HAL
open science

Un modèle générique pour les organisations dynamiques en univers multi-agent

Laurent Lacomme

To cite this version:

Laurent Lacomme. Un modèle générique pour les organisations dynamiques en univers multi-agent. Système multi-agents [cs.MA]. Université de Grenoble, 2011. Français. NNT : 2011GRENM067 . tel-01328138

HAL Id: tel-01328138

https://theses.hal.science/tel-01328138v1

Submitted on 27 Jun 2016

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Laurent LACOMME

Thèse dirigée par **Yves DEMAZEAU** et
codirigée par **Valérie CAMPS**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale MSTII**

Un modèle générique pour les organisations dynamiques en univers multi-agent

Thèse soutenue publiquement le **13 décembre 2011**,
devant le jury composé de :

M. Olivier BOISSIER
Professeur à l'École des Mines de Saint-Etienne, Rapporteur
Mme Valérie CAMPS
Maître de conférences à l'UPS Toulouse III, Co-Directrice
M. Yves DEMAZEAU
Directeur de Recherches au CNRS, Grenoble, Directeur
M. René MANDIAU
Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, Examinateur
M. Juan PAVÓN
Professeur à l'Universidad Complutense Madrid, Rapporteur
M. Patrick REIGNIER
Professeur à Grenoble-INP, Président

Résumé

Les systèmes multi-agents (SMA) mettent en relations des entités autonomes, les agents, dont les actes et les interactions participent à la mise en place d'un comportement et d'une fonctionnalité globale du système. La structure et le fonctionnement global du système sont décrits à travers la notion d'organisation : celle-ci regroupe l'ensemble des places occupées par les agents, de leurs relations et de leur importance dans la fonctionnalité attendue du système. Usuellement, ces concepts sont formalisés par des notions empruntées à l'analyse des organisations humaines et animales : rôles, groupes, normes, etc. Cependant, une part importante des organisations des SMA n'est que partiellement décrite par ces notions : la dynamique. En effet, dans les SMA ouverts, dans lesquels des agents, parfois inconnus à la conception du système, peuvent entrer ou sortir de celui-ci à tout instant, et dans les SMA auto-organisés, où la structure du système se forme et se reforme en fonction du contexte, la dynamique des organisations, c'est-à-dire l'analyse de leur évolution au cours du temps, est un champ important qui est difficile à décrire avec les notions de haut-niveau usuellement utilisées pour formaliser les organisations.

Ce que nous proposons dans cette thèse est donc un modèle de description des organisations pour les SMA, axé sur la possibilité de décrire une vaste variété de système, à la fois dans leurs aspects statiques et dynamiques. Pour atteindre cela, nous nous appuyons sur une conception émergentiste et calculatoire, et nous créons notre modèle autour de trois éléments typés et de bas-niveau : les agents, les relations et les tâches. Nous proposons ensuite des méthodes descriptives des organisations fondée sur l'agrégation de contraintes sur les systèmes exprimées à partir de ces éléments, ainsi que des méthodes calculatoires destinées à permettre l'extraction de propriétés globales sur les organisations à partir de ces descriptions ; tout cela dans l'optique de fournir une aide méthodologique à la conception et à l'analyse d'organisations pour les SMA. Nous appliquons ensuite l'utilisation de notre modèle sur des applications précises afin de démontrer son intérêt et sa pratique dans la formalisation et dans la comparaison d'organisations dynamiques pour les SMA.

Mots clés :

intelligence artificielle ; intelligence artificielle distribuée ; systèmes multi-agents ; organisations ; dynamique ; auto-organisation ; émergence ;

Abstract

Multiagent systems (MAS) are composed of interacting autonomous entities called agents. Their behaviors and interactions take part in the generation of a global functionality in the system. The notion of organization describes the structural and functional aspects of such systems: it includes the objectives of each agent, the way they can interact and create relationships and their importance in the system's global function. These concepts are usually formalized through notions derived from human and animal organizations: roles, groups, norms... However, an important part of MAS' organization can only be partially described with these notions: dynamics. In open MAS – where agents can enter or exit the system at any time, and where their number and characteristics are often not known at the time of the system's design – and in self-organized MAS – where the system's structure evolves with relation to context and environment – organization's dynamics is very difficult to formally describe with these high-level notions.

In this thesis, we propose a model for MAS organizations' description, which is focused toward the description of a wide range of systems and the modeling of both their static and dynamic aspects. In order to achieve this, we ground our model on an approach based on emergence and computation. We then create a model based on three basic, low-level, typed static elements: agents, relations and tasks, and one low-level dynamic element: organizational mechanisms. We then propose some methods for organization description based on our model and the principle of system's constraints aggregation. We also provide some computational methods allowing the calculation of some global properties over described organizations, in order to provide a methodological help for MAS design and analysis. We then apply our model and the proposed methods on practical applications in order to show its pertinence in MAS organizations' formalization and comparison.

Keywords:

artificial intelligence; distributed artificial intelligence; multiagent systems; organizations; dynamics; self-organization; emergence;

Publications

Laurent Lacomme. *Vers un modèle générique pour les organisations dynamiques en univers multi-agent.* Atelier FUTURAMA : le Futur des Agents et des Multi-Agents, plateforme AFIA 2011, mai 2011.

Laurent Lacomme, Valérie Camps, Yves Demazeau, Florent Hautefeuille, Bertrand Jouve. *Middle Age Social Networks: a dynamic organizational study.* Advances in Practical Applications of Agents and Multiagent Systems, 9th International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS'1A), Advances in Soft Computing, apr 2011.

Laurent Lacomme, Yves Demazeau, Julie Dugdale. *CLIC: An Agent-Based Interactive and Autonomous Piece of Art.* Advances in Practical Applications of Agents and Multiagent Systems, 8th International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS'10), Yves Demazeau, Frank Dignum, J.-M. Corchado, Javier Bajo eds. 70:25-34, Advances in Soft Computing, apr 2010.

Laurent Lacomme, Yves Demazeau, Valérie Camps. *Classification des mécanismes organisationnels dans les réseaux d'agents.* 17ème Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09) Zahia Guessoum, Salima Hassas eds. Lyon, France, oct 2009.

Laurent Lacomme, Yves Demazeau, Valérie Camps. *Personalization of a Trust Network.* Proceedings of the 1st International Conference on Agents and Artificial Intelligence (ICAART'09), 408-415, Porto, Portugal, jan 2009.

“That orbed maiden with white fire laden,
Whom mortals call the Moon,
Glides glimmering o'er my fleece-like floor,
By the midnight breezes strewn;”

Percy Bysshe Shelley – The Cloud

Remerciements

Je tiens ici à remercier toutes les personnes qui m'ont accompagné et soutenu durant ces trois années. Tout d'abord, je remercie Yves Demazeau de m'avoir donné l'envie, par ses cours et sa passion, de m'intéresser et de m'investir dans le domaine du multi-agent ; je le remercie également de son soutien, de sa disponibilité, et de ses conseils avisés tout au long de mon doctorat. J'espère que nous serons amenés à travailler à nouveau ensemble. Je remercie également Valérie Camps pour avoir toujours, malgré la distance et mon indépendance naturelle, gardé un œil sur mes avancées ; je la remercie grandement pour ses conseils et ses remarques, qui m'ont également beaucoup apporté tout au long de mon travail. J'espère aussi que nous pourrons travailler à nouveau ensemble un jour.

Je remercie évidemment l'équipe MAGMA du LIG, pour son accueil, les discussions et les échanges ; je remercie pour cela les permanents, les doctorants et les multiples secrétaires, sans qui les obligations administratives eussent été des tortures ! Un grand merci également à l'équipe SMAC de l'IRIT pour son accueil chaleureux et bienveillant à chacune de mes visites. Je remercie de même les différents partenaires des projets menés dans cette thèse, pour leurs apports respectifs à mon travail et à ses applications.

Je remercie mes parents pour leur soutien indéfectible malgré le peu que je laisse à voir de mon travail, et surtout pour m'avoir toujours poussé dans la direction où je souhaitais aller. Je tiens à présenter par ailleurs un remerciement spécial à M. Roussel, un professeur de mathématiques grâce auquel je me suis retrouvé sur la voie qui m'a conduit à cette thèse.

Je remercie Baptiste, a.k.a A.K.A, pour tous les échanges que nous avons eus et pour ceux que nous aurons, pour son amitié et son soutien, et surtout pour le graiiil ! Je remercie Yann, avec qui j'ai pu décompresser dans les moments difficiles, rire et délirer dans – tous – les moments et partager nos expériences respectives. Merci Shangy ! Merci aussi à Sir Bedevere l'Équarisseur de Nains (alias Yves), avec qui j'ai partagé des pensées, des fous rires, de nombreux moments fort sympathiques et surtout DE L'EAU !

Et enfin le plus grand des mercis à Céline, qui m'a accompagné durant ces trois années et a partagé avec moi mes joies, mes difficultés, mes avancées et mes idées farfelues... bref, ma vie. Merci à toi pour ta gentillesse, ta bonté et tous les échanges que nous avons eus.

Et à tous : attention aux flèches dans le genou !

Sommaire

Sommaire 13

Figures 17

Tableaux 17

Liste des notations 19

Chapitre 1. Introduction 23

1.1 Le multi-agent dans l'informatique 23

1.2 Problématique et intérêts 24

1.2.1 Notion d'organisation 24

1.2.2 La problématique et son importance pour le domaine 25

1.3 Objectifs 27

1.3.1 Un modèle organisationnel à fort pouvoir expressif 27

1.3.2 Un modèle formel de description des organisations 28

1.3.3 Un modèle adapté au calcul et à l'évaluation de propriétés 28

1.4 Méthode suivie 29

1.4.1 Fondements méthodologiques 29

1.4.2 Approche du problème 30

1.5 Plan du mémoire 32

Chapitre 2. État de l'art 35

2.1 Méthodologie de conception des SMA 35

2.1.1 Critères d'analyse 36

2.1.2 Méthodes de conception des SMA 36

2.1.3 Conclusion sur les méthodes 41

2.2 Modèles organisationnels 42

2.2.1 Critères d'étude 42

2.2.2 Modèles organisationnels 43

2.2.3 Synthèse sur les modèles organisationnels 48
2.3 Auto-organisation et réorganisation dans les SMA 50
2.3.1 Critères de classification des mécanismes 52
2.3.2 Mécanismes structuraux sociaux 54
2.3.3 Mécanismes structuraux situationnels 57
2.3.4 Mécanismes fonctionnels de délégation 58
2.3.5 Mécanismes fonctionnels de répartition 60
2.3.6 Synthèse sur les mécanismes organisationnels 62
2.4 Lacunes et problématiques de l'état de l'art 63
Chapitre 3. Notre approche 65
3.1 Une approche émergentiste 65
3.1.1 L'émergence 66
3.1.2 Les niveaux de représentation 68
3.1.3 Une construction bottom-up 70
3.1.4 L'indépendance de la description organisationnelle 72
3.2 Une approche calculatoire 74
3.2.1 Évaluer et calculer 74
3.2.2 Peut-on calculer l'émergence ? 75
3.2.3 Cas-test et probabilités 76
3.2.4 Une approche calculatoire est possible 77
Chapitre 4. Modèle organisationnel 79
4.1 Éléments du modèle 79
4.1.1 Agents 80
4.1.2 Relations 81
4.1.3 Tâches 83
4.1.4 Mécanismes organisationnels 84
4.2 Description des mécanismes 85
4.2.1 Mécanismes structuraux 86

4.2.2 Mécanismes fonctionnels 88
4.2.3 Définition du pas de temps 92
4.2.4 Identification de l'émergence 93
4.3 Description des organisations 94
4.3.1 Le principe des réservoirs 95
4.3.2 Spécifications à partir des réservoirs 102
4.3.3 Définitions formelles des notions globales 105
4.4 Possibilités expressives du modèle 110
4.4.1 Rôles 110
4.4.2 Groupes 113
4.4.3 Coalitions, équipes 114
4.4.4 Normes 116
Chapitre 5. Méthodes calculatoires pour les organisations 121
5.1 Représentation dynamique dans le modèle 121
5.2 Approche probabiliste du calcul de propriétés 123
5.2.1 Justification de l'approche probabiliste 124
5.2.2 Utilisation des éléments du modèle 126
5.2.3 Objectifs des calculs et représentation markovienne 127
5.3 Méthode de description des mécanismes 129
5.3.1 Représentation par chaînes de Markov 129
5.3.2 Méthodes de description selon les types de mécanismes 133
5.3.3 Agrégation de mécanismes et rétroactions 136
5.4 Calcul de propriétés dynamiques 138
5.4.1 Définition et intérêt des propriétés globales 139
5.4.2 Validité du calcul des propriétés globales 142
5.4.3 Intérêt et utilisation de cette solution 143
Chapitre 6. Valeur méthodologique 145
6.1 Méthodologie SMA orientée organisations dynamiques 145

6.2 Correspondance en POO 161
6.3 Utilisation pratique des spécifications de l'organisation 165
6.3.1 Génération semi-automatique à partir des spécifications 165
6.3.2 Intégration à un modèle général d'un SMA 167
6.4 Utilisation des évaluations de dynamique 168
Chapitre 7. Applications 171
7.1 Conception de système : Projet CLIC 171
7.1.1 Contexte applicatif 172
7.1.2 Contraintes logicielles 173
7.1.3 Modélisation organisationnelle 176
7.1.4 Exploitation du logiciel et exposition réalisée 184
7.2 Analyse de système : Projet MASNET 187
7.2.1 Contexte applicatif 188
7.2.2 Approche 190
7.2.3 Modélisation organisationnelle 192
7.2.4 Résultats et perspectives 196
Chapitre 8. Conclusion 199
8.1 Modèle théorique 199
8.2 Méthodes et applications 201
8.2.1 Conception de système 201
8.2.2 Analyse de systèmes 202
8.3 Perspectives 203
Références 205

Figures

Figure 2.1 - Classification des mécanismes organisationnels 54

Figure 3.1 - Représentation de l'émergence et des niveaux de description 69

Figure 3.2 - Représentation du principe d'indépendance organisation - population..... 73

Figure 4.1 - Action des mécanismes structuraux 87

Figure 4.2 - Actions des mécanismes fonctionnels 90

Figure 4.3 - Éléments constitutifs du modèle 94

Figure 4.4 - Composantes de l'organisation d'un SMA S(t) dans le modèle proposé 102

Figure 5.1 - Représentation d'un exemple de chaîne de Markov à travers ses états et probabilités de transition ainsi que sa matrice de transition associée 128

Figure 5.2 - La modélisation markovienne implique la transformation de l'espace d'états du mécanisme vers un espace d'état de chaîne de Markov..... 130

Figure 6.1 - Décomposition des objectifs en tâches et définition des rôles effectifs 149

Figure 6.2 - Rôles normatifs et types des relations associées 152

Figure 6.3 - Description UML des équivalences aux notions organisationnelles 165

Figure 7.1 - L'un des affichages de l'installation autonome et interactive CLIC 171

Figure 7.2 - Architecture et intégration du SMA de contrôle de l'installation 175

Figure 7.3 - Décomposition des objectifs en tâches et définition des rôles effectifs et normatifs pour le projet CLIC..... 178

Figure 7.4 - Rôles normatifs et types de relations associées pour le projet CLIC 181

Figure 7.5 - Images de l'installation - matériel utilisé 185

Figure 7.6 - Affichage de la séquence 4 du script..... 185

Figure 7.7 - Images de l'installation en activité 187

Figure 7.8 - Photographie d'un acte notarié des archives utilisées dans le projet 188

Figure 7.9 - Modèle de la base de données utilisée dans les projets GRAPHCOMP et MASNET 189

Figure 7.10 - Représentation du réseau social induit par extraction de la base de données à une date choisie..... 190

Figure 7.11 - Correspondance entre notions organisationnelles et éléments du réseau social étudié 192

Figure 7.12 - Instanciation du modèle organisationnel à l'analyse du réseau social du projet MASNET 196

Tableaux

Tableau 2-A - Caractéristiques des méthodes de conception des SMA 41

Tableau 2-B - Caractéristiques des modèles organisationnels 50

Tableau 2-C - Exemples classifiés de mécanismes organisationnels 62

Tableau 5-A - Algorithme itératif d'estimation des distributions d'états 136

Tableau 6-A - Étapes de conception 146

Tableau 6-B - Correspondance entre notions organisationnelles et concepts de POO 162

Liste des notations

notion	notation	page
agent	$a = (n_a, \alpha), n_a \in \mathrm{N}$	81
agent du système à un instant donné	$A(t)$	96
agents du système	A	96
autorité normative	$A\mathcal{N} \subseteq \mathrm{A}$	118
champ d'action	$P_a(t) \in \{ x = (B, j, \phi), j \in N\}$	84
champ de coopération	$C_a(t) = \{ x \mid x \in E_a(t) \wedge x \in P_a(t) \} = \bigcup_{a' \in A(t)} C_a^{a'}(t)$	100
champ de coopération avec un agent	$C_a^{a'}(t) = \{ x \mid (x, a) \in D_{a'}(t) \wedge x \in P_a(t) \}$	100
champ de non-coopération	$NC_a(t) = \{ x \mid x \in E_a(t) \wedge x \notin P_a(t) \} = \bigcup_{a' \in A(t)} NC_a^{a'}(t)$	100
champ de non-coopération avec un agent	$NC_a^{a'}(t) = \{ x \mid (x, a) \in D_{a'}(t) \wedge x \notin P_a(t) \}$	100
champ extérieur	$E_a(t) = \{ x \mid (x, a) \in D_{a'}(t) \wedge a' \in A(t) \}$	100
champ relationnel	$R_a(t) \in \{r = (a, a_c, \lambda)\}$	82
coalition	$\mathcal{C}_C^{t_1,t_2} = \left(A_C, P_C^{(t_1,t_2)}\right) = \left(A_C, \bigcap_{t \in [t_1,t_2[} P_C(t)\right)$	115
coalition à un instant donné	$\mathcal{C}_C(t) = \left(A_C(t), P_C(t)\right)$	115
composition des mécanismes	$m^2 \circ m^1$	86
connectivité	$c_\alpha^h = \sum_{\beta \in h} c_\alpha^\beta$	140
connectivité pour un type d'agents	$c_\alpha^\beta = \left(\overrightarrow{w_\alpha^S} \middle\vert \overrightarrow{V_\alpha^\beta}\right)$	140
dérivation des organisations	$O_1 \vDash O_2$	107
distribution stationnaire	$\overrightarrow{w_\alpha^S}$	139
ensemble d'états de la modélisation markovienne	$\mathcal{W}_\alpha$	131
ensemble des relations du système	$R(t) = \bigcup_{a \in A(t)} R_a(t)$	98
ensemble des tâches du système à un instant	$P(t) = \bigcup_{a \in A(t)} P_a(t)$	99
équivalence des organisations	$O_1 \equiv O_2$	107
état de la modélisation markovienne	$w_a(t) \in \mathcal{W}_\alpha$	130
états modifiés par un mécanisme	$\mathcal{E}_a(t)$	130
états non modifiés par un mécanisme	$\mathcal{Y}_a(t)$	130

extension sociale	$D_a(t) \subseteq \{ (x, a_c) \mid a_c \in V_a(t) \}$	89
graphe structurel	$\mathbb{G}(t) = < A(t), R(t) >$	98
groupe	$\forall a, b \in G, \exists r_1, r_2 \in R(t) \mid r_1 = (a, b, \lambda_1) \wedge r_2 = (b, a, \lambda_2) \wedge \lambda_1, \lambda_2 \in T_R$	114
mécanisme d'insertion	$\forall R_a(t) \neq \emptyset, s_a\big(R_a(t), P_a(t), \xi_a(t)\big) = R_a(t)$	88
mécanisme fonctionnel	$f_a : \big(D_a(t), R_a(t+\Delta t), P_a(t), \xi_a(t)\big) \longmapsto D_a(t+\Delta t)$	90
mécanisme fonctionnel négligent	$f_a : \big(R_a(t+\Delta t), P_a(t), \xi_a(t)\big) \longmapsto D_a(t+\Delta t)$	90
mécanisme structural	$s_a : \big(R_a(t), P_a(t), \xi_a(t)\big) \longmapsto R_a(t+\Delta t)$	87
mécanisme structural homogène	$s_a : \big(R_a(t), P_a(t)\big) \longmapsto R_a(t+\Delta t)$	87
mécanisme structural libre	$s_a : R_a(t) \longmapsto R_a(t+\Delta t)$	88
mécanisme structural réactif	$s_a : \big(R_a(t), \xi_a(t)\big) \longmapsto R_a(t+\Delta t)$	87
modélisation markovienne d'un mécanisme	$\mathcal{M} : M \longmapsto \mathcal{M}(M)$	130
norme autoritaire	$\mathcal{N}^A \subseteq \mathcal{G}M$	118
norme relationnelle	$\mathcal{N}^R \subseteq \mathcal{G}M$	118
période d'existence	$L(a) = \{ t \in R^+ \mid a \in A(t) \}$	97
précision	$\mathfrak{p}(M, M, \alpha)$	142
relation	$r = (a_s, a_c, \lambda)$	82
réservoir d'action global	$\mathcal{G}P = \bigcup_{\alpha \in T_A} \mathcal{S}P(\alpha) = \{ x \mid \exists t \in \mathrm{R}^+, \exists a \in A(t), x \in P_a(t) \}$	100
réservoir d'action spécifique	$\mathcal{S}P(\alpha) = \{ x \mid \exists t \in R^+, \exists a \in A(t), t(a) = \alpha, x \in P_a(t) \}$	100
réservoir fonctionnel global	$\mathcal{G}F = \bigcup_{\alpha \in T_A} \mathcal{S}F(\alpha)$	101
réservoir fonctionnel spécifique	$\mathcal{S}F(\alpha) \subseteq \mathcal{S}M(\alpha)$	91
réservoir organisationnel global	$\mathcal{G}M = \mathcal{G}S \cup \mathcal{G}F = \bigcup_{\alpha \in T_A} \mathcal{S}M(\alpha)$	101
réservoir organisationnel spécifique	$\mathcal{S}M(\alpha)$	85
réservoir structurel global	$\mathcal{G}S = \bigcup_{\alpha \in T_A} \mathcal{S}S(\alpha)$	101
réservoir structurel spécifique	$\mathcal{S}S(\alpha) \subseteq \mathcal{S}M(\alpha)$	88
rôle effectif	$q \subseteq P(GP)$	111
rôle normatif	$h \subseteq T_A$	113
spécification	$\Theta \equiv \forall t \in I^T, C(t)$	103
spécificité	$\mathfrak{s}(M, M, \alpha)$	143
tâche	$x = (B_x, j, \phi), j \in N$	84
tâches d'une coalition	$P_C(t) = \bigcap_{a \in A_C} P_a(t) \neq \emptyset$	115
température du système pour un type d'agent	$\mathbb{T}_\alpha = \frac{1}{\dim\left(\overrightarrow{V_\alpha^R}\right)} \left\lVert \overrightarrow{V_\alpha^R} \right\rVert_1$	141
transformation des états	$T : \big(\mathcal{Y}_a(t), \mathcal{E}_a(t)\big) \longmapsto w_a(t)$	131

type	$\mathfrak{t}(e)$	97
types d'agents	T_A	96
types de relations	T_R	98
types de tâches	T_P	99
vecteur fonctionnel	$\overrightarrow{D_\alpha} = \overrightarrow{w_\alpha^S} \times \begin{pmatrix} \overrightarrow{D_{\alpha,w_1}} \\ \vdots \\ \overrightarrow{D_{\alpha,w_N}} \end{pmatrix}$	142
voisinage	$V_a(t) = \{\ a' \mid \exists r \in R_a(t) \wedge\ r = (a, a', \lambda)\}$	82

Chapitre 1. Introduction

1.1 Le multi-agent dans l'informatique

Le **multi-agent** est un domaine à part de l'**intelligence artificielle** et plus généralement de l'informatique. Ce domaine réunit en effet en un même paradigme des vues diverses et parfois divergentes, s'attachant au génie logiciel, à l'intelligence artificielle, à la robotique, aux systèmes distribués et trouvant des applications dans de nombreux domaines, du web sémantique à l'informatique ubiquitaire.

L'idée que nous retenons comme fondamentale de ce domaine est celle de l'**émergence** (Deguet *et al.*, 2006). Elle se présente comme suit :

> Il existe des systèmes étudiables dont la fonction globale est supérieure à la somme des fonctions des parties qui composent ces systèmes.

L'émergence décrit donc le phénomène qui, au niveau de l'observation, permet de passer du niveau local des comportements individuels au niveau global de la fonction du système. Elle définit un gain de complexité dans l'observable suivant les niveaux d'observation.

> *L'analogie première du domaine est celle de la colonie de fourmis : bien que le comportement de chaque fourmi soit extrêmement basique (et donc simple à décrire), la complexité engendrée par les comportements cumulés de toutes les fourmis n'est pas immédiatement déductible de ces comportements individuels. Cette complexité peut prendre l'aspect de la formation de pistes menant aux sources de nourriture, mais aussi de la construction, sur le long terme, d'une fourmilière complexe et fonctionnelle.*

L'émergence apparait donc comme un moyen, pour le programmeur, d'obtenir des programmes complexes et riches en économisant sur la complexité du code ; pour cela, il lui faut déterminer des comportements locaux qui, agrégés, mènent à l'émergence d'un comportement global souhaité.

Le domaine du multi-agent a ainsi trouvé de nombreux domaines d'application, en dehors même de son objectif premier : la simulation de l'*intelligence collective.* Ainsi, la simulation, notamment la simulation sociale ou du comportement animal, a été un champ d'application important pour le domaine. Par ailleurs, ce paradigme est naturellement adapté à la conception de systèmes distribués et modulaires, puisque le programme y est considéré comme l'assemblage, en parallèle, du code de chacun des **agents**, c'est-à-dire d'entités distinctes, séparées et indépendantes. Ceci a ainsi engendré de nombreux autres champs d'applications, dans des domaines tels que les architectures orientées service (SOA) (Erl,

2005), l'informatique ambiante (Huhns *et al.*, 2005) et, par exemple, l'entreprise ouverte (Papazoglou, 2001) et le Web sémantique (Hendler, 2001).

1.2 Problématique et intérêts

1.2.1 Notion d'organisation

Pour décrire un système fondé sur ce paradigme, que nous nommons **système multi-agent (SMA)**, il peut être considéré de prime abord que la description des agents eux-mêmes est suffisante. Les agents sont les entités actives du système : ce sont les programmes, individuels, qui s'exécutent en parallèle et communiquent pour former un système entier.

> Un agent est une entité réelle ou virtuelle, plongée dans un environnement, capable de percevoir cet environnement, d'agir dans cet environnement, d'interagir avec les autres agents, et qui manifeste un comportement autonome. (Demazeau et Müller, 1990a)

Il paraît donc également nécessaire, au vu de la définition d'un agent, de décrire l'**environnement** au sein duquel les agents évoluent, qu'il soit physique ou simulé, afin d'être à même de comprendre les fonctionnalités et les propriétés du système. A la description des agents – dits **situés** lorsque l'environnement au sein duquel ils se trouvent est topologique –, nous ajoutons donc la description de leur environnement.

Cependant, il est apparu que ces descriptions ne sont pas toujours suffisantes pour décrire et comprendre un système. Le principe d'émergence est en effet au cœur du paradigme multi-agent. Il en découle un problème qui s'est logiquement posé au cours du développement et de l'expansion du domaine : le comportement global du système n'étant pas une sommation directe des comportements locaux des entités du système – les agents –, il apparait que d'autres éléments entrent en jeu dans la détermination de la fonctionnalité globale du système. Ces éléments sont les **interactions** entre ces agents. L'étude des interactions est donc une étape importante dans la compréhension des SMA.

> *Pour poursuivre l'analogie présentée ci-dessus, quand les comportements des fourmis peuvent être décrits en termes algorithmiques tels que « explore le terrain tant que tu n'as pas trouvé de nourriture » ou « retourne à la fourmilière avec la nourriture », les interactions résultent, quant à elles, des échanges entre les individus. Dans le cas des fourmis, ces interactions sont principalement indirectes, résultant du dépôt de phéromones dans l'environnement et de la perception de ces phéromones par d'autres fourmis, résultant en des modifications de leur comportement (par exemple de la trajectoire suivie).*

Dans des systèmes simples, comme dans l'exemple des fourmis présenté ci-dessus, il est possible de décrire les interactions possibles entre les différents agents (ou individus) de manière simple et générale, car les agents ont des comportements identiques les uns avec les autres : on dit qu'ils sont **homogènes** (*i.e.* construits sur le même modèle). Cela n'est pas toujours le cas. En effet, dans de nombreux systèmes, tels, par exemple, que les SOA ou le Web sémantique, les agents sont **hétérogènes** (*i.e.* construits sur des modèles différents) : ils n'ont pas tous les mêmes comportements, mais surtout, ils ne possèdent pas les mêmes capacités, les mêmes objectifs, ni les mêmes fonctions au sein du système. Pour être capables de décrire un SMA dans son ensemble, et d'en comprendre le fonctionnement, la description locale des comportements et des interactions peut, dès lors, ne plus suffire. Par analogie avec les communautés humaines ou animales souvent modélisées par ces systèmes, c'est la notion d'**organisation** qui s'est révélée être à même de permettre une description plus globale et plus compréhensible des systèmes étudiés, en complément des descriptions locales (Costa et Demazeau, 1996).

Ce que nous définissons comme le concept d'organisation, est la description, éventuellement partielle, et située à un niveau global, c'est-à-dire du point de vue du concepteur ou de l'observateur (et non du programmeur), de la manière dont est organisée la fonctionnalité du système et de comment celle-ci est distribuée entre les différents éléments et leurs différentes interactions. Pour cela, pour décrire l'organisation d'un SMA, il faut spécifier de quelle manière les agents sont supposés se comporter : quels objectifs locaux ils doivent atteindre, quelles règles comportementales ils doivent respecter. Mais il faut également spécifier de quelle manière les interactions entre ces agents peuvent ou doivent avoir lieu. Ainsi, nous définissons l'organisation de la manière suivante :

> L'organisation est une spécification au niveau des fonctionnalités et des comportements des agents, des possibilités d'interactions entre agents et des méthodes et objectifs de ces interactions.

1.2.2 La problématique et son importance pour le domaine

Avec l'évolution du domaine du multi-agent et l'ouverture à des champs d'application de plus en plus nombreux, l'ingénierie des SMA s'est faite de plus en plus complexe. Les systèmes peuvent être très variés, allant de l'exemple de la colonie de fourmis, ou les agents sont homogènes et réactifs – ils réagissent algorithmiquement aux stimuli extérieurs – et le système bien délimité à l'instant de sa conception, à l'exemple de l'entreprise ouverte, ou des agents cognitifs hétérogènes échangent entre eux, et où de nouveaux agents et de nouveaux types d'agents sont ajoutés à chaque instant au système.

La description des SMA, tant au niveau de leur conception que de l'analyse de systèmes existants, se fait à plusieurs niveaux : les concepteurs d'un système peuvent vouloir spécifier indépendamment l'organisation d'un système, les méthodes d'interaction entre les agents et l'ensemble de tous les agents possibles et de leurs structures internes. En effet, chacun de ces éléments peut être connu à l'avance à la conception ou bien défini incrémentalement lors de l'utilisation du système. De ce fait, de nombreux modèles ont été développés pour décrire les organisations des SMA. Fondés sur des notions « sociales » telles que les rôles, les groupes, les équipes, ils cherchent à permettre la description précise de systèmes à un niveau global, donnant sens à l'agencement des agents entre eux et à leurs interactions possibles.

Par ailleurs, les SMA, au vu de la place importante qu'y jouent l'interaction et l'échange, sont naturellement des systèmes à forte dynamique. On retrouve cette place de la dynamique dans de nombreux systèmes et dans les applications qui y sont associées : simulations sociales, agents assistants, réseaux sociaux,... Cette dynamique s'intègre à la description des systèmes au niveau des agents, par l'apprentissage et l'adaptation à l'environnement, au niveau des interactions, par les échanges d'informations et d'activités entre agents, et au niveau de l'organisation par la formation de liens « sociaux » entre agents et l'évolution des activités de chaque agent au sein du système. Ceci a donné lieu à de nombreux travaux sur les méthodes employées par les agents pour s'organiser au sein du système, que ce soit dans des buts collaboratifs – tels que les systèmes multi-agents adaptatifs (AMAS) (Gleizes *et al.*, 2007) – ou dans des buts d'optimisation individuelles – tels que les processus de confiance (Castelfranchi et Falcone, 1998).

En revanche, tandis que ces travaux apparaissent comme des descriptions ad hoc de modèles d'auto-organisation ou de réorganisation pour des systèmes spécifiques, ou tout au moins vis-à-vis de problèmes spécifiques, il apparait, comme le montre l'état de l'art (cf. section 2.4), que les modèles descriptifs des organisations ne sont pas adaptés à la description de la dynamique organisationnelle des systèmes. En effet, ces modèles sont généralement orientés vers la description des aspects statiques et figés à la conception des organisations, et sont donc peu adaptés à la description de **systèmes ouverts** (*i.e.* dans lesquels les agents entrent ou sortent du système à loisir) ou partiellement spécifiés (comme dans le cas où des agents au comportement inconnu peuvent entrer dans le système). Nous cherchons donc ici à déterminer une manière simple de décrire précisément et formellement la dynamique des organisations dans les SMA. Nous souhaitons en cela être capables de spécifier des propriétés globales des systèmes, leurs évolutions et leurs invariants.

1.3 Objectifs

La problématique que nous posons est donc celle de la **formalisation des propriétés organisationnelles statiques et dynamiques des SMA** à travers un modèle unique et générique, applicable à tous types de systèmes, totalement ou partiellement spécifiés.

1.3.1 Un modèle organisationnel à fort pouvoir expressif

Le but poursuivi est ainsi de définir un modèle[1] de description des organisations. Ce modèle doit permettre de décrire de manière générique tout type d'organisation, et donc de spécifier au niveau organisationnel quels sont les points connus du système et quelles sont ses inconnues. Il s'agira de définir les propriétés spécifiées du système ainsi que d'énoncer l'ensemble des possibles pour les propriétés non totalement spécifiées.

> *Par exemple, pour le cas de la colonie de fourmis, le nombre maximal d'agents et leurs comportements possibles peuvent être connus, tandis que le nombre d'agents à un instant donné et les comportements adoptés par les agents à cet instant là seraient inconnus a priori.*

Par ailleurs, ce modèle, en plus de décrire les invariants des organisations, doit permettre de décrire les mécanismes, les méthodes, qui définissent sa dynamique, c'est-à-dire la manière dont les agents réarrangent au cours du temps la connaissance qu'ils ont du système, leurs interactions avec les autres, leurs échanges, et les fonctions qu'ils remplissent au sein du système.

> *Toujours pour l'exemple de la colonie de fourmis, il s'agira de décrire quels sont les « rôles » sociaux joués par les fourmis au cours du temps, et comment le comportement des fourmis et leurs échanges évoluent au cours du temps et influent sur le système dans son ensemble.*

Il s'agit là d'un problème à la fois local, de part le fait que ce sont les agents eux-mêmes qui sont les acteurs du système, et donc prennent les décisions et agissent en conséquence, mais également global, car c'est en relation avec le fonctionnement du système dans son ensemble que la description doit être faite.

---

[1] Il s'agit ici, plus justement, d'un **métamodèle** d'organisations, dans le sens où c'est le modèle d'un langage de description des organisations ; il permet donc de décrire une manière de modéliser les organisations des SMA. Pour des raisons évidentes de simplicité, nous parlerons ici de modèle organisationnel.

L'objectif premier est donc de **définir un modèle à fort pouvoir expressif**, permettant de décrire au mieux, et de manière générique, l'organisation d'un SMA et sa dynamique.

Nous évaluons le pouvoir expressif des modèles organisationnels par rapports à deux critères : la variété de systèmes que le modèle permet de décrire, et la précision de la description pour chacun de ces systèmes – c'est-à-dire le nombre de notions organisationnelles représentables. Le but de ce modèle est d'améliorer la variété des systèmes modélisables par rapport aux modèles organisationnels existants tout en maintenant une précision descriptive équivalente à ces derniers pour les systèmes modélisables par ceux-ci (cf. section 2.2).

1.3.2 Un modèle formel de description des organisations

La description des organisations sert deux buts distincts : la **conception de SMA** et l'**analyse de systèmes** existants. Dans les deux cas, un besoin commun de formalisme et de précision s'impose. Dans le premier cas, ce besoin de formalisme provient de la nécessité d'être à même de relier la description qui peut être faite des SMA avec des contraintes et des connaissances sur le futur code des agents, et plus généralement, de l'ensemble du système, afin d'être capable de concrétiser facilement les assertions sur les notions organisationnelles en les transposant directement vers la programmation du système. Dans le second cas, ce besoin de formalisme provient du besoin d'être capable de comparer, de la manière la plus simple et la plus complète possible, différents systèmes. En effet, la comparaison et l'évaluation des SMA dans un contexte donné n'est pas une activité aisée, et l'analyse des systèmes en est donc rendue complexe.

Ainsi, le deuxième objectif est d'**obtenir un formalisme suffisant** pour notre modèle pour garantir la possibilité de comparer les organisations entre elles sur des propriétés précises, et de relier les spécifications organisationnelles à des contraintes et à des modèles concrets lors de la conception de systèmes.

1.3.3 Un modèle adapté au calcul et à l'évaluation de propriétés

Enfin, l'étude et l'ingénierie des SMA requièrent des outils pour la comparaison, l'analyse et l'évaluation des SMA. Dans le but de procurer une base d'outils visant à préciser et à évaluer l'adéquation des organisations avec des applications ou à des contextes donnés, nous visons à fournir, avec notre modèle, des moyens de calculer, évaluer ou prédire des propriétés globales des systèmes reposant sur une organisation donnée. Pour cela, nous procédons également à l'étude de certains exemples de méthodes permettant de déterminer, à partir d'une description d'une organisation donnée, des propriétés, variantes ou invariantes, du système,

dans des cas généraux (si cela est possible) ou dans des cas particuliers (*i.e.* pour une population donnée ou dans un contexte fixé).

Ainsi, le troisième objectif est de **rendre notre modèle calculable**, afin de lui permettre d'exprimer des propriétés sur les SMA résultants d'une organisation donnée, et d'extrapoler des informations dynamiques à partir de la description d'une organisation. En cela, nous requérons qu'il soit possible d'effectuer au sein du modèle des calculs, probabilistes ou exacts, sur les organisations elles-mêmes, *a priori* de toute implémentation du système et en vue d'obtenir des informations sur les invariants ou la dynamique des systèmes régis par les organisations étudiées.

1.4 Méthode suivie

1.4.1 Fondements méthodologiques

Pour cerner notre sujet, nous appuyons notre travail sur deux considérations principales :

- la description des SMA suivant les dimensions proposées par la méthode VOYELLES (Demazeau, 1996) ;
- l'indépendance de la description organisationnelle vis-à-vis de la description des agents eux-mêmes, proposée par le modèle PopOrg (Demazeau et Costa, 1996).

La première considération sur laquelle se fonde notre approche est donc la séparation entre les différents niveaux de description d'un SMA proposée dans le paradigme VOYELLES. Ce paradigme définit en effet 4 niveaux de description des SMA : les agents (A), s'attachant aux modèles internes de comportement des entités actives du système, l'environnement (E), décrivant les entités passives du système et la situation des agents, les interactions (I), s'attachant aux canaux et aux protocoles de communication entre agents et l'organisation (O), décrivant la structuration du système, les relations entre agents et leurs évolutions. Nous suivons cette séparation afin de poser un cadre à la description des SMA permettant de rechercher de manière indépendante pour chaque dimension, les éléments descriptifs requis pour former un modèle cohérent. Dans notre modèle, nous nous intéressons ainsi uniquement à la dimension organisationnelle (O), délaissant volontairement les autres dimensions à des travaux ultérieurs, et ceci afin que dans la description générale d'un SMA, les différentes dimensions explicitées soient décrites de manière indépendante et distincte. Cette indépendance permet tout d'abord une analyse possible des systèmes pour lesquels certaines des dimensions (par exemple le code des agents eux-mêmes) ne peuvent pas être observées ; elle permet également de concevoir, expliciter et

réutiliser les différentes dimensions des SMA séparément, afin de faciliter la conception, l'optimisation et la réutilisation de systèmes complexes.

La seconde considération que nous prenons comme base pour notre travail est l'indépendance de l'organisation vis-à-vis de la population réelle qui l'implémente, telle qu'explicitée dans le modèle PopOrg. En d'autre termes, nous voulons que la description de l'organisation d'un SMA soit indépendante (et complémentaire) de la connaissance des agents qui la composent réellement dans une implémentation du système : une même organisation, c'est-à-dire un ensemble de fonctions dans le système et d'interactions possibles entre agents, peut, dans ce cadre, être réalisée par différents ensembles d'agents, chacun de ces ensembles pouvant être constitué d'agents dotés d'architectures internes et de protocoles différents. Notre modèle organisationnel devra donc être spécifié indépendamment de l'architecture interne (du code) des agents qui composeront les implémentations du système.

> *Dans l'exemple des ensembles de fourmis, par exemples, l'organisation, décrit l'ensemble des fonctions remplies par les fourmis, leurs possibilités de communication – les cas d'utilisation des phéromones – et leurs objectifs globaux – construire une fourmilière, stocker la nourriture, se reproduire. Cette même organisation peut être réalisée par de nombreux types d'agents différents, codés à l'aide de diverses architectures : programmation à base de règles, programmation à base de cas, programmation fonctionnelle, programmation logique, réseaux bayésiens, etc.*

Le choix de ces deux principes est motivé par une volonté double : créer un modèle d'organisations générique et ouvrir la possibilité d'intégrer par la suite ce modèle dans un modèle général de description des SMA. La généricité est assurée par l'indépendance des organisations de toute spécificité des agents du système, en cela que l'organisation et les agents sont deux dimensions distinctes de la description d'un SMA et que nous garantissons dans notre seconde considération que l'une n'influe pas sur l'autre. L'intégration à un modèle plus générique est également rendue possible par une distinction formelle entre les différentes dimensions de la description des SMA et par l'absence d'influences réciproques (*i.e.* l'indépendance) entre ces dimensions.

1.4.2 Approche du problème

C'est en gardant ces principes à l'esprit que nous commençons notre travail par une analyse de l'état de l'art articulée autour de la comparaison de deux ensembles distincts : d'une part, les modèles d'organisation des SMA, et d'autre part, l'ensemble des algorithmes et mécaniques de réorganisation et d'auto-organisation développés pour les SMA. C'est en étudiant la portée et la complémentarité de ces deux ensembles, en comparant leurs

éléments, leurs méthodes et leurs objectifs que nous cherchons à déterminer les éléments de base de notre modèle organisationnel. Nous visons ainsi à définir des éléments constitutifs d'un modèle organisationnel permettant de maximiser la variété d'organisations pouvant être décrites, tout en maintenant une précision suffisante pour spécifier de la manière la plus complète possible les organisations décrites. Ces éléments devront répondre à la double problématique de décrire les aspects statiques et dynamiques des organisations.

Par la suite, notre objectif est de construire, à l'aide de ces éléments de base, un modèle complet des organisations définissant de quelle manière une organisation peut être représentée et formalisée. Cette formalisation, tout en prenant en compte la diversité des organisations existantes, vise à rassembler dans un même langage l'expression des nombreuses propriétés organisationnelles des SMA. Nous cherchons à évaluer l'expressivité de notre modèle en exprimant les notions organisationnelles relevées dans l'état de l'art et en comparant les possibilités expressives de notre modèle à celles des autres modèles existants à travers ces notions (cf. section 4.4). Nous cherchons aussi à évaluer l'expressivité du modèle vis-à-vis des propriétés significatives des SMA dans le contexte des algorithmes de réorganisation et d'auto-organisation étudiés dans l'état de l'art. Cette évaluation sera réalisée dans le but de déterminer dans quelle mesure notre modèle permet de formaliser et d'étudier des propriétés organisationnelles supplémentaires par rapport aux autres modèles organisationnels connus.

Nous visons également, pour suivre nos objectifs, à rendre possible la conversion de cette formalisation en termes d'éléments concrets d'ingénierie des SMA, c'est-à-dire d'éléments intégrables à la création de squelette de code ou de code final – lorsque associés aux éléments descriptifs des autres dimensions – lors de la conception d'un SMA. Par ailleurs, nous tentons de donner à notre modèle, à travers sa formalisation, des possibilités calculatoires suffisantes pour exprimer des propriétés plus complexes sur les organisations considérées. Pour cela, nous cherchons à déterminer au moins un modèle possible des mécanismes mis en jeu dans la dynamique organisationnelle des SMA qui permette le calcul sur cette dynamique. Un tel modèle doit permettre l'agrégation d'informations suivant la dimension temporelle et l'évaluation de propriétés organisationnelles dans des situations génériques et particulières. Il nous faut pour cela trouver des méthodes mathématiques adéquates pour évaluer des situations incertaines et en extraire des propriétés globales ; de ces méthodes, nous pouvons déterminer certaines des capacités analytiques de notre modèle et les appliquer à des exemples précis.

Dans ce travail, nous proposons donc, en réponse à la problématique posée, un modèle organisationnel, à notre connaissance original, fondé sur des éléments atomiques bas-niveau, et respectant les différentes hypothèses posées dans la description de l'approche suivie.

La thèse que nous défendons dans ce travail est qu'il est possible de construire un modèle descriptif des organisations pour les SMA fondé sur des éléments descriptifs locaux et répondant aux exigences suivantes :

- la capacité de décrire tous types d'organisations, qu'elles soient fermées ou ouvertes, totalement ou partiellement spécifiées ;
- la possibilité de décrire les propriétés globales, statiques et dynamiques, des organisations à partir de la connaissance des spécifications locales ;
- la capacité de servir comme outil pour la conception et l'analyse des SMA, à travers, notamment, son inclusion dans des méthodes de conception et dans des méthodes de comparaisons et d'évaluation de systèmes.

1.5 Plan du mémoire

Ce mémoire s'organise en trois parties[2], dans lesquelles nous présentons successivement les différents éléments annoncés en introduction.

- Dans la première partie, à la suite de l'introduction, nous présentons dans le chapitre 2 l'état de l'art, suivant les axes définis ci-dessus : méthodologie de conception des SMA, modèles organisationnels, techniques d'auto-organisation et de réorganisation.
- Dans une deuxième partie, nous exposons la contribution apportée par notre travail (Lacomme, 2011). Elle s'organise comme suit :
  - Dans le chapitre 3, nous présentons les différentes hypothèses et les fondements de notre approche.
  - Dans le chapitre 4, nous exposons la formalisation et la sémantique de notre modèle organisationnel.
  - Dans le chapitre 5, nous présentons la méthode de calcul de propriétés globales sur les mécanismes organisationnels que nous avons choisi d'étudier.
  - Dans le chapitre 6, nous proposons une méthode de conception des organisations pour les SMA fondée sur notre modèle, et étudions son applicabilité pour des cas pratiques.

[2] Nous utiliserons, tout au long de ce mémoire, la notation en gras pour les mots-clés et les concepts que nous introduisons et la notation en italique pour les termes empruntés à d'autres modèles ou domaines et les exemples.

- Dans une troisième et dernière partie, nous présentons des exemples, conclusions et perspectives pour notre modèle :
  - Dans le chapitre 7, nous présentons deux applications différentes de notre modèle, mettant en jeu respectivement chacun des deux objectifs de notre modèle : la conception et l'analyse.
  - Enfin, nous concluons sur les travaux présentés et développons quelques perspectives de recherche quant aux apports présentés et aux résultats obtenus.

Chapitre 2. État de l'art

Dans ce chapitre, nous faisons un état de l'art orienté vers la représentation de la dynamique des organisations dans les SMA. Cet état de l'art est agencé en trois parties, correspondant chacune à un domaine différent du multi-agent. Chacun de ces domaines est chargé de représentations spécifiques de la dynamique organisationnelle, et correspond à des positionnements différents vis-à-vis de la problématique annoncée :

1) l'étude des méthodes de conception pour les SMA permet d'avoir une vue des problèmes soulevés par la conception des SMA et de positionner notre modèle sur l'axe de conception
2) l'étude des modèles organisationnels existants permet de répertorier un panel de notions organisationnelles utiles et utilisées dans la littérature, de soulever les lacunes descriptives des modèles existants, et de positionner notre travail par rapport à ces manques sur l'axe de description
3) enfin, l'étude des méthodes d'auto-organisation et de réorganisation dans les SMA permet de prendre en compte la variété des dynamiques organisationnelles existant dans le domaine, de définir les besoins et les objectifs de représentation de ces dynamiques et de comparer ces besoins avec les modèles organisationnels répertoriés précédemment ; elle permet de positionner notre travail sur l'axe d'analyse.

2.1 Méthodologie de conception des SMA

Notre but n'est pas, dans ce travail, de proposer une nouvelle méthode de conception des SMA, mais seulement une méthode de description limitée aux organisations. Nous nous intéressons cependant aux méthodes de conception existantes pour plusieurs raisons :

- l'identification des différents concepts et méthodes clés utilisés au niveau organisationnel dans la conception des SMA ;
- l'analyse de la possibilité de concevoir des SMA ouverts, émergents, auto-adaptatifs ;
- l'étude des prérequis architecturaux imposés par les différentes méthodes ;
- les possibilités existantes de comparaison et d'évaluation des SMA créés, notamment au niveau des organisations.

Ensemble, ces informations nous permettent de poser les bases des besoins et des problèmes existants dans l'état de l'art du paradigme SMA, et ainsi d'orienter notre étude ultérieure de cet état de l'art, au niveau des modèles organisationnels et des mécanismes d'auto-organisation et de réorganisation.

2.1.1 Critères d'analyse

Nous passons donc en revue, en nous appuyant principalement sur des analyses antérieures (Picard, 2004), les méthodes reconnues de conception des SMA, et nous les analysons selon des critères correspondant aux raisons de notre étude :

- Quelles notions organisationnelles sont utilisées par la méthode ?
- Est-il possible de concevoir des SMA ouverts à travers cette méthode ?
- La méthode rend-elle possible l'inclusion, dans la conception du système, d'une dynamique organisationnelle (auto-organisation, réorganisation) ?
- La méthode permet-elle de concevoir un système indépendamment de l'architecture interne des agents qui l'instancient (ou bien impose-t-elle une architecture précise) ?
- Les modèles de représentation utilisés par la méthode permettent-ils une évaluation de la pertinence et de la performance du système proposé par rapport aux attentes ?
- Les modèles de représentation utilisés permettent-ils la comparaison des systèmes entre eux, notamment au niveau organisationnel ?

2.1.2 Méthodes de conception des SMA

AAII

Le modèle AAII (*Australian Artificial Intelligence Institute Methodology*) (Kinny *et al.*, 1996) est une méthode de conception multi-agent fondée sur le concept des agents BDI (Rao et Georgeff, 1995). Il distingue les points de vue *interne* et *externe* aux agents, c'est-à-dire l'architecture interne et l'organisation, mais limite cette dernière à des concepts dérivés de l'architecture interne : hiérarchie de classes, responsabilités, services. Il n'est ainsi possible de concevoir, grâce à cette méthode, que des SMA fondés sur des agents BDI, c'est-à-dire cognitifs et dotés d'un raisonnement sur les *croyances*, les *désirs* et les *engagements* – ainsi qu'autour de la notion de *plans*. Cette méthode ne laisse pas la possibilité de considérer des systèmes ouverts, ni de spécifier des liens et des propriétés dynamiques sur le système.

Aalaadin

La méthode Aalaadin (Ferber et Gutknecht, 1998) est axée autour du modèle AGR, prédécesseur d'AGRE (cf. section 2.2.2). Elle présente ainsi les avantages et les inconvénients de ce modèle, notamment l'indépendance de la description organisationnelle vis-à-vis de l'architecture agent, ainsi que la description à partir de concepts normatifs des organisations humaines : *rôles* et *groupes*. La méthode de conception, orientée autour de descriptions UML, se limite principalement à l'organisation elle-même ; celle-ci définit la structure et la fonction du système. En revanche, du fait des notions utilisées dans la description organisationnelle (rôles et groupes), qui sont des notions statiques d'inspiration humaine, celle-ci reste incapable de décrire des organisations à forte dynamique ou des systèmes auto-organisés.

ADELFE

La méthode ADELFE (Bernon *et al.*, 2003) s'oriente vers la conception de systèmes auto-organisés à forte dynamique. Elle prend en compte le choix et l'évaluation itérative des mécanismes de réorganisation et d'auto-organisation pour garantir la fonction attendue du système : elle repose sur l'approche AMAS (*Adaptative MultiAgent Systems*), qui s'appuie sur des mécanismes organisationnels locaux visant à éliminer les situations de *non-coopération* (notamment les situations de conflits ou d'incompréhension entre agents). Elle prend ainsi en compte les systèmes ouverts et tente de s'abstraire des architectures internes des agents sans toutefois arriver à une véritable indépendance (la spécification organisationnelle impliquant un contrôle sur l'implémentation des agents). Cependant, elle perd en généricité ce qu'elle gagne en capacité à décrire et comparer les mécanismes locaux : les concepts organisationnels plus généraux et les organisations statiques sont difficiles à concevoir avec cette méthode. Par ailleurs, elle se fonde sur des méthodes flexibles, bien qu'intégrées dans des processus de conception répandus (RUP, description en SPEM), et non sur une véritable formalisation de l'organisation des systèmes.

Cassiopée

La méthode Cassiopée (Drogoul et Collinot, 1998 ; Collinot et Drogoul, 1998) – et son extension Andromède (Drogoul et Zucker, 1998) – s'appuient sur une représentation des organisations multi-agents fondée sur les rôles et les groupes, mais surtout sur la notion fonctionnelle de dépendance pour proposer une méthode *bottom-up* (ce que nous nommons dans ce travail une approche émergentiste) de la conception des SMA. Suivant ce principe, les rôles et les groupes sont définis à partir des dépendances entre les tâches nécessaires à la fonction globale du système et leur attribution aux différents agents. Si cette approche est

particulièrement adaptée à une modélisation totalement indépendante des architectures internes des agents et très génériques dans les notions qu'elle permet de modéliser, la méthode Cassiopée est en revanche limitée dans sa modélisation des organisations dynamiques ou ouvertes par un manque de possibilité de formalisation de ces notions : s'il est possible de décrire des organisations dynamiques à travers les descriptions de dépendances et des populations d'agents, il est en revanche difficile de concevoir, évaluer et réutiliser les mécanismes qui contrôlent cette dynamique.

DESIRE

La méthode DESIRE (*DEsign and Specification of Interacting REasoning framework*) (Brazier *et al.*, 1997) est une autre méthode orientée autour de l'architecture BDI, au même titre qu'AAII. Bien que davantage axée sur la pragmatique de la conception, et orientée autour de modèles formels permettant une bonne évaluation des choix de conception par rapport aux objectifs de conception, elle n'en reste pas moins limitée, de la même manière qu'AAII, par le lien important entre les notions organisationnelles utilisées et les principes de l'architecture BDI. Elle se limite ainsi également aux systèmes fermés, dotés d'une organisation statique et d'un objectif global dérivable en une hiérarchie de tâches précises.

GAIA

La méthode GAIA (Wooldridge *et al.*, 2000 ; Zambonelli *et al.*, 2003) est une méthode fondée sur une représentation organisationnelle multi-niveaux ; la distinction entre niveau micro et niveau macro y est ici faite, apportant les avantages d'une réelle indépendance de la méthode vis-à-vis de toute architecture agent précise, ainsi que la possibilité de réutiliser les éléments définis pour une organisation dans d'autres situations (en particulier les éléments de niveau macro, suffisamment génériques pour prendre sens dans différentes organisations). En revanche, le modèle organisationnel étant construit entièrement autour d'une représentation sous forme de *services*, ainsi que de concepts normatifs associés, tels que *responsabilités* et *permissions*, la méthode ne permet pas de décrire des systèmes dont les organisations soient dynamiques, ou même ouvertes. Elle se limite ainsi à la description de systèmes dont les organisations sont figées et les rôles et comportements des agents définis complètement à la conception.

INGENIAS

La méthode INGENIAS (Pavón *et al.*, 2005 ; Gomez-Sanz *et al.*, 2009), dérivée de la méthode MESSAGE (*Methodology for Engineering Systems of Software AGEnts*) (Caire *et al.*, 2002), est une méthode de conception fondée sur des concepts organisationnels haut-niveau, tels que les *rôles* et les *groupes* et dont la représentation fonctionnelle s'articule autour d'une

notion complexe de *tâche*. Elle s'insère dans des processus de conception bien définis, et propose des modèles de description dérivés d'UML, afin de faciliter la réutilisation de composants et la génération de code à partir des spécifications. Elle se veut indépendante des architectures internes des agents et tolère les systèmes ouverts et la dynamique dans la structure, c'est-à-dire dans les occurrences d'*interactions* entre agents. En revanche, sa limitation se trouve dans la difficulté d'exprimer, à travers les notions organisationnelles utilisées, les cas précis et les propriétés nécessaires de la dynamique organisationnelle au niveau micro, tant au niveau des interactions possibles que des délégations et répartition de tâches entre agents.

MaSE

La méthode MaSE (*Multiagent Systems Engineering*) (Deloach, 1999) est une méthode complète d'ingénierie des SMA qui s'appuie sur UML. De la même façon que pour AUML (Odell *et al.*, 2000), une extension d'UML pour la représentation des concepts multi-agents (cf. section 2.2.2), cela permet d'intégrer la représentation des agents et de leurs composants à une méthode de conception de systèmes. En revanche, si cette méthode offre une certaine indépendance vis-à-vis des architectures internes choisies pour les agents, la description organisationnelle sur laquelle s'appuie MaSE est limitée à des organisations statiques et fermées, à cause d'une approche *top-down* de la décomposition du système en *rôles* ; elle est de plus limitée, par la complexité de la description et de l'analyse, à des systèmes ne requérant qu'un faible nombre d'agents différents. En revanche, il est à noter que le formalisme utilisé a permis la création d'un outil de génération partielle de code à partir des spécifications : *agentTool* (Deloach et Wood, 2001).

MASSIVE

La méthode MASSIVE (*Multi Agent SystemS Iterative View Engineering*) (Lind, 2001) est fondée sur un processus itératif de conception sur un modèle de vues. Ce processus s'appuie sur des spécifications bas-niveau (comportements et état locaux) permettant à la fois un contrôle précis des différents objectifs du système, une conception fondée sur les besoins utilisateur et une indépendance fonctionnelle vis-à-vis des architectures et langages choisis. Cette approche atteint son plein potentiel dans la conception de systèmes composés majoritairement d'agents réactifs, permettant alors une conception efficace et une évaluation des choix facilitée par un formalisme modulaire basé sur UML adapté à la génération de code et à l'analyse rapide. En revanche, en l'absence, dans cette méthode, d'un réel modèle organisationnel indépendant des mesures de performances et de qualité, l'expression des

notions haut-niveau des organisations (rôles, groupes, normes) et la dynamique de celles-ci sont plus difficilement exprimables.

PASSI

La méthode PASSI (*Process for Agent Societies Specification and Implementation*) (Cossentino et Potts, 2002) est à nouveau une méthode de conception qui s'appuie sur des descriptions UML provenant d'une analyse *top-down* des besoins liés au système. Il s'agit d'une méthode peu générique : elle ne permet de définir que des organisations fermées et statiques, dans lesquelles les agents correspondent nécessairement à l'implémentation des besoins liés à leurs rôles (elle ne dispose donc que de peu d'indépendance vis-à-vis des architectures internes de ces agents, qui doivent être directement capables de répondre aux attentes de conception).

Prometheus

La méthode Prometheus (Padgham et Winikoff, 2002), fondée au départ sur les concepts associés à l'architecture d'agents BDI, a évolué pour atteindre l'indépendance des descriptions effectuées à la conception vis-à-vis de l'architecture interne des agents. Cependant, l'héritage de cette création se trouve dans la nécessité descriptive, au sein du modèle, de l'association fonctionnelle statique entre agents et objectifs liés au système. Si elle permet dans une certaine mesure la validation et la vérification des modèles d'organisation conçus, elle se limite donc elle aussi à la conception d'organisations statiques et fermées.

TROPOS

La méthode TROPOS (Castro *et al.*, 2001) est également une méthode fondée sur des concepts BDI. Si elle permet une approche itérative impliquant une évaluation successive des choix de conception effectués, elle est très limitée quant à l'aspect organisationnel, puisqu'elle ne propose qu'une représentation des *dépendances* et des liens entre classes, et aucun modèle complet des organisations des systèmes conçus. Elle ne permet donc que très peu d'expressivité organisationnelle, et n'autorise donc pas la conception efficace de systèmes ouverts, auto-organisés, ou même de systèmes normatifs (accessibles aux modèles utilisant des notions de rôles, de groupes ou de normes).

VOYELLES

La méthode VOYELLES (Demazeau, 1995 ; Demazeau, 1996) est une méthode entière orientée autour des notions propres au domaine du multi-agent ; elle repose sur la décomposition de la description du système en quatre dimensions : agent, environnement, interaction, organisation. La description organisationnelle y est donc potentiellement riche et

fortement indépendante de la vue agent et donc des architectures internes choisies, comme exprimé dans le modèle associé *PopOrg* (Demazeau et Costa, 1996). Elle admet la description de tous types de systèmes, y compris les systèmes ouverts et ceux dotés d'organisations dynamiques. En revanche, il s'agit d'une méthode haut-niveau, pour laquelle les modèles de description ne sont pas spécifiés. Elle offre donc beaucoup de possibilités tout en n'étant pas dotée des outils appropriés pour leur réalisation. Le manque de formalisme implique notamment une faible réutilisabilité des modèles conçus en raison d'une tendance, résultant du manque d'outils, à l'utilisation d'outils et de formalisation *ad hoc* pour chaque cas de conception.

Tableau 2-A - Caractéristiques des méthodes de conception des SMA

	Ouverture	Dynamique	Indépendance	Évaluation	Comparaison
AAII	×	×	×	×	×
Aalaadin	✓	×	✓	×	×
ADELFE	✓	✓	×	✓	×
Cassiopée	×	✓	✓	×	×
DESIRE	×	×	×	✓	×
GAIA	×	×	✓	×	✓
Ingenias	✓	✓	✓	✓	×
MaSE	×	×	✓	✓	×
MASSIVE	✓	×	✓	✓	✓
PASSI	×	×	✓	×	×
Prometheus	×	×	✓	✓	×
TROPOS	×	×	✓	✓	×
VOYELLES	✓	✓	✓	✓	×

2.1.3 Conclusion sur les méthodes

Dans notre analyse des méthodes de conception des SMA, nous avons essayé de déterminer des caractéristiques (cf. Tableau 2-A) qui nous paraissent essentielles dans la description et la conception des SMA, principalement du point de vue de l'expression des organisations : possibilité de décrire les organisations ouvertes, capacité à spécifier la dynamique du système, indépendance de la conception organisationnelle vis-à-vis des choix d'architecture

interne des agents, possibilité d'évaluation des systèmes en cours de conception et enfin comparaison des choix de conception entre eux.

De cette analyse, il ressort que la méthode la plus appropriée, car la moins contraignante, à nos objectifs de modélisation, est la méthode VOYELLES. Le seul critère qu'elle ne remplit pas est la possibilité de comparer les choix de conception entre eux, car elle ne définit pas de cadre suffisamment formel pour la mise en place de telles comparaisons. En revanche, elle adopte une approche validant tous les autres critères qui nous semblaient nécessaires à la mise en place de notre modèle. Par ailleurs, c'est une méthode haut-niveau, n'imposant pas de langage ou de modèle particulier, et qui se prête donc à la mise en place d'un nouveau modèle organisationnel sans problème de liens ou de corrélation avec les modèles déjà associés aux méthodes. C'est donc à partir des concepts et des procédures développés par la méthode VOYELLES que nous construisons par la suite notre approche d'un modèle organisationnel.

2.2 Modèles organisationnels

Un point important de notre état de l'art, étant donné notre objectif dans ce travail, est l'étude des modèles organisationnels existants, et la comparaison de leurs capacités expressives. Nous nous intéressons aux types d'organisations qui peuvent être décrites à travers chacun de ces modèles, mais aussi à l'ensemble des notions organisationnelles mises en jeu dans les descriptions obtenues à travers ces modèles.

En effet, notre objectif principal consiste à déterminer les limitations des modèles existants par rapport aux organisations rencontrées dans l'univers multi-agent actuel, et en particulier par rapport aux organisations dynamiques conçues autour de mécanismes d'auto-organisation ou de réorganisation (cf. section 2.3). De ces limitations nous pouvons tirer les fondements et les justifications de notre apport.

2.2.1 Critères d'étude

Nous étudions les modèles organisationnels existants en nous appuyant sur l'analyse de leurs propriétés expressives et des notions organisationnelles qu'ils manipulent.

Pour chaque modèle étudié, nous nous intéressons ainsi à :

- Déterminer si l'ouverture des organisations (la possibilité pour les agents d'entrer et sortir à tout moment du système) est descriptible.

- Déterminer si la dynamique organisationnelle est prise en compte, tant d'un point de vue structurel que fonctionnel, et s'il est possible de la représenter de manière précise à l'aide du modèle.
- Déterminer si le modèle organisationnel implique une connaissance des architectures internes des agents, ou s'il les laisse libres.
- Énumérer l'ensemble des notions organisationnelles de base sur lesquelles s'appuie le modèle pour construire des descriptions organisationnelles.

2.2.2 Modèles organisationnels

AGRE

Le modèle AGRE (Ferber *et al.*, 2005), extension du modèle AGR (Ferber *et al.*, 2003) prenant en compte l'environnement, est construit autour de trois concepts clés : les agents, les groupes et les rôles. Les *agents* sont les entités actives et communicantes du système ; les *groupes* sont les ensembles d'agents partageant des caractéristiques communes, et définissant les lieux de communication possible entre agents ; les *rôles* sont les représentations des fonctions jouées par les agents au sein d'un groupe.

L'avantage d'un tel modèle est sa capacité à décrire les organisations indépendamment de toute implémentation ou architecture interne des agents. Les inconvénients en sont que les organisations descriptibles dans ce modèle sont réduites aux organisations statiques, fermées (ou tout au moins dans lesquelles la population d'agents est spécifiable d'avance), et dans lesquelles les agents jouent chacun une fonction qui est une part de l'objectif global du système ; cela provient du fait que les notions organisationnelles utilisées sont haut-niveau, prennent leur origine dans les organisations humaines, et en cela conservent un aspect statique et normatif. Ces notions ont en revanche l'avantage de permettre une description très fine et précise des types d'organisations qu'elles recouvrent.

AUML

AUML (Odell *et al.*, 2000) n'est pas à proprement parler un modèle d'organisations mais un langage de description des SMA qui se présente comme une extension du langage UML pour les agents. Ce langage permet cependant d'exprimer des concepts organisationnels, principalement autour des notions d'agent, de rôles et de protocoles. Les *rôles* sont ici considérés comme de simples notations pour les comportements adoptés par les agents, tandis que les *protocoles* servent à définir les méthodes d'organisation et de communication des agents.

Un tel langage est très limité du point de vue de la description des organisations, puisqu'il ne permet d'exprimer que des considérations statiques sur des entités bien définies ; il est ainsi limité aux organisations statiques et fermées, mais possède de plus une expressivité très restreinte quant aux propriétés de ces organisations. Enfin, l'imbrication des descriptions AUML implique une dépendance importante entre les différents niveaux de description, et en particulier entre organisations et spécifications des agents.

ISLANDER

Le modèle et langage ISLANDER (Esteva *et al.*, 2002) réfère aux organisations multi-agents d'une manière particulière : les *institutions électroniques* (Esteva *et al.*, 2001). Cette vision des organisations s'articule autour de concepts spécifiques : les structures performatives, les scènes qui les composent, et les normes[3]. Les *scènes* correspondent à des zones de dialogue possible entre agents (qui pourraient être comparées aux groupes du modèle AGRE, cf. *supra*). La *structure performative* regroupe les scènes d'un système et les *rôles* qui peuvent y être joués, au sein d'une structure dans laquelle les agents peuvent se mouvoir. Les *normes* décrivent la manière dont les agents sont autorisés à se comporter, soit au sein d'une scène, soit entre les scènes, dans la structure performative.

Cette approche des organisations est très différente des autres approches descriptives de l'état de l'art, car il ne s'agit pas d'un modèle constructif mais d'un modèle de contraintes. Cependant, ce modèle ne répond quand même qu'imparfaitement au problème des systèmes ouverts, car, s'il autorise la définition d'institutions pour de tels systèmes, il ne propose pas nécessairement les outils pour valider, à la réalisation du système, le maintien effectif de ces normes. Par ailleurs, la structure performative définie l'est de manière statique, et n'autorise donc pas la définition d'organisations fonctionnellement dynamique (les rôles et leurs interactions sont statiques), bien que la dynamique des agents eux-mêmes soit possible.

MAS-ML

Le langage MAS-ML (da Silva *et al.*, 2004) est un dérivé du langage UML s'attachant à la description complète de systèmes multi-agents. Cette description inclut un niveau organisationnel, dont les éléments principaux sont les rôles.

Cette description est, de la même manière qu'AUML (cf. *supra*), limitée dans son étendue par l'imbrication des niveaux de description et par le manque d'expressivité du modèle d'organisation. De plus, les *rôles* définis dans MAS-ML sont fondés sur un modèle BDI de l'agent, et sont donc décomposés en concepts de *buts*, de *croyances*, d'*actions* et de *plans*. La

[3] *Normative rules.*

description organisationnelle en est rendue plus précise mais limitée aux systèmes composés d'agents BDI.

MOISE+

Le modèle MOISE+ (Hubner *et al.*, 2002), extension du modèle MOISE (Hannoun *et al.*, 2000), est un modèle de description de SMA orienté organisations. Il s'agit d'un modèle normatif (la dimension normative est ici nommée *déontique*), de même qu'ISLANDER, mais construit autour des notions de groupes, de rôles et de relations. Les *groupes* définissent dans ce modèle également les lieux où l'interaction entre agents est possible ; les *rôles* représentent les comportements possibles ou autorisés pour les agents ; les *relations* représentent les liens entre agents jouant des rôles spécifiques ; elles sont typées selon les liens représentés : connaissance, communication ou autorité. Il s'agit par ailleurs d'un modèle multi-niveaux : chaque organisation est spécifiable au niveau *individuel* (les comportements d'agents autorisés pour chaque rôle), au niveau *social* (les interactions possibles entre rôles), et au niveau *collectif* (les structures résultant de ces rôles interconnectés). Les possibilités d'interactions entre agents peuvent également être normées, comme décrit dans (Boissier *et al.*, 2010).

Il s'agit d'un modèle permettant de spécifier de manière très complète les organisations, indépendamment des agents qui les instancient : les agents entrant dans le système peuvent choisir de jouer différent rôles, mais sont dès lors contraints par les normes de comportement et d'interaction liées à ce rôle. Le défaut de ce modèle est le fait que la description des organisations est entièrement statique et que, si la population d'agents peut évoluer, les contraintes d'organisations sont définies statiquement pour le système et ne peuvent en aucun cas s'adapter dans le temps. Ce modèle ne permet donc pas de définir des systèmes auto-organisés ou réorganisés, dans lesquels la description organisationnelle doit être dynamique.

ODML

Le modèle ODML (*Organizational Design Modeling Language*) (Horling et Lesser, 2005) est un modèle d'organisations à part, car il est conçu expressément pour l'évaluation de la dynamique organisationnelle des systèmes. Il est construit autour de notions bas-niveau, et en particulier autour de la notion de *nœud*, un nœud pouvant représenter de nombreuses entités différentes, telles que les agents, les rôles, les relations entre agents, etc. Ces nœuds sont reliés entre eux par deux types de liens : *has-a* et *is-a*, qui représentent respectivement l'agrégation et la dérivation (entendues au sens qu'elles ont dans le langage UML). De plus, les nœuds possèdent des *paramètres*, des *variables* et des *constantes*, représentant différentes

propriétés, mais aussi des *modificateurs*, représentant la manière dont leurs propriétés sont affectées par celles des autres nœuds. Ce modèle permet, à partir d'une description organisationnelle fondée syntaxiquement sur ces notions, et sémantiquement sur des notions provenant d'autres modèles organisationnels (par exemple les agents, les groupes, les relations, les rôles, etc.), de dériver un ensemble d'équations sur les propriétés contenues dans les nœuds. Ces équations modélisent la qualité de l'organisation vis-à-vis des attentes et des mesures qui peuvent être effectuées dessus. Elles permettent donc de modéliser, d'évaluer et de comparer les dynamiques organisationnelles de plusieurs organisations possibles pour un même problème.

Ce modèle propose une approche originale par rapport aux autres modèles organisationnels, en cela qu'elle apporte les notions d'évaluation et de dynamique aux organisations multi-agents. Cependant, elle possède des limites importantes. Tout d'abord, la dynamique organisationnelle ainsi étudiable n'est qu'une dynamique fonctionnelle du système : les organisations modélisées doivent en effet former un graphe fixe pour que l'étude puisse avoir lieu. Ainsi, les organisations considérées n'en sont pas moins nécessairement structurellement fixes, et les systèmes étudiés sont nécessairement des systèmes fermés. Ensuite, l'étude des propriétés des organisations se fait non pas sur les organisations elles-mêmes mais sur des instances spécifiques. Cette dernière condition limite la portée des évaluations et des comparaisons qu'il est possible de faire avec ce modèle.

OMNI

Le modèle OMNI (*Organizational Model for Normative Institutions*) (V. Dignum *et al.*, 2005) présente une approche similaire à celle du modèle ISLANDER (cf. *supra*) : il s'agit d'une représentation fondée sur la notion d'*institution*. Il distingue le modèle organisationnel du modèle normatif, et se positionne comme un modèle multi-niveaux : niveau *abstrait* des spécifications à partir des objectifs du système, niveau *concret* de définition des objets organisationnels et niveau *implémentation*, définissant les objets et méthodes réels du système. Le modèle organisationnel contient une représentation des *rôles* qui représentent les attentes fonctionnelles vis-à-vis des agents, des *groupes* comme ensembles de rôles et des *dépendances entre rôles*, qui représentent les dépendances fonctionnelles entre les rôles ; au plus bas niveau, le modèle organisationnel est instancié par une population d'agents. Il contient également la représentation d'une structure d'*interactions*, représentant les interactions possibles entre rôles. Le modèle normatif, quant à lui, décrit l'ensemble des *normes* qui règlent le fonctionnement de l'institution, ainsi que les *règles* locales qui instancient ces normes pour les agents.

La force de ce modèle est qu'il combine les capacités expressives des modèles organisationnels comme AGR et des modèles normatifs comme ISLANDER, tout en donnant les moyens de spécifier complètement les réalisations effectives des normes dans une organisation. Il permet donc de décrire finement l'ensemble d'une organisation normée et de ses populations, et gère donc très bien les systèmes ouverts. Les limitations de ce modèle proviennent des limitations communes à ses différentes composantes, à savoir l'inaptitude à la description de la dynamique ; il est donc impossible de spécifier, à travers ce modèle, des systèmes auto-organisés ou réorganisés.

OperA

Le modèle OperA (V. Dignum, 2004) présente une approche inspirée de la notion d'*institutions* et ressemblant au modèle OMNI (cf. *supra*). Il contient, de la même façon, plusieurs modèles différents, représentant : i) la structure sociale, fondée sur les notions de *rôle*, de *groupe* et de *dépendances* ; ii) la structure d'interactions, basée sur les notions de *scènes* et de *transitions entre scènes* ; iii) la structure normative, représentant les *normes* suivies par les agents ; et enfin iv) la structure de communication, définissant les langages de communication. Ce modèle s'appuie, dans sa dimension sociale, sur la notion de *contrats* pour modéliser les engagements et les contraintes appliqués aux agents selon les rôles qu'ils acceptent au sein d'une organisation.

Ce modèle présente globalement les mêmes avantages et inconvénients que le modèle OMNI. Il apporte une modélisation plus fine de l'acceptation de rôles et des engagements sociaux entre agents à travers la notion de contrats. S'il possède un partitionnement horizontal similaire à OMNI, il lui manque, en revanche, le partitionnement vertical de ce dernier, relatif aux niveaux de spécification, et qui apporte à OMNI une plus grande flexibilité, en particulier au niveau de la spécification du modèle d'interactions, vis-à-vis de la prise en compte des systèmes ouverts.

STEAM

Le modèle STEAM (*a Shell for TEAMwork*) (Tambe *et al.*, 1999) est un modèle organisationnel fondé sur la notion d'*équipe*. Il se décompose en deux ensembles qui s'associent pour former la description organisationnelle : la *hiérarchie d'organisation,* qui définit la structure (hiérarchique ou non) des rôles de l'équipe, et la *hiérarchie d'activité*, qui représente les activités en cours pour l'équipe (les objectifs et les plans). Les rôles considérés sont de deux types : les *rôles persistants*, qui définissent des assignations durables d'objectifs, et les *rôles spécifiques à une tâche*, qui définissent les rôles que peuvent jouer les membres de l'équipe lorsque qu'une tâche spécifique est en cours.

Le modèle STEAM propose une approche dynamique de la réponse fonctionnelle d'une organisation aux tâches qui lui sont assignées. Le modèle permet de présenter la répartition des tâches aux différents membres d'une équipe en fonction des rôles assignés à chacun. Sa force est de permettre la comparaison de différentes assignations et de rechercher ainsi la répartition optimale pour chaque tâche. En revanche, les aspects structurels des organisations (relations et interactions possibles entre agents) ainsi que ses aspects normatifs en sont totalement absents. De plus, les équipes considérées se rapportent uniquement à des agents dont il est possible de spécifier les compétences et les comportements à la conception, et le modèle ne permet donc pas la modélisation des organisations ouvertes.

TÆMS

Le modèle TÆMS (Decker, 1996 ; Lesser *et al.*, 2004) est construit autour du concept de tâche. Il s'agit donc d'une définition purement fonctionnelle des organisations. Il décrit, pour une organisation, un ensemble de *tâches*, représentant des objectifs du système, reliées entre elles par des *relations* de dérivation ou de précédence. Ces tâches sont directement assignées aux agents.

Le modèle TÆMS est orienté vers la mesure de la progression des tâches dans le temps pour une organisation donnée. Il permet d'évaluer et de comparer des organisations à travers la progression vers leurs objectifs. Cependant, bien qu'indépendante vis-à-vis des spécifications des agents et que calculable sur des populations d'agents évolutives (systèmes ouverts), cette évaluation ne peut être menée que sur des instanciations des organisations. Les propriétés évaluées (la progression des différentes tâches du système) ne peuvent pas être calculées globalement pour les organisations, et par ailleurs, le modèle ne permet pas l'expression des dimensions structurelles et normatives des organisations.

2.2.3 Synthèse sur les modèles organisationnels

Nous avons comparé ici un nombre important de modèles organisationnels pour le multi-agent dans le but de déterminer les possibilités expressives offertes par chacun d'entre eux ainsi que les différentes approches suivies pour l'expression des notions organisationnelles (cf. Tableau 2-B).

Nous pouvons en retenir, en premier lieu, que les notions utilisées sont variées, mais que certaines notions, bien que variables, sont proches sémantiquement et fréquemment utilisées :

- les rôles, comme ensemble d'objectifs et/ou de capacités ;
- les groupes et équipes comme ensembles fonctionnels d'agents ou de rôles ;

- les normes, comme expression de spécifications comportementales et sociales.

Il apparaît également que, si ces notions décrivent de manière très adaptée les organisations ressemblant à des organisations humaines – d'où ces notions sont dérivées – elles ne permettent de définir que très difficilement et incomplètement les organisations ouvertes et les organisations à forte dynamique – auto-organisées, adaptatives, etc. Si les modèles normatifs permettent de s'abstraire des populations d'agents (et donc de se rendre indépendant des architectures agent et aptes à décrire des systèmes ouverts), c'est au prix de l'impossibilité de la description d'une quelconque dynamique organisationnelle.

Nous en déduisons que, pour construire un modèle capable de décrire tous types d'organisations, tant sur le plan statique que dynamique, il semble nécessaire de repenser l'ensemble des éléments de base sur lequel fonder notre modèle, afin d'intégrer l'ensemble des exigences d'expressivité et d'indépendance que nous recherchons.

Par ailleurs, il nous faut remarquer les seuls modèles réellement axés sur l'évaluation dynamique des organisations par le calcul de propriétés définies, sont les modèles ODML et TÆMS. Le modèle TÆMS étant approprié pour le calcul de propriétés fonctionnelles sur des instances de systèmes dotés d'organisations et de populations d'agents entièrement spécifiées, il est limité à la description fonctionnelle et ne permet aucune spécification ou évaluation des organisations *a priori* de leur instanciation. Quant au modèle ODML, bien qu'il soit le plus approprié pour exprimer des informations quantitatives sur des organisations spécifiables fonctionnellement et structurellement, il est en revanche limité dans son expressivité, notamment vis-à-vis des organisations à structures dynamiques ou à population changeante. De plus, les types de propriétés exprimables et calculables à travers ce modèle sont limités par le type d'expression qu'elles doivent prendre : ce sont des calculs d'expressions à partir de variables réparties dans un graphe évolutif, relativement à une instance précise d'une organisation. Ainsi, il semble difficile d'exprimer à travers ce modèle des propriétés complexes et non restreintes à des instances, comme le temps qu'il faut à un système pour se réorganiser ou encore la réussite des délégations de tâches entre agents (cf. exemples de mécanismes organisationnels, section 2.3). Il est enfin impossible de spécifier des organisations ouvertes à travers ce modèle.

Tableau 2-B - Caractéristiques des modèles organisationnels

	Ouverture	Dynamique	Indépendance	Notions organisationnelles[4]
AGRE	NON	NON	OUI	Groupes, Rôles
AUML	NON	NON	NON	Protocoles, Rôles
ISLANDER	limité	NON	OUI	Institutions, Scènes, Structures performatives, Normes
MAS-ML	NON	NON	NON	Rôles
MOISE+	OUI	NON	OUI	Rôles, Relations, Groupes
ODML	NON	limité	OUI	Nœuds, Liens, Contraintes, Modificateurs, Variables
OMNI	OUI	NON	OUI	Rôles, Groupes, Dépendances, Interactions, Normes, Règles
OperA	limité	NON	OUI	Rôles, Groupes, Dépendances, Communications, Normes, Contrats
STEAM	NON	limité	OUI	Équipe, Rôles persistants, Rôles spécifiques, Tâches
TAEMS	OUI	limité	OUI	Tâches, Relations entre tâches

Nous concluons de cela que, si ODML est un modèle très intéressant pour certains types de propriétés, et cela grâce à ses éléments et opérateurs bas-niveau, il nous faut, pour atteindre un plus grand degré d'expressivité, conserver la possibilité d'exprimer, dans notre modèle, des notions organisationnelles de haut niveau telles que rôles, groupes ou normes, afin de s'abstraire des agents eux-mêmes, et de prendre notamment en compte les systèmes ouverts.

2.3 Auto-organisation et réorganisation dans les SMA

Avec l'essor des systèmes multi-agents, la notion d'organisation a progressivement pris de plus en plus d'importance, passant de la simple description globale d'un système à une composante à part entière du domaine du multi-agent. Aux organisations figées et définies *a priori* se sont progressivement substituées des organisations dynamiques, évolutives et adaptatives, visant à une optimisation constante du système face à un environnement changeant et aux besoins fluctuants des utilisateurs.

L'évolution temporelle de l'organisation d'un système multi-agent a tout d'abord pu être vue comme relevant d'un algorithme global et centralisé, gérant les relations entre agents d'un point de vue omniscient. Mais avec le développement de réseaux totalement décentralisés et ouverts, contenant des agents hétérogènes, en nombre et en qualités variables, aux buts parfois antagonistes, comme dans le cadre des réseaux sociaux, cette

[4] Hors notion d'*agent* (seul le modèle ODML n'utilise pas cette notion).

approche a très vite montré ses limites – un exemple en est trouvé dans (Yu et Singh, 2003). Ainsi, il a fallu charger les agents eux-mêmes de la gestion locale de l'organisation du réseau. Pour cela, d'autant que ces agents n'ont souvent que des connaissances partielles et incertaines du système, il a été nécessaire de développer des mécanismes intégrés aux agents capables d'agir sur l'organisation même du réseau.

Ces mécanismes, que nous appelons *mécanismes organisationnels*, sont en fait des *méta-interactions*, dans le sens où ils modifient l'état interne des agents en influant sur la connaissance qu'ils possèdent du réseau dont ils font partie et donc sur les possibilités et les choix d'interactions de ceux-ci. Chaque mécanisme est lié à des interactions particulières : un mécanisme organisationnel s'applique à chaque contexte (des types d'interactions précis) de manière indépendante.

Dans le cadre des réseaux ouverts et évolutifs, comme les réseaux sociaux, dans lesquels chaque agent représente un utilisateur et est en interaction avec de nombreux autres agents, il est nécessaire pour les concepteurs du système d'être à même de contrôler la réorganisation dynamique du réseau afin d'obtenir une structure globale permettant l'optimisation des échanges et une satisfaction maximale des utilisateurs. En effet, dans un tel système, ce n'est pas la satisfaction d'un seul utilisateur (ce qui est gérable au niveau de l'agent) mais de tous les utilisateurs (ce qu'il faut prévoir au niveau du système) qui est recherchée. Pour cela, il est fondamental de connaître les mécanismes organisationnels mis en jeu dans le réseau et d'être capable de les représenter de manière globale et non seulement du point de vue de l'agent.

Nous nous proposons dans cette section de déterminer les caractéristiques essentielles de ces mécanismes organisationnels et d'en établir une classification (Lacomme, Demazeau et Camps, 2009a). Nous élaborons celle-ci dans la perspective de déterminer les points essentiels qui doivent être modélisables dans une description organisationnelle de SMA. En effet, pour être à même de proposer un modèle calculatoire cohérent incluant les effets sur l'organisation de tous les types de mécanismes organisationnels des agents, il nous paraît nécessaire d'étudier les différents types de mécanismes existants et de combiner leurs modèles lorsque plusieurs sont présents dans un réseau. Il nous faut pour cela commencer par classer ces mécanismes, afin de pouvoir étudier les caractéristiques de chaque catégorie de mécanismes séparément. Après avoir présenté les critères de classification, nous présentons une analyse, selon ces critères, des grandes familles et des principaux exemples de mécanismes organisationnels.

2.3.1 Critères de classification des mécanismes

Les organisations des systèmes multi-agents sont souvent décrites selon deux vues différentes (Hubner *et al.*, 2002) :

1) la description structurale, qui regroupe les relations entre agents (*i.e.* les possibilités d'interactions), les rôles et les groupements attribués aux agents, et qui décrit ainsi, souvent sous forme de graphes, la "forme" du système ;

2) la description fonctionnelle, qui regroupe les tâches attribuées à chaque rôle, et donc à chaque agent, leur participation dans le fonctionnement global du système, et l'intérêt pour le système des interactions mises en jeu.

> Ce constat nous permet de poser un critère de classification principal selon la portée des mécanismes : un mécanisme est *structural*[5] s'il influe sur la connaissance des autres agents et les possibilités d'interactions ; il est *fonctionnel* s'il influe sur la distribution des tâches aux agents ou des buts à accomplir pour les différents acteurs du système.

Mécanismes structuraux

Pour classifier les mécanismes structuraux (cf. Figure 2.1), nous nous attachons d'abord à déterminer si la recherche d'informations sur les autres agents est située (c'est-à-dire dépendante de l'environnement) ou non, car les moyens dont disposent les agents pour communiquer (directement ou via l'environnement) influent sur la manière dont ils peuvent découvrir ou perdre des partenaires potentiels.

> Si le mécanisme est dépendant de l'environnement, il s'agit d'un mécanisme *situationnel*. Sinon, il s'agit d'un mécanisme *social*.

Pour les mécanismes *situationnels*, la modification de la structure du système à travers l'environnement implique que l'agent puisse agir *directement* ou *indirectement* (*e.g.* par des traces laissées dans l'environnement) avec d'autres agents du système. Il peut s'agir d'un simple environnement virtuel (une simple possibilité de broadcast à tous les agents), mais aussi d'un environnement physique dans lequel l'agent a des perceptions. Les agents rencontrés dans l'environnement étant indistinctement connus ou non, il s'agit d'un mécanisme permettant de découvrir de nouveaux partenaires.

En revanche, dans le cas où le mécanisme est indépendant de tout environnement – mécanismes *sociaux* – les communications se font directement d'agent à agent, et la

[5] Nous parlons ici de mécanismes structuraux (*i.e.* dont l'objet est la structure) et non structurels (*i.e.* qui est déterminé par la structure).

découverte de nouveaux partenaires ne peut se faire que par l'intermédiaire d'un autre agent. Il s'agit donc de mécanismes impliquant la transmission d'informations organisationnelles d'agent à agent.

Nous notons que les mécanismes *structuraux* sont constitués de deux processus distincts : la découverte et l'oubli. La découverte correspond à la capacité d'un agent à ajouter de nouveaux partenaires potentiels à ceux qu'il connaît déjà, et l'oubli correspond à la capacité de supprimer des agents de l'ensemble de ses partenaires potentiels. Ces deux capacités sont souvent étroitement liées dans le cas des mécanismes *situationnels*, étant donné qu'elles dépendent souvent de conditions extérieures (capacité ou non à communiquer, à percevoir les autres agents) ; en revanche, elles sont souvent indépendantes dans les mécanismes *sociaux*, pour lesquels elles relèvent d'objectifs différents (découvertes de partenaires potentiels d'un côté, oubli des moins bons partenaires connus de l'autre, par exemple).

Mécanismes fonctionnels

Pour le cas des mécanismes fonctionnels, il est tout d'abord nécessaire de déterminer si le mécanisme s'applique dans un système où chaque agent possède des buts propres, parfois antagonistes avec ceux des autres, *i.e.* dans un cadre d'intelligence artificielle décentralisée (Demazeau et Müller, 1990b), ou si le fonctionnement du mécanisme implique que le système ait un but global et que chaque agent participe à ce but, *i.e.* dans un cadre d'intelligence artificielle distribuée.

Dans le cas où le système n'a pas de but global et où les agents peuvent posséder des objectifs antagonistes, nous voyons que l'attribution des tâches ne peut être imposée et doit être acceptée au cas par cas par chaque agent. Nous parlons alors de mécanisme de *délégation*, dans lequel chaque agent peut tenter de transférer une partie de sa fonction à d'autres agents, ou au contraire accepter pour siennes les tâches d'un autre agent.

Dans le cas où le système possède un but global et où les agents coopèrent (ou non) à sa réalisation, la distribution des tâches peut être imposée ou proposée, à travers différents types de relations entre agents, des relations hiérarchiques à des relations équitables de négociation. Nous parlons dans ce cas de mécanismes de *répartition*.

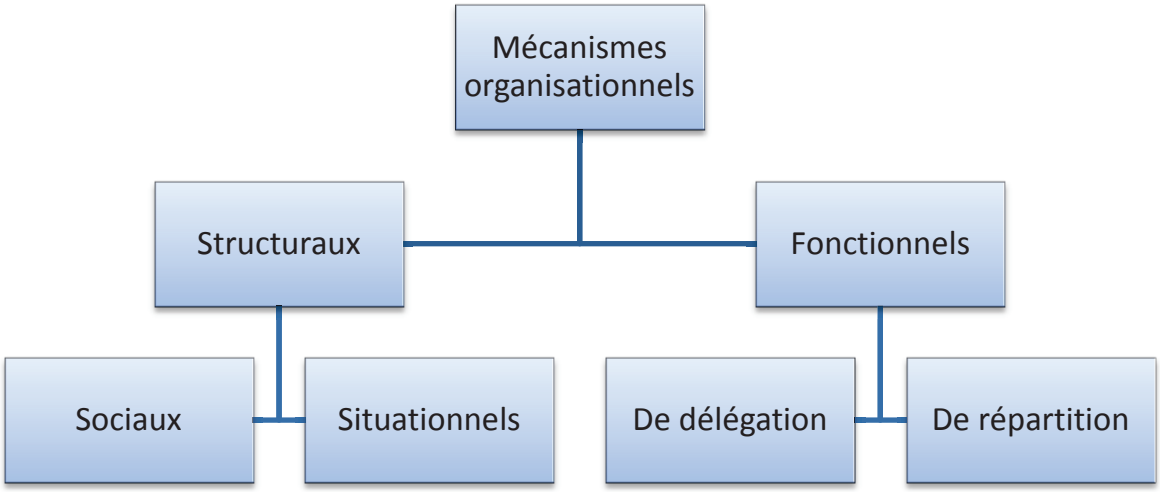


Figure 2.1 - Classification des mécanismes organisationnels

2.3.2 Mécanismes structuraux sociaux

Protocoles épidémiques

Dans des réseaux à connaissance partielle (*i.e.* dans lesquels chaque agent n'a connaissance que d'un voisinage et non de tout le réseau), il est nécessaire pour les agents de disposer d'un moyen de découvrir les autres agents qui leur sont au départ inconnus. Le premier exemple de ce type de mécanismes *sociaux* est inspiré de la propagation des rumeurs dans la société (ou des infections contagieuses dans une population, qui suivent le même modèle) : il s'agit des protocoles épidémiques (dits aussi Gossip) (Allavena, 2006).

Leur principe est simple : il revient pour chaque agent à partager régulièrement une part de la connaissance qu'il a de son voisinage avec ses voisins ; chaque voisin garde alors la partie de cette information qui l'intéresse. Ainsi, la connaissance que chaque agent a du réseau augmente progressivement, jusqu'à ce que la topologie atteigne une forme d'équilibre, prévisible au départ si tous les agents partagent le protocole. Par ailleurs, en général, lorsque les agents ajoutent un nouvel agent à leur voisinage, ils en oublient un ancien, et ce en vue de conserver un voisinage d'une taille bornée.

Relaxation restreinte

Un deuxième exemple de mécanismes *sociaux*, beaucoup plus économe en messages échangés, est le mécanisme de relaxation restreinte (Camps et Gleizes, 1995). Son principe est simple : chaque fois qu'un agent reçoit un message (requête, information, etc.) qui ne le

concerne pas ou auquel il ne peut pas répondre, il le transmet (de façon transparente, sans en changer l'émetteur) à l'agent de son voisinage qui lui semble le plus apte à répondre.

Le choix de l'agent "le plus apte" implique qu'en plus de la relaxation restreinte, l'agent dispose également d'un mécanisme *fonctionnel* permettant de choisir une interaction. L'agent contacté répondra alors à l'agent ayant envoyé le message le premier, qui aura alors connaissance d'un nouvel agent qu'il pourra ajouter à son voisinage. Afin de limiter la cardinalité des voisinages des agents du système, ce protocole est étendu en un mécanisme à processus d'*oubli*, dans lequel les voisins les moins pertinents d'un agent pour une recherche donnée sont oubliés.

Systèmes multi-agents adaptatifs

L'approche par AMAS (*Adaptative MultiAgent Systems*) vise (Camps, 1998 ; Gleizes *et al.*, 1998 ; Gleizes *et al.*, 2007) à contrôler localement les organisations des systèmes multi-agents en vue d'atteindre l'adéquation fonctionnelle (la fonction attendue du système du point de vue d'un observateur connaissant sa finalité). Il s'agit d'une approche de conception multi-agent qui s'appuie sur l'identification et la compréhension des *situations non coopératives* (SNC). Ces SNC sont des violations des méta-règles de conception suivantes : 1) tout signal perçu par un agent doit être compris par lui sans ambiguïté ; 2) l'information provenant de ses perceptions doit être utile à son raisonnement ; 3) son raisonnement doit le mener à effectuer des actions utiles aux autres et à l'environnement. La conception des systèmes vise alors à donner aux agents les moyens de percevoir, d'éviter et de pallier les occurrences possibles de SNC.

L'une des trois méthodes dont dispose l'agent pour corriger une SNC découverte (*tuning*, *réorganisation*, *évolution*), la réorganisation, est un mécanisme structural social générique[6], qui étend le procédé de *relaxation restreinte* présenté précédemment. Il consiste en effet à modifier la manière dont il interagit, c'est-à-dire l'ensemble de ses relations avec les autres agents. Il s'agit d'un mécanisme générique dans le sens ou son implémentation dépend de la spécification des SNC possibles pour le système considéré et de la manière dont il est possible de les éviter. Il a pour objectif d'optimiser, de manière locale, le fonctionnement global du système et implique la création de nouvelles relations entre agents et *oubli* d'autres relations inadéquates.

[6] Les autres méthodes agissent, l'une sur le comportement de l'agent, l'autre sur l'existence de l'agent lui-même ou la création d'un nouvel agent. Elles ne sont donc, pour nous, pas analysables dans la seule dimension organisationnelle, mais requièrent de plus l'étude de la dimension agent.

Composition de services

Un autre domaine dans lequel se sont beaucoup développés les mécanismes distribués d'organisation des réseaux est celui de la composition de services. En effet, le principe même de la composition tend à induire la présence dans les systèmes d'agents ayant pour rôle d'associer des prestataires et des clients de services pour qu'ils interagissent. Il s'agit de mécanismes dotés d'un processus d'*oubli* : les informations concernant les prestataires ne sont pas connues des clients, et donc les interactions sont "oubliées" dès qu'elles se terminent. Nous prendrons l'exemple des agents "broker" et "front-agent" (A. Badica et C. Badica, 2008).

Algorithmes de réorganisation médiats : RTS et matchmaker

Sur le même principe que les mécanismes cités précédemment, il existe un nombre important d'algorithmes décentralisés visant à réassocier les agents avec d'autres en propageant l'information au moyen d'agents "intermédiaires". Ceux-ci fournissent à l'agent considéré des informations concernant d'autres agents qui lui sont inconnus.

Nous pouvons citer l'algorithme RTS (Randomized Local Search) utilisé pour de l'agrégation d'agents similaires ou dissimilaires dans des réseaux à connaissance partielles (Michiardi *et al.*, 2007).

Un autre exemple est l'agent dit "matchmaker" pour la composition de services, qui, contrairement aux agents "broker" et "front-agent", transmet l'information pertinente des prestataires de services aux clients lors de leurs requêtes afin de leur permettre d'interagir par eux-mêmes (A. Badica et C. Badica, 2008). Il s'agit donc d'un mécanisme *social* sans processus d'*oubli*.

Algorithmes d'appariement

Nous prendrons également comme exemple le cas des systèmes multi-agents minimaux, dans lequel ont été étudiés plusieurs algorithmes d'appariement des agents et leur impact sur la topologie du réseau formé par les agents (Van Aeken, 1999). Les mécanismes d'appariement visent à trouver, pour un agent, le meilleur partenaire connu avec qui former une équipe (une paire, dans le cadre des systèmes multi-agents minimaux). Ils consistent à étudier la similitude des agents et à appareiller les agents de la meilleure manière possible pour optimiser l'équilibre et la configuration du système. Uniquement applicables dans le cadre d'agents totalement coopératifs, ces mécanismes ont la particularité d'imposer des structures de réseau très particulières aux systèmes où ils sont mis en œuvre (structures d'arbres binaires).

Agents mobiles

Enfin, nous citerons un cas particulier des mécanismes *sociaux* qui se place dans le cadre où les agents sont situés dans des environnements logiciels contrôlés. En effet, dans ce cas là, les partenaires avec qui interagir peuvent tout simplement être communiqués par un agent particulier à chaque environnement, qui dispose d'une liste de tous les agents présents dans cet environnement.

Ce cas de figure se produit dans le cadre des agents mobiles (Dorra et Nejla, 2005), qui, lorsqu'ils migrent d'un environnement vers un autre, découvrent de nouveaux partenaires d'interactions par l'intermédiaire d'un nouvel agent référent de l'environnement et oublient les anciens. Il s'agit donc d'un mécanisme avec processus d'*oubli*.

2.3.3 Mécanismes structuraux situationnels

Perception et interaction *in situ*

Le second type de mécanisme *structural* que nous proposons oblige à une vision particulière des relations entre agents *in situ*. En effet, pour de tels agent qui peuvent être des robots, communiquant à travers, par exemple, des signaux visuels (Groß *et al.*, 2006), il n'est pas possible de représenter l'organisation globale du système sous la forme d'un graphe de relations (contrairement à l'organisation de systèmes tels que les réseaux sociaux). Pour ces systèmes, les interactions se forment et se rompent avec les rencontres des agents dans l'environnement : une relation se forme entre deux agents lorsqu'ils se rencontrent et sont visibles l'un pour l'autre et se brise lorsqu'ils se séparent. Ainsi, ce sont ces aléas des rencontres et des séparations qui deviennent les mécanismes organisationnels mêmes.

Dès lors, le mécanisme organisationnel lui-même est décrit par le comportement physique (les déplacements) des agents dans l'environnement ainsi que la portée et les conditions de visibilité d'un agent pour un autre. Il s'agit de mécanismes dans lesquels le processus d'*oubli* est lié de manière forte au processus de *découverte* (puisqu'un agent "oublie" ceux qu'il ne voit plus). Des exemples typiques de mécanismes fondés sur les déplacements peuvent être trouvés dans les modèles proies/prédateurs ou dans les modèles de comportement de groupe des robots (Reynolds, 1987) : poursuite, fuite, déplacements coordonnés, déplacement en nuées, attraction / répulsion... Tous ces comportements prédéfinis permettent de modéliser comme des mécanismes organisationnels les déplacements des agents liés à la présence d'autres agents. Ainsi, les agents sont capables de pérenniser des relations établies (déplacement en groupe ou poursuite) ou de les briser s'ils

ne les satisfont pas (fuite, répulsion). Modéliser ainsi les déplacements permet donc de généraliser la notion de mécanismes *organisationnels structuraux* à des agents situés.

Stigmergie

Il existe un autre type de contact entre agents situés qui, une fois modélisé, entre également dans la catégorie des mécanismes *situationnels*. Il s'agit des contacts indirects entre agents par l'intermédiaire de modifications symboliques de l'environnement : la stigmergie (Hadeli *et al.*, 2004). La métaphore usuelle de la stigmergie est le dépôt de phéromones dans l'environnement par les fourmis d'une colonie pour signaler les emplacements de ressources disponibles aux autres fourmis. Il s'agit d'un moyen de coordonner des actions, en créant des interactions entre des agents qui ne se perçoivent pas directement dans l'environnement.

Définir le mécanisme organisationnel associé à la stigmergie revient à définir les conditions dans lesquelles le dépôt de phéromones se produit. En effet, les phéromones sont utilisées dès qu'elles sont perçues et spécifier leur utilisation revient à spécifier l'interaction elle-même ; le mécanisme organisationnel revient uniquement à la création du contact par le dépôt de phéromone, et, comme dans les exemples précédents, le déplacement des agents.

2.3.4 Mécanismes fonctionnels de délégation

Réseaux de dépendance

Contrairement aux mécanismes *structuraux*, les mécanismes *fonctionnels* n'influent pas sur les relations existantes ou à venir entre les agents mais sur leurs objectifs et donc les tâches qu'ils ont l'intention d'accomplir. La première catégorie de mécanismes *fonctionnels* à laquelle nous nous intéressons regroupe les mécanismes de *délégation*.

La délégation consiste pour un agent à se départir d'une tâche qu'il compte accomplir et à la confier à un autre agent. Il peut faire cette opération s'il n'est pas capable d'exécuter la tâche lui-même (dépendance forte) ou s'il pense qu'elle sera exécutée plus vite, mieux ou à moindre coût s'il la délègue (dépendance faible). Un réseau d'agents utilisant la délégation de tâches selon les dépendances entre agents (Sichman *et al.*, 1994) est capable d'améliorer significativement les résultats obtenus, voire de permettre aux agents d'effectuer des opérations qui leur seraient impossibles sans délégation, même coordonnés (dans le cas où les agents qui ont connaissance des tâches et ceux qui savent les accomplir ne sont pas les mêmes).

Confiance

Le deuxième, et certainement le plus significatif, exemple de mécanisme de *délégation* que nous présentons est celui des modèles de confiance. La confiance est en effet un élément de jugement de l'agent sur les autres qui est déjà pris en compte comme une méthode d'organisation dans (Foisel *et al.*, 1997) et dont il existe de nombreuses variantes.

L'utilisation d'un modèle de confiance par un agent consiste en l'évaluation des partenaires possibles connus pour une interaction donnée en fonction de la satisfaction estimée de l'agent après l'interaction (Bouron *et al.*, 2006). L'agent en déduit un classement, et donc un choix de partenaires privilégiés pour l'interaction à venir. Il s'appuie pour cela sur le modèle de confiance, qui comporte une méthode d'apprentissage à partir de sources de confiance, une méthode d'évaluation des partenaires possibles et une méthode de choix du partenaire à partir des évaluations, le tout pour chaque contexte (interaction) particulier. Un mécanisme de confiance ne permet donc pas de découvrir de nouveaux partenaires, mais permet de modifier les affectations de tâches dans le système en influant sur les partenaires choisis pour les interactions des agents.

Les mécanismes de confiance peuvent prendre différents aspects, selon que les sources de confiance utilisées par l'agent sont limitées à l'expérience directe ou bien sont au contraire étendues à des notions de réputation (Sabater et Sierra, 2001) ou à des modèles incluant des tiers de confiance (Kreps, 1990). Ils peuvent également s'associer aux mécanismes de personnalisation décrits ci-dessous pour former des mécanismes unifiés prenant ces deux aspects en compte (Lacomme, Demazeau et Camps, 2009b).

Personnalisation

Le troisième exemple de mécanisme de délégation que nous proposons, tout aussi varié dans ses formes que le précédent, est la personnalisation (Montaner, 2003).

Dans un réseau centré utilisateur (où certains agents représentent des utilisateurs réels du système), les agents peuvent posséder une représentation symbolique des préférences que les utilisateurs ont exprimées sur leurs comportements préférés, tant dans le cadre de choix de haut niveau d'objectifs et de méthodes que dans des choix plus directs comme les partenaires à privilégier. Dans la mesure où ces préférences sont utilisées pour évaluer les différentes interactions possibles et en choisir une (ce que nous retiendrons comme définition d'un mécanisme de personnalisation) il s'agit également d'un mécanisme de *délégation*. Il est intéressant de noter qu'il s'agit d'un des rares exemples de mécanismes organisationnels de la littérature qui ne soit pas prévu pour optimiser le fonctionnement du

système dans son ensemble, mais bien d'un seul agent, parfois au détriment des autres agents du réseau, voire des performances globales du système.

Contrats

Un autre protocole, utilisé principalement lorsque les réseaux considérés sont ouverts et que des enjeux (financiers, de confidentialité, etc.) sont présents est celui des contrats (Kreps, 1990). Il peut s'agit de contrats auto-surveillés, mais dans la majorité des cas les agents s'en remettent à des tiers de confiance pour surveiller les contrats et punir les fraudeurs. On peut citer des protocoles connus, comme les *contract nets* (Smith, 1980), qui se fondent sur ce mécanisme. Les contrats impliquent le demandeur de service et le prestataire, et contraignent les deux parties à des obligations. En général, il s'agit d'un paiement (sous quelque forme que ce soit) en retour d'un service rendu (*i.e.* de l'accomplissement d'une tâche déléguée).

2.3.5 Mécanismes fonctionnels de répartition

Hiérarchie pyramidale

Un premier protocole d'attribution de buts, c'est-à-dire de rôles, aux agents de systèmes entièrement collaboratifs se fonde sur des mécanismes dérivant de la notion de hiérarchie. Ces mécanismes *de répartition* consistent en la définition de relations hiérarchiques entre agents (relations qui peuvent, par ailleurs, être contrôlées par des mécanismes *structuraux*), donnant l'autorité à un agent sur un autre. L'agent qui possède l'autorité peut dès lors donner des ordres à ses "subordonnés", et ainsi déléguer des tâches du système à d'autres agents de manière impérative. Ces mécanismes fonctionnent au sein de groupes où les agents subordonnés ne sont par exemple pas capables de décider par eux-mêmes quelles sont les tâches prioritaires à exécuter (par exemple dans (Odubiyi *et al.*, 1997)). En revanche, ils ne sont prévus pour fonctionner que si les agents subordonnés obéissent aveuglément aux ordres qui leur sont donnés.

Hiérarchie complexe

Contrairement à ce qu'il est tout d'abord naturel de concevoir, les relations hiérarchiques ne forment pas nécessairement de structure pyramidale. Ainsi, il est possible que, au sein d'un système multi-agent, des relations de hiérarchies se créent et se détruisent au cours du temps selon le rôle que tient chaque agent. Ainsi, un mécanisme de *répartition* intéressant est l'attribution de tâches à des agents choisis aléatoirement dans le système, qui obtiennent en même temps le pouvoir nécessaire à imposer les tâches qu'ils ont reçues à n'importe quel autre agent (Terabe *et al.*, 1997). Ce mécanisme est également un mécanisme *hiérarchique*,

puisque le système possède nécessairement un but global et que la répartition des tâches est imposée aux autres par un certain nombre d'agents.

Répartition descendante

Une autre variante des mécanismes de *répartition* présentés existe, et a l'avantage d'être applicable dans le cadre de systèmes ouverts. Elle consiste à répartir les tâches globales au sein du système multi-agent, mais chaque agent est libre d'accepter les tâches (ou les rôles) qui lui conviennent et de refuser les autres. Les mécanismes liés à des modèles d'organisations tels que MOISE+ (Hubner *et al.*, 2002) sont fondés sur ce principe et consistent en une chaîne de délégations ou d'affectations de rôles à des agents. L'opération peut être faite *a priori* pour un modèle d'organisation défini, mais c'est l'affection *a posteriori* pendant l'exécution du système qui constitue un mécanisme de *répartition*. Dans ces systèmes, les tâches sont découpées par chaque agent qui les reçoit en sous-tâches distinctes, et chaque sous-tâche est proposée à des agents tiers avec qui l'agent est en relation. Si les agents tiers acceptent la tâche, ils obtiennent un nouveau rôle au sein du système, modifiant ainsi son organisation globale.

Négociation

Il est à noter que, contrairement à ce qui est décrit dans les mécanismes précédents, l'attribution de tâches ne se fait pas nécessairement d'agent à agent. Il est possible que tous les agents coopèrent pour déterminer lequel d'entre eux est le plus apte à exécuter chacune des tâches et quelle répartition amènera le système à fonctionner de manière optimale. Pour définir les affectations des tâches à des agents, ces derniers peuvent ainsi utiliser des protocoles de négociation ou de vote, qui permettent de déterminer une répartition optimale des tâches entre les agents capables de les remplir. Ils peuvent aussi utiliser des méthodes d'optimisation (exploration d'espaces de solutions) pour déterminer la configuration optimale du système (Prasad *et al.*, 1998). Ces mécanismes impliquent tous les agents dans la répartition des tâches, mais posent le problème d'être moins résilients que les mécanismes de répartition descendante car le résultat des négociations donne une solution optimale qui doit être suivie par tous pour être valide ; ainsi, si un agent s'engage à accomplir une tâche et ne l'accomplit pas, le système s'en trouve perturbé.

Holons

Un exemple typique de structure autorisant ce type de mécanismes organisationnels est l'holon (Fischer, 1999 ; Adam *et al.*, 2001). Il s'agit d'une structure dans lequel un groupe d'agents (nommé holon) agit collectivement au sein d'un système. Il peut par exemple être

représenté par l'un de ses agents, élu ou nommé, qui interagit pour le groupe avec l'extérieur de celui-ci. Dans un holon, l'action collective permet au groupe (ou à son représentant) de se substituer aux agents mêmes pour parler en leurs noms et accepter des tâches pour eux. Un dialogue à la fois externe (pour choisir les tâches que l'holon accepte) et interne (pour déléguer les tâches aux agents de l'holon) se crée et assure donc la répartition des tâches dans l'holon, généralement à travers des principes de négociation et de vote.

Tableau 2-C - Exemples classifiés de mécanismes organisationnels

Catégorie	Sous-catégorie	Exemples
Structural	Social	Agents mobiles Algorithmes d'appariement Composition de services (broker, front-agent) Protocoles épidémiques Relaxation restreinte RTS / matchmaker Systèmes multi-agents adaptatifs
	Situationnel	Perception et interaction *in situ* Stigmergie
Fonctionnel	Délégation	Confiance Contrats Personnalisation Réseaux de dépendance
	Répartition	Hiérarchie complexe Hiérarchie pyramidale Holons Négociation Répartition descendante

2.3.6 Synthèse sur les mécanismes organisationnels

Il est remarquable de noter que les mécanismes organisationnels sont très variés (cf. Tableau 2-C), tant dans leur forme que dans leurs objectifs, allant de l'optimisation du réseau global (protocoles épidémiques) à l'amélioration de l'adéquation des résultats d'un agent avec les besoins de l'utilisateur qu'il représente (personnalisation). Tous ces mécanismes sont mis en jeu dans des types de réseaux très différents, allant du réseau social, ouvert et à connaissance partielle, à des systèmes de résolution de problème, fermés et à connaissance totale. Au vu de la nécessité de contrôler de tels réseaux au niveau des agents, nous avons guidé notre étude, au contraire de la majorité des travaux récents (V. Dignum, 2009), vers le

point de vue local d'une étude des mécanismes organisationnels applicables à tous les types d'agents et de réseaux.

La classification (cf. Figure 2.1) ainsi obtenue nous permet, par la suite, d'identifier les mécanismes mis en jeu dans un système multi-agent comme une somme de mécanismes élémentaires appartenant à l'une des catégories distinguées dans ce chapitre. L'idée sous-jacente est de combiner les analyses disponibles pour chaque type de mécanisme organisationnel connu afin de calculer l'effet global sur l'organisation du système de l'ensemble des mécanismes présents dans le réseau. Il apparaît nécessaire pour cela de dissocier les mécanismes structuraux et les mécanismes fonctionnels, en cela qu'ils ne manipulent pas les mêmes éléments et que les seconds sont dépendants des premiers. En revanche, il ne paraît pas nécessaire de distinguer les sous-catégories de mécanismes que nous avons précisées, car elles s'attachent à des différences sémantiques plutôt qu'à de réelles différences d'expression ou de représentation.

2.4 Lacunes et problématiques de l'état de l'art

Nous voyons, par comparaison entre les capacités expressives des modèles organisationnels étudiés et le développement varié des mécanismes organisationnels dans la conception de SMA, que la modélisation et la conception ne se rejoignent que sur la partie statique des organisations. En effet, si les organisations statiques sont bien étudiées et formalisées par de nombreux modèles, ceux-ci ont une composante dynamique très limitée et souvent incapable de représenter correctement la complexité des mécanismes d'auto-organisation et de réorganisation. La raison nous semble être que ces modèles organisationnels sont souvent fondés sur des concepts organisationnels provenant d'organisations humaines ou des architectures d'agents qui font référence au raisonnement humain (comme l'architecture BDI), et qu'ils ne prennent donc qu'imparfaitement en compte les problèmes de dynamique, tant au niveau de la population d'agents des systèmes (ouverture) qu'au niveau de la structure et de la fonction du système. Enfin, la grande variété de mécanismes organisationnels implique, pour des descriptions d'organisations dynamiques, un grand choix au niveau de la conception, qui doit être compensé par des possibilités d'évaluation et de comparaison des organisations entre elles lors de leur description et avant toute implémentation du système.

Ainsi, il existe un manque au niveau de la représentation, formelle et générique, de la dynamique organisationnelle. Ce manque se retrouve dans la difficulté de trouver un modèle de représentation adéquat pour la comparaison et l'évaluation de multiples organisations

dynamiques de SMA, souvent conçues de manière *ad hoc*, hors des modèles organisationnels usuels.

Chapitre 3. Notre approche

Dans cette partie, nous définissons les bases de l'approche suivie pour construire un modèle organisationnel des SMA. Cette approche provient des incomplétudes de l'état de l'art, des besoins expressifs des SMA ouverts et auto-organisés, et vise à fournir des méthodes et critères de construction qui guideront le processus de création et d'utilisation de notre modèle. Elle se développe suivant deux axes :

1) la représentation de l'émergence comme donnée fondamentale de la description des SMA ;
2) la possibilité du calcul sur les organisations comme nécessité pour leur analyse.

3.1 Une approche émergentiste

Notre objectif est de développer un modèle organisationnel générique pour les SMA. C'est à travers la prise en compte de l'émergence pour guider la construction de notre modèle que nous nous démarquons de l'état de l'art.

Comme nous l'avons vu, les modèles existants prennent appui sur les notions organisationnelles de la société humaine, qui sont des notions de haut niveau, décrivant des abstractions déjà complexes, et, par là même, ils émettent des présupposés sur les agents du système. La description des organisations à travers ces modèles est limitée en expressivité à cause de deux points essentiels :

- Le premier est la faible prise en compte des systèmes ouverts et des systèmes auto-organisés : la description de la dynamique, la possibilité d'apparition d'agents inconnus et la description des mécanismes d'auto-organisation sont en effet les points faibles de ces modèles.
- Le second est la corrélation entre les notions mises en jeu et des architectures internes des agents qui leur correspondent. Ainsi les notions de rôles ou de coalitions sont généralement liées à des architectures particulières d'agents cognitifs, comme par exemple les agents BDI (Rao et Georgeff, 1995). Ces corrélations limitent la portée du modèle en rendant difficile ou sémantiquement faible la description de systèmes contenant d'autres types d'agents.

Notre approche, fondée sur la méthode VOYELLES (Demazeau, 1996) et sur une vision émergentiste du multi-agent, nous pousse à nous préoccuper d'abord des systèmes eux-mêmes pour en extraire les composants essentiels d'une organisation. Il est ainsi possible de

s'abstraire des notions usuelles pour décrire de manière plus générique les organisations et leur dynamique.

3.1.1 L'émergence

Le premier point important de notre représentation des SMA est l'émergence. Elle est définie dans (Deguet et Demazeau, 2005) comme :

> un gain de complexité observable entre : i) un premier processus responsable de la production d'un phénomène inscrit dans une trace d'exécution et ii) un second processus chargé de sa détection

L'émergence décrit le processus selon lequel un ensemble d'entités interagissant est capable de produire un comportement global et donc une fonctionnalité globale plus importante que la somme des fonctionnalités locales. Elle est uniquement mesurée à travers l'**observation**. Il s'agit d'un élément fondamental du domaine des SMA, car la répartition de l'intelligence artificielle en entités indépendantes et autonomes a pour effet de permettre, justement, l'émergence d'une nouvelle forme d'intelligence collective à travers la construction des échanges entre entités.

> *Ainsi, dans l'exemple de la fourmilière, si chaque fourmi n'est capable que de comportements basiques, et par ailleurs, d'une communication basique (le dépôt de phéromones), les fonctions du système lui-même, comme la création de pistes approchant le chemin le plus court entre le nid et les sources de nourriture est un comportement complexe qui résulte de l'agrégation des comportements individuels et des interactions entre les fourmis. Les fonctions résultantes du système, visant à la création et au maintien du fonctionnement d'une fourmilière, y compris à travers l'optimisation de l'apport en nourriture, sont liées à ces processus émergents et en dépendent directement.*

Il existe plusieurs niveaux d'émergence. La première distinction se fait selon que la structure causale est conservée du niveau le plus bas au plus haut, c'est l'émergence dite « faible », ou que le lien causal n'est, tout du moins en apparence, pas conservé, c'est l'émergence dite « forte » (Deguet *et al.*, 2006). Nous nous plaçons hors du débat sur l'existence de l'émergence forte, en considérant que, puisque, au résultat, tout le comportement des systèmes est contenu dans le code des agents, qui sont les entités actives, il ne s'agit ici que d'émergence faible.

En revanche, une seconde distinction se fait selon que l'émergence est observée verticalement ou horizontalement (Deguet, 2008) :

- l'émergence de structures de haut niveau : les phénomènes observés à haut niveau ne peuvent être **déduits** des constituants et processus de bas niveau ;
- l'émergence relative à un modèle : l'apparition d'un comportement haut-niveau du système ne peut être expliquée par le modèle décrivant ce système.

Puisque nous décrivons des systèmes conçus autour de l'association d'entités interagissant, et puisque nous créons nous-mêmes le modèle global du système, ce qui nous intéresse ici, c'est donc la description d'une émergence du point de vue des interactions, et plus précisément des possibilités et des règles d'interactions entre les agents (les entités actives) du système. Il s'agit donc d'une émergence de structures de haut niveau. Selon ce point de vue, l'émergence est la capacité du système à engendrer, à travers les interactions entre agents, un comportement global complexe. Ce comportement peut être voulu ou bien simplement découvert « accidentellement ».

Puisque nous souhaitons l'utiliser ici dans le cadre d'un système descriptif des organisations, nous retenons comme définition de l'émergence celle de (Deguet et Demazeau, 2005), qui correspond le mieux avec le point de vue de l'observation du système. Elle correspond à l'observation de l'apparition de fonctionnalités de haut niveau complexes à partir de la mise en interaction de sous-systèmes impliquant des fonctionnalités de bas niveau moins complexes. Formellement, nous pouvons définir l'émergence par une simple réécriture de l'équation fonctionnelle (Demazeau, 1995) de la méthode VOYELLES, équation dans laquelle elle est nommée *fonction collective*.

> Dans ce cadre, l'émergence se définit comme la différence entre la fonction du système et la somme des fonctions des éléments du système :
>
> $$Em = F_{sys} - \sum_{a \in sys} F_a$$

où F_a est la fonction remplie par l'agent a, F_{sys} la fonction totale remplie par le système sys, et Em la mesure de l'émergence fonctionnelle du système sys. Cette différence de fonction provient des possibilités d'interactions entre les éléments du système, et est donc représentée par la spécification de l'organisation du système. C'est donc à partir de ce principe et de ses implications sur la description des systèmes que nous fondons notre modèle organisationnel.

3.1.2 Les niveaux de représentation

La complexité descriptive de la résultante globale des interactions entre les éléments d'un SMA rend celui-ci difficile à décrire dans son ensemble par un modèle unique. En effet, les comportements et structures observables sur le système dans son ensemble sont généralement très différentes des structures et des comportements des agents eux-mêmes. C'est pourquoi plusieurs niveaux de description (cf. Figure 3.1) peuvent être définis et utilisés dans la conception d'un modèle.

L'hétérogénéité des comportements et des fonctions simples, mais également de ceux qui découlent de l'émergence rend nécessaire la description selon différents niveaux.

> *Il n'est par exemple pas aisé de définir dans un même modèle les comportements et les capacités des fourmis et les structures et mécanismes de fonctionnement observables de la fourmilière dans son ensemble. Si les premiers sont facilement descriptibles sous la forme de mécanismes réactifs, et donc d'automates à états finis, la fourmilière, dans son ensemble, possède des entrées pseudo-aléatoires (provenant de l'environnement) et respecte des contraintes précises : son état (remplissage des réserves, création et stabilisation des pistes,...) réagit ainsi plutôt comme un système à asservissement. Ainsi, le modèle liant ces deux descriptions est loin d'être trivial.*

Ainsi, nous pouvons définir plusieurs niveaux de description en nous inspirant des définitions existantes (House *et al.*, 1995 ; Klein et Kozlowski, 2000) :

- niveau **micro** : il concerne les éléments descriptifs locaux – autrement dit, atomiques ; il comprend les agents eux-mêmes, leur comportement propre, leurs actions sur l'environnement, les communications qu'ils mettent en place entre eux, et leur représentation d'eux-mêmes et des autres.
- niveau **méso** : il concerne les phénomènes d'agrégation des éléments locaux, qui mènent à l'émergence ; il comprend les notions d'adaptation et d'apprentissage, les modifications de l'environnement, les interactions entre agents (en tant que communications suivies et non instantanées), la permanence d'interactions entre agents, le partage de tâches ou d'objectifs, les déplacements communs (pour des agents situés),...
- niveau **macro** : il concerne les propriétés émergentes résultantes ; il comprend l'évolution temporelle des agents, l'effet du système sur son environnement, les schémas d'interaction résultants (problèmes de saturation, traces,...), la fonctionnalité même du système, son organisation globale et ses évolutions dans le temps.

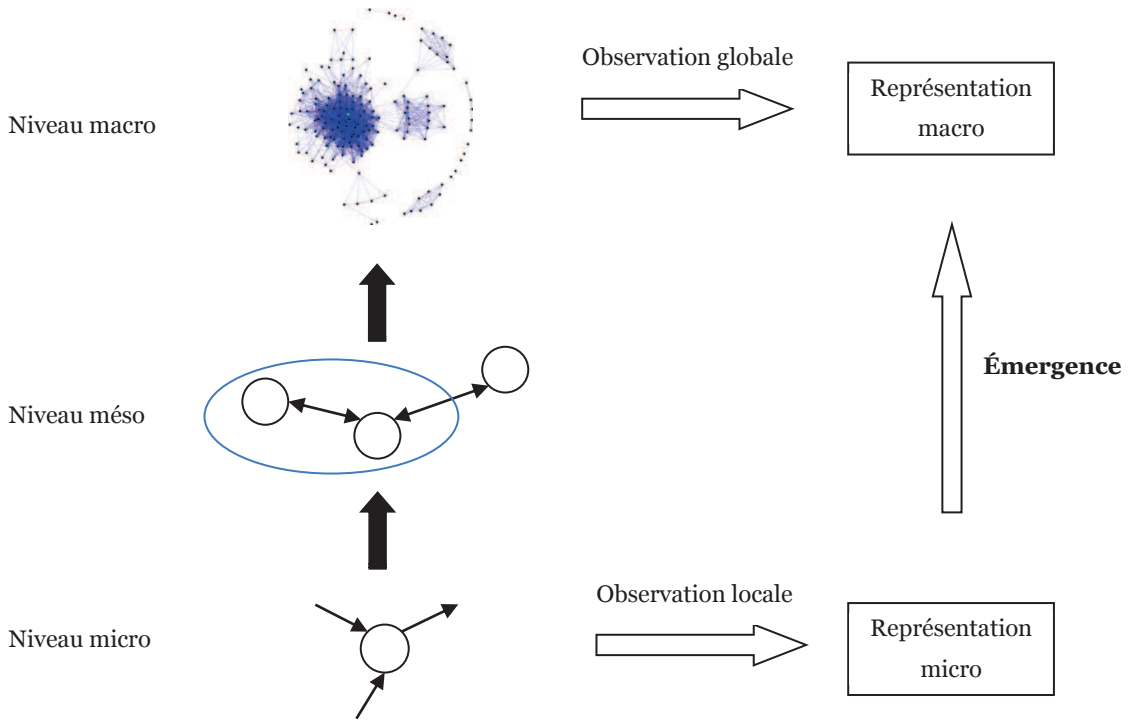


Figure 3.1 - Représentation de l'émergence et des niveaux de description

À ces différents niveaux de description, nous cherchons à faire correspondre des éléments de description différents. Nous répartissons ces éléments selon les différentes dimensions de la méthode VOYELLES, qui nous sert à distinguer les composantes descriptives des SMA. Une manière simple de définir le niveau auquel appartient un élément descriptif est le nombre d'agents qu'il implique dans la description : le niveau micro correspond à la description d'un seul agent, le niveau méso de plusieurs agents, et le niveau macro du système ou d'un sous-système.

Niveau micro

- *Agents* : architecture interne, états, comportement, représentation de soi, représentation et traitement de l'information.
- *Environnement* : description des artéfacts (entités passives), topologie, situation absolue des agents (positions), déplacements.
- *Interactions* : canaux de communication, protocoles d'interactions, langages de communication.
- *Organisations* : relations entre agents, réorganisation locale des voisinages, répartition de buts et de tâches.

Niveau méso

- *Agents* : synchronisation, représentation d'autrui.
- *Environnement* : situations relatives des agents (distances), déplacements en groupe, modifications coordonnées, dépôts de phéromones, interactions partagées avec les artéfacts.
- *Interactions* : engorgement des communications, négociations, alignements des ontologies, intercompréhension.
- *Organisations* : formation ou existence d'équipes, de coalitions, délégation, partage des représentations et jugements sur les autres agents entre plusieurs agents.

Niveau macro

- *Agents* : représentations propres du système, réflexivité du système.
- *Environnement* : créations de structures dans l'environnement, altération durable de l'environnement, distribution et consommation des ressources, positionnement dans l'environnement.
- *Interactions* : occurrences et fiabilité des interactions, quantité de données échangées intérieurement et extérieurement.
- *Organisations* : structure et fonction globale du système, possibilités et évolution des interactions, répartition de la fonction du système, auto-organisation et réorganisation, intégrité fonctionnelle (Kozlak, 2000).

Il est ensuite possible d'utiliser ces éléments descriptifs pour construire des descriptions des systèmes qui soient multi-échelles, et qui prennent ainsi en compte les spécificités à la fois locales et globales du système.

Dans l'optique de construire un modèle descriptif des organisations et de leurs dynamiques pour les SMA, nous nous intéressons ici plus particulièrement à la dimension organisationnelle des systèmes, et souhaitons donc développer un modèle apte à décrire les différents niveaux de spécification et d'observation des systèmes.

3.1.3 Une construction bottom-up

L'objectif est de construire un modèle descriptif prenant en compte les trois niveaux descriptifs distingués ci-dessus. Pour arriver à cette capacité descriptive, il est nécessaire que le modèle soit construit à partir d'éléments simples, c'est-à-dire bas-niveau. En effet, s'il est possible d'agréger des éléments bas-niveau pour obtenir des descriptions de plus haut niveau, l'opération inverse est impossible dans de nombreux cas : les éléments haut-niveau, tels que ceux proposés par la majorité des modèles organisationnels (rôles, groupes,...) ne peuvent

être décomposés que selon une architecture d'agents associés (suivant des plans, des raisonnements associés, par exemple, pour une architecture BDI) ; il n'est donc pas possible d'exprimer de manière générique des propriétés bas-niveau avec de tels éléments, d'autant que les éléments bas-niveau obtenus par décomposition n'existent pas indépendamment des notions haut-niveau auxquelles ils appartiennent.

Dans le sujet qui nous intéresse, les éléments bas-niveau sont ceux du niveau micro et définissent ainsi des propriétés locales. Plutôt que d'imposer, comme dans les autres modèles organisationnels, des éléments haut-niveau dérivés des organisations humaines, il nous faut identifier des éléments locaux permettant la description au niveau micro des organisations. Ces éléments doivent permettre la description statique (c'est-à-dire les invariants) mais aussi dynamique (l'évolution possible au cours du temps) des organisations ; il peut donc y avoir besoin de plusieurs types différents d'éléments dédiés réciproquement à la description statique et dynamique des organisations. Puis, il nous faut leur ajouter des opérateurs d'agrégation permettant de décrire les propriétés méso et macro.

Une organisation peut donc être décrite, en suivant cette méthode, comme une somme :

- d'éléments locaux – représentant des propriétés micro ;
- d'opérateurs d'agrégation – représentant des propriétés méso ;
- de la résultante de l'agrégation d'éléments locaux – représentant des propriétés macro.

C'est l'agrégation des éléments locaux qui permet de dériver les propriétés macro. Ceci correspond à notre vision émergentiste des SMA en cela que c'est par la somme des propriétés locales (comportements et interactions) que les propriétés globales apparaissent : c'est aussi par la somme (l'agrégation) des propriétés locales que nous décrivons les propriétés globales. C'est la complexité des possibilités d'interactions entre agents qui empêche de décrire le système d'un point de vue seulement local. Ainsi, l'agrégation des propriétés locales est la correspondance formelle à cette montée en complexité des niveaux méso et macro.

Or, il apparaît que les propriétés locales sont contenues, de fait, dans le code même des agents. Cela implique que si la description des organisations se fait par une spécification totale des propriétés locales au niveau du code des agents, il n'est pas possible de décrire les organisations indépendamment d'une implémentation spécifique ; il existerait alors une corrélation entre le code et la description des organisations. Pour éviter cette corrélation, il nous faut affirmer l'existence des concepts organisationnels de bas niveau indépendamment du code même des agents, c'est-à-dire des agents eux-mêmes. Nous atteignons cela en posant

comme hypothèse supplémentaire l'indépendance des différentes dimensions descriptives des SMA.

3.1.4 L'indépendance de la description organisationnelle

Il nous parait en effet important de souligner la nécessité de l'indépendance, dans notre modèle, de la description organisationnelle vis-à-vis de toute autre partie de la description des SMA.

Pour être capables de décrire complètement les SMA, de les comparer entre eux et d'analyser leurs capacités, leurs limites et leurs fonctionnalités, il est nécessaire d'avoir accès, comme pour toute expérience, à un panel de **variables indépendantes** sur lesquelles agir, et desquelles dérivent un autre ensemble de variables observables permettant d'identifier les propriétés recherchées sur le système. Les variables indépendantes manipulables sont pour nous les éléments descriptifs de nos modèles de SMA : elles permettent de décrire et de spécifier un SMA à partir du modèle. Ces éléments de base doivent être indépendants, pour garantir la portée expressive du modèle, et la possibilité d'agrégation des éléments de niveau micro en propriétés de niveau macro.

> *Il est par exemple nécessaire de pouvoir spécifier, pour les systèmes ouverts, des règles de comportements sociaux des agents, sans pour autant connaître à l'avance le nombre et le comportement exact des différents agents du système. Sans cela, il n'est pas possible de décrire les structures possibles du réseau d'agents formé au sein du système sans connaître les détails de l'implémentation exacte de l'organisation.*

C'est cette nécessité qui implique l'existence de dimensions indépendantes clairement identifiables pour la description des systèmes : l'étude et la spécification d'un système ne peuvent être complètes que s'il est possible d'isoler les différents éléments constitutifs qui influent sur sa fonction, sur son fonctionnement. Nous avons choisi comme dimensions celles proposées par la méthode VOYELLES. Cette dernière propose en effet, à notre connaissance, la seule distinction claire et pragmatique entre différentes dimensions des SMA : les agents (A), l'environnement (E), les interactions (I) et les organisations (O). Cette distinction est déjà suivie par plusieurs modèles de descriptions (Demazeau et Costa, 1996 ; Ricordel et Demazeau, 2001), qui ont prouvé la cohérence et la pertinence des dimensions définies.

L'indépendance des dimensions descriptives ainsi définies les unes par rapport aux autres dans le processus de formation d'un modèle complet d'un SMA correspond donc à notre hypothèse complète. Cette hypothèse est formulée, tout au moins pour les dimensions A et O, dans le modèle PopOrgs (Demazeau et Costa, 1996). Il est en effet exprimé dans ce modèle que la description d'une organisation elle-même est différente et indépendante de la

description de la population d'agents qui l'implémente dans un cas précis (cf. Figure 3.2). Il n'est donc pas nécessaire de connaître les agents du système, ni leur implémentation pour décrire l'organisation de celui-ci. Cela permet deux choses :

- de décrire de manière générale un ensemble de systèmes possibles partageant une même organisation ;
- de décrire des systèmes dans lesquels les agents ne peuvent pas être connus à l'avance : les systèmes ouverts, dans lesquels des agents entrent et sortent du système à volonté sans qu'il y ait nécessairement un contrôle sur la provenance ou le code de ces agents.

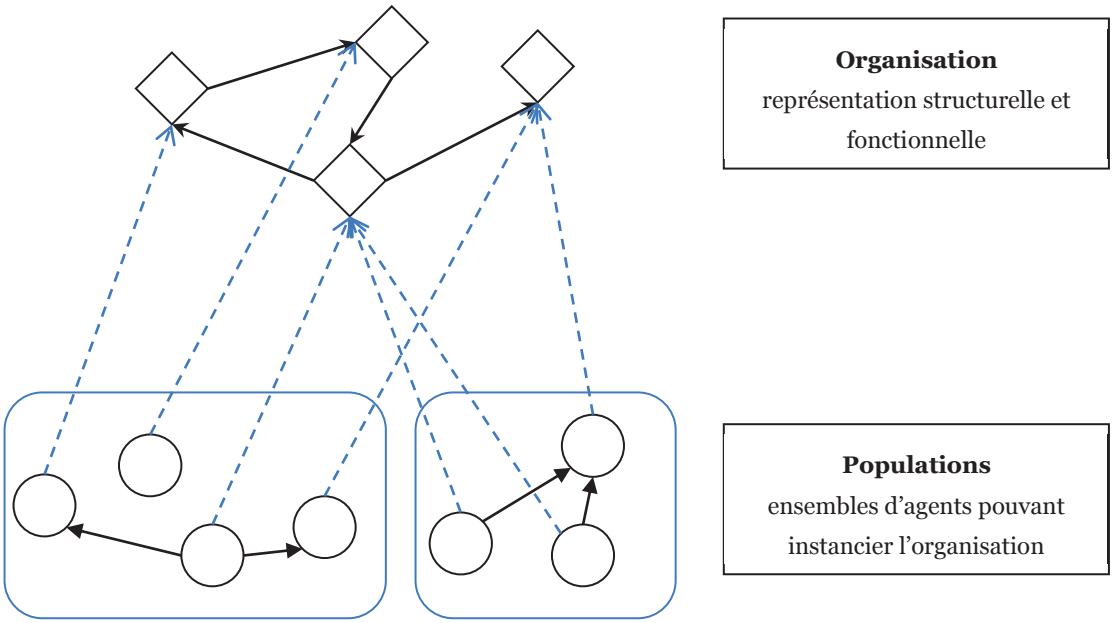


Figure 3.2 - Représentation du principe d'indépendance organisation - population

Par ailleurs, l'indépendance vis-à-vis de l'implémentation des agents, donc de leur architecture interne, permet de spécifier des systèmes en toute indépendance des méthodes utilisées pour créer les agents eux-mêmes.

> *Ainsi, une même organisation peut être instanciée (peuplée) par des agents réactifs ou au contraire par des agents BDI sans que la description organisationnelle n'en soit modifiée.*

Dans ce travail, nous faisons donc nôtre cette hypothèse d'indépendance entre dimensions descriptives agent (A) et organisation (O) des SMA. Il est à noter que, bien que cela paraisse plus évident, la dimension organisationnelle des SMA se doit aussi d'être indépendante des autres dimensions de la description des SMA (E et I). Nous proposons donc

un modèle organisationnel qui vise à rester indépendant de l'ensemble des autres dimensions de la méthode VOYELLES.

3.2 Une approche calculatoire

Le second aspect de l'approche que nous proposons pour la description des organisations est d'être calculatoire : nous cherchons à rendre possible l'expression de propriétés globales sur les organisations à travers le calcul sur les éléments à notre disposition.

Dans un modèle qui suit la structure émergentiste que nous proposons, les propriétés du niveau micro sont le lieu de la spécification, tandis que c'est au niveau macro que la description des structures profondes et des fonctions du système est la plus lisible. Le principe d'émergence impliquant une montée en complexité des phénomènes décrits avec la montée du niveau de description, la description des propriétés méso et macro nécessite une méthode pour sélectionner et agréger l'information. C'est donc à travers le calcul que la possibilité d'exprimer ces propriétés se présente.

3.2.1 Évaluer et calculer

Notre second objectif est donc de construire un modèle autorisant l'existence de méthodes permettant de formaliser une notion de calcul sur les organisations : l'obtention d'indications sur les propriétés globales à partir de spécifications des propriétés locales.

La raison derrière cet objectif est le manque de possibilités de calcul sur les organisations dans les modèles existants (cf. section 2.2.3). Le plus souvent, les descriptions des systèmes auto-organisés, ou plus généralement des SMA à forte dynamique sont effectuées de manière ad hoc, sans modèle générique pour fournir la possibilité de comparaison ou d'analyse profonde des modèles proposés. La cause de cette disparité des descriptions est le manque d'adéquation entre les mécanismes réellement employés dans ces systèmes et les possibilités expressives des modèles d'organisations. Ces modèles haut-niveau n'offrent en effet pas les moyens de calculer, à partir d'une description organisationnelle, des propriétés statiques et dynamiques sur cette organisation.

Ainsi, il apparaît nécessaire de créer un modèle qui propose des méthodes permettant d'exploiter ses éléments dans un but calculatoire, pour être à même d'obtenir des informations supplémentaires sur les organisations par le calcul de propriétés globales des systèmes. Cet objectif pose donc le problème de la calculabilité de ces propriétés, dans le cadre de l'analyse *a priori* des organisations, c'est-à-dire l'analyse de l'organisation elle-même et des différents systèmes qu'elle peut engendrer, sans s'attacher à des instanciations ou des simulations spécifiques de ces systèmes.

3.2.2 Peut-on calculer l'émergence ?

La complexité du principe même d'émergence et la découpe multi-niveau de la description des systèmes mènent à un problème de complexité vis-à-vis du calcul de propriétés globales de ces systèmes qui peut être résolu en simplifiant l'expression de ces propriétés à travers l'utilisation des probabilités.

L'objectif des calculs organisationnels est de calculer les états possibles d'un système créé en accord avec une description organisationnelle préétablie. Plusieurs problèmes apparaissent face à cet objectif.

- Le premier d'entre eux est la **complexité calculatoire** : dans un système à n agents, chaque agent pouvant être dans p états différents, la grandeur d'un voisinage d'agent est de l'ordre de $O(n)$, et une transition de voisinage d'un instant t à $t+1$ est une fonction sur un espace de grandeur $O(np)$; sa résolution dans le temps est donc d'ordre $O(np\,\Delta t)$, où Δt est le nombre de pas temporels. La résolution simultanée pour tout le système (dans un système simple, donc, où les états des agents sont en nombre bornés) est donc un calcul de complexité $O(n^2 p\,\Delta t)$.
- Un deuxième problème est celui de l'**incomplétude des données**. En effet, dans la plupart des cas, il sera intéressant de connaître certaines propriétés du système sans connaître tous les détails de son implémentation : par exemple, sans savoir combien il contient d'agents, ou bien quel est son état initial…
- Le troisième problème qui apparaît est l'existence de **dépendances** entre les mécanismes organisationnels. Ces dépendances compliquent d'autant le calcul qu'elles peuvent exister au sein d'un agent, mais aussi entre agents. C'est donc un réseau d'états et de voisinages multiples qu'il faut faire évoluer au cours du temps pour calculer les propriétés désirées, ce qui revient presque à simuler le comportement du système dans un cas précis. Or, il est souhaitable d'être capable de prédire des propriétés des systèmes, même lorsque leur organisation est sous-spécifiée et que l'implémentation n'est pas décidée.

Cet ensemble de problèmes quant au calcul des propriétés rend difficile la possibilité de calcul exact sur les mécanismes ; d'autant plus avec l'augmentation du nombre d'états possibles pour les agents (par exemple pour des agents cognitifs dotés de mémoire), du nombre d'états possibles du voisinage (par exemple pour des mécanismes utilisant des variables réelles, comme certains protocoles de confiance), et du nombre d'agents possibles dans le système. Les solutions pour calculer sur de tels espaces apparaissent comme étant la logique floue et les probabilités. La logique floue est un paradigme approprié pour l'expression d'inexactitudes ou de raisonnements. Or ici, c'est plutôt incertitudes et

occurrences d'évènements que les calculs doivent prendre en compte. Et ces conditions de calcul poussent naturellement à l'usage des probabilités comme moyen d'exprimer la possibilité d'occurrence de comportements et de situations spécifiques dans un système qui n'est pas entièrement connu et où ce qui est inconnu peut être représenté comme de l'aléatoire (tout ce qui n'est pas spécifié dans l'organisation pouvant être fixé à l'implémentation où laissé libre à l'exécution, et étant donc aléatoire). Il est ainsi possible d'utiliser les probabilités pour calculer, à partir d'une spécification organisationnelle locale, des propriétés globales des systèmes correspondants. Il s'agit donc de déterminer comment le modèle peut permettre d'utiliser les probabilités dans l'expression de spécifications locales, et quelles propriétés globales peuvent être calculées et sont significatives pour l'analyse des organisations.

3.2.3 Cas-test et probabilités

Pour calculer des propriétés significatives sur les systèmes représentés par une organisation donnée, plusieurs approches sont possibles : le calcul de propriétés générales (un des objectifs de notre modèle étant de le rendre possible), le calcul de propriétés plus spécifiques selon des choix d'implémentation précis (cf. section 2.2.3) ou plus précisément encore, le calcul de propriétés suivant des cas-tests (Nguyen *et al.*, 2008). Chacune de ces méthodes est appropriée pour des cas différents, selon les caractéristiques des systèmes étudiés, et surtout selon les propriétés globales recherchées.

Les propriétés que nous proposons de calculer à partir de notre modèle sont les propriétés méso et macro liées aux organisations. Elles se distinguent en deux catégories distinctes : les propriétés statiques et les propriétés dynamiques. Les propriétés statiques décrivent l'état possible du système à un instant déterminé : composition des voisinages, répartition de la fonction du système entre les agents, topologie du réseau induit par les relations entre agents. Les propriétés dynamiques décrivent les évolutions possibles du système au cours du temps : vitesse et modes de réorganisation du réseau induit, changements dans la fonction du système et dans sa répartition.

Le calcul de propriétés a pour objectif de fournir des estimations plus ou moins spécifiques du comportement moyen du système dans une situation donnée. Ainsi, des comportements moyens peuvent être obtenus à partir de la simple description organisationnelle du système. Par ailleurs, en se rapprochant du cas-test, par une spécification de plus en plus précise de l'implémentation du système et des conditions initiales de son exécution, il est possible d'analyser plus précisément les comportements possibles du système dans certaines conditions, et ainsi de tester, de comparer et d'évaluer les organisations par rapport à des situations ou des objectifs précis.

3.2.4 Une approche calculatoire est possible

L'utilisation appropriée d'éléments bas-niveau pour construire notre modèle organisationnel doit donc nous ouvrir une possibilité supplémentaire : la capacité de produire, dans un but d'analyse, de test et de comparaison, une estimation des propriétés globales de tous les systèmes engendrés par une organisation définie.

L'indépendance de la description organisationnelle vis-à-vis des autres dimensions descriptives des SMA, combinée à un modèle algorithmique de la dynamique des organisations fondé sur des éléments de niveau micro, permettent d'envisager des méthodes de réexpression des éléments dynamiques sous forme probabiliste. De l'existence de ces méthodes découle la possibilité d'une évaluation fonctionnelle des organisations, *a priori* de toute implémentation, à travers l'estimation de propriétés globales, dans des cas généraux ou plus spécifiques, jusqu'au cas-test.

Ainsi, nous visons à proposer un modèle organisationnel permettant à la fois une expression générique des organisations des SMA, y compris des systèmes ouverts, et de leur dynamique, mais permettant également l'analyse et la comparaison d'organisations à travers le calcul de propriétés globales sur ces organisations mêmes.

Chapitre 4. Modèle organisationnel

A partir des multiples aspects que nous avons définis pour notre approche, il nous est donc possible de discuter et de sélectionner les différents éléments et les différentes méthodes qui nous seront utiles pour former un modèle générique complet des organisations et de leur dynamique.

La première étape de cette construction passe par l'identification des éléments atomiques qui sont associés, dans notre modèle, aux différentes notions manipulées par les organisations. Il s'agit donc de déterminer à la fois les éléments descriptifs de la constitution d'une organisation et les éléments descriptifs de sa dynamique. Nous introduisons en même temps les premiers éléments de notation en vue d'une écriture formalisée du modèle.

Les étapes suivantes de notre démarche correspondent à plusieurs points importants. Tout d'abord, il s'agit de l'identification de la description formelle des éléments dynamiques, plus complexes à appréhender que les éléments statiques. Puis il s'agit de mettre en place une méthode de description formelle des organisations à partir des éléments identifiés, à travers la description des ensembles de propriétés nécessaires à la description correcte d'une organisation. Enfin, la comparaison avec les modèles de l'état de l'art nous amène au questionnement de la formalisation, au sein de notre modèle, des notions usuelles des organisations que sont, par exemple, les rôles et les groupes.

4.1 Éléments du modèle

Notre approche nous conduit donc à commencer la construction de notre modèle organisationnel (Lacomme, 2011) par l'étude du niveau micro, afin d'en extraire les concepts nécessaires et suffisants pour la description organisationnelle des SMA, tant sur le plan statique que sur le plan dynamique.

Suivant notre approche émergentiste (cf. section 3.1), la première étape de l'élaboration d'un modèle des organisations est pour nous l'identification des moyens de description de l'organisation au niveau micro, c'est-à-dire, en considérant l'organisation du point de vue d'un unique agent. En gardant à l'esprit que, suivant notre hypothèse de séparation des dimensions descriptives, nous ne supposons rien sur la structure interne des agents, nous ne supposons en particulier pas de compétence cognitive particulière. Il est donc impossible de représenter, au niveau micro, l'organisation comme un ensemble de concepts cognitifs complexes, tels que des obligations, des contrats ou des attitudes comportementales, etc. Il devient donc nécessaire d'appuyer la description organisationnelle sur des concepts

élémentaires capables d'étayer la réalité du système représenté en faisant abstraction des autres dimensions descriptives – des concepts concrets.

L'objectif que nous poursuivons nous amène donc à faire, en suivant celle que proposent (Demazeau et Costa, 1996) dans le modèle PopOrg, une distinction entre le modèle organisationnel et son instanciation – ou sa « population » – par un ensemble d'agents. Ce faisant, il apparait donc que nos entités élémentaires – du niveau micro – doivent donc exister sous deux modalités différentes : celle du concept organisationnel et celle de son actuation dans une instance de système. Pour ce faire, nous utilisons donc la notion de type. En effet, à travers la définition d'entités typées, nous pouvons concevoir à la fois le concept organisationnel – le type – et son instanciation dans un système réel – l'entité considérée elle-même.

Notre première étape se résume donc en l'identification des entités typées permettant la description d'une organisation multi-agent au niveau micro.

4.1.1 Agents

Nous définissons tout d'abord un premier élément qui est l'**agent**, qui sera un élément descriptif existentiel. C'est lui qui décrit la réalité de l'organisation à travers les entités qui la composent.

Dans un premier temps, donc, il apparait que le prérequis à toute existence d'organisation est, de toute évidence, la présence d'agents actifs, interagissant et fondant, par leur existence, celle de l'organisation. Ces entités coexistantes sont la base de toute considération d'organisation d'un système. Elles assurent à la fois le lieu et l'action de ce qui se produit au sein d'une organisation – sa réalisation temporelle.

Notre première définition sera donc celle de l'agent. Il est d'abord nécessaire de réitérer ici un point important de notre approche : nous ne connaissons pas les détails de l'implémentation réelle et de l'architecture interne des agents ; ils sont donc, pour nous, des « boîtes noires » qui sont définies d'un point de vue strictement externe. Ce qui nous permettra de manipuler ces éléments, c'est de donner une définition de la notion de type pour cet élément qui nous permette de les distinguer les uns des autres.

> Ainsi, nous définissons le **type d'un agent** comme une classe d'équivalence fonctionnelle des agents. Pour préciser, nous dirons que deux agents de même type sont identifiables – et donc remplaçables – dans toute simulation – *i.e.* dans toute instanciation de système.

Les agents de même type partagent donc une architecture identique : ils possèdent le même code à l'instant de leur création respective. Ils se comportent ainsi de la même façon

dans les mêmes situations s'ils partagent un même historique de situations et d'états. On notera que deux agents de même type peuvent se comporter différemment dans une même situation pour un instant donné si leur passé est différent – ils peuvent avoir appris, vécus différentes choses, ou encore être dans un état différent au moment choisi – y compris pour leur état initial.

> Pour définir formellement un agent, que nous noterons couramment par les lettres a, b, c..., nous devons donc définir un couple composé d'un identifiant choisi dans $\mathbb{N}$ – pour des raisons de simplicité[7] –, et du type de l'agent, que nous noterons couramment par les lettres α, β, γ... Ainsi :
>
> $$a = (n_a, \alpha), n_a \in \mathbb{N}$$

Nous ne cherchons pas pour l'instant à donner une sémantique particulière à l'ensemble des types d'agents. Ceux-ci prendront en effet signification comme possibles d'une organisation donnée, et nous verrons (section 4.3) comment ils sont utilisés pour décrire formellement une organisation. L'ensemble de tous les types possibles est un ensemble infini, puisqu'il regroupe l'ensemble des classes d'équivalence sur les sorties d'automates correspondant au modèle de machine mis en jeu – qui est, pour nous le modèle de la machine de Turing. Il comporte au moins autant d'éléments que de langages reconnaissables sur l'alphabet {0,1} c'est-à-dire une infinité.

4.1.2 Relations

Le deuxième élément que nous définissons est un élément de définition structurelle : la **relation**. Elle permet de spécifier la structure locale d'un réseau, c'est-à-dire d'un système d'agents communicants.

Afin de décrire une organisation à un niveau micro, il nous est donc nécessaire, après nous être assurés de pouvoir spécifier l'existence des agents eux-mêmes, d'être à même de spécifier la relation des agents à l'altérité, c'est-à-dire les différents liens qui peuvent exister entre les agents et qui forment la structure de l'organisation du système. En effet, les interactions entre agents ne peuvent exister que si les agents se forment une représentation – symbolique ou sub-symbolique – des autres agents du système. C'est à travers cette représentation que la coopération, l'échange – en un mot : l'organisation – peut exister.

> Nous définirons ainsi la notion de relation comme la possibilité d'interaction entre deux agents, ou plus précisément, entre un agent source et un agent cible : l'agent source est à

[7] Il peut également être choisi dans tout espace dénombrable qui remplace $\mathbb{N}$ pour une description particulière ; il peut s'agit, par exemple d'un « nom », en toutes lettres.

même d'engager une ou plusieurs interactions, directes ou indirectes, avec l'agent cible, et, au besoin (pour une interaction directe notamment), « connait » et se représente l'agent cible. Il s'agit donc d'une notion orientée et potentiellement durable dans le temps.

En revanche, de même que nous avons rejeté l'inclusion de la description des agents au sein de la description organisationnelle, nous en excluons également, sachant notre approche, la description des interactions. Ainsi, afin de disposer d'un lien entre cette dimension et la description organisationnelle proposée, nous devons inclure une spécification des interactions possibles au niveau des relations.

Ainsi, nous définissons un **type de relations** comme l'ensemble des interactions auxquelles ces relations peuvent donner lieu. Deux relations du même type impliquent donc les mêmes possibilités d'interaction.

A partir de cela, nous pouvons donc définir formellement une relation, que nous noterons couramment par les lettres r, s, t...

Pour cela, il nous faut donc définir un triplet composé des agents source et cible et du type de la relation, que nous noterons par λ, μ, ν... – deux mêmes agents pouvant en effet être en relation plusieurs fois simultanément pour des types de relations différents. Ainsi :

$$r = (a_s, a_c, \lambda)$$

Les relations définissant les possibilités d'interaction entre deux agents, l'ensemble des relations entretenues par un agent définit donc l'ensemble de ses possibilités d'interactions actuel.

Nous appelons l'ensemble des relations entretenues par un agent son **champ relationnel**; nous notons[8] le champ relationnel de l'agent a à l'instant t : $R_a(t)$. Nous avons donc :

$$R_a(t) \in \{\, r = (a, a_c, \lambda) \,\}$$

Pour des raisons de compréhension, nous préférerons parfois parler du **voisinage** d'un agent dans le système. Cette notion définit le voisinage externe – au sens de la théorie des graphes – d'un agent dans le graphe composé des agents du système (nœuds) et des relations entre agents (arcs).

Le voisinage d'un agent a est donc défini comme :

$$V_a(t) = \{\, a' \mid \exists r \in R_a(t) \wedge\ r = (a, a', \lambda)\}$$

[8] La définition formelle de cet ensemble dans le cadre d'un système donné est précisée section 4.3.2

Cette notion est dérivée de celle de champ relationnel et est moins riche, puisqu'elle ne prend pas en compte dans sa définition le type et le nombre des relations entre l'agent considéré et chacun de ses voisins.

4.1.3 Tâches

Le troisième élément que nous définissons complète les deux premiers éléments en permettant la description de la composante fonctionnelle des organisations : c'est la **tâche**.

Une organisation implique, en plus de l'existence des entités qui la composent et de liens entre ces entités permettant les interactions, que ces entités possèdent, échangent, partagent des objectifs, communs ou individuels, en vue desquels ils agissent ou interagissent. Puisque nous ne supposons rien sur la structure interne des agents, la décomposition formelle de ces objectifs en sous-entités est impossible. En effet, les objectifs des agents peuvent ici autant être des « buts », pour des agents BDI (Rao et Georgeff, 1995) par exemple, que simplement la sémantique associée, pour le programmeur, au code d'un agent réflexe. Ainsi, nous sommes obligés de considérer une représentation atomique des différents objectifs pouvant être suivis par les agents : la tâche.

> Une tâche représente donc l'entité atomique associée à la description fonctionnelle du système. Les tâches représentent ainsi les fonctions des agents à des instants précis.

Afin d'être à même de spécifier la dimension fonctionnelle d'un système, nous avons à nouveau besoin de recourir à la notion de type afin de distinguer les tâches les unes des autres.

> Pour les tâches, la notion de **type de tâches** recouvre la similitude des objectifs : deux tâches de même type recouvrent les mêmes objectifs pour les agents qui les possèdent, bien que ces objectifs soient instanciés par rapport aux agents.

Ainsi, deux agents désirant obtenir une même information, chacun pour leur usage personnel, possèderont une tâche chacun, les deux tâches étant de même type. En revanche, une tâche peut être partagée par plusieurs agents si elle les concerne tous. Ainsi, deux agents désirant en commun mettre à disposition une information à un même tiers, possèderont tous deux une unique tâche.

Afin de spécifier les différences entre des tâches communes et des tâches identiques – de même type mais indépendantes – il est nécessaire d'inclure dans la définition d'une tâche les agents bénéficiaires de cette tâche. Nous noterons donc B_x l'ensemble des agents bénéficiaires d'une tâche x. Cet ensemble peut être vide, réduit à un singleton ou composé de multiples agents ; s'il est vide, la tâche est « désintéressée » : elle peut impliquer une action sur l'environnement, un retour à l'utilisateur, etc. ; s'il est réduit à un singleton, la tâche est

individuelle : elle ne bénéficie qu'à un unique agent, qui peut être celui qui possède la tâche ou non – dans le cas de délégation – ; s'il est multiple, la tâche est commune à plusieurs agents. De plus, une tâche peut être présente plus d'une fois par agent – dans le simple cas de tâches répétitives, par exemple – et il faut donc lui ajouter un identifiant, dans $\mathbb{N}$ ou un autre ensemble dénombrable quelconque, différent pour chaque tâche de même type et de même ensemble de bénéficiaires.

> Ainsi, une tâche, que nous noterons par x, y, z..., est définie par son identifiant j, son ensemble de bénéficiaires B_x et son type, noté ϕ, χ, ψ,... :
>
> $$x = (B_x, j, \phi), j \in \mathbb{N}$$

Les tâches définissant les possibilités d'action d'un agent, l'ensemble des tâches possédé par un agent à un instant donné définit donc l'ensemble de ses possibilités d'action actuel.

> Nous appelons l'ensemble des tâches d'un agent son **champ d'action**; nous notons[9] le champ d'action de l'agent a à l'instant t : $P_a(t)$, avec
>
> $$P_a(t) \in \{\, x = (B, j, \phi), j \in \mathbb{N} \,\}$$

Nous verrons par la suite (section 6.1) qu'il peut être intéressant d'exprimer des relations entre tâches, ou entre types de tâches, afin de dénoter des concepts de sous-objectifs ou d'objectifs partiels, mais dans un premier temps, et puisque ces concepts ne seront utiles que dans l'objectif d'étudier les instanciations précises de systèmes, nous nous contentons de souligner la possibilité de le faire à partir de notre définition de tâche.

4.1.4 Mécanismes organisationnels

Enfin, il nous faut définir un élément atomique de description de la dynamique des systèmes. Nous appellerons ces éléments, en suivant les analyses effectuées dans l'état de l'art, des **mécanismes organisationnels**.

En effet, l'association des concepts d'agents, de relations et de tâches permet de définir un cadre formel de description des systèmes, à travers les notions d'entités actives, de structure de réseau et de répartition fonctionnelle. Il est donc possible de décrire à la fois des éléments généraux – les types – capables de représenter les possibles d'une organisation et d'utiliser leur instanciation pour définir l'état organisationnel d'un système à un instant donné. Ce qu'il nous manque est la possibilité de décrire les réorganisations, modifications et changements – tant dans la structure du système que dans sa répartition fonctionnelle – subis par l'organisation au cours du temps.

---

[9] De même que pour le champ relationnel, la définition formelle de cet ensemble dans le cadre d'un système donné est précisée section 4.3.2

Afin de répondre à ce problème, nous devons introduire la notion supplémentaire de mécanisme organisationnel. Un mécanisme organisationnel décrit la manière dont un agent réorganise localement – la description s'effectue toujours à l'échelle micro – les composantes structurelle et fonctionnelle de son environnement organisationnel.

> Il s'agit donc, formellement, d'un algorithme défini sur un pas de temps du système – nous discutons la définition de ce pas de temps à la section 4.2.3 – associant les états de l'agent et de l'organisation locale à l'état suivant de l'organisation locale, modifiée par l'agent.

Étant donnée notre définition des types d'agents, deux agents de même type se comportent de la même manière dans les mêmes circonstances : ils doivent donc posséder les mêmes mécanismes organisationnels.

> Nous pouvons donc associer chaque type d'agent α avec l'ensemble $SM(\alpha)$ des mécanismes organisationnels que les agents de ce type possèdent – nous l'appelons **réservoir organisationnel spécifique** de α.

Une telle définition permet donc d'obtenir un élément apte à décrire les dynamiques locales mises en place par les agents. Cet élément est atomique dans notre modèle organisationnel, mais complexe dans sa description profonde : il s'agit d'un algorithme complètement défini, et dont l'expression et le calcul posent ainsi des problèmes importants. Il nous faut donc préciser les descriptions des différents mécanismes organisationnels envisageables et ainsi préparer la possibilité de les utiliser d'une manière plus calculatoire par la suite.

4.2 Description des mécanismes

Les mécanismes organisationnels sont donc les éléments de base à l'aide desquels nous pouvons décrire la dynamique d'un système. Il est ainsi possible de décrire, grâce à la combinaison de ces mécanismes, les dynamiques locales imposées par les agents – *i.e.* la dynamique au niveau micro. C'est à travers leur association avec les types d'agents qu'il est possible de décrire, à un niveau plus général, l'ensemble des dynamiques locales potentiellement présentes dans une organisation. Associés à la composition et à la structure initiale d'un système, ils permettent donc de spécifier la dynamique complète d'une instanciation de ce système. Les mécanismes organisationnels sont des algorithmes, applicables par un agent, dont l'objet est l'organisation locale de cet agent. Ils associent donc un état donné de l'agent et de son organisation locale avec un autre état de cette organisation locale. Du point de vue méthodologique, il ne peut rien être imposé de plus sur la définition générale du mécanisme organisationnel.

En revanche, dans un premier temps, en tant que fonction dont l'espace de destination est contenu dans l'espace de départ, il est possible de définir un opérateur de **composition** sur les mécanismes organisationnels pour simplifier l'expression de mécanismes complexes. Ainsi, nous définissons $m^2 \circ m^1$ comme le mécanisme dont le résultat à t est le résultat de m^2 appliqué à l'agent dans l'état où il se trouve à t et dont l'organisation locale est dans l'état où elle se trouverait à $t + \Delta t$ après l'application de m^1 à t. C'est donc le résultat de l'application successive de m^1 puis de m^2 à l'instant t sans changement de l'état interne de l'agent.

Dans un second temps, en suivant notre analyse de l'état de l'art (section 2.3), nous avons classifié les mécanismes observés dans les exemples classiques de SMA. Il est possible de ce servir de cette classification pour particulariser les mécanismes et ainsi déterminer des expressions plus spécifiques à ces mécanismes. L'objectif étant de préciser l'expression de ces mécanismes à des fins ultérieures d'application au calcul de propriétés organisationnelles, tout en conservant une bonne généricité de la définition, nous nous limiterons au premier critère posé sur les mécanismes : leur portée. Nous distinguerons donc les mécanismes dont l'objet est de définir l'évolution de l'aspect structural de l'organisation locale de l'agent, c'est-à-dire les relations qu'il entretient avec d'autres agents, des mécanismes définissant l'évolution de l'aspect fonctionnel de l'organisation locale, c'est-à-dire le partage de ses tâches avec d'autres agents.

Nous décrivons dans cette section la formalisation de ces deux types de mécanismes, puis nous traitons du problème de la définition du pas de temps Δt qui sert à leur définition. Nous définissons également les notions de mécanismes partiels et de mécanismes totaux, ces derniers étant les résultats de la sommation des mécanismes partiels – la sommation des mécanismes étant l'addition des modifications effectuées en parallèle sur l'état courant de l'organisation.

4.2.1 Mécanismes structuraux

> Nous définissons les mécanismes structuraux comme les mécanismes organisationnels qui modifient le champ relationnel d'un agent, c'est-à-dire l'ensemble des relations qu'il entretient avec d'autres agents.

L'aspect local de la dimension structurelle d'une organisation se retrouve, sachant les définitions que notre approche nous a amené à proposer, dans l'ensemble des relations que possède un agent avec d'autres agents du système. Un mécanisme organisationnel structural est donc un algorithme permettant, pour un agent donné, de passer de l'état du champ relationnel de l'agent à t au champ relationnel de cet agent à $t + \Delta t$ (cf. Figure 4.1) en prenant en compte l'état de l'agent à t, que nous noterons $\xi_a(t)$ – en effet, cet état et en particulier sa mémoire, peut modifier la manière dont l'agent souhaite modifier son organisation locale – et

le champ d'action de l'agent à t - car celui-ci définit les objectifs courants de l'agent et engage donc potentiellement l'agent à des choix précis.

> Un mécanisme structural est donc défini comme une fonction :
>
> $$s_a : \left(R_a(t), P_a(t), \xi_a(t)\right) \longmapsto R_a(t + \Delta t)$$

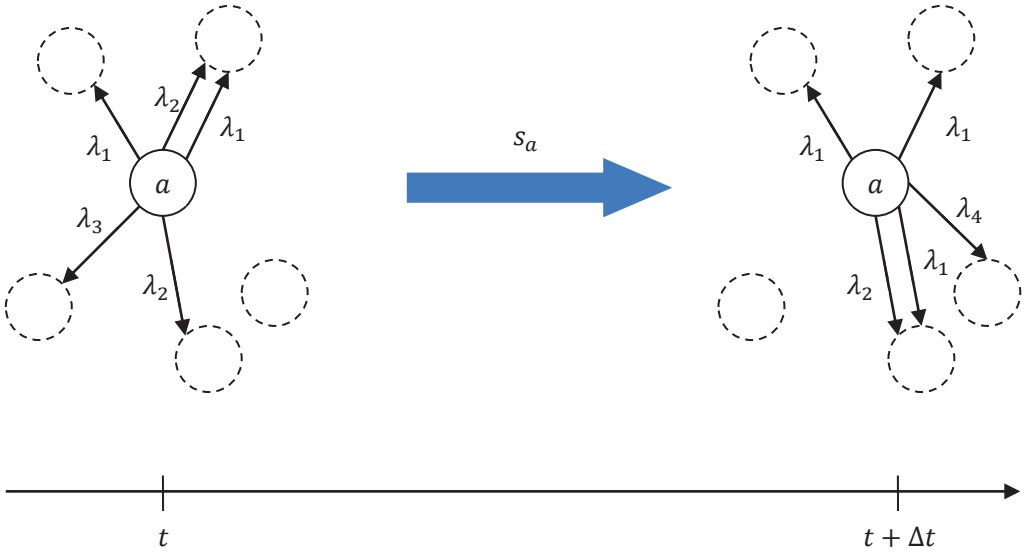


Figure 4.1 - Action des mécanismes structuraux

La prise en compte dans un mécanisme organisationnel de l'état actuel de l'agent et de ses tâches courantes – son champ d'action – peut varier. Dans le cas le plus général, les deux sont pris en compte, ce qui donne des mécanismes complexes et variables. Cependant nous distinguons les cas où :

- l'état de l'agent n'est pas pris en compte. Nous parlons de mécanismes structuraux **homogènes** (en opposition aux mécanismes **hétérogènes** où l'état de l'agent est pris en compte) :

$$s_a : \left(R_a(t), P_a(t)\right) \longmapsto R_a(t + \Delta t)$$

- le champ d'action de l'agent n'est pas pris en compte. Nous parlons de mécanismes structuraux **réactifs** (en opposition aux mécanismes **téléonomiques** où le champ d'action est pris en compte) :

$$s_a : \left(R_a(t), \xi_a(t)\right) \longmapsto R_a(t + \Delta t)$$

- dans le cas particulier où ni le champ d'action, ni l'état de l'agent ne sont pris en compte (mécanismes homogènes et réactifs), nous parlons de mécanismes **libres** (en opposition aux mécanismes **liés**, qui sont au moins hétérogène ou téléonomiques) :

$$s_a : R_a(t) \longmapsto R_a(t + \Delta t)$$

Pour ajouter à cette classification, mais de manière spécifique aux mécanismes structuraux[10], nous définissons également un cas particulier de mécanismes structuraux : les mécanismes décrivant le champ relationnel initial d'un agent introduit dans le système à un instant $t \neq 0$.

Cette classification nous permet par la suite de mesurer la complexité et la spécificité d'un mécanisme organisationnel et ainsi de définir dans quelle mesure il est possible d'extraire des informations globales de ce mécanisme et quels points il est nécessaire de spécifier pour obtenir ces informations (cf. section 5.4.2).

Nous nommons pour cela **mécanisme d'insertion** tout mécanisme structural s_a n'effectuant des modifications que sur le champ relationnel vide, c'est-à-dire tel que :

$$\forall R_a(t) \neq \emptyset, s_a\big(R_a(t), P_a(t), \xi_a(t)\big) = R_a(t)$$

Pour de tels mécanismes, les seules valeurs significatives, c'est-à-dire$s_a\big(\emptyset, P_a(t), \xi_a(t)\big)$, représentent les valeurs initiales possibles du champ relationnel de l'agent a introduit dans le système, selon les conditions de son insertion (tâches initiales et état initial).

Nous notons $\mathcal{SS}(\alpha)$ le **réservoir structurel spécifique** de α, c'est-à-dire le sous-ensemble de tous les mécanismes structuraux partagés par les agents de type α. Il s'agit d'un sous-ensemble du réservoir organisationnel spécifique (cf. section 4.1.4) de α, $\mathcal{SM}(\alpha)$:

$$\mathcal{SS}(\alpha) \subseteq \mathcal{SM}(\alpha)$$

Par ailleurs, sachant qu'un agent peut cumuler plusieurs mécanismes structuraux différents et en parallèle, nous notons S_a le mécanisme structural total, défini comme la résultante des mécanismes s_a^i ($i \in [\![1, |\mathcal{SS}(\alpha)|]\!]$) utilisés par l'agent a de type α. On a donc, la somme des mécanismes organisationnels étant définie comme l'union des résultats de ces mécanismes effectués en parallèle :

$$S_a = \sum_{i=1}^{|\mathcal{SS}(\alpha)|} s_a^i$$

4.2.2 Mécanismes fonctionnels

Nous définissons ensuite les mécanismes fonctionnels comme étant les mécanismes organisationnels qui modifient la manière dont un agent partage des tâches avec les agents de son voisinage.

[10] Le reste de la classification étant, comme nous le voyons à la section suivante, applicable à l'ensemble des mécanismes organisationnels.

La dimension fonctionnelle d'un système présuppose sa dimension structurelle. En effet, ce qui est décrit dans la dimension fonctionnelle d'un système est l'ensemble des requêtes, partages, délégations et attributions de tâches entre un agent et les agents de son voisinage. Et il est nécessaire à un agent de posséder un voisinage pour pouvoir développer des échanges fonctionnels avec d'autres agents. Cela suppose donc que les mécanismes fonctionnels, décrivant l'évolution fonctionnelle de l'organisation locale pour un agent, soient calculés à un instant donné après les mécanismes structuraux. L'application des mécanismes fonctionnels entre t et $t + \Delta t$ suppose donc la connaissance préalable de $R_a(t + \Delta t)$, champ relationnel de l'agent à $t + \Delta t$, c'est-à-dire résultat de l'application des mécanismes structuraux de l'agent à t.

Ce que vont modifier les mécanismes fonctionnels, c'est l'échange fonctionnel de l'agent avec ses voisins (cf. Figure 4.2). Nous représentons cet échange par l'attribution de tâches par l'agent à ses voisins. Cette attribution peut avoir plusieurs significations, par exemple :

- *L'attribution par a d'une tâche qu'il possède $x \in P_a(t)$ et dont il est bénéficiaire ($a \in B_x$) à un agent de son voisinage peut être une délégation.*
- *L'attribution par a d'une tâche qui est nécessaire ou bénéfique à la résolution d'une de ses tâches propres peut-être considérée comme une requête ou une demande de service.*
- *L'attribution par a d'une tâche qui est bénéfique au système entier peut être un ordre, une recommandation ou le résultat d'une répartition de tâches dans le système.*

Malgré les diverses interprétations possibles, le processus mis en jeu reste similaire : l'attribution par l'agent a d'une tâche x à un agent de son voisinage dans le but d'interaction « sociale », c'est-à-dire de partage de fonction.

Nous décrivons ce processus par un ensemble $D_a(t)$ que nous nommons **extension sociale** de a et qui regroupe l'ensemble des couples formés par les tâches attribuées et les agents du voisinage auxquels elles ont été attribuées :

$$D_a(t) \subseteq \{(x, a_c) \mid a_c \in V_a(t)\}$$

Les mécanismes fonctionnels sont descriptibles à partir de cet ensemble.

Nous les définissons donc comme des mécanismes qui décrivent les modifications effectuées sur l'extension sociale d'un agent en prenant en compte l'évolution de son

champ relationnel[11], son état actuel $\xi_a(t)$ et son champ d'action $P_a(t)$. Nous les représentons donc comme une fonction :

$$f_a : \big(D_a(t), R_a(t+\Delta t), P_a(t), \xi_a(t)\big) \longmapsto D_a(t+\Delta t)$$

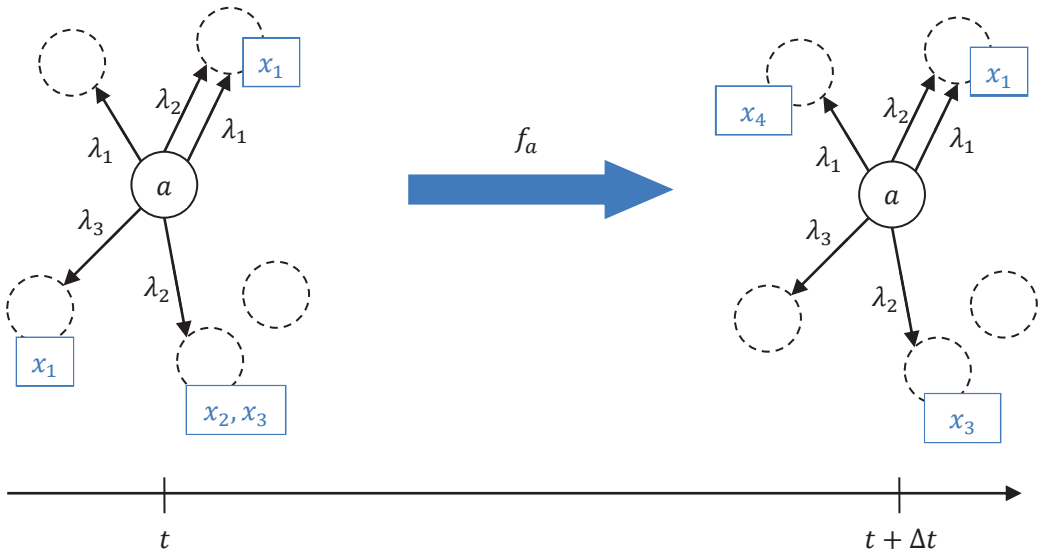


Figure 4.2 - Actions des mécanismes fonctionnels

Nous faisons les mêmes distinctions que pour les mécanismes structuraux entre mécanismes homogènes et hétérogènes et entre mécanismes réactifs et téléonomiques selon la prise en compte respective de l'état de l'agent et de son champ d'action à l'instant t. De plus, nous faisons une distinction supplémentaire dans le cas où :

– l'état actuel de l'extension sociale de a n'est pas pris en compte pour calculer le suivant ; nous parlerons alors de mécanismes **négligents** (en opposition aux mécanismes **projectifs** qui le prennent en compte) :

$$f_a : \big(R_a(t+\Delta t), P_a(t), \xi_a(t)\big) \longmapsto D_a(t+\Delta t)$$

Nous notons $\mathcal{SF}(\alpha)$ le **réservoir fonctionnel spécifique** de α, c'est-à-dire le sous-ensemble de tous les mécanismes fonctionnels partagés par les agents de type α. Il s'agit d'un sous-ensemble du réservoir organisationnel spécifique (cf. section 4.1.4) de α, $\mathcal{SM}(\alpha)$:

$$\mathcal{SF}(\alpha) \subseteq \mathcal{SM}(\alpha)$$

[11] Le champ relationnel étant une notion qui véhicule plus d'informations que le voisinage (à cause du typage et de la possible multiplicité des relations entre deux agents), c'est lui et non le voisinage qui est à prendre en compte comme donnée des mécanismes fonctionnels.

Par ailleurs, sachant qu'un agent peut cumuler plusieurs mécanismes fonctionnels différents et en parallèle, nous notons F_a le mécanisme fonctionnel total, défini comme la résultante des mécanismes f_a^i ($i \in [\![1, |\mathcal{S}\mathrm{F}(\alpha)|]\!]$) utilisés par l'agent a de type α. On a donc, la somme des mécanismes organisationnels étant définie comme l'union des résultats de ces mécanismes effectués en parallèle :

$$F_a = \sum_{\mathrm{i}=1}^{|\mathcal{S}F(\alpha)|} f_a^i$$

On note que l'opération d'agrégation entre mécanismes structuraux et mécanismes se doit, par cohérence, de correspondre à la composition de mécanismes (comme définie au début de la section 4.2). En effet, les effets des mécanismes fonctionnels sont calculés à partir du résultat des mécanismes structuraux à chaque pas de temps. Au vu de cela, on note qu'il est possible de séparer tout mécanisme organisationnel en une partie structurale et une partie fonctionnelle, puisque l'application de l'une se doit d'être consécutive à l'autre[12]. Et puisque les deux composantes organisationnelles modifiables par l'agent localement sont la structure et la fonction, c'est-à-dire son champ relationnel et son extension sociale, nous posons comme principe de construction que **tout mécanisme organisationnel partiel doit être soit structural soit fonctionnel, jamais les deux**.

A partir de cela, nous avons donc, pour un type d'agents α :

$$\mathcal{S}M(\alpha) = \mathcal{S}S(\alpha) \cup \mathcal{S}F(\alpha)$$

Et il est donc possible de définir le mécanisme organisationnel total pour un agent a comme la composition de son mécanisme fonctionnel total avec son mécanisme structural total :

$$M_a = F_a \circ S_a$$

Ce qui revient, *in fine*, à le définir comme :

$$M_a = \left(\sum_{j=1}^{|\mathcal{S}F(\alpha)|} f_a^j \right) \circ \left(\sum_{i=1}^{|\mathcal{S}S(\alpha)|} s_a^i \right)$$

[12] La preuve en est que l'extension sociale de a en $t + \Delta t$ ne peut être définie qu'en connaissance du champ relationnel de a en $t + \Delta t$, puisque les tâches ne peuvent être déléguées qu'à des agents présents dans le voisinage au moment de la délégation.

4.2.3 Définition du pas de temps

Toutes les définitions de ces algorithmes se fondent sur un élément commun qu'est le pas de temps Δt. Nous nous assurons ici de l'existence d'un tel élément et discutons sa définition dans les divers types de systèmes qu'il est possible d'étudier.

Dans un premier temps, nous affirmons qu'il est toujours possible de définir, au niveau de l'agent, un pas de temps cohérent qui permette l'expression de ces algorithmes. Le premier choix se porte évidemment sur le laps de temps au bout duquel les mécanismes organisationnels sont réévalués par l'agent, mais ce laps de temps n'est pas nécessairement constant. En revanche, le top d'horloge lié au **temps processeur** du processus d'exécution de l'agent est une mesure cohérente du temps pour l'exécution d'un agent, et donc permet de définir convenablement un pas de temps pour ces algorithmes. Il est à noter que, dès lors, le résultat des mécanismes organisationnels a de grandes chances d'être identique à l'état de l'organisation avant leur application, car ils ne seront pas évalués par l'agent à chaque pas de temps[13].

Cependant, dans un système multi-agent, il va nous intéresser de définir formellement les mécanismes indépendamment de l'agent qui les utilise. Pour cela, il nous faut définir un pas de temps Δt indépendant de l'agent choisi et donc commun pour le système entier. Cela est possible de deux manières différentes :

– **Si le SMA est un système synchrone**

Alors il existe un pas de temps commun à tous les agents qui règle la communication entre agents et avec l'extérieur – environnement, utilisateur. Connaître l'état des agents aux instants de communication suffit à connaître le système – puisqu'ils se comportent, dans l'intervalle, comme des processus indépendants. Il est donc possible, dans ce cas, de choisir pour Δt le **pas de temps synchrone** du système. Ainsi, les états des champs relationnels et des extensions sociales des agents sont connus à tous les pas de temps, et donc à tous les moments possibles d'interaction entre agents.

– **Si le SMA est un système asynchrone**

Alors, il a été prouvé dans (Mattern, 1989) qu'il existe un pas de temps définissable pour le système entier dans lequel sont incluses toutes les étapes d'interactions entre les agents. Ainsi, si nous définissons Δt comme ce **pas de temps réparti**, nous retrouvons les propriétés énoncées pour les systèmes synchrones et nous disposons d'un pas de temps

[13] Ce qui ne pose pas de problème, car l'évaluation sera déclenchée ou non selon l'état actuel de l'agent, qui relève déjà des entrées des mécanismes.

commun à tous les agents qui permet d'exprimer de manière unifiée les mécanismes organisationnels. Il est à noter que ce pas de temps, contrairement à celui des systèmes synchrones, n'est pas directement lié au temps processeur – ni par ailleurs, au temps réel – ce qui pourra avoir de l'importance dans le cadre de l'interprétation des calculs effectués sur les organisations (cf. chapitre 5).

Il est donc, dans tous les cas, possible de définir un pas de temps Δt commun pour tout le système, et donc de définir les mécanismes organisationnels de façon indépendante et unifiée pour l'organisation en entier.

4.2.4 Identification de l'émergence

Il est intéressant, à ce niveau de la construction, d'identifier l'expression de l'émergence à travers les différents éléments du modèle proposé. Pour cela, nous commençons par définir la fonction globale du système comme la somme des fonctions de ses éléments[14].

$$F_{sys}(t) = \sum_{a \in A(t)} F_a(t) + \sum_{r \in R(t)} F_r(t) + \sum_{m \in \mathcal{G}M} F_m(t)$$

Sachant que les relations sont, pour $t \neq 0$, dépendantes uniquement des mécanismes structuraux et de l'instant initial, nous pouvons réécrire :

$$F_{sys}(t) = \sum_{a \in A(t)} F_a(t) + \sum_{r \in R(0)} F_r(t) + \sum_{m \in \mathcal{G}M} F_m(t)$$

En comparant à l'équation fonctionnelle (cf. section 3.1.1), nous pouvons écrire :

$$Em(t) = \sum_{r \in R(0)} F_r(t) + \sum_{m \in \mathcal{G}M} F_m(t)$$

Soit, en intégrant par rapport à t :

$$Em = \sum_{r \in R(0)} F_r + \sum_{m \in \mathcal{G}M} F_m$$

Nous voyons ainsi, que, dans notre modèle et en suivant les définitions posées dans notre approche, l'émergence d'un SMA peut se définir comme la somme des fonctions des mécanismes organisationnels, auxquelles il faut ajouter la fonction jouée par le réseau de relations initiales. Cela montre que la *fonction collective* (Demazeau, 1995) est décrite, dans notre modèle, par l'ensemble formé des relations initiales et des ensembles de mécanismes

[14] Nous ne donnons pas de fonction aux tâches, puisqu'elles sont justement la représentation de la fonction des agents. On peut en revanche noter que :

$$\forall a \in A(t), F_a(t) = \sum_{x \in P_a(t)} x$$

organisationnels pour chaque type d'agent. C'est associés à la connaissance des agents du système et des tâches de ces agents qu'il est ainsi possible de décrire la fonction totale du système.

4.3 Description des organisations

Après avoir explicité les éléments (cf. Figure 4.3) que nous utilisons pour décrire, à partir du niveau micro, les différents aspects des organisations, nous devons assembler ces éléments dans une structure permettant la description complète des organisations et des systèmes qu'elles sous-tendent.

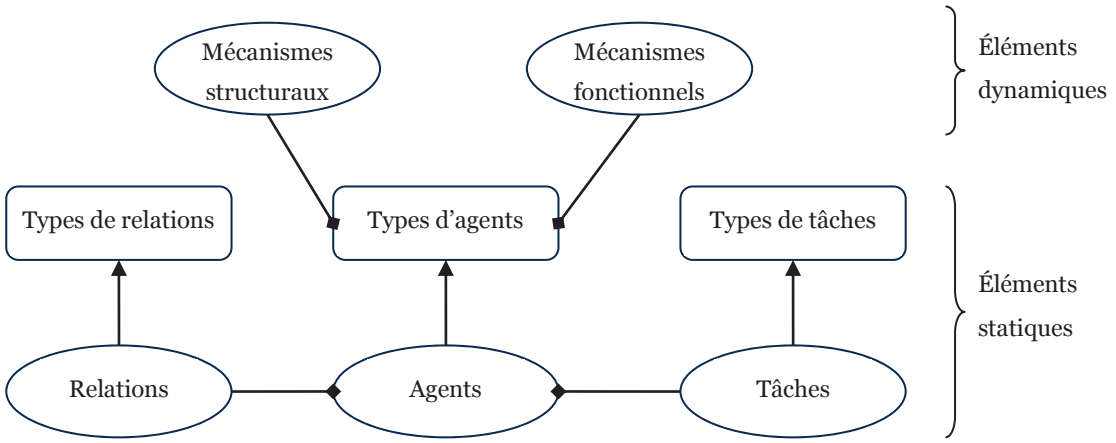


Figure 4.3 - Éléments constitutifs du modèle

Notre objectif étant de permettre la description de systèmes variés, allant de systèmes fermés et totalement spécifiés à des systèmes ouverts dans lesquels le nombre et le type des agents peuvent être totalement inconnus des concepteurs, nous devons établir une structure permettant de définir des systèmes à travers un ensemble d'éléments organisationnels, mais également à partir de contraintes plus ou moins fortes sur l'organisation et les systèmes résultants.

Par ailleurs, il apparait nécessaire de distinguer la spécification d'une organisation abstraite et celle d'un système instancié. En effet, en suivant la distinction effectuée dans le modèle PopOrg (Demazeau et Costa, 1996), nous avons choisi de séparer la description d'une organisation et celle de la population qui l'instancie. Ainsi, nous voulons spécifier, au niveau organisationnel, à la fois la description de l'organisation elle-même, avec les inconnues que cela représente, mais aussi la description d'une instanciation de cette organisation dans un

système, en ajoutant les spécifications requises[15]. En revanche, ce sont les éléments que nous avons définis (section précédente), et eux seulement, qui nous serviront à décrire à la fois les organisations et les instances de systèmes.

Il s'agit donc d'utiliser ces éléments dans l'optique de définir l'ensemble des possibles pour une organisation donnée, tout en spécifiant du mieux possible les contraintes et les degrés de liberté. Et, dans un second temps, il s'agit de spécifier ce qui a été laissé comme degrés de liberté pour définir une instance précise d'un système.

4.3.1 Le principe des réservoirs

La méthode que nous suivons pour rendre compte de ces divers degrés de spécification et de contrainte est de grouper les éléments organisationnels définis en « réservoirs[16] », *i.e.* en ensembles, qui pourront être définis plus ou moins précisément en fonction du niveau de spécification.

> Nous appelons **réservoir** tout ensemble définissable logiquement à partir de la variable temporelle et des éléments de base de l'organisation : agents et types d'agents, relations et types de relations, tâches et types de tâches, mécanismes organisationnels – structuraux et fonctionnels[17].

Ainsi, il est possible de définir, pour chaque organisation – que ce soit au niveau abstrait ou au niveau instance, que nous traitons tous deux dans cette sous-partie – des ensembles d'éléments représentant des informations de spécification. Pour chaque réservoir ainsi exprimé, deux informations apparaissent comme informatives :

- la cardinalité du réservoir (*i.e.* son nombre d'éléments), qui, spécifiée complètement ou seulement contrainte, apporte de l'information quantitative sur l'organisation ;

---

[15] Il est à noter que, comme présenté dans notre approche, nous ne décrivons, même pour une instance de système, que la dimension organisationnelle : nous n'incluons pas, même à ce stade, les autres composantes de la description des SMA.

[16] D'où les appellations de « réservoirs organisationnels » spécifiques et globaux choisies précédemment.

[17] Techniquement parlant, les définitions des réservoirs peuvent contenir les quantificateurs universel et existentiel, les opérations logiques (négation, et, ou), les relations d'inclusion (et d'appartenance), les opérations sur les ensembles (union, complémentaire, intersection, produit cartésien) et les relations d'ordre sur les ensembles ordonnés (notamment le temps).

– le contenu du réservoir (*i.e.* ses éléments), qui, spécifié complètement ou partiellement, apporte de l'information qualitative sur l'organisation.

Nous pouvons spécifier ces propriétés sur chacun des réservoirs que nous pouvons définir. Ces ensembles exprimant les possibilités d'existence de chacun des éléments organisationnels atomiques que nous avons précédemment introduits, nous pouvons donc en définir pour :

- chacun des éléments de base de notre modèle ;
- chacun des types correspondant à ces éléments.

Les réservoirs de types permettent ainsi un degré d'abstraction supérieur aux réservoirs d'éléments lors de la spécification.

Agents

Concernant les agents, nous pouvons tout d'abord définir l'ensemble des types d'agents présents dans l'organisation. Nous le notons T_A. Il permet de définir l'ensemble des types d'agents rencontrés dans le système.

Dans un système fermé – où l'ensemble des agents est connu dès le départ – l'ensemble des types d'agents est un ensemble fini et entièrement spécifié pour l'organisation : $| T_A | < +\infty$

Dans un système ouvert – où l'ensemble des agents n'est pas connu dès le départ, la cardinalité de T_A dépend de la spécification de l'organisation : l'ensemble des types d'agents peut être limité par la conception du système ou bien laissé libre jusqu'à l'exécution. En revanche, lors d'une exécution en temps limité, T_A sera un ensemble fini pour l'instance choisie.

Nous définissons ensuite les ensembles d'agents présents dans l'organisation. L'ensemble des agents présents dans le système à au moins un instant donné[18] est noté A. L'ensemble des agents présent dans le système à un instant t donné est noté $A(t)$. On a bien sûr :

$$\forall t\ A(t) \subseteq \mathrm{A}$$

Pour les instances, ces deux ensembles sont toujours des ensembles finis, car les instances de systèmes sont considérées bornées dans le temps. Pour les organisations, seul l'ensemble des agents à un instant donné est un ensemble fini – s'il est connu – tandis que la cardinalité

[18] On parlera simplement de « l'ensemble des agents », en opposition à « l'ensemble des agents à l'instant t » défini juste après.

de l'ensemble des agents A peut, elle, être inconnue. Nous pouvons, à partir de cela, redéfinir la notion d'ouverture des systèmes[19]. Ainsi :

- Si l'ensemble A des agents du système est inconnu, il s'agit d'un **système ouvert** : les agents peuvent entrer et sortir du système, et ils ne sont pas connus à l'avance.
- Si l'ensemble A est connu ou est inclus dans un ensemble connu, mais que l'ensemble $A(t)$ n'est pas connu quel que soit t, on peut définir le système comme étant un **système semi-ouvert** : les agents peuvent entrer et sortir du système, mais l'ensemble total des agents est connu d'avance.
- Si l'ensemble $A(t)$ est connu pour tout t, il s'agit d'un **système fermé** : l'appartenance des agents au système est préprogrammée ou bien constante.

Pour des raisons pratiques, il sera parfois favorable de considérer d'abord l'agent lui-même avant les ensembles auxquels il appartient.

Dans cette perspective, il est notable que l'on peut définir, et de manière équivalente – il est possible à tout moment de passer d'une définition à l'autre –, pour chaque agent $a \in \mathrm{A}$ la notion de **période d'existence** de l'agent comme :

$$L(a) = \{\, t \in \mathbb{R}^+ \mid a \in A(t) \}$$

Nous définissons également deux ensembles supplémentaires pour être à même de préciser la description d'une organisation dans certains cas. Pour des raisons pratiques, nous notons $\mathrm{t}(e)$ le type de l'élément e (e étant un agent, une relation ou une tâche).

Le premier est l'ensemble des agents d'un type $\alpha \in T_A$ donné :

$$\mathrm{A}_\alpha = \{\, a \in \mathrm{A} \mid \mathrm{t}(a) = \alpha \} \subseteq \mathrm{A}$$

Le second est l'ensemble des agents d'un type $\alpha \in T_A$ donné présents dans le système à l'instant t :

$$A_\alpha(t) = \{\, a \in A(t) \mid \mathrm{t}(a) = \alpha \} \subseteq A(t)$$

Ces ensembles partagent les propriétés et les sémantiques des ensembles dans lesquels ils sont inclus.

[19] On notera qu'alors, la connaissance de T_A permet de définir différents degrés d'ouverture selon sa cardinalité : un système sera d'autant plus ouvert que la cardinalité de T_A est grande si A n'est pas connu. En effet, plus il y a de types possibles d'agents, plus les entrées et sorties d'agents peuvent avoir des effets variés sur le système, et donc moins le comportement du système sera prédictible à partir d'une organisation non instanciée.

Relations

De la même manière que pour les agents, nous définissons pour les relations des ensembles permettant la description des organisations.

> Le premier ensemble que nous définissons est donc l'ensemble des types de relations présents dans l'organisation. Nous le notons T_R.

De même, son cardinal peut être fini ou infini, selon que l'ensemble des relations possibles entre agents est connu au départ ou bien que les agents peuvent développer entre eux des relations non prévues par les concepteurs du système. Cependant, puisque les relations sont créées et modifiées par les mécanismes organisationnels, et que ces mécanismes sont attribués de manière fixe aux types d'agents pour une organisation donnée, la cardinalité de l'ensemble des types de relations ne peut être infinie que si celle de l'ensemble des types d'agents l'est également ; soit :

$$| T_A | < +\infty \Rightarrow | T_R | < +\infty$$

En revanche, la cardinalité de T_R n'impose pas de contrainte sur celle de T_A. De la même manière que pour les agents, dans le cadre de la description d'une instance précise en temps fini, la cardinalité de T_R sera toujours finie[20].

> Nous définissons ensuite l'ensemble des relations présentes dans l'organisation à un instant donné. Nous le notons $R(t)$. Par rapport à nos définitions précédentes, cet ensemble est obtenu par l'union des champs relationnels (cf. section 4.1.2) de tous les agents présents dans le système à l'instant t :
>
> $$R(t) = \cup_{a \in A(t)} R_a(t)$$

Pour les instances comme pour les organisations, $R(t)$ est toujours un ensemble fini, bien qu'il ne soit pas nécessairement pleinement défini pour une organisation non instanciée.

> Nous pouvons, à partir de cet ensemble, définir le **graphe structurel** $\mathbb{G}(t)$ de l'organisation comme le graphe orienté dynamique formé des agents pour les nœuds et des relations pour les arcs :
>
> $$\mathbb{G}(t) = < A(t), R(t) >$$

Nous définissons également d'autres ensembles plus spécifiques à partir de celui-ci, ainsi que les graphes associés, en spécifiant par rapport au type de la relation. Nous notons par exemple $R_\lambda(t)$ l'ensemble des relations de types λ entre les agents du système, et le graphe associé $\mathbb{G}_\lambda(t) = < A(t), R_\lambda(t) >$.

---

[20] Cela formalise le fait qu'un nombre fini d'agents ne peut, en un temps fini mettre en jeu qu'un nombre fini de relations.

Tâches

Ensuite, nous définissons les mêmes catégories d'ensembles concernant les éléments tâches. Le premier ensemble défini est donc l'ensemble des types de tâches possibles dans le système. Nous le notons T_P[21].

Sa cardinalité exprime cette fois la diversité des objectifs possibles pour les agents du système. Ainsi, dans un système purement coopératif, où tous les agents participent au même but global et ne disposent d'aucun but personnel, tous les objectifs possibles sont reliés à la fonctionnalité même du système. Il sera donc possible de définir T_P comme un ensemble fini. En revanche, dans un système où les agents peuvent suivre leurs propres objectifs, il sera peu courant de pouvoir spécifier totalement T_P pour une organisation non instanciée, et, dans le cas d'un système ouvert, son cardinal sera infini – bien qu'une fois encore, son cardinal soit fini pour toute instance.

Nous définissons ensuite l'ensemble des tâches présentes dans l'organisation à un instant donné. Nous le notons $P(t)$ et l'obtenons par union de l'ensemble des champs d'actions $P_a(t)$ (cf. section 4.1.3) des agents du système à cet instant :

$$P(t) = \bigcup_{a \in A(t)} P_a(t)$$

L'ensemble $P(t)$ est de cardinal fini pour les instances, mais en revanche, son cardinal peut être fini ou infini, connu ou inconnu pour les organisations non instanciées, car des tâches de même type pouvant être multipliées à l'infini par un agent donné, aucune limite ne s'impose quand au cardinal de $P_a(t)$ pour chacun des agents d'une organisation – non instanciée.

Cet ensemble peut également être spécifié selon le type des tâches, pour définir des sous-ensembles $P_\phi(t) \subseteq P(t),\ \phi \in T_P$.

Il est ensuite intéressant de définir des ensembles de tâches plus généraux pour le système.

On définit ainsi d'abord les **réservoirs d'actions spécifiques**, $\mathcal{S}P(\alpha)$ définis de même comme les ensembles de toutes les tâches possibles pour un type d'agents donné[22] :

$$\mathcal{S}P(\alpha) = \{\, x \mid \exists t \in \mathbb{R}^+, \exists a \in A(t), \mathfrak{t}(a) = \alpha, x \in P_a(t) \,\}$$

[21] De l'anglais *purpose*.

[22] Eux-mêmes ne sont qu'exceptionnellement totalement connus, mais peuvent être utilisés pour spécifier certaines propriétés fonctionnelles de l'organisation.

Il est ensuite possible de définir l'ensemble de toutes les tâches possibles pour le système. On le nomme le **réservoir d'actions global** et on le note $\mathcal{G}P$:

$$\mathcal{G}P = \bigcup_{\alpha \in T_A} \mathcal{S}P(\alpha) = \{\, x \mid \exists t \in \mathbb{R}^+, \exists a \in A(t), x \in P_a(t) \,\}$$

Cet ensemble n'est connu que dans le cadre d'un système fermé et totalement spécifié, mais même dans ce cas, il n'est pas nécessairement de cardinal fini.

Par ailleurs, il est intéressant de définir l'ensemble des tâches imposées par des agents extérieurs à un agent a donné, c'est-à-dire l'ensemble des tâches affectées à l'agent a par d'autres agents $a' \in A(t)$ dans leurs extensions sociales $D_{a'}(t)$ (cf. section 4.2.2) respectives. Nous le nommons **champ extérieur** de a :

$$E_a(t) = \{\, x \mid (x, a) \in D_{a'}(t) \wedge a' \in A(t) \}$$

Cet ensemble décrit les attentes que les autres agents ont vis-à-vis de l'agent a. Il permet de décrire les liens fonctionnels unissant les agents du réseau.

Il est ainsi possible, à partir de cet ensemble, des extensions sociales $D_a(t)$, et des champs d'actions $P_a(t)$ des agents, de déterminer la coopération entre l'agent a et les autres agents du système. Nous pouvons définir deux ensembles, chacun soit au niveau de l'agent soit au niveau du système, qui font état de cela.

Les premiers d'entre eux sont le **champ de coopération** de a avec a' et le champ de coopération de a, respectivement :

$$C_a^{a'}(t) = \{\, x \mid (x, a) \in D_{a'}(t) \wedge x \in P_a(t) \}$$

$$C_a(t) = \{\, x \mid x \in E_a(t) \wedge x \in P_a(t) \,\} = \bigcup_{a' \in A(t)} C_a^{a'}(t)$$

Les suivants sont le **champ de non-coopération** de a avec a' et le champ de non-coopération de a, respectivement :

$$NC_a^{a'}(t) = \{\, x \mid (x, a) \in D_{a'}(t) \wedge x \notin P_a(t) \}$$

$$NC_a(t) = \{\, x \mid x \in E_a(t) \wedge x \notin P_a(t) \,\} = \bigcup_{a' \in A(t)} NC_a^{a'}(t)$$

Ces ensembles décrivent respectivement les situations de coopérations de l'agent avec ce que les autres attendent de lui et les situations de non-coopérations. Les situations de non-coopération correspondent exactement aux cas où un agent a du système est tel que $NC_a(t) \neq \emptyset$. Ils permettent ainsi de décrire ou d'identifier des situations souhaitées ou non souhaitées. Leurs cardinaux sont limités par les cardinaux des champs d'actions et des extensions sociales des agents, et uniquement par cela ; ces cardinaux peuvent ainsi être infinis dans le cas d'organisations non instanciées et sont nécessairement finis pour des instances.

On a, bien sûr, les relations suivantes :

$$\forall t, \forall a, a' \in A(t), C_a^{a'}(t) \cup NC_a^{a'}(t) = \{ x \mid (x, a) \in D_{a'}(t) \}$$
$$\forall t, \forall a \in A(t), C_a(t) \cup NC_a(t) = E_a(t)$$

Sachant que, au vu des définitions :

$$\forall t, \forall a, a' \in A(t), C_a^{a'}(t) \cap NC_a^{a'}(t) = \emptyset$$
$$\forall t, \forall a \in A(t), C_a(t) \cap NC_a(t) = \emptyset$$

Mécanismes organisationnels

Enfin, en ce qui concerne les mécanismes organisationnels, nous nous trouvons également face à plusieurs ensembles qui doivent être définis pour donner la possibilité de spécifier, cette fois d'un point de vue dynamique, les caractéristiques des organisations et de leurs instances.

> Les premiers ensembles que nous définissons sont les ensembles de tous les mécanismes organisationnels disponibles dans le système. Ils sont au nombre de deux : le **réservoir structurel global** $\mathcal{GS}$ et le **réservoir fonctionnel global** $\mathcal{GF}$. Ils sont définis par l'union (sans redondance) des différents réservoirs structurels (resp. fonctionnels) spécifiques des différents types d'agents présents dans l'organisation :
>
> $$\mathcal{GS} = \bigcup_{\alpha \in T_A} \mathcal{SS}(\alpha)$$
> $$\mathcal{GF} = \bigcup_{\alpha \in T_A} \mathcal{SF}(\alpha)$$

Puisque nous avons partitionné l'ensemble des mécanismes organisationnels en mécanismes structuraux et fonctionnels, la réunion de ces deux ensembles définit sans ambigüité l'ensemble des mécanismes organisationnels du système.

> On définit ainsi le **réservoir organisationnel global** $\mathcal{GM}$ comme l'union des deux précédents. Il est identique à l'union des réservoirs organisationnels spécifiques $\mathcal{SM}(\alpha)$ de tous les types d'agents $\alpha \in T_A$:
>
> $$\mathcal{GM} = \mathcal{GS} \cup \mathcal{GF} = \bigcup_{\alpha \in T_A} \mathcal{SM}(\alpha)$$

Il apparaît ici possible de formaliser, à partir des réservoirs considérés, les notions d'organisation statique et dynamique. Pour cela, nous proposons les définitions suivantes, à partir des classifications de mécanismes proposées (cf. section 4.2) :

1. Une organisation est **structurellement dynamique** si et seulement si son réservoir structurel global contient au moins un élément qui ne soit pas un mécanisme d'insertion.

   Inversement, une organisation est **structurellement statique** si et seulement si son réservoir structurel global est vide ($\mathcal{GS} = \emptyset$) ou ne contient que des mécanismes d'insertion.

2. Une organisation est **fonctionnellement dynamique** si et seulement si son réservoir fonctionnel global contient au moins un mécanisme fonctionnel hétérogène ou projectif.

   Inversement, une organisation est **fonctionnellement statique** si et seulement si soit son réservoir fonctionnel global est vide ($\mathcal{G}F = \emptyset$), soit il ne contient que des mécanismes fonctionnels homogènes et négligents.

3. Une organisation est **globalement dynamique** (on dira juste dynamique) si et seulement si elle est structurellement dynamique ou fonctionnellement dynamique.

   Inversement, une organisation est **globalement statique** (on dira juste statique) si et seulement si elle est structurellement statique et fonctionnellement statique.

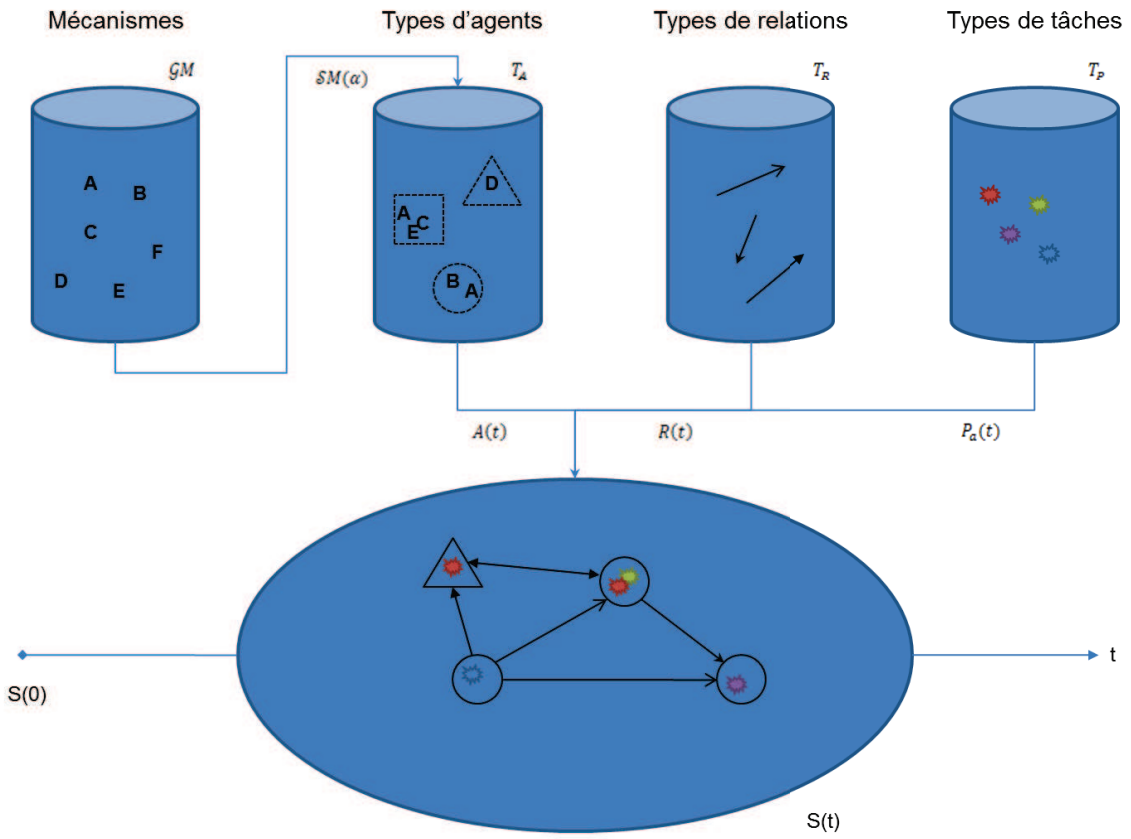


Figure 4.4 - Composantes de l'organisation d'un SMA S(t) dans le modèle proposé

4.3.2 Spécifications à partir des réservoirs

A partir des ensembles présentés ci-dessus, nous proposons une méthode de spécification par contraintes permettant de décrire une organisation avec tous ses points de spécifications et tous ses points d'incertitudes. La même méthode permettra de décrire aussi bien les organisations elles-mêmes que leurs instances.

Nous avons décrit dans la section précédente des ensembles variés ayant pour point commun d'être construits autour des éléments que nous avons choisis comme bases de notre modèle organisationnel et d'être sémantiquement reliés à la définition d'une organisation : les réservoirs (cf. Figure 4.4).

Afin de représenter les organisations comme des ensembles de possibles sur un système – ce qui suit les constatations amenées par notre approche –, nous considérons que les organisations doivent être spécifiées par des ensembles de contraintes exprimées sur leurs éléments de base, c'est-à-dire sur les réservoirs ainsi définis. En effet, il nous faut proposer une définition des organisations, et donc une méthode de description, qui permette, à partir des éléments de base que nous avons choisis – au niveau micro, dans le but même de la généricité de la description – de formaliser les organisations en laissant la liberté de ne spécifier que les contraintes nécessaires. C'est donc une approche restrictive que nous choisissons : les organisations ne sont ainsi pas définies par des accumulations d'éléments, car cela obligerait à penser, évaluer et formaliser l'ensemble de tous les degrés de libertés disponibles pour chaque organisation. A la place de cela, nous choisissons de laisser au départ l'ensemble des possibilités présentes[23], et de proposer une méthode consistant à réduire les possibilités par l'ajout successif de contraintes – ce qui n'oblige à penser et formaliser que les contraintes, c'est-à-dire le connu, le voulu, à la conception d'un système, ou bien encore l'observé, le su, à l'analyse d'un système.

Nous distinguerons donc quatre types de **contraintes** que nous pourrons poser sur les réservoirs : contraintes sur le nombre d'éléments du réservoir, contraintes sur les éléments possibles dans le réservoir, contraintes sur les éléments nécessairement inclus dans le réservoir et contraintes sur les valeurs (ensembles) possibles pour le réservoir. Chaque type possède une version positive et une version négative, ce qui en double le nombre total. Par ailleurs, les contraintes ainsi proposées peuvent être vérifiées soit à tout instant ($\forall t$), soit seulement sur un ensemble temporel défini. D'une manière générale, pour une contrainte donnée $C(t)$ – *i.e.* une proposition C définie sur un réservoir de l'une des quatre manières décrites ci-dessous et valable à l'instant t –, une **spécification** Θ est l'énoncé de la validité de cette contrainte sur un ensemble d'instants $I^T \in \mathbb{R}^+$ quelconque ; une spécification prend donc la forme d'une expression :

$$\Theta \equiv \forall t \in I^T, C(t)$$

[23] Cela rejoint l'idée naturelle que l'absence d'organisation correspond bien à un « chaos » où tout est envisageable, et non à l'absence de tout, au « néant »…

Étant donné Λ un réservoir quelconque, les différents types de spécification possibles, en fonction de la forme que prend la contrainte exprimée, sont :

- spécification **cardinale** : seul le cardinal du réservoir est spécifié, soit exactement, soit par une ou plusieurs inégalités[24] ;

  e.g. $|\Lambda| \leq 12$

- spécification **limitative** : un ensemble fini incluant le réservoir est spécifié ; le réservoir est nécessairement un sous-ensemble de l'ensemble fourni ;

  e.g. $\Lambda \subseteq \{a, b, c, d, e\}$

- spécification **constructive** : un ensemble fini inclus dans le réservoir est spécifié ; le réservoir contient donc au moins l'ensemble fourni ;

  e.g. $\{(1, \alpha), (2, \gamma)\} \subseteq \Lambda$

- spécification **alternative** : un ensemble fini contenant le réservoir est spécifié ; le réservoir fait partie des éléments de l'ensemble fourni.

  e.g. $\Lambda \in \{\{x, y\}, \{x, z\}, \{y, z\}\}$

Chacune de ces spécifications peut être prise à la négative. La sémantique change alors pour chaque type de spécification :

- la négation de la spécification cardinale permet de spécifier les cardinaux que ne peut pas prendre le réservoir – rien ne change quant aux possibilités expressives des inégalités,
- la négation de la spécification limitative permet d'exprimer que le réservoir contient nécessairement des éléments extérieurs à un ensemble,
- la négation de la spécification constructive permet d'exclure des éléments ou des ensembles d'éléments du réservoir,
- la négation de la spécification alternative permet de spécifier les valeurs que ne peut pas prendre le réservoir.

[24] On notera que malgré sa notation différente pour des raisons pratiques, il s'agit bien d'une spécification ensembliste : on peut réécrire l'exemple donné en $|\Lambda| \in [\![0\,;12]\!]$. Seule l'utilisation de l'opérateur « cardinal » en est la particularité.

Pour des raisons évidentes, nous appelons spécification **totale** toute spécification alternative positive dont l'ensemble spécifiant est un singleton. Elle définit ainsi l'unique valeur que peut prendre un réservoir.

e.g. $\Lambda \in \{\{\phi, \chi, \omega\}\}$

soit $\Lambda = \{\phi, \chi, \omega\}$

Ces spécifications peuvent donc être utilisées pour définir des propriétés voulues ou attendues sur les organisations. Nous nous en servons donc pour décrire chacune des contraintes imposées par une organisation sur le système sous-jacent. C'est donc les ensembles de spécifications – toutes reliées par des « et » logiques – qui forment les descriptions de niveau organisationnel.

4.3.3 Définitions formelles des notions globales

A partir des spécifications organisationnelles ainsi définies, nous pouvons proposer une définition complète de différentes notions haut-niveau – niveau macro – qu'elles rendent représentables, dont les notions d'organisation et d'instance.

Pour cela, il nous faut d'abord distinguer plusieurs types de réservoirs selon le rôle qu'ils jouent par rapport aux descriptions d'organisations.

> Nous appelons ainsi **réservoirs primaires** les réservoirs définissant les ensembles d'éléments du système sur lesquels les concepteurs disposent des choix de spécifications.

Ainsi, font partie des réservoirs primaires : les ensembles de types T_A, T_R, T_P, les réservoirs d'agents $A(t)$, les réservoirs de tâches pour chaque agent $P_a(t)$, les ensembles de mécanismes organisationnels pour chaque type d'agent $\mathcal{SS}(\alpha)$ et $\mathcal{SF}(\alpha)$, et enfin les ensembles de relations à $t = 0$ $R(0)$ – et seulement à $t = 0$, car si l'initialisation est bien laissée au choix des concepteurs, l'évolution des relations dans le système est spécifiée entièrement par les ensembles de mécanismes structuraux.

> Nous incluons dans l'ensemble des **réservoirs secondaires** ceux qui sont spécifiés à partir des réservoirs primaires par intégration ou restriction selon une variable : temporelle, de type – d'agent, de relation, de tâches –, ou d'élément particulier – agent, relation, tâche, mécanisme.

Ces ensembles sont donc bien plus nombreux, et comprennent, notamment : les réservoirs d'agents A, $A_\alpha(t)$, A_α, les réservoirs de tâches, $P(t)$, $P_\phi(t)$, $\mathcal{SP}(\alpha)$, $\mathcal{GP}$, les ensembles intégrés de mécanismes organisationnels $\mathcal{GS}$ et $\mathcal{GF}$, et les ensembles de relations $R_a(0)$.

> Enfin, nous caractérisons de **réservoirs tertiaires** tous les autres réservoirs (cf. définition section 4.3.1).

Ils incluent notamment ceux dont le contenu dépend de l'opération au cours du temps de mécanismes organisationnels. Ils ne peuvent pas, dans notre modèle, être spécifiés *a priori* par les concepteurs, car ils sont la résultante d'autres spécifications existantes – notamment les valeurs initiales des ensembles de relations et les attributions de mécanismes organisationnels aux types d'agents.

Parmi ces réservoirs tertiaires se trouvent : les ensembles de relations pour $t \neq 0$ $R_a(t)$ et $R(t)$, les extensions sociales des agents[25] $D_a(t)$ et les ensembles qui en sont dérivés, tels que $E_a(t)$, $C_a^{a'}(t)$, $C_a(t)$, $NC_a^{a'}(t)$ et $NC_a(t)$. S'y trouvent également tous les ensembles dérivés par intégration ou par restriction de ceux-ci ainsi que les réservoirs obtenus à partir d'autres réservoirs par des opérations d'agrégation autres que l'intégration ou la restriction selon une variable – par exemple, ceux construits par l'utilisation de l'ensemble des parties d'un ensemble connu (noté $\mathcal{P}(E)$) ou incluant des quantificateurs existentiels.

Ainsi, nous avons posé la distinction entre les réservoirs devant être connus pour définir un système unique (primaire), aptes à être utilisés dans la spécification d'une organisation (primaires et secondaires), et seulement observables car dépendants d'autres spécifications et du déroulement lui-même des instances de systèmes (tertiaires). Nous pouvons ainsi définir les notions de niveau macro dont nous voulions permettre la description.

Organisation

Pour définir une organisation, nous utilisons donc les spécifications sur les ensembles qui se prêtent à la spécification au niveau de la conception, c'est-à-dire les réservoirs primaires et secondaires.

En effet, pour décrire une organisation, nous avons tout d'abord identifié les éléments du niveau micro qui nous permettaient de fonder une description organisationnelle suivant notre approche. Ensuite, nous avons proposé des ensembles sémantiquement riches nous permettant de décrire les éléments essentiels d'une organisation. Enfin, parmi ces ensembles, en sélectionnant uniquement ceux qui apparaissent comme indépendants d'autres spécifications en amont, nous sommes à même d'exprimer des spécifications complètes ou incomplètes et pleinement cohérentes. Pour cela, il nous faut utiliser les spécifications organisationnelles, telles que définies dans la section précédente pour spécifier tous les points connus ou décidés des organisations tout en laissant libres les autres points.

---

[25] Contrairement aux ensembles de relations, les extensions sociales des agents sont calculées, pour $t = 0$, à l'aide des mécanismes fonctionnels, en supposant que $\forall a \in A(0), D_a(-1) = \emptyset$, car, étant calculées après les relations, elles n'ont pas besoin d'être initialisées *a priori*.

> Nous définissons comme **organisation** tout ensemble cohérent (i.e. n'impliquant pas de contradiction) de spécifications organisationnelles (cardinales, limitatives, constructives ou alternatives, positives ou négatives) exprimées sur des réservoirs primaires ou secondaires.

Une organisation est ainsi un ensemble de contraintes sur les ensembles d'éléments qui permettent la description organisationnelle des systèmes. Elle définit donc un ensemble de systèmes possibles, qui correspond à l'ensemble des systèmes dont la description organisationnelle valide l'ensemble des spécifications de l'organisation.

Il nous est utile de définir ici une notion d'**organisation dérivée** : une organisation $\mathcal{O}_2$ est dérivée d'une organisation $\mathcal{O}_1$ si et seulement si l'ensemble des spécifications de $\mathcal{O}_1$ subsume celui de $\mathcal{O}_2$; on dira de manière équivalente que l'ensemble des systèmes validant $\mathcal{O}_2$ est contenu dans l'ensemble des systèmes validant $\mathcal{O}_1$. On note $\mathcal{O}_1 \vDash \mathcal{O}_2$.

Deux organisations sont dites **équivalentes** si et seulement si elles dérivent l'une de l'autre – c'est-à-dire si les ensembles de systèmes qui les valident sont identiques. On note $\mathcal{O}_1 \equiv \mathcal{O}_2$.

Il est intéressant de noter ici que l'organisation vide $\emptyset$ représente donc l'absence de contraintes, et par conséquent valide tout système. Cela correspond à notre intuition, qui consiste à opposer l'organisation et le chaos : spécifier une organisation est un moyen de contraindre le système à un fonctionnement voulu. On remarquera également que, de fait, toute organisation dérive de l'organisation vide, et donc que l'ensemble des organisations, définies à l'équivalence près, forment une structure de semi-treillis supérieur pour la dérivation[26] :

$$\forall \mathcal{O}, \emptyset \vDash \mathcal{O}$$

Instance organisationnelle

Nous avons aussi parlé précédemment de la notion d'instance organisationnelle. Une instance organisationnelle est un exemple de système – donc totalement spécifié – qui représente, et donc – en des termes plus précis – qui valide une organisation particulière.

Une instance organisationnelle partage donc la structure d'une organisation – il s'agit toujours d'une description uniquement organisationnelle d'un système –, sa particularité, par

[26] On peut noter que, en revanche, il n'existe pas nécessairement de borne inférieure à un ensemble d'organisations, le contre-exemple le plus simple étant l'ensemble de deux organisations contenant respectivement une spécification positive et sa négative : aucune organisation ne peut donc dériver des deux à la fois. Nous n'obtenons donc pas une structure de treillis complète.

rapport à une organisation, étant d'être totalement spécifiée, puisqu'il s'agit d'une description complète[27] d'un système réel. Il nous est donc facile, à l'aide des éléments que nous avons utilisés pour définir une organisation, de définir une instance organisationnelle.

> Nous définissons comme **instance organisationnelle** toute organisation totalement spécifiée, c'est-à-dire contenant une spécification totale sur $t \in \mathbb{R}^+$ pour chaque réservoir principal.

Ainsi, une instance organisationnelle, par redondance, peut se réduire à la donnée de spécifications totales sur quatre ensembles statiques (T_A, T_R, T_P et $R(0)$) plus deux par type d'agents ($\mathcal{SS}(\alpha)$ et $\mathcal{SF}(\alpha)$), ainsi que sur les ensembles dynamiques sur $\mathbb{R}^+$ $A(t)$ et $P_a(t)$[28].

On dira qu'une instance organisationnelle $\mathcal{I}$ est **instance de l'organisation** $\mathcal{O}$ si et seulement si $\mathcal{I}$ est dérivée de $\mathcal{O}$ (*i.e.* si $\mathcal{O} \vDash \mathcal{I}$).

On peut noter que, du fait de l'exclusion de l'organisation nulle – qui n'est évidemment pas une instance organisationnelle –, l'ensemble des instances organisationnelles n'hérite pas de la structure de l'ensemble des organisations.

Modèle normatif

Nous avons, pour définir les notions organisationnelles que nous souhaitions, simplifié l'ensemble des éléments à notre disposition et conservé uniquement ceux qui relevaient des éléments pouvant être spécifiés à la conception. En revanche, dans l'objectif de prévoir des formules de tests pour les SMA, d'analyser et de regrouper les systèmes et leurs organisations entre eux, une notion plus générale pourrait nous être utile. C'est pourquoi nous généralisons la notion d'organisation en proposant celle de modèle normatif[29].

---

[27] On notera qu'il ne s'agit que d'une spécification organisationnelle : l'organisation est totalement spécifiée, en revanche, une instance organisationnelle peut valider plusieurs systèmes possédant exactement la même organisation, mais dont les comportements internes sont différents – notamment lorsque ceux-ci dépendent des spécifications de l'environnement ou des interactions.

[28] On peut redéfinir ce dernier comme l'ensemble dynamique $\Pi(t) = \{ (a, x) \mid a \in A(t) \wedge x \in P_a(t)\}$. Il s'agit donc bien d'un unique ensemble dynamique, même si on préfèrera, pour des raisons de pratique et de compréhensibilité, son expression sous la forme des ensembles $P_a(t)$.

[29] Nous choisissons ce nom car il correspond à un ensemble de « normes », c'est-à-dire de contraintes haut-niveau, exprimée sur une description organisationnelle. Il ne s'agit pas ici de « normes » telles que définies syntaxiquement dans les modèles organisationnels existants, et étudiées à la section 4.4.4 (bien que nous verrons, dans cette section, que certains aspects des

> Nous définissons comme **modèle normatif** tout ensemble cohérent (i.e. n'impliquant pas de contradiction) de spécifications organisationnelles (cardinales, limitatives, constructives ou alternatives, positives ou négatives) exprimées sur des réservoirs.

Ainsi, nous étendons les spécifications aux réservoirs tertiaires. Il est à noter que toute organisation (ou instance organisationnelle) est un modèle normatif. Nous réserverons cependant en général cette appellation aux modèles normatifs qui comportent des spécifications sur les réservoirs tertiaires et qui ne sont donc, à ce titre, pas des organisations. Ces modèles permettent d'exprimer des contraintes, structurelles et fonctionnelles, sur des systèmes, qui doivent pouvoir être observées à l'exécution. Ils permettent de formuler des contraintes dynamiques plus variées que les organisations, mais pour lesquelles les concepteurs de systèmes ne disposent pas nécessairement de moyen direct d'implémentation.

Nous posons les mêmes définitions de dérivation et d'équivalence sur les modèles normatifs que sur les organisations. Puisque l'ensemble des organisations fait partie de l'ensemble des modèles normatifs, l'organisation nulle $\emptyset$ est donc un modèle normatif également. Nous retrouvons ainsi – la démonstration est la même que pour les organisations – la structure de semi-treillis supérieur pour l'ensemble des modèles normatifs définis à l'équivalence près et la relation de dérivation.

Utilisation du modèle

Notre modèle propose ainsi une description des aspects organisationnels des systèmes fondée sur des ensembles d'éléments du niveau micro et déclinée en trois **représentations organisationnelles** selon le niveau de description visée : les modèles normatifs permettent de décrire des spécifications attendues, à tester ou à valider, sur les systèmes ; les organisations permettent de décrire des spécifications directement implémentables pour les systèmes ; les instances organisationnelles permettent de spécifier des systèmes précis, dans leur totalité et dans leur durée.

Nous pouvons donc, dès lors, utiliser ces trois représentations pour décrire aussi précisément que possible une grande variété de systèmes, et pour spécifier la dimension organisationnelle de systèmes ouverts ou fermés, connus à la conception ou implémentés à l'exécution, grâce à la flexibilité de la description par contraintes sur les ensembles d'éléments. Les notions de dérivation et d'équivalence permettent déjà de comparer formellement, *a priori* de toute implémentation, les représentations entre elles, et la structure

---

normes ne peuvent être exprimées que dans les modèles normatifs et non dans les organisations, car leur formalisation fait appel à des réservoirs tertiaires).

de semi-treillis supérieur nous permet d'opérer sur les représentations et d'en extraire les points communs et les différences.

4.4 Possibilités expressives du modèle

Ainsi, nous avons montré la possibilité de définir des représentations organisationnelles à partir d'éléments locaux et de spécifier d'une manière unique des organisations ouvertes ou fermées, rigides ou flexibles. Nous nous intéressons donc, dans cette partie, à comparer l'expressivité organisationnelle de notre modèle fondé sur des éléments bas-niveau avec les notions haut-niveau usuellement rencontrées dans les modèles organisationnels.

Pour effectuer cette comparaison, nous gardons à l'esprit les hypothèses de notre approche, telles que l'indépendance des dimensions descriptives et l'indépendance organisation-population (cf. section 3.1.4). Nous cherchons, à partir de notre modèle, à réexprimer en termes d'éléments locaux et de spécifications organisationnelles les notions importantes de l'état de l'art et comparons la sémantique de ces réexpressions à leur sémantique originelle. Ainsi, nous recréons, par agrégation, les notions haut-niveau usuelles des organisations, et, en suivant notre approche, nous proposons de cette manière un modèle émergentiste, c'est-à-dire capable de redéfinir les éléments sémantiques de niveau macro à partir des éléments de base de niveau micro. Ces éléments de haut-niveau sur lesquels les concepteurs de SMA peuvent, évidemment, vouloir émettre des spécifications, nous les relions ainsi à des concepts de niveau micro, ce qui permet leur prise en compte et leur implémentation dans tous types de systèmes. Cette généricité nécessaire des concepts haut-niveau obtenus, provenant de la généricité des éléments bas-niveau les constituant, opère un déplacement sémantique du modèle – comme nous le verrons sur les différents exemples – qui garantit l'indépendance de notre modèle organisationnel vis-à-vis des autres dimensions descriptives des SMA en limitant nécessairement sa portée sémantique à de pures notions organisationnelles indépendantes des codes et représentations internes des agents.

4.4.1 Rôles

Les plus courantes des notions organisationnelles apparaissant dans l'état de l'art sont les notions de rôle et de groupe. Nous les étudions ici à partir des définitions proposées dans les modèles MOISE$^+$ (Hubner *et al.*, 2002) et AGRE (Ferber *et al.*, 2005), qui dérivent de la définition de cette notion sociale originalement donnée dans (Werner, 1989) et qui nous apparaissent comme les définitions utilisées de ces notions les plus précises et les plus représentatives.

Nous commençons donc, pour positionner l'expressivité de notre modèle par rapport à l'état de l'art, par chercher à redéfinir, à partir des éléments utilisés dans notre modèle, le concept organisationnel de rôle tel que défini dans ces deux modèles. Pour cela, il nous faut étudier les définitions données dans ces modèles et en retirer ce qui est exclusivement organisationnel.

La définition posée pour le modèle MOISE+ se résume ainsi (Hubner *et al.*, 2002) :

A role means a set of constraints that an agent ought to follow when it accepts to enter a group playing that role. [..] these constraints are defined in two ways: in relation to other roles (in the collective structural level) and in a deontic relation to global plans (in the functional dimension).

Ainsi, un rôle est défini d'une manière double : à la fois à travers les relations entre agents qu'il implique et, de manière fonctionnelle, à travers les objectifs qu'il doit faire siens pour être considéré comme « jouant » ce rôle. Cette seconde approche, si elle est complémentaire à la première dans un modèle comme MOISE+, correspond bien mieux à notre approche, car elle considère le rôle comme un état de fait dynamique : les rôles peuvent ne pas être prévus à la conception, mais intervenir spontanément lors de l'exécution, de même que dans des exemples tels que l'émergence de rôles spécifiques pour des communautés humaines (Krupa *et al.*, 2009). De la même manière, dans AGRE, le rôle est posé comme fonctionnel (Ferber *et al.*, 2005) :

The role is the abstract representation of a functional position of an agent in a group.

Nous proposons donc de séparer la définition d'un rôle en deux, afin de rendre compte à la fois de cette dynamique de l'existence des rôles joués par les agents lors de l'exécution, qu'ils aient été prévus ou non, et en même temps, de la capacité à prévoir des spécifications explicites des comportements requis pour jouer un rôle particulier.

Nous définissons ainsi le **rôle effectif** q comme un ensemble d'ensemble de tâches qu'un agent peut remplir dans le système à un instant donné :

$$q \subseteq \mathcal{P}(\mathcal{G}P)$$

L'ensemble ainsi défini correspond à tous les champs d'action possibles qui valident le rôle q. Ainsi, un agent a jouera à l'instant t le rôle q au sein du système si et seulement si :

$$P_a(t) \in q$$

Pour des raisons d'efficacité lors de l'utilisation de la notion de rôle effectif dans la conception de système, on simplifie la notation des rôles lorsqu'ils représentent simplement l'obligation de présence d'au moins une tâche d'un certain type dans le champ d'action de l'agent. On note ainsi $q \equiv Q$, avec $Q \subseteq T_P$ si et seulement si :

$$q = \{ P \in \mathcal{P}(\mathcal{G}P) \mid \forall \phi \in Q, \exists x \in P, \mathrm{t}(x) = \phi \}$$

Il est intéressant de noter que la définition d'un rôle effectif pour un agent à un instant donné est une spécification de niveau organisation[30]. En effet, il s'agit d'une spécification alternative sur le champ d'action d'un agent, qui est un réservoir primaire.

Nous notons cependant que cette notion ne recouvre pas exactement la partie fonctionnelle de la notion de rôle proposée par le modèle MOISE+. La différence tient au fait que ce dernier définit comme entités fonctionnelles des « missions » (Hannoun *et al.*, 2000), qui contiennent non seulement les objectifs des agents, mais également les moyens précis de remplir ces objectifs (plans, actions, ressources). Or, la description de ces moyens est en dehors du champ descriptif de notre modèle, car appartenant à la dimension agents (et éventuellement environnement) de la description du système. Nous nous limitons donc à la description à base de tâches, ce qui réduit l'expressivité de la notion de rôle, mais seulement aux frontières de la description organisationnelle, pour conserver l'indépendance voulue entre les différentes dimensions. Cependant, il est à de noter que, par exemple, la notion d'héritage entre rôles est aisément conservée par l'utilisation de l'inclusion des ensembles de tâches que sont les rôles effectifs pour nous.

En ce qui concerne les contraintes définies entre les rôles, elles peuvent être nombreuses, et ainsi définies de manières variées dans notre modèle. Cependant, nous proposons une méthode simple pour réduire la majorité d'entre elles à une unique spécification. En effet, de nombreuses contraintes définies sur les rôles sont à la frontière de la dimension organisationnelle, en cela qu'elles contraignent le fonctionnement de l'agent qui va jouer un rôle. Or, nous n'émettons, dans notre modèle, aucune spécification sur les agents eux-mêmes. Une manière de contourner cela pour redéfinir dans notre modèle cette notion de rôle est d'utiliser des spécifications non sur les agents mais sur les types d'agents.

---

[30] A la condition qu'il s'agisse bien de dire qu'un agent en particulier joue un rôle donné, ce qui est spécifiable directement par le concepteur. En revanche, dire qu'au moins un agent du système joue ce rôle ($\forall t \in \mathbb{R}^+,\ r \in P^{part}(t)$ avec $P^{part} = \bigcup_{a \in A(t)} \mathcal{P}\big(P_a(t)\big)$) est une spécification de niveau normatif, car $P^{part}(t)$ est un réservoir tertiaire (cf. section 4.3.3). Cela est cohérent avec le fait qu'il n'est pas possible de spécifier directement cette propriété : elle doit être spécifiée, soit pour des agents précis, à travers leur comportement, soit seulement attendue.

Nous spécifions un ensemble de types d'agents conformes à l'exécution d'un rôle, ce qui définit une notion que nous appelons **rôle normatif**[31] h :

$$h \subseteq T_A$$

Un agent a est normativement conforme[32] à h si et seulement si son type appartient à l'ensemble h :

$$\mathrm{t}(a) \in h$$

Il est ainsi possible, dans un modèle normatif, de spécifier que tous les agents jouant un rôle effectif q soit du rôle normatif h. Avec $\bar{h} = T_a \setminus h$ et $\mathrm{A}_{\bar{h}} = \{a = (n_a, \alpha) \in \mathrm{A} | \alpha \notin h\}$ le réservoir correspondant à l'ensemble des agents qui ne sont pas normativement conformes à h, cela s'exprime sous la forme de l'ensemble de spécifications constructives négatives[33] :

$$\forall a \in \mathrm{A}_{\bar{h}}, \forall t\ P_a(t) \notin q$$

Par ailleurs, il est possible de définir, séparément, des ensembles de spécifications concernant les contraintes de relations existantes entre agents de certains rôles. Les contraintes entre les rôles normatifs sont naturellement définies dans notre modèle à travers les spécifications sur $R(0)$ et les réservoirs structurels et fonctionnels globaux, au niveau organisationnel ; leur expression comme des règles sur les rôles n'est donc qu'une représentation de plus haut niveau de propriétés définissables à bas niveau. Les contraintes entre les rôles efficients pouvant être énoncées au niveau normatif par des spécifications sur les ensembles de relations des agents jouant ces rôles (qui sont des réservoirs tertiaires).

4.4.2 Groupes

Les modèles utilisant la notion de rôle, que nous venons de redéfinir dans notre modèle, utilisent également souvent celle de groupe. Les groupes décrivent des ensembles d'agents ou de rôles liés d'une certaine façon.

---

[31] En opposition au rôle effectif, qui décrit une réalité organisationnelle (l'attribution de tâches), le rôle normatif décrit un ensemble permettant de spécifier des normes de comportement pour les agents (notamment de posséder les capacités et comportements adaptés pour jouer un rôle effectif).

[32] Il s'agit seulement d'une description et non d'une spécification.

[33] Avec $\Pi_\alpha(t)$ la restriction de $\Pi(t)$ aux agents de type α, on peut réécrire cela sous la forme de l'ensemble de spécifications constructives négatives $\forall \alpha \notin h, \forall t \in \mathbb{R}^+, r \notin P_\alpha^{part}(t)$, avec $P_\alpha^{part}(t) = \cup_{a \in A_\alpha(t)} \mathcal{P}\big(P_a(t)\big)$.

La plus simple à adapter des définitions de rôles provient de la propriété formulée dans le modèle AGRE (Ferber *et al.*, 2005) :

Two agents may communicate if and only if they belong to the same group, but an agent may belong to several groups.

Plus précisément, cette propriété, implique, dans le contexte de ce modèle, que deux agents ne peuvent interagir que s'ils appartiennent au même groupe.

Ainsi, un groupe est pour nous une clique du graphe structurel $\mathbb{G}(t) = < A(t), R(t) >$, puisque pour nous, deux agents ne peuvent interagir que s'ils sont en relation. Un ensemble $G \subseteq A(t)$ d'agents sera donc un groupe si et seulement si :

$$\forall a, b \in G, \exists r_1, r_2 \in R(t) \mid r_1 = (a, b, \lambda_1) \wedge r_2 = (b, a, \lambda_2) \wedge \lambda_1, \lambda_2 \in T_R$$

Il est donc aisé de spécifier l'appartenance à au moins un même groupe pour deux agents par l'existence d'une relation d'un type quelconque entre eux dans chaque sens – il s'agit d'une spécification de niveau organisationnel.

Dans le modèle AGRE, les groupes sont spécifiés par le type de groupe, qui définit les rôles et les types de relations en jeu. L'appartenance d'un agent à un groupe de ce type, dans un rôle particulier, est donc également tout à fait descriptible à travers un ensemble de spécifications sur le rôle effectif de l'agent ainsi que ses relations. L'existence du groupe lui-même est la somme des spécifications sur tous les agents qui le composent.

Bien qu'ils comprennent des éléments plus nombreux, les groupes définis dans le modèle MOISE+ suivent la même logique. Ils se composent d'un ensemble de rôles, effectifs et normatifs, et de l'existence de relations entre ces rôles. Comme nous l'avons vu, ces spécifications sont à même d'être décrites avec précision au niveau organisationnel de notre modèle.

Ainsi, la notion de groupe, bien qu'elle perde une partie de sa sémantique, à cause de l'absence de spécification de la dimension agent dans notre modèle purement organisationnel, est tout à fait exprimable à partir des éléments de notre modèle, et peut être analysée de la même manière.

4.4.3 Coalitions, équipes

Certaines notions organisationnelles sont cependant intrinsèquement plus dynamiques dans leur définition, et donc plus adaptées à être exprimées dans notre modèle. L'exemple caractéristique de cela est celui de la notion de coalition.

Une coalition est un ensemble d'agents qui partagent un objectif commun et qui sont capables de collaborer en vue de cet objectif. Cette notion est utilisée dans un certain nombre

de travaux sur les SMA dans le but de définir les ensembles d'agents concernés par des mécanismes de répartition ou de délégation de tâches (cf. sections 2.3.4 et 2.3.5) ; en revanche, elle peut aussi être utilisée de manière plus statique, avec une vue plus hiérarchique des objectifs et sous-objectifs, dans la notion d'équipe, qui se différencie par le fait qu'elle suppose une spécification de l'ensemble d'agents et des tâches à remplir qui soit préalable à l'exécution du système(cf. modèle TÆMS, section 2.2.2).

> Une coalition est donc représentable à travers deux éléments distincts : tout d'abord un ensemble d'agents, qui se doivent d'avoir la propriété d'être tous capables d'interagir entre eux ; et ensuite, un ensemble de tâches, qui représente l'ensemble des objectifs partagés par les agents.

Pour spécifier, dans notre modèle, qu'un ensemble d'agents $A_C \subseteq A(t)$ forme une coalition à l'instant t, il faudra donc tout d'abord s'assurer qu'il forme une clique dans le graphe structurel de l'organisation – on note $R_{a_1}^{a_2}(t)$ l'ensemble des relations entre l'agent source a_1 et l'agent cible a_2, *i.e.* le réservoir secondaire formé par restriction de $R_{a_1}(t)$ à l'agent cible a_2 :

$$\forall a_1, a_2 \in A_C, R_{a_1}^{a_2}(t) \neq \emptyset$$

Il est ensuite nécessaire que l'ensemble des agents de A_c partage un ensemble de tâches communes qui sont les objectifs de la coalition :

$$P_C(t) = \bigcap_{a \in A_C} P_a(t) \neq \emptyset$$

> La coalition est donc définie par la paire :
>
> $$\mathcal{C}_C(t) = \big(A_C(t), P_C(t)\big)$$

Dans la durée, on définira la coalition comme l'intégration sur tous les instants considérés. A condition que les agents considérés appartiennent au système sur toute la durée, *i.e.* $\forall t \in [t_1, t_2[\ A_C \subseteq A(t)$, on a :

$$\mathcal{C}_C^{t_1,t_2} = \left(A_C, P_C^{(t_1,t_2)}\right) = \left(A_C, \bigcap_{t \in [t_1,t_2[} P_C(t)\right)$$

Cette coalition n'existe entre t_1 et t_2 que si l'ensemble des tâches communes conservées du début à la fin de la coalition n'est pas vide :

$$P_C^{(t_1,t_2)} \neq \emptyset$$

Il est à noter que l'existence et la description d'une coalition et éventuellement de sa composition en terme d'agents, de tâches, mais également de relations est entièrement effectuée, dans notre modèle, au niveau organisationnel. Ainsi, les coalitions sont, comme leur définition usuelle le laissait penser, de très bons outils de descriptions de structures pour notre modèle.

4.4.4 Normes

Une dernière notion organisationnelle usuelle nous semble importante ici pour rendre compte de la manière dont notre modèle permet de reformer les concepts habituels des organisations en des termes plus génériques et bas-niveau. Il s'agit de la notion de norme[34].

La notion de norme est un concept organisationnel de très haut niveau, complexe dans sa définition comme dans ses applications (Hexmoor *et al.*, 2006). Il décrit un ensemble d'obligations et de comportements requis pour les agents dans une organisation, selon les places qu'ils occupent. Cette notion recouvre des termes aussi variés que le respect des contrats et des obligations sociales, les conventions et protocoles, les comportements nécessaires des agents et leur surveillance par le système.

Au premier abord, il semble donc s'agir d'une notion fort éloignée de notre modèle, en ce qu'elle implique nécessairement des spécifications sur les dimensions agent et interactions des SMA, et est donc hors de portée de notre modèle qui s'attache uniquement à la dimension organisation. Cependant, de la même façon que pour les rôles normatifs, il nous est possible de contourner cette limitation en ne décrivant jamais formellement les comportements attendus, mais en les regroupant sous forme de spécifications sur les types d'agents et les mécanismes organisationnels qu'ils utilisent.

Pour cela, il nous faut tout d'abord distinguer les différents aspects de la notion de norme, afin de pouvoir décrire précisément ce qu'ils recouvrent et comment les spécifier dans notre modèle.

A la suite de l'étude de la description des normes dans l'état de l'art, nous distinguons trois aspects dans la notion de norme :

- *L'aspect comportemental* : il s'agit de définir, pour un agent, le comportement qu'il doit adopter d'un point de vue fonctionnel par rapport au système ; il spécifie les tâches qu'il doit réaliser, selon sa place dans le système et ses interactions antérieures.
- *L'aspect relationnel* : il s'agit de définir de quelle manière chaque agent a le droit de nouer ou rompre des relations avec les autres agents, et quelles sont les contraintes qui lui sont imposées dans ce domaine.
- *L'aspect autoritaire* : il s'agit là de définir comment, au sein du système, le respect des normes est assuré ; les spécifications de cet aspect concernent les autorités –

[34] Nous parlons ici de la norme telle que décrite et utilisée dans les modèles organisationnels (cf. section 2.2), en opposition à la notion générale de norme menant aux diverses appellations de notre modèle (modèle normatif, rôle normatif, etc.).

éventuellement des agents du système – chargés de la vérification du respect des normes et les sanctions éventuelles que les agents du système peuvent prendre à l'encontre de ceux qui ne respectent pas les normes.

Ces trois aspects sont très différents du point de vue des spécifications qu'ils impliquent, bien qu'ils soient les différentes parties d'un même tout qui est la spécification et la régulation des comportements sociaux au sein d'une organisation donnée. Nous déduisons de cela que nous devons séparer la description d'une norme dans notre modèle en trois définitions différentes, correspondant aux spécifications de chacun de ces aspects.

Aspect comportemental

L'aspect comportemental d'une norme est tout d'abord représentable, pour nous, dans une certaine mesure, par des spécifications sur les tâches que doivent accepter ou non les agents dans différentes conditions : seule la dimension organisationnelle de cet aspect est représentable, tandis que la dimension agent ne l'est pas[35] (le fait, notamment, qu'un agent soit réellement capable de remplir les tâches qu'il accepte ou non). Il se fonde sur des ensembles de spécifications portant sur les champs de coopération et de non-coopération des agents. Ces spécifications portent soit directement sur ces réservoirs, soit sur des sous-ensembles de ceux-ci concernant uniquement certains types d'agents sources ou certains types de relations – par exemple des types de relations correspondant à des relations hiérarchiques.

Une manière simple de spécifier une obligation de répondre à une demande peut être notamment de spécifier que le champ de non-coopération des agents d'un certain type α, limité[36] aux relations hiérarchiques de type λ, doit être vide :

$$\forall a \in A \mid \mathrm{t}(a) = \alpha, \forall t \in \mathbb{R}^+, NC_{a,\lambda}(t) = \emptyset$$

Aspect relationnel

L'aspect relationnel des normes impose des comportements sociaux particuliers : il définit la manière dont les agents peuvent nouer ou rompre des relations, et la manière dont ils sont autorisés à collaborer, selon la nature des agents et des relations considérées. En règle générale, nous ne souhaitons pas, dans notre modèle, spécifier directement les relations entre agents, vu que ce sont les mécanismes organisationnels qui sont à l'origine de l'évolution de ces éléments.

---

[35] Du fait de notre approche volontairement limitée à la dimension organisationnelle.

[36] Pour définir cet ensemble, il suffit de remplacer, dans sa définition, $E_a(t)$ par $E_{a,\lambda}(t)$, qui est lui-même défini en fonction de $D_{a,\lambda}(t) \subseteq \{(x, a_c) \mid (a, a_c, \lambda) \in R_a(t)\}$ et non de $D_a(t)$.

Ainsi, nous décrirons simplement les normes relationnelles comme les ensembles de mécanismes organisationnels qui, à partir d'une condition initiale respectant les normes, conservent le respect de ces normes au cours du temps. Nous définirons donc une **norme relationnelle** $\mathcal{N}^R$ comme un sous-ensemble du réservoir organisationnel global $\mathcal{GM}$:

$$\mathcal{N}^R \subseteq \mathcal{GM}$$

Un agent de type α sera respectueux[37] de la norme si et seulement si il ne met en jeu que des mécanismes validant cette norme, *i.e.* dont le réservoir organisationnel spécifique $\mathcal{SM}(\alpha)$ est inclus dans l'ensemble des mécanismes respectant la norme :

$$\mathcal{SM}(\alpha) \subseteq \mathcal{N}^R$$

Aspect autoritaire

Enfin, pour représenter l'aspect autoritaire de la norme, nous pouvons proposer deux ensembles définissant les éléments qui permettent, au sein du système, d'assurer le maintien des normes.

Le premier s'applique dans le cas où un certain nombre d'agents, bien définis, sont chargés de veiller à faire respecter les normes établies pour l'organisation.

Nous définissons alors l'**autorité normative** $\mathcal{AN}$ comme l'ensemble des types d'agents capables d'assurer la surveillance normative du système :

$$\mathcal{AN} \subseteq \mathrm{A}$$

Le second s'applique en plus dans les cas où le maintien des normes est assuré par des fonctions sociales, telles que des fonctions de confiance ou de punition sociale.

Nous définissons alors la **norme autoritaire** $\mathcal{N}^A$ comme l'ensemble des mécanismes organisationnels permettant aux agents d'assurer le respect des normes dans le système ou au moins de punir les contrevenants :

$$\mathcal{N}^A \subseteq \mathcal{GM}$$

Nous sommes ainsi capables de définir, à travers les éléments de notre modèle, un ensemble d'éléments normatifs, qui, d'ailleurs, relèvent pour la plupart du niveau normatif des spécifications, qui permettent de spécifier la partie purement organisationnelle de la description des normes. Encore une fois, nous perdons une partie de la sémantique originelle de ces notions, puisqu'elle relève des autres dimensions descriptives des SMA. Cependant,

[37] Il est à noter que nous ne sommes à même de décrire ici que la potentialité qu'un agent a, de par les mécanismes qu'il met en jeu, de violer la norme. Il nous est impossible de représenter ici une violation effective de la norme, puisque la représentation de cette violation n'est pas du domaine organisationnel, mais de celui de la représentation du comportement de l'agent.

malgré le relatif éloignement entre notre approche et l'approche normative des organisations, il apparait qu'il nous est tout de même possible de spécifier un nombre important d'informations liées aux normes à travers notre modèle.

Chapitre 5. Méthodes calculatoires pour les organisations

A partir des descriptions organisationnelles que notre modèle permet, nous souhaitons, dans notre approche, être à même de calculer des propriétés globales, statiques ou dynamiques, sur les organisations, afin de rendre possibles l'étude et la comparaison objective de ces dernières selon des critères définissables. Les apports d'une telle possibilité seraient tout d'abord la capacité d'exprimer des propriétés dynamiques sur les organisations et ensuite la possibilité de comparer entre elles des organisations diverses dans leur nature – qu'elles mettent en jeu des architectures d'agents différents, des environnements dissemblables, des protocoles de communications variés, etc.

La difficulté d'une telle entreprise provient de deux sources : l'incomplétude des connaissances sur le système et la complexité des mécanismes organisationnels. Les connaissances sur le système formalisé sont en effet incomplètes, au vu de notre approche de la description organisationnelle : les descriptions étant des ensembles de contraintes spécifiant l'ensemble du connu sur un système, l'étude d'une organisation repose sur des connaissances incomplètes – sauf dans le cas de l'étude d'une instance organisationnelle précise. D'autre part, les mécanismes organisationnels sont des algorithmes quelconques, dont les entrées peuvent dans certains cas prendre leur valeur dans des espaces très grands. Leur étude peut donc se révéler difficile, d'autant plus que les connaissances sur ces valeurs d'entrée peuvent être incomplètes.

Il est ainsi nécessaire, pour être à même d'effectuer des calculs organisationnels dans le plus grand nombre de cas possibles, de développer des méthodes permettant de simplifier l'expression des mécanismes organisationnels et d'agréger les connaissances sur une organisation afin de permettre l'étude calculatoire des propriétés de cette organisation. Nous montrons qu'ainsi, il est possible de faire des analyses globales des organisations, mais aussi, en sur-spécifiant les organisations, d'étudier des cas particuliers de fonctionnement de ces organisations dans des situations choisies.

5.1 Représentation dynamique dans le modèle

Dans notre modèle, la représentation de la dynamique s'effectue d'abord localement – au niveau micro. C'est lors des étapes d'analyse des organisations qu'il faudra donc effectuer les agrégations nécessaires à l'obtention de propriétés globales – *i.e.* de niveau macro.

Au niveau micro, c'est-à-dire du point de vue d'un unique agent a, la dynamique extérieure de cet agent peut être décrite par :

- sa période d'existence $L(a)$;
- ses objectifs au cours du temps $P_a(t)$;
- l'ensemble de ses relations avec d'autres agents $R_a(t)$;
- l'influence qu'il a sur les autres : son extension sociale $D_a(t)$;
- l'influence que les autres ont sur lui : son champ extérieur $E_a(t)$.

Si le champ extérieur $E_a(t)$ est calculable à partir de la connaissance des extensions sociales des autres agents (puisqu'il s'agit des tâches affectées à l'agent a par les autres agents du système), tous les autres ensembles listés ici sont propres à l'agent a choisi. La période d'existence et les objectifs de l'agent lui sont propres et sont spécifiables indépendamment pour chaque agent. En revanche, l'ensemble des relations et l'extension sociale d'un agent ne sont spécifiables dans une organisation que pour $t = 0$, car leur évolution – *i.e.* leur dynamique, ce qui nous intéresse ici – est en réalité déterminée par les mécanismes organisationnels de l'agent.

De ce fait, les mécanismes organisationnels jouent une part très importante de la description de la dynamique organisationnelle. Ils reflètent en effet de manière synthétique l'évolution des ensembles dynamiques liés à des notions organisationnelles : les ensembles de relations pour chaque agent $R_a(t)$ et les extensions sociales des agents $D_a(t)$ – et ainsi sur les champs extérieurs $E_a(t)$, par agrégation. Puisque les mécanismes organisationnels sont affectés de manière univoque à chaque type d'agents, ce sont les ensembles de mécanismes affectés à chaque type d'agents, c'est-à-dire les réservoirs structurels spécifiques $\mathcal{SS}(\alpha)$ et les réservoirs fonctionnels spécifiques $\mathcal{SF}(\alpha)$ de chaque type d'agent[38] α qui définissent la partie spécifiable de la dynamique structurelle et fonctionnelle des organisations.

Nous étudions la dynamique des organisations en nous attachant à ces ensembles qui permettent de spécifier d'une manière générique le comportement organisationnel des agents, et donc la dynamique au niveau micro. Il est ainsi nécessaire de connaître, au moins de façon partielle, le contenu de ces ensembles pour les types d'agents présents dans le système pour être capables d'exprimer des propriétés organisationnelles globales qui soient significatives. Il est ensuite possible d'étudier la dynamique des organisations avec plusieurs niveaux différents de connaissance – aucune, statistique, partielle et statistique, complète –

[38] Ou plus synthétiquement le réservoir organisationnel spécifique $\mathcal{SM}(\alpha)$ de chaque type d'agent.

sur les autres ensembles prenant part à la description de l'évolution du système, selon que l'on souhaite étudier l'organisation dans sa plus grande généralité, ou étudier cette organisation dans un cas de fonctionnement plus particulier[39].

Les ensembles sur lesquels des connaissances variables peuvent être définies selon le niveau de connaissance voulu (ou possible) pour l'étude sont :

- les agents présents dans le système $A(t)$, définissant ce que nous nommons la **dynamique existentielle** du système, c'est-à-dire sa composition en terme d'agents ;
- les objectifs de chaque agent dans le système $\forall a, P_a(t)$[40], définissant ce que nous nommons la **dynamique téléonomique** du système, c'est-à-dire l'évolution des buts des agents du système et ainsi du système lui-même ;
- l'ensemble initial des relations entre agents $R(0)$, définissant les conditions structurelles initiales du système.

Le principe que nous suivons pour le calcul de propriétés dynamiques sur les organisations consiste premièrement en un calcul d'informations au niveau micro pour chaque type d'agent à partir des mécanismes organisationnels de ce type d'agent et des autres informations dynamiques sur le système, puis en l'agrégation des informations pour les différents types d'agents, et enfin en l'agrégation des différents types d'informations en propriétés compréhensibles et exploitables. Il est à noter que l'agrégation des informations entre types d'agents peut donner lieu à un apport de connaissances permettant de boucler et de recalculer des informations plus précises pour chaque type d'agents – on s'en réfèrera alors à des méthodes itératives (cf. section 5.3.2) pour obtenir l'information la plus précise possible.

5.2 Approche probabiliste du calcul de propriétés

La complexité potentielle des espaces d'état des mécanismes organisationnels ainsi que les incertitudes sur les organisations – c'est-à-dire le manque de connaissances *a priori* sur les systèmes instanciant une organisation – impliquent d'utiliser une méthode pour à la fois

---

[39] Plus le cas d'utilisation est spécifique, plus les propriétés calculées seront significatives et précises, l'extrême étant les propriétés calculées sur des instances ; cependant, les propriétés perdent en généralité ce qu'elles gagnent en précision.

[40] On peut remplacer l'ensemble des $P_a(t)$ par l'ensemble $\Pi(t)$, tel que défini section 4.3.3 – instances.

réduire la complexité de l'exploration de ces espaces d'état et acquérir de l'information dans un contexte incertain.

Nous proposons dans cette section l'utilisation d'une approche probabiliste pour le calcul de propriétés organisationnelles. Une telle approche nous permet tout d'abord d'exprimer de manière précise l'incertitude vis-à-vis des systèmes à travers l'attribution de valeurs probabilistes plutôt que de valeurs fixes. Elle nous permet également de simplifier les espaces d'états des mécanismes en réduisant ces algorithmes exacts à espaces d'états étendus à des algorithmes probabilistes à espaces d'états plus réduits.

L'approche probabiliste implique que les propriétés globales obtenues reflètent majoritairement non pas des occurrences exactes d'évènements ou de situations, mais plutôt des probabilités d'occurrences, ou encore des moyennes[41]. C'est en effet le type de propriétés le plus facile à obtenir avec certitude sur n'importe quelle organisation – y compris les organisations seulement partiellement spécifiée. Par ailleurs, les probabilités permettent malgré tout de spécifier complètement les propriétés qui peuvent l'être – simplement à travers l'utilisation de probabilités binaires : 0 ou 1 –, et donc de calculer éventuellement des propriétés globales exactes, par exemple sur les instances organisationnelles.

5.2.1 Justification de l'approche probabiliste

Nous choisissons notre approche de modélisation des mécanismes à travers la nécessité de répondre à deux problèmes posés par le calcul sur les organisations : la grandeur des espaces d'états et l'incertitude.

Plusieurs approches ont été envisagées, dont la modélisation probabiliste des incertitudes (Lindley, 2006), la modélisation des mécanismes organisationnels parallèles par des réseaux d'automates cellulaires (Saidani, 2003 ; Saidani, 2004 ; Nehnouh et Debbat, 2009), ou encore l'expression des états du système à l'aide de la logique floue (Celikyilmaz et Türksen, 2009). Si chacune de ces approches peut être retenue avec avantage pour des cas particuliers (cf. *infra*), les deux dernières manquent de la généricité d'expressivité de l'approche probabiliste.

Nous voyons en effet que, sur une organisation qui n'est pas instanciée, les informations dont nous disposons sont incomplètes. Que ce soit l'existence des différents agents et leurs entrées et sorties du système, les conditions initiales ou encore les tâches des agents au cours du temps, des éléments manquent pour être capable de calculer de manière exacte sur ces

[41] De la même manière que la température d'un objet reflète une moyenne de l'agitation de ses atomes.

organisations. L'utilisation de la logique floue pour répondre à ce problème implique d'être capable de classifier les situations en ensembles sémantiquement cohérents ; or, classifier, par exemple, les différents cas d'entrées et de sortie des agents du système, apparait comme problématique, car aucune notion générale ne permet de regrouper différents cas.

Il apparait donc, face à des problématiques de description temporelle et d'incomplétude des informations, que ce n'est pas la logique floue mais la théorie des probabilités qui est la plus appropriée: en effet, les autres méthodes envisagées ne permettent pas de répondre à la fois à l'objectif d'agrégation des informations et au traitement de l'incertitude sur certaines informations locales. On retiendra cependant, pour les perspectives d'utilisation de notre modèle, que certains cas d'étude – par exemple, les systèmes fermés à mécanismes homogènes pour la modélisation par automates cellulaires, et les systèmes dotés d'une fonction de contrôle globale pour la logique floue – favorisent les autres méthodes, qui peuvent alors apporter des résultats plus précis et complets que l'approche probabiliste.

Les probabilités sont une réponse usuelle à la difficulté du raisonnement sur un problème incertain (Lindley, 2006). Elles permettent, dans ce contexte, de définir certaines valeurs de manière non unique en utilisant des variables aléatoires, d'exprimer l'effet de certains paramètres sur des propriétés à travers les probabilités conditionnelles et d'agréger des valeurs incertaines à travers des indicateurs statistiques, tels qu'espérance, variance, etc. Par ailleurs, l'utilisation des probabilités nous offre un moyen de simplifier l'expression des mécanismes organisationnels pour réduire la complexité du calcul de propriétés tout en conservant une information suffisante pour assurer la précision de ces propriétés.

> *Pour donner un exemple de ce qu'il est possible de faire, nous prenons le cas d'un algorithme $\mathcal{F}(t)$ qui, à partir de l'historique d'un agent, compte le nombre d'occurrences passées d'un évènement e particulier. L'espace d'états d'origine de l'algorithme est très grand, car pour chaque $t \in \mathbb{R}^+$ un ou plusieurs évènements différents peuvent être mémorisés, avec pour chacun une certaine quantité d'informations supplémentaires. À partir de cette présentation de l'algorithme, la prédiction et le calcul sont très difficiles, car ils impliquent de générer des historiques cohérents et significatifs des propriétés qui doivent être testées. Dans une approche probabiliste, en revanche, si nous connaissons la probabilité p_e d'occurrence de l'évènement e à un pas de temps donné, nous pouvons affirmer que l'algorithme se résume à :*
>
> $$\mathcal{F}(t+\Delta t) = \begin{cases} \mathcal{F}(t) \text{ avec une probabilité } 1-p_e \\ \mathcal{F}(t)+1 \text{ avec une probabilité } p_e \end{cases}$$

Une telle approche permet d'obtenir une expression plus compacte et plus générale de l'algorithme, tout en conservant la majorité de l'information qu'il contient, dans un contexte probabiliste permettant d'effectuer ensuite des calculs sur cette représentation, puisqu'il n'y a

plus besoin de prendre en compte l'espace d'état d'origine de l'algorithme, mais seulement son espace de résultats – qui est ici $\mathbb{N}$.

Nous choisissons donc de modéliser, plutôt que des connaissances exactes sur les organisations spécifiées, des connaissances probabilistes décrivant les chances pour un agent quelconque d'un type donné de se retrouver dans une situation particulière au sein d'un système quelconque instanciant l'organisation spécifiée. Cela nous permet ainsi d'obtenir des connaissances directes sur les organisations et non sur leurs instances.

5.2.2 Utilisation des éléments du modèle

Afin d'effectuer une étude probabiliste des organisations, il nous faut préciser la méthode suivie pour obtenir de l'information à partir des spécifications composant l'organisation.

L'étude de la dynamique d'une organisation se fonde donc ici sur la connaissance que l'on peut avoir des réservoirs organisationnels spécifiques $\mathcal{SM}(\alpha)$ pour chaque type d'agents α. Il est donc nécessaire, dans un premier temps, de calculer des propriétés agrégées temporellement séparément sur chaque type d'agent, pour ensuite agréger les résultats obtenus en propriétés globales.

Puisque l'étude ne s'effectue pas dans le cas d'un système instancié et connu, mais d'un ensemble de spécifications incomplètes, il est nécessaire de faire de calculs sur des agents « théoriques », représentants de leur type, mais dont les détails d'existence ne peuvent pas être spécifiés. Ainsi, il sera impossible de définir précisément la période d'existence $L(a)$ et les tâches acceptées $P_a(t)$ de l'agent a de type α sur lequel nous effectuons un calcul général concernant la dynamique des agents de type α. Il sera en revanche possible de prendre pour hypothèse l'existence de l'agent a dans le système, et de spécifier ces ensembles – pour l'agent a et pour les autres agents du système – de manière probabiliste[42] de manière à avoir accès à l'information nécessaire au calcul sur cet agent. Le calcul effectué concerne alors les ensembles représentatifs de la dynamique organisationnelle au niveau de cet agent : les relations $R_a(t)$ et l'extension sociale $D_a(t)$. C'est la connaissance de l'évolution au cours du temps de ces ensembles qui permet de définir des propriétés dynamiques pour l'organisation, car il est par la suite possible d'agréger les connaissances sur les différents types d'agents en fonction de l'existence – supposée – des agents de ces types dans le système pour obtenir de l'information sur celui-ci.

[42] De la même manière que pour l'occurrence de l'évènement e dans l'exemple de la section précédente.

Après avoir effectué cette étude pour chacun des types d'agents du système, il est possible d'utiliser les connaissances obtenues pour étudier rétroactivement la dynamique de structure de l'organisation considérée – à travers l'évolution des $R_a(t)$ – ainsi que sa dynamique fonctionnelle – à travers l'évolution des extensions sociales $D_a(t)$, des champs extérieurs $E_a(t)$ calculables à partir de ces dernières, et des autres réservoirs dérivés de ceux-ci et des champs d'actions $P_a(t)$. C'est un ensemble de connaissances représentées sous forme de probabilités qui donnera accès à des informations générales sur les organisations, qui peuvent ensuite être formalisées sous une forme agrégée de propriétés numériques globales.

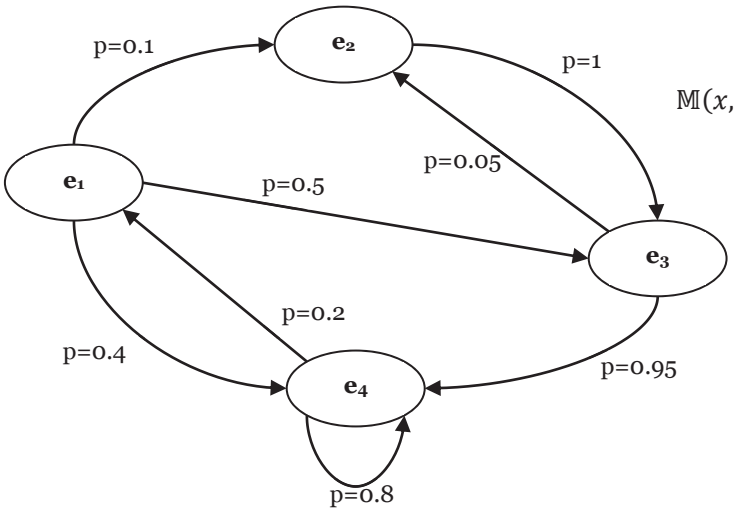
Dans certains cas, il sera nécessaire d'itérer le processus de description et de calcul pour chaque type d'agents, afin de réutiliser l'information obtenue pour les autres types d'agents dans les calculs pour un type donné – par exemple lorsque les autres agents influencent les probabilités d'acceptations de certaines tâches par l'agent. Dans ce cas, nous ferons appel à des processus itératifs pour assurer la calculabilité et la meilleure précision possible des informations désirées (cf. section 6.3.3).

5.2.3 Objectifs des calculs et représentation markovienne

L'objectif des calculs probabilistes effectués sur les organisations est d'obtenir des informations sur les états possibles et l'évolution des ensembles $R_a(t)$ et $D_a(t)$, représentant respectivement les aspects structurels et fonctionnels de l'organisation considérée, au niveau d'un agent d'un type donné. Seulement, il ne s'agit pas d'un agent réel, mais d'une connaissance moyenne, sur un agent abstrait, représentant moyen de son type, ce qui oblige à rendre probabilistes toutes les informations sur les agents qui ne sont pas communes à tous les agents d'un même type. Les informations attendues sur $R_a(t)$ et $D_a(t)$ sont donc également des informations probabilistes, synthétiques, sur les états possibles que peuvent prendre ces ensembles. Il nous faut adopter une méthode permettant, à partir des informations locales du point de vue temporel que sont les mécanismes organisationnels, de calculer des informations globales du point de vue temporel que sont les probabilités des différents états de ces ensembles.

Pour atteindre cet objectif, il nous faut disposer d'une méthode permettant d'agréger temporellement l'information afin de créer des indicateurs statistiques sur l'évolution de ces ensembles. Il n'existe finalement que peu de moyens de procéder à ces agrégations ; parmi les processus stochastiques, qui sont le moyen de répondre à ce problème, le plus simple et puissant est l'utilisation d'un **processus de Markov** (Stroock, 2005). En effet, lorsqu'il est possible de modéliser un processus temporel comme un modèle markovien, il est ensuite possible d'agréger les informations temporelles en calculant un état « moyen » du processus, dit **état stationnaire**, qui reflète les probabilités du système de se trouver, à un instant

donné, dans un état particulier. Une chaîne de Markov est définie par un ensemble d'états, représentant les états possibles du système étudié, et par l'ensemble des probabilités de transition d'un état à un autre à chaque pas de temps (cf. Figure 5.1). Ces probabilités, qui représentent de manière explicite la dynamique du système, sont exprimées sous forme matricielle – la matrice dite « de transition ». Cette représentation est un avantage dans notre situation où nous souhaitons conserver une représentation la plus précise possible de la dynamique de l'organisation.



$$\mathbb{M}(x,y) = \begin{pmatrix} 0 & 0.1 & 0.5 & 0.4 \\ 0 & 0 & 1 & 0 \\ 0 & 0.05 & 0 & 0.95 \\ 0.2 & 0 & 0 & 0.8 \end{pmatrix}$$

Figure 5.1 - Représentation d'un exemple de chaîne de Markov à travers ses états et probabilités de transition ainsi que sa matrice de transition associée

La difficulté de cette méthode est qu'elle requiert que les processus considérés soient dotés de la propriété markovienne[43] : il s'agit de l'indépendance de l'état suivant l'état actuel vis-à-vis des états passés – autres que l'état actuel. Un processus est markovien si et seulement si la probabilité conditionnelle d'un état suivant donné, sachant l'historique des états du processus est égal à la probabilité conditionnelle de cet état suivant sachant uniquement l'état actuel. Ainsi, la séquence $X_n, n \in \mathbb{N}$ de variables aléatoires est un processus de Markov si et seulement si, pour x un état quelconque du processus :

$$\forall n \in \mathbb{N}, \forall x, \mathbb{P}(X_{n+1} = x \mid X_0, X_1, \dots, X_n) = \mathbb{P}(X_{n+1} = x \mid X_n)$$

Le problème qui se pose à nous est que de nombreux mécanismes organisationnels ne valident pas naturellement cette propriété. C'est donc à travers leur simplification lors de leur transformation en expression probabiliste qu'il nous faut veiller à obtenir des processus

[43] Plus précisément, qu'ils soient des processus **homogènes** dotés de la propriété markovienne, *i.e.* dont la matrice de transition ne change pas au cours du temps.

aléatoires qui valident la propriété markovienne. Nous montrerons ensuite (section 5.3.2) comment nous atteignons cet objectif selon les propriétés des mécanismes considérés. Pour l'introduire simplement, l'opération effectuée consiste à ramener, lorsque nécessaire, les ensembles d'états définissant les différents passés menant au même futur en un seul afin d'obtenir des connaissances sur les probabilités de transitions à partir du seul état présent.

Une fois les mécanismes organisationnels exprimés sous la forme markovienne souhaitée, il est alors possible de calculer les états stationnaires de ces processus afin d'agréger l'information temporelle disponible et de déduire des propriétés globales sur les organisations considérées – d'abord par type d'agent, puis pour l'organisation dans son ensemble.

5.3 Méthode de description des mécanismes

Nous décrivons dans cette section la méthode suivie pour exprimer les mécanismes organisationnels sous une forme permettant le calcul probabiliste : des chaînes de Markov. Il s'agit de processus markoviens discrets dans le temps et dans l'espace d'états dont les états sont directement observables. Cela correspond à nos mécanismes qui :

- agissent suivant un pas de temps Δt donné (cf. section 4.2.3) ;
- disposent d'un espace d'états que nous voulons ramener à un espace discret fini, car il s'agit du seul moyen de récupérer une information sémantiquement significative sur l'organisation ; nous décrivons dans les méthodes proposées les moyens de transformer cet espace d'état ;
- représentent des états organisationnels locaux, qui sont donc directement exprimables – donc « observables » – à partir des mécanismes organisationnels.

L'expression des mécanismes sous forme de chaînes de Markov se fait ainsi plus ou moins difficilement selon les espaces d'états manipulés initialement par ces mécanismes. Cette complexité correspond à la difficulté d'obtenir de l'information générale sur les organisations impliquant ce mécanisme, et à la nécessité de sur-spécifier le système analysé pour obtenir une information précise : nous appelons ce paramètre la **spécificité** du mécanisme.

5.3.1 Représentation par chaînes de Markov

Une chaîne de Markov est un automate probabiliste exprimant le comportement d'un système à travers les probabilités de celui-ci de passer de son état actuel à n'importe quel autre état possible en fonction de ce seul état actuel – sans référence à l'historique des états

précédents[44]. Les chaînes de Markov sont donc décrites par deux éléments : l'ensemble des états possibles et la **matrice de transition**, correspondant à la matrice bidimensionnelle $\mathbb{M}(x, y)$ des probabilités de transitions entre un état x à l'instant t et un état y à l'instant $t + \Delta t$.

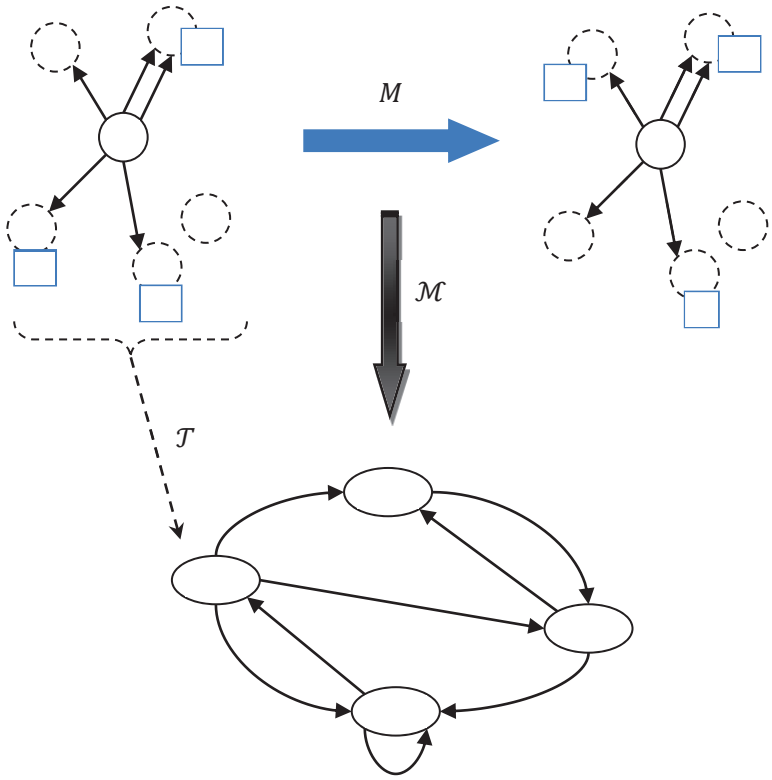


Figure 5.2 - La modélisation markovienne implique la transformation de l'espace d'états du mécanisme vers un espace d'état de chaîne de Markov

De cette façon, la modélisation d'un mécanisme correspond à la définition d'une transformation $\mathcal{M}$ telle que :

$$\mathcal{M} : M \longmapsto \mathcal{M}(M)$$

Soit, avec $\mathcal{E}_a(t)$ l'ensemble correspondant aux états manipulés ($R_a(t)$ pour les mécanismes structuraux, $D_a(t)$ pour les mécanismes fonctionnels), $\mathcal{Y}_a(t)$ l'ensemble des autres états pris en compte par le mécanisme ($\mathcal{Y}_a(t) \subseteq \xi_a(t) \times P_a(t)$ pour les mécanismes

[44] Nous considérons uniquement, comme précisé précédemment (section 5.2.3), les chaînes de Markov homogènes, dans lesquelles les probabilités de transition ne dépendent pas de l'instant considéré (elles sont indépendantes de la variable t).

structuraux et $\mathcal{Y}_a(t) \subseteq \xi_a(t) \times P_a(t) \times R_a(t)$ pour les mécanismes fonctionnels) et $w_a(t) \in \mathcal{W}_\alpha$ l'état de la chaîne de Markov à l'instant t :

$$\mathcal{M}: \Big(\big(\mathcal{Y}_a(t), \mathcal{E}_a(t)\big) \mapsto \mathcal{E}_a(t + \Delta t)\Big) \longmapsto \big(w_a(t) \mapsto w_a(t + \Delta t)\big)$$

Pour modéliser un mécanisme sous forme de chaîne de Markov, il faut donc définir tout d'abord (cf. Figure 5.2) l'ensemble $\mathcal{W}_\alpha$ des états qu'il met en jeu, pour le type d'agents α. C'est cet ensemble qu'il est nécessaire de réduire le plus possible pour simplifier le calcul des états stationnaires de la chaîne (Pardoux, 2007 ; Marek et Szyld, 2007) et ainsi le calcul des propriétés de niveau macro sur l'organisation.

Cette définition correspond formellement à l'énoncé d'une application $\mathcal{T}$ telle que :

$$\mathcal{T}: \big(\mathcal{Y}_a(t), \mathcal{E}_a(t)\big) \longmapsto w_a(t)$$

Cette application permet de construire la chaîne de Markov à partir du principe suivant. La probabilité de transition entre un état $w_1 \in \mathcal{W}_\alpha$ et un état $w_2 \in \mathcal{W}_\alpha$ est égale à la moyenne sur l'ensemble des états (y, e) tels que $\mathcal{T}(y, e) = w_1$, pondérée par la probabilité d'occurrence de l'état (y, e) dans cet ensemble, de la somme des probabilités de transition entre l'état (y, e) et les états (y', e') tels que $\mathcal{T}(y', e') = w_2$. Soit, avec $x(t) = \big(\mathcal{Y}_a(t), \mathcal{E}_a(t)\big)$:

$$\begin{aligned}
&\mathbb{P}\big(w_2(t + \Delta t) \mid w_1(t)\big) \\
&\quad = \sum_{(y,e) \in \mathcal{T}^{-1}(w_1)} \left(\frac{\mathbb{P}\big(x(t) = (y, e) \mid x(t) \in \mathcal{T}^{-1}(w_1)\big)}{\sum_{x(t) \in \mathcal{T}^{-1}(w_1)} \mathbb{P}\big(x(t)\big)} \right. \\
&\quad \left. \times \sum_{\mathcal{T}(y',e')=w_2} \mathbb{P}\big(x(t + \Delta t) = (y', e') \mid x(t) = (y, e)\big) \right)
\end{aligned} \tag{1}$$

Il nous faut d'abord considérer l'ensemble des éléments mis en jeu par les mécanismes organisationnels – c'est-à-dire qui peuvent influer sur le résultat de ces mécanismes – et qui doivent donc être pris en compte dans la définition des états de la chaîne de Markov :

- l'état actuel $\xi_a(t)$ de l'agent a considéré,
- l'état du champ relationnel $R_a(t)$ de l'agent,
- l'état du champ d'action $P_a(t)$ de l'agent,
- l'état de l'extension sociale $D_a(t)$ de l'agent (uniquement pour les mécanismes fonctionnels).

Bien qu'il faille distinguer les méthodes de simplification selon les types de mécanismes considérés, une première simplification sera à effectuer quel que soit le type de mécanisme : **la réduction aux types**. Il s'agit de simplifier l'expression de l'état du

champ relationnel et du champ d'action de l'agent en réduisant les éléments considérés à leur type.

Nous l'effectuons en premier, car elle conditionne l'expression des états des processus stochastiques modélisés, et doit donc être préalable aux autres simplifications qui peuvent être effectuées sur ces ensembles d'états (c'est-à-dire qu'aucune simplification ne peut être réalisée sur les relations avec un agent en particulier, mais seulement sur des relations avec des agents « génériques », réduits à leur type).

Ainsi, le champ relationnel d'un agent est exprimé comme un ensemble de paires : type de la relation, type de l'agent cible.

Le champ d'action de l'agent est lui exprimé comme un ensemble de : type de la tâche, ensemble des types des bénéficiaires de la tâche. Les types des bénéficiaires sont, eux, inclus dans l'ensemble comprenant l'ensemble des types d'agents et le type spécial σ désignant l'agent étudié lui-même.

Formellement, on a donc égalité des états du modèle markovien :

$$\mathcal{T}\left(\left(\xi_a(t), R_a(t), P_a(t), D_a(t)\right)\right) = \mathcal{T}\left(\left(\xi'_a(t), R'_a(t), P'_a(t), D'_a(t)\right)\right)$$

si et seulement si :

- $\xi_a(t) = \xi'_a(t)$ (les états de l'agent sont les mêmes) ;
- $R_a(t) \sim R'_a(t)$, soit $\forall \alpha \in T_A, |\{b \in R_a(t) \mid \mathrm{t}(b) = \alpha\}| = |\{b \in R'_a(t), \mid \mathrm{t}(b) = \alpha\}|$ (les champs relationnels sont équivalents, *i.e.* contiennent, pour chaque type d'agents, un nombre d'agents identiques) ;
- $P_a(t) \sim P'_a(t)$, soit $\exists \mathfrak{B}: \widehat{B_a} \leftrightarrow \widehat{B'_a} \mid \mathfrak{B}(B) = B' \Rightarrow B \sim B' \wedge (\forall \phi \in T_P, |\{x = (B, j, \phi) \in D_a(t), j \in \mathbb{N}\}| = |\{x = (B', j, \phi) \in D'_a(t), j \in \mathbb{N}\}|$ avec $\widehat{B_a} = \{B \subseteq A(t) \cup \sigma \mid \exists x = (B, j, \phi) \in P_a(t), j \in \mathbb{N}, \phi \in T_P\}$ (les champs d'action sont équivalents, *i.e.* il existe une bijection entre les sous-ensembles des champs d'actions définis par l'équivalence des bénéficiaires des tâches telle que cette bijection conserve le nombre de tâches de chaque type entre les ensembles qu'elle associe) ;
- $D_a(t) \sim D'_a(t)$, soit $\exists \mathfrak{B}: A(t) \leftrightarrow A(t) \mid (\mathfrak{B}(b) = b' \Rightarrow \{x \mid (x, b) \in D_a(t)\} \sim \{x \mid (x, b') \in D'_a(t)\})$ (les extensions sociales sont équivalentes, *i.e.* les ensembles de tâches associés à un même agent sont équivalents au sens de l'équivalence des champs d'action).

Il est nécessaire de procéder ainsi car il s'agit avec cette méthode d'analyser une organisation localement sans réellement spécifier le système en entier, puis d'agréger les connaissances locales en propriétés globales. Ainsi, l'état du système n'est connu que

localement – agent, champ relationnel, champ d'action, extension sociale – et non globalement : les autres agents du système ainsi que les relations et les tâches de ces agents sont inconnus. Il n'est donc possible de procéder à l'analyse qu'à travers une vision locale du système, sans présupposer d'autres connaissances sur les autres agents, mis à part l'agent étudié, que leur type.

Ensuite, il nous faut faire une distinction importante : si les mécanismes structuraux et les mécanismes fonctionnels projectifs – *i.e.* qui prennent en compte l'état précédent de l'extension sociale (cf. section 4.2.2) – sont itératifs, et donc modélisés sous forme de chaîne de Markov, les mécanismes fonctionnels négligents – *i.e.* non projectifs (cf. section 4.2.2) –, quant à eux, ne sont pas itératifs : ils sont de simples fonctions appliquées à l'état du système à l'instant t sans prise en compte de l'état antérieur ; ainsi, ils n'ont pas besoin d'être exprimés sous une forme autre. Ils nécessitent seulement d'être transformés selon les égalités de probabilités exprimées ci-dessus pour prendre comme espace de départ l'espace d'état $\mathcal{W}_\alpha$ des chaînes de Markov des mécanismes itératifs du type d'agents étudié.

5.3.2 Méthodes de description selon les types de mécanismes

Au vu de la complexité potentielle des mécanismes organisationnels, il nous est nécessaire de différencier les méthodes de description selon le type de mécanisme considéré. Nous nous référons donc aux types que nous avons utilisés (section 4.2) pour décrire la complexité des espaces d'états des mécanismes organisationnels. Nous considérons donc ici uniquement les mécanismes itératifs, qui peuvent ainsi être décrits comme des chaînes de Markov : mécanismes structuraux et mécanismes fonctionnels projectifs.

Le cas le plus simple est celui des **mécanismes libres**. Leur espace d'états initial est limité au champ relationnel à l'instant précédent (pour les mécanismes structuraux) ou au champ relationnel à l'instant donné et à l'extension sociale à l'instant précédent (pour les mécanismes fonctionnels). Une fois la réduction aux types effectuée (cf. section 5.3.1), le mécanisme est déjà dans une forme correspondant à une modélisation par chaîne de Markov : processus discret sur un ensemble fini. Cela est vrai à la condition que le cardinal du champ relationnel des agents soit **borné** ; l'ensemble des types de tâches étant fini, il suffit donc que le nombre d'agents dans tout voisinage soit borné. Cela s'obtient grâce à la condition suffisante que le nombre total d'agents du système soit lui-même borné, ce qui, en condition réelle, est réaliste. Nous supposons donc cette condition vérifiée dans toutes les modélisations effectuées. Le mécanisme considéré prend alors généralement la forme d'une chaîne de Markov dont les probabilités de transition sont binaires – car les transitions sont

définies de manière exactes. Seule la présence de transitions non spécifiques à des types[45] peut introduire des probabilités non binaires ; celles-ci requièrent alors des informations extérieures au mécanisme – telles que la proportion des types d'agents dans le système – qui peuvent, lors de la modélisation, être laissées sous forme d'inconnues ou être spécifiées pour l'étude d'un système particulier ; nous les appelons **valeurs extérieures**.

Le cas des **mécanismes homogènes** est assez similaire à celui des mécanismes libres. En effet, après réduction aux types, l'ensemble des tâches possibles pour un agent est, comme l'ensemble de ses champs relationnels possibles, un ensemble fini. Ainsi, l'ensemble d'états du modèle markovien de ces mécanismes se réduit aisément au produit cartésien de ces deux ensembles[46], d'autant qu'il est possible de limiter l'ensemble des champs relationnels à la considération des tâches réellement impliquées dans les mécanismes, en négligeant l'ensemble des tâches non impliquées. Les probabilités de transition entre les états sont alors le produit de deux valeurs : la probabilité de transition entre les états, définie comme pour les mécanismes libres, et la probabilité de transition entre les différents champs d'actions pour chaque agent, qui est une valeur extérieure. Pour ces mécanismes comme pour les mécanismes libres, une simplification de l'expression de la chaîne de Markov par regroupement d'états sémantiquement similaires est possible bien que non nécessaire[47].

Enfin, les plus compliqués des mécanismes sont les **mécanismes hétérogènes**. Comme nous avons vu que la distinction entre mécanismes réactifs et téléonomiques (les mécanismes libres étant réactifs et les mécanismes homogènes non libres étant donc téléonomiques) n'était pas significative dans la complexité de la construction du modèle markovien, nous ne ferons pas la distinction entre les deux dans le cadre de l'étude de la modélisation des mécanismes hétérogènes. La complexité de cette modélisation provient du fait que l'ensemble ξ des états de l'agent peut potentiellement prendre une infinité de valeurs, ou tout du moins un très grand nombre de valeurs, et qu'ainsi, la définition des transitions pour chacune de ces valeurs et le calcul à partir de la chaîne de Markov ainsi engendrée sont des opérations très coûteuses. L'objectif est donc d'être capables de regrouper les informations contenues à la fois dans l'état de l'agent, dans son champ d'action et dans son

---

[45] Un mécanisme peut par exemple faire entrer un nouvel agent dans le voisinage sans que le type de cet agent soit nécessairement spécifié.

[46] Champs relationnels et champs d'action pour les mécanismes structuraux ; champs relationnels, champs d'action et extension sociale pour les mécanismes fonctionnels.

[47] La méthode de regroupement des états étant similaire à celle décrite ci-après pour les mécanismes hétérogènes.

champ relationnel – et dans son extension sociale pour les mécanismes fonctionnels – en un ensemble d'états, fini et de cardinal le plus petit possible, dont les états soient :

- sémantiquement signifiants : un état doit pouvoir être décrit et compris ;
- significatifs par rapport au mécanisme organisationnel : le déroulement du mécanisme doit pouvoir être prévu à partir de l'état ;
- cohérents : une situation précise d'un agent ne doit pouvoir être représentée que par un seul état du modèle[48].

Bien que de nombreux moyens puissent permettre d'arriver à ce résultat, la plupart consistent à décrire les éléments significatifs des états et de l'organisation locale de l'agent qui conservent une signification après réduction aux types. Il s'agit ensuite d'agréger les éléments ainsi trouvés pour former un espace d'états multidimensionnel.

Enfin, lorsqu'il est nécessaire d'agréger des états présents en états plus compacts, en vue de calcul, il est possible de regrouper des états entre eux. L'expression des transitions se fait alors de deux manières, en suivant l'égalité posée précédemment (cf. équation 1, p. 119). Les transitions vers les états regroupés sont simplement la somme de chaque transition vers les états qui ont été groupés. En revanche, les transitions à partir des états regroupés nécessitent la connaissance des probabilités d'occurrence de chacun des états qui ont été groupés. Puisque celles-ci sont, *a priori*, inconnues, il est généralement nécessaire d'utiliser une méthode itérative – dérivée des méthodes d'approximation sur les modèles bayésiens (Rosenthal, 1994 ; Jordan *et al.*, 1999) – pour atteindre ces valeurs, c'est-à-dire l'ensemble des probabilités d'occurrence $\mathbb{P}\big(x(t)\big)$ des états $x(t) = \big(\mathcal{Y}_a(t), \mathcal{E}_a(t)\big)$.

Ces méthodes (cf. Tableau 5-A) consistent à prendre une distribution uniforme pour les états groupés, à calculer la chaîne de Markov, puis la distribution stationnaire liée à celle-ci ; il s'agit ensuite d'utiliser les informations de répartition moyenne des états groupés pour calculer, de manière exacte, les transitions vers chacun des états originaux, puis d'utiliser les probabilités d'arriver dans chacun de ces états pour reconstruire itérativement, une nouvelle chaîne de Markov. Après un certain nombre d'itérations – différent selon la méthode exacte choisie – on arrête les itérations, et on utilise les valeurs, plus ou moins précises selon le nombre d'itérations effectuées, comme répartition définitive des états originaux dans les états groupés. Les transitions à partir de ces états sont donc la moyenne des transitions à partir des états originaux qu'ils contiennent, pondérée par la probabilité d'occurrence de ces états originaux relativement à l'état groupé.

---

[48] Autrement dit, les états du modèle doivent former une partition des états réels de l'agent et de son organisation locale.

Il est possible d'appliquer le même algorithme, dans le cas où les probabilités d'occurrences de sous-états des états $x \in \mathcal{T}^{-1}(w)$ sont connues, en supposant la distribution uniforme sur les sous-états inconnus et en calculant, par multiplication des probabilités de tous les sous-états, la distribution initiale supposée des états x.

Tableau 5-A - Algorithme itératif d'estimation des distributions d'états

1 Supposer une distribution uniforme pour les états x groupés en un même état w de la chaîne de Markov.

$$\forall w \in \mathcal{W}_\alpha, \forall x \in \mathcal{T}^{-1}(w), \mathbb{P}(x \mid w) = \frac{1}{|\mathcal{T}^{-1}(w)|}$$

2 Calculer les probabilités de transition de la chaîne de Markov.

(cf. équation 1, p. 119)

3 Calculer l'état stationnaire $\overrightarrow{w_\alpha^S}$ de la chaîne de Markov, c'est-à-dire le vecteur propre de la matrice de transition associé à la valeur propre 1 et de norme 1 égale à 1.

4 ↻ Répéter les étapes 1 à 4 pour les mécanismes totaux de tous les types d'agents du système.

5 Déterminer, par des méthodes *ad hoc*, les probabilités des états x pour chaque type d'agent à partir des informations moyennes obtenues par le calcul des distributions stationnaires.

6 ↻ Répéter les étapes 1 à 5 en remplaçant les distributions stationnaires par les probabilités moyennes calculées à l'étape 5.

5.3.3 Agrégation de mécanismes et rétroactions

Bien qu'il soit possible de modéliser directement les mécanismes totaux utilisés par les agents, il est souvent plus intéressant de modéliser les mécanismes partiels, souvent partagés par plusieurs types d'agents, puis de les agréger pour chaque type d'agent séparément. Il faut pour cela être capable de modéliser plusieurs types d'interactions entre mécanismes :

- la composition : deux mécanismes appliqués successivement à un état (cf. section 4.2) ;
- la mise en parallèle : deux mécanismes appliqués en même temps à un même état et dont les modifications sont cumulées (cf. sections 4.2.1 et 4.2.2) ;
- le partage d'informations organisationnelles entre agents : plusieurs agents échangeant des informations organisationnelles dépendant de leurs mécanismes et

réutilisées par les mécanismes des autres.

Tout d'abord, la composition de mécanismes se modélise particulièrement facilement si les états des modèles markoviens de ces deux mécanismes sont les mêmes. Dans ce cas, il s'agit d'effectuer la simple multiplication matricielle des matrices de transition des deux mécanismes. Si les états d'un des mécanismes sont des sous-états de ceux de l'autre mécanisme, il suffit d'étendre la matrice du second en dupliquant les états les plus généraux pour pouvoir effectuer la multiplication. En revanche, si les mécanismes n'ont pas les mêmes états, il est nécessaire de prendre le produit cartésien des états mis en jeu par les mécanismes, et de dupliquer les transitions existantes, afin d'obtenir un espace d'états commun dans lequel les transitions des deux mécanismes aient un sens.

Pour modéliser la mise en parallèle de deux mécanismes m_1 et m_2, il suffit de disposer d'un opérateur $\oplus$ modélisant le cumul des modifications sur les états. Avec cet opérateur $\oplus$ à disposition, l'opération s'effectue aisément si les ensembles d'états $\mathcal{W}_a$ des mécanismes sont identiques et clos par rapport à $\oplus$. Sinon, il est nécessaire d'étendre les ensembles d'états afin d'obtenir ces propriétés. La probabilité de transition d'un état e_o à un état e_d est alors la somme des produits des probabilités de transitions de e_o vers les états e_1 et e_2 respectivement pour les mécanismes 1 et 2, pour tous les états e_1 et e_2 tels que le cumul des modifications apportées par chacun des mécanismes donne l'état final e_d, c'est-à-dire[49] :

$$\mathbb{P}_{m_1+m_2}\big(e_d(t+\Delta t) \mid e_0(t)\big) = \sum_{(e_1-e_o)\oplus(e_2-e_o)=(e_d-e_o)} \Big(\mathbb{P}_{m_1}\big(e_1(t+\Delta t) \mid e_0(t)\big) \times \mathbb{P}_{m_2}\big(e_2(t+\Delta t) \mid e_0(t)\big)\Big)$$

Enfin, lorsqu'il y a échange d'informations organisationnelles entre différents agents du système, et que ces informations sont réutilisées au sein des mécanismes organisationnels – par exemple, dans le mécanisme de relaxation restreinte (Camps et Gleizes, 1995) –, un autre problème se pose lors de la modélisation : l'évaluation des probabilités liées à ces informations. Deux cas se présentent alors :

- soit les informations proviennent d'un autre type d'agents, qui, lui, n'utilise pas d'informations fournies par le type étudié, auquel cas, il suffit de modéliser d'abord le second type d'agents, et d'utiliser les informations moyennes sur son état pour en déduire les probabilités des informations fournies ;
- soit les informations proviennent d'un autre agent du même type, ou bien d'un agent d'un autre type, mais qui utilise également des informations fournies par le type

[49] L'opérateur - faisant référence à l'ensemble des modifications permettant de passer d'un état à un autre.

d'agent étudié ; dans ce cas, il est nécessaire d'utiliser également des méthodes itératives (cf. section 5.3.2) pour partir d'une répartition uniforme des probabilités des informations fournies et itérer sur les types d'agents étudiés pour aller vers de plus en plus de précision.

Enfin, un dernier cas d'agrégation d'informations sur les mécanismes est à prendre en compte : la composition de mécanismes fonctionnels négligents avec des mécanismes itératifs. Cette composition s'effectue bien sûr *a posteriori* du calcul des états stationnaires, puisqu'il s'agit de l'application d'une fonction à des ensembles d'états. En revanche elle nécessite que l'espace de départ de la fonction soit équivalent à l'espace d'état des mécanismes itératifs. Or, cette propriété est facilement obtenue en agrégeant, là encore, si besoin, les états multiples de la fonction en états groupés dont les valeurs sont les moyennes des sous-états, ou alors en répartissant les états des mécanismes itératifs comme sous-états d'états de départ de la fonction, si ces derniers sont moins spécifiques.

Ainsi, à travers un ensemble d'outils : réduction aux types, limitation des cardinaux, estimation des valeurs extérieures, agrégations d'états et méthodes itératives, nous sommes à même de proposer une méthode permettant de modéliser l'ensemble des mécanismes itératifs d'un système comme des chaînes de Markov, et de tirer de cette modélisation des informations sur l'état moyen du système correspondant. La pertinence de cette modélisation est limitée par l'adéquation des mécanismes avec une représentation markovienne, ce qui est conforme dans le cas des mécanismes homogènes, mais peut ne pas l'être pour des mécanismes hétérogènes. Pour cela, nous introduisons dans la suite des mesures de l'écart entre la représentation sous forme de chaînes de Markov et le fonctionnement réel des mécanismes. Dans le cas où cet écart est trop important, il sera alors préférable de s'orienter vers d'autres moyens de représentation, tels que ceux présentés précédemment : modélisation par automates cellulaires parallèles ou en logique floue (cf. section 5.2.1).

5.4 Calcul de propriétés dynamiques

A partir des modélisations sous forme markovienne, il est ensuite possible de déduire des propriétés globales de l'organisation. C'est en effet à travers le calcul de l'état stationnaire des chaînes de Markov que l'information locale se transforme en une information utilisable de manière globale, car elle définit un état statistiquement moyen pour tous les agents d'un certain type. Cette information peut être utilisée de trois manières différentes :

- la distribution stationnaire, elle-même, fournit des informations sur l'état moyen du champ relationnel et donc du voisinage de chaque type d'agents ; agrégée selon les

types d'agents, elle permet d'obtenir des informations sur la topologie du réseau, ainsi que sur les différentes topologies locales ;

- par comparaison avec la matrice de transition, et à travers la sémantique des états considérés, la distribution stationnaire permet aussi de connaître la dynamique moyenne locale pour chaque type d'agent : des informations sur l'évolution des champs relationnels, et donc plus globalement la dynamique, l'évolution et même la stabilité du réseau formé par les agents peuvent être étudiées ;
- enfin, par application des mécanismes fonctionnels – en particulier des mécanismes non-itératifs – il est possible de déterminer des propriétés plus particulières des systèmes formés, telles que les propriétés fonctionnelles liées à l'attribution de tâches, leurs probabilités de succès, et la fréquence à laquelle elles sont réassignées ; cette étude peut se faire par la prise en compte de valeurs extérieures supposées ou réelles, et leur utilisation comme hypothèse dans l'étude de cas spécifiques des organisations considérées.

5.4.1 Définition et intérêt des propriétés globales

Ainsi, différents types de propriétés globales peuvent être exprimées à partir de telles modélisations : propriétés structurelles, propriétés dynamiques, propriétés fonctionnelles. Il est évident que les propriétés recherchées et leur expression dépendent des systèmes considérés et des objectifs de l'étude, cependant, elles suivent un même principe : l'utilisation de la distribution stationnaire des états de chaque type d'agents permet l'obtention d'une information globale moyenne sur l'organisation ; combinée à des valeurs extérieures, telles que la population d'agents moyenne présente dans le système, les objectifs fixés pour celui-ci, sa fonction attendue, etc., elle permet d'évaluer différents aspects de l'organisation, pour des cas généraux comme pour des cas particuliers.

Information moyenne statique

Les informations fournies par l'ensemble des distributions stationnaires $\overrightarrow{w_\alpha^S}$ permettent par exemple de connaître la connectivité (West, 2001) moyenne des agents de chaque type, ou encore leur connectivité vis-à-vis de sous-réseaux constitués seulement de certains types d'agents. Il est ainsi déjà possible, en combinant ces informations à une hypothèse sur la population moyenne du système, d'évaluer la répartition des connectivités et donc la topologie du graphe constitué par les agents, ou de certains de ses sous-graphes.

Pour connaître la connectivité moyenne de l'agent, il est nécessaire de connaître le nombre $\left|V_\alpha^\beta\right|$ d'agents de chaque type $\beta \in T_A$ contenus dans le voisinage de l'agent, et cela

pour tout état $w \in \mathcal{W}_\alpha$ – il est évidemment possible de ne s'intéresser qu'aux types pour lesquels cette information est connue, si elle ne l'est pas pour tous, et l'information peut être uniquement une moyenne et non une valeur exacte. Par construction du vecteur $\overrightarrow{V_\alpha^\beta}$ constitué de ces cardinaux pour chaque type β, on obtient facilement la connectivité moyenne pour chacun de ces types par le produit scalaire de ces deux vecteurs. Ainsi, la connectivité des agents de type α vis-à-vis des agents de type β est :

$$c_\alpha^\beta = \left(\overrightarrow{w_\alpha^S} \middle| \overrightarrow{V_\alpha^\beta}\right)$$

De cette façon, il est possible de construire un réseau moyen, selon les proportions d'agents présents dans le système pour chaque type d'agents et les connectivités de chacun des types d'agents vis-à-vis de chaque type. Par ailleurs, il est possible de calculer des propriétés de connectivités plus générales, pour des ensembles de types $h \subseteq T_A$:

$$c_\alpha^h = \sum_{\beta \in h} c_\alpha^\beta$$

Information moyenne dynamique

Par suite, en considérant à la fois la distribution stationnaire $\overrightarrow{w_\alpha^S}$ et la matrice de transition, c'est-à-dire l'ensemble des probabilités $\mathbb{P}\big(w_2(t + \Delta t) \mid w_1(t)\big)$ pour tous $w_1, w_2 \in \mathcal{W}_\alpha$, il est possible d'obtenir des informations sur la dynamique structurelle du système : en effet, la matrice de transition décrivant les probabilités de réorganisation à partir de chaque état, et la distribution stationnaire décrivant la probabilité de chaque état, on peut donc en déduire les différentes probabilités globales de réorganisation – bien qu'il faille prendre en compte la signification donnée aux états lors de la modélisation.

Par exemple, si l'ensemble des états est égal à l'ensemble des états du voisinage (mécanismes structuraux libres, par exemple), alors la multiplication de la distribution stationnaire par la matrice de transition dont la diagonale a été annulée donne un vecteur indiquant les probabilités de réorganisation à partir de chaque état :

$$\overrightarrow{V_\alpha^R} = \overrightarrow{w_\alpha^S} \times \begin{pmatrix} 0 & m_{12} & \cdots & m_{1p} \\ m_{21} & 0 & \cdots & m_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & 0 \end{pmatrix}$$

La moyenne des composantes de ce vecteur donne la probabilité totale de réorganisation pour un type d'agent à chaque pas de temps, que nous nommons, par analogie, la **température en α** :

$$\mathbb{T}_\alpha = \frac{1}{\dim\left(\overrightarrow{V_\alpha^R}\right)} \left\| \overrightarrow{V_\alpha^R} \right\|_1$$

Une telle information, extraite pour tous les types d'agents, donne une idée précise de la stabilité du réseau d'agents et de la mobilité de chaque type d'agents à l'intérieur de celui-ci. Si l'ensemble des états n'est pas égal à l'ensemble des états du voisinage, il convient d'annuler l'ensemble des composantes de la matrice de transition correspondant à des changements entre états n'impliquant pas de réorganisation du champ relationnel de l'agent.

Ainsi, la **température** du système, dans sa totalité est égale à la moyenne des températures en α pour chaque type $\alpha \in T_A$, pondérée par la proportion d'agents de ce type dans le système :

$$\mathbb{T}(t) = \sum_{\alpha \in T_A} \mathbb{T}_\alpha \times \mathbb{P}\big((a = (n_a, \alpha), n_a \in \mathbb{N}) \mid a \in \mathrm{A(t)} \big)$$

De la même manière, il est possible de définir diverses propriétés liées à la réorganisation dynamique du système à partir des informations fournies par la matrice de transition et la distribution stationnaire. Il est notamment possible de définir un ensemble d'opérations permettant de trouver des informations de réorganisation à partir de l'ensemble d'états $\mathcal{W}_\alpha$ en définissant un vecteur de réorganisation $\overrightarrow{t_\alpha}$ pour lequel la fonction s'écrirait :

$$\mathbb{T}_\alpha = \left(\overrightarrow{V_\alpha^R} \middle| \overrightarrow{t_\alpha} \right)$$

Dans le cas de la température, cela correspond à définir, en posant $d = dim\left(\overrightarrow{V_\alpha^R}\right)$:

$$\overrightarrow{t_\alpha} = \begin{pmatrix} d^{-1} \\ \vdots \\ d^{-1} \end{pmatrix}$$

Information moyenne fonctionnelle

Enfin, l'application des mécanismes fonctionnels – ou plutôt de leur modèle – à la distribution stationnaire donne une information conséquente sur la répartition moyenne de la fonction dans le système. Elle permet de mettre à jour les motifs de délégation et de répartition des tâches dans le système, et, cumulée à des valeurs extérieures comme la probabilité d'acceptation par les agents des tâches attribuées, permet d'évaluer l'efficacité moyenne de la répartition des tâches et ainsi de proposer une évaluation *a priori* de l'intégrité fonctionnelle du système dans le temps – *i.e.* la capacité de celui-ci à assurer sa fonction, c'est-à-dire à progresser vers son objectif, en tout instant (Kozlak, 2000).

L'application des mécanismes fonctionnels consiste à calculer les vecteurs $\overrightarrow{D_{\alpha,w}}$ des probabilités d'états de l'extension sociale D_a d'un agent pour chacun des états $w \in \mathcal{W}_a$, à travers la modélisation $\mathcal{M}(F)$ du mécanisme fonctionnel total pour le type d'agents α. Il s'agit d'une simple application de fonction pour les mécanismes fonctionnels négligents, mais d'une opération de calcul de distribution stationnaire de chaîne de Markov pour les mécanismes fonctionnels projectifs. A partir de la donnée de ces vecteurs, il est déjà possible

d'exprimer des propriétés dynamiques sur la dimension fonctionnelle de l'organisation, en étudiant notamment, l'évolution des composantes de ces vecteurs, c'est-à-dire des probabilités d'états de D_a, en fonction de la dynamique structurelle étudiée précédemment.

Puis, il est possible d'effectuer la multiplication de la distribution stationnaire par la matrice formée par les vecteurs calculés pour l'ensemble des états w :

$$\overrightarrow{D_\alpha} = \overrightarrow{w_\alpha^S} \times \begin{pmatrix} \overrightarrow{D_{\alpha,w_1}} \\ \vdots \\ \overrightarrow{D_{\alpha,w_N}} \end{pmatrix}$$

Ce **vecteur fonctionnel** définit la probabilité moyenne d'occurrence de chaque état de l'extension sociale d'un agent de type α. Il est, à partir de cela, possible de définir des propriétés moyennes sur la fonction. Par exemple, si on dispose d'un sous-ensemble d'états de D_a correspondant à des états non souhaités, il suffit de faire l'addition des probabilités de ces états – c'est-à-dire des composantes du vecteur $\overrightarrow{D_\alpha}$ qui leur correspondent – pour connaître la probabilité moyenne qu'un agent de type α se retrouve dans une situation non souhaitée.

5.4.2 Validité du calcul des propriétés globales

Toutes ces propriétés globales doivent être évaluées selon deux critères supplémentaires, pour être comprises correctement lors d'une analyse d'organisation : leur **précision** et leur **spécificité**.

Nous définissons tout d'abord ces deux notions pour les modélisations des mécanismes organisationnels, puis nous étendons ces définitions, par agrégation, aux propriétés calculées à partir des données fournies par la modélisation des mécanismes.

La précision d'une modélisation d'un mécanisme est une mesure de son exactitude vis-à-vis des approximations effectuées dans le modèle pour parvenir au calcul de cette propriété. Elle dépend, dans notre méthode de l'exactitude des agrégations effectuées dans la modélisation du mécanisme.

Nous définissons la *précision* $\mathfrak{p}(\mathcal{M}, M, \alpha)$ comme la différence moyenne entre les probabilités markoviennes et les probabilités non markoviennes de transitions entre états ξ_a de l'agent a, normalisée par le nombre total d'états possibles considérés dans la modélisation $\mathcal{M}$ du mécanisme M pour le type d'agents α représenté par a (soit Ξ l'ensemble de ces états) :

$$\mathfrak{p}(\mathcal{M}, M, \alpha) =$$

$$\frac{1}{|\Xi|^2} \sum_{\xi_a, \xi_a' \in \Xi} \mathbb{E}\left[\left|\mathbb{P}\left(\xi_a(t+\Delta t) \mid \xi_a'(t)\right) - \mathbb{P}(\xi_a(t+\Delta t) \mid \xi_a'(t), \xi_a''(t-\Delta t), \dots)\right|\right]$$

La précision d'une propriété est ainsi définie comme la moyenne des précisions des mécanismes impliqués dans le calcul de cette propriété. Plus une propriété est précise, plus son espérance mesurée lors de simulations de systèmes instanciant l'organisation considérée sera proche de son espérance calculée par modélisation markovienne.

La *spécificité* d'une modélisation mesure, quant à elle, la dépendance de cette propriétés vis-à-vis de valeurs extérieures non spécifiques à l'organisation et donc dépendantes des instances organisationnelles étudiées. Elle évalue la nécessité de sur-spécifier l'organisation analysée pour obtenir une précision suffisante pour les propriétés dérivant de cette modélisation. Elle dépend donc de l'utilisation de valeurs extérieures dans le calcul de cette propriété.

> Nous définissons la spécificité $\mathfrak{s}(\mathcal{M}, M, \alpha)$ d'une modélisation $\mathcal{M}$ d'un mécanisme M pour un type d'agents α comme le nombre d'états possibles entrant en compte dans son calcul, mis à part les états engendrés par les mécanismes eux-mêmes, c'est-à-dire le nombre d'états possibles pour $\mathcal{Y}(t)$ (cf. section 5.3.1). Elle est exprimée en échelle logarithmique de base 2. Soit, avec $\mathcal{Y}(t) \in \mathcal{Y}$, *i.e.* $\mathcal{Y}$ l'ensemble des états possibles de $\mathcal{Y}(t)$:
>
> $$\mathfrak{s}(\mathcal{M}, M, \alpha) = \log_2|\mathcal{Y}|$$

La spécificité d'une propriété est, dès lors, définie comme la somme, pour chaque type d'agents, des spécificités des mécanismes utilisés dans son calcul. Plus une propriété est spécifique, plus la variance de cette propriété étudiée sur un grand nombre d'instances différentes de l'organisation considérée sera élevée.

L'ensemble de ces deux mesures permet de spécifier la précision et donc la validité, tout d'abord d'une modélisation organisationnelle par chaînes de Markov dans son ensemble, puis des propriétés qui sont calculées grâce à cette modélisation. Dans un contexte où la mesure de l'erreur et des variances est difficile, il est ainsi possible de disposer d'outils permettant l'expression de la pertinence d'une modélisation ; d'autres outils, liés au calcul sur les chaînes de Markov, pourraient être utilisés (Munos et Moore, 2009), mais semblent disposer de moins de sémantique vis-à-vis de la compréhension organisationnelle engendrée par la modélisation – car il s'agit d'outils plus généraux, ne produisant d'information que sur la chaîne de Markov elle-même.

5.4.3 Intérêt et utilisation de cette solution

Ainsi, nous proposons une méthode permettant d'exprimer, de manière probabiliste, des propriétés globales sur les organisations, sans étudier de manière spécifique des instances de ces organisations. Ces propriétés globales découlent de la modélisation des mécanismes

organisationnels itératifs sous la forme de chaînes de Markov et de l'utilisation du calcul de la distribution stationnaire comme méthode d'agrégation temporelle de l'information locale.

Par agrégation spatiale ensuite, sur les types d'agents, il est ainsi possible d'exprimer des propriétés globales pour l'organisation sur :

- la structure et la fonction moyennes de l'organisation ;
- la dynamique et la stabilité de cette organisation ;
- les cas particuliers de fonctionnement à travers la sur-spécification des propriétés dans l'analyse.

Cette méthode permet donc, en veillant à produire des modélisations des organisations qui soient sémantiquement équivalentes, d'obtenir des propriétés globales permettant non seulement de caractériser des organisations, mais également de comparer entre elles différentes organisations sur des critères précis, formalisés et objectifs, et cela *a priori* de toute instanciation de ces organisations. La notion de spécificité permet de mesure la capacité de modéliser les organisations en dehors de toute instanciation spécifique.

Cette méthode permet tout cela de manière exacte sur toutes les organisations impliquant des mécanismes organisationnels homogènes ou des mécanismes hétérogènes dotés de la propriété markovienne. Elle permet également cela, de manière approchée, sur les organisations impliquant des mécanismes hétérogènes non markoviens ; il est alors possible de définir l'acuité de la modélisation à travers la notion de précision.

Chapitre 6. Valeur méthodologique

Nous avons, dans les chapitres précédents, décrit un modèle organisationnel complet ainsi que les méthodes analytiques applicables pour obtenir des informations sur les organisations explicitées avec ce modèle. Ces considérations théoriques sont applicables à l'analyse de systèmes existants, mais également à la conception d'organisations pour les SMA. Nous montrons dans ce chapitre comment notre modèle s'intègre à la méthode VOYELLES et permet de donner aux concepteurs les moyens de spécifier de manière formelle et structurée les organisations des SMA, et comment cette spécification se positionne dans les étapes de la conception d'un SMA.

Nous insistons préalablement sur le fait que l'approche choisie pour la création de notre modèle, à savoir une approche émergentiste et calculatoire nous permet de garantir des propriétés d'expressivité de notre modèle (cf. section 3.1.4 et 3.2.4), notamment l'indépendance du modèle vis-à-vis des architectures internes possibles des agents et la calculabilité de propriétés sur les organisations. Les conséquences en sont que notre modèle peut être utilisé pour la conception de SMA :

- indépendamment du domaine d'application ;
- indépendamment de l'environnement et du langage de programmation ;
- avec la capacité de fournir des outils de validation du modèle avant l'implémentation.

Nous décrivons donc ci-après la manière dont le modèle organisationnel proposé permet d'instancier la méthode VOYELLES et d'utiliser de manière efficace les spécifications organisationnelles dans la conception d'un SMA.

6.1 Méthodologie SMA orientée organisations dynamiques

La méthode VOYELLES (Demazeau, 1996) propose une approche de la conception des SMA à travers 4 dimensions distinctes : Agent, Environnement, Interaction, Organisation. Chacune de ces dimensions peut être spécifiée indépendamment et de leur mise en commun provient le modèle complet du SMA conçu. Dans cette approche, notre modèle permet la formalisation de la dimension « Organisation ». La modélisation de cette dimension est une étape de la conception, parallèle à la spécification des autres dimensions ; elle permet de définir l'agencement et la fonction globale du système, mais aussi, nous l'avons dit, sa dynamique (cf. exemple d'application section 7.1.3).

Nous proposons une méthode linéaire pour la création des types d'éléments et des éléments d'une organisation: les Agents, les Relations et les Tâches. Nous proposons ensuite une méthode itérative pour la définition et la validation des spécifications d'une organisation, et en particulier pour la définition des mécanismes organisationnels et leur validation (cf. Tableau 6-A).

> *Tout au long de la description de cette méthode, nous proposons en illustration un exemple de conception d'organisation. Cet exemple est la description de la conception d'une organisation possible répondant au problème de l'extraction de minerai – décrit dans* (Joumaa *et al.*, 2009)*. La description succincte du problème est la suivante : étant donné un environnement physique réel, comprenant des gisements de minerai dans le sol, un système multi-agent, composé d'agents situés (des robots), a pour objectif de localiser les gisements de minerai dans l'environnement, d'en extraire le minerai et de transporter et stocker celui-ci dans un lieu prédéfini (la base). Le problème posé est la conception d'un tel SMA. Nous nous intéressons uniquement ici à la conception d'une organisation possible pour celui-ci.*

Tableau 6-A - Étapes de conception

Étape	Description	Ensembles spécifiés
1	Définition fonctionnelle du système	T_P
2	Définition des types d'agents	T_A
3	Définition des types de relations	T_R
4	Choix des agents du système	A, $A(t)$
5	Définition des relations à t=0 pour les agents connus	$R(0)$
6	Spécifications sur les tâches	$P_a(t),\ P(t)$
7 à 10	Spécifications des mécanismes organisationnels	$\mathcal{SM}(\alpha), \mathcal{GM}$

Spécification statique

Les étapes linéaires de la spécification statique de l'organisation d'un SMA que nous proposons sont :

1) Définition fonctionnelle du système

Cette étape permet d'énoncer des spécifications sur l'ensemble des types de tâches du système : T_P.

a. Définition des objectifs globaux du système

Il s'agit dans cette étape de définir un ensemble $P_1, P_2, \ldots, P_N$ d'objectifs correspondant aux attentes fonctionnelles sur l'ensemble du système, dans le cas où celles-ci existent – dans des cas de simulations ou de modèles émergents sans attente globale, ces objectifs n'ont pas besoins d'être définis.

> *L'ensemble des objectifs du système répondant au problème des agents transporteurs est réduit à un seul objectif :*
>
> P_1 *: collecter le minerai et le stocker à la base*

b. Décomposition des objectifs en arbre de tâches

Cette étape correspond à la décomposition des objectifs principaux en un arbre dont les feuilles sont les types de tâches – description des objectifs élémentaires. Ceci permet d'obtenir une décomposition utilisable par la suite dans la conception de l'organisation, car elle donne des objets (des types de tâches) qui sont des éléments de notre modèle organisationnel. Pour effectuer cette décomposition, nous proposons de suivre une méthode de décomposition, telle que celles énoncée dans (Souchon *et al.*, 2002), ou dans le modèle TÆMS (Decker, 1996)), qui permettent d'obtenir simplement la décomposition complète.

> *L'objectif global* P_1 *est décomposable en :*
>
> – *Repérer les gisements de minerai :* P_2
>
> ▪ *Explorer l'environnement :* ϕ_E
>
> ▪ *Signaler l'emplacement des gisements :* ϕ_S
>
> – *Collecter le minerai :* P_3
>
> ▪ *Rejoindre l'emplacement des gisements :* ϕ_R
>
> ▪ *Récupérer le minerai (forage) :* ϕ_F
>
> – *Transporter le minerai jusqu'à la base :* ϕ_T

c. Définition des tâches annexes des agents

Cette étape consiste à définir les tâches personnelles des agents. Il peut s'agir de tâches utiles pour le système ou les agents qui le composent, mais aussi de tâches parasites ou nocives qui peuvent être spécifiées négativement par la suite.

Des tâches personnelles existent principalement dans deux cas :

– lorsque les agents possèdent des besoins personnels (notamment lorsqu'il s'agit d'entités réelles et non virtuelles) ;

– lorsque les agents font partie d'un système ouvert ou sont associés à des utilisateurs, auxquels cas, ils ne sont pas seulement une partie fonctionnelle autonome d'un système, mais peuvent posséder des objectifs propres (non systémiques) liés à leur concepteur ou possesseur.

Dans l'exemple considéré, les agents doivent être des robots mobiles, situés dans un environnement. Nous devons donc définir au moins deux tâches personnelles, liée à l'autonomie de chaque robot :

– *veiller à son alimentation (maintenance d'une charge des batteries non nulle à tout instant) :* ϕ_A

– *se situer dans l'espace :* ϕ_L

d. Définition des rôles effectifs

La définition de rôles effectifs permet l'expression des ensembles d'objectifs globaux du système attribuables aux agents. Les regroupements peuvent être effectués selon des besoins fonctionnels – mêmes informations utilisées, mêmes capacités requises – ou selon des logiques d'exécution – tâches successives, tâches alternatives, etc. Les types de tâches concernés sont généralement les tâches systémiques (*i.e.* dérivées des objectifs globaux du système), mais il arrive que des tâches personnelles des agents gagnent à être exprimées comme rôles.

Les tâches systémiques peuvent être regroupées selon différentes modalités (cf. Figure 6.1), mais une proposition simple et cohérente du point de vue fonctionnel est :

– *rôle d'explorateur :* $q_E \equiv \{\phi_E, \phi_S\}$

– *rôle de foreur :* $q_F \equiv \{\phi_R, \phi_F\}$

– *rôle de transporteur :* $q_T \equiv \{\phi_R, \phi_T\}$

– *rôle de localisateur :* $q_L \equiv \{\phi_L\}$

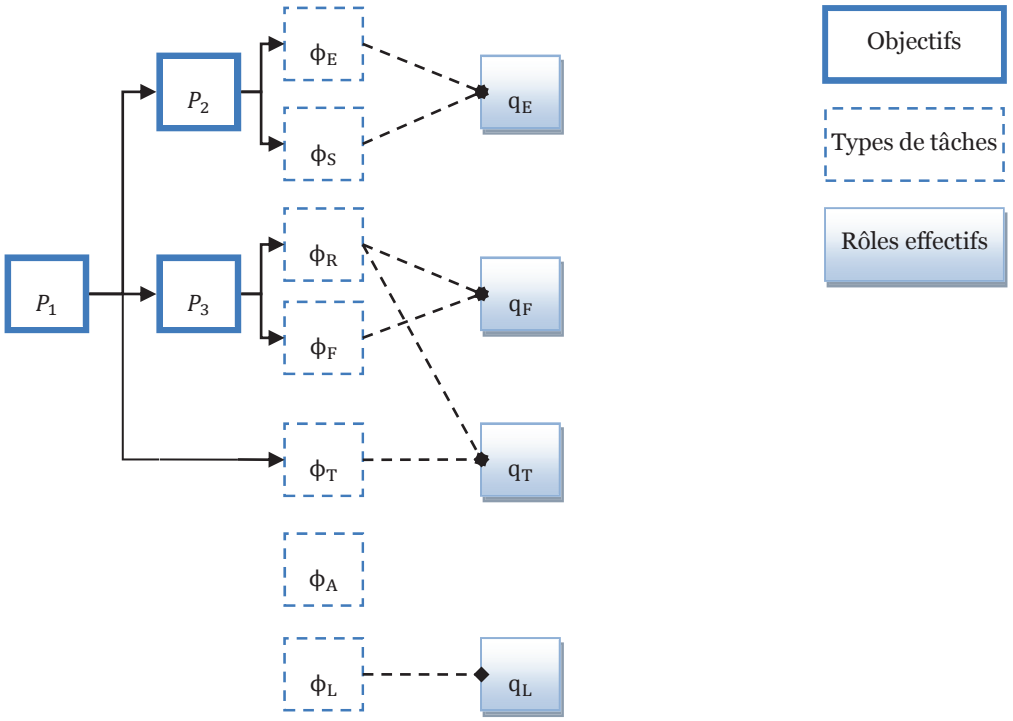


Figure 6.1 - Décomposition des objectifs en tâches et définition des rôles effectifs

e. Spécifications sur les types de tâches

Cette étape, optionnelle, consiste à énoncer d'autres types de spécifications sur l'ensemble T_p des types de tâches du système. Ceci est fait à partir des tâches définies aux étapes précédentes et implique la définition de spécifications qui ne soient pas constructives positives (cf. section 4.3.2).

Elle peut mettre, par exemple en jeu la définition d'alternatives entre l'existence, dans le système, de deux types de tâches, selon les conditions d'implémentation, ou encore des limitations cardinales sur le nombre de types de tâches existant dans le système.

> *Aucune spécification supplémentaire sur T_P n'est requise dans l'exemple considéré.*

2) Définition des types d'agents

Cette étape consiste à spécifier l'ensemble T_A des types d'agents du système.

a. Description des types d'agents et rôles normatifs souhaités

Le principe de construction utilisé dans cette étape est la définition de types d'agents considérés comme dotés des capacités ou des comportements adaptés pour répondre à chacun des rôles effectifs définis précédemment. Les rôles effectifs peuvent être associés un à un à des types d'agents ou regroupés de manière à former des ensembles cohérents pour des conditions d'exécution (plateformes, outils, etc.) spécifiées.

Il est possible, pour un système fermé, de définir directement des types d'agents. Pour un système ouvert, ou pour une plus grande flexibilité même dans des systèmes fermés, il est possible de définir plutôt des rôles normatifs (ensembles de types d'agent), non définis en extension.

> *Par soucis de simplicité et d'efficacité, nous proposons dans l'exemple considéré, une association bijective entre rôles effectifs et rôles normatifs. Nous utilisons des rôles normatifs plutôt que des types d'agents pour prendre en compte la possibilité que les agents dotés des mêmes capacités ne soient pas tous identiques – nouvelles versions d'un même robot, moyens de déplacements variés, par exemple.*
>
> *Les rôles normatifs ainsi définis sont :*
>
> – *explorateur* $[e]$ *: correspond au rôle effectif* q_E
>
> – *foreur* $[f]$ *: correspond au rôle effectif* q_F
>
> – *transporteur* $[t]$ *: correspond au rôle effectif* q_T
>
> – *base* $[b]$ *: correspond au rôle effectif* q_L
>
> *Nous spécifions cependant également que le rôle de base* $[b]$ *correspond à un unique type que l'on nomme agent-base* α_b *:*
>
> $$[b] = \{\alpha_b\}$$
>
> *Il est à noter que le système ainsi décrit n'est pas figé quant aux types et aux répartitions de rôles considérés ; il est modulaire, du fait même que les rôles soient des ensembles de types d'agents, et que ces ensembles ne soient pas spécifiés. Il est ainsi par exemple possible d'introduire, lors de la réalisation du système, un agent foreur-transporteur dont le type* α_{ft} *soit à la fois compris dans les rôles normatifs foreur et transporteur :* $\alpha_{ft} \in [f] \cap [t]$.
>
> *Pour spécifier que deux types normatifs sont incompatibles, par exemple les types base et transporteur, il faut ajouter la condition :*
>
> $$[b] \cap [t] = \emptyset$$

b. Description des autres types d'agents possibles

Cette étape équivaut à la description des types d'agents possibles ne correspondant pas à des entités prévues du système. Elle implique la spécification la plus complète possible de T_A.

Les types d'agents couramment définis dans cette étape sont les agents utilisateurs, les agents fraudeurs ou encore les entités actives extérieures au système, mais pouvant y entrer et interagir en son sein – par exemple, des fourmis d'une autre colonie dans l'exemple de la colonie de fourmis.

Il n'est pas nécessaire de définir d'autres types d'agents dans l'exemple considéré, sauf s'il existe la possibilité pour les agents de croiser, dans l'environnement, d'autres équipes de récupération de minerai, partenaires ou concurrentes. Il serait nécessaire, dans ce cas, de formaliser les interactions entre ces équipes au niveau organisationnel, et donc, par exemple, de définir pour les autres équipes des rôles normatifs équivalents aux rôles définis pour l'équipe considérée.

3) Définition des types de relations

Cette étape consiste à spécifier l'ensemble T_R des types de relations.

a. Définition des relations associées aux rôles normatifs

Dans cette étape, il s'agit de définir l'ensemble des types de relations nécessaires au bon fonctionnement du système. Pour cela, il est nécessaire de commencer par analyser les besoins d'échanges entre les rôles effectifs définis précédemment, puis de projeter ces besoins comme des dépendances entre les rôles normatifs instanciant ces rôles effectifs.

Cela permet de définir un ensemble de types de relations répondant aux besoins d'échanges entre agents pour assurer l'intégrité fonctionnelle du système – la capacité de celui-ci à progresser vers son objectif en tout instant (cf. section 5.4.1) (Kozlak, 2000).

Il est intéressant de définir en même temps, lorsque cela est approprié, les restrictions sur les types d'agents sources et cibles possible pour les relations d'un type déterminé[50].

Dans notre exemple, les types de relations que nous définissons (cf. Figure 6.2) permettent d'assurer les échanges d'informations nécessaires à la continuité des opérations exécutées par les différents types d'agents.

Ainsi, nous définissons comme relations :

- *relation de localisation λ_l (entre n'importe quel agent et une base $[b]$) : permet à l'agent de demander sa position à un agent capable de la déterminer ; sémantiquement, l'existence d'une relation de localisation implique que la base connaisse l'agent source et les informations qui lui sont associées ;*
- *relation de signalisation λ_s (entre les agents explorateurs $[e]$ et une base $[b]$) : permet à l'agent source de signaler la position d'un gisement de minerai détecté ;*
- *relation de proximité λ_p (entre agents explorateurs $[e]$) : permet aux agents de percevoir la position des autres agents et d'agir en conséquence (dans le but de*

[50] Ces relations prennent la forme de spécifications constructives négatives sur les réservoirs secondaires $R_\alpha^\beta = \{ r = (a, b, \lambda) \mid a = (n_a, \alpha) \wedge b = (n_b, \beta), n_a, n_b \in \mathbb{N} \}$.

permettre un contrôle sur les agents explorateurs afin qu'ils n'explorent pas les mêmes endroits à plusieurs, par exemple) ; une telle relation peut, de même, servir à définir des déplacements coordonnés (dimension Environnement) comme le vol en nuée, par exemple ;

– *relation d'information λ_i (entre une base $[b]$ et des agents foreurs) : permet aux agents bases d'informer les agents foreurs de l'existence d'un gisement de minerai ou de l'affectation d'un agent foreur à celui-ci ;*

– *relation d'affectation λ_a (entre un agent foreur $[f]$ et un agent transporteur $[t]$) : permet aux agents foreurs d'affecter un agent transporteur au transport du minerai qu'ils ont récolté, lorsque cela est nécessaire.*

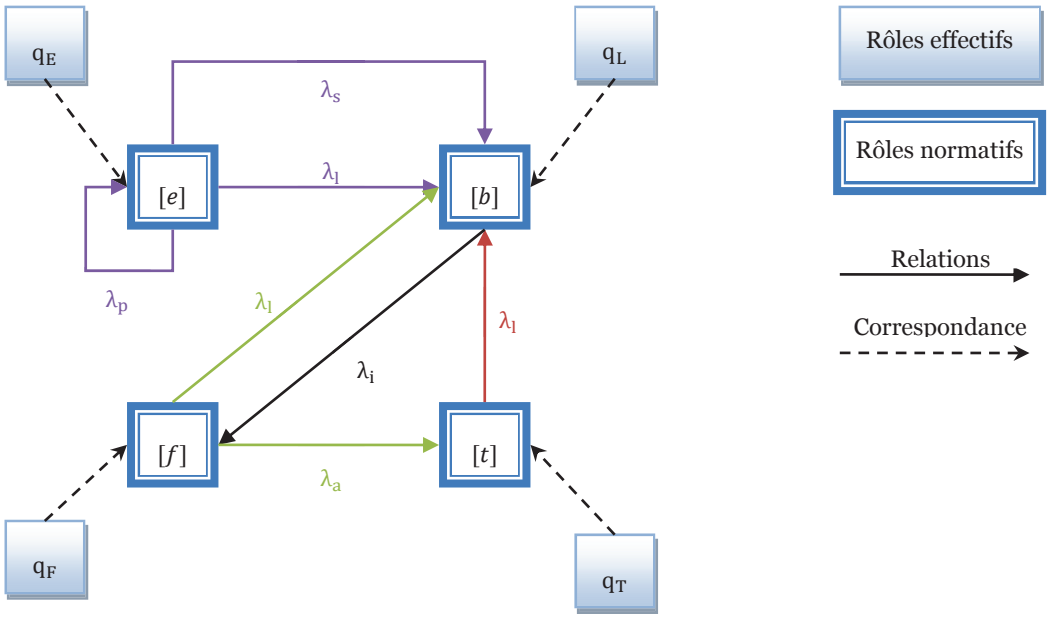


Figure 6.2 - Rôles normatifs et types des relations associées

b. Définition des autres relations possibles entre agents

Cette étape correspond à la définition d'un ensemble de types de relations supplémentaires, correspondant aux relations liées à des aspects non systémiques. Elle inclut la définition des relations conflictuelles, des relations liées aux objectifs personnels – accords, contrats, délégations, etc. – et des relations fondées en dehors de la conception du système – connaissances préalables des agents utilisateur, coalitions extérieures, etc.

Dans l'exemple décrit, les relations systémiques suffisent à la description du système, sauf, comme précédemment, dans le cas où des équipes concurrentes peuvent se

> *retrouver dans le même environnement, et où des relations conflictuelles entre les agents des différentes équipes pourraient être décrites.*

4) Choix des agents du système

Cette étape correspond à la définition, partielle dans le cas d'un système ouvert, de l'ensemble A et éventuellement des ensembles $A(t)$ des agents du système – resp. globalement et pour un instant $t \in \mathbb{R}^+$.

a. Description en extension des agents connus du système

Il s'agit, dans cette étape, de définir les agents du système connus à la conception. Elle concerne donc potentiellement tous les agents des systèmes fermés, et seulement un nombre restreint d'agents des systèmes ouverts ou semi-ouverts (cf. section 4.3.1). Lorsque cela est possible, il est intéressant de définir la période d'existence $L(a)$ de l'agent dans le système.

> *Dans le système concerné, nous savons qu'il est nécessaire que le minerai soit stocké à un endroit prédéfini ; pour cela, les agents doivent être capables de retrouver leur chemin vers ce lieu. Étant donnée la conception organisationnelle que nous avons suivie jusque là, il est nécessaire que les agents disposent en permanence d'un agent base capable de les localiser. Nous choisissons de définir cet agent $a_b \in \Gamma$ comme présent en permanence dans le système[51] :*
>
> $$L(a_b) = \mathbb{R}^+$$

b. Description des spécifications sur les agents

Il s'agit, dans cette partie, d'énoncer l'ensemble des spécifications restantes sur les ensembles d'agents du système, globalement, par types ou par rôles normatifs, en tout instant ou sur des intervalles temporels spécifiques. Il peut s'agit de spécifications fonctionnelles, pratiques, ou liées à l'implémentation du système.

Il est ainsi possible, notamment, de :

- limiter le nombre d'agents d'un certain type présents dans le système ;
- présenter des alternatives d'implémentation en laissant le choix entre plusieurs ensembles d'agents possibles pour implémenter les mêmes fonctions ;
- limiter les agents possibles pour un certain rôle normatif à un choix fini d'implémentations différentes ;

[51] La spécification est équivalente à $\forall t \in \mathbb{R}^+, a_b \in A(t)$.

– affirmer l'existence nécessaire d'un certain nombre d'agents ou de certains agents dans le système.

> *Dans l'exemple développé ici, il est nécessaire, pour garantir la fonctionnalité du système, d'affirmer l'existence d'un certain nombre d'agents (donc au moins un) pour chaque rôle normatif :*
>
> – $\forall t \in \mathbb{R}^+, \left|A_{[t]}(t)\right| \geq 1$
> – $\forall t \in \mathbb{R}^+, \left|A_{[e]}(t)\right| \geq 1$
> – $\forall t \in \mathbb{R}^+, \left|A_{[f]}(t)\right| \geq 1$
>
> *D'autres spécifications peuvent être ajoutées selon les choix de développement, par exemple l'unicité de la base ou encore la nécessité fonctionnelle de disposer de plus de transporteurs que de foreurs (pour assurer le fait que les foreurs puissent optimiser leur temps en minage sans être dans le besoin d'attendre des transporteurs pour les décharger) :*
>
> – $\forall t \in \mathbb{R}^+, \left|A_{[b]}(t)\right| = 1$
> – $\forall t \in \mathbb{R}^+, \left|A_{[t]}(t)\right| \geq \left|A_{[f]}(t)\right|$

5) Définition des relations à $t = 0$ pour les agents connus

Cette étape a pour objet la définition de la structure initiale du SMA. Elle consiste en l'énoncé de spécifications sur $R(0)$.

Elle revêt une signification différente selon le type d'organisation conçue : dans le cas d'une organisation structurellement statique, il s'agit de définir l'ensemble des relations du système, et donc la structure complète de l'organisation considérée ; dans le cas d'une organisation structurellement dynamique, il s'agit de définir les relations initiales, c'est-à-dire l'initialisation du processus d'évolution de la structure défini par l'ensemble des mécanismes structuraux.

L'étape consiste, dans les deux cas, à rechercher l'ensemble des relations des types définis précédemment qui sont nécessaires au bon fonctionnement du système – sur le long terme ou seulement d'un point de vue de l'initialisation – c'est-à-dire qui répondent à l'existence de dépendances ou à la nécessité d'échanges entre agents.

> *Pour initier la structure du système conçu, il est suffisant que les robots forment un ensemble connexe ; il leur suffit ensuite, par le biais d'échanges (cf. infra, définition des mécanismes organisationnels), d'établir les autres relations qui leur sont nécessaires.*

Ainsi il est suffisant, dans cet exemple, d'assurer l'existence d'une relation initiale de type relation de localisation λ_l entre chaque agent du système et l'agent base a_b :

$$\forall a \in A(0), \exists r \in R(0) \mid r = (a, a_b, \lambda_l)$$

6) Spécifications sur les tâches

Enfin, la dernière étape consiste à énoncer les éventuelles spécifications sur les tâches présentes dans le système. Il s'agit de définir des spécifications sur $P_a(t)$ ou $P(t)$.

Cette étape permet de décrire les attentes fonctionnelles vis-à-vis du système, et ainsi de spécifier des fonctions principales ou continues du système. Il s'agit donc d'énoncer, dans cette étape, l'ensemble des attentes fonctionnelles vis-à-vis des agents connus et des rôles normatifs décrits.

Dans notre exemple, nous définissons une seule spécification sur les tâches, correspondant à la nécessité pour chaque agent de veiller au maintien de sa charge. Tout agent possède ainsi une tâche x dont il est l'unique bénéficiaire ($B_x = \{a\}$) et qui est de type ϕ_A (on lui donne l'identifiant 0) :

$$\forall a \in A, \exists x = (\{a\}, 0, \phi_A) \mid \forall t \in L(a), x \in P_a(t)$$

Il serait possible de définir d'autres spécifications sur les tâches des agents, comme l'obligation d'explorer l'espace en permanence, par exemple, mais en l'absence de conditions particulières, ces spécifications ne semblent pas nécessaires pour garantir la cohérence de l'organisation en cours de conception.

Spécification dynamique

A la fin de ces étapes de conception, nous obtenons la spécification d'une organisation statique : il s'agit de la valeur initiale de l'organisation. Afin de compléter la description de l'organisation, il faut donc ajouter les spécifications dynamiques de celle-ci. D'un point de vue de conception, il peut être difficile de trouver directement la dynamique optimale complétant une organisation initiale. Pour trouver une dynamique répondant aux attentes de conception, il est donc avantageux d'utiliser une méthode itérative permettant de concevoir puis de valider ou d'invalider le modèle proposé et de l'améliorer. Les capacités calculatoires prédictives de notre modèle nous permettent de répondre à cette problématique.

Dans l'ensemble de ces étapes, l'objectif est d'énoncer des spécifications sur les réservoirs organisationnels globaux et spécifiques : $\mathcal{GM}$ et $\mathcal{SM}(\alpha)$ pour tous types α d'agents.

Les étapes itératives de la définition d'une dynamique organisationnelle que nous proposons sont exprimées ci-dessous.

7) Identification des problèmes dynamiques de l'organisation

Cette étape consiste à définir les problèmes dynamiques posés par l'ensemble de spécifications statiques déjà énoncées. L'objectif est de quantifier et de formaliser les besoins d'invariants, de contrôle et de stabilité pour les dimensions structurelle et fonctionnelle de l'organisation.

a. Définition de propriétés générales significatives

La première étape que nous proposons est de définir un ensemble de propriétés sur le système qui mesurent, d'une manière ou d'une autre, un aspect de l'organisation qui pose problème du point de vue de la dynamique. Ces propriétés peuvent être de différents types, entre autres : binaires, entières, décimales, vectorielles,...

En règle générale, il parait plus simple de commencer par la description des propriétés liées à la structure, et de continuer par les propriétés fonctionnelles.

Dans notre exemple, nous pouvons ainsi définir :

- $Ppté1$ *(binaire) : existence d'une relation de signalisation* λ_s *entre tous les agents explorateurs* $[e]$ *et les bases* $[b]$[52] *auxquelles ils sont liés par une relation de localisation* λ_l.
- $Ppté2$ *(binaire) : existence d'une relation d'information* λ_i *entre les agents bases* $[b]$ *et les agents foreurs* $[f]$ *qui leur sont liés par une relation de localisation* λ_l.
- $Ppté3$ *(décimale) : nombre moyen d'agents explorateurs* $[e]$ *en relation de proximité* λ_p *avec chaque agent explorateur.*
- $Ppté4$ *(décimale) : nombre moyen d'agents transporteurs* $[t]$ *affectés* (λ_a) *à chaque agent foreur* $[f]$.
- $Ppté5$ *(décimale) : probabilité qu'un agent foreur ne puisse plus extraire de minerai par impossibilité d'affecter la tâche de transport* ϕ_T *de ce minerai à un agent transporteur* $[t]$.

b. Expression des problèmes dynamiques sous forme d'invariances de propriétés

L'étape suivante consiste à exprimer les attentes structurelles et fonctionnelles sur le système comme des contraintes sur les propriétés définies préalablement. Ces contraintes peuvent

[52] Même si l'on choisit de n'avoir qu'une seule base dans le système, il n'est pas nécessaire d'utiliser cette connaissance dans la description des mécanismes organisationnels ; cela permet de les rendre plus génériques, et ainsi potentiellement plus réutilisables.

prendre la forme d'invariants stricts à valider en tout instant ou bien d'objectifs à atteindre (contraintes souples), tels que minimiser ou maximiser une valeur.

> *A partir des propriétés définies, nous pouvons, selon les attentes que nous fixons pour la conception de l'organisation, proposer divers ensembles de contraintes :*
>
> – *La validation de la* $\mathrm{Ppt\acute{e}1}$ *pour tout* $t \in \mathbb{R}^+$*, pour garantir les échanges d'information dans le système :*
>
> $$\forall t \in \mathbb{R}^+, Ppt\acute{e}1(t)$$
>
> – *La validation de la* $\mathrm{Ppt\acute{e}2}$ *pour tout* $t \in \mathbb{R}^+$*, pour garantir les échanges d'information dans le système :*
>
> $$\forall t \in \mathbb{R}^+, Ppt\acute{e}2(t)$$
>
> – *Le contrôle de la* $\mathrm{Ppt\acute{e}3}$*, par exemple sa minimisation vers* 0*, pour garantir la plus vaste zone de couverture possible pour l'exploration :*
>
> $$minimiser(Ppt\acute{e}3)$$
>
> – *La maximisation de la* $\mathrm{Ppt\acute{e}4}$*, et éventuellement sa normalisation à une valeur supérieure à* 1*, pour assurer la possibilité de transport de minerai et l'activité efficace des agents transporteurs :*
>
> $$\forall t \in \mathbb{R}^+, Ppt\acute{e}4(t) \geq 1$$
> $$maximiser(Ppt\acute{e}4)$$
>
> – *La minimisation de la* $\mathrm{Ppt\acute{e}5}$*, pour tendre à la plus grande efficacité possible dans l'extraction du minerai (en évitant les blocages) :*
>
> $$minimiser(Ppt\acute{e}5)$$

8) Proposition de mécanismes organisationnels

Cette étape consiste à proposer, en fonction des contraintes dynamiques énoncées à l'étape précédente, des mécanismes organisationnels permettant de répondre aux attentes sur la dynamique organisationnelle du système. Les contraintes structurelles doivent être vérifiées par la définition de mécanismes structuraux, tandis que la vérification des contraintes fonctionnelles peut impliquer la définition de tous types de mécanismes organisationnels.

> *Dans l'exemple considéré, nous pouvons assez facilement répondre aux contraintes explicitées par la définition d'un petit nombre de mécanismes organisationnels simples :*
>
> – *Pour répondre à « $\forall t \in \mathbb{R}^+, Ppt\acute{e}1$ », nous définissons un mécanisme structural (libre)* S_1 *qui assure la création d'une relation de type* λ_s *lors de l'existence d'une relation* λ_l *avec un agent particulier :*
>
> $$S_1 : r_1 = (a, b, \lambda_l) \in R_a(t) \Rightarrow \exists r_2 = (a, b, \lambda_s) \in R_a(t + \Delta t)$$

- *Pour répondre à « $\forall t \in \mathbb{R}^+, Ppté2$ », nous définissons de même un mécanisme (hétérogène réactif) S_2 construisant les relations λ_i lors de l'inscription[53] des agents foreurs (ce qui correspond à un changement d'état de l'agent base) :*

$$S_2: inscription(a,b,t) \Rightarrow \exists r = (a,b,\lambda_i) \in R_a(t+\Delta t)$$

- *Pour répondre à « $minimiser(Ppté3)$ », nous définissons un mécanisme (hétérogène réactif) S_3 correspondant à la construction d'une relation de type λ_p lorsqu'un agent explorateur est perçu[54], dans l'environnement, par l'agent ; cette relation doit servir à entamer une correction de trajectoire prenant en compte la position des autres agents (dimension Environnement) :*

$$S_3:\ vue(a,b,t) \Leftrightarrow \exists r = \left(a,b,\lambda_p\right) \in R_a(t+\Delta t)$$

- *Pour répondre à « $maximiser(Ppté4)$ », nous devons définir un mécanisme (hétérogène téléonomique) S_4 gérant l'affectation des agents transporteurs. Nous pouvons par exemple définir le mécanisme comme entraînant la génération d'une relation d'affectation λ_a lorsqu'un agent foreur est à un gisement en train de forer[55] et qu'il n'a pas d'agent transporteur affecté, ou alors lorsqu'un agent foreur est en attente de transport et qu'un agent transporteur est inactif (il ne transporte pas de minerai). Dès lors, l'agent transporteur affecté est soit l'agent libre (non affecté) le plus proche s'il existe, soit l'agent inactif[56] le plus proche s'il existe, soit un agent aléatoire parmi ceux affecté à l'agent foreur disposant du plus d'agents transporteurs associés sinon :*

$$S_4:$$
$$\left(mine(a,t) \wedge \nexists r = (a,b,\lambda_a) \in R_a(t)\right) \Rightarrow \exists r' = (a,b',\lambda_a) \in R_a(t+\Delta t)$$
$$\left(\left(\forall b \mid (a,b,\lambda_a) \in R_a(t), occupé(b)\right) \wedge \left(\exists c \in A_{[t]}(t) \mid \neg occupé(c)\right)\right) \Rightarrow \exists r = (a,b',\lambda_a) \in R_a(t+\Delta t)$$

- *Pour répondre à « $minimiser(Ppté5)$ », il faut définir un mécanisme fonctionnel (hétérogène réactif projectif) F_1 qui décide de l'allocation des tâches de transport (ϕ_T)*

---

53 Nous nommons $inscription(a,b,t)$ la propriété sur $\xi(t)$ qui est vraie si et seulement si l'agent b a construit une relation de type λ_l avec l'agent a et le lui a signalé à l'instant t considéré.

54 $vue(a,b,t)$est vrai si et seulement si l'agent a perçoit l'agent b à l'instant t.

55 C'est-à-dire qu'il possède une tâche d'extraction de minerai : $mine(a,t) \equiv \exists x = (\emptyset, n_x, \phi_F) \in P_a(t), n_x \in \mathbb{N}$

56 L'activité est définie par le prédicat $occupé(a) \equiv \exists x = (B_x, n_x, \phi) \in P_a(t)$, $n_x \in \mathbb{N}, \phi = \phi_R \wedge \phi_T$

par les agents foreurs[57] *aux agents transporteurs qui leur sont affectés. Ce mécanisme peut être, par exemple, le choix de l'agent transporteur affecté inactif le plus proche, s'il existe, puis le choix de l'agent transporteur affecté le plus proche sinon :*

$$F_1\colon\ besointransport(a,t) \wedge\Rightarrow \exists b \mid (a,b,\lambda_a) \in R_a(t), \qquad \big(b,(\{a\},0,\phi_T)\big) \in D_a(t+\Delta t)$$

Ces mécanismes sont affectés aux agents pour lesquels ils ont été conçus :

$$\{S_1,S_3\} \subseteq \mathcal{SM}([e])$$
$$S_2 \in \mathcal{SM}([b])$$
$$\{S_4,F_1\} \subseteq \mathcal{SM}([f])$$

Nous ajoutons à cet ensemble de mécanismes le mécanisme d'insertion I_1 *(cf. section 4.2.1). Il définit l'ensemble des relations initiales de tout agent qui entre dans le système à un instant* $t \neq 0$ *; cet ensemble de relations est ici réduit à la seule relation de localisation* λ_l *avec une base du système (plus précisément dans l'organisation décrite, avec l'unique base du système) :*

$$I_1\colon a \in A(t) \wedge a \notin A(t-\Delta t) \Rightarrow \exists r \in R(t) \mid r = (a,a_b,\lambda_l)$$

Ce mécanisme est affecté à tous les types d'agents :

$$\forall \alpha \in T_A, I_1 \in \mathcal{SM}(\alpha)$$

9) Évaluation des mécanismes selon les propriétés choisies

Cette étape consiste à évaluer l'impact des mécanismes sur la dynamique organisationnelle à travers les indications qu'il est possible d'obtenir sur les propriétés énoncées précédemment et les contraintes associées.

Cette évaluation consiste à déterminer, de la manière la plus exacte possible, la validité ou l'invalidité des contraintes définies. Ceci peut être effectué à travers un certain nombre de méthodes différentes, adaptées à des types de mécanismes et de propriétés souhaitées différents :

[57] Le prédicat $besointransport(a,t)$ définit si oui ou non l'agent foreur a requiert un transport de minerai à l'instant t. Sa valeur dépend de l'état $\xi_a(t)$ de l'agent foreur (cette dépendance est définie par les capacités et les propriétés propres de a, qui ne sont pas des spécifications organisationnelles, mais agent).

- en logique du premier ordre :

  Les propriétés binaires ou entières peuvent être validées par une analyse logique, notamment dans le cas où les mécanismes correspondent à des mécanismes exacts, assurant l'existence d'une relation sous certaines conditions par exemple – c'est le cas généralement pour des mécanismes homogènes.

- en suivant la méthode de modélisation par chaînes de Markov (cf. chapitre 5) :

  Les propriétés décimales et éventuellement vectorielles peuvent être validées par le calcul de moyennes théoriques en suivant la méthode de modélisation des mécanismes par des chaînes de Markov (comme présenté au chapitre 5) ; ceci est généralement approprié pour des mécanismes hétérogènes dont la précision (cf. section 5.4.2) est suffisante – c'est-à-dire dépendant d'états de l'agent suivant le plus possible la propriété de Markov.

- en suivant d'autres méthodes analytiques (logique floue, réseaux d'automates cellulaires, etc., cf. section 5.2.1) :

  Dans le cas où les propriétés (décimales ou vectorielles, principalement), sont assurées par des mécanismes organisationnels hétérogènes dont la précision est trop faible, il peut être préférable de sélectionner une méthode analytique alternative à la modélisation markovienne ; ces autres méthodes peuvent alors être utilisées pour approximer les valeurs des propriétés recherchées, et valider ou invalider ainsi les contraintes spécifiées.

- par simulation :

  Dans le cas où les approches analytiques ne sont pas concluantes, par exemple pour des propriétés assurées par des mécanismes hétérogènes à faible précision et à forte spécificité (cf. section 5.4.2), il peut être plus intéressant d'utiliser la simulation du système suivant différents choix de situations pour étudier la validité des contraintes proposées dans ces différentes situations ; il est alors nécessaire de se référer à des méthodes de test par simulation (Fortino *et al.*, 2005).

10)Sélection et correction des mécanismes

La dernière étape de cette conception dynamique consiste à utiliser les résultats des évaluations calculées à l'étape précédente pour corriger et sélectionner les mécanismes organisationnels. Suite à l'application de ces corrections, il est possible, si nécessaire, de boucler de manière itérative les étapes de conception dynamique (étapes 7 à 10 – spécifications des mécanismes organisationnels) afin tout d'abord d'ajouter des conditions sur des propriétés dynamiques dont l'importance aurait été découverte lors des analyses

dynamiques (étape 9 – évaluation des mécanismes) précédentes, puis d'adapter l'ensemble des mécanismes organisationnels à ces nouvelles conditions.

La correction des mécanismes organisationnels peut prendre plusieurs aspects différents, dans un ordre de moindre à plus grande modification :

- optimisation des paramètres des mécanismes qui en sont dotés, afin d'optimiser les propriétés qui en dépendent ;
- ajustement des conditions des mécanismes existants, afin d'approcher le plus possible des conditions requises ;
- ajout de mécanismes pour garantir les conditions qui ne le sont pas encore ;
- suppression de mécanismes erronés qui induisent des problèmes organisationnels comme l'impossibilité de garantir des conditions souhaitées.

L'application itérative de ces étapes permet de compléter la spécification de l'organisation par l'adjonction de la dynamique organisationnelle. Elle permet ainsi l'obtention d'une spécification complète d'une organisation pour le système. Ensuite, la dernière étape pratique de l'utilisation de notre modèle pour la conception de SMA est la conversion de la spécification organisationnelle obtenue en squelette de code permettant par la suite d'implémenter les autres dimensions de la méthode VOYELLES en disposant d'un cadre organisationnel préétabli. Nous avons pour ce faire besoin d'établir une correspondance entre les spécifications organisationnelles et des contraintes de conception sur le code du SMA qui les instancie.

6.2 Correspondance en POO

Il est possible d'établir des correspondances entre les concepts organisationnels utilisés dans notre modèle et les éléments de tout langage de programmation, en reliant la sémantique des concepts organisationnels à l'expression sémantique des langages considérés. Les langages de programmation utilisés dans la création de SMA sont actuellement majoritairement des langages orientés objet : si les codes des agents gagnent parfois à être exprimés dans d'autres types de langages, notamment fonctionnels, les agents et messages étant des entités indépendantes et bien définies, la programmation de SMA s'est en effet dirigée vers les langages orientés objet (Ricordel, 2001 ; Gutknecht et Ferber, 2000). Au vu de cela, nous choisissons de donner ici une correspondance générique entre nos concepts organisationnels et les éléments usuels de programmation orientée objet.

Pour cela, nous partons des sémantiques des spécifications exprimables dans notre modèle pour les convertir en un ensemble d'éléments pouvant servir à la réalisation

automatisée d'un squelette de code[58] à partir d'une description organisationnelle. Les correspondances (cf. Tableau 6-B) proviennent ainsi d'un lien sémantique, tout en conservant dans le code les relations de dépendance qu'elles entretiennent dans le modèle.

Tableau 6-B - Correspondance entre notions organisationnelles et concepts de POO

Élément	Notion POO	Dérivé de	Implémentant
Agent	Objet		
Type d'agent	Classe	Classe *Agent*	*Runnable*
Relation	Objet		
Type de relation	Classe	Classe *Relation*	
Tâche	Objet		
Type de tâche	Classe	Classe *Tâche*	*Serializable*
Mécanisme organisationnel	Interface et méthode	Interface *mecanisme*	

Les premiers éléments à associer sont les éléments constitutifs des organisations : **les agents, les relations, les tâches**. Il est également nécessaire, dans le même temps, de définir les notions qui correspondent aux types de ces éléments. Chacun de ces éléments est défini par un ensemble de valeurs et de fonctions spécifiques : les agents ont un type, c'est-à-dire un code, et un identifiant ; les relations ont des agents source et cible et un type ; les tâches ont un ensemble d'agents bénéficiaires spécifié, un type et un identifiant. La solution de représentation la plus simple et la plus expressive pour de telles notions en POO est l'**objet**. Ils sont des instances de classes dérivant de trois **classes abstraites** distinctes : une classe Agent, une classe Relation et une classe Tache. Par extension, les types des éléments considérés sont donc représentés par des **classes instanciables**, dérivées des classes abstraites correspondantes : les types d'agents sont des classes dérivées de la classe Agent, les types de relations des classes dérivées de la classe Relation et les types de tâches des classes dérivées de la classe Tache.

Certaines propriétés sémantiques devant être garanties par la correspondance proposée, des propriétés supplémentaires doivent être spécifiées :

[58] Il ne s'agit que d'un « squelette » et non de code complet, étant donnée l'absence, dans une description organisationnelle, de spécification sur les autres dimensions descriptives du multi-agent : code des agents, description de l'environnement, canaux et protocoles de communication.

- La classe Agent est **exécutable en tant que thread** : un agent étant une entité autonome active, elle doit être exécutée dans un thread indépendant. En Java, par exemple, la classe Agent implémentera l'interface *Runnable*.
- La classe Tache est **sérialisable** : elle doit en effet pouvoir être transmise d'un objet à l'autre de manière identifiable et unique, lors de la création par les agents de leurs *extensions sociales* – éventuellement sur une machine distance via le réseau si le système multi-agent est distribué sur plusieurs machines. En Java, par exemple, la classe Tache implémentera l'interface *Serializable*.

Ensuite, ce sont les **mécanismes organisationnels** qui doivent être définis comme des éléments de code. Les mécanismes organisationnels sont des processus de calcul permettant, à partir d'informations internes à l'agent, de calculer les états de son *champ relationnel* et de son *extension sociale* (cf. sections 4.1.2 et 4.2.2), c'est-à-dire, de créer et manipuler des relations et de dupliquer et transmettre des tâches, mais aussi de stocker de l'information sur celles-ci. En tant que mécanismes internes à un agent, ils seront naturellement codés comme des **méthodes** propres aux classes correspondant aux types d'agents auxquels ils sont associés. Ces méthodes ont besoin, pour fonctionner, de l'existence, dans l'objet agent auquel elles appartiennent, d'un certain nombre de structures, représentées d'une manière similaire :

- le *champ relationnel* de l'agent, représenté par un objet d'une classe spécifique, doté des accesseurs et des mutateurs appropriés ;
- l'*extension sociale* de l'agent, également représentée par un objet d'une classe spécifique, doté d'accesseurs et de mutateurs appropriés ;
- les informations stockées par l'agent et concernant son état courant – historiques d'interactions, situation,... –, représentées par des objets divers mais dotés d'accesseurs et de mutateurs spécifiquement requis par ce mécanisme et représentés par une interface spécifique (au sens de l'*interface* Java) ;
- les objectifs actuels de l'agent, c'est-à-dire son *champ d'action* (cf. section 4.1.3), représenté par un objet d'une classe spécifique, également doté d'accesseurs et de mutateurs appropriés.

Pour plus de clarté et la possibilité d'une validation formelle du code généré, ces différents composants, ainsi que la méthode définissant le mécanisme organisationnel, sont regroupés dans une **interface** spécifique, implémentée par les classes d'agents dont les types possèdent ce mécanisme. L'interface d'un mécanisme étend, selon le type de celui-ci, soit l'interface correspondant aux mécanismes structuraux, soit celle correspondant aux

mécanismes fonctionnels, qui toutes deux étendent une interface correspondant aux mécanismes organisationnels[59].

Par ailleurs, dans le cas où les besoins des développeurs se portent sur des descriptions de plus haut niveau, pourront être ajoutées à ces correspondances, celles permettant de décrire de manière concrète les notions haut-niveau intégrables à notre modèle. Ainsi, pourront être décrits :

- les *rôles effectifs* comme des fonctions booléennes des agents, sur les champs d'action, retournant si oui ou non, l'agent regroupe couramment l'ensemble des actions définissant le rôle ;
- les *rôles normatifs* comme des interfaces implémentées par l'ensemble des classes – et uniquement celles-ci – qui peuvent « jouer » ce rôle ;
- les *groupes* comme des méthodes des agents retournant si oui ou non, l'agent fait partie du groupe en question – par le test récursif de l'existence des relations requises pour l'agent en question et tous les autres agents du groupe ;
- les *coalitions* comme des méthodes des agents retournant si oui ou non l'agent fait partie du groupe correspondant et si le groupe partage des objectifs communs – par l'intersection récursive des tâches des agents du groupe.

Ces correspondances (cf. Figure 6.3), bien qu'elles ne se réfèrent pas directement à des éléments organisationnels de base de notre modèle, peuvent être intégrées utilement aux descriptions organisationnelles et aux codes générés, dans le but de faciliter l'évaluation des organisations du système lors de son exécution, ou encore de faciliter, pour des agents cognitifs, un niveau plus élevé de représentation organisationnelle en temps réel à l'exécution.

[59] Toujours pour des raisons de clarté et de modularité du code, il peut être intéressant de spécifier davantage d'interfaces étendant ceux proposés ci-dessus, et correspondant aux différentes complexités possibles, comme vues section 4.2.

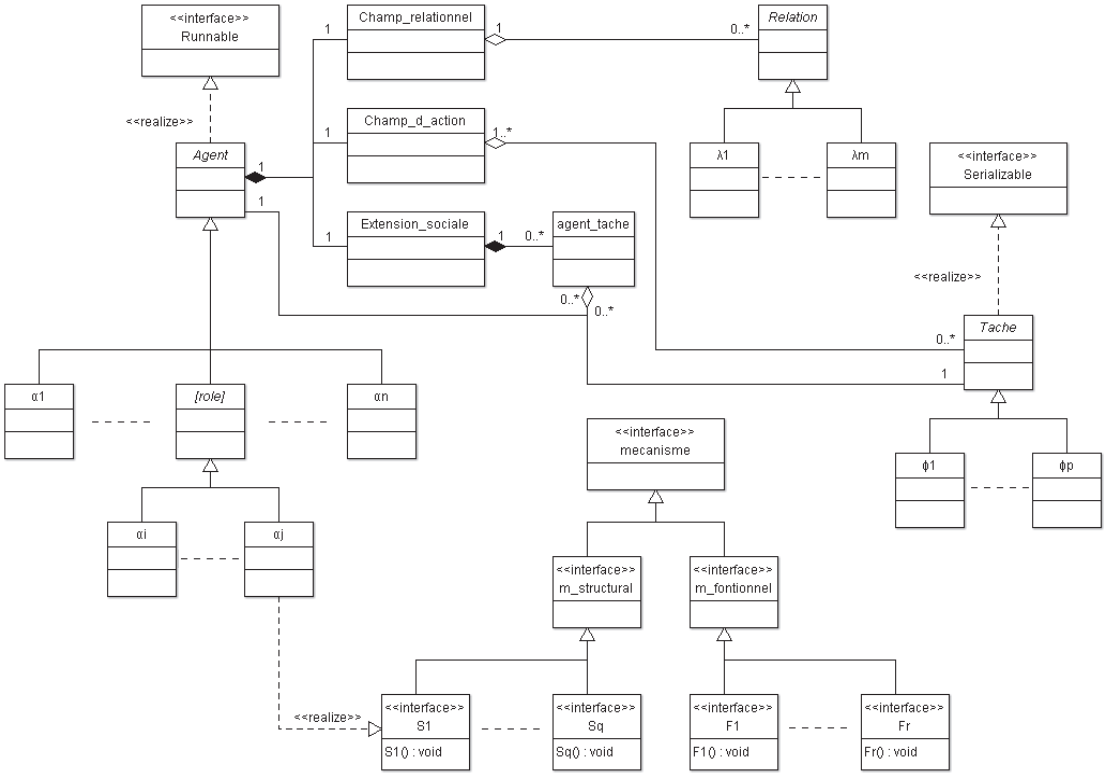


Figure 6.3 - Description UML des équivalences aux notions organisationnelles

6.3 Utilisation pratique des spécifications de l'organisation

6.3.1 Génération semi-automatique à partir des spécifications

A partir de la correspondance établie entre éléments organisationnels et notions de programmation orientée objet – à titre d'exemple, car nous avons vu qu'il était possible de spécifier le même type de correspondance avec d'autres paradigmes de programmation –, il est ensuite possible d'instrumenter la création de code de SMA à partir d'une description organisationnelle.

Pour cela, nous utilisons le fait que les spécifications – limitatives, constructives et alternatives – composant une organisation sont établies à partir des éléments constitutifs étudiés ci-dessus. Plus précisément, une organisation est un ensemble de spécifications, chacune ne mettant en jeu que des ensembles formés des éléments de base et de leurs types : agents, relations, tâches, mécanismes. Pour générer un squelette de code aidant par la suite les développeurs à créer le code final du SMA voulu, il suffit donc d'extraire de l'ensemble des spécifications d'une organisation la totalité des éléments de base définis dans ces spécifications. Tous ces éléments sont ensuite convertis dans leur représentation

correspondante en code. Ce squelette, comprenant classes d'objets, objets, interfaces, champs et méthodes, peut être fourni aux développeurs pour les assister dans leur travail, ou bien peut être associé et complété par des données provenant de la modélisation d'autres dimensions descriptives du SMA à travers des modèles appropriés. Ensuite, les spécifications en elles-mêmes peuvent être garanties par deux moyens, selon leur nature :

- Les spécifications statiques ou qui concernent les mécanismes organisationnels peuvent être vérifiées lors de la réalisation du code du système. Il faut pour cela qu'un **plugin** à l'environnement de développement (IDE) vérifie, en temps réel ou à la demande, la validité des spécifications organisationnelles par rapport au code proposé. Cette vérification ne peut être effectuée que si les développeurs fondent leur code sur les bases posées par le code généré automatiquement[60].
- Les spécifications dynamiques – en particulier celles qui s'appuient sur les agents présents dans le système $A(t)$ et sur les champs d'action $P_a(t)$ – ne sont pas nécessairement vérifiables, car elles mettent en jeu des dynamiques – dynamique existentielle et dynamique téléonomique –, qui, bien qu'organisationnelles, prennent leur origine dans des faits extérieurs à l'organisation elle-même. Il est en revanche envisageable de vérifier ces spécifications à l'exécution de deux manières différentes :
  - soit par le biais d'une sonde extérieure – analyse de la mémoire de travail du SMA, capture d'information par les protocoles d'interaction, etc. – si le système tourne sur une seule machine ou un nombre fini de machines connues et synchronisées ;
  - soit par le moyen d'un mécanisme de *snapshot distribué*, implémenté par les agents – donc inclus dans une méthode de la classe Agent – tel que présenté dans (Chandy et Lamport, 1985 ; Mattern, 1989), et renvoyant une valeur de sortie correspondant à la propriété cherchée – nombre d'agents, tâches en cours d'exécutions, etc.

Outre la génération du code correspondant à tous les éléments définis dans une spécification organisationnelle, un module de génération semi-automatique de code pourra donc proposer également un plugin permettant la validation des spécifications organisationnelles au cours de la réalisation du SMA ; il pourra également créer une sonde de validation à l'exécution des propriétés dynamiques – pour, par exemple, des applications de

[60] Par exemple, que tous leurs types d'agents dérivent de la classe Agent, que tous leurs mécanismes organisationnels étendent l'interface correspondant, etc.

simulation – ou des mécanismes de snapshot distribués intégrés aux agents et permettant cette vérification à la volée quel que soit l'implantation du système.

6.3.2 Intégration à un modèle général d'un SMA

Bien qu'il soit, de cette façon, possible d'utiliser des spécifications uniquement organisationnelles pour l'analyse et la conception d'un SMA, la réalisation complète de celui-ci implique des choix sur les autres dimensions descriptives de ces systèmes : Agent, Environnement, Interaction.

Pour formaliser, normaliser et justifier ces choix de réalisation, il est utile, sinon nécessaire, d'intégrer ces notions lors de la conception du système. C'est pourquoi notre modèle se place, dans une optique de conception des SMA, comme l'une des briques descriptives nécessaires à la conception d'un système. Ces briques correspondent aux dimensions définies dans notre approche, et se regroupent par la mise en relation de ces descriptions dans la réalisation d'un système complet.

Comme exemple de mise en relation complète des descriptions d'un SMA pour sa conception, nous nous situons par rapport à la méthode développée pour la plateforme VOLCANO (Ricordel, 2001) et utilisant le langage de description MADEL. Dans ce langage (Ricordel et Demazeau, 2001), la description organisationnelle que nous proposons correspond principalement à la description de la *brique Organisations*, mais également à la description d'*instances* et d'*éléments*.

Dans un tel modèle, nous avons les représentations suivantes :

- Les **agents** sont représentés par des *entités*.
- Les **types d'agents** sont représentés par des *composants distribués* de la brique *Agents* ; il est à noter que, comme la description précise des architectures liées aux types d'agents n'est pas contenue dans la description organisationnelle, les *composants distribués* ainsi définis ne comprennent aucune *propriété*.
- Les **relations** et les **tâches** ne sont pas représentables dans le modèle, car il ne permet de spécifier comme lien entre les entités que des objets de la *brique Interactions*, c'est-à-dire des protocoles, canaux ou langages.
- Les **mécanismes organisationnels** peuvent être représentés par des *composants distribués* de la *brique Organisations*. Leurs instanciations dans les agents sont représentées par des *instances*, qui doivent être reliées par des *interbriques* aux types d'agents qui les supportent.

Nous voyons donc tout d'abord que l'intérêt de la description organisationnelle que nous proposons est accru si celle-ci est associée aux descriptions des autres dimensions du multi-agent. D'autre part, nous notons qu'il sera nécessaire de développer un autre langage de description global, étant donné que MADEL, celui proposé avec la méthode VOYELLES, ne permet pas d'intégrer l'ensemble des descriptions organisationnelles proposées, et surtout, ne permet pas de gérer la validité des spécifications.

6.4 Utilisation des évaluations de dynamique

L'un des avantages de notre modèle est la possibilité d'obtenir des évaluations de la dynamique d'une organisation, dès la conception. Cette évaluation peut être faite à plusieurs moments clés : lors de la définition des mécanismes organisationnels, pour valider des prérequis sur l'organisation générale et améliorer de manière itératives les choix de mécanismes lorsque les propriétés globales estimées ne correspondent pas avec les attentes de conception (cf. étape 9, section 6.1.1) ; ou bien lors de l'analyse d'une instance particulière d'un SMA, pour comprendre les problèmes organisationnels qui ont pu se poser dans cette instance particulière et leurs origines.

La pratique de cette évaluation dynamique se fait suivant les modalités présentées dans le chapitre précédent : par la modélisation des mécanismes sous forme de chaînes de Markov et la réalisation des matrices de transition associées, puis par le calcul des états stationnaires. A cela s'ajoute la définition de propriétés liées à ces états moyens estimés et à la matrice de transition. Ces propriétés – par exemple le nombre moyen d'agents de type α en relation γ avec un agent de type β dans le système – permettent de spécifier des attentes, des besoins ou des risques liés à l'organisation dans le SMA résultant. Ces propriétés sont donc construites à partir des probabilités de transitions en états, elles-mêmes exprimées :

- soit de manière absolue, pour n'importe quel système instanciant l'organisation ;
- soit de manière relative à une ou plusieurs variables concernant l'organisation – par exemple, dans l'exemple donné, la proportion d'agents de type β présents dans le système.

A partir de l'expression de ces propriétés et du calcul de leur valeur pour une organisation donnée – et en particulier pour un ensemble de mécanismes donnés – il est possible de tirer des conclusions quant à l'adéquation de cette organisation dynamique avec les prérequis formulés préalablement sur les propriétés dynamiques de cette organisation.

Deux types de réactions peuvent suivre cela dans le processus méthodologique de conception de SMA (cf. étape 10, section 6.1.1) :

- Soit il est possible d'ajuster l'ensemble de mécanismes organisationnels pour corriger les propriétés de l'organisation, soit par l'amélioration absolue de la propriété, soit par l'amélioration de celle-ci pour certaines valeurs des variables qui entrent dans son calcul – en général, pour le « pire cas » – et cela de différentes façons :
  - par ajout de mécanismes supplémentaires destinés à réguler des cas jusqu'ici non pris en compte ;
  - par suppression de mécanismes ineffectifs – et leur éventuel remplacement par de meilleurs ;
  - par modification des mécanismes, ajouts de cas particuliers, prise en compte de différents critères ou ajustement de paramètres internes aux mécanismes.
- Soit il est envisageable de rajouter d'autres types de contraintes organisationnelles – c'est-à-dire de nouvelles spécifications – pour restreindre ou supprimer les risques organisationnels évalués précédemment, notamment en jouant sur la limitation du champ des éventuelles variables organisationnelles présentes dans le calcul de l'estimation de la propriété concernée.

Il est ainsi possible d'intégrer les estimations fournies par la modélisation markovienne que nous proposons de lier à notre modèle, à une méthodologie globale permettant d'élaborer une organisation, de prévoir, estimer et optimiser sa dynamique et de générer, à partir de cela, les éléments de code et de méta-code – modules de surveillance de spécification notamment – qui pourront aider à la réalisation complète d'un SMA prenant en compte des nécessités organisationnelles spécifiques.

Chapitre 7. Applications

7.1 Conception de système : Projet CLIC

Le projet CLIC – Conception d'un Logiciel Interactif Collaboratif – (Lacomme *et al.*, 2010) s'inscrit dans le cadre d'un événement biennal organisé par le théâtre de Grenoble (Meylan) l'Hexagone, nommé les « Rencontres-i ». Cet événement s'articule autour de la collaboration entre arts et sciences, et dispose à chaque édition d'un thème différent, le thème dans lequel s'insère le projet CLIC étant intitulé « butinage et essaimage ». Le projet CLIC lui-même implique l'équipe MAGMA et un collectif d'artistes nommé Coincoin Production[61].

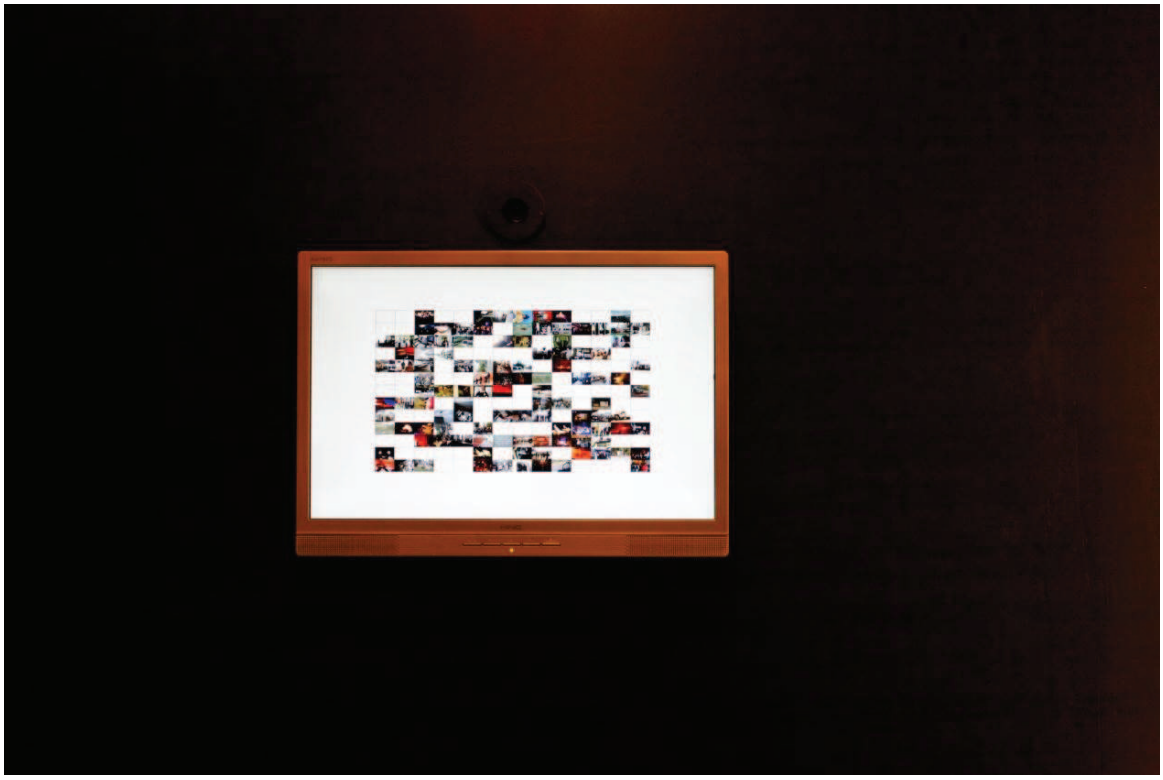


Figure 7.1 - L'un des affichages de l'installation autonome et interactive CLIC

L'idée artistique poursuivie dans ce travail consiste en la présentation, la réorganisation et la mise en relation d'images et de sons capturés depuis un lieu bien connu des visiteurs de l'exposition : le campus universitaire de Grenoble. Le résultat escompté prend la forme d'une

[61] http://www.collectif-coin.com

installation multimédia dynamique et interactive, exposée et accessible au public pendant 3 semaines (cf. Figure 7.1).

L'objectif informatique dans ce projet est donc de définir, de concevoir et de développer un logiciel, incorporé à la plateforme de programmation graphique multimédia *Max/MSP/Jitter*[62], capable de supporter la création et le déploiement d'une exposition multimédia autonome. L'idée majeure derrière la création du logiciel est de permettre à celui-ci de n'être pas seulement un logiciel *ad hoc* destiné à cette exposition particulière, mais un système flexible, réutilisable, et aisément manipulable par des artistes, ceci afin de permettre la réutilisation du logiciel par différents artistes pour la construction de multiples expositions multimédia autonomes.

L'exposition réalisée par les artistes, quant à elle, permet de montrer l'adéquation du logiciel avec les besoins des artistes. Et, bien qu'il soit conçu dans un cadre plus vaste que celui de l'exposition réalisée, son application réussie montre l'utilité et la fonctionnalité du système créé.

7.1.1 Contexte applicatif

L'objectif artistique du projet étant de construire une installation autonome et dynamique, les caractéristiques sont que le logiciel contrôlant l'installation doit être capable de contrôler son évolution et sa dynamique au cours du temps, sans contrôle direct des artistes. Le but de la conception du logiciel peut donc être spécifié comme étant de concevoir un logiciel autonome, évolutif, dynamique, dont le comportement et l'évolution peuvent contenir à la fois des contraintes scriptées et une part aléatoire. Ainsi, les artistes préparant l'exposition peuvent définir un comportement global pour celle-ci, afin d'exprimer leur créativité artistique, mais l'évolution précise de l'installation au cours du temps dépend directement du logiciel et des interactions de celui-ci avec son environnement – logiciel et, indirectement, matériel et humain.

La contrainte duale provenant d'une nécessité de contrôle global, de liberté de choix et de spécification dans le rendu artistique d'une part, mais aussi d'aléatoire, de réactivité aux conditions et aux interactions d'autres part, est la raison pour laquelle l'ajout de capacités d'intelligence artificielle apparaît comme nécessaire à la conception du système. En effet, celles-ci permettent au système d'être à même d'appréhender des contraintes et des situations toujours changeantes, avec toujours une grande liberté de choix des règles comportementales et une capacité de spécifications de haut niveau inégalée dans les autres

[62] http://cycling74.com/products/maxmspjitter/

modes de conception logicielle. La non-linéarité des algorithmes d'intelligence artificielle garantit en effet la *variabilité* du système selon les situations environnementales et les aléas pris en compte. De plus, la *stabilité* vis-à-vis des contraintes et objectifs globaux du système est garantie par ces mêmes algorithmes, à travers leur capacité à adapter le comportement du système à la satisfaction de spécifications globales quelles que soient les conditions locales.

Afin d'obtenir un système utilisable et configurable par les artistes eux-mêmes, qui soit capable de gérer avec flexibilité un grand nombre d'items – médias audio, picturaux et vidéos – et un grand nombre de situations, le logiciel créé nécessite également une grande modularité et une approche intuitive de la configuration. Déjà utilisés dans plusieurs applications artistiques auparavant (Cohen, 1988 ; Hutzler *et al.*, 1997 ; Gufflet et Demazeau, 2004), les systèmes multi-agents sont une réponse satisfaisante à ces prérequis. En effet, les SMA offrent la possibilité de maintenir une organisation structurée au cours du temps, garantissant une configuration globale stable, tout en permettant des résultats locaux variables en fonction des configurations locales existantes. De plus, contrairement à la majorité des paradigmes d'IA, pour la plupart centralisés, les SMA autorisent la modularité complète du système par rapport aux objets manipulés – qui deviennent des *agents* – et la configuration intuitive du système par l'identification de situations précises sans connaissance préalable du domaine – contrairement par exemple aux systèmes experts (Shortliffe, 1976), qui nécessitent une connaissance et une compréhension du domaine pour l'expression des règles décisionnelles. Le paradigme multi-agent est également l'un des paradigmes d'IA les mieux adaptés à l'étude et à la spécification de la dynamique, étant par nature un réseau potentiellement évolutif d'entités, tandis que les IA centralisées ne prennent en général en compte la dynamique qu'à travers la notion d'*apprentissage* (Mitchell, 1997). La notion d'apprentissage est limitée en cela qu'elle demande à nouveau à ce que les concepteurs soient capables de définir ce qui, dans le domaine, peut être appris, et comment cela doit être appris.

Ainsi, la conception logicielle liée au projet CLIC prend la forme de la création d'un SMA adapté aux contraintes logicielles et environnementales du projet – et en particulier des utilisateurs auxquels il est destiné – et capable d'être paramétré de manière simple et efficace pour maintenir une cohérence et une dynamique organisationnelle sur le long terme, et en totale autonomie.

7.1.2 Contraintes logicielles

Du point de vue logiciel, de nombreuses contraintes proviennent de l'application elle-même et de l'environnement logiciel associé. En effet, le SMA produit vise à être intégré comme

contrôleur logiciel (cf. Figure 7.2) au sein d'une plateforme de programmation graphique multimédia nommée Max/MSP/Jitter (Blum, 2007).

Les premières contraintes apportées par le projet relèvent donc de l'environnement d'exécution :

- Le SMA doit respecter une API externe écrite en C, permettant son inclusion comme objet de la plateforme de programmation.
- Pour respecter cette API, le programme doit être codé en C/C++.
- Afin d'être inclus à l'exécution de la plateforme, le programme doit être compilé sous forme d'une bibliothèque dynamique (.dll ou .dylib).

La plateforme d'exécution Max/MSP/Jitter étant disponible et déployée à la fois sous Windows et sous Mac OS, le logiciel doit également être compatible pour la compilation et l'exécution avec les deux systèmes d'exploitation. Pour cela, seules des bibliothèques de fonctions disponibles sous les deux OS à la fois peuvent être utilisées dans sa réalisation.

De plus, le SMA étant le module de contrôle de l'ensemble de l'exposition, il doit être capable de calculer **en temps réel** l'ensemble des sorties de contrôle. Pour cela, la complexité calculatoire du système doit être suffisamment basse, en accord avec les capacités des matériels sur lesquels le logiciel doit tourner. Pour cela, il est nécessaire de conserver une complexité calculatoire faible sur l'ensemble des mécanismes organisationnels présents dans les agents, mais aussi de garantir une complexité faible pour les *processus métiers* des agents.

Ensuite, à cause du besoin de flexibilité quant à la disponibilité des agents, leur paramétrage, mais surtout pour la gestion des entrées-sorties avec le système, requérant des temps de latence définis et une gestion algorithmique précise pour éviter les surcharges des canaux de communications, il a été fait le choix d'un SMA fondé sur un ensemble d'agents *asynchrones*. En effet, l'activation, la désactivation, la gestion des messages et des files de commandes sont grandement facilitées par l'application des principes de programmation parallèle – tels que l'utilisation de moniteurs et de sémaphores – au sein des architectures des agents.

Enfin, le logiciel étant destiné à un usage flexible orienté vers un public d'artistes, non spécialistes des outils informatiques et en particulier des concepts d'IA, sa configuration, et en particulier la configuration de la *population* précise d'agents le constituant, doit conserver un caractère simple et fluide. Il a ainsi été fait le choix d'initialiser le système à son exécution à partir d'un fichier de configuration XML validant un schéma XSD prédéterminé. Un second fichier, séparé, a été introduit pour définir les comportements des agents (cf. section 7.1.5).

Le choix de XML a été fait, car il permet :

- de paramétrer le système sans prérequis de lecture d'une documentation complexe et sans outil spécifique (contrairement aux formats binaires) ;
- de créer, au besoin, des logiciels d'aide au paramétrage du système, manipulant aisément le fichier de configuration à partir de la simple connaissance du schéma associé (contrairement à d'autres formats textuels tels que JSON) ;
- d'effectuer facilement une vérification formelle de la conformité du fichier de paramètres à l'aide du schéma XSD à l'exécution.

Par ailleurs, pour les mêmes raisons de flexibilité, de simplicité et de standardisation, il a été décidé que la majorité des agents devraient être codés en interne à l'aide de programmation à base de règles. Cela permet une configuration simple par le biais des fichiers XML, en indiquant simplement les conditions, les effets et les priorités des règles voulues.

Étant donnée l'indépendance de notre modèle organisationnel vis-à-vis des architectures internes des agents, ce choix de conception n'entrave en rien la modélisation organisationnelle effectuée pour la conception du système. En effet, notre modèle permet la description et l'analyse organisationnelle indépendamment de la dimension Agent des systèmes. Le choix de la programmation à base de règles comme architecture interne des agents, choix inclus dans la dimension Agent de la description du système, n'influe donc pas sur les possibilités de conception offertes par notre modèle organisationnel.

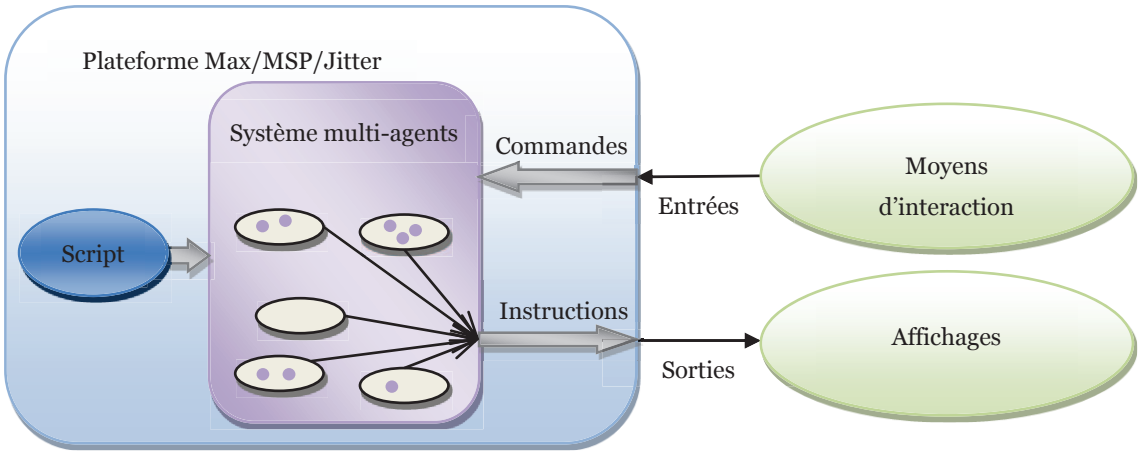


Figure 7.2 - Architecture et intégration du SMA de contrôle de l'installation

7.1.3 Modélisation organisationnelle

Dans cette section, nous cherchons à décrire, à l'aide de notre modèle, la dimension organisationnelle du système que nous concevons. Pour cela, nous suivons les différentes étapes énoncées dans notre méthodologie de conception des SMA (cf. section 6.1). Nous expliquons ci-dessous chacune des différentes étapes de conception organisationnelle.

1) Définition fonctionnelle du système

a. Définition des objectifs globaux du système

Le système conçu répond à un objectif global flexible, qui intègre son rôle de contrôleur de la plateforme multimédia et d'outil paramétrable. Il peut être formulé comme suit :

<O_G> : Le système contrôle et gère l'affichage, au cours du temps, de différents types de médias sur plusieurs périphériques de types connus, en suivant un script défini à sa création et en fonction des interactions avec l'environnement et les utilisateurs.

b. Décomposition des objectifs en arbre de tâches

Les sous-objectifs du système sont différents pour chaque exposition à laquelle il est intégré. Ils relèvent de la définition même des paramètres de l'objectif global : médias, périphériques, script, interactions extérieures. Ces objectifs sont spécifiés particulièrement en fonction des fichiers de configuration fournis par les utilisateurs à l'exécution du système. Ils se formulent de manière générique ainsi :

<O_1> : Le système gère un ensemble de médias potentiellement variable au cours du temps.

<O_2> : Le système gère l'affichage ou l'activation d'un certain nombre de périphériques connus.

<O_3> : Le système suit un script prédéfini lors de son exécution, et contraint les actions et interactions possibles en fonction de ce script.

<O_4> : Le système réagit en adaptant son comportement aux actions extérieures des visiteurs et aux paramètres d'environnements, en fonction des capteurs et moyen d'interactions proposés et des éléments de script associés.

Les **tâches** attribuées aux agents découlent toutes de ces 4 objectifs globaux. Elles distribuent dans le système la gestion de ces objectifs.

- Dérivent de <O_1> les tâches individuelles de gestion de chacun des médias présents dans le système à un instant donné : <M_x>, x étant le type de média concerné.

- Dérivent de <O_2> :
    - Les tâches de contrôle d'accès pour l'affichage d'un média sur un périphérique donné : <P_x>, x étant le type de périphérique concerné.
    - La tâche de transmission des messages en sortie vers la plateforme multimédia environnante : <ES_s>.
- Dérivent de <O_3> :
    - La tâche de contrôle du temps d'exécution pour les parties de script chronométrées : <T>.
    - Les tâches particulières dérivées des objectifs liés à chacune des parties du script : <S_x>, x étant l'objectif en question.
    - Les tâches particulières de détection de situations, dérivées des objectifs du script et des situations qui y sont à prendre en compte : <D_x>, x étant la situation concernée.
- Dérivent de <O_4> :
    - Les tâches liées aux réponses aux événements extérieurs : <E_x>, x étant le type d'événements concerné.
    - La tâche de réception des messages en entrée depuis la plateforme multimédia environnante : <ES_e>.

c. Définition des tâches annexes des agents

Le système considéré étant un système complètement coopératif, conçu et contrôlé par le même utilisateur, toutes les tâches des agents relèvent de la décomposition des objectifs globaux du système. Il n'a donc pas été nécessaire de définir de tâches annexes spécifiques aux agents.

d. Définition des rôles effectifs

Les rôles effectifs proviennent de l'identification de groupes de tâches pouvant être affectées ensemble à des agents. Des tâches énoncées pour le système, nous pouvons créer un certain nombre de rôles effectifs prédéterminés (cf. Figure 7.3). D'autres peuvent être ajoutés en fonction de la configuration du système. Les rôles identifiables sont :

- Le rôle m_x de gestionnaire de média associé aux médias de type x (tâches : <M_x>)
- Le rôle p_x de contrôleur de ressource, associé au périphérique x (tâches : <P_x>)
- Le rôle de *timer* (tâches : <T>)

- Le rôle *cs* de contrôleur de script (tâches : <S_x> ∀x)
- Le rôle d_x de déclencheur situationnel (tâches : <D_x>)
- Le rôle e_x de gestionnaire d'évènements extérieurs, associé au type d'évènement x (tâches : <E_x>)
- Le rôle *es* de gestionnaire d'entrée-sortie avec la plateforme multimédia (tâches : <ES_e> et <ES_s>).

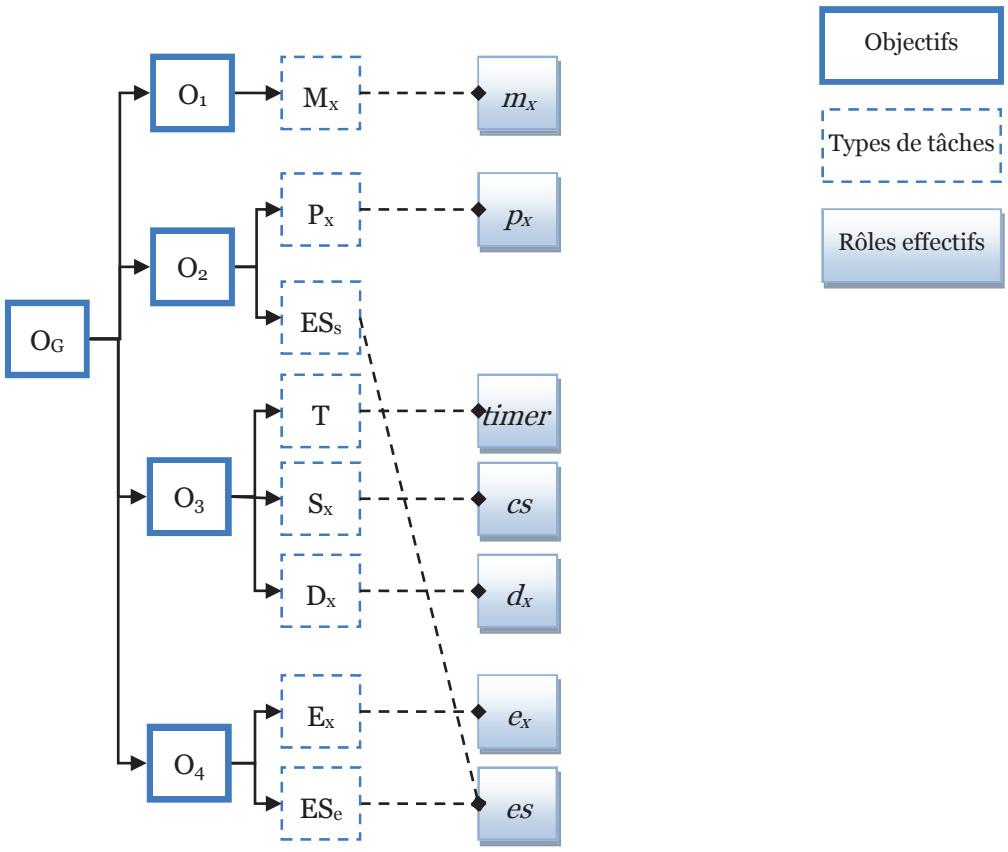


Figure 7.3 - Décomposition des objectifs en tâches et définition des rôles effectifs et normatifs pour le projet CLIC

2) Définition des types d'agents

a. Description des types d'agents et rôles normatifs souhaités

Pour répondre aux besoins énoncés dans les objectifs du système, un certain nombre de rôles statiques et de types d'agents doivent être définis. Ceux-ci permettent de répartir les tâches sur différents agents, et de répondre aux contraintes logicielles imposées sur le système. Sont ainsi définis :

- Le rôle normatif d'*agent-média [am$_x$]*, qui correspond à un agent associé à un média particulier de type *x,* et qui dérive d'un rôle normatif supérieur, l'*agent-média [am].* Il lui est confié la gestion de ce média, c'est-à-dire le rôle effectif m_x. *Ce* rôle permet de distinguer les types d'agents associés à des médias par types de média, et ceci afin de préserver la simplicité de configuration requise pour l'utilisateur (cf. section 7.1.2). Les types d'agents regroupés sous ce rôle normatif dépendent des spécificités du système.

  Dans le système réalisé, les *agents-images* sont par exemple séparés en *images diaporama*, *images réservoir* et *photos des visiteurs*.

- Le rôle normatif d'*agent-environnement [ae]*, qui correspond à un seul type d'agents éponyme, est un agent chargé de la gestion d'un certain nombre de périphériques associés. Il a le rôle effectif de contrôleur de ressource p_x pour potentiellement plusieurs types de ressources différents.

  Dans le système réalisé, par exemple, les *agents-environnements* sont chargés du contrôle de deux ressources : un écran LCD et un ensemble de haut-parleurs.

- Les rôles normatifs correspondant aux rôles effectifs *timer, cs, es* ete_x sont tous limités, par soucis de simplicité, et parce qu'ils ne nécessitent pas de modularité supplémentaire, à un même type d'agent : l'*agent-contrôleur [ac].*

- Le rôle normatif d'*agent-déclencheur [ad]*, qui correspond à un agent moniteur d'une situation particulière et permettant la gestion locale des parties de script associées à ces situations, en association avec l'*agent-contrôleur*. Plusieurs types d'agents peuvent être dérivés de ce rôle, en fonction des spécificités du script.

b. Description des autres types d'agents possibles

Le système étant entièrement spécifié avant son exécution, et n'étant pas un système *ouvert*, il n'y a pas d'autres types d'agents que ceux créés par le concepteur à considérer. Ainsi, il n'existe pas d'autres types d'agents que ceux cités au paragraphe précédent.

3) Définition des types de relations

a. Définition des relations associées aux rôles normatifs

Nous définissons les relations possibles (cf. Figure 7.4) entre agents selon le rôle joué par l'agent source d'abord et l'agent cible ensuite.

- *Agent-média* :

  - vers *agent-média* : relations spécifiques au script. Elles correspondent aux structures définies dans le script pour le réseau de médias interagissant.

- vers *agent-environnement* : relation d'*inscription*. Elle correspond à la capacité d'engager une interaction visant à affecter des périphériques à une commande d'affichage choisie par l'agent source.
- vers *agent-contrôleur* : aucune.
- vers *agent-déclencheur* : relations spécifiques au script.

- *Agent-environnement* :
  - vers *agent-média* : relation d'*enregistrement*. Elle correspond au fait que l'agent source connaisse l'agent cible, c'est-à-dire que l'agent cible soit inscrit auprès de l'agent source. Elle permet à celui-ci de lister les agents inscrits qu'il connait. Elle est la réciproque de la relation d'*inscription*.
  - vers *agent-environnement* : aucune.
  - vers *agent-contrôleur* : relation d'*intermédiaire*. Elle correspond à la possibilité des agents sources de communiquer des commandes de sortie à l'agent cible afin que celui-ci les transmette à l'environnement, après passage dans une file d'attente.
  - vers *agent-déclencheur* : relation d'*enregistrement*.
- *Agent-contrôleur* :
  - vers tous les autres types d'agents : relation de *contrôle*. Elle correspond au pouvoir d'imposer des objectifs précis à l'agent cible.
- *Agent-déclencheur* :
  - vers *agent-média* : relations spécifiques au script.
  - vers *agent-environnement* : relation d'*inscription*. Elle correspond à la capacité d'engager une interaction visant à affecter des périphériques à une commande d'affichage choisie par l'agent source.
  - vers *agent-contrôleur* : aucune.
  - vers *agent-déclencheur* : relations spécifiques au script.

b. Définition des autres relations possibles entre agents

De même que pour les tâches, le nombre de relations possibles entre agents est limité par le fait que le système conçu est fermé, c'est-à-dire limité à sa conception à un ensemble défini d'agents coopérants et d'objectifs communs. Il n'y a donc pas lieu de définir des relations

possibles entre agents qui sortent du cadre des rôles que jouent ces agents dans l'organisation.

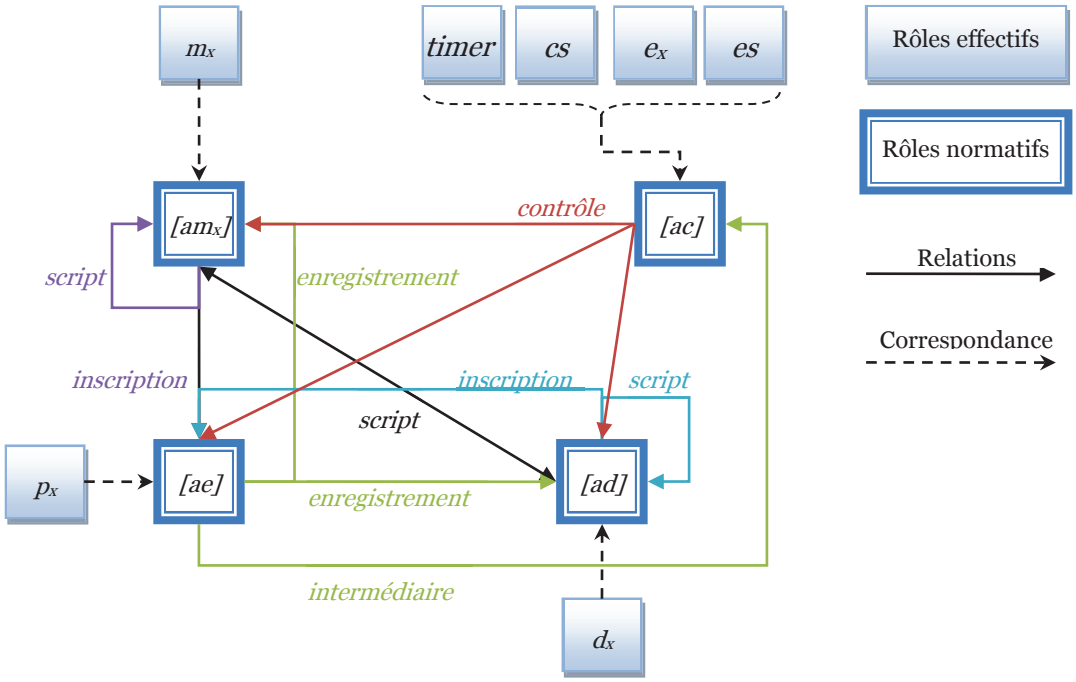


Figure 7.4 - Rôles normatifs et types de relations associées pour le projet CLIC

4) Choix des agents du système

a. Description en extension des agents connus du système

Tout d'abord, un agent a_{ac} de type *agent-contrôleur* est créé. L'agent-contrôleur est en charge de la gestion de l'ensemble du script, du timer et des entrées-sorties, et correspond donc en réalité au thread principal du programme. En raison de sa fonction de gestion du script, c'est lui qui lit les fichiers de configuration à la création du système pour créer les autres agents.

Ensuite, des agents a_{e_i} de type *agent-environnement* sont créés pour chacun des environnements d'affichage définis dans les fichiers de configuration. Ils gèrent chacun l'environnement e_i auquel ils sont associés, et transmettent les signaux de sortie à l'agent-contrôleur. Un agent a_{e_0} de type *agent-environnement* est créé et correspond à un environnement « neutre » auquel sont associés – relation d'*inscription* – les agents qui, à un moment donné, n'ont pas d'objectif et sont par conséquent « désactivés » pour les besoins du script.

Enfin, des agents a_{m_i} sont créés pour chacun des médias spécifiques de l'installation. Ces agents sont créés à la lecture du fichier de configuration[63]. Leur nombre et leurs types sont dépendants de l'installation précise.

b. Description des spécifications sur les agents

L'*agent-contrôleur* créé à l'exécution du système est le seul agent de ce type pour le système, puisqu'il est en charge du contrôle de celui-ci :

$$\forall t \; \mathrm{A}_{[ac]}(t) = \{a_{ac}\}$$

Par ailleurs, puisqu'il s'agit d'une installation dont les entrées-sorties doivent être gérées, les ressources d'affichage doivent y être représentées. Il existe donc au moins un *agent-environnement* dans le système, l'agent correspondant à l'environnement neutre (associé à aucune ressource) :

$$\forall t, \{a_{e_0}\} \subseteq A(t)$$

Enfin, tous les agents du système appartiennent à l'un des types dérivant des rôles définis ci-dessus. Autrement dit, l'ensemble des types des agents du système est inclus dans l'union des types d'agents inclus dans chacun des rôles normatifs définis :

$$T_A \subseteq [ac] \cup [ae] \cup [am] \cup [ad]$$

5) Définition des relations à $t = 0$ pour les agents connus

Les relations à $t = 0$ sont mises en place en fonction de l'organisation souhaitée et du script précisé. Elles correspondent, pour chacun des types de relations présents, à :

- la présence de relations de *contrôle* entre l'*agent-contrôleur* a_{ac} et chacun des autres agents créés ;
- la présence de relations d'*intermédiaires* entre les *agents-environnements* créés et l'*agent-contrôleur* ;
- l'*inscription* de chacun des *agents-médias* et *agents-déclencheurs* créés à l'*agent-environnement* neutre a_{e_0} ;
- l'initialisation, si besoin, de relations spécifiques au script entre chacun des *agents-médias* et des *agents-déclencheurs*.

---

[63] A cause de contraintes logicielles, en particulier de gestion d'espace mémoire et d'interaction avec la plateforme multimédia, il a été prévu que tous les agents du système soient spécifiés dans le fichier de configuration. Des mécanismes de création et de destruction d'agents à l'exécution pourraient cependant être implémentés sans modification de la formalisation organisationnelle présentée ici.

L'ensemble de ces étapes permet de définir une organisation partielle, complétée par les définitions des mécanismes organisationnels, définis sous forme de règles comportementales des agents dans le script.

6) Spécifications dynamiques sur les tâches

Les spécifications dynamiques sur les tâches reflètent la nécessité, pour le système, d'assurer certaines tâches spécifiques en continu ou à des instants donnés. Ici, ce sont les tâches assurées par l'*agent-contrôleur* qui sont nécessaires au bon fonctionnement du système (gestion du temps, du script, des entrées-sorties et des évènements extérieurs) :

$$\forall \phi \in timer \cup cs \cup es \cup \left(\bigcup_x e_x \right), \exists x \mid \mathrm{t}(x) = \phi \wedge \forall t \in \mathbb{R}^+, x \in P(t)$$

7) à 10) Identification des mécanismes organisationnels

La définition des mécanismes organisationnels dépend essentiellement des spécificités du script. La méthode diverge ici de celle proposée de manière canonique dans le chapitre précédent, car elle s'inscrit ici comme une part essentielle de la création artistique. Ainsi, elle devient une activité partagée entre le concepteur du système et l'utilisateur du système : les principes énoncés dans la description de la méthode proposée (cf. section 6.1) sont axés sur l'obtention d'une organisation validant des propriétés exprimables en termes organisationnels ; ici, l'aspect créatif et itératif des choix artistiques mène à une méthode moins formelle dans les choix de conception en raison d'une plus grande difficulté d'une expression formelle des besoins.

Les mécanismes organisationnels sont définis en fonction du script afin de garantir les propriétés voulues à la configuration du système[64]. Nous pouvons citer en exemple ces mécanismes, prévus dans le cas d'utilisation étudié et significatifs de comportements attendus :

- pour les *agents-médias* et les *agents-déclencheurs*, la limitation du voisinage aux agents associés à un même *agent-environnement*. Le mécanisme se compose de deux processus : la terminaison de toutes les relations antérieures et la création de nouvelles relations avec les agents enregistrés auprès de l'*agent-environnement*. Soit[65], pour un *agent-média* ou *agent-déclencheur* a :

[64] Il n'est pas nécessaire d'inclure des mécanismes d'insertion, puisque tous les agents du système sont ici créés à $t = 0$.

[65] Pour des raisons de clarté, nous exprimons uniquement ici les voisinages et non les champs relationnels.

$$V_a^{[ae]}(t) = \{a_{e_i}\} \Rightarrow V_a^{[am],[ad]}(t+1) = V_{a_{e_i}}^{[am],[ad]}(t) \setminus \{a\}$$

- pour les *agents-environnements*, l'obligation de n'avoir qu'un *agent-déclencheur* d'un certain type à la fois dans son voisinage. Pour implémenter cette obligation de la manière la plus fluide possible, la solution peut être de faire en sorte de déplacer l'*agent-déclencheur* qui trouve un autre agent du même type que lui dans son nouveau voisinage. Pour un agent a de type d, on voudrait :

$$V_a^d(t) \neq \emptyset \Rightarrow V_a^d(t+1) = \emptyset$$

Combinée à la règle précédente sur les environnements, deux solutions permettent d'éviter les départs simultanés des deux *agents-déclencheurs* à chaque fois :

  - l'établissement de probabilités non nulles et non unitaires de départ pour les deux agents (répétées successivement jusqu'à ce que la situation ait changé, c'est-à-dire qu'un des agents soit parti)

$$0 < P\left(V_a^d(t) \neq \emptyset \Rightarrow V_a^{[ae]}(t+1) \cap V_a^{[ae]}(t) = \emptyset\right) < 1$$

  - la négociation, par l'interaction des deux agents, résultant en l'attribution à un des deux agents en relation d'une tâche <*dep*> de déplacement ; dès lors, seul l'agent ayant cette tâche se déplace, en suivant un mécanisme téléonomique :

$$< dep >: V_a^d(t) \neq \emptyset \Rightarrow V_a^{[ae]}(t+1) \cap V_a^{[ae]}(t) = \emptyset$$

La dynamique engendrée par les mécanismes choisis est donc très variable d'une configuration du système à l'autre, et le modèle proposé peut de plus, notamment dans les situations gérées par de l'aléatoire, être très efficace pour prédire l'évolution organisationnelle du système. Mais, il permet déjà ici d'exprimer les propriétés voulues et de les convertir en mécanismes exploitables pour le code – ou ici la configuration du SMA – sans passer par la modélisation markovienne, puisque les propriétés exprimées ne sont pas probabilistes mais exactes.

7.1.4 Exploitation du logiciel et exposition réalisée

L'exposition réalisée par les artistes comprend deux types de médias : des images et des sons. Les images proviennent d'une part de photographies du campus universitaire récupérées au préalable, et d'autre part de photographies des visiteurs prises au moyen d'une webcam placée à l'entrée de l'exposition. L'installation elle-même (cf. Figure 7.5) est composée de trois ordinateurs, dédiés l'un à l'exécution du SMA, les deux autres à la gestion des affichages, d'un buzzer et d'un clavier d'ordinateur – comme moyens d'interactions avec les visiteurs – d'une webcam, d'une mini-imprimante, de quatre haut-parleurs et de six

écrans LCD – l'un d'entre eux affichant une représentation du SMA en activité, les autres affichant les médias.

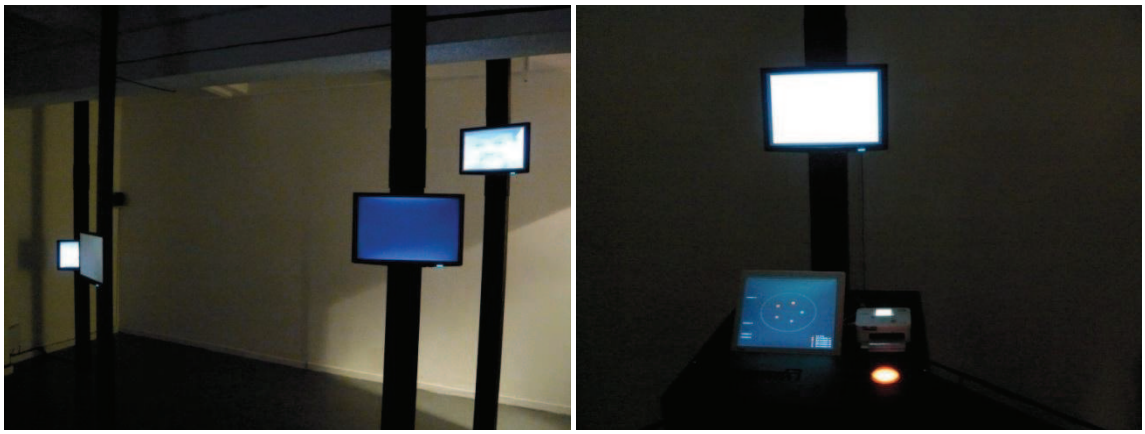


Figure 7.5 - Images de l'installation - matériel utilisé

La configuration du système correspondant à cette installation inclut tout d'abord 30 types d'agents, dont l'agent-contrôleur, les agents-environnements, et 2 types d'agents-déclencheurs, correspondant à deux parties distinctes du script. Le nombre total d'agents avoisine 500, majoritairement des *agents-médias* liés à des images, mais aussi des *agents-médias* liés à des sons, des *agents-déclencheurs* et bien sûr l'*agent-contrôleur* ; les périphériques d'affichage sont quant à eux modélisés par 5 *agents-environnements*, correspondant chacun à un écran LCD particulier et aux 4 haut-parleurs – mis en commun.

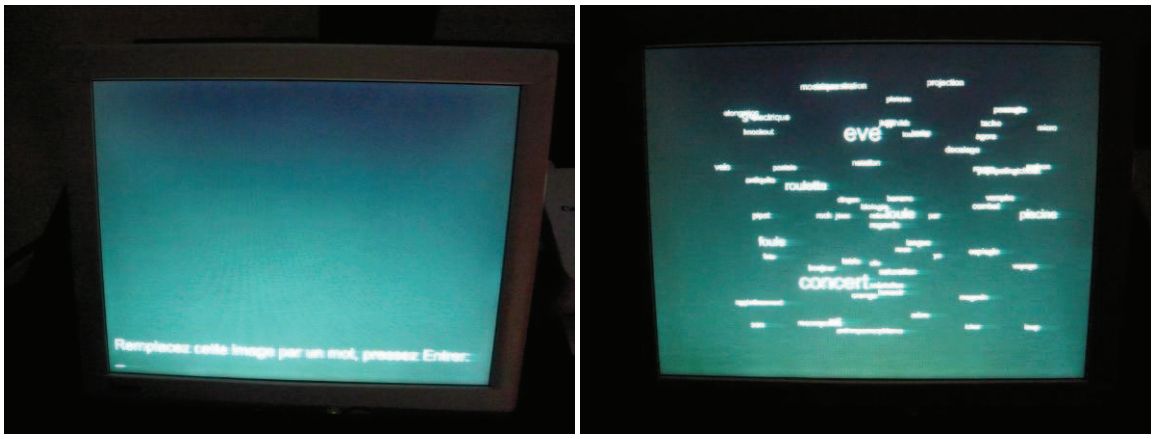


Figure 7.6 - Affichage de la séquence 4 du script

Le script est séparé en 4 parties distinctes, et inclut la définition de 30 règles de comportement pour les agents – les mécanismes organisationnels. Il suit un cycle dans lequel la situation est globalement remise à zéro à chaque début de cycle – mis à part la mémoire

des agents – et qui compte donc plus sur l'aléatoire que sur la continuité dans le temps[66]. Brièvement décrit, le script procède comme suit :

1) Deux agents (parmi environ 40) représentant des groupes d'images interagissent avec un déclencheur pour être sélectionnés. Les agents liés aux images des groupes sélectionnés interagissent ensuite par paires pour, sur chaque écran, être affichées simultanément au moyen de transformations texturales sur les images.

2) Après un temps déterminé, des sons interagissent avec un ensemble d'images – les photos des visiteurs – pour être affichées – et joués – sur chaque affichage brièvement, afin de créer une impression d'explosion progressivement accélérée.

3) Après un temps déterminé, des agents représentant des groupes d'images interagissent pour démarrer des diaporamas sur les différents affichages.

4) Après l'appui sur le buzzer, un agent-image est sélectionné aléatoirement – mais sans répétition au cours du temps[67] – parmi environ 200 agents par un agent-déclencheur. Il est affiché, et le public est invité à entrer un mot évoquant cette image tandis que l'image fait un fondu vers sa couleur moyenne unie. Puis l'ensemble des mots entrés dans le passé est affiché (cf. Figure 7.6), tandis qu'est imprimée une carte de la couleur unie et comportant le mot entré.

Dans un premier temps, cette réalisation nous a permis de constater la possibilité, pour les artistes, de comprendre, manipuler et paramétrer le logiciel créé. Cependant, il est apparu clairement qu'en l'absence d'une interface complémentaire d'assistance au paramétrage, les spécificités techniques – manipulation de XML, programmation à base de règles – et la difficulté conceptuelle de l'appropriation des notions de dynamique organisationnelle requièrent l'assistance de personnes qualifiées pour fournir les éléments nécessaires aux artistes pour manipuler correctement le logiciel et sa configuration. Nous croyons qu'un logiciel complémentaire d'aide à la configuration pourrait résoudre cette difficulté en fournissant aux utilisateurs des explications et des représentations plus manipulables des données et des concepts sous-jacents.

---

[66] Il s'agit ici d'un choix artistique et non technique, puisque, comme nous l'avons décrit dans la conception du système, les deux orientations sont indifféremment possibles dans le SMA créé.

[67] Il est intéressant de noter qu'il s'agit ainsi du seul mécanisme *hétérogène* utilisé dans cette configuration du logiciel, car l'état de l'agent – ici, la mémoire des associations agent-mot entré – est utilisé pour le choix du partenaire d'interaction par *l'agent-déclencheur*.

Figure 7.7 - Images de l'installation en activité

Dans un deuxième temps, il nous a été permis de constater que le programme répondait aux exigences logicielles et techniques imposées : intégré à la plateforme Max/MSP/Jitter, il a tourné pendant plusieurs semaines de suite sans problèmes significatifs ni ralentissements (cf. Figure 7.7), manipulant plus de 500 agents – 200 actifs en simultané maximum – sur un ordinateur raisonnablement puissant – 3GHz quad-core. Ainsi, bien que probablement limité – si exécuté sur un unique ordinateur et non distribué – par sa gestion parallèle des nombreux médias présents dans une exposition de ce type, il semble que le système soit capable d'assurer un traitement efficace d'un nombre d'agents suffisants pour les besoins de la plupart des artistes souhaitant mettre en place des installations de ce type.

Enfin, et dans un autre registre, bien que les artistes aient souhaité mettre la technique du multi-agent au centre de la sémantique de leur projet – reliée bien évidemment au thème *butinage et essaimage* de la biennale – il semble que l'intérêt, le fonctionnement, et plus généralement les principes, des SMA et de leur utilisation dans ce cadre aient été difficiles à appréhender pour le public.

7.2 Analyse de système : Projet MASNET

Le projet MASNET – *Middle Age Social NETworks* – (Lacomme *et al.*, 2011) est un projet multidisciplinaire impliquant l'équipe MAGMA du LIG, l'équipe SMAC de l'IRIT, le laboratoire TRACES et l'IMT, de l'université du MIRAIL. Il s'agit de l'extension de l'étude d'un réseau social médiéval extrait d'une base de données de documents notariés analysés et numérisés. L'extension de cette étude porte sur la dynamique du réseau, sa structure statique et certains de ses invariants ayant déjà été étudiés dans un précédent projet : GRAPHCOMP (Boulet, 2008).

7.2.1 Contexte applicatif

L'étude dirigée par les historiens vise à reconstruire et à étudier les réseaux sociaux de la société paysanne médiévale (Hautefeuille et Jouve, 2004). Elle est fondée sur l'analyse d'un nombre important de documents notariés couvrant les nombreux contrats (plus de 8000) effectués et archivés dans une zone géographique restreinte du sud-ouest de la France. Ces documents (cf. Figure 7.8), conservés pour la plupart aux archives départementales du Lot, font actes de différents types de contrats (contrats agraires, testaments, mariages ou hommages féodaux), et cela tout au long de trois siècles (de 1250 à 1550).

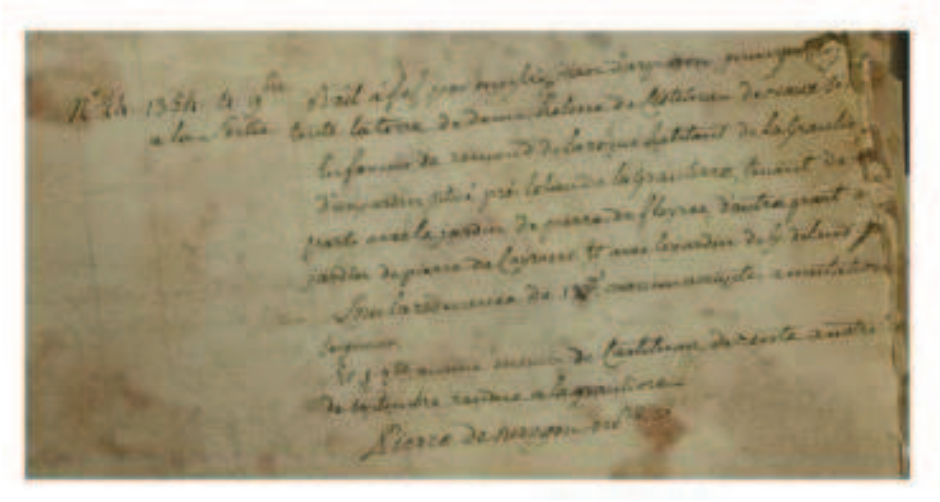


Figure 7.8 - Photographie d'un acte notarié des archives utilisées dans le projet

L'hypothèse de travail historique est que ces contrats sont le marqueur de phénomènes relationnels entre différentes strates sociales de la société rurale médiévale : paysans, église, seigneurs,... Ainsi, de ces documents, peut être extrait un graphe de relations représentant la part visible de l'organisation de la société rurale médiévale, et cela pour chaque instant donné. L'intérêt de l'étude par cette approche réside dans le fait que les paysans, représentant environ 90 à 95% de la population de l'époque, ont laissé fort peu d'écrits permettant de pénétrer l'organisation et l'histoire de cette couche de la société ; ces actes notariés, axés sur la population paysanne, forment donc un ensemble de documents historiques rare. Pour faciliter l'extraction et l'analyse des réseaux sociaux à partir des archives historiques, une base de données a été créée (cf. Figure 7.9). Elle contient 8725 individus, répartis sur 250 ans et situés géographiquement sur différentes parcelles de terrain. La population de base – pour un instant donné – est estimée entre un maximum de 3000 individus (en 1330), et un minimum de 1500 individus (en 1420).

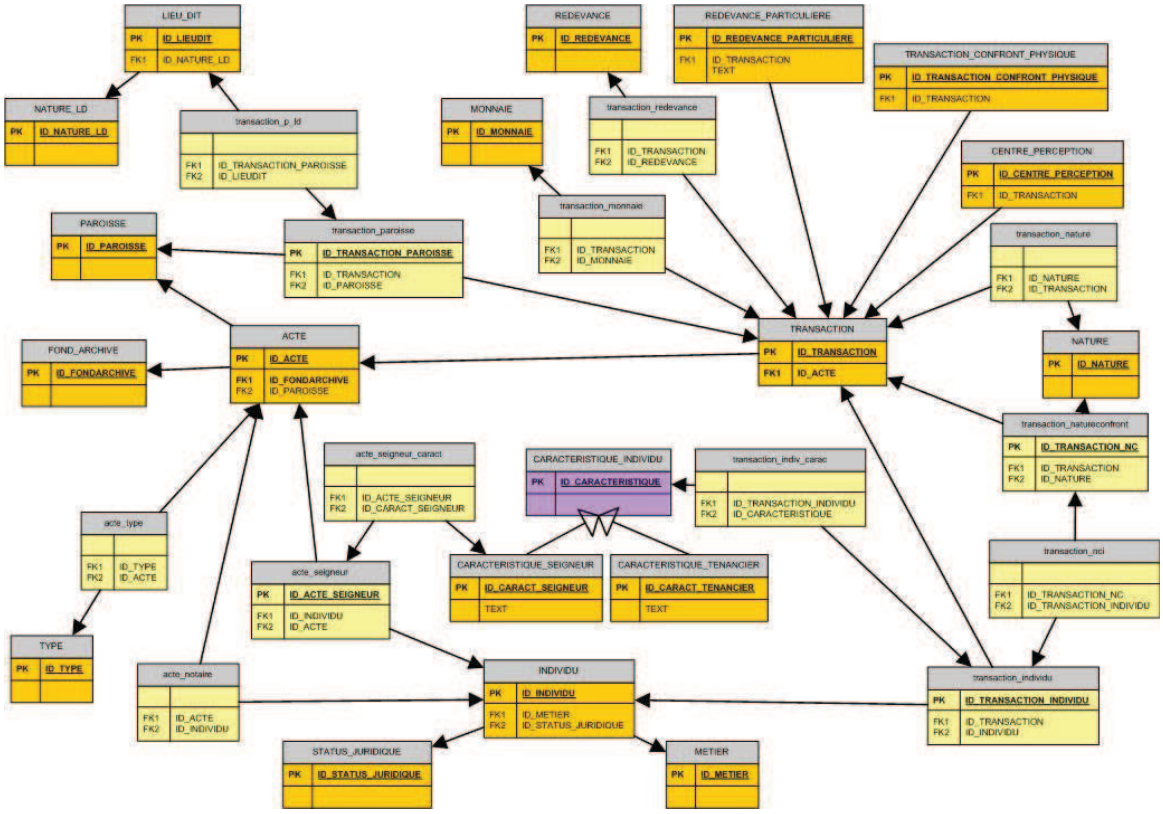


Figure 7.9 - Modèle de la base de données utilisée dans les projets GRAPHCOMP et MASNET

Une première analyse statique a été effectuée sur ce réseau social induit dans le projet GRAPHCOMP (Boulet *et al.*, 2007). Pour cela, un graphe représentant le réseau social a été construit (cf. Figure 7.10) pour chaque instant dans le temps – par interpolation sur la date des contrats, les individus étant considérés vivre 30 ans, l'espérance de vie de l'époque pour cette population. Les nœuds en sont les individus supposés vivants à l'instant considéré. Les nœuds sont reliés par une arête si et seulement si les deux individus représentés apparaissent dans un même contrat ou bien dans deux contrats différents séparés de moins de 15 ans et reliés au même seigneur ou notaire. Des études réalisées (Boulet et Jouve, 2006 ; Villa et Boulet, 2007 ; Villa *et al.*, 2008) sur des territoires plus ou moins restreints et portant sur le *degré* (nombre de voisins) et la *centralité* (fréquence des chemins les plus courts entre deux nœuds quelconques passant par le nœud concerné) des nœuds ont montré plusieurs propriétés du réseau :

- une distribution en *petit monde* ;
- une faible connectivité globale et une forte connectivité locale ;
- l'existence d'un *rich-club*, ensemble très fortement connexe, principalement constitué de non-paysans ;

- l'existence de communautés paysannes, liées aux répartitions géographiques ;
- la connexion des communautés entre elles par des *individus relais*, identifiés comme des individus à la fois paysans et marchands.

L'objectif de cette seconde étude se focalise donc sur l'analyse de la dynamique du réseau social, de son évolution au cours du temps et de l'influence des facteurs extérieurs – tels que la guerre – sur sa structure. Deux niveaux différents peuvent être étudiés dans cette optique :

1) l'évolution du réseau, et l'existence des différentes structures identifiées par l'analyse statique dans différents contextes – paix, guerre, population croissante ou décroissante, etc.
2) l'évolution d'une sous partie du réseau, telle qu'une famille donnée, et la dynamique de son mouvement au sein des structures établies.

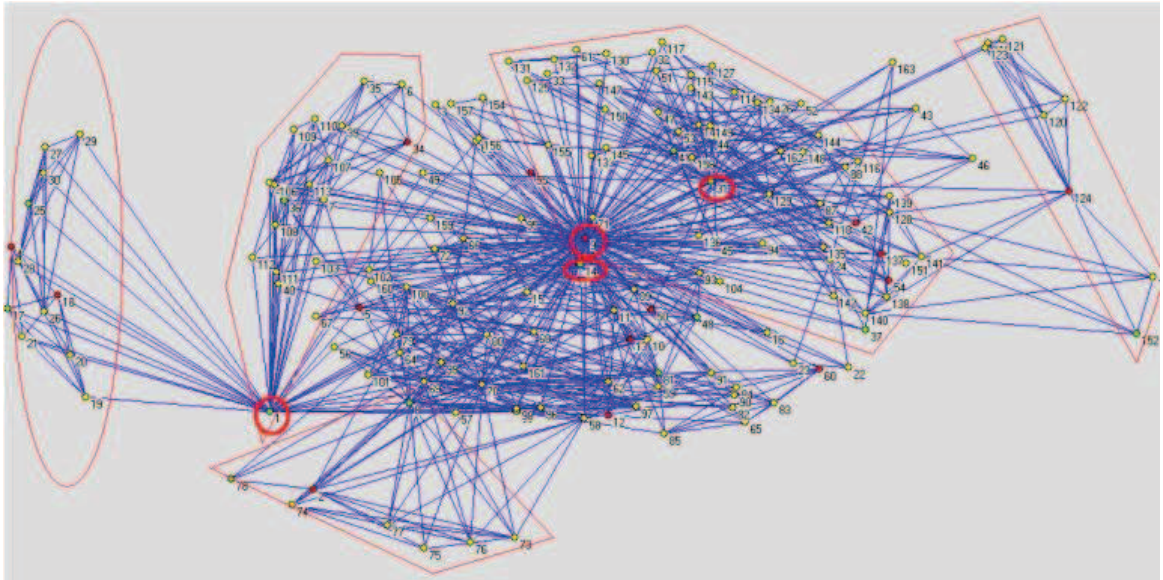


Figure 7.10 - Représentation du réseau social induit par extraction de la base de données à une date choisie

7.2.2 Approche

L'étude réalisée dans le cadre du projet MASNET se concentre donc sur la dynamique du système, dans ses aspects globaux et locaux. Pour cela, il s'agit d'analyser l'évolution et la dynamique d'un système simulé, régit par des règles comportementales locales historiquement plausibles, et de comparer cette dynamique à celle du réseau social réel, induit depuis les données historiques. Bien que d'autres méthodes, telles que les automates cellulaires, puissent être envisagées, l'incomplétude des données manipulées et les incertitudes sur ces données (existence d'ambigüité sur les noms, par exemple) oriente

l'approche vers un paradigme plus flexible, capable d'apporter des résultats sans requérir une connaissance complète et parfaite de la situation explorée. De plus, aucune autorité supérieure n'existant historiquement pour « diriger » le réseau social, celui-ci dérive nécessairement des interactions locales entre individus ; cet aspect émergent est parfaitement exploité à travers la modélisation par les SMA, qui fonctionnent naturellement selon le même principe.

L'approche consiste donc à créer un SMA à partir d'un modèle organisationnel permettant d'exprimer l'ensemble des propriétés locales significatives :

- métiers et statut des individus : paysan libre, serf, marchand, notaire, seigneur,...
- nature des relations : familiale, contrat agraire, mariage, vassalité,...

L'évolution du SMA est guidée par un ensemble de règles comportementales fondées sur des hypothèses historiquement plausibles. Par exemple : *deux paysans voisins se connaissent*.

Enfin, les propriétés étudiées du réseau sont les propriétés déjà étudiées du point de vue statique : répartition des *degrés* et de la *centralité* des nœuds – c'est-à-dire des agents.

L'analyse consiste alors à réaliser des hypothèses sur les règles comportementales locales possibles, et à ensuite analyser les effets de ces règles sur le système à travers une méthode parmi plusieurs possibles :

1) la création d'un système simulé à partir des données historiques sur les **individus** et les **relations** à un instant donné, puis la comparaison de l'évolution du système simulé avec l'évolution du réseau réel après l'instant choisi ;
2) la création d'un système simulé à partir des données historiques uniquement sur les **individus** à un instant donné, et la comparaison de l'auto-organisation du système simulé grâce aux règles locales avec l'état réel du réseau à l'instant choisi ;
3) la création d'un système simulé à partir des données historiques sur une **famille** particulière (Hautefeuille, 2006), et aléatoirement pour le reste, et la comparaison de l'évolution de la famille considérée entre le réseau simulé et le réseau réel.

A partir des résultats obtenus par comparaison entre un réseau simulé à partir de règles comportementales choisies et le réseau historique extrait de la base de données, il est ensuite possible de corriger l'ensemble des règles comportementales afin de sélectionner les règles produisant des effets conformes à la vérité historique, d'éliminer les règles incohérentes et d'affiner les éventuels paramètres des règles afin d'ajuster au mieux la conformité du réseau simulé au réseau réel. C'est l'analyse *a posteriori* par les historiens des sélections de règles et des paramètres optimaux découverts qui donnera sens à l'analyse dynamique du système.

7.2.3 Modélisation organisationnelle

Dans le cadre de cette application, nous ne nous référons pas à la méthode proposée pour la conception de système (cf. section 6.1), qui est uniquement adaptée au domaine de la conception de systèmes disposant d'objectifs exprimables : nous sommes ici dans l'analyse d'un système préexistant par recréation et imitation. Nous partons plutôt des éléments connus du système et de son fonctionnement supposé pour bâtir le SMA d'analyse. Nous voyons donc ici le deuxième domaine d'utilisation possible pour notre modèle : l'analyse de SMA.

La construction du SMA consiste en la définition des éléments qui le composent (cf. Figure 7.11) et en l'élaboration des spécifications sur ces éléments. Ainsi, nous cherchons d'abord à définir les types d'objets rencontrés, puis les objets eux-mêmes, et enfin les spécifications éventuelles à ajouter pour contraindre le système dans le fonctionnement souhaité.

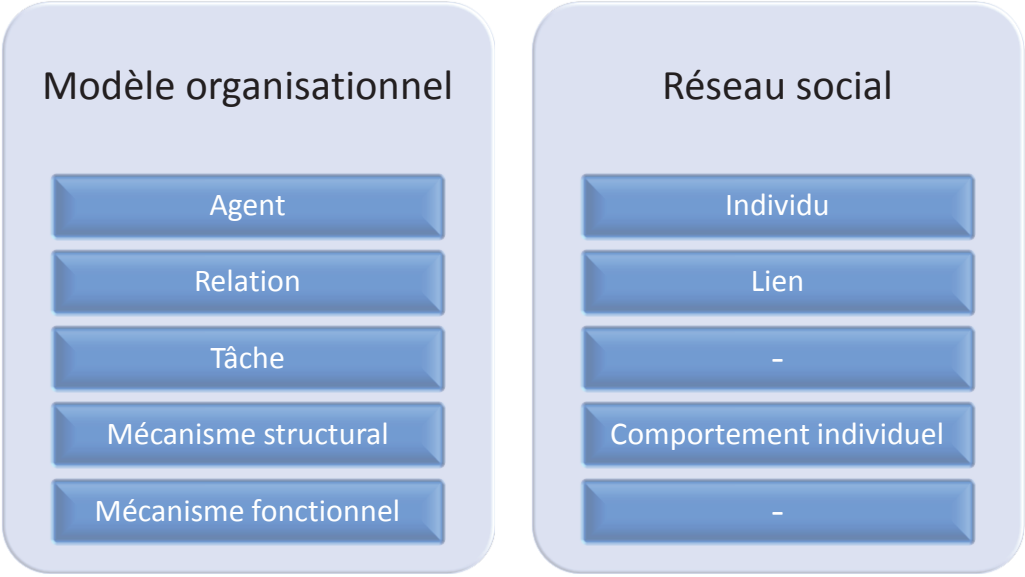


Figure 7.11 - Correspondance entre notions organisationnelles et éléments du réseau social étudié

- **Types d'agents**

Les agents doivent représenter les acteurs du système. Le système étant ici un réseau social en évolution, les agents correspondent aux individus participant à ce réseau. Nous distinguons les individus par les informations dont nous disposons sur eux. Il s'agit de leur métier ou statut social.

Il existe ainsi un type d'agent pour chaque métier répertorié dans la base de données : α_p = paysans libres, α_n = notaires, α_{ser} = sergents, α_m = marchands, etc. Il est évidemment

possible, pour des raisons pratiques, de regrouper les types présentés en différentes catégories, qui seront donc des rôles normatifs et correspondront à des attitudes communes lorsque le niveau de précision requis sur les comportements est suffisamment faible pour ne pas nécessiter une modélisation séparée, métier par métier. Il sera alors par exemple possible de regrouper en rôles normatifs : *[p]* pour les paysans, *[n]* pour les notaires, *[m]* pour les marchands, *[s]* pour les seigneurs, *[r]* pour les ecclésiastiques et *[a]* pour les autres professions.

- **Types de relations**

Nous identifions les relations existantes entre les individus en fonction de leur sémantique et des interactions possibles entre les individus liés. Les différentes relations découlent de plusieurs procédés d'identification de leur existence historique à partir des bases de données. Chaque relation représente un lien social entre deux individus ; elle n'est pas directement liée à l'existence, dans la base de données, d'un acte liant ces deux personnes, pour deux raisons : la première est que les données historiques sont incomplètes et parcellaires, et ne dénotent pas de l'entièreté du réseau social ; la seconde raison, plus fondamentale, est que l'existence d'une relation n'induit pas nécessairement l'existence d'une interaction notariée (deux voisins peuvent se connaître, échanger des produits et des services, sans jamais avoir procédé à un acte notarié). Nous ne déduisons donc pas les types de relations des types d'actes répertoriés, mais de leurs causes possibles.

On répertorie ainsi des relations :

- familiales γ_f entre frères, sœurs et cousins d'une même génération,
- maritales γ_m entre époux,
- parentales γ_p entre les enfants et leurs parents,
- communautaires γ_c entre les voisins et habitants d'une même paroisse,
- professionnelles γ_y entre commerçants et clients et entre employeurs et employés,
- notariales γ_n entre un notaire et les personnes se référant à lui pour leurs actes,
- seigneuriales γ_s entre un seigneur et ceux qui dépendent de lui ou le reconnaissent comme tel.

Dans ce cas, les interactions étant seulement preuves de l'existence d'une relation sociale, sans précision sur cette relation, nous simplifions l'analyse en considérant les relations sociales comme réciproques. Pour cela, à chaque type de relation γ définie ci-dessus, que nous appelons de forme directe, et qui lie un agent source à un agent cible, nous associons un type de relation $\bar{\gamma}$ de une forme réciproque qui les lie dans l'autre sens. Pour chaque

occurrence d'une relation de forme directe, nous imposons que la relation réciproque existe également (de même, si la relation réciproque existe, alors la relation de forme direct aussi) :

$$r = (a, b, \gamma) \in R(t) \Rightarrow \exists r' \in R(t) \mid r' = (b, a, \bar{\gamma})$$

Exemple : une relation maritale γ_m lie l'agent source a (le mari) à sa femme l'agent cible b. Alors, nécessairement, une relation maritale inverse $\overline{\gamma_m}$ lie la femme b à son mari a.

- **Types de tâches**

Étant donnés l'aspect spéculatif de la simulation analytique effectuée et l'absence de données historiques sur ces informations, nous ne pouvons rien supposer des objectifs partagés ou individuels des acteurs du réseau social étudié. Nous ne pouvons donc pas attribuer des tâches particulières à nos agents sans risque inconsidéré d'erreur : nous n'avons aucun moyen fiable, à partir des informations de la base de données, pour définir les objectifs des acteurs du système. De plus, la précision des descriptions – approximation des périodes de vie des individus, incomplétude des données – est incompatible avec une description fonctionnelle précise, qui demande de prendre en compte l'évolution des objectifs personnels au cours de la vie des individus. Par ailleurs, l'aspect étudié du système est son aspect structurel, puisqu'il s'agit de modéliser et de comprendre les structures identifiées dans le réseau et leurs dynamiques. Cela implique que :

- Nous ne définissons pas de tâches ni de types de tâches dans le système. L'étude s'effectuera uniquement au niveau structurel.
- Les mécanismes structuraux définis lors de l'étude seront nécessairement **réactifs** (*i.e.* indépendant du *champ d'action* de l'agent, cf. section 4.2.1).

- **Mécanismes organisationnels**

La spécification des mécanismes organisationnels est un élément complexe de cette analyse, dans le sens où les mécanismes doivent non seulement nous permettre d'émuler ou de simuler le comportement global du réseau social historique, mais aussi faire sens dans ce contexte. Ainsi, si nous pouvons agir sur les paramètres des mécanismes pour affiner le comportement du système (cf. Figure 7.12), il nous faut avant tout définir des mécanismes qui soient compréhensibles d'un point de vue historique.

L'approche suivie est de définir un grand nombre de mécanismes possibles, puis d'étudier leur impact sur le réseau en faisant varier éventuellement des paramètres internes de ces mécanismes, mais surtout et principalement leur probabilité d'occurrence. L'étude de leur impact peut ensuite s'effectuer dans diverses conditions de réseaux, tant du point de vue de sa configuration initiale que de sa dynamique existentielle, à travers les paramètres de

natalité et de mortalité, notamment ; ceci permet en effet d'étudier le réseau dans différentes circonstances, en particulier en période de guerre.

L'étude des impacts sur le réseau nécessite cependant une autre étape, qu'il est important de préciser. Il s'agit du passage du réseau social réel au réseau social apparent, par la génération de pseudo-interactions et de leurs traces historiques. Il est ainsi nécessaire de définir en plus des probabilités d'occurrence de chaque interaction particulière pour chaque type de relation sociale ; il est également nécessaire de définir la probabilité que chaque type d'interaction soit consigné en acte notarié. C'est le passage à travers ces deux opérations de projection qui donne, à partir du réseau social simulé, l'image de réseau social à comparer avec le réseau historique.

Pour définir des mécanismes organisationnels, nous partons donc de faits sociaux considérés comme plausibles, puis nous les modélisons sous forme de comportements organisationnels pour les individus – les agents – composant le système.

On peut prendre pour exemple les mécanismes suivants :

- (M1) Un enfant possède les relations communautaires de ses parents.

$$e \xrightarrow{\gamma_p} p \Rightarrow V_e^{\gamma_c}(t+1) = V_p^{\gamma_c}(t)$$

- (M2) Un paysan connait les autres paysans du village.

$$village(a)(t) = village(b)(t) \Rightarrow b \in V_a^{\gamma_c}(t+1)$$

- (M3) Un paysan connait les marchands qui passent dans son village.

$$village(a)(t) \subseteq zone(m)(t) \Rightarrow m \in V_a^{\gamma_y}(t+1)$$

- (M4) Un paysan connait les notaires dont il dépend.

$$village(a)(t) \subseteq zone(n)(t) \Rightarrow n \in V_a^{\gamma_n}(t+1)$$

- (M5) Un paysan connait le seigneur dont il dépend.

$$village(a)(t) \subseteq terres(s)(t) \Rightarrow s \in V_a^{\gamma_s}(t+1)$$

- (I1) Tout individu entrant dans le système (naissance) est le fils de deux individus du système.

$$V_e(t-1) = \emptyset \Rightarrow e \xrightarrow{\gamma_p} p_1 \wedge e \xrightarrow{\gamma_p} p_2$$

Chacun de ces mécanismes peut être associé à une probabilité d'occurrence. Par exemple, on peut décider que tout paysan est lié à son seigneur (M5 : 100%), mais qu'un paysan n'a qu'une chance sur 5 de connaître, pour chaque période de temps, un nouveau voisin (M2 : 20%). C'est principalement la variation de l'ensemble des probabilités d'occurrence qui nous permet d'agir sur la structure et la dynamique du système et du réseau observé. Et c'est par

les observations qu'il nous est donné d'apporter de nouvelles données et une nouvelle compréhension possible à l'analyse historique.

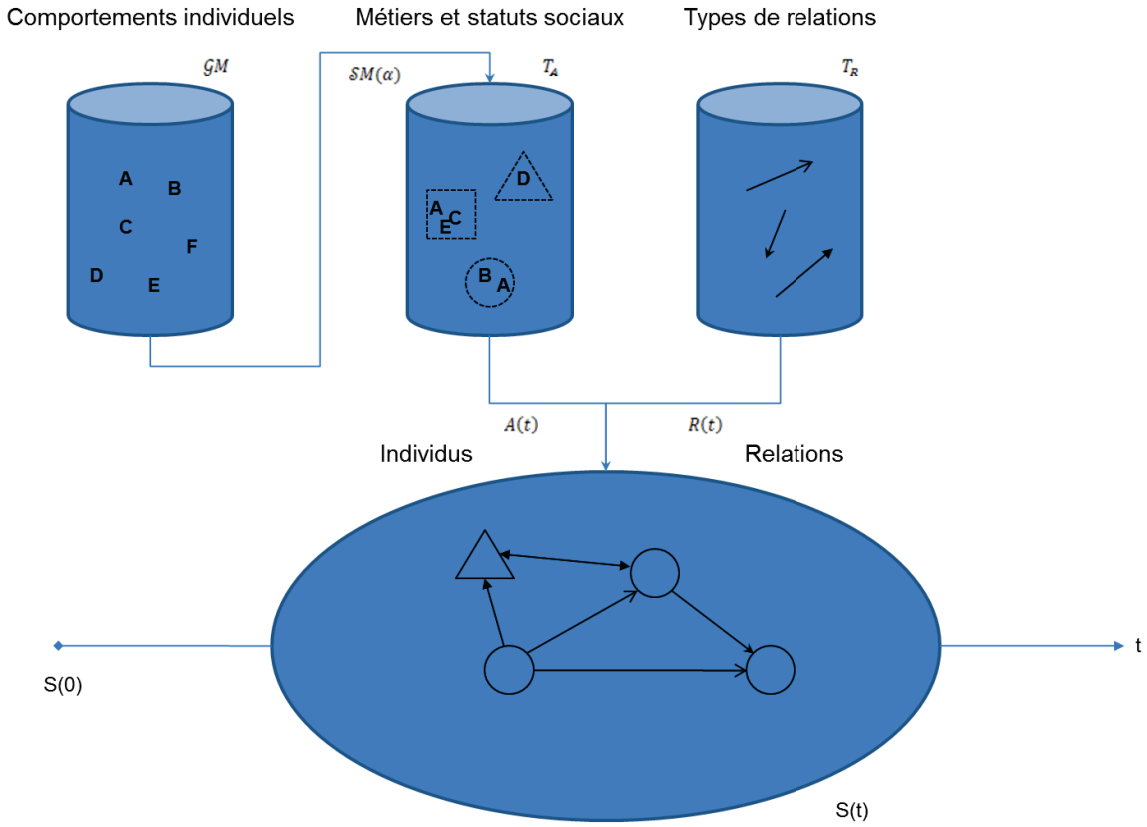


Figure 7.12 - Instanciation du modèle organisationnel à l'analyse du réseau social du projet MASNET

7.2.4 Résultats et perspectives

La conception d'un SMA approprié, destiné à la recréation et la simulation d'un réseau historique observé et de sa dynamique, nous a permis de fournir un outil théorique et technique approprié à l'analyse de ces données historiques et à une éventuelle meilleure compréhension de leur évolution. L'approche choisie est mixte : entre prédiction théorique, possible à travers l'utilisation des chaînes de Markov à partir des probabilités attribuées à chaque mécanisme organisationnel, et simulation, possible à travers l'implémentation du SMA ainsi défini sous une plateforme de simulation multi-agent. Elle permet la mise en place d'un outil efficace dans ses capacités prédictives et analytiques vis-à-vis d'un réseau connu partiellement et globalement. Elle permet ainsi de rendre possible l'explication des phénomènes globaux à partir de réponses locales, tant dans l'étude de l'évolution globale du réseau, d'abord en temps de paix, ensuite en temps de guerre, et dans la persistance de

structures identifiées, que dans l'étude de l'évolution des différentes familles et de leur place dans la société.

La complexité des possibilités de comportements historiquement plausibles, combinée à un manque d'opportunités d'interactions entre historiens et informaticiens pour cerner, discuter et préciser ces comportements possibles, a rendu pour l'instant difficile l'énumération des différents mécanismes plausibles, et donc des différents axes d'analyse intéressants. En effet, en l'absence de contraintes supplémentaires sur les mécanismes locaux, les degrés de libertés sur le paramétrage de ceux-ci sont tellement importants qu'il n'est pas difficile de recréer un réseau stable, adoptant, du point de vue des indicateurs utilisés – degré, centralité – des caractéristiques similaires au réseau historique. Il est évident cependant qu'une telle liberté est factice du point de vue de sa sémantique historique, et qu'un recadrage des mécanismes locaux est donc nécessaire, en collaboration avec les historiens, pour disposer d'un modèle de base crédible sur lequel expérimenter à travers les paramètres disponibles. A la suite de cette étape, il sera possible et fructueux d'implémenter réellement le SMA conçu, sur une plateforme de simulation multi-agent adaptée, telle que Repast (North *et al.*, 2005), et d'utiliser ce système pour étudier des cas précis, à partir des données historiques ou de données fictives modifiées.

Chapitre 8. Conclusion

Dans cette thèse, nous avons tenté d'aborder l'un des défis actuels du domaine des systèmes multi-agents : l'expression de la dynamique organisationnelle des systèmes. Cette problématique se situe à l'intersection des difficultés de représentation de l'émergence dans les systèmes complexes et de la formalisation conceptuelle et analytique des SMA. Son importance réside dans la place vitale de la notion d'organisation dans la capacité expressive des modèles descriptifs des SMA, combinée au besoin de plus en plus important de formaliser les modèles de représentation en vue d'assurer de manière fiable et communicable des propriétés générales, dynamiques ou invariantes sur les systèmes créés ou étudiés.

Pour répondre à cette problématique, nous avons proposé un nouveau modèle générique descriptif des organisations des SMA, orienté vers l'expression de la dynamique organisationnelle. Ce modèle tente de répondre au problème théorique et technique soulevé : l'expression et la formalisation, en vue de l'analyse, de la comparaison et de la conception, des organisations des SMA et de leurs propriétés statiques et dynamiques.

Nous présentons dans ce chapitre nos conclusions sur l'étendue et la réussite de notre travail, puis nous présentons le positionnement de celui-ci dans un cadre plus général et apportons des perspectives quant à son extension.

8.1 Modèle théorique

Dans ce travail, nous nous sommes fondés sur le hiatus dans l'état de l'art entre d'une part les modèles de représentation usuels des organisations dans les SMA, orientés vers la spécification des invariants et des propriétés statiques, et d'autre part la grande diversité de modèles locaux de mécanismes organisationnels, destinés au contrôle de la dynamique organisationnelle des systèmes. De cette opposition, nous avons déduit la nécessité d'introduire un nouveau modèle organisationnel pour les SMA, fondé sur une approche favorisant l'énonciation et la formalisation de la dynamique.

Nous nous sommes pour ce faire fondés sur la dissociation de la description des SMA en multiples dimensions en suivant la méthode VOYELLES (Demazeau, 1996), et sur l'indépendance nécessaire entre ces dimensions descriptives, pour assurer un **modèle générique**, capable de représenter tous types de systèmes et d'inclure tous types d'agents. Nous avons également pris comme hypothèse l'indépendance entre l'organisation et la population d'agents qui l'implémente, comme proposée par le modèle PopOrg (Demazeau et Costa, 1996), et cela afin de garantir la possibilité de définir, spécifier et comparer des

organisations à haut niveau, sans nécessité de s'attacher spécifiquement à une instanciation précise de celles-ci.

Par ailleurs, afin de garantir la possibilité de **spécifier et de calculer la dynamique**, tant locale que globale, pour le système, nous avons retenu une approche fondée sur le principe de l'émergence, qui propose de construire la description du système en partant des propriétés locales pour aller vers les propriétés globales.

C'est à partir de ces spécificités d'approches que nous avons déduit que notre modèle devait reposer sur un ensemble d'éléments bas-niveau, réduit à l'expression des propriétés essentielles d'une organisation : sa structure, sa fonction, la dynamique de celles-ci. Ainsi, notre modèle s'appuie sur quatre éléments atomiques : l'**agent**, la **relation**, la **tâche** et le **mécanisme organisationnel**. A partir de ces éléments de base, nous avons explicité la possibilité de spécifier les organisations par adjonction de **contraintes** sur des ensembles significatifs de ces éléments, les **réservoirs**. Les ensembles de contraintes ainsi construits permettent d'expliciter et de formaliser l'ensemble des possibilités de configuration des systèmes relevant de l'organisation étudiée. Ils permettent donc, comme recherché, de spécifier également des organisations partielles comprenant des degrés de liberté, et de décrire ainsi des **systèmes ouverts**.

Nous avons ensuite montré comment il était possible d'exprimer dans notre modèle, à partir des éléments de base, des **notions haut-niveau** usuelles des organisations : **rôles**, **groupes**, **coalitions**, **normes**. La possibilité, souvent aisée, de l'expression de ces concepts dans notre modèle montre la portée expressive de celui-ci, et donc sa potentialité dans une utilisation pratique pour l'analyse ou la conception de systèmes.

Ainsi, la possibilité de décrire des systèmes ouverts et fermés, grâce à un langage de description fondé sur des éléments bas-niveau, ajoutée à la possibilité d'agréger les éléments pour produire une description haut-niveau plus riche, permettent ensemble de répondre à notre premier objectif de **définir un modèle à fort pouvoir expressif**.

Tout au long des chapitres 4 à 6, nous avons décrit formellement le modèle descriptif que nous proposons et les méthodes associées. Cette formalisation s'adapte, comme nous l'avons montré, à la fois à la description de systèmes et à leur conception. Elle permet d'unifier la description organisationnelle de nombreux types de systèmes, et ainsi de rendre possible leur étude conjointe et leur comparaison. À ce titre, la formalisation proposée répond à notre deuxième objectif d'**obtention d'un formalisme suffisant** à l'analyse et à la conception de systèmes.

Enfin, l'exemple de méthode de calcul proposée au chapitre 5 vise à illustrer le fait que la description organisationnelle statique et dynamique à partir d'éléments locaux s'adapte

correctement aux exigences pratiques pour le calcul de propriétés globales sur les organisations. La possibilité de définir plusieurs modèles différents à différents niveaux de spécification (organisations, instances organisationnelles, modèles normatifs), reliés entre eux par des relations de dérivation, permet, de plus, d'analyser une organisation sous plusieurs aspects, du plus général à la situation la plus particulière, et de comparer et d'associer les résultats de ces analyses entre eux. Ceci répond donc à notre troisième et dernier objectif : l'exigence de **calculabilité du modèle**.

8.2 Méthodes et applications

Afin d'étudier et de montrer l'applicabilité de notre modèle à des problèmes théoriques réels, nous avons ensuite proposé des méthodes et appliqué ces méthodes à deux cas d'applications concrètes. Ces cas présentent les deux situations auxquelles nous avons voulu répondre à travers les méthodes proposées : la conception de systèmes suivant des objectifs définis et l'analyse et la comparaison de systèmes existants observés.

8.2.1 Conception de système

Pour ce qui est de la conception, nous avons proposé une méthode progressive pour la conception de système, partant de la décomposition des objectifs haut-niveau, c'est-à-dire des besoins du système, en objectifs bas-niveau – les tâches. Cette décomposition permet ensuite de définir l'organisation elle-même à partir d'une spécification bas-niveau, dont les propriétés haut-niveau sont déduites par agrégation – ce en quoi cette méthode suit l'approche émergentiste que nous proposons.

La méthode proposée comprend ainsi les spécifications organisationnelles complètes correspondant aux besoins énoncés sur le système, tant sur le plan fonctionnel que structurel, d'un point de vue statique et dynamique. Cette spécification se fait par la définition, formelle et sémantique, de l'ensemble des éléments connus du système : types et instances d'agents, de relations, de tâches et mécanismes organisationnels ; elle se complète par l'expression des spécifications nécessaires sur les ensembles de ces objets.

Nous proposons, pour faciliter la transition du modèle organisationnel vers un squelette de code permettant la création du système et des agents le composant, l'exemple d'une équivalence entre concepts organisationnels de notre modèle et notions de programmation orientée objet. Cette équivalence, implémentable sous forme algorithmique d'un générateur de squelette de code et d'un vérificateur de propriétés à la compilation, donne à la méthodologie proposée en complément de notre modèle un appui pratique suffisant pour

envisager son utilisation concrète, et procure ainsi à notre modèle la possibilité réelle d'une utilisation pour la conception de système.

Il est également envisageable, de la même façon, de proposer des équivalences similaires pour d'autres paradigmes de programmation – fonctionnel, impératif, etc. – puisque le travail théorique effectué est indépendant de l'architecture et du code des agents concernés, et d'étendre ainsi la portée pratique de notre modèles à d'autres situations.

Par ailleurs, il est envisageable de pouvoir intégrer notre modèle, de par sa généricité et son expressivité, à d'autres méthodes existantes ou à venir. Il est notamment notable que la formalisation des champs de coopération et de non-coopération des agents permet d'exprimer les propriétés nécessaires à l'utilisation de notre modèle pour de la conception de systèmes effectuée dans le cadre de la théorie des AMAS (*Adaptative MultiAgent Systems*) (Glize *et al.*, 1998), fondée sur l'analyse des cas de coopération et de conflit dans les systèmes.

8.2.2 Analyse de systèmes

Pour ce qui est de l'analyse, notre modèle permet une analyse statique complète de nombreuses propriétés du système, de même que la comparaison entre différentes organisations. Formellement, cela est rendu possible par la représentation du système sous forme de réseaux d'éléments, comparables statiquement entre eux par le biais des méthodes usuelles de comparaison de réseaux. La formalisation proposée permet également d'exprimer, dans certains cas, des invariants organisationnels directement sur les spécifications.

En revanche, l'analyse et la comparaison des dynamiques organisationnelles présentant une complexité bien plus importante, nous proposons une méthode pour réduire cette complexité, et, au prix d'une analyse probabiliste plutôt qu'exacte, d'obtenir des informations quantitatives sur le comportement moyen du système, dans des conditions générales ou dans des conditions spécifiques. Ces informations, utilisables pour exprimer des propriétés globales permettant de caractériser et de comparer les organisations, sont extraites par une modélisation des mécanismes organisationnels sous forme de chaînes de Markov, organisées et simplifiées par divers procédés selon les types et les complexités des mécanismes en jeu. Par le calcul des états stationnaires des chaînes de Markov ainsi exprimées, il est possible de transformer les informations locales connues en informations globales sur l'organisation.

Bien que cette proposition de méthode pour l'analyse et la comparaison d'organisations à partir de la représentation de ces organisations dans notre modèle ne soit qu'une méthode possible parmi d'autres envisageables – analyse par algorithmes cellulaires, réseaux dynamiques, etc. – éventuellement plus adaptées à certaines situations particulières, elle

permet de montrer que le modèle organisationnel proposé réunit les qualités nécessaires à l'utilisation des spécifications organisationnelles seules pour l'analyse et la comparaison formalisée d'organisations. Elle est également adaptée, d'où son choix, à un grand nombre de situations, et en particulier à un grand nombre de types de mécanismes organisationnels différents, et permet dans de nombreux cas d'obtenir une information utilisable, même dans des situations très peu spécifiées.

8.3 Perspectives

Les perspectives de travail autour du modèle organisationnel proposé dans cette thèse s'articulent autour de deux axes : le premier axe, théorique, consiste au développement d'un modèle descriptif complet des SMA permettant d'instancier et de rendre pratique et applicable la méthode VOYELLES (Demazeau, 1995) ; le second axe, pratique, vise au développement et à l'amélioration des méthodes associées au modèle pour l'exploitation de ses possibilités analytiques des SMA.

Dans un premier temps, donc, le modèle s'inscrit, comme nous l'avons défini dans notre approche, dans la séparation des SMA en différentes dimensions descriptives proposée par la méthode VOYELLES. Nous avons posé l'hypothèse de rendre ces dimensions indépendantes, afin de garantir la meilleure généricité aux modèles de chacune des dimensions. Ainsi, pour obtenir un modèle descriptif complet d'un SMA, faudrait-il disposer d'un modèle pour chacune de ses dimensions : Agents, Environnement, Interactions, Organisations. Il faudrait ensuite combiner ces représentations en les liants par leurs articulations théoriques en un modèle commun global. C'est dans cette optique que se développe le premier axe des perspectives du travail présenté, qui est celui du développement des modèles descriptifs correspondant aux trois autres dimensions des SMA : agents (existence, comportement et code), environnements (situation, artefacts, représentations) et interactions (canaux, protocoles). Le but recherché est bien évidemment la création d'un modèle global de représentation des SMA permettant la comparaison et l'analyse des systèmes, et procurant un cadre général à la conception des SMA.

Dans un second temps, le modèle proposé est déjà en lui-même un outil théorique et pratique permettant d'unifier et de formaliser la représentation des organisations dynamiques pour les SMA, et donc de favoriser l'analyse et la comparaison des systèmes par leurs organisations. Cet outil ne pourra être pleinement exploité dans toutes les situations que s'il est possible d'adapter les méthodes, de conception comme d'analyse et de comparaison, à une vaste palette de cas possibles. Cette adaptation nécessitera le raffinement de la modélisation markovienne pour des cas précis, ainsi que l'étude de possibilités alternatives de modélisation, et pourra mener à l'implémentation d'outils spécifiques à ces

modélisations et aux analyses qu'elles visent à produire. Ces travaux s'inscrivent dans la continuité des méthodes et des outils proposés dans cette thèse et ont pour but de donner au modèle proposé un champ d'application plus vaste à travers la possibilité de résoudre, par des méthodes spécifiques, un large spectre de situations.

Références

Adam E., Mandiau R. et Kolski C., 2001. Application of a Holonic Multi-Agent System for Cooperative Work to Administrative Processes. *Journal of Applied Systems Studies*, vol. 2, p. 100-115.

Van Aeken F., 1999. *Les systèmes multi-agents minimaux*. Thèse de doctorat. Institut National Polytechnique de Grenoble.

Allavena A.A.P., 2006. *On the correctness of gossip-based membership protocols*. Ithaca.

Badica A. et Badica C., 2008. Formal specification of matchmakers, front-agents, and brokers in agent environments using FSP. In *Proceedings of the 6th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS 2008)*. Barcelona, Spain, p. 9–18.

Bernon C., Gleizes M.-P., Peyruqueou S., *et al.*, 2003. ADELFE: A Methodology for Adaptive Multi-agent Systems Engineering. In P. Petta, R. Tolksdorf, & F. Zambonelli (éd.) *Engineering Societies in the Agents World III*. Springer Berlin / Heidelberg, p. 70-81.

Blum F., 2007. Digital Interactive Installations: Programming interactive installations using the software package Max/MSP/Jitter.

Boissier O., Balbo F. et Badeig F., 2010. Controlling multi-party interaction within normative multi-agent organizations. In *Proceedings of the 6th international conference on Coordination, organizations, institutions, and norms in agent systems (COIN@AAMAS'10)*. p. 357-376.

Boulet R., 2008. *Comparaison de graphes, applications à l'étude d'un réseau de sociabilité paysan au Moyen Âge*. Toulouse 2 - Le Mirail.

Boulet R. et Jouve B., 2006. Partitionnement d'un réseau de sociabilité à fort coefficient de clustering. *Revue des Nouvelles Technologies de l'Information*, n°9, p. 569-574.

Boulet R., Hautefeuille F., Jouve B., *et al.*, 2007. Sur l'analyse de réseaux de sociabilité de la société paysanne médiévale. In *Actes des journées MASHS 2007*. Brest, France, p. 1-11.

Bouron T., Demazeau Y. et Melaye D., 2006. Which Adequate Trust Model for Trust Networks? *IFIP Artificial Intelligence Applications and Innovations*, vol. 204, p. 236–244.

Brazier F.M.T., Dunin-Keplicz B.M., Jennings N.R., *et al.*, 1997. *DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework. Int. Journal of Cooperative Information Systems*.

Caire G., Coulier W., Garijo F., *et al.*, 2002. Agent Oriented Analysis Using Message/UML. In M.J. Wooldridge, G. Weiß, & P. Ciancarini (éd.) *Agent-Oriented Software Engineering II*. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 119-135.

Camps V., 1998. *Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : application à la recherche d'information dans un système d'information répartie*. Université Paul Sabatier - Toulouse.

Camps V. et Gleizes M.-P., 1995. Principes et évaluation d'une méthode d'auto-organisation. In *Actes des troisièmes journées francophones IAD&SMA, St Baldoph-Savoie, 15/03/95-17/03/95*. p. 337-348.

Castelfranchi C. et Falcone R., 1998. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of the 3rd International Conference on Multi Agent Systems*. p. 72.

Castro J., Kolp M. et Mylopoulos J., 2001. A Requirements-Driven Development Methodology. In *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE '01)*. Springer-Verlag, p. 108-123.

Celikyilmaz A. et Türksen I.B., 2009. *Modeling Uncertainty with Fuzzy Logic* Studies in. 400 p.

Chandy K.M. et Lamport L., 1985. Distributed snapshots: determining global states of distributed systems. *ACM Transactions on Computer Systems*, vol. 3 n°1, p. 63-75.

Cohen H., 1988. How To Draw Three People In A Botanical Garden. *AAAI*, vol. 89, p. 846-855.

Collinot A. et Drogoul A., 1998. Approche orientée agent pour la conception d'organisations : application à la robotique collective. *Revue d'intelligence artificielle*, vol. 12 n°1, p. 125-147.

Cossentino M. et Potts C., 2002. A CASE tool supported methodology for the design of multi-agent systems. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP'02)*.

Costa A.C. da R. et Demazeau Y., 1996. Toward a formal model of multi-agent systems with dynamic organizations. In *Proceedings of 2nd. International Conference on Multi-agent Systems (ICMAS'96)*. p. 431.

Decker K.S., 1996. TÆMS: a framework for environment centered analysis and design of coordination mechanisms. In New York, NY, USA : John Wiley & Sons, Inc., p. 429-447.

Deguet J., 2008. *Intégration de l'émergence au sein des systèmes multi-agent - Une étude appliquée à la recherche heuristique*. Joseph Fourier - Grenoble 1.

Deguet J. et Demazeau Y., 2005. A Complexity Based Feature to Support Emergence in MAS. In *Multi-agent systems and applications IV*. Budapest, Hungary, p. 616-619.

Deguet J., Demazeau Y. et Magnin L., 2006. Elements about the Emergence Issue: A Survey of Emergence Definitions. *Complexus*, vol. 3 n°1-3, p. 24-31.

Deloach S.A., 1999. Multiagent systems engineering: A methodology and language for designing agent systems. In *Proceedings of Agent Oriented Information Systems*. p. 45–57.

Deloach S.A. et Wood M.F., 2001. Developing multiagent systems with agenttool. *Intelligent Agents VII Agent Theories Architectures and Languages*, vol. 1986, p. 30–41.

Demazeau Y., 1995. From interactions to collective behaviour in agent-based systems. In *Proceedings of the 1st. European Conference on Cognitive Science*. Saint-Malo, France, p. 117-132.

Demazeau Y., 1996. Vowels. Invited lecture, 1st Ibero-American Workshop on Distributed AI and Multi-Agent Systems (IWDAIMAS'96).

Demazeau Y. et Costa A.C. da R., 1996. Populations and organizations in open multi-agent systems. In *Proceedings of the 1st National Symposium on Parallel and Distributed AI*. Hyderabad , India, p. 1-13.

Demazeau Y. et Müller J.-P., 1990a. Decentralized AI. *1st European Workshop on Modelling Autonomous Agents in Multi-Agent World, MAAMAW'89, London, August 1989, reviewed in Decentralized AI*, vol. 2, p. 3-13.

Demazeau Y. et Müller J.-P. (éd.), 1990b. *Decentralized Artificial Intelligence*. Elsevier Science Publishers.

Dignum V., 2004. *A Model for Organizational Interaction : Based on Agents, Founded in Logic*. SIKS Dissertation Series.

Dignum V. (éd.), 2009. *Handbook of Research on Multi-agent Systems: Semantics and Dynamics of Organizational Models*. Hershey, PA, USA : Information Science Reference. 602 p. ISBN 978-160566256-5.

Dignum V., Vázquez-Salceda J. et Dignum F., 2005. Omni: Introducing social structure, norms and ontologies into agent organizations R. Bordini, M. Dastani, & J. Dix (éd.). *Programming MultiAgent Systems*, p. 181-198.

Dorra M. et Nejla A., 2005. Intégration d'un agent mobile dans un Système multi-agent hiérarchique. In *Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC (MajecSTIC 2005)*. Rennes, France, p. 426-429.

Drogoul A. et Collinot A., 1998. Applying an Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer. *Autonomous Agents and Multi-Agent Systems*, vol. 1 n°1, p. 113 - 129.

Drogoul A. et Zucker J.-D., 1998. *Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences from the RoboCup Challenge*.

Erl T., 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. ISBN 0131858580.

Esteva M., Padget J. et Sierra C., 2002. Formalizing a Language for Institutions and Norms. In J.-J. Meyer & M. Tambe (éd.) *Intelligent Agents VIII*. Springer Berlin / Heidelberg, p. 348-366.

Esteva M., Rodríguez-Aguilar J.-A., Sierra C., *et al.*, 2001. On the Formal Specification of Electronic Institutions. In F. Dignum & C. Sierra (éd.) *Agent Mediated Electronic Commerce*. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 126-147.

Ferber J. et Gutknecht O., 1998. A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In Y. Demazeau (éd.) *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*. Paris, France : IEEE Press, p. 128-135.

Ferber J., Gutknecht O. et Michel F., 2003. From agents to organizations: an organizational view of multi-agent systems. *Agent-Oriented Software Engineering IV*, vol. 2935pp, p. 214-230.

Ferber J., Michel F. et Baez J., 2005. AGRE: Integrating environments with organizations. *Lecture Notes in Computer Science*, vol. 3374, p. 48-56.

Fischer K., 1999. Holonic Multiagent Systems — Theory and Applications —. In P. Barahona & J. Alferes (éd.) *Progress in Artificial Intelligence*. Springer Berlin / Heidelberg, p. 850.

Foisel R., Chevrier V. et Haton J.-P., 1997. Un modèle pour la réorganisation de système multi-agent. In *Actes des 5èmes journées francophones IAD-SMA*. La Colle Sur Loup, France, p. 261-277.

Fortino G., Garro A. et Russo W., 2005. An integrated approach for the development and validation of multi-agent systems. *Computer Systems Science & Engineering (CSSE)*, vol. 20 n°4, p. 259-280.

Gleizes M.-P., Camps V., Georgé J.-P., *et al.*, 2007. Engineering Systems which Generate Emergent Functionalities. In D. Weyns, S. Brueckner, & Y. Demazeau (éd.) *Engineering Environment-Mediated Multiagent Systems - Satellite Conference held at The European Conference on Complex Systems (EEMMAS'07)*. Dresden, Germany : Springer-Verlag, p. (on line).

Gleizes M.-P., Glize P. et Camps V., 1998. A self-organization process based on cooperation theory for adaptive artificial systems. In *1st International Conference on Philosophy and Computer Science « Processes of evolution in real and Virtual Systems. »*Krakow, Poland.

Glize P., Gleizes M.-P. et Camps V., 1998. Une théorie de l'apprentissage fondée sur l'auto-organisation par coopération . In Ritschard, Berchtold, Duc, *et al.* (éd.) *Apprentissage : des principes naturels aux méthodes artificielles* . Paris : Hermes, p. 117-131.

Gomez-Sanz J.J., Fernández C.R. et Pavón J., 2009. Quick Prototyping and Simulation with the INGENIAS Agent Framework. In *Proceedings of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops*. Turin, Italy.

Groß R., Bonani M., Mondada F., *et al.*, 2006. Autonomous self-assembly in swarm-bots. *Robotics, IEEE Transactions on*, vol. 22 n°6, p. 1115–1130.

Gufflet Y. et Demazeau Y., 2004. Applying the PACO paradigm to a three-dimensional artistic creation. In *5th International Workshop on Agent-Based Simulation (ABS'04)*. Lisbon, Portugal, p. 121-126.

Gutknecht O. et Ferber J., 2000. *MadKit : une architecture de plate-forme multi-agent générique*. Montpellier, France. 13 p.

Hadeli, Valckenaers P., Kollingbaum M., *et al.*, 2004. Multi-agent coordination and control using stigmergy. *Computers in Industry*, vol. 53 n°1, p. 75-96.

Hannoun M., Boissier O., Sichman J.S., *et al.*, 2000. MOISE: An organizational model for multi-agent systems. In M.C. Monard & J.S. Sichman (éd.) *Proceedings of IBERAMIA-SBIA*. Springer, p. 156-165.

Hautefeuille F., 2006. L'extraordinaire ascension d'une famille de marchands de Castelnau-Montratier (46) : Les Trapas (1250-1350). In *Minorités juives, pouvoirs, littérature politique en péninsule ibérique, France et Italie au Moyen Age, études offertes à Béatrice Leroy*. p. 51-64.

Hautefeuille F. et Jouve B., 2004. Analyse d'une base de données de contrats agraires pour comprendre l'espace et la société paysanne du Moyen-Age.

Hendler J., 2001. Agents and the Semantic Web. *IEEE Intelligent Systems*, vol. 16 n°2, p. 30-37.

Hexmoor H., Venkata S.G. et Hayes D., 2006. Modelling social norms in multiagent systems. *Journal of Experimental Theoretical Artificial Intelligence*, vol. 18 n°1, p. 49-71.

Horling B. et Lesser V., 2005. Using ODML to model multi-agent organizations. *Intelligent Agent Technology, IEEE/WIC*.

House R., Rousseau D.M. et Thomas-Hunt M., 1995. The Meso Paradigm: A Framework for the Integration of Micro and Macro Organizational Behavior L.L. Cummings & B. Staw (éd.). *Review of Organization Behavior*, vol. 17, p. 71-114.

Hubner J.F., Sichman Jaime Simão et Boissier O., 2002. A model for the structural, functional, and deontic specification of organizations in multiagent systems. *Lecture notes in computer science*, p. 118–128.

Huhns M.N., Singh M.P., Burstein M., *et al.*, 2005. Research Directions for Service-Oriented Multiagent Systems. *IEEE Internet Computing*, vol. 9 n°6, p. 65-70.

Hutzler G., Gortais B. et Drogoul A., 1997. Le Jardin Des Hasards: peinture abstraite et Intelligence Artificielle Distribuée réactive. In J.-P. Muller & J. Quinqueton (éd.) *Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents*. La Colle Sur Loup, France, p. 295-306.

Jordan M.I., Ghahramani Z., Jaakkola T.S., *et al.*, 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, vol. 37 n°2, p. 183-233.

Joumaa H., Demazeau Y. et Vincent J.-M., 2009. Performance Visualization of a Transport Multi-agent Application. In Y. Demazeau, J. Pavón, J. Corchado, *et al.* (éd.) *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*. Springer Berlin / Heidelberg, p. 188-196.

Kinny D., Georgeff M.P. et Rao A., 1996. A methodology and modelling technique for systems of BDI agents. In W. de Velde & J. Perram (éd.) *Agents Breaking Away*. Springer Berlin / Heidelberg, p. 56-71.

Klein K.J. et Kozlowski S.W.J., 2000. From Micro to Meso: Critical Steps in Conceptualizing and Conducting Multilevel Research. *Organizational Research Methods*, vol. 3 n°3, p. 211-236.

Kozlak J., 2000. *Maintien de l'intégrité fonctionnelle dans les systèmes multi-agents ouverts à ressources renouvelable*. Institut national polytechnique de Grenoble.

Kreps D., 1990. Corporate culture and economic theory. In *Perspectives on Positive Political Economy*. Cambridge, Massachussets.

Krupa Y., Vercouter L., Hubner J.F., *et al.*, 2009. Trust Based Evaluation of Wikipedia ' s Contributors. In *10th International Workshop on Engineering Societies in the agents world (ESAW 2009)*. Utrecht, Netherlands.

Lacomme L., 2011. Vers un modèle générique pour les organisations dynamiques en univers multi-agent. In *Actes de FUTURAMA 2011, Plateforme AFIA*.

Lacomme L., Camps V., Demazeau Y., *et al.*, 2011. Middle Age Social Networks : a dynamic organizational study. In Y. Demazeau, F. Dignum, J.M. Corchado, *et al.* (éd.) *Advances in practical applications of agents and multiagent systems: 9th international conference (PAAMS'11)*. Salamanca, Spain : Springer-Verlag.

Lacomme L., Demazeau Y. et Camps V., 2009a. Classification des mécanismes organisationnels dans les réseaux d'agents. In Guessoum & Hassas (éd.) *17ème Journées Francophones des Systèmes Multi-Agents (JFSMA'09)*. Lyon, France, p. 79-88.

Lacomme L., Demazeau Y. et Camps V., 2009b. How to Integrate Personalization and Trust in an Agent Network. In J. Filipe, A. Fred, & B. Sharp (éd.) *Proceedings of the 1st International Conference on Agents and Artificial Intelligence (ICAART'09), Reviewed in Best papers of Agents and Artificial Intelligence 2009*. Springer-Verlag.

Lacomme L., Demazeau Y. et Dugdale J., 2010. CLIC: an agent-based interactive and autonomous piece of art. In Y. Demazeau, F. Dignum, J.M. Corchado, *et al.* (éd.) *Advances in practical applications of agents and multiagent systems: 8th international conference (PAAMS'10)*. Salamanca, Spain : Springer-Verlag.

Lesser V., Decker K.S., Wagner T., *et al.*, 2004. Evolution of the GPGP/TÆMS Domain-Independent Coordination Framework. *Autonomous Agents and Multi-Agent Systems*, vol. 9 n°1, p. 87-143.

Lind J., 2001. *Iterative Software Engineering for Multiagent Systems: The MASSIVE Method* Lecture No. Springer. 303 p. ISBN 3540421661.

Lindley D.V., 2006. *Understanding uncertainty*. John Wiley and Sons. 250 p. ISBN 0470043830.

Marek I. et Szyld D.B., 2007. Iterative and Semi-Iterative Methods for Computing Stationary Probability Vectors of Markov Operators.

Mattern F., 1989. Virtual time and global states of distributed systems. *Parallel and Distributed Algorithms*, p. 215–226.

Michiardi P., Marrow P., Tateson R., *et al.*, 2007. Aggregation Dynamics in Service Overlay Networks. In *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*. Cambridge, Massachussets : IEEE, p. 129-140.

Mitchell T.M., 1997. *Machine Learning*. McGraw-Hill Science/Engineering/Math. 432 p. ISBN 0070428077.

Montaner M., 2003. A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review*, vol. 19 n°4, p. 285-330.

Munos R. et Moore A., 2009. Influence and variance of a Markov chain: application to adaptive discretization in optimal control. *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 2.

Nehnouh C. et Debbat F., 2009. *Modélisation de l'émergence avec les Automates Cellulaires dans les Systèmes Multi Agents*.

Nguyen C.D., Perini A. et Tonella P., 2008. eCAT: a tool for automating test cases generation and execution in testing multi-agent systems. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers*. Richland, SC : International Foundation for Autonomous Agents and Multiagent Systems, p. 1669-1670.

North M.J., Howe T.R., Collier N.T., *et al.*, 2005. The repast simphony runtime system. In C.M. Macal, M J North, & D. Sallach (éd.) *Proceedings of the Agent 2005 Conference on Generative Social Processes Models and Mechanisms*. p. 151-158.

Odell J., Parunak H.V.D. et Bauer B., 2000. Extending UML for agents. In *Proceedings of the Agent Oriented Information Systems (AOIS) Workshop at the 17th National Conference on Artificial Intelligence (AAAI)*.

Odubiyi J.B., Kocur D.J., Weinstein S.M., *et al.*, 1997. SAIRE - A Scalable agent-based information retrieval engine. In *Proceedings of the first international conference on Autonomous agents*. Marina del Rey, CA, USA : ACM, p. 292–299.

Padgham L. et Winikoff M., 2002. Prometheus: A Methodology for Developing Intelligent Agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02*. New York, New York, USA : ACM Press, p. 37.

Papazoglou M.P., 2001. Agent-oriented technology in support of e-business. *Communications of the ACM*, vol. 44 n°4, p. 71-77.

Pardoux E., 2007. *Processus de Markov et applications : Algorithmes, réseaux, génome et finance, cours et exercices corrigés* Sciences S. Dunod. 320 p. ISBN 210051217X.

Pavón J., Gómez-Sanz J. et Fuentes-Fernández R., 2005. The INGENIAS Methodology and Tools. In *Agent-Oriented Methodologies*. Idea Group Publishing, p. 236-276.

Picard G., 2004. *Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente*. Thèse de doctorat. Université Paul Sabatier de Toulouse III.

Prasad V.M.N., Lesser V. et Lander S.E., 1998. Learning organizational roles for negotiated search in a multiagent system. *International Journal of Human-Computer Studies*, vol. 48 n°1, p. 51-67.

Rao A. et Georgeff M.P., 1995. BDI Agents : From Theory to Practice. In *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*. Melbourne, Australia.

Reynolds C.W., 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH'87)*. p. 25-34.

Ricordel P.-M., 2001. *Programmation Orientée Multi-Agents : Développement et Déploiement de Systèmes Multi-Agents Voyelles*. INPG.

Ricordel P.-M. et Demazeau Y., 2001. *Madel : un langage descriptif pour le développement et le déploiement de systèmes multi-agents*. *IEEE Transactions on Software Engineering*.

Rosenthal J.S., 1994. Markov chain Monte Carlo algorithms. In *Proceedings of 1994 Workshop on Information Theory and Statistics*. IEEE, p. 46.

Sabater J. et Sierra C., 2001. REGRET: A reputation model for gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents - AGENTS '01*. New York, New York, USA : ACM Press, p. 194-195.

Saidani S., 2003. *Modélisation de Réseaux Dynamiques par Automates Cellulaires*. Caen, France.

Saidani S., 2004. *Topodynamique de Graphe*. Caen, France.

Shortliffe E.H., 1976. *Computer-based medical consultations: MYCIN*. New York, NY, USA : Elsevier.

Sichman Jaime Simão, Conte R., Demazeau Y., *et al.*, 1994. A Social Reasoning Mechanism Based On Dependence Networks. In *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI'94)*. Amsterdam, The Netherlands, p. 188-192.

da Silva V.T., Choren R. et de Lucena C.J.P., 2004. A UML based approach for modeling and implementing multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS' 04)*. IEEE Computer Society, p. 914-921.

Smith R.G., 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, vol. C-29 n°12, p. 1104-1113.

Souchon N., Limbourg Q. et Vanderdonckt J., 2002. Task Modelling in Multiple Contexts of Use. In P. Forbrig, Q. Limbourg, J. Vanderdonckt, *et al.* (éd.) *Interactive Systems:Design, Specification, and Verification*. Springer Berlin / Heidelberg, p. 59-73.

Stroock D.W., 2005. *An Introduction to Markov Processes (Graduate Texts in Mathematics)*. Springer. 171 p. ISBN 3540234519.

Tambe M., Adibi J., Al-Onaizan Y., *et al.*, 1999. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, vol. 110 n°2, p. 215-239.

Terabe M., Washio T., Katai O., *et al.*, 1997. A study of organizational learning in multiagents systems. In G. Weiß (éd.) Springer Berlin / Heidelberg, p. 168-179.

Villa N. et Boulet R., 2007. Clustering a medieval social network by SOM using a kernel based distance measure. In *Proceedings of ESANN*. Bruges, Belgique, p. 31-36.

Villa N., Rossi F. et Truong Q.D., 2008. Mining a medieval social network by kernel SOM and related methods. In *Proceedings of MASHS 2008*. Créteil, France.

Werner E., 1989. Cooperating Agents: A Unified Theory of Communication and Social Structure. *Distributed Artificial Intelligence*, vol. 2.

West D.B., 2001. *Introduction to graph theory* Upper, Saddle, & River (éd.). ISBN 0-13-014400-2.

Wooldridge M., Jennings N.R. et Kinny D., 2000. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, vol. 3 n°3, p. 285–312.

Yu B. et Singh M.P., 2003. Searching social networks. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS '03)*. Melbourne, Australia : ACM, p. 65–72.

Zambonelli F., Jennings N.R. et Wooldridge M., 2003. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 12 n°3, p. 317–370.