

UNIVERSITE DE YAOUNDE I
UNIVERSITY OF YAOUNDE I

FACULTE DES SCIENCES
FACULTY OF SCIENCES

DEPARTEMENT D'INFORMATIQUE
DEPARTMENT OF COMPUTER SCIENCE



**DES SYSTEMES MULTI-AGENTS AUX
RESEAUX DE PETRI POUR LA GESTION DES
RESSOURCES NATURELLES :
Le cas de la faune à l'Est-Cameroun**

THESE

Présentée par

Innocent BAKAM TCHIAKAM

(Maître ès-sciences)

Sous la Co-Direction de :

Maurice TCHUENTE

Professeur

François BOUSQUET

HDR

Soutenue le **28 mars 2003**

devant le jury composé de :

EMMANUEL TONYE, Professeur, Université de Yaoundé I
MAURICE TCHUENTE, Professeur, Université de Yaoundé I
FRANÇOIS BOUSQUET, Chercheur, CIRAD-IRRI, Bangkok
FABRICE KORDON, Professeur, Université de Paris VI
ERIC BADOUEL, Chargé de Recherche INRIA, Université de Yaoundé I
MARCEL FOU DA, Chargé de Cours, Université de Yaoundé I
EMMANUEL KAMGNIA, Chargé de Cours, Université de Yaoundé I
CHRISTOPHE LE PAGE, Chercheur, CIRAD Montpellier
JACQUES WEBER Chercheur CIRAD. Paris

Président du Jury
Co-Directeur de thèse
Co-Directeur de thèse
Rapporteur
Rapporteur
Examineur
Examineur
Examineur
Examineur

En vue de l'Obtention du
Doctorat/PhD

Année Académique 2002-2003

DES SYSTEMES MULTI-AGENTS AUX
RESEAUX DE PETRI POUR LA GESTION DES
RESSOURCES NATURELLES :
Le cas de la faune à l'Est-Cameroun

Innocent BAKAM TCHIAKAM

Juin 2002

A mes enfants, Isabelle et Fabrice
A Alliance, ma compagne

Remerciements

Au bout de ce long travail, mes premiers mots de remerciement vont tout naturellement à mon directeur de thèse. Le Professeur Maurice Tchunte a supervisé ce projet. Il m'a prodigué tout au long de cette thèse, et même bien avant, de précieux conseils. Je n'ai certes pas la prétention de les avoir tous suivis à la lettre, mais j'espère qu'il trouvera dans ce document quelque raison de croire qu'il n'a pas perdu son temps.

Je remercie Fabrice Kordon et Eric Badouel pour avoir accepté de rapporter sur cette thèse. Merci également à Emmanuel Tonye, Richard Canal, Marcel Fouda, Christophe Le Page et Jacques Weber pour leur participation au jury.

François Bousquet a été de tous les instants : il a tenu la carotte – et parfois aussi le bâton – pour que ce travail soit mené à son terme. Il a guidé mes pas dans l'univers des multi-agents et de la gestion des ressources renouvelables. Je rends hommage à sa patience et à son humanisme. Qu'il trouve à travers ces pages l'expression de ma profonde reconnaissance. Cette reconnaissance s'adresse également à la chaleureuse équipe Gestion des Ressources rEnouvelables et ENvironnement (GREEN) du Centre de coopération Internationale en Recherche Agronomique pour le Développement (CIRAD) à Montpellier pour toute la sollicitude qu'elle m'a témoignée. Je pense à Jacques Weber, Christophe Le Page, Ani Takforyan, Martine Antona, Olivier Barreteau, Hubert Proton, Juliette Rouchier et Mélanie Requier-Desjardins.

Je remercie le Professeur Claude Girault, responsable du thème Systèmes Répartis Coopératifs (SRC) du Laboratoire d'Informatique de Paris VI (LIP6), de m'avoir accueilli dans son équipe pour la réalisation de la partie de la thèse qui touche aux réseaux de Petri. Le Professeur Fabrice Kordon m'a permis de donner forme à une idée qui m'a hanté depuis le début de cette thèse : la validation formelle des modèles multi-agents. Je lui en suis infiniment reconnaissant. Je remercie également toute l'équipe SRC pour son accueil.

J'exprime ici toute ma gratitude à l'endroit de mes enseignants du Département d'Informatique de l'Université de Yaoundé 1, et notamment à son chef M. Emmanuel Kamgnia. Ils m'ont beaucoup appris.

Merci à mes parents, à ma famille et à mes amis. Ce travail ne serait pas achevé sans la contribution de chacun de vous.

L'amour de mon épouse Alliance a été un agréable stimulant tout au long de cette thèse. De même que le sourire de notre fille Isabelle, durant les derniers mois de la rédaction. Je vous renouvelle à toutes les deux ma profonde affection.

Cette thèse a bénéficié du soutien logistique et financier :

- du CNRS (Centre National de la Recherche Scientifique en France) à travers le projet SEAH (Systèmes écologiques et Action de l'Homme) du Programme Environnement, Vie et Société
- du CIRAD (Centre de Coopération Internationale en Recherche Agronomique pour le Développement) par l'ATP « Systèmes multi-agents et Systèmes d'Information Géographiques » et l'AIC « Validité des Systèmes multi-agents et Aide à la décision »
- du programme de recherche STICASE (Sciences et Techniques de l'Information et de la Communication et leurs Applications aux Sciences de l'Education) de l'Université de Douala.

**LISTE DES ENSEIGNANTS PERMANENTS
DE LA FACULTE DES SCIENCES
Année académique 2001/2002
(Par Département et par Grade)**

1- DEPARTEMENT DE BIOCHIMIE (BC)	
1. ETOA François Xavier (Chef de Départ)	MC
2. ATOGHO Barbara Mma	CC
3. FOKOU Elie	CC
4. BENG Née NINTCHOM PENLAP V	CC
5. BIYITI BI ESSAM née AKAM ADA L.F	CC
6. DEMMANO Gustave	CC
7. EVEHE BEBANDOUE Marie-Solange	CC
8. FEKAM BOYOM Fabrice	CC
9. KANSCI Germain	CC
10. MBACHAM Wilfred	CC
11. MOFOR née TEUGWA Clotilde	CC
12. MOUNDIPA FEWOU Paul	CC
13. MULUH John PAYNE	CC
14. NGUEFACK Julienne	CC
15. NKENFOU NGUEFEU Céline	CC
16. NJAYOU MOUNDJOHOU	CC
17. OBEN Julius ENYONG	CC
18. RIWOM SARA Honorine	CC
19. TCHANA NKOUATCHOUA Angèle	CC
20. TCHUINMOGNE SIMO Jules	CC
21. WAKAM née NANA Louise	CC
22. BELINGA née NDOYE FOE	Ass
23. DJOKAM TAMO Rosine Rachel	Ass
24. AWOANE Célestin	Ass
25. SHANG Judith DZELAMBONG	Ass
26. SOUOPGUI Jacob	Ass
2- DEPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE ANIMALES (B.P.A)	
27. NJIKAM NJIFUTIE (Chef de Départ)	CC
28. NJINE Thomas	PR
29. BILONG BILONG Charles Félix	MC
30. FOMENA Abraham	MC
31. KAMTCHOUING Pierre	MC
32. MESSI Jean	MC
33. NGASSAM Pierre	MC
34. BAPFUBUSA Benoît Alain	CC

35. CHUNGAG ANYE née NKEH B.	CC
36. DIMO Théophile	CC
37. DJIETO Lordon Champlain	CC
38. ESSOMBA née NTSAMA MBALLA	CC
39. FOTO MENBOHAN Samuel	CC
40. KAMGANG René	CC
41. MONAYOUNG René	CC
42. MONKIEDJIE Adolphe	CC
43. NJIETO Lordon Champlain	CC
44. NOLA Moïse	CC
45. RAKOTONIRINA Vincent Silvère	CC
46. TCHUEM TCHUENTE Louis	CC
47. YEWAH PANJO Maurice	CC
48. BELLET EDIMO Oscar Roger	ASS
49. ENO Anna Aret	ASS
50. MBOUDJEKA Innocent	ASS
51. NJAMEN Dieudonné	ASS
3- DEPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE VEGETALES (B.P.V)	
52. AMOUGOU AKOA (Chef de Départ)	MC
53. AHANDA TSOUNGUI née MBOLO M.	CC
54. ATANGANA ETENE Roger (Vice-Doyen No 3)	CC
55. BELL Joseph Martin	CC
56. BONGEN OTTOKO Bruno	CC
57. BOYOMO ONANA	CC
58. KENLA Jean Victor	CC
59. NGALA ZOGO Dieudonné M.	CC
60. NKONMENECK Bernard A.	CC
61. NWAGA Dieudonné M.	CC
62. YOUMBI Dieudonné M.	CC
63. ZAPFACK Louis	CC
64. MOSSEBO Dominique Claude	CC
65. NGOUO Lucas Vincent	CC
66. TSOATA Esaïe	CC
67. ABOMO NDONGO Solange	ASS
68. AMBANG Zachée	ASS
69. BIYE Elvire Hortense	ASS
70. DJOGOUE Pierre François	ASS
71. MBARGA BINDZI Marie A.	ASS
72. NGALLE Hermine BILLE	ASS
73. TEMGO TCHIASSEU Jean Charlem	ASS

4. DEPARTEMENT DE CHIMIE INORGANIQUE(C .I .)	
74. NJOPWOUO Daniel (Chef de Dépt. Vice Doyen No 1)	PR
75. NOAH NGAMVENG Joseph	PR
76. WANDJI Roland	PR
77. NEMBA Robert	MC
78. NGAMENI Emmanuel	MC
79. YOUNANG Elie	MC
80. AGWARA ONDOH	CC
81. AVOM Jérôme	CC
82. BABALE née DJAM DOUDOU	CC
83. BAIZOUMI ZOUA	CC
84. DJOUFAC WOUMFO Emmanuel	CC
85. DOUBLA AVALY	CC
86. ELIMBI Antoine	CC
87. GHOGOMU Paul MINGO	CC
88. GWET Simon-Pierre	CC
89. KAMWA Jean Dagobert	CC
90. KETCHA MBADCAM Joseph	CC
91. KONG SAKEO	CC
92. LAMINSI Samuel	CC
93. LIBOUM	CC
94. MELO née CHIJE Uphie F.	CC
95. MUOLIOM Christophe	CC
96. NDIFON Peter TEKE	CC
97. NDIKONTAR Maurice KOR	CC
98. NGOMO Hoarce MANGA	CC
99. PAYOM Gaston	CC
100. SIGNING Pierre	CC
101. NANSEU Charles Péguy	ASS
102. NENWA Justin	ASS
5-DEPARTEMENT DE CHIMIE ORGANIQUE(CO)	
103. SONDEGAM B. Lucas (Chef de départ.)	PR
104. FONUM TANEE Zacharias	PR
105. NGADJUI TCHALEU Bonaventure	PR
106. TSAMO Etienne	PR
107. FON KIMBU Samuel	MC
108. GHOGOMU TIH Raphaël	MC
109. LONTSI David	MC
110. NKENFACK AUGUSTIN Ephraïm	MC

111. NYASSE Barthélemy	MC
112. WANDJI Jean	MC
113. DONGO Etienne	CC
114. FOLEFOC Gabriel NGOSONG	CC
115. KAPNANG Henriette	CC
116. KEUMEDJIO Félix	CC
117. KOUAM Jacques	CC
118. MBAFOR Joseph	CC
119. MBAZOA née DJAMA Céline	CC
120. MESSANGA Bernard Blaise	CC
121. MOFO née NGOUNOU Fernande	CC
122. NGOUELA Silvère Augustin	CC
123. NTEDE NGA Hypolite	CC
124. NYEMBA Jean-CLAUDE	CC
125. PEGNYEMBA née ETOUNOU Anne-M	CC
126. TCHOUANKEU Jean-Claude	CC
127. TIH née NGO BILONG E.Anastasia	CC
128. YANKEP Emmanuel	CC
129. Alex de Théodore ATCHADE	ASS
130. NOUNGOUE TCHAMO Diderot	ASS
6-DEPARTEMENT D'INFORMATIQUE	
131. KAMGNIA Emmanuel (Chef de Départ)	CC
132. BOYOM SOP Flaubert	CC
133. FOTSO Pauline Laure	CC
134. FOUA NDJODO Marcel	CC
135. KAMDOUM Basile	CC
136. LOUKA Basile	CC
137. NZALI Jean-Pierre	CC
138. TINDO Gilbert	CC
139. ATSA ETOUNDI Roger	ASS
140. NDOUNDAM René	ASS
141. TADONKI Claude	ASS
7-SERVICE DES LANGAGES(LA)	
8-DEPARTEMENT DE MATHEMATIQUES(MA)	
142. BEKOLLE David (Chef de Départ)	MC
143. DOSSA COSSY Marcel	MC
144. MOULEN Joël	MC

145. NGUETSENG Gabriel	MC
146. NJOCK Georges Edward	MC
147. NOUTCENGUEME Norbert	MC
148. TAYOU SIMO	MC
149. WOUAFO KAMGA Jean (Doyen)	MC
150. BATE EYA Hans MBI	CC
151. BEBBE NJOH Etienne	CC
152. BINZOULI Etienne Jean-Jacques	CC
153. FOMEKONG Christophe	CC
154. KIKI Maxime Armand	CC
155. MBIANDA Gilbert	CC
156. MEWOLI Boulchard	CC
157. MDAKBO Victor	CC
158. NGUIMTSA Charles	CC
159. NKUIMI JUGNIA Célestin	CC
160. TCHANGANG Roger Duclos	CC
161. TIAYA TSAGUE N. Anne-Marie	CC
162. TOLEN Joséphine	CC
163. TONGA Marcel	CC
164. WAMON François	CC
165. ZAME Alfred	CC
166. EMVUDU WONO Yves S.	ASS
167. TAN AHANDA Barnabé	ASS
168. NOUNDJEU Pierre	ASS
169. TCHAPNDA NJABO Sophonie Blaise	ASS
170. TSEBO Alexis	ASS
9- DEPARTEMENT DE PHYSIQUE (PH)	
171. OWONO ATEBA (Chef de Départ)	MC
172. DOMNGANG Samuel	PR
173. KOFANE Timoléon Crépin	MC
174. KWATO NJOCK Moïse Godefroy	MC
175. MANGUELLE-DICOUM Eliézer	MC
176. EKOBENA FOUA Henri Paul	CC
177. ESSIMBI ZOBO Bernard	CC
178. JONGWANE DIPOKO	CC
179. KWENDE MBANWI Teddy	CC
180. MAGA Emire Mondésir	CC
181. MBANE BIOUELE	CC
182. MBOUNGA Jean-Marie	CC
183. MKANKAM KAMGA François	CC
184. MOUSSA Idolko	CC

185. NDJAKA Jean Marie Bienvenu	CC
186. NDOHI Raphaël	CC
187. NJOMO Donatien	CC
188. NOUAYOU Robert	CC
189. PEMHA Elkana	CC
190. TABOD Charles TABOD	CC
191. TCHAWOUA Clément	CC
192. TCHOFFO Fidèle	CC
193. WOAFO Paul	CC
194. ZEKENG Serge Sylvain	CC
195. BEN-BOLIE Germain Hubert	CC
196. BIYA MOTTO Frédéric	CC
197. MBONO SAMBA Yves Christian U.	CC
10- DEPARTEMENT DES SCIENCES DE LA TERRE (S.T)	
198. ENO BELINGA Samuel Martin (Chef de Départ)	PR
199. TCHOUA M.Félix	PR
200. BILONG Paul	MC
201. EKODECK Georges E.	MC
202. NDJENG Emmanuel	MC
203. NZENTI Jean-Paul	MC
204. ABOSSOLO née ANGUE Monique	CC
205. BEKOA Etienne	CC
206. BITOM Dieudonné	CC
207. BONDJE Charles	CC
208. DJEUDA TCHAPNGA Henri Bosco	CC
209. GHOGOMU Richard TANWI	CC
210. KAMGANG Pierre	CC
211. LAMILLEN BILLA Daniel	CC
212. LEGER Jean Marc	CC
213. MOUAFO Lucas	CC
214. MOUNDI Amidou	CC
215. MVONDO ONDOA Joseph	CC
216. NANA Richard	CC
217. NGONGE Emmanuel Donald	CC
218. NGOS III Simon	CC
219. NJILAH Issac KONFOR	CC
220. NJOM Bernard de Lattre	CC
221. NKONGUIN NSIFA Emmanuel	CC
222. NKOUMBOU Charles	CC
223. NYECK Bruno	CC
224. TCHOUANKOUE Jean-pierre	CC

225. TEMDJIM Robert	CC
226. VICAT Jean-Pierre	CC
227. YENE ATANGANA Joseph Q	CC
228. YONGUE née FOUATEU Rose	CC
229. BISSO Dieudonné	ASS
230. ESSONO Jean	ASS
231. MINYEM Dieudonné	ASS
232. MVOGO Jean Kisito	ASS
233. NDAM NGOUPAYOU Jules-Remy	ASS
234. MDJIGUI Jean - Désiré	ASS
235. NGO BIDJECK Louise Marie	ASS
236. ZO ?O ZAME Philémon	ASS

Table des matières

Remerciements	iii
Introduction et problématique générale	1
0.1 Introduction	1
0.2 Dynamique des populations	2
0.3 La gestion des ressources naturelles renouvelables	4
0.3.1 Enjeux	4
0.3.2 Outils de gestion des ressources communes	6
0.3.3 L'approche patrimoniale	8
0.4 Problématique de la thèse	8
0.5 Plan de la thèse	11
I LA MODELISATION MULTI-AGENT	13
1 Les systèmes multi-agents	15
1.1 Introduction	15
1.2 La notion d'agent	15
1.3 Typologies d'agents	17
1.3.1 Agents cognitifs vs agents réactifs	18
1.3.2 Agents situés vs agents communicants	21
1.4 Les architectures d'agents	21
1.4.1 Les architectures réactives	21
1.4.2 L'architecture de subsomption	23
1.4.3 L'architecture BDI	23
1.4.4 Les architectures logiques	25
1.5 Notion de système multi-agent	26
1.5.1 Définition	26
1.5.2 Modes d'interaction	27
1.5.3 Les domaines d'application	28
1.6 La simulation multi-agent	29

1.6.1	Principe	29
1.6.2	Quelques simulateurs	30
1.6.3	Quelques extensions des SMA	33
1.7	SMA et Gestion de ressources renouvelables	34
1.8	Conclusion	35
2	La plate-forme de simulation CORMAS	37
2.1	Introduction	37
2.2	La simulation	38
2.2.1	Principe général	38
2.2.2	Modèles de simulations	39
2.3	Les plate-formes multi-agents	40
2.3.1	Madkit	40
2.3.2	Dima	40
2.3.3	Swarm	41
2.4	Pourquoi CORMAS ?	41
2.5	Les concepts de l'environnement CORMAS	42
2.5.1	L'espace	42
2.5.2	La cellule	42
2.5.3	Le canal	43
2.5.4	Les entités situées, les entités communicantes	44
2.5.5	La dynamique	44
2.5.6	Le modèle	45
2.5.7	La simulation	46
2.6	Démarche de conception de modèles pour CORMAS	47
2.6.1	Construction du modèle	47
2.6.2	Construction des simulations	48
2.7	Implémentation	49
2.7.1	Implémentation du modèle	49
2.7.2	Initialisation de la simulation	56
2.7.3	Déroulement de la simulation	57
2.7.4	Résultats de la simulation	58
2.8	Applications	58
2.8.1	Le feu de brousse	58
2.8.2	Dynamiques de transformation des paysages.	62
2.8.3	Autres applications	64
2.9	Conclusion	65

II	MODELISATION DE LA CHASSE	69
3	La chasse à l'Est-Cameroun	71
3.1	Introduction	71
3.2	La chasse dans l'espace et dans le temps	72
3.2.1	La zone d'étude	72
3.2.2	L'activité de chasse	72
3.2.3	Organisation sociale	74
3.2.4	Le temps	75
3.2.5	L'espace	75
3.3	Le Céphalophe bleu	77
3.3.1	Connaissances éthologiques	77
3.3.2	Etude du comportement social	77
3.4	Conclusion	78
4	Modélisation multi-agent de la chasse	81
4.1	Introduction	81
4.2	Modélisation	82
4.2.1	Le céphalophe	82
4.2.2	Le sentier de chasse	87
4.2.3	Le chasseur et le groupe de chasseurs	89
4.3	Simulations	90
4.3.1	Calibrage	90
4.3.2	Scénarios	91
4.3.3	Résultats des simulations	92
4.3.4	Interprétation	94
4.4	Discussion	95
4.5	Conclusion	96
III	VALIDATION PAR LES RESEAUX DE PETRI	97
5	Modélisation par les Réseaux de Petri	99
5.1	Introduction	99
5.2	Les Réseaux de Petri	100
5.2.1	Définition informelle	100
5.2.2	Définition formelle	101
5.2.3	Règle de franchissement	101
5.2.4	Théorème fondamental (Condition nécessaire d'accessibilité)	101
5.2.5	Exemple	102

5.3	Systèmes multi-agents et réseaux de Petri	104
5.3.1	Le formalisme BRIC	104
5.3.2	Le formalisme VAL	105
5.4	Vers une approche basée sur les réseaux colorés	106
5.4.1	Les réseaux de Petri colorés	106
5.4.2	Modélisation par les réseaux de Petri colorés	107
5.4.3	CPN-AMI	107
5.5	Le modèle Réseau de Petri coloré	108
5.5.1	Les hypothèses du modèle	109
5.5.2	Description des places principales	109
5.5.3	Module de déplacement	110
5.5.4	Module de dynamique de la population	112
5.5.5	Module de croissance des individus	112
5.5.6	Module de séquençement	116
5.5.7	Les paramètres du modèle	116
5.6	Conclusion	118
6	Validation formelle	119
6.1	Introduction	119
6.1.1	Analyse structurelle	119
6.1.2	Le graphe des marquages accessibles	120
6.2	Optimisation du modèle	121
6.2.1	Module de déplacement révisé H'1	121
6.2.2	Les autres modules	124
6.3	Analyse du modèle optimisé	124
6.4	Conclusion	126
	Conclusion générale	129
	Bibliographie	137
IV	Annexe : Le guide de l'utilisateur de CORMAS	139

Table des figures

1	Dynamique d'une ressource renouvelable	3
2	Modèle proie-prédateur de Lotka-Voltera	3
3	Le modèle bio-économique de Schaffer	5
4	Effet d'une taxe	6
5	Instauration de quotas	7
6	Instauration de licences	7
1.1	Modèle d'agent générique	16
1.2	Des agents cognitifs	18
1.3	Les agents réactifs dans un jeu de taquin	19
1.4	Des agents purement réactifs aux agents purement cognitifs, un continuum	19
1.5	un agent à états	20
1.6	Exemple de réseau de neurones	22
1.7	L'architecture de subsomption	23
1.8	Modèle de robot explorateur à base d'architecture de subsomption	24
1.9	Un exemple de manipulation de cubes	26
1.10	L'interface de SHADOC	31
1.11	L'interface de SEALAB	32
1.12	Les fourmis de MANTA	33
2.1	Les voisinages d'ordre un (ou immédiat) d'une cellule	43
2.2	Le passage de la réalité au modèle	47
2.3	Le passage du modèle à l'implémentation	49
2.4	L'écran principal de CORMAS	50
2.5	L'écran de définition des entités	50
2.6	L'écran de définition des messages.	53
2.7	L'écran de définition des observations.	54
2.8	L'écran de définition de l'affichage des communications.	56
2.9	L'écran de définition des procédures d'initialisation et de contrôle global.	57
2.10	L'espace de simulation	58
2.11	L'observateur des communications et échanges	59

2.12	L'affichage des statistiques	59
2.13	Evolution asynchrone et synchrone du modèle feu de brousse	61
2.14	Simulation de la dynamique de transformation des paysages	64
3.1	Carte des villages d'étude	73
4.1	Un agent Céphalophe	82
4.2	Carte de Djemiong avec les zones chassées en rouge	88
4.3	Un agent Chasseur	88
4.4	Convergence de la dynamique sans chasse	91
4.5	Scénario 1 : chasse en continu et sentiers fixes	92
4.6	Scénario 2 : chasse périodique et sentiers fixes	92
4.7	Scénario 3 : chasse périodique et sentiers redéfinis chaque année	93
4.8	Scénario 4 : rotation individuelle des chasseurs	93
4.9	Scénario 5 : Chasseurs qui se coordonnent en groupes.	93
5.1	Exemple de réseau de Petri marqué	100
5.2	Modélisation du partage d'une ressource par un réseau de Petri	102
5.3	Franchissement de transitions dans un réseau de Petri	103
5.4	Structure d'un composant BRIC et de ses bornes	105
5.5	Module de déplacement et de prélèvement par piégeage	111
5.6	Module de croissance de la population	113
5.7	Module de croissance des individus	114
5.8	Module de séquençement	115
6.1	Un réseau de Petri marqué et son GMA	120
6.2	Le choix de la direction de déplacement	122
6.3	Module de déplacement et de prélèvement par piégeage, sans les permutations	123
6.4	Module révisé de croissance de la population	125
6.5	Evolutions comparatives des tailles du GMA	125
6.6	Statistiques des graphes de marquages accessibles	126

Introduction et problématique générale

0.1 Introduction

La protection de l'environnement en général et de la faune en particulier est d'une brûlante actualité. En effet, selon les experts [Bauer, 1998], 2 espèces animales disparaissent de la surface du globe toutes les trois heures. Pour apporter une réponse à cette question cruciale, et parfois simplement pour calmer une opinion publique de plus en plus alarmiste sur les questions touchant à l'environnement, les pouvoirs publics mettent en place de part le monde des mesures drastiques de protection des espèces « menacées ». C'est ainsi que les concepts comme « espèces protégées », « réserve de la biosphère », « classé patrimoine de l'humanité », « ouverture de la période de chasse » entre autres sont entrés dans le vocabulaire quotidien. Des portions entières d'écosystèmes considérées comme uniques ou menacées sont ainsi mises sous cloche afin de les préserver contre les agressions de l'homme [Takforyan, 1995]. L'expérience a montré qu'une telle procédure est une solution imparfaite. D'une part la mobilité de nombreuses espèces ne permet pas de les isoler totalement et d'autre part les conflits engendrés avec les populations adjacentes qui perdent l'accès à des ressources qu'elles ont toujours exploitées ne cessent de s'aggraver [Weber, 1992].

De « nouvelles » politiques de gestion des ressources communes sont aujourd'hui envisagées, et l'informatique se met au service du politique pour l'aider à évaluer l'impact des stratégies de gestion sur la viabilité de la faune. Pour être pertinente, cette évaluation s'appuie sur la modélisation et la simulation. Un modèle étant la représentation simplifiée d'un phénomène réel, la simulation permet d'étudier le comportement du modèle et d'extrapoler ensuite sur le phénomène initial pour en comprendre les propriétés. De façon classique, les phénomènes sont décrits par un ensemble de variables numériques et les équations sont utilisées pour établir des relations entre ces variables. On parle alors de modèles mathématiques. Grâce à la puissance de calcul des ordinateurs, la modélisation mathématique a connu un énorme succès et est largement utilisée dans de nombreux domaines, notamment en biologie des populations pour l'étude de leur dynamique. Nous présentons de façon plus détaillée dans le paragraphe suivant les modèles de dynamique de populations.

0.2 Dynamique des populations

Les travaux sur la dynamique des populations sont nombreux et au fil des décennies, on a vu apparaître des modèles mathématiques plus ou moins complexes décrivant les variations des tailles et des densités de populations.

Un des modèles les plus simples de croissance des populations est donné par la relation

$$N_i = N_{i-1} + (n + i - m - e) \quad (1)$$

où N_i et N_{i-1} représentent les effectifs de la population respectivement à la fin et au début de l'intervalle de temps considéré, et n, i, m, e les nombres, respectivement, de naissances, d'immigrants, de décès et d'émigrants, enregistrés dans l'intervalle de temps considéré [Barbault, 1992]. Ce modèle exprime le fait que l'effectif des populations varie en fonction de l'équilibre entre les processus de recrutement (natalité + immigration) et les processus de disparition (mortalité + émigration).

Les ressources communes nécessaires à la survie des espèces sont généralement en quantité limitée. Par ailleurs, même si les ressources sont en abondance, les individus en concurrence se nuisent mutuellement lors de la recherche de ces ressources. Il advient alors une compétition intraspécifique qui entraîne la régulation de la croissance de la population, phénomène connu sous le nom de densité-dépendance. Cette régulation du niveau d'abondance a été introduite par le mathématicien belge F. Verhulst en 1838 [Le Page, 1996], avec l'équation logistique

$$\frac{dx}{dt} = rx \frac{K - x}{K} \quad (2)$$

où $x(t)$ est la taille de la population à l'instant t , r est le taux d'accroissement maximum de la population et K est la capacité biotique du milieu.

Cette évolution est représentée sous deux formes équivalentes [Takforyan, 2001] : la courbe en « S », qui rend compte de l'évolution du stock biologique (x) dans le temps, et la courbe en « cloche », qui présente le taux de croissance du stock selon sa taille (dx/dt).

Ce modèle a été par la suite étendu à la compétition interspécifique et aux interactions proie/prédateur. L'exemple le plus célèbre étant le système de Lotka-Volterra

$$\begin{cases} \frac{dN_1}{dt} = r_1 N_1 - P N_1 N_2 \\ \frac{dN_2}{dt} = a P N_1 N_2 - d_2 N_2 \end{cases} \quad (3)$$

Dans le système de Lotka-Volterra, N_1 et N_2 représentent respectivement les populations de proies et de prédateurs, P est le coefficient de prédation, a est l'efficacité avec laquelle les prédateurs convertissent la nourriture en descendance, r_1 est le taux de fécondité des proies et d_2 le taux de mortalité des prédateurs [Volterra, 1926].

Certains modèles de dynamique de populations utilisent les équations aux dérivées partielles permettant d'exprimer la densité de population par unité d'espace. La résolution

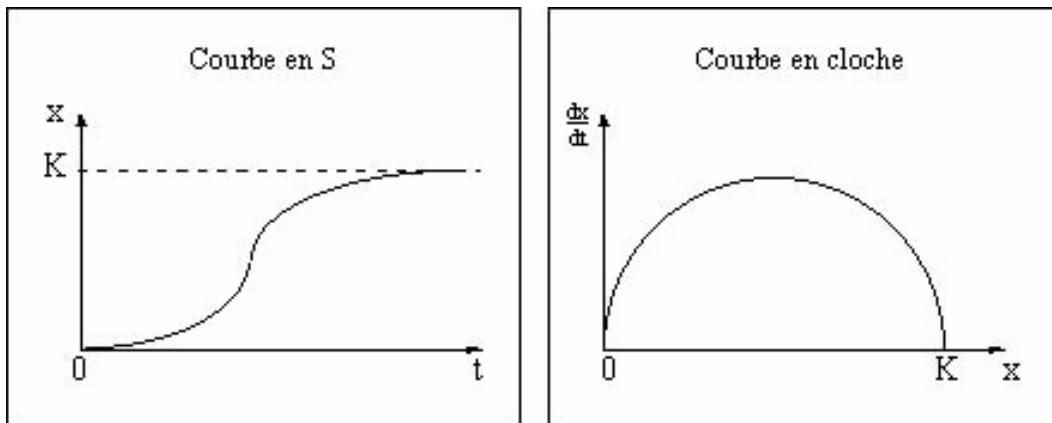


FIG. 1 – Dynamique d'une ressource renouvelable

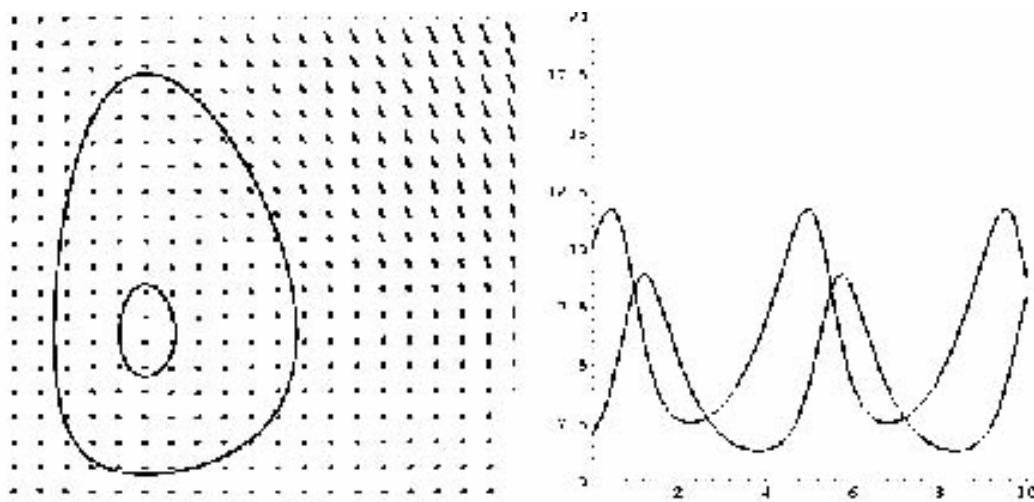


FIG. 2 – Modèle proie-prédateur de Lotka-Volterra

de ces équations par des méthodes analytiques ou numériques permet d'analyser le modèle qu'elles représentent.

Toutefois, la modélisation mathématique de la dynamique des populations présente de nombreux inconvénients. Dans le système de Lotka-Volterra ci-dessus, on peut par exemple s'interroger sur le réalisme et la pertinence des paramètres P , a , r_1 , et d_2 . En général, le modélisateur attribue à ces paramètres une valeur initiale fondée sur une estimation empirique, valeur qui est par la suite ajustée de sorte que les résultats s'accordent avec une forme régulière. Quelle signification exacte attribue-t-on à « l'efficacité avec laquelle les prédateurs convertissent la nourriture en descendance » et comment l'évalue-t-on numériquement ? Plus généralement, quelle est la pertinence des paramètres utilisés dans ces modèles ? Comment intégrer dans les modèles mathématiques des niveaux d'abstraction différents d'un même phénomène ? Comment prendre en compte des aspects non quantifiables d'un phénomène ?

Pour étudier la dynamique des systèmes complexes, l'approche multi-agent est aujourd'hui utilisée comme une alternative aux modèles mathématiques classiques. Cette approche consiste à représenter le système non pas en termes de variables et d'équations, mais plutôt en termes d'entités et d'interactions. Elle permet de substituer aux paramètres globaux dont la justification n'est pas toujours évidente, des comportements locaux directement observables dans la réalité. L'approche multi-agent permet également de prendre en compte dans les modèles des comportements qualitatifs et individuels.

0.3 La gestion des ressources naturelles renouvelables

0.3.1 Enjeux

Dans son célèbre article « The Tragedy of the Commons » paru dans le journal *Science* en 1968 [Hardin, 1968] et repris dans [Hardin and Baden, 1977], Garrett Hardin soutient que la gestion des biens communs conduit inéluctablement à une surexploitation de la ressource jusqu'à sa disparition. Cette surexploitation est due au fait que chaque usager essaie de maximiser son profit individuel, étant donné que les conséquences sont équitablement partagées par tous. La thèse de Hardin, même si elle a été vivement critiquée par d'autres théoriciens de la gestion des biens communs (notamment par Berkes et al. dans [Berkes and Farvar, 1999]) pose très bien le problème de la gestion des ressources naturelles renouvelables, surtout en situation d'accès libre à la ressource.

La théorie économique de la gestion des ressources renouvelables considère les hypothèses suivantes :

- la ressource renouvelable est constituée d'une seule espèce dont l'évolution est représentée par la loi logistique et dont le prix est fixe ;

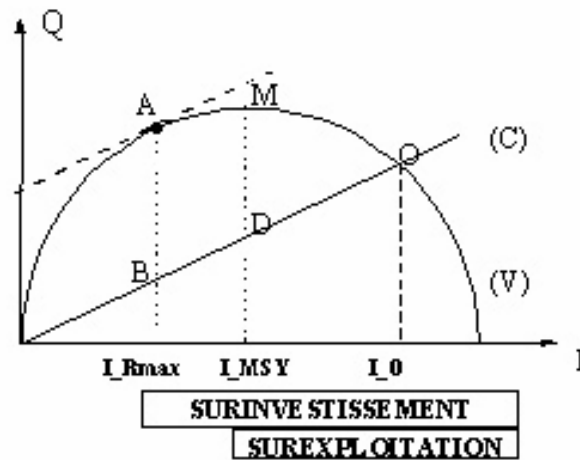


FIG. 3 – Le modèle bio-économique de Schaffer

- les agents sont des producteurs rationnels qui maximisent le profit tiré de l'exploitation de la ressource ;
- le marché est en situation d'accès libre : de nombreux exploitants ont accès sans entrave à la ressource.

Ce modèle dû à Schaffer est illustré par la figure 3.

Dans cette figure, l'axe des abscisses correspond à l'investissement, ou de manière plus générale à l'effort que les populations fournissent pour exploiter la ressource. L'axe des ordonnées représente le produit de cet effort. C'est par exemple la quantité de ressource prélevée ou simplement sa valeur sur le marché. La droite (C) est la courbe des coûts, qui est proportionnelle à l'effort. La courbe (V) correspond à la valeur de la production. Les prix sont supposés constants.

A l'idéal, la quantité exploitée devrait se situer au niveau correspondant au point maximum de la courbe en cloche. Il s'agit du taux de croissance maximal de la population et, par conséquent, du maximum que l'on puisse prélever à perpétuité de cette population sans en réduire la taille ; on parle de « rendement maximum soutenable » (maximum sustainable yield, MSY). Un niveau de prélèvement situé à gauche du MSY représente une situation de sous-exploitation relative car un taux supérieur serait possible et plus avantageux pour les hommes ; à droite du MSY, le stock est considéré comme relativement surexploité car le taux de renouvellement de la ressource est comparativement plus faible [Takforyan, 2001].

En situation d'accès libre, l'investissement augmente jusqu'à l'égalisation des revenus et des coûts. La production croît avec l'investissement jusqu'au point M correspondant à I_{MSY} . Puis la production décroît lorsque I continue à augmenter. En l'absence d'entrave, I augmentera jusqu'à I_0 pour lequel (C) et (V) se croisent en O : la rente, surface comprise entre (C) et (V) a disparu. Mais dès I_{RMax} , l'investissement excède ce qui

est nécessaire pour tirer le profit maximal de l'exploitation, excédent ici représenté par [AB]. Le surinvestissement commence avant la surexploitation de la ressource, qui survient lorsque l'investissement est supérieur à I_{MSY} [Weber, 1997].

On remarque que le MSY, situé à droite du niveau d'effort d'équilibre A, ne représente pas une pratique de gestion désirable sur le plan économique, puisqu'il correspond à une situation de surinvestissement. De plus, la poursuite d'objectifs économiques entraîne une moindre pression sur la ressource renouvelable, par rapport à des objectifs purement biologiques.

0.3.2 Outils de gestion des ressources communes

L'élaboration de la réglementation en ce qui concerne la gestion des ressources communes relève des pouvoirs publics. A cet effet, ils utilisent une panoplie d'outils [Weber, 1997] tels que les taxes, les quotas, les licences ou les droits d'usage exclusifs.

Les taxes, proportionnelles à l'investissement (figure 4), ont pour effet d'augmenter les coûts d'exploitation.

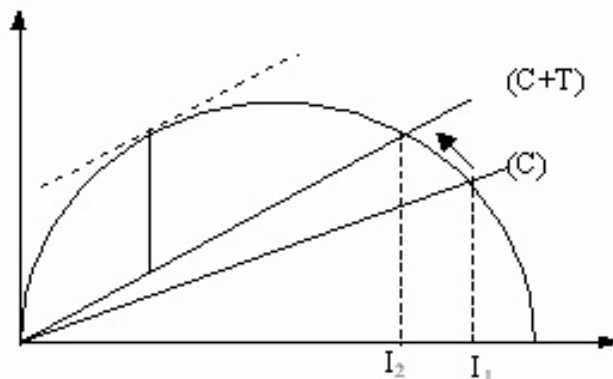


FIG. 4 – Effet d'une taxe

En accès libre, l'investissement entre dans l'exploitation jusqu'au niveau correspondant à une rente nulle. Une taxe relève la droite des coûts de (C) en (C+T). Tout investissement au-delà de I_2 cesse d'être rentable.

L'instauration de quotas (figure 5), sans limitation de l'accès à la ressource, revient à diminuer l'approvisionnement pour un investissement qui n'a pas diminué. Les coûts restent proportionnels à la ressource prélevée. Cette politique préserve la ressource au prix d'un gaspillage économique.

Instaurer les licences revient à définir qui a le droit d'exploiter la ressource. Il s'agit donc d'exclure des exploitants. La droite des coûts ne change pas (figure 6). L'investissement

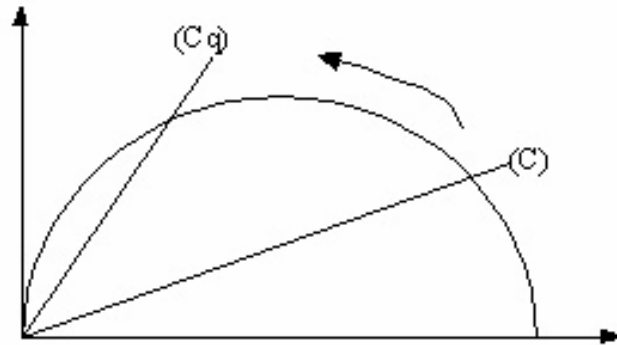


FIG. 5 – Instauration de quotas

présent diminue en même temps que l'effort. Si la situation initiale est celle du point C (profit nul), l'instauration de licences restreint l'investissement de I_C à I_L . La rente était nulle en C, elle est positive en L. Les licences ont pour effet de diminuer la pression sur la ressource et de restaurer la rentabilité.

Un exemple simple est celui de la politique de gestion des taxis dans la ville de Paris. La municipalité attribue un nombre déterminé de licences. Une licence peut être transférée d'un exploitant à l'autre, mais le nombre de licences reste constant.

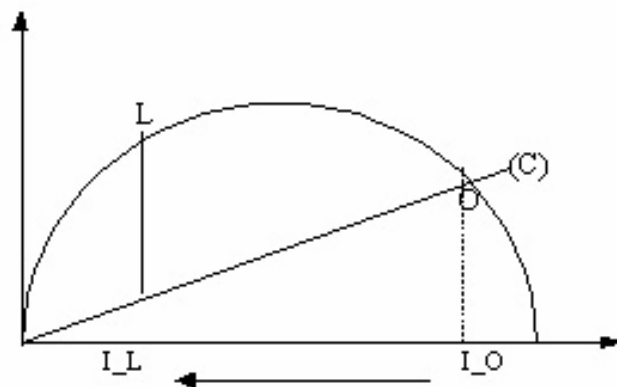


FIG. 6 – Instauration de licences

Les droits d'usage correspondent à un « coût d'accès » à payer pour entrer dans l'exploitation [Weber, 1997].

Toutes ces mesures relèvent de l'approche « économique » et tendent à éviter la sur-

exploitation de la ressource. Cette approche présente cependant de graves inconvénients. D'une part, ces outils pris isolément se sont souvent avérés inefficaces, d'autre part ils concentrent tout entre les mains de l'Etat qui n'a pas toujours les moyens de contrôle de sa politique. En même temps, les autres acteurs du système ne se sentant plus impliqués, cette perte d'identité conduit ceux-ci à ne plus considérer les ressources comme faisant partie de leur patrimoine.

C'est par exemple le cas lorsqu'une aire protégée est créée dans une forêt qui était jusqu'alors exploitée par les habitants des villages avoisinants. L'interdiction d'accès à la forêt amène ces villageois à ne plus exercer de « contrôle » sur les ressources de la forêt, laissant ainsi libre cours aux braconniers.

Il faut donc intégrer les outils de l'approche économique dans une approche plus globale, inspirée de l'analyse des systèmes et de la sociologie des organisations [de Mongolfier and Natali, 1987].

0.3.3 L'approche patrimoniale

C'est une approche participative qui requiert un médiateur. Celui-ci doit animer un processus de dialogue et de négociation dont le résultat doit être constitué [Weber, 1996] :

- d'objectifs de très long terme (une génération) ;
- de scénarios de gestion a moyen terme ;
- de l'élaboration d'une structure de gestion négociée.

L'approche patrimoniale relève donc de l'aide à la décision qui permet au plus grand nombre des titulaires d'un patrimoine de garder le souci du long terme, de prendre conscience des multiples interdépendances entre acteurs, et de se rencontrer pour décider en commun les mesures qui permettront au plus juste prix, de conserver ou d'augmenter ce patrimoine tout en en tirant le maximum d'avantages [de Mongolfier and Natali, 1987].

0.4 Problématique de la thèse

Dans cette thèse, nous nous intéressons à la mise en oeuvre de l'approche patrimoniale. Cela suppose qu'il y ait un consensus entre les différents acteurs impliqués dans le gestion d'une ressource, sur la dynamique à long terme de celle-ci, les conditions initiales et les scénarios d'évolution étant définis. Il s'avère donc nécessaire de simuler les interactions entre dynamiques écologiques, sociales et économiques afin de comprendre les liens entre processus de décision des acteurs, modes d'arbitrage entre intérêts multiples et mise en oeuvre des politiques environnementales. Dans une optique d'aide à la décision, les systèmes multi-agents sont utilisés pour l'aide à la négociation, l'analyse et la modélisation des politiques environnementales, ou en couplage avec des systèmes d'information géographique.

Notre travail s'inscrit donc dans le cadre de la problématique de la gestion des ressources naturelles renouvelables, en s'appuyant sur le cas de la chasse

dans la forêt villageoise de Djemiong à l'Est-Cameroun. Cette thèse se veut une illustration de l'apport d'une méthodologie très récente en informatique : l'approche multi-agent, à l'étude des modes de gestion des ressources naturelles renouvelables. Elle est en même temps une contribution à l'étude formelle des systèmes multi-agents par les réseaux de Petri, l'objectif de cette étude étant notamment la preuve formelle de certaines propriétés observées lors des simulations multi-agents.

Partant des enquêtes de terrain réalisées par Ani Takforyan dans le cadre d'une thèse de socio-économie [Takforyan, 2001], nous proposons un modèle multi-agent qui intègre autant les connaissances éthologiques sur l'espèce chassée que les stratégies de prise de décision des chasseurs. Ce modèle prend également en compte l'hétérogénéité spatiale, en utilisant des informations sur la nature du sol provenant d'un SIG, dans la mesure où ces informations sont pertinentes pour le déroulement des simulations. La modélisation multi-agent nous a donc permis d'**intégrer des connaissances pluridisciplinaires (écologie, sociologie, économie) et de fournir un cadre conceptuel pour comprendre la gestion villageoise de la faune sauvage à Djémiong.**

Nous avons ensuite procédé à plusieurs séries de simulations de ce modèle. Les simulations de la dynamique hors chasse produisent des courbes d'évolution de la population qui s'apparentent aux dynamiques observées dans les zones non chassées. C'est ainsi que **nous validons en partie le modèle multi-agent**, puisque nous pouvons conclure que les connaissances éthologiques qui déterminent la dynamique de la population sont bien formulées dans notre modèle.

Les simulations mettent ensuite en oeuvre les différents scénarios de gestion identifiés lors des enquêtes et permettent d'étudier sur le long terme la viabilité de la faune. Nous testons l'impact d'une gestion « villageoise » de la faune et, dans la suite de l'analyse socio-économique qu'en fait Takforyan dans sa thèse, **nous utilisons les simulations multi-agents pour montrer que les modes de gestion locale de la faune sauvage constituent un moyen de préservation de l'espèce.**

Pour implémenter ce modèle multi-agent et pour mener à bien les simulations, **nous avons développé une plate-forme de simulation multi-agent orientée vers les problèmes de gestion de ressources naturelles.** L'originalité de la plate-forme CORMAS réside dans le fait qu'elle **permet d'intégrer à la fois dans un modèle des dynamiques spatiales et des dynamiques sociales.** Lors de la définition de l'espace, le modélisateur a le choix entre une grille spatiale théorique dont il précise les caractéristiques (taille, forme des cellules, connexité, attributs et dynamique des cellules) et un espace issu d'une carte produite par un SIG. CORMAS permet de gérer par les attributs des cellules les différentes couches d'information que le SIG peut conserver sur une carte.

L'utilisateur de CORMAS peut également suivre la dynamique des groupes sociaux. Les échanges de messages ou de biens entre les agents sont traduits par un graphe dynamique

qui permet ainsi de suivre pendant la simulation l'évolution des groupes. Des primitives d'interactions entre les groupes et leurs composantes permettent au modélisateur de gérer des changements d'échelle dans le modèle.

Nous pouvons aussi souligner que CORMAS est d'une extrême souplesse et d'une grande facilité d'utilisation, au regard du temps (une semaine) que mettent des chercheurs non informaticiens à prendre en main la plate-forme et à développer en toute autonomie leurs propres modèles. Nous en avons fait l'expérience lors des diverses sessions de formation à CORMAS qui sont régulièrement organisées par le CIRAD à l'intention des chercheurs en agronomie.

nous avons poursuivi notre travail en abordant

la question de la validation formelle des modèles multi-agents. A cet effet, nous avons pris contact avec le Laboratoire d'Informatique de Paris VI (Université Pierre et Marie Curie) qui abrite une équipe renommée dans la recherche sur les Systèmes Répartis Coopératifs.

Comme nous l'avons vu plus haut, l'objectif poursuivi par la gestion des ressources renouvelables est d'assurer la viabilité de celle-ci. Les simulations multi-agents que nous avons menées permettent d'étudier le comportement du modèle sur un temps fini. Au moment où on arrête une simulation, rien ne nous permet d'affirmer que la tendance qui s'affiche serait maintenue si on poursuivait la simulation. Nous nous sommes interrogés sur la validité des résultats obtenus dans le cas où on considère le très long terme.

Bien que les équations différentielles nous offrent d'énormes possibilités d'analyse formelle, il est très difficile de traduire le modèle multi-agent dans ce formalisme sans le dénaturer totalement. Nous avons choisi comme formalisme pour la validation du modèle multi-agent les réseaux de Petri. C'est à la fois un formalisme mathématique et un langage graphique permettant de modéliser les systèmes complexes. Ils nous semblent être un bon compromis entre les systèmes multi-agents qui offrent un riche pouvoir d'expression mais ne permettent pas de vérification formelle, et les modèles d'équations différentielles très adaptées à une analyse formelle, mais inefficaces lorsqu'il faut représenter des entités autonomes en interactions ou des comportements qualitatifs. Les réseaux de Petri offrent les avantages de ces deux approches extrêmes.

La viabilité de la ressource se traduit dans le modèle réseau de Petri par la non existence de configuration dans le graphe des marquages accessibles correspondant à une taille nulle de la population. **Nous avons ainsi montré que le système est viable sous certaines hypothèses. Cette étude, en proposant des techniques de formalisation de phénomènes liés à la référence spatiale comme les déplacements des animaux, est aussi une contribution à l'étude formelle des systèmes multi-agents spatialisés.**

0.5 Plan de la thèse

Dans le cadre de cette thèse, nous avons étudié un système de gestion « villageoise » de la faune en nous appuyant sur la théorie des systèmes multi-agents. Ce travail s'est fait en alternance au Département d'Informatique de la Faculté des Sciences de l'Université de Yaoundé I et au sein de l'équipe GREEN (Gestion des Ressources rEnouvelables et ENvironnement) du CIRAD à Montpellier. Notre contribution porte essentiellement sur trois points :

- la mise en oeuvre d'un environnement de simulations multi-agents ;
- la modélisation et la simulation multi-agents d'un phénomène complexe : la gestion d'une ressource naturelle renouvelable ;
- la formalisation de modèles multi-agents par les réseaux de Petri en vue de valider les résultats des simulations multi-agents.

Cette thèse comporte donc trois parties subdivisées en six chapitres.

Le premier chapitre présente le domaine des systèmes multi-agents en retraçant l'histoire de cette approche et en précisant les grands axes de recherche actuels. Il insère notre travail dans l'optique des « simulations multi-agents » vues comme l'exécution d'un modèle multi-agent dans un environnement traditionnel de programmation séquentielle. La théorie des univers multi-agents est aujourd'hui utilisée dans de très nombreux domaines. Nous en citons quelques uns et nous établissons leur apport dans le domaine de la gestion des ressources renouvelables en montrant en quoi la spécificité des problèmes dans ce domaine permet de bénéficier des concepts de l'approche multi-agent.

Le chapitre II est consacré à la plate-forme de simulation multi-agent CORMAS. Nous avons développé cet environnement au sein de l'équipe Green du Cirad pour répondre à un besoin d'outils adaptés à la modélisation des systèmes de gestion de ressources naturelles renouvelables. Il est aujourd'hui utilisé par des dizaines de chercheurs à travers le monde et un site web (<http://cormas.cirad.fr>) ainsi qu'une liste de discussion (cormas@cines.fr) lui sont dédiés.

L'environnement CORMAS est utilisé pour modéliser et simuler par une approche multi-agent l'activité de chasse dans la forêt villageoise de Djemiong à l'Est-Cameroun. Cette étude fera l'objet des chapitres III et IV de cette thèse. Nous nous sommes appuyés sur des données recueillies lors des enquêtes sur le terrain par Ani Takforyan [Takforyan, 2001] pour élaborer un modèle multi-agent du phénomène. Plusieurs séries de simulations sont effectuées à partir des divers scénarios de gestion de la ressource que nous avons identifiés, et les résultats de ces simulations sont discutés au regard des observations faites sur le terrain.

Dans le chapitre V, nous présentons une modélisation de la chasse par les Réseaux de Petri. C'est surtout un essai de formalisation du modèle multi-agent de l'activité de chasse présenté plus haut. A partir d'un groupe d'hypothèses extraites du modèle multi-

agent, nous construisons des modules de réseaux de Petri qui modélisent chacun une partie du système multi-agent. Les modules sont assemblés par fusion des places. Enfin, nous analysons le modèle en étudiant les propriétés structurelles du réseau ainsi que le graphe des marquages accessibles du système.

Le chapitre VI présente l'analyse formelle du modèle réseau de Petri qui nous a permis de valider les résultats de la modélisation multi-agent en montrant que sous certaines hypothèses, le modèle formel confirme les tendances obtenues lors des simulations.

Nous concluons cette thèse par une étude comparative des deux expériences de modélisation, ce qui permet de mettre en évidence les forces et les faiblesses de chacune des approches. Nous proposons quelques directives susceptibles de guider le modélisateur dans le choix de l'approche qui conviendrait le mieux à une situation donnée.

Première partie

LA MODELISATION MULTI-AGENT

Chapitre 1

Les systèmes multi-agents

1.1 Introduction

Les systèmes multi-agents se situent dans le prolongement de l'Intelligence Artificielle Distribuée (IAD) qui se définit comme l'étude et la conception d'organisations d'agents artificiels pour obtenir des systèmes intelligents [Demazeau and Müller, 1991]. Les systèmes multi-agents se veulent de portée plus générale que l'IAD, puisqu'ils concernent la résolution de problèmes au sens large, la robotique distribuée, la simulation et la conception de mondes hypothétiques [Ferber, 1995].

De la notion de système multi-agent se dégage immédiatement l'idée d'un système qui est fait de plusieurs agents. Le concept d'agent reste donc le pivot de ce domaine et l'interprétation qu'on a de ce terme doit être le premier pas dans l'exploration des univers multi-agents. La façon dont les agents interagissent entre eux permettra ensuite de cerner comment les systèmes multi-agents sont construits.

1.2 La notion d'agent

La notion d'agent s'apparente beaucoup au concept classique d'objet qui est une structure encapsulant les données (attributs) et les procédures (méthodes). Beaucoup de concepts de l'approche agent peuvent d'ailleurs être mis en oeuvre par des concepts objet [Boulanger et al., 1995]. La programmation objet, basée sur le paradigme

« donnée + méthodes = objets »

se distingue clairement du schéma de la programmation structurée, traduite par

« programme = données + procédures ».

L'agent étend l'objet en lui ajoutant les notions d'autonomie et d'objectifs propres. Plus concrètement, lorsqu'un agent reçoit un message, contrairement à l'objet qui active simplement la méthode associée si elle a été implémentée lors de sa définition, l'agent met

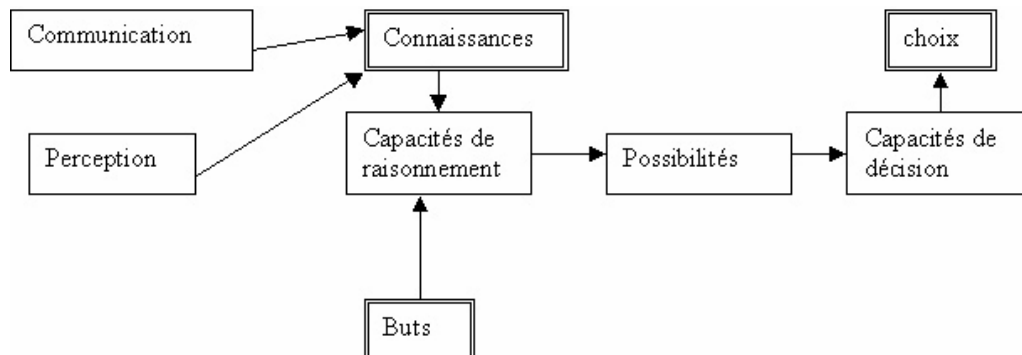


FIG. 1.1 – Modèle d'agent générique

en oeuvre avec ses accointances un véritable protocole de communication. Il délibère de façon interne et décide ou non de réagir au message, et éventuellement de la qualité de la réponse. Wooldridge caricature très bien cette différence entre agents et objets par la boutade « *objects do it for free, agents do it for money* » [Wooldridge, 1999].

Le modèle qui se rapproche le plus de l'agent est le modèle d'acteur qui est une extension du modèle d'objet par ajout de la notion d'activité. En effet, un acteur est un objet actif, autonome, communiquant par envoi de messages asynchrones avec les autres acteurs [Labidi and Lejouad, 1993].

Demazeau et Muller proposent un modèle d'agent générique en exhibant les caractéristiques minimales qui permettront d'expliquer les diverses interactions dans un univers multi-agent [Demazeau and Müller, 1991]. L'agent se trouve ainsi explicité en terme de buts, de connaissances, de perceptions, de choix, de capacités de raisonnement et de décision, de communications. Ce modèle est résumé dans la figure 1.1.

Dans ce modèle, les connaissances de l'agent sont en général un modèle de l'environnement, et peuvent inclure un modèle de l'agent lui-même. Les plans ou chemins possibles vers une solution sont générés à partir de ses connaissances et de ses capacités de raisonnement. Le choix d'un plan parmi les plans possibles est lié à ses capacités de décision.

Cependant, la définition la plus couramment admise de la notion d'agent est celle proposée par Ferber dans l'un des tout premiers livres de référence sur le domaine [Ferber, 1995]. Un agent y est défini comme une entité physique ou virtuelle qui

- est capable d'agir dans un environnement
- peut communiquer directement avec d'autres agents
- est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie qu'elle cherche à optimiser)
- possède des ressources propres
- est capable de percevoir (mais de manière limitée) son environnement
- ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune)

- possède des compétences et offre des services
- peut éventuellement se reproduire
- a un comportement tendant à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Il s'en suit que l'agent se définit généralement par rapport à un environnement. Un agent peut ainsi modéliser une voiture dans un système d'étude du trafic routier, un individu ou un groupe d'individus dans un système représentant une société, un jeton dans un jeu de dames.

L'environnement est assimilé à l'univers dans lequel évoluent les agents. Il peut être simplement un espace métrique pour des agents situés (nous reviendrons sur cette notion) ou un environnement d'exécution (variables d'environnement) pour des agents représentant des processus.

Wooldridge [Wooldridge, 1999] propose un cadre formel dans lequel les notions d'agent et d'environnement sont définis sans aucune ambiguïté. Ainsi, un environnement est caractérisé par un ensemble

$$S = \{s_1, s_2, \dots\} \quad (1.1)$$

d'états (de l'environnement). Etant donné un ensemble

$$A = \{a_1, a_2, \dots\} \quad (1.2)$$

d'actions possibles, un agent se définit comme une fonction

$$action : S^* \longrightarrow A \quad (1.3)$$

où

$$S^* = \bigcup_{i=0}^{+\infty} S^i \quad (1.4)$$

est l'ensemble des historiques ou des suites d'états de l'environnement. A chacune de ces suites, la fonction *action* associe une action particulière de l'agent. Cette approche fonctionnelle de l'agent offre des outils appropriés pour caractériser formellement les typologies d'agents et d'environnements.

1.3 Typologies d'agents

En fonction du degré de détail des connaissances d'un agent ou de la complexité de ses fonctionnalités (on parlera aussi de granularité d'un agent), on distingue dans les systèmes multi-agents deux principales catégories d'agents : les agents réactifs et les agents cognitifs. Une autre typologie des agents est liée à leurs relations à l'environnement spatial dans lequel ils évoluent et à leurs interactions avec les autres agents : on parlera d'agents situés et d'agents communicants.



FIG. 1.2 – Des agents cognitifs

1.3.1 Agents cognitifs vs agents réactifs

Les systèmes d'agents cognitifs sont fondés sur la coopération d'agents capables à eux seuls d'effectuer des opérations complexes [Labidi and Lejouad, 1993]. Les agents sont donc dotés d'une grande capacité de perception, de raisonnement et d'action. Ils sont assez proches des systèmes experts de l'Intelligence Artificielle classique, dans leur aptitude à manipuler la connaissance

Chaque agent dispose d'une base de connaissances comprenant l'ensemble des informations et du savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec les autres agents et avec son environnement. Les agents cognitifs sont aussi dits intentionnels car ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts [Ferber, 1995]. On assimile la notion d'agent cognitif à la conception sociale des agents.

Les agents réactifs par contre sont perçus comme une approche plutôt biologique de la notion d'agent. On part ici du principe que dans les SMA, il n'est plus nécessaire que chaque agent soit individuellement intelligent pour parvenir à un comportement global intelligent. Cette approche propose donc la coopération d'agents de faible granularité qui seraient beaucoup plus nombreux, ne disposant que d'un protocole et d'un langage de communication réduits et dont les capacités répondent uniquement à la loi stimulus/action [Labidi and Lejouad, 1993]. Par des mécanismes de réaction aux événements, ne prenant en compte ni une explicitation des buts, ni des mécanismes de planification, les systèmes d'agents réactifs peuvent ainsi résoudre des problèmes complexes.

C'est par exemple le cas dans un jeu de taquin (figure 1.3) où chaque agent, correspondant à une case, met en oeuvre un protocole très simple : si une case voisine est vide et qu'un agent souhaite l'occuper, il s'y déplace. Nous reviendrons sur cet exemple dans le cadre de la résolution des problèmes par une approche multi-agent (section 1.5.3.1 ci-dessous).

En réalité, les systèmes mettent en oeuvre généralement des agents dits hybrides, qui

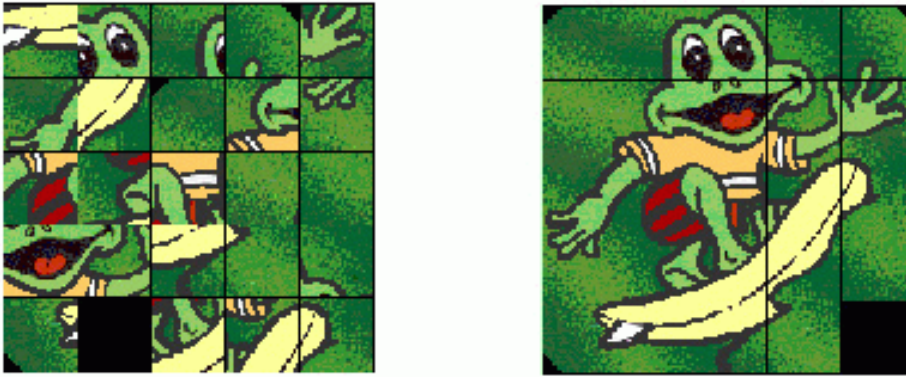


FIG. 1.3 – Les agents réactifs dans un jeu de taquin

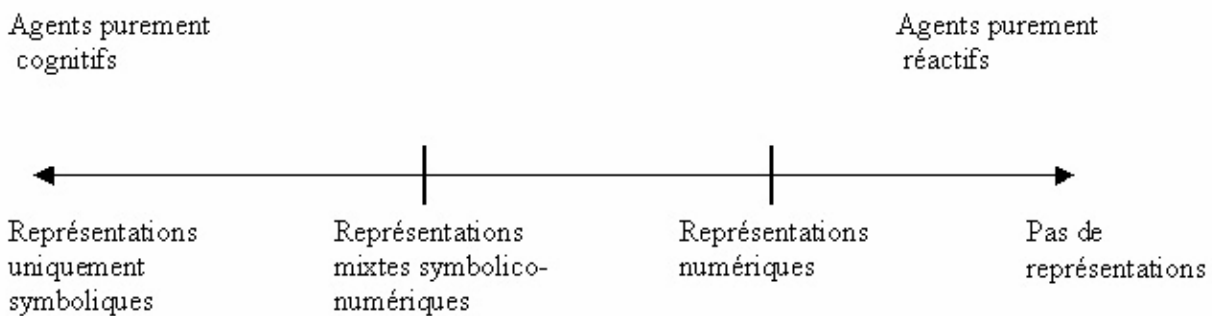


FIG. 1.4 – Des agents purement réactifs aux agents purement cognitifs, un continuum

se situent entre les agents réactifs et les agents cognitifs et qui intègrent plus ou moins les caractéristiques de l'une ou l'autre des tendances. On peut considérer que les agents purement cognitifs et les agents purement réactifs représentent deux écoles situées aux deux extrémités d'une large gamme d'architectures possibles pour la conception des agents (figure 1.4).

Si l'on reprend la définition fonctionnelle de l'agent par $action : S^* \rightarrow A$, un agent purement réactif est défini comme un agent dont les actions ne prennent en compte que l'historique immédiat (d'ordre un), c'est-à-dire l'état courant de l'environnement. La fonction caractéristique d'un agent réactif sera équivalente à sa projection sur S . Un agent purement réactif est défini par une fonction $action : S \rightarrow A$.

Par contre, le comportement d'un agent cognitif est déterminé par l'historique des états de l'environnement. Cela suppose que l'agent cognitif intègre un processus plus complexe de prise de décision. De façon plus naturelle, soit P un ensemble non vide de percepts, on peut considérer une fonction de perception

$$see : S \rightarrow P \quad (1.5)$$

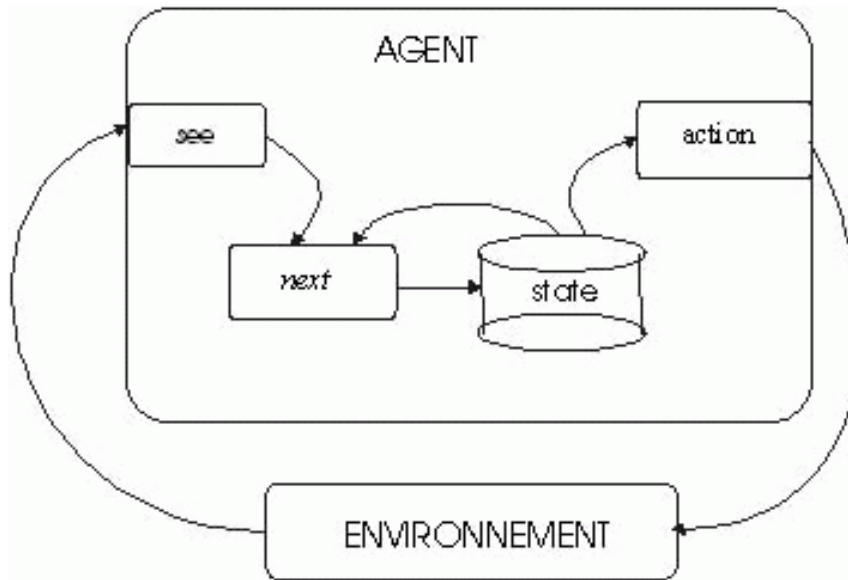


FIG. 1.5 – un agent à états

qui associe à chaque état de l'environnement un percept. Le percept ici correspond à la perception (partielle) que l'agent a de l'environnement. Soit par ailleurs I l'ensemble des états internes de l'agent, une fonction

$$next : I \times P \longrightarrow P \quad (1.6)$$

associée à chaque agent décrit l'évolution de son état interne. L'agent (cognitif) pourrait donc être décrit par une fonction

$$action : I \longrightarrow A \quad (1.7)$$

Cette représentation est équivalente à la description de l'agent par $action : S^* \longrightarrow A$, et présente l'avantage de se rapprocher plus d'une vision naturelle de l'agent. En effet, l'agent perçoit de façon partielle son environnement (*see*) et conserve dans son état interne une trace de l'historique du système. Cet état interne évolue en fonction de nouvelles perceptions (*next*). L'agent sélectionne une action simplement en considérant son état interne.

Cette vision d'un agent est très proche de la théorie des automates finis. Un automate fini est défini par la donnée d'un alphabet A , d'un ensemble fini d'états E , d'une relation de transition, sous-ensemble de $E \times A \times E$, d'un état initial $I \in E$ et d'un ensemble d'états finaux $F \subseteq E$. Ainsi, un agent à états peut être vu comme un automate fini enrichi, dans lequel les symboles de l'alphabet d'entrée sont assimilés à des perceptions de l'environnement, et les transitions d'états portent des actions sur l'environnement.

1.3.2 Agents situés vs agents communicants

Une classification courante des types d'agents est celle qui est fondée sur leur rapport à l'espace (agents situés) et aux autres agents (agents communicants).

Comme nous le verrons plus loin, un des éléments clés d'un SMA est l'environnement. Cet environnement est souvent un espace métrique et la donnée des références spatiales d'un agent est pertinente pour la modélisation d'un grand nombre de phénomènes : planification de trajectoire en robotique, détection d'indices visuels complexes en analyse d'images, détection de contours en vision artificielle, localisation spatiale d'aménagements dans un territoire, généralisation cartographique etc. Ces problèmes seront alors dits spatialisés [Ferrand et al., 1998].

Les agents sont également caractérisés par leurs rapports aux autres agents. Lorsqu'un agent possède des compétences ou des services qu'il peut offrir aux autres agents, ou lorsqu'un agent doit solliciter pour satisfaire un but, des compétences ou des services d'autres agents, il s'avère nécessaire d'établir entre eux des communications. La communication est à la base des interactions directes entre les agents dans un système multi-agent. Un agent communicant sera donc un agent disposant des aptitudes à échanger, (c'est-à-dire envoyer et recevoir) des messages avec d'autres agents. Un agent n'a généralement qu'une perception partielle du système. Il ne lui est donc pas souvent possible de communiquer avec tous les autres agents du système. Dans un système multi-agent communicant, chaque agent dispose d'un ensemble d'accointances, terme désignant l'ensemble des autres agents du système avec lesquels il est susceptible de communiquer. De plus, communiquer implique la mise en oeuvre de langages et de protocoles appropriés.

1.4 Les architectures d'agents

L'architecture d'un agent définit la manière dont il est conçu, c'est-à-dire son organisation interne. Les applications utilisant l'approche multi-agent étant de plus en plus nombreuses, on voit émerger dans la communauté des architectures d'agents, plus ou moins dédiées à des systèmes particuliers. Bien que ce ne soit pas encore le cas, l'enjeu est d'aboutir à terme à une véritable méthodologie normalisée de conception et d'implémentation des systèmes à base d'agents. Nous présentons quelques unes des architectures les plus courantes.

1.4.1 Les architectures réactives

La modélisation par agents réactifs s'oppose à l'approche des agents qui raisonnent en manipulant des informations symboliques, tels qu'on en rencontre dans les systèmes de planification ou de résolution de problèmes. Dans les systèmes d'agents réactifs, l'accent est

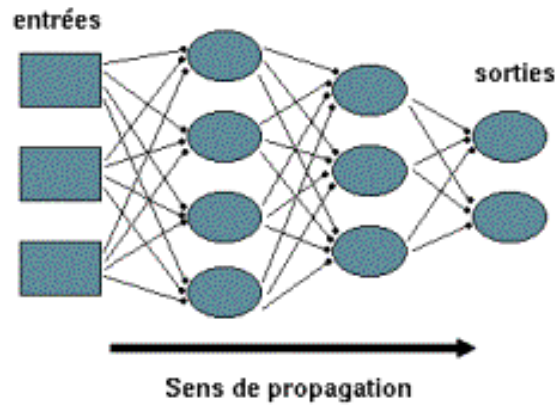


FIG. 1.6 – Exemple de réseau de neurones

mis sur la réalisation de tâches élémentaires par chaque agent, les interactions entre ceux-ci devant conduire à l'émergence d'un comportement global intelligent. Plusieurs architectures ont été proposées pour mettre en oeuvre les agents réactifs [Ferber, 1994].

Un agent réactif peut être décrit par un ensemble fini de règles simples qui lient les perceptions de l'agent à ses actions. Ces règles sont de la forme

```
if <perceived situation>
then <specific action>
```

On peut aussi représenter les agents réactifs par des automates à états finis, c'est-à-dire en décrivant l'ensemble (fini) de leurs états et leur fonction de transition. La fonction de transition d'un automate spécifie l'état suivant en fonction de l'état actuel et des informations qu'il reçoit en entrée.

Les réseaux de neurones (figure 1.6) permettent également de décrire le comportement d'un agent réactif. Un neurone peut être décrit comme un petit automate de décision. Il se compose de 2 parties ayant des fonctions distinctes : évaluation de la stimulation reçue et évaluation de son état interne. Un neurone est donc caractérisé par : son état, le niveau d'activation qu'il reçoit en entrée, sa fonction de transition, sa fonction d'entrée. L'état que peut prendre un neurone est en général binaire mais peut aussi bien être une valeur discrète ou continue. Selon le type de cet état, on aura donc différentes formes pour la fonction de transition. Les neurones doivent être interconnectés pour former un réseau de neurones.

La topologie des réseaux de neurones peut être très variée. On peut concevoir plusieurs types de réseaux seulement en modifiant les règles de connexion. Dans le modèle le plus classique, les neurones sont organisés en couches successives. La particularité de la connexion est d'être pondérée. On simule ainsi la relation entre les neurones, c'est à dire l'influence de l'activation (désactivation) d'un neurone sur un autre. Dans la majorité des cas, les connexions servent à calculer l'entrée du neurone tandis que les fonctions

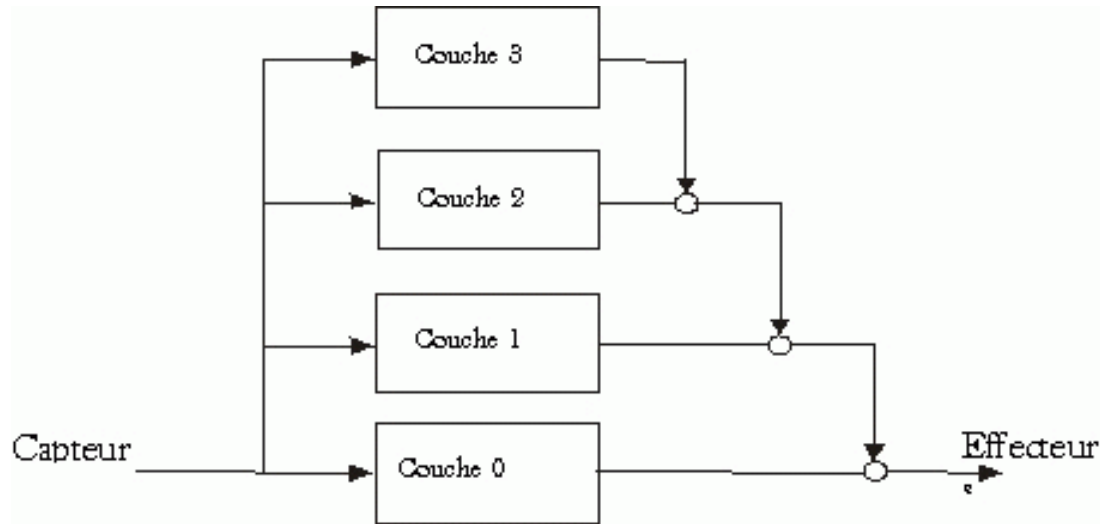


FIG. 1.7 – L'architecture de subsumption

d'activation détermineront la sortie.

1.4.2 L'architecture de subsumption

L'architecture de subsumption (figure 1.7) est très proche des architectures réactives, et certains auteurs la cite comme telle. Introduite par Brooks [Brooks, 1991], cette architecture suggère de décomposer un agent en modules verticaux, chacun d'eux n'étant responsable que d'un type de comportement très limité. Ces modules effectuent leurs tâches en parallèle et il existe une priorité dans les modules, telle que lorsque deux modules fournissent des résultats contradictoires, seuls ceux du module dominant soient pris en compte. La figure 1.8 extraite de [Ferber, 1995] présente un exemple caractéristique d'architecture de subsumption pour un robot explorateur.

L'architecture de subsumption peut être étendue de manière à la rendre plus flexible. Jonathan Connell [Connell, 1990] propose une variante qui, en faisant toujours appel aux liens d'inhibition et de suppression entre tâches, définit plutôt un ordre de priorité partiel entre les modules, permettant ainsi d'avoir des comportements de même niveau hiérarchique. L'architecture de subsumption dynamique proposée par [Hideyuki and Itsuki, 1998] facilite la prise en compte de la dynamique de l'environnement de l'agent.

1.4.3 L'architecture BDI

C'est le modèle le plus utilisé pour décrire les agents intelligents. L'agent ne possède qu'une connaissance partielle de son environnement. Cependant, pour choisir une action, l'agent doit faire des hypothèses sur les éléments de l'environnement qu'il ne connaît pas. Le modèle BDI tente d'axiomatiser l'évolution du point de vue (croyances, désirs, intentions)

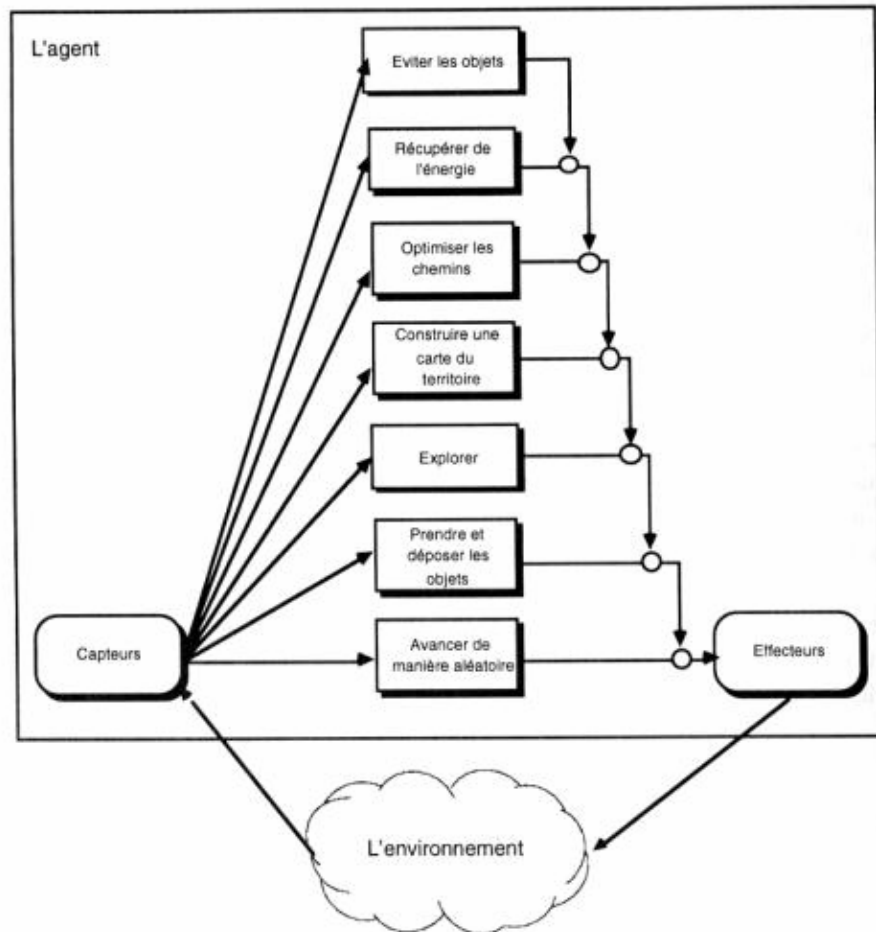


FIG. 1.8 – Modèle de robot explorateur à base d'architecture de subsumption

d'un agent en fonction du résultat de ses actions.

Le modèle du comportement d'un système à agents dans une approche BDI peut se définir incrémentalement comme suit [El Fallah et al., 1997] :

- un ensemble de proposition $X = \{p, q, \dots\}$ qui traduisent l'état du monde et l'état de l'agent
- un modèle du monde qui est l'ensemble des propositions vraies et des propositions fausses à un instant donné
- un ensemble d'événements A partitionné entre les actions de l'agent et celles (spontanées) de l'environnement
- un graphe d'états où chaque noeud est un modèle du monde et chaque arc reliant ces noeuds est étiqueté par une occurrence d'événement.

Selon l'approche BDI, chaque type d'attitude de l'agent est défini par l'ensemble des mondes que l'agent associe à son type d'attitude. On introduit dans ce modèle trois types d'attitudes :

- BELIEF : c'est l'ensemble des croyances de l'agent. Il décrit les mondes possibles dans lesquels l'agent croit être.
- DESIRE : c'est l'ensemble des buts de l'agent. Il définit les mondes dans lesquels l'agent atteint ses buts.
- INTENTION : c'est l'ensemble des choix de l'agent. Il définit les mondes où l'agent adopte une manière d'atteindre ses buts.

Une situation d'un modèle BDI est définie par un état du monde réel et par les attitudes courantes de l'agent. Lorsque l'agent effectue une action, l'état du monde est modifié et le résultat de cette action influence à nouveau les attitudes de l'agent.

L'intérêt du modèle BDI vient d'une part du fait qu'il est intuitif car les notions de croyances, désirs et intentions correspondent bien à l'acceptation courante que nous en avons, d'autre part de ce que cette approche offre une décomposition fonctionnelle simple de l'agent en un ensemble de sous-systèmes.

1.4.4 Les architectures logiques

Elles correspondent à l'approche traditionnelle en Intelligence Artificielle. A partir d'une représentation symbolique des faits et des règles de raisonnement, un système est capable par des manipulations syntaxiques de ces représentations de produire un comportement dit « intelligent ».

Cette architecture peut être utilisée pour définir des agents dans les problèmes de planification. Un exemple connu est celui du monde des cubes [Ferber, 1995]. Le problème consiste à empiler des cubes les uns sur les autres de manière à obtenir un emplacement précis à partir d'un emplacement initial en ne bougeant qu'un seul cube à la fois (figure 1.9).

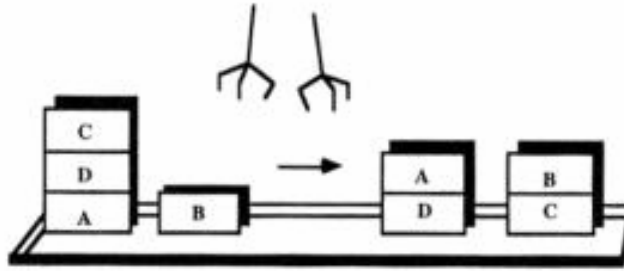


FIG. 1.9 – Un exemple de manipulation de cubes

Ces situations sont définies simplement sous la forme d'un ensemble de formules atomiques :

Sinit = {sur(C,D), sur(D,A), sur(A,Table), sur(B,Table)}

Sfin = {sur(A,D), sur(D,Table), sur(B,C), sur(C,Table)}

Les opérateurs correspondant aux deux actions classiques d'empilement d'un cube sur l'autre et de dépôt du cube sur la table peuvent s'écrire ainsi :

```
Opérateur poser(x,y)
  pré : libre(x), libre(y), sur(x,z)
  suppr : libre(y), sur(x,z)
  ajouts : libre(z), sur(x,y)
fin
```

```
Opérateur poserTable(x)
  pré : libre(x), sur(x,y)
  suppr : sur(x,y)
  ajouts : libre(y), sur(x,Table)
fin
```

1.5 Notion de système multi-agent

1.5.1 Définition

Un agent est amené à interagir avec d'autres agents et éventuellement avec son environnement. C'est l'ensemble de ces éléments (agents, environnement et interactions) qu'on appellera système multi-agent. Plus précisément, on appelle système multi-agent [Ferber, 1995] (ou SMA) un système composé des éléments suivants :

- un environnement E , c'est-à-dire un espace disposant généralement d'une métrique
- un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que pour tout objet, il est possible à un moment donné d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents
- un ensemble A d'agents qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système
- un ensemble de relations R qui unissent les objets (et donc les agents) entre eux
- un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O
- des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Il faut noter que tous les systèmes n'auront pas nécessairement besoin d'être décrits avec tous ces éléments. Il arrive par exemple que les ensembles E ou $O - A$ soient vides dans des situations où il n'est pas nécessaire de prendre en compte un environnement pour les agents, ou qu'il n'existe pas dans le systèmes des objets inertes. On parlera d'ailleurs lorsque E est vide d'un système multi-agent purement communicant.

Les relations R qui unissent les objets entre eux ainsi que les opérateurs Op recourent deux notions clés de la théorie des SMA : les interactions et les communications.

Dans un système de gestion du trafic aérien par exemple, les agents seraient les avions, l'environnement étant l'espace aérien dans lequel la position de chaque avion est définie par ses coordonnées. Ces agents suivent des trajectoires et doivent mettre en oeuvre des stratégies pour éviter les collisions. Ils peuvent ainsi communiquer directement entre eux ou à travers une tour de contrôle.

1.5.2 Modes d'interaction

Les interactions sont au centre de la problématique des systèmes multi-agents car elles représentent la « glu » qui permet aux agents de produire ensemble le comportement attendu du système. On appelle situation d'interaction dans un SMA [Ferber, 1995] un ensemble de comportements résultant de regroupements d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles. Il s'en suit que le type d'interactions entre des agents dans un système sera fonction de la compatibilité de leurs buts, ainsi que de la disponibilité des ressources et de leurs aptitudes propres. On aura donc selon le cas des situations de collaboration (simple ou coordonnée), d'encombrement ou de compétition (individuelle ou collective).

La collaboration simple peut se mettre en oeuvre dans le cas d'un système multi-agent cognitif où un agent sollicite les services d'un autre agent pour atteindre son but. Elle est coordonnée dans un système représentant des activités industrielles, comme la fabrication

de produits. On pourrait rencontrer des situations d'encombrement dans un système de trafic aérien comme celui évoqué plus haut. La compétition se retrouve par exemple dans le cas d'un système de dynamique d'une population, où des agents ont besoin de ressources elles-mêmes en quantité limitée.

1.5.3 Les domaines d'application

L'approche multi-agent est une véritable révolution dans la conception des systèmes. C'est la raison pour laquelle les domaines d'application sont nombreux.

1.5.3.1 La résolution de problèmes

L'intelligence artificielle s'intéresse aux problèmes pour lesquels il est difficile de trouver une solution en un temps raisonnable par l'approche algorithmique classique. L'Intelligence Artificielle Distribuée et par la suite les SMA proposent une approche décentralisée de résolution des problèmes, qui consiste à répartir l'expertise sur plusieurs agents qui coopèrent pour trouver la solution du problème initial. La difficulté réside ici dans la décomposition des tâches. Cependant, cette approche offre sur certains problèmes comme le tri collectif ou le taquin de meilleurs résultats que les approches classiques [Drogoul, 1993].

Le taquin est un problème célèbre en Intelligence Artificielle car extrêmement simple à modéliser en terme d'états et de transition entre états, mais très difficile à résoudre du fait du grand nombre d'états possibles. C'est un jeu de patience qui consiste en un damier carré de dimension $n \times n$ contenant $n^2 - 1$ palets et un emplacement vide appelé le « blanc ». La seule opération autorisée consiste à déplacer un palet adjacent au blanc sur le blanc, la tâche globale étant de réarranger les palets pour obtenir une configuration globale souhaitée à partir d'une configuration initiale quelconque.

Pour résoudre ce problème par une approche multi-agent, on considère que les cases sont des agents qui ne sont satisfaits que s'ils se trouvent sur la case correspondant à la configuration désirée. Lorsqu'un palet n'est pas satisfait, il « agresse » les palets voisins et les pousse ainsi à se déplacer eux-mêmes. Par un processus d'auto-organisation, chaque palet se positionne après plusieurs déplacements sur la case qui lui est associée dans la configuration finale.

Les SMA peuvent également être utilisés dans la généralisation cartographique, procédé qui consiste à créer une carte à partir d'un lot de données géographiques, en tenant compte de la focalisation envisagée ainsi que des contraintes graphiques de représentation comme la symbolisation et la résolution graphique du support [Baeijs et al., 1995].

Il existe de nombreuses applications des SMA aux technologies liées au réseau Internet comme l'aide à la recherche d'informations sur le web [Jieh Hsiang, 1999] ou la médiation dans le commerce électronique [Carles and Dignum, 2001].

1.5.3.2 La robotique

Construire des robots capables de réaliser des tâches de plus en plus complexes, et parfois mal définies comme l'exploration d'un environnement incertain sont les enjeux majeurs de la robotique. Les systèmes multi-agents offrent à cette discipline une nouvelle façon de « penser » le robot. Plutôt que d'essayer de fabriquer un robot complexe, il serait plus aisé de concevoir plusieurs petits robots qui interagissent pour réaliser une tâche complexe.

C'est ainsi que dans son projet d'exploration de la planète Mars, la NASA a décidé d'utiliser des robots. Puisque la communication entre la terre et Mars met environ 5 mn par liaison radio, ce délai est bien trop grand pour télécommander un robot à partir de la terre. Il est donc nécessaire que ce robot soit autonome. Comme il est difficile de construire un robot suffisamment complexe pour être autonome et qui s'adapte à un environnement incertain, la NASA a envisagé la mise au point d'un grand nombre de petits robots très simples dans leurs capacités et bon marché [Ferber, 1995].

1.5.3.3 La simulation

La simulation suivant une approche multi-agent consiste à reproduire sur ordinateur un modèle multi-agent d'un phénomène et à l' « exécuter » pour étudier son évolution. C'est sur la simulation multi-agent que portera notre propos tout au long de cette thèse, c'est pourquoi nous la développons plus longuement au paragraphe suivant.

1.6 La simulation multi-agent

1.6.1 Principe

La simulation est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier expérimentalement certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. Le processus de simulation passe donc par l'élaboration d'un modèle. On distingue classiquement les modèles analogiques (qui reproduisent une maquette du phénomène à étudier) et les modèles mathématiques ou numériques dans lesquels les phénomènes sont représentés par des variables (variables-causes et variables-effets) et par des fonctions mathématiques qui les mettent en relation.

La simulation analogique est un exercice délicat dès lors que l'on cherche à simuler des domaines où subsistent de nombreuses inconnues. Elle nécessite du chercheur qu'il connaisse parfaitement la nature et la forme des phénomènes à modéliser afin de pouvoir les simuler par des dispositifs physiques analogues. Bien adaptée, dans l'esprit, à la simulation de systèmes complexes dont les éléments agissent et interagissent simultanément,

elle s'avère dans la pratique peu employée en dehors de certains secteurs très particuliers comme l'aéronautique.

Le succès de l'approche numérique s'explique d'une part par la possibilité qu'elle procure de prendre en compte des phénomènes aléatoires (par la méthode dite de Monte-Carlo, ou par des méthodes probabilistes semblables) et d'autre part par une maniabilité et une généralité bien plus importantes que celles offertes par l'approche analogique.

Les modèles mathématiques obligent à définir des paramètres agrégés qui ne sont pas directement observables sur le phénomène réel (comme le taux de mortalité d'une population par exemple) et dont la justification n'est pas toujours évidente.

L'approche multi-agent de la simulation (et donc de la modélisation) est radicalement différente. La modélisation d'un phénomène dans une perspective multi-agent se traduit par [Drogoul, 1993] :

- une décomposition du phénomène en un ensemble d'éléments discrets et autonomes dont les interactions reproduisent le phénomène
- la modélisation de chacun de ces éléments par un agent, en précisant éventuellement ses connaissances, ses capacités fonctionnelles, ses comportements et les modes d'interaction qu'il adoptera à l'encontre des autres agents
- la définition de l'environnement des agents, espace topologique et/ou temporel dans lequel les agents évoluent, ainsi que des lois qui le gouvernent et de ses réactions aux actions des agents.

1.6.2 Quelques simulateurs

Depuis quelques années, la simulation multi-agent est de plus en plus utilisée pour mieux appréhender des phénomènes dont la complexité ne pouvait pas jusqu'alors être cernée par les modèles classiques. Nous présentons à titre d'illustration quelques exemples de simulateurs : SIM-DELTA, SHADOC, SEALAB et MANTA. Les deux premiers ont en commun de s'intéresser aux mécanismes de gestion d'une ressource commune : les poissons dans SIM-DELTA et l'eau dans SHADOC. Le simulateur SEALAB teste l'influence des variations du milieu sur le comportement reproductif d'une population, et MANTA simule le fonctionnement d'une fourmilière pour comprendre comment l'organisation (sociale) émerge des comportements élémentaires (des individus). SEALAB et MANTA mettent en oeuvre une approche « vie artificielle ».

1.6.2.1 Le simulateur SIM-DELTA

Le simulateur SIM-DELTA a été réalisé par F. Bousquet [Bousquet, 1994] dans le but d'étudier la pêche dans le delta central du fleuve Niger au Mali. Il permet de simuler un écosystème soumis à une intensification d'exploitation. Un pêcheur dans SIM-DELTA est un agent cognitif qui a des croyances et élabore des stratégies (représentées par un système

de règles) pour exploiter les poissons. Ces derniers sont regroupés en bancs, et chaque banc de poissons est un agent caractérisé entre autres par sa taille (nombre de poissons) et son processus de migration.

Les expériences menées ont permis d'observer l'évolution des captures et la dynamique du peuplement au cours du temps. Ce simulateur est basé sur une architecture à base de tableau noir et prend en compte autant les facteurs biologiques et topologiques que les processus de prise de décision des pêcheurs.

Cette modélisation a montré l'importance du mécanisme de décision sur la dynamique des poissons. Elle a permis d'illustrer le fait que les stratégies fondées sur des critères de rationalité économique sont globalement moins performantes à terme que celles qui intègrent des règles d'accès à l'espace.

1.6.2.2 Le simulateur SHADOC

Le simulateur SHADOC (Simulateur Hydro-Agricole Décrivant les modes d'Organisation et de Coordination), réalisé par Olivier Barreteau [Barreteau and Bousquet, 2000] illustre l'exemple d'une recherche sur les modes de gestion de l'eau dans les périmètres irrigués de la zone de Podor sur le fleuve Sénégal. L'espace est découpé en parcelles appartenant aux paysans. Le simulateur permet de prendre en compte différentes règles d'accès à l'eau : le libre service (chaque paysan ouvre et ferme sa parcelle quand il en a envie) ou le tour d'eau séquentiel (les paysans ont accès à l'eau suivant un ordre prédéfini).

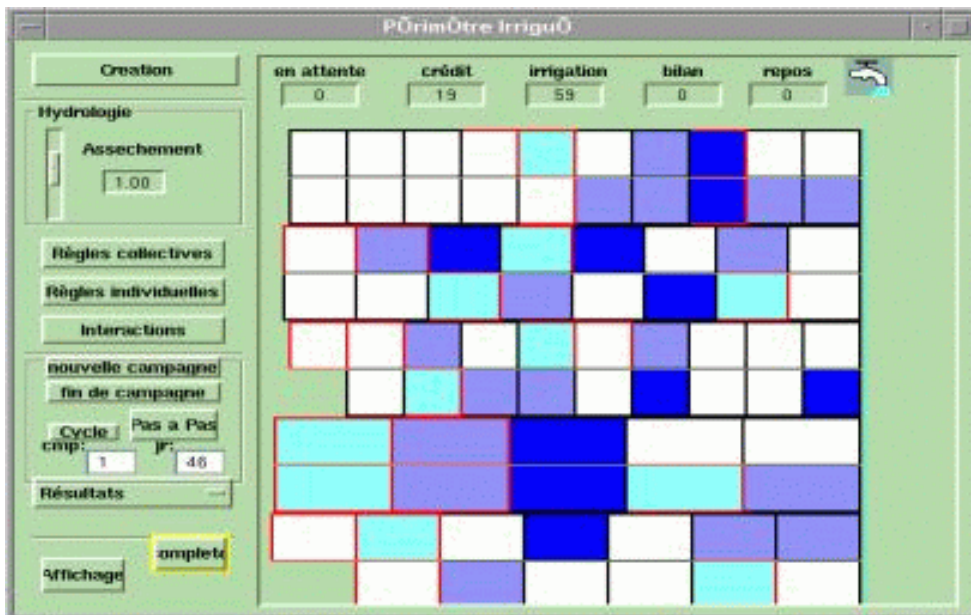


FIG. 1.10 – L'interface de SHADOC

En plus de ces deux situations extrêmes, on peut également considérer les contextes individuels dans lesquels s'inscrivent les processus de décision de chaque paysan et les

règles collectives mises en place pour adapter les deux règles extrêmes ci-dessus. C'est ainsi que des interactions directes entre les paysans comme des échanges d'informations ou des échanges de services leur permettent de mieux élaborer leurs décisions sur l'utilisation de l'eau. Les simulations des différentes règles ont permis d'observer dans chaque cas le nombre de parcelles à sec et de montrer l'importance du lien entre les divers niveaux d'organisation.

1.6.2.3 Le simulateur SEALAB

SEALAB (Sea Laboratory) [Le Page, 1996] est un outil de simulations individus-centrées, développé par Ch. Le Page pour tester l'influence d'un milieu hétérogène et fluctuant, s'exprimant par l'intermédiaire de comportements reproductifs, sur l'abondance d'une population. Il permet de suivre sur une grille les déplacements des entités représentant des poissons jusqu'au moment de leur reproduction. Chaque cellule de la grille est caractérisée par un indice qui traduit l'influence de l'ensemble des facteurs hydro-climatiques sur le processus reproductif. En tenant compte des fluctuations temporelles de la composition de l'environnement et de sa structuration spatiale, le simulateur aide à comparer les stratégies de reproduction des poissons sous l'angle de leur sensibilité à une hétérogénéité environnementale.

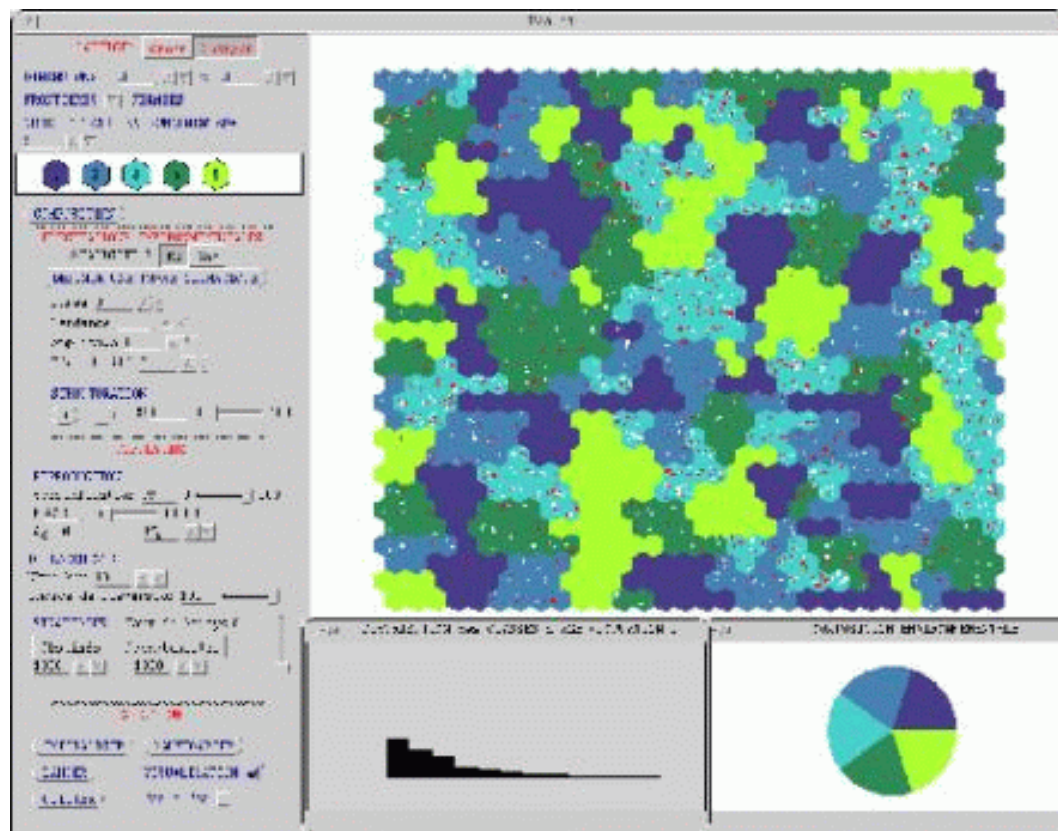


FIG. 1.11 – L'interface de SEALAB

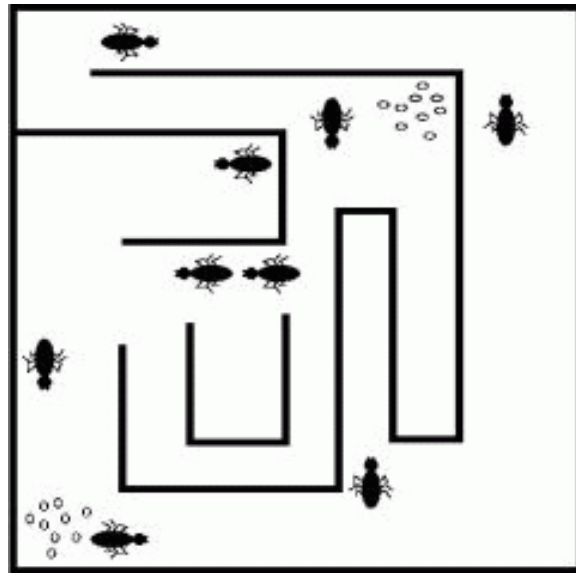


FIG. 1.12 – Les fourmis de MANTA

1.6.2.4 Le simulateur MANTA

L'objectif du projet MANTA (par Alexis Drogoul, [Drogoul, 1993]) était de coder dans des agents réactifs simples les comportements des fourmis et d'essayer d'obtenir par simulation des phénomènes émergents comme l'organisation sociale ou la division du travail, caractéristiques des colonies naturelles. Cette étude aidait à la compréhension des mécanismes individuels qui permettent par interaction, coopération et compétition avec les autres individus, de générer la structure sociale stable que l'on observe chez les fourmis, alors même que ces fourmis n'ont pas individuellement les capacités de penser le niveau social.

L'intérêt majeur de ce travail est que le paradigme de la fourmilière pourra par la suite être reproduit en robotique. On peut ainsi réaliser par exemple des milliers de petits robots très simples qui interagissent entre eux pour réaliser une tâche complexe comme l'exploration d'une planète, au lieu d'essayer de construire un robot complexe susceptible de tomber plus souvent en panne et incapable de s'adapter aux changements de son environnement.

1.6.3 Quelques extensions des SMA

1.6.3.1 Les SMA génétiques

Canal et Bah [Bah et al., 1998] ont proposé une extension des systèmes multi-agents vers les systèmes multi-agents génétiques, en utilisant les opérateurs de l'algorithmique génétique (reproduction, mutation et croisement) pour créer des populations d'agents qui, de génération en génération, évoluent entendant vers la population la mieux adaptée à un contexte donné. Cette approche est utilisée dans l'étude d'un phénomène dont l'impact

environnemental pose des problèmes cruciaux : la mobilité pastorale en zone intertropicale sèche. Les simulations ont ainsi permis de définir les caractéristiques en termes de mémoire, communication et stratégie, des troupeaux les mieux adaptés à l'environnement sahélien. Les auteurs montrent que la démarche peut être généralisée en l'appliquant au problème du voyageur de commerce [Bah, 2000].

1.6.3.2 Les SMA spatialisés

Plutôt qu'une extension, les SMA spatialisés se définissent mieux comme une spécialisation des SMA. C'est un courant qui est né de la rencontre des problématiques de la simulation multi-agent et des systèmes d'information géographique. L'enjeu est de proposer un cadre conceptuel pour mettre en commun les acquis des deux disciplines (représentation d'entités autonomes et simulation d'interactions d'une part, modélisation, stockage, extraction et représentation de données géo-référencées d'autre part). Tout ceci a pour but de donner de la dynamique aux cartes [Le Page et al., 1999].

1.7 SMA et Gestion de ressources renouvelables

Les problèmes de dynamiques environnementales sont complexes, en ce sens qu'ils prennent en compte plusieurs processus en interactions : dynamiques naturelles (écologiques, climatiques, ...) et dynamiques sociales (processus de prise de décision des acteurs politiques et sociaux).

Les situations de conflit dans la gestion des ressources renouvelables mettent en jeu des acteurs qui se font des représentations différentes de la même réalité et qui ont donc des modèles différents. Il existe entre ces perceptions un écart, une sorte de distance qui résume la difficulté de la négociation. Par exemple, dans le cas des périmètres irrigués, chacun des paysans a une représentation spécifique de sa parcelle et de ses besoins en eau. Les uns, qui voient en leur parcelle un simple accès au foncier peuvent être assez flexible sur le respect d'un calendrier d'irrigation tandis que les autres, pour qui leur parcelle est un moyen de production économique, seront plus rigoureux sur ce point [Barbault, 1992].

Le processus de négociation consiste pour les différents acteurs (politiques, sociaux, ...) à étudier ensemble les scénarios d'évolution de la ressource en fonction d'un corps d'hypothèses sur les modes de gestion. Ces hypothèses incluent également les comportements des différentes entités du système : dynamique des ressources, interactions, autant de concepts qui se prêtent très bien à une modélisation multi-agent. Dans ce contexte et plus particulièrement dans le cas de la gestion des ressources renouvelables, les SMA se révèlent être une approche très adaptée à la modélisation des différentes dynamiques.

Les SMA sont également utilisés dans un processus de modélisation d'accompagnement de la recherche, en suscitant un cycle d'observation – modélisation – simulation –

questionnement – observation [Bousquet et al., 1996]. On distinguera plusieurs étapes :

- Construire un monde artificiel.

Il s'agit d'élaborer un modèle dans un processus constructiviste qui passe par de nombreux allers et retours entre le terrain et la modélisation.

- La deuxième phase est une étape de restitution que l'on pourrait aussi appeler validation du modèle cognitif.

Il s'agit de tester le modèle proposé pour le processus de prise de décision. C'est la mise à plat des représentations et des processus d'interactions entre les agents. En effet il est difficile de faire comprendre ce qui « a été mis dans la machine ». Par contre il est possible de mettre l'acteur en situation de l'agent qui est dans la machine, avec les hypothèses de représentation, ses communications et les contrôles constitutifs du modèle. Pour mener cette opération à bien nous proposons de nous tourner vers la méthodologie du jeu de rôle expérimentée par les auteurs cités ci-dessus. On fait évaluer le monde artificiel en y plongeant les acteurs c'est-à-dire en créant un monde semblable au modèle. Ces acteurs peuvent être parties prenantes du système de gestion en tant qu'utilisateurs de la ressource (des paysans), régulateurs de cette gestion (gestionnaires ou administrateurs) ou observateurs du système (chercheurs). Le monde artificiel dans lequel sont plongés les acteurs ressemble-t-il au monde réel ? Cette étape peut participer à la phase d'initialisation de la démarche patrimoniale car elle permet de dresser une carte des différents types d'acteurs, des différentes perceptions et des interactions, et d'en faire un savoir partagé.

- Une troisième phase est celle de la simulation.

La simulation montre comment la dynamique du système est issue des interactions entre des acteurs qui ont des poids et des représentations différentes. On peut ici distinguer deux sous- phases. Dans un premier temps la simulation peut être effectuée sous forme de jeu de rôle, ce qui permet aux acteurs de valider le fait que c'est bien dans les interactions entre représentations différentes que se trouve le moteur de la dynamique du système. Cette première sous-phase permet aussi de faire émerger les différents scénarios intéressants à tester. Puis, cette phase étant acquise on peut utiliser le modèle multi-agent pour faire des simulations sous différents scénarios. Les simulations, tant "in ludo" que "in silico", participent ainsi à une autre phase de la démarche patrimoniale qui consiste, après avoir défini des objectifs de long terme, à envisager les différents scénarios qui pourraient y mener, à les tester et à en discuter les résultats.

1.8 Conclusion

Ce chapitre nous a permis de faire le point sur les systèmes multi-agents. Nous avons présenté les grands concepts qui définissent le domaine, les architectures d'agents ainsi que les typologies de SMA. Nous avons montré quel est l'intérêt de cette approche dans l'étude des phénomènes complexes.

Nous avons terminé par la problématique de la gestion des ressources naturelles renouvelables. La faiblesse des outils classiques utilisés pour aborder cette question a été mise en évidence et nous avons montré comment les systèmes multi-agents pouvaient être une approche plus efficace tant dans le cadre d'une démarche patrimoniale que comme outil d'accompagnement de la recherche.

Chapitre 2

La plate-forme de simulation CORMAS

2.1 Introduction

Les problèmes que pose la gestion des ressources naturelles renouvelables exploitées par l'homme sont complexes, en ce sens qu'ils résultent de plusieurs processus en interaction : ces ressources obéissent à des dynamiques naturelles contraintes par leur répartition spatiale et les acteurs sociaux (hommes ou groupes humains) qui les exploitent mettent en oeuvre différentes stratégies, allant de la coopération à la compétition.

Pour simuler la dynamique des systèmes complexes, l'approche multi-agent est depuis ces dernières années utilisée comme une alternative aux modèles mathématiques classiques. Elle permet de substituer aux paramètres globaux dont la justification n'est pas toujours évidente, des comportements locaux directement observables dans la réalité. Comme nous l'avons vu au chapitre précédent, cette approche consiste à représenter le système non pas en termes de variables et d'équations, mais plutôt en termes d'entités et d'interactions. La particularité de l'approche multi-agent est de mettre l'accent sur l'émergence de comportements globaux à partir de l'implémentation des comportements locaux et ainsi de ne pas présupposer le comportement global du système. Ce comportement global résulte plutôt des interactions entre les différentes composantes.

La communauté scientifique propose différentes plate-formes d'implémentation des modèles multi-agents. Nous citerons parmi les plus connues JavaNetAgents [Merlat and Seyrat, 1997], Madkit [Ferber and Gutknecht, 1998], Lalo [Gauvin, 1995] qui sont davantage orientées vers la communication entre des systèmes distribués. L'environnement Swarm développé au Santa-Fé Institute [Hiebeler, 1994] est un exemple de plate-forme orientée vers la simulation de systèmes, qui s'appuie principalement sur le concept d'automate cellulaire.

Dans le contexte de la gestion des ressources naturelles renouvelables, l'utilisation des plate-formes existantes pose de nombreux problèmes, notamment :

- elles se focalisent en général sur la mise en oeuvre des protocoles de communication entre les agents sur des systèmes distribués, laissant ainsi de côté les aspects liés à la

spatialisation des ressources, aspects clés dans la compréhension de la dynamique de celles-ci ;

- du fait de leur portée générique, elles obligent le modélisateur à manipuler les concepts des univers multi-agents, notions parfois difficiles à cerner pour des chercheurs non informaticiens.

Nous avons développé un environnement de simulation d'interactions entre dynamiques écologiques et sociales [Bousquet et al., 1998]. CORMAS (COMmon-pool Resources and Multi-Agent Systems) utilise l'approche des systèmes multi-agents pour la modélisation des différents concepts liés à l'usage des ressources renouvelables. Nous présentons dans la suite de ce chapitre la notion de simulation et les différents modèles de simulation, du point de vue de la prise en compte du temps. Ensuite, nous décrivons quelques plate-formes de simulation multi-agent, ainsi que les motivations de la conception de CORMAS. Pour éviter toute ambiguïté sur les termes, nous précisons notre compréhension des concepts utilisés dans l'environnement, puis nous proposons une démarche de construction de modèles pour CORMAS, ainsi que la procédure d'implémentation de ces modèles. Nous terminons le chapitre par quelques exemples d'applications développées sous CORMAS.

2.2 La simulation

2.2.1 Principe général

La simulation, comme nous l'avons déjà définie précédemment, est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier expérimentalement certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. On a recours aux techniques de simulation essentiellement dans les deux situations suivantes :

- soit l'on est dans l'impossibilité de recourir à l'expérimentation directe en raison de considération éthiques ou d'impossibilité techniques : un candidat aux élections n'a ainsi jamais la possibilité de mener de front dix campagnes différentes, de même qu'il n'est pas envisageable de construire dix autoroutes pour n'en garder qu'une ou de vérifier la portance des ailes d'un avion lors de son premier vol.
- soit l'on ne dispose pas de bases théoriques solides sur un phénomène donné et l'on cherche à élaborer une théorie, par voie de simulation, qui permette de rendre compte des données d'observation. Il s'agit ici d'une utilisation à posteriori du processus de simulation dans le but de formaliser, souvent de manière mathématique, un ensemble de propriétés du phénomène réel [Drogoul, 1993].

2.2.2 Modèles de simulations

Habituellement un système à simuler est caractérisé par un certain nombre de variables évoluant dans le temps. Cette évolution ne pouvant pas être toujours décrite par des équations mathématiques valables et calculables sur toute la durée recherchée, l'ordinateur va permettre d'effectuer les calculs en discrétisant le temps. Ainsi, s'il est possible d'obtenir pour une suite U_{t_n} d'états du système aux instants t_n , une fonction F telle que $U_{t_{n+1}} = F(U_{t_n}, t_n)$, et connaissant U_{t_0} alors il est possible de connaître, par calculs successifs, tout état du système U_{t_n} ; ce calcul par un ordinateur des états intermédiaires constituant la simulation.

La simulation d'un SMA est en général plus complexe à décrire. Ceci pour deux raisons : d'une part l'état du système ne peut pas être réduit à une simple compilation de « variables d'état » et d'autre part les informations recherchées comme résultat de la simulation peuvent ne pas se limiter à une liste de valeurs de variables préalablement établie (il peut y avoir eu, par exemple, création d'un nouvel agent...).

Dès lors il est difficile de calculer l'état d'un SMA à un instant t sans avoir décomposé ce calcul en utilisant des instants intermédiaires. D'où la nécessité d'effectuer une simulation ; ou plus exactement n simulations en parallèle, à chaque agent correspondant une simulation propre. Au niveau global, peu importe ce qui se passe à l'intérieur d'un agent entre deux de ses « actions », ainsi pour simuler un agent A_{g_i} , seule est nécessaire une fonction G_i capable de déterminer la prochaine action $A_{i,t_{n+1}}$ d'un agent et son état $E_{i,t_{n+1}}$ à la fin de cette action, en fonction des « perceptions » $P_{i,t_n,t_{n+1}}$ perçues par l'agent pendant ce laps de temps :

$$G : (Etat \times Perceptions \times Temps) \longrightarrow (Etat \times Action \times Temps) \quad (2.1)$$

avec

$$(E_{i,t_{n+1}}, A_{i,t_{n+1}}, t_{n+1}) = G_i(E_{i,t_n}, P_{i,t_n,t_{n+1}}, t_n). \quad (2.2)$$

2.2.2.1 Les systèmes à pas de temps constants

La simulation à pas de temps constants est très utilisée pour les simulations de SMA : au temps t toutes les actions – déduites des délibérations des agents – sont exécutées ; puis l'horloge est incrémentée d'une valeur constante (passage au temps $t + dt$), le calcul de nouvelles actions étant alors relancé en fonction de cette nouvelle heure.

Ce mode de simulation pose le problème de la définition a priori d'une unité de temps satisfaisante (i.e. suffisamment petite pour coller à des actions de durée minimale, suffisamment importante pour ne pas trop ralentir la simulation) et commune à toutes les entités du système qui, elles, peuvent être de granularités diverses et avoir des échelles temporelles différentes dans leurs dynamiques.

2.2.2.2 Les systèmes événementiels

L'idée est ici de partir directement des interactions entre les agents et leur environnement, c'est-à-dire faire évoluer la simulation d'interaction en interaction. Ce principe est déjà bien connu sous le vocable de « simulation à événements discrets ».

Le coeur de tout système événementiel est constitué par un échéancier d'événements, triés par leurs dates associées. Le simulateur possède également une horloge centrale. Un événement est composé d'une heure de déclenchement, ainsi que d'une procédure d'activation. Cette procédure permet de connaître les actions à effectuer lors du déclenchement de l'événement associé [Magnin, 1996].

2.3 Les plate-formes multi-agents

2.3.1 Madkit

MadKit (Multi-Agent Development Kit) est une plate-forme générique de conception et d'exécution de systèmes multi-agents, développée au LIRMM (Laboratoire d'informatique, de Robotique et de Microélectronique de Montpellier) par l'équipe de Jacques Ferber [Gutknecht et al., 2000]. Elle est basée sur un modèle organisationnel plutôt qu'une architecture d'agent ou un modèle d'interaction spécifique. L'utilisation de groupes et de rôles associés à des agents est mis en oeuvre tant en tant qu'outil de modélisation et de conception pour les développeurs de systèmes multi-agents, que de principe d'architecture de la plate-forme elle-même.

Cette architecture est basée sur un noyau agent minimal découplé de tout modèle individuel d'agent. Dans cette plate-forme, les services classiques de passage de message distribués, de migration ou de surveillance sont fournis au meta-niveau par des agents spécialisés afin d'obtenir un maximum de flexibilité. Une interface graphique componentielle et découplée du noyau et des agents permet de supporter différentes modes d'utilisation et d'exploitation de la plate-forme.

2.3.2 Dima

La plate-forme DIMA développée au LIP6 par Zahia Guessoum [Guessoum, 1998] propose de décomposer chaque agent en différents modules dans le but d'intégrer des paradigmes d'intelligence artificielle. Ces modules représentent les différents comportements d'un agent tels que la perception (interaction agent-environnement), la communication (interaction agent-agent) et la délibération. Un agent peut ainsi avoir un ou plusieurs modules qui peuvent être réactifs ou cognitifs. Un comportement réactif est décrit par un ensemble de méthodes (au sens objets) alors qu'un comportement cognitif peut avoir des méthodes qui activent des mécanismes de raisonnement. Pour gérer les interactions entre

ces différents comportements, DIMA propose un module de supervision représentant le méta-comportement de l'agent. Ce méta-comportement réifie le mécanisme de contrôle de l'agent, il gère les interactions entre les différents modules et permet à l'agent d'observer ces comportements.

Un agent dans DIMA est une entité pro-active et autonome. Cette architecture permet de dépasser la dichotomie classique en offrant la possibilité de développer des applications multi-agents dont la granularité des agents est variable. Elle offre ainsi la possibilité de concevoir et réaliser différents types d'agents :

- des agents réactifs, décrits par des comportements réactifs,
- des agents cognitifs, décrits par des comportements cognitifs,
- des agents hybrides, conçus pour allier des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de leur univers.

Un système multi-agent peut comporter un nombre quelconque d'agents hétérogènes (tailles différentes, mécanismes de décision/raisonnement différents, etc.). Des agents peuvent être créés dynamiquement et des agents peuvent également mourir parce qu'ils ont atteint leur but ou qu'ils n'ont plus de ressources. Ces agents sont également dotés de mécanismes pour adapter leurs comportements aux changements de leur univers. Un agent dans DIMA peut ainsi avoir une base de règles figée mais il peut également l'acquérir automatiquement en se basant sur sa mémoire et en utilisant un raisonnement à base de cas par exemple. D'autre part, pour optimiser le contrôle un méta-comportement utilisant les algorithmes génétiques a été introduit dans cette plate-forme.

2.3.3 Swarm

Le projet de la plate-forme Swarm a été lancé au Santa Fe Institute par Chris Langton en 1994, dans le but de fournir un environnement de simulation facilitant le développement de systèmes multi-agents. Son principe de base est la simulation des collections d'agents agissant concurremment l'un sur l'autre. Un agent dans Swarm est décrit par :

- une structure de donnée contenant ses variables d'état
- une fonction discrète (optionnelle) qui est exécutée à chaque pas de temps
- des fonctions définissant les actions de l'agent lors de la réception de messages en provenance d'autres agents.

2.4 Pourquoi CORMAS ?

Le besoin de mettre au point la plate-forme CORMAS est né du fait que les plate-formes existantes n'offraient pas toujours un cadre de modélisation adapté aux problèmes de gestion de ressources naturelles renouvelables. En effet, la prise en compte dans un même

système des dynamiques spatiales et des dynamiques des échanges sociaux est un des points clés que nous avons mis en oeuvre dans notre plate-forme. De plus, du fait de leur portée générique, les plate-formes multi-agents ne permettent pas au chercheur en écologie ou en socio-économie de manipuler des concepts liés à sa discipline. CORMAS s'est voulu dédié aux problèmes de gestion des ressources naturelles renouvelables.

Cet environnement offre une base pour la discussion des points suivants :

- Quel est l'impact de la répartition spatiale sur la dynamique des ressources ?
- Quelles stratégies de gestion peuvent permettre d'assurer une exploitation soutenable de ces ressources ?
- Quelles organisations sociales émergent de la dynamique des acteurs sociaux et quelle est l'influence de ces organisations sur l'exploitation de la ressource ?

La simulation doit permettre de mesurer ou du moins d'estimer la sensibilité de la dynamique du système à certains paramètres et par là d'aider à l'élaboration des stratégies de gestion des ressources naturelles renouvelables.

La plate-forme CORMAS permet d'implémenter des entités situées et/ou communicantes, en proposant des structures de gestion de la spatialisation et des communications. On peut observer à la fois la dynamique des réseaux sociaux (observateur graphique) et la dynamique spatialisée des ressources, ainsi que des statistiques sur la simulation. L'environnement est entièrement écrit en Smalltalk sous VisaulWorks. Smalltalk est également utilisé comme langage de mise en oeuvre des modèles sous CORMAS.

2.5 Les concepts de l'environnement CORMAS

Il nous semble utile de préciser notre entendement des concepts qui sont utilisés. Le but est d'une part de mettre en lumière les similitudes et les nuances pour lever toute ambiguïté et d'autre part de trouver un vocabulaire minimal (et donc facilement accessible) qui puisse unifier les langages du concepteur, du modélisateur et de l'utilisateur final.

2.5.1 L'espace

L'espace fera référence à l'espace physique (territoire, zone, région) sur lequel sont situés les agents. Il est l'environnement des agents. Cet environnement est distribué (non centralisé) en ce sens qu'il est constitué de cellules liées par des relations de voisinage.

2.5.2 La cellule

C'est la plus petite unité spatiale ayant une référence. Il s'identifie par cette référence (coordonnées, numéro). Certaines entités du système sont dites situées par le fait qu'elles sont à un instant donné positionnées sur une cellule (ou plusieurs quand on prend en compte



FIG. 2.1 – Les voisinages d'ordre un (ou immédiat) d'une cellule : la forme des cellules et la connexité de chaque cellule avec le voisinage sont variables

des organisations spatiales). Les entités se trouvant sur une même cellule à un instant peuvent interagir (agir, percevoir, communiquer) par le fait de leur proximité spatiale. Cette interaction peut s'étendre aux entités situés sur les cellules voisines (figure 2.1). La cellule offre ainsi à l'agent l'accès à son champ deperception.

Pour assurer ce rôle de média, d'intermédiaire entre ses occupants et entre ceux-ci et le voisinage, la cellule doit posséder une structure appropriée. Ainsi elle détient une liste de ses occupants et implémente des primitives permettant la communication entre eux. Elle offre aussi des primitives d'accès à son voisinage (immédiat ou étendu).

En tant que portion d'espace, la cellule est le lieu de la modélisation de la nature (ou de l'état) du milieu qu'elle représente. Elle possède donc un (ou des) attribut(s) d'état (eau, lieu-dit, forêt, ...). Les objets géographiques peuvent donc être modélisés par des attributs dans les cellules qu'ils recouvrent. En résumé, la cellule possède des informations sur ce qu'elle a (ses occupants) et sur ce qu'elle est (ses attributs).

2.5.3 Le canal

Le canal modélise une voie de communication entre des agents qui ne sont pas nécessairement (physiquement) voisins. On pourra même dire à ce propos qu'il introduit une notion de voisinage logique entre les agents. C'est l'image d'un groupe de discussion sur Internet. Il implémente des primitives de transmission (transmission point à point, transmission sélective, diffusion) de messages entre les agents qui lui sont connectés.

Il joue le rôle d'un module d'acheminement de messages et définit un champ de communication autant que la cellule définit un champ de perception. La cellule possède une sémantique, une propriété intrinsèque qui est sa position dans l'espace. Elle existe indépendamment de ses occupants et est pour eux un contenant. La sémantique du canal, par contre, est définie par le modélisateur. Ce peut être le type de messages véhiculés, une propriété sur les agents entre lesquels il peut établir une communication, ou un objet extérieur sur lequel portent les communications. Pour définir le canal, le modélisateur doit répondre à la question suivante : Qu'est-ce qui fait qu'un agent A puisse communiquer avec B, mais

pas avec C?

2.5.4 Les entités situées, les entités communicantes

Nous avons commencé à expliquer cette dichotomie à la section 1.3.2 du chapitre précédent. Une entité est dite située si sa position dans l'espace est une donnée pertinente dans le modèle. Les entités du système qui modélisent des objets, des ressources ou des êtres humains seront généralement situées par le fait que leurs dynamiques dépendent souvent de leur environnement physique. Lorsqu'une entité est déclarée située, cela entraîne qu'elle possède une référence spatiale qui est sa cellule. De plus, elle possède un attribut qui signifie l'étendue de son champ de perception.

Une entité peut, suivant les aspects qui sont pris en compte dans le modèle, être considérée tantôt comme située tantôt comme non située. C'est le cas par exemple d'un marché qui peut être non situé en tant que référence des équivalences entre les produits et leurs coûts d'une part, et d'autre part situé comme lieu d'échanges, ce qui entraîne la prise en compte des déplacements des agents pour s'y rendre.

Une entité est dite communicante quand elle peut percevoir, agir ou communiquer avec une autre entité qui ne lui est pas voisine. La notion d'entité communicante est assez proche de celle d'agent cognitif car les entités communicantes ont généralement besoin d'une délibération interne, ne serait-ce que pour la conception et le traitement des messages.

La communication ne doit pas être perçue ici seulement comme transmission d'information. Nous lui donnons le sens plus large de transmission, l'objet de cette transmission pouvant être une information ou un bien. Dans le contexte de la gestion des ressources renouvelables, les dons, les échanges, les négociations, les contrats sont autant de situations qui mettent en avant la communication entre des entités. Notons par ailleurs que les notions d'agents situées et communicantes ne sont pas exclusives l'une de l'autre. Certaines entités pourront vérifier simultanément les propriétés de ces deux classes et être à la fois situées et communicantes.

2.5.5 La dynamique

2.5.5.1 Dynamique de l'espace

L'espace qui modélise ici l'environnement physique des agents peut être sujet à une dynamique propre. Des phénomènes comme l'écoulement de l'eau, la propagation d'un signal ou d'un feu de brousse sont pris en compte comme faisant partie de la dynamique propre de l'espace, dans la mesure où ils ne résultent pas directement de l'action d'autres agents. Dans notre approche de modélisation, la dynamique de l'espace sera essentiellement locale. Cela veut dire que le comportement d'une cellule va dépendre de son état et de celui de son voisinage immédiat (ou étendu).

Le modèle d'espace est celui des réseaux d'automates cellulaires de dimension deux. La dynamique de la cellule spatiale correspond alors à la fonction de transition de l'automate. A la différence des automates cellulaires, l'évolution de la cellule sera généralement asynchrone.

2.5.5.2 Dynamique des agents

Chaque agent du système possède une dynamique propre qui se résume dans la description de son comportement à chaque pas de temps. D'une façon générale, un agent peut accomplir les actions suivantes :

- Réception des messages
- Délibération interne
- Mise à jour des variables de son champ d'action
- Envoi des messages

Les variables du champ d'action d'un agent regroupent ses attributs propres (comme son âge, sa taille), les attributs de la portion d'espace à laquelle il a accès ainsi que les attributs des autres agents du système qui sont modifiables de l'extérieur. Ces différents comportements doivent être pris en compte en fonction du type d'agent que l'on est en train de décrire. Ainsi, un agent réactif n'aura pas besoin de recevoir ni d'envoyer des messages. D'autre part, la phase de délibération interne d'un agent peut contenir des communications (nécessitant des réponses immédiates) avec d'autres agents.

2.5.5.3 Dynamique des canaux

Les canaux de communication doivent, en fonction de leur sémantique, être mis à jour régulièrement. La dynamique d'un canal est la mise à jour de sa population, c'est-à-dire de la liste des agents du système qui ont la possibilité d'utiliser ce canal. Si l'appartenance d'un agent à cette population est fonction des valeurs des attributs de l'agent, il sera nécessaire de reporter sur la structure du canal les modifications subies par ces valeurs au cours de la dynamique de l'agent. Comme la mise à jour des canaux peut être très coûteuse (en temps de calcul), le choix de la fréquence de mise à jour (qui peut être différente du pas de temps de la simulation) sera fonction de la finesse du modèle que l'on est en train de construire.

2.5.6 Le modèle

Un modèle dans l'environnement CORMAS est composé

- des entités
 - leurs attributs
 - leurs dynamiques
 - leurs perceptions étendue

- leurs communications (traitement des messages)
- leurs actions
- les procédures d’initialisation
- les procédures de contrôle local
- de l’espace
 - les attributs des cellules
 - leur dynamique propre
- des canaux
 - leurs sémantiques
 - leurs dynamiques
- des observations
 - sur la grille : pour chaque entité observable et pour les cellules, les points de vue
 - sur l’observateur graphique : les fonctions d’association, d’écoute et d’attirance
- des statistiques
- des procédures d’initialisation et de contrôle global

2.5.7 La simulation

La simulation d’un modèle dans CORMAS est une instance de ce modèle. Elle se définit par le choix

- pour chaque entité
 - la procédure d’initialisation
 - la procédure de contrôle local
- pour l’espace
 - la taille
 - la forme des cellules, la connexité...
 - l’initialisation des attributs des cellules (aléatoire, à la main, fichier de configuration issu d’un SIG)
- pour l’observation
 - sur la grille spatiale : pour chaque entité observée et pour les cellules, le point de vue
 - sur l’observateur graphique : les fonctions d’association, d’écoute et d’attirance
- les statistiques à observer
- la procédure d’initialisation et de contrôle global



FIG. 2.2 – Le passage de la réalité au modèle

2.6 Démarche de conception de modèles pour CORMAS

2.6.1 Construction du modèle

Nous proposons dans cette partie une démarche structurée de modélisation d'un problème de gestion de ressources pour l'environnement CORMAS. Cette démarche est inspirée de l'approche multi-agents qui préconise la modélisation d'un système comme un ensemble d'entités en interactions.

Les entités sont ici les objets du monde réel. Le choix des entités du modèle sera fonction de la granularité que le concepteur lui attribue. Ainsi, dans un modèle de dynamique de populations par exemple, une entité peut autant être un individu, un ménage, un quartier que la population elle-même. Ce choix dépend du niveau de détail que l'on souhaite prendre en compte dans le modèle. Cette étape est d'autant plus importante qu'elle détermine le choix des comportements à prendre en compte ainsi que le niveau des observations du modèle.

Pour chaque entité, il est important de répondre aux questions suivantes :

- La position spatiale de l'entité est-elle pertinente pour le modèle ?

La réponse à cette question nous permet de savoir si l'entité sera située ou non. Une entité située le sera par rapport à l'intérêt que l'on donne à sa position physique dans le modèle et non par rapport à sa structure réelle.

- L'entité a-t-elle besoin de communiquer avec d'autres entités du système ?

La communication ici sera tantôt l'échange de messages, tantôt l'échange de biens. Le modélisateur doit identifier dans les interactions de la réalité celles qui mettent en oeuvre des échanges d'objets ou d'informations pertinents. L'analyse de ces échanges sur l'observateur graphique pourra renseigner sur la structuration des réseaux sociaux.

L'identification des différentes entités doit être suivie du choix des dynamiques de celles-ci. Il faudra répertorier les différents aspects du comportement de l'entité qui influencent la dynamique du tout.

Cette étape est suivie :

- de la modélisation de l'espace
- de la modélisation des communications et des échanges
- du choix des points de vue pour l'observation
- de la définition de la dynamique globale du système

La démarche de conception peut se résumer dans le tableau 1 ci-dessous :

Ce que fait le modélisateur	Ce que fait l'environnement CORMAS	Exemples
Identification des entités de la réalité Nom Attributs	Construction des classes Nom de la classe Variables d'instances	Chasseur, biche... poids, âge, sexe...
Pour chaque entité : La localisation est-elle pertinente? Participe-t-elle à des communications ?	Pour chaque classe d'entités : Héritage d'un objet situé Héritage d'un objet communicant	oui pour la biche. oui pour le chasseur
Pour chaque entité située, comment observe-t-on son évolution ?	Construction des méthodes de point de vue sur les entités	l'âge de la biche
Si l'espace est pris en compte dans le modèle, quels sont ses attributs propres? Comment observe-t-on l'espace ?	Construction de la classe représentant la cellule spatiale, ajout des attributs. Méthodes de point de vue sur l'espace.	Portion de forêt Attributs : eau, route...
Comment évoluent les attributs de l'espace au cours du temps	Méthodes de dynamique de la cellule	Écoulement de l'eau
Pour les entités communicantes, quelle est la structure des messages utilisés ?	Construction de la classe des messages avec ses variables d'instance	Message avec pour attributs : émetteur, récepteur, type, information
Quelles sont les configurations initiales possibles du système ?	Méthodes d'initialisation des simulations	Création de 20 chasseurs et 100 biches
Quelle est la dynamique du système à chaque pas de temps ?	Méthodes de dynamique des simulations	Les chasseurs courent après les biches
Que désire-t-on observer : La dynamique spatiale? Les communications ? Les statistiques ?	Différentes fenêtres d'observation de la simulation	L'espace et les statistiques sur le nombre de biches.

Tableau 1 : Démarche de conception de modèles pour CORMAS

2.6.2 Construction des simulations

Les simulations seront déterminées par

- L'identification des paramètres et des procédures (comportements) de test de sensi-



FIG. 2.3 – Le passage du modèle à l'implémentation

bilité

- Le choix des plages de valeur de test des paramètres et des scénarios de tests pour les comportements des différentes entités
- Le choix des résultats à observer pour l'analyse des simulations

2.7 Implémentation

L'utilisateur de la plate-forme Cormas, après avoir bien défini son modèle, peut alors le mettre en oeuvre. Cormas peut être vu comme une boîte noire qui reçoit en entrée le modèle multi-agent et produit en sortie des observations de la dynamique du modèle.

2.7.1 Implémentation du modèle

Après le lancement de l'application, l'utilisateur se trouve face à l'écran principal de CORMAS (figure 2.4). C'est à partir de cet écran qu'il construira son modèle, qu'il définira les diverses simulations et qu'il choisira les observations.

Pour implémenter un modèle il faut choisir dans le menu Modele, puis Nouveau... Cette opération ouvre une fenêtre de dialogue où l'utilisateur est invité à entrer le nom du modèle.

A partir de ce moment l'utilisateur peut

- définir les entités du modèle
- définir l'espace
- définir les communications/échanges

2.7.1.1 Définition des entités

Pour implémenter une entité de la société, il faut choisir Ajouter... dans le menu contextuel de la liste des entités Sociétés.

Après avoir donné le nom de l'entité, une fenêtre s'ouvre (figure 2.5) dans laquelle les différentes caractéristiques de l'entité seront définies. A ce niveau, l'utilisateur doit choisir une entité à partir de laquelle la nouvelle entité est construite. Celle-ci peut ainsi des attributs et des méthodes d'une entité existante. Le système propose quatre entités de base à partir desquelles toutes les entités seront construites. Il s'agit de ObjetStandard, ObjetSitué, ObjetCommunicant, ObjetSituéCommunicant. Par la suite, il faudra compléter les

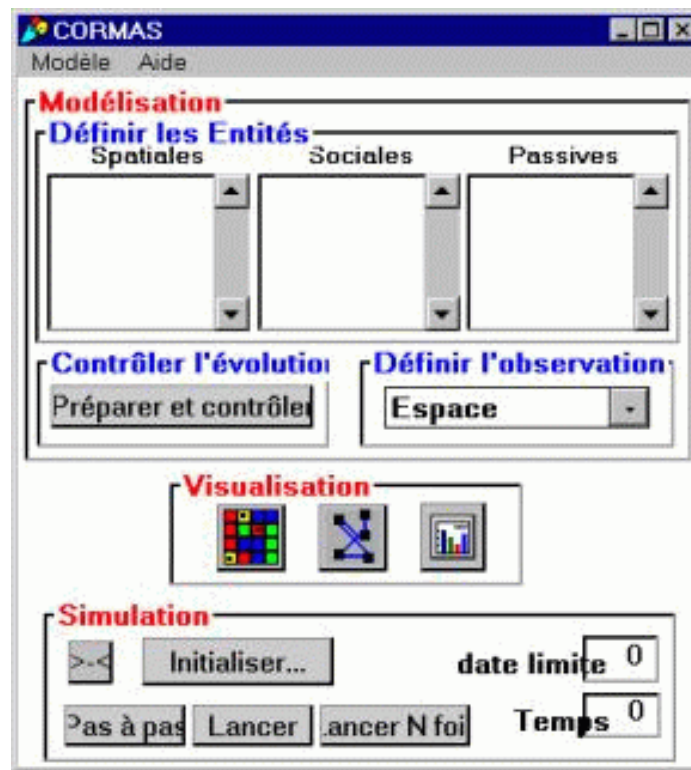


FIG. 2.4 – L'écran principal de CORMAS

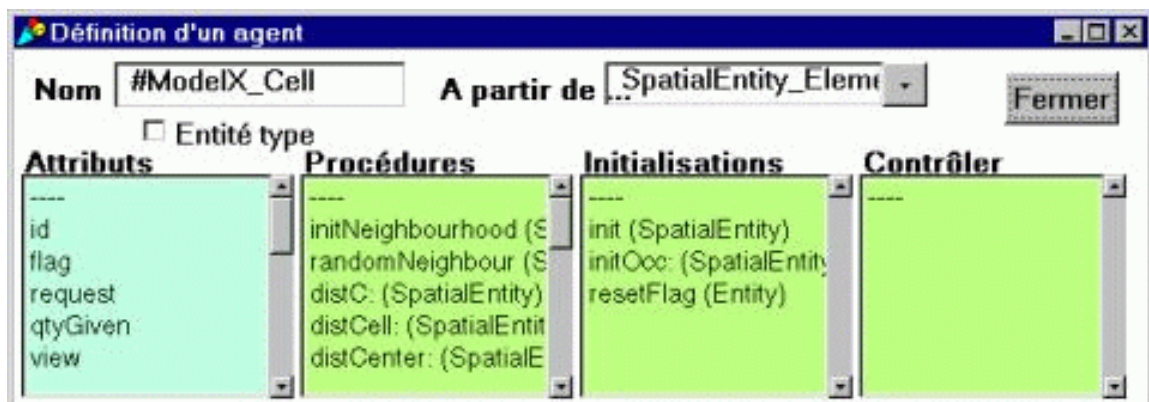


FIG. 2.5 – L'écran de définition des entités

cinq listes quicaractérisent la nouvelle entité. Ces listes contiennent les propriétés(attributs et méthodes) qui sont acquises par héritage. Les propriétés ajoutées à la nouvelle entité seront séparées des précédentes par une barre horizontale (—).

La liste Attributs Cette liste possède un menu contextuel qui permet d'ajouter ou de supprimer les attributs de la nouvelle entité. L'ajout d'un attribut se limite à la définition de son nom. Son type et sa valeur initiale seront précisés dans les méthodes appropriées. Il n'est pas possible de supprimer un attribut appartenant à l'entité à partir de laquelle la nouvelle entité est construite. Si l'entité père possède des attributs dont on n'a pas besoin dans la nouvelle entité, il suffit de ne pas les utiliser. Cette contrainte permet de protéger les informations contenues dans l'entité père.

Les listes de méthodes Les trois listes suivantes (Procedures, Initialisation, Dynamique) sont des listes de méthodes qui définissent le comportement de la nouvelle entité. Elles possèdent un menu contextuel qui permet d'ajouter, de modifier ou de supprimer une méthode. Lors de l'ajout/modification d'une méthode, une fenêtre d'édition ayant pour titre « Méthodes » s'ouvre pour en permettre la saisie. Après avoir saisi/modifié le texte d'une méthode, il faut l'inclure dans le système en choisissant Accepter dans le menu contextuel de la fenêtre d'édition.

La liste Procedures contient les méthodes qui définissent la dynamique de l'entité. Ce sont des procédures d'accès aux attributs et les différentes actions de l'entité.

Exemples de méthodes d'action d'un agent :

```
grandit
seDeplace
seNourrit
litMessage
repondMessage
```

La liste Initialisation contient les méthodes d'initialisation. Ce sont des procédures qui définissent les types ainsi que les valeurs initiales des attributs qui seront appliqués à toute nouvelle instance de l'entité

```
init_X
    self age: 4.
    self revenus: 100.
    self satisfait: false.
    self perception: 2.
```

La liste Dynamique contient les méthodes de contrôle local de l'entité. Le contrôle local est le processus suivi par l'entité à chaque étape (pas de temps élémentaire) de la

simulation. Il correspond à un séquençement particulier des actions de l'entité à chaque pas de temps.

Exemple :

```
controle_local_X
    self grandit.
    self seDeplace.
    self seNourrit.
    self litMessage.
    self repondMessage
```

```
controle_local_Y
    self repondMessage.
    self grandit.
    self seDeplace.
    self seNourrit.
    self litMessage.
```

Lorsqu'une entité est ainsi définie, il est possible de lui associer l'attribut d'entité type. Une entité type est une entité à partir de laquelle on peut construire d'autres entités. C'est le cas par exemple des entités `ObjetStandard`, `ObjetSitué`, `ObjetCommunicant` et `ObjetSituéCommunicant`.

2.7.1.2 Définition de l'espace

Pour définir l'espace, il faut utiliser le bouton Espace de l'écran principal de CORMAS. La procédure de définition de l'espace est en tous points identique à celle des entités. Le nom de la cellule est le nom que l'on donne à la portion d'espace qu'elle représente. Ce pourrait être par exemple un patch, un lieu... ou un mot plus approprié du domaine d'étude.

La cellule est construite à partir de l'entité `Patch` ou de toute autre entité de même type définie dans le système. Lors de la définition de la cellule, il est possible de lui associer la qualité d'entité type pour qu'elle puisse servir de base pour la construction de cellules dans d'autres modèles.

Les attributs de la cellule Les attributs de l'espace représentent les caractéristiques possibles d'une cellule. Ils modélisent l'état d'une portion d'espace (ce qu'elle est). Une cellule pourra par exemple avoir les attributs eau, route, forêt.

Les occupants Les objets situés du modèle sont à chaque instant positionnés sur une cellule. Cette dernière possède une variable qui contient l'ensemble de ses occupants. La variable `lesOccupants` est structurée comme un dictionnaire dont chaque index est un



FIG. 2.6 – L'écran de définition des messages.

symbole représentant une des entités situées du modèle et dont l'élément associé est la liste des instances de cette entité se trouvant sur la cellule.

2.7.1.3 Définition des communications

Le bouton Communication ouvre l'écran de définition des messages (figure 2.6) à partir de laquelle l'utilisateur peut créer (ou modifier) une classe de messages pour les entités communicantes du système. Un message est défini par la liste de ses attributs. Les différentes classes de messages héritent de la classe `Msg` qui a déjà quelques attributs clés comme l'émetteur du message, le récepteur ainsi qu'un symbole qui peut servir à préciser l'objet du message. L'utilisateur pourra ajouter d'autres champs dans ses messages en fonction des informations qui seront transmises.

2.7.1.4 Définition de l'affichage sur l'espace

Cet affichage concerne l'espace et tous les objets observables du modèle. Lors de la simulation, il est nécessaire d'observer l'évolution de ces différents éléments pour mieux comprendre leur dynamique. Le bouton Affichage Espace ouvre une fenêtre (figure 2.7) dans laquelle on définit la façon d'observer la simulation sur l'espace. Cette fenêtre (Observation de l'espace) contient trois listes : Entités Situées, Methodes et Symboles.

La liste Entités Situées contient toutes les entités du système qui sont susceptibles d'être affichées. Il s'agit des agents, des objets situés et de la cellule.

La liste Methodes doit contenir les méthodes qui définissent l'observation de l'entité dans le modèle. Au cours de la simulation, les différentes instances de l'entité peuvent prendre un nombre considérable (voire infini) d'états distincts. Un point de vue est une fonction qui associe à chaque état de l'entité un symbole d'observation. Les méthodes doivent donc retourner des symboles. L'ensemble des symboles doit être suffisamment petit pour faciliter l'observation des simulations.

A partir du menu contextuel de la liste Symboles, il faut choisir Ajouter... pour ajouter

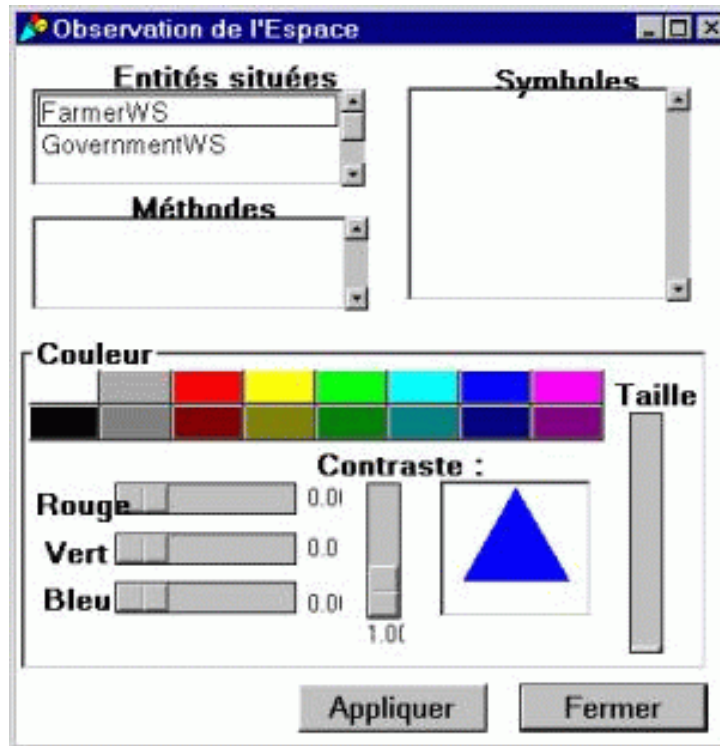


FIG. 2.7 – L'écran de définition des observations.

un symbole de point de vue. Pour chaque entité, il est nécessaire de définir une couleur particulière pour chaque symbole de point de vue qui est retourné par les méthodes de point de vue qui ont été précédemment spécifiées. Le choix d'une couleur se fait par composition à partir des trois couleurs rouge, vert et bleu. On peut aussi faire varier la brillance pour changer de couleur. Les symboles pour lesquels aucune couleur n'a été définie se verront attribuer une couleur par défaut.

La notion de point de vue est capitale pour l'observateur du modèle parce que son choix est lié aux comportements émergents que l'observateur présuppose au système. Plus formellement, un point de vue est défini par :

$$pdv : Etats = \prod Attributes \longrightarrow Symboles \quad (2.3)$$

où les ensembles *Attributes* sont les ensembles des valeurs des différents attributs de l'entité ; *Symboles* est un ensemble fini de valeurs symboliques.

Exemple :

```
point_de_vue_X
  age < 18 ifTrue: [^#jeune].
  (18 <= age and: [age <= 35]) ifTrue: [^#adulte].
  age > 35 ifTrue: [^#vieux]
```

```

point_de_vue_Y
  (age < 15 and: [revenus > 100]
   ifTrue: [#jeuneRiche]
   ifFalse:
     [(age <= 35 and: [revenus > 200])
      ifTrue: [#adulteRiche]
      ifFalse: [#pauvre]])

```

Pour chaque entité située, il faudra définir une image qui est ici un polygone. Dans l'écran de définition de l'affichage, le menu contextuel sur l'image propose la possibilité de modifier le nombre de cotés du polygone (par défaut fixé à 3). Pour modifier l'image d'une entité, il faut la sélectionner dans la liste des entités et choisir nbre côtés dans le menu de l'affichage. Il faut ensuite choisir la taille du polygone avec le curseur qui se trouve à droite de l'image.

Les points de vue sur la cellule spatiale Les points de vue sur les cellules pourront être fonction soit de leurs attributs, soit de leurs occupants. Dans le premier cas, l'espace serait vu comme une carte des objets géographiques pris en compte dans le modèle, et dans le second, il traduirait son état d'occupation. On peut par exemple envisager d'observer la répartition de la densité de la population. C'est lors de la définition de ces points de vue que l'utilisateur précise l'ordre de superposition des couches d'information sur la cellule. Si nous considérons le cas d'une cellule spatiale qui est tantôt une cellule d'eau et tantôt une cellule de route, on peut d'une part envisager que lors de l'affichage on ait :

- soit le point de vue suivant la présence de l'eau, ce qui correspondrait à la carte des eaux,
- soit le point de vue suivant la présence de la route, ce qui correspondrait à la carte des routes,
- soit la superposition des deux cartes. C'est dans ce dernier cas qu'il faudra choisir un ordre sur les points de vue. Cela permet pour les cellules ayant à la fois la route et l'eau de préciser l'information à afficher.

2.7.1.5 Définition de l'affichage des communications et des échanges : l'observateur graphique

La dynamique des communications et des échanges pendant la simulation peut renseigner sur la structuration des réseaux sociaux. Les dynamiques sociales ont un impact très important sur celles des ressources. CORMAS propose une interface de visualisation des communications. L'utilisateur définit les types de messages ainsi que la façon dont ils seront analysés pour l'affichage à partir de l'interface de définition des communications (figure 2.8).

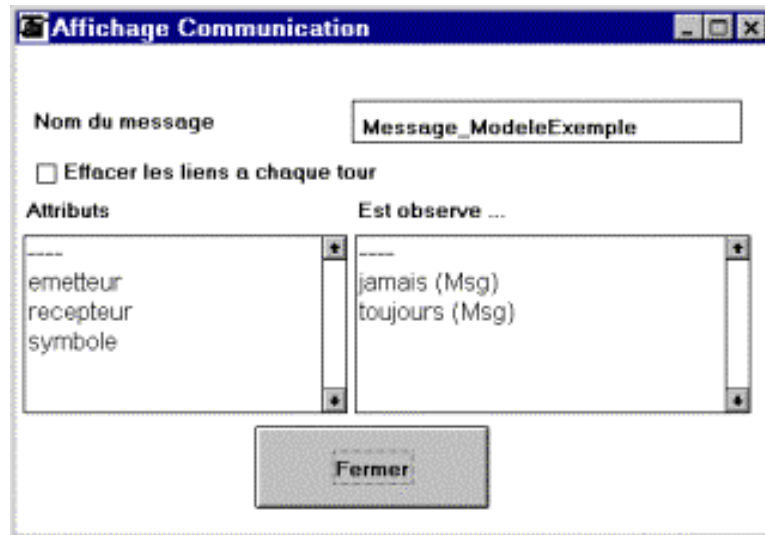


FIG. 2.8 – L'écran de définition de l'affichage des communications.

2.7.1.6 Initialisation et contrôle global

Avant de lancer la simulation, celle-ci doit être initialisée (figure 2.9). C'est pendant la phase d'initialisation que les instances des entités qui entrent dans la simulation sont créées. Celles qui sont situées doivent être réparties dans l'espace. Il est proposé des procédures de répartition spatiale d'objets qui peuvent être simplement réutilisées.

Il est possible que la répartition des entités se trouve sur une carte. Celle-ci peut être chargée en utilisant la procédure de chargement des cartes. Lorsqu'une carte est lue, ses données sont systématiquement affectées à un attribut de la cellule. L'utilisateur doit donc prévoir une procédure qui fait la correspondance entre cet attribut et l'information qui est réellement traduite sur la carte. Par exemple, si l'on charge une carte des densités de populations, il faudra mettre la densité d'une cellule donnée dans un attribut de la cellule (densité) et écrire une procédure qui génère le nombre d'individus se trouvant réellement sur la cellule en fonction de la taille de celle-ci.

Le bouton Init et Dyn (Initialisation et Dynamique) permet de créer les différentes procédures de contrôle global de la simulation. Ces procédures décrivent l'évolution des entités du système pendant l'unité de temps choisie pour la simulation. C'est dans cette procédure qu'est défini l'ordre dans lequel les différentes entités évoluent, étant donné que le choix de cet ordre peut fortement influencer les résultats des simulations.

2.7.2 Initialisation de la simulation

Le choix de la procédure d'initialisation d'une entité revient à la sélection dans la liste Initialisation de cette entité. Il en est de même du choix d'une procédure de contrôle local et d'une procédure de point de vue. Pour initialiser une simulation, ces différents

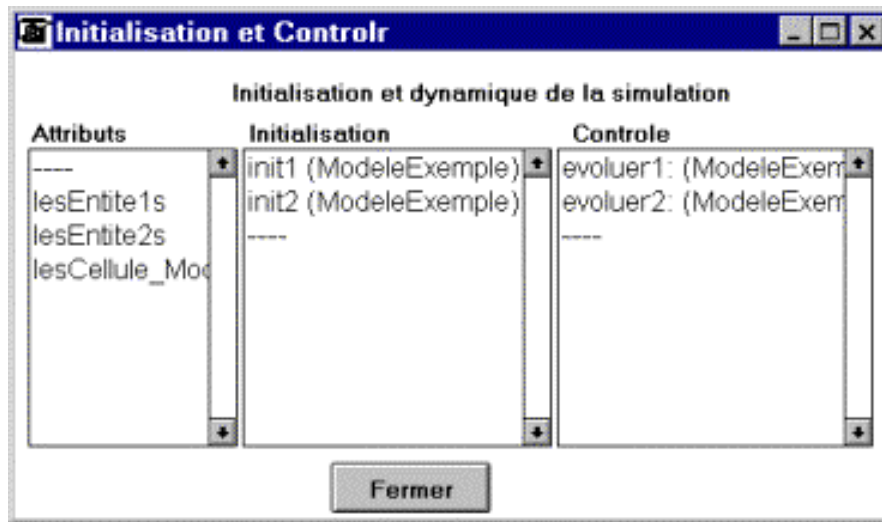


FIG. 2.9 – L'écran de définition des procédures d'initialisation et de contrôle global.

choix doivent être faits pour toutes les entités du modèle et pour la cellule. Ensuite, il faudra choisir la procédure d'initialisation de la simulation elle-même en sélectionnant une procédure dans la liste des procédures prévues à cet effet (bouton Initialisation de l'écran principal). Cette procédure doit créer et distribuer les instances d'entités qui existent au début de la simulation. De même, la procédure de contrôle global qui est retenue pour la simulation courante doit être sélectionnée dans la liste des procédures de contrôle de la simulation.

Avant de lancer la simulation, il faudra initialiser l'espace (pour les simulations qui offrent des dynamiques spatiales). L'espace peut être abstrait : il est alors défini par un découpage régulier en cellules. Le modélisateur doit préciser dans ce cas

- la taille de l'espace (nombre de lignes et de colonnes)
- la forme des cellules (carrée ou hexagonale)
- la connexité des cellules (4, 6 ou 8 suivant la forme)
- le type de clôture de l'espace (fermé sur les bords ou toroïdal)

2.7.3 Déroulement de la simulation

Pour lancer la simulation, l'utilisateur doit choisir soit une exécution sur une période donnée, soit une exécution pas à pas. Dans le cas d'une exécution sur une période, il est nécessaire d'entrer la durée de la simulation, exprimée en nombre de pas de temps, dans le champ Durée de l'écran principal de CORMAS.

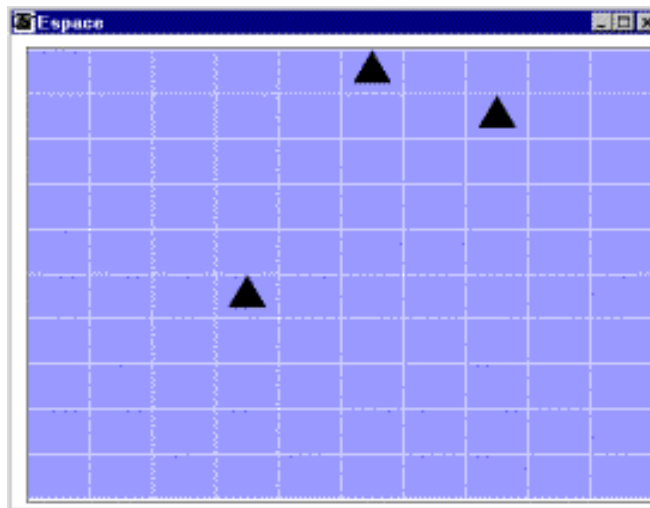


FIG. 2.10 – L'espace de simulation. On peut voir le découpage en cellules ainsi que des entités sur ces cellules

2.7.4 Résultats de la simulation

Pendant le déroulement de la simulation, il est possible d'observer trois types de résultats :

2.7.4.1 La dynamique des entités sur l'espace

Voir figure 2.10

2.7.4.2 Les communications et échanges

Voir figure 2.11

2.7.4.3 Les statistiques

Voir figure 2.12

2.8 Applications

2.8.1 Le feu de brousse

Ce modèle illustre la propagation d'un feu de brousse. C'est l'une des utilisations les plus simples de la plate-forme CORMAS. Il n'y a pas ici d'agent à proprement parler. Toute l'information est localisée sur les cellules de l'espace. Chaque cellule est un automate qui peut posséder un des états suivant : `#fire`, `#tree`, `#ash` ou `#empty`. Les cellules évoluent de

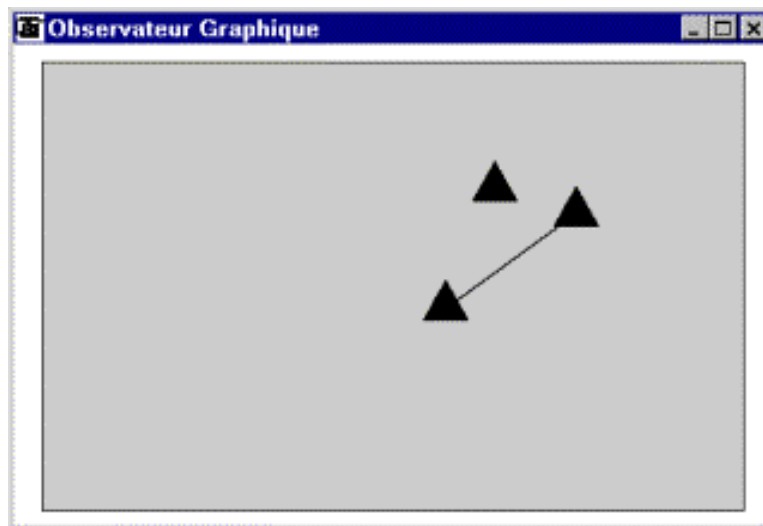


FIG. 2.11 – L'observateur des communications et échanges. Un trait entre deux entités représente un lien de communication entre elles

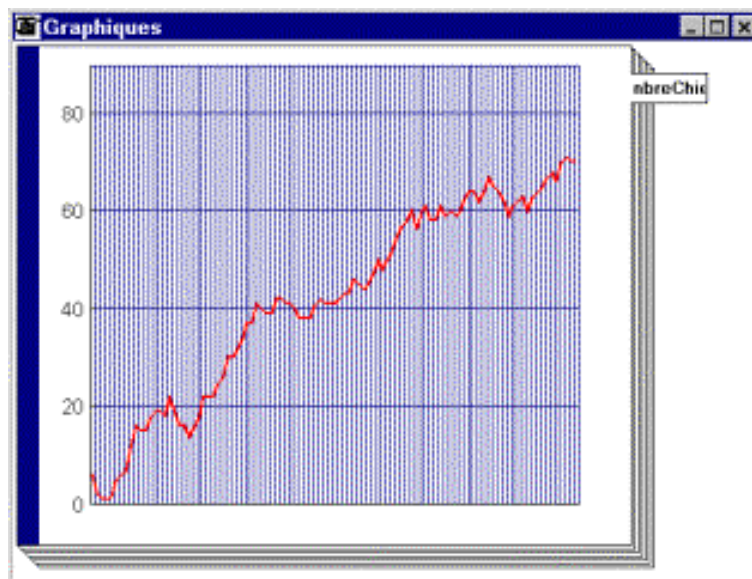


FIG. 2.12 – L'affichage des statistiques

façon synchrone ou asynchrone (le choix est fait à l'initialisation de la simulation), suivant la méthode ci-dessous.

```
newState
    self state = #tree ifTrue: [(self neighbourhood contains:
[:a | a state = #fire])
        ifTrue: [self bufferState: #fire]
        ifFalse: [self bufferState: #tree]].
    self state = #fire ifTrue: [self bufferState: #ash].
    self state = #ash ifTrue: [self bufferState: #empty].
    self state = #empty ifTrue: [self bufferState: #empty]
```

Pour initialiser l'espace à un certain pourcentage de cellules vides et de cellules d'arbre, on utilise le générateur de nombres aléatoires de CORMAS. L'exemple ci-dessous montre le cas où on souhaite avoir 53% d'arbres sur l'espace.

```
init53
    Cormas random < 0.53
        ifTrue: [self state: #tree]
        ifFalse: [self state: #empty]
```

On définit par la suite les caractéristiques de l'espace : cellules carrées, 8- connexes. L'évolution est dite synchrone lorsque toutes les cellules calculent leur nouvel état, puis changent d'état. Elle est dite asynchrone si chaque cellule calcule son nouvel état et change d'état avant que la cellule suivante calcule à son tour son nouvel état. Dans l'évolution synchrone, le calcul de l'état d'une cellule à l'instant t tient compte de l'état des cellules voisines à l'instant $t-1$, tandis que dans le cas de l'évolution asynchrone, le calcul de l'état d'une cellule à l'instant t prend en compte l'état de ses voisines à l'instant t pour celles qui ont déjà calculé leur nouvel état, et de l'état à l'instant $t-1$ pour celles qui ne l'ont pas encore fait.

Du point de vue de la modélisation d'un feu de brousse, le choix d'un mode d'évolution peut être motivé par la volonté de faire apparaître dans le modèle certains aspects du phénomène comme le sens du vent. En simulant une dynamique asynchrone, le sens de parcours des cellules pourrait donc être assimilé au sens de propagation du vent. L'évolution synchrone correspond alors au cas où le modélisateur ne tient pas compte du vent dans le modèle. La figure 2.13 montre quelques étapes des deux modes d'évolution du modèle.

Nous utilisons les méthodes suivantes pour implémenter sous CORMAS les deux modes d'évolution.

Méthode d'évolution asynchrone :

```
stepAsynchronously: t
    self theESE do:
        [:a | a newState; updateState]
```

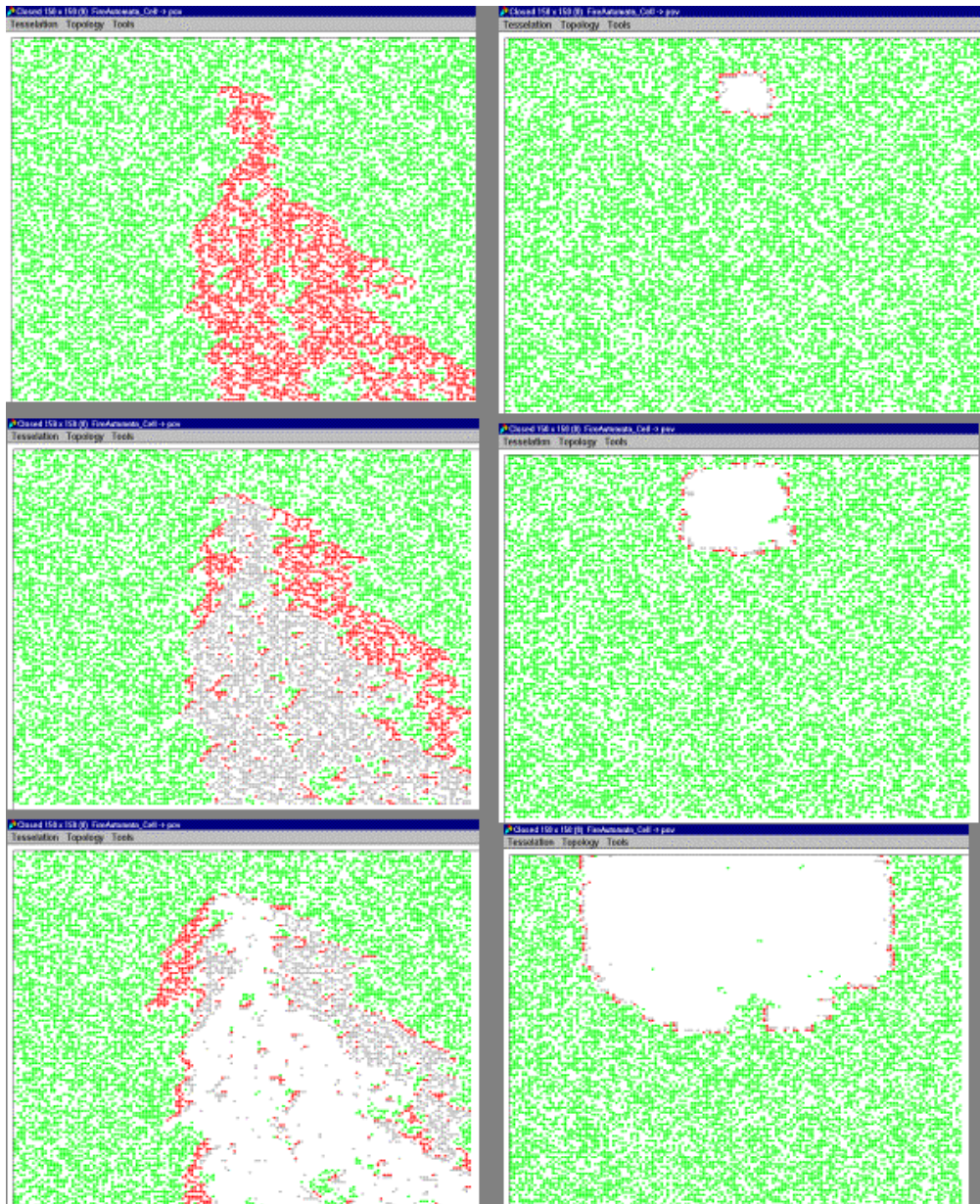


FIG. 2.13 – Evolution asynchrone (à gauche) et synchrone (à droite) du modèle feu de brousse

Méthode d'évolution synchrone :

```
stepSynchronously: t
    self theESE do: [:a | a newState].
    self theESE do: [:a | a updateState]
```

2.8.2 Dynamiques de transformation des paysages.

Zunga et al. [Zunga et al., 1998] ont développé sous CORMAS un modèle avec pour objectif à la fois :

- d'étudier les règles d'installation des agents humains qui conduisent à des situations réalistes en terme d'occupation des sols dans la Moyenne Vallée du Zambèze, mais aussi les déplacements de la grande faune sauvage,
- de disposer d'un outil de simulation des interactions entre des dynamiques foncières (découlant des relations sociales entre les habitants de la zone d'étude) et des dynamiques de populations d'animaux sauvages.

2.8.2.1 L'espace

L'espace dans ce modèle est une grille carrée de 100 ? 100 cellules, chaque cellule représentant une zone de 100 m². La grille dans son ensemble représente donc une superficie de 10 Km². Le voisinage de chaque cellule est l'ensemble des 8 cellules adjacentes, les frontières de la grille étant considérées fermées. Pour initialiser les attributs des cellules à partir des informations contenues dans la base de données géoréférencées (gérée par le Cirad à Harare en utilisant le SIG MapInfo), un fichier de données binaires (présence codée par 1 et absence par 0) est chargé au moment de l'initialisation de l'espace. Les attributs initialisés de cette façon sont : rivière, route, village, champ, jess. Ce dernier argument représente l'habitat préférentiel des grands éléphants mâles qui sont la cause des dégâts les plus graves dans la région. Les cellules qui ne sont d'aucun de ces types seront dites "vides", et susceptibles de se voir transformées par l'activité humaine.

2.8.2.2 Les entités du modèle

Pour représenter les habitants de la moyenne vallée du Zambèze, deux types d'agent ont été développés : l'agent entitéVillageoise, et l'agent fauneSauvage.

L'agent entitéVillageoise L'agent entitéVillageoise représente un ensemble de paysans qui vivent ensemble. Les deux principaux attributs de l'agent entitéVillageoise sont population et cultures (collection de champs). Ces deux attributs sont connectés par un paramètre qui assigne un nombre d'individus par champ, et permet donc par une simple multiplication de connaître le besoin en champs d'une entitéVillageoise.

Une des deux méthodes de l'agent entitéVillageoise qui va directement influencer l'évolution du paysage concerne ainsi le choix de la localisation des nouveaux champs. Dans le prototype qui est présenté ici, cette recherche s'effectue parmi les cellules vides dans le voisinage proche de l'entité villageoise.

La seconde méthode à la base de la transformation du paysage concerne le positionnement de nouveaux agents de type entitéVillageoise lorsque l'attribut population dépasse un certain seuil. Le choix de cette localisation dépend d'un triple test :

- Dans tous les cas, la localisation ne pourra se faire dans les zones où la végétation est de type "jess", cette zone faisant l'objet d'un tabou par les populations locales (sans doute lié au fait que ces zones sont intrinsèquement plus difficile à mettre en valeur, mais aussi l'habitat préférentiel des grands mâles éléphants durant la journée).
- Si dans la voisinage proche de l'entité villageoise dépassant le seuil de population il existe une cellule libre, et que la disponibilité (% de cellules libres) de cette étendue restreinte ne dépasse pas un certain niveau, alors la nouvelle localisation se fera dans ce voisinage.
- Si cette dernière condition n'est pas remplie, alors la nouvelle localisation dépend d'un critère de contraintes, noté cc , qui est défini comme une fonction combinant les distances à la rivière, à la route, ainsi qu'au centre administratif de la région, Mushumbi Pools. Cette localité est située en dehors de la zone d'étude, dans la direction Sud-Est. Pour simplifier, la combinaison est linéaire, l'importance relative de chacun des éléments attractifs étant paramétrée par des coefficients multiplicatifs :

$$cc = A.d_w + B.d_r + C.d_m \quad (2.4)$$

où d_w , d_r et d_m sont respectivement les distances à la rivière, à la route et au centre administratif. Plus la valeur de l'indice de contraintes cc est élevée, moins la cellule a de chance d'être sélectionnée comme lieu d'établissement d'un nouveau village.

La dynamique des agents de type entitéVillageoise est décrite à un niveau de granularité suggérant un pas de temps assez important. Il a été choisi d'effectuer la mise à jour de ces agents tous les 6 mois.

L'agent animalSauvage Les agents de type animalSauvage sont sensés représenter les individus de la grande faune sauvage qui occasionne des dégâts aux cultures. Dans cette zone il s'agit principalement des éléphants et des buffles. Dans ce modèle, l'entité animalSauvage a été défini à partir du comportement des grands éléphants mâles, qui sont les animaux qui occasionnent les dégâts les plus sévères, en raison d'un comportement de recherche active des champs cultivés. De manière très régulière lors de la saison des pluies (qui coïncide avec la culture vivrière du maïs), ces animaux, qui sont situés de jour dans les zones de jess, vont se rassasier de nuit dans les champs puis se désaltérer dans la rivière. Pour modéliser ce comportement, un agent réactif a été créé : partant d'une cellule

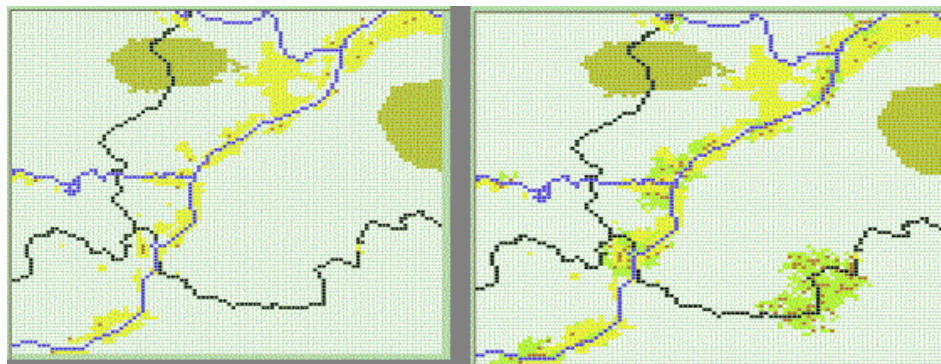


FIG. 2.14 – Simulation de la dynamique de transformation des paysages : les zones de jess, habitat préférentiel des éléphants, sont représentées en marron

aléatoirement sélectionnée dans les zones de jess, cet agent se déplace vers le champ le plus proche, puis de là se rend par le plus court chemin également jusqu'à la rivière.

Pour modéliser ce type de comportement de déplacement, décrit à une échelle spatio-temporelle fine, le pas de temps de la dynamique doit être journalier. Le pas de temps du modèle, qui doit se caler sur le plus petit pas de temps défini pour les dynamiques des différents agents, est donc le jour. Les différentes méthodes des agents ne seront déclenchées que lorsque le pas de temps élémentaire correspond effectivement à la période correspondante.

2.8.2.3 Simulations

La figure 2.14 représente la grille spatiale du modèle. A gauche, la configuration au début de la simulation et à droite l'état de l'espace après 7300 pas de temps (environ 20 ans) de simulation.

2.8.3 Autres applications

De nombreux modèles ont été développés sous CORMAS, mettant en exergue l'approche de gestion des ressources à travers l'étude des interactions entre dynamique des ressources et dynamiques sociales. Ces modèles sont, entre autres :

- Le modèle Raphia élaboré par Vestalys Herimandimby et Eline Randriajaona qui traite de la modélisation de la filière raphia à Madagascar [Herimandimby et al., 1998].
- Le modèle de Martine Antona [Antona, 2000] qui étudie la filière à la gestion du bois énergie au Niger.
- Le modèle Kayanza élaboré par P. Guizol qui traite de l'évolution du paysage d'une région du Burundi sous l'effet de la collecte de bois de feu. Ce modèle prend en compte les dynamiques spatiales sur différentes échelles et examine l'effet des configurations spatiales, de l'imposition exogène de droits d'accès et d'échange de droits d'usage [Bousquet and Le Page, 2000].

- Le modèle Pasteur élaboré par P. d’Aquino et A. Bah sur la transhumance en région sahélienne [Bah, 2000], [Bah et al., 1998]. Ce modèle met en oeuvre des représentations des bergers et leurs décisions. Dans un processus dereprésentation de connaissances, les deux auteurs ont tenté de construire des algorithmes de décision. Les questions posées sont celles de la construction d’une carte mentale par les agents, et d’une capacité de décision pour évaluer des zones viables (agrégats). C’est à cette occasion que A. Bah a défini des agents génétiques dont les traits correspondent aux tâches de perception, de décision, de déplacement et de communication. Cela se rapproche des travaux de J. Holland [Holland, 1995] ou de la plateforme Echo [Hraber et al., 1997].
- Le modèle SINUSE [Feuillette and Bousquet, 2000], [Feuillette and Garin, 2000] élaboré par Sarah Feuillette pose la question des relations entre une nappe phréatique à Kairouan (Tunisie), et les comportements de pompage individuel ou collectif de paysans. Il prend en compte des échanges entre agents et plusieurs dimensions spatiales pour le partage de l’eau.
- Le modèle WErosion élaboré par G. Trebuil et C. Baron s’intéresse à l’érosion sur des bassins versants du Nord Thaïlande, en fonction des stratégies des agriculteurs.
- Le modèle Didy, élaboré par G. Abrami [Abrami, 1999] cherche à modéliser un écosystème forestier où se confrontent différents usages.
- Le modèle Camargue élaboré par R. Mathevet [Mathevet, 1998], [Mathevet, 2000], sur l’activité de chasse en Camargue intègre une structuration de l’espace en différents milieux, des décisions individuelles d’aménagements, des contrats de chasse entre agriculteurs et chasseurs.
- Le modèle proposé par M. Bonin [Bonin and Le Page, 2000] prolonge les réflexions théoriques sur les dynamiques d’entités spatiales et les applique en envisageant différents scénarios sur le massif de Tanargues.
- Le modèle SylvoPast élaboré par M. Etienne, sur l’aménagement de forêts méditerranéennes prend en compte les représentations spatiales de plusieurs acteurs qui négocient pour aménager ensemble un même espace dynamique soumis au risque de feu.

2.9 Conclusion

L’environnement CORMAS que nous avons présenté dans ce chapitre est une illustration de l’apport des systèmes multi-agents dans l’étude de la dynamique des systèmes complexes. Cet environnement permet de modéliser et de simuler les interactions entre dynamiques écologiques et dynamiques sociales, en vue d’aborder sous un angle nouveau les problèmes liés à la gestion des ressources naturelles renouvelables. En effet, ces questions étaient jusqu’alors traitées tantôt par l’approche biologique, tantôt par l’approche économique.

L’approche biologique de la gestion des ressources naturelles renouvelables repose sur

un modèle théorique constitué d'une part de la loi logistique (figure 1 dans le chapitre d'introduction), décrivant l'évolution d'une population dans le temps, et d'autre part de la prise en compte des activités humaines sous la forme d'une pression exogène. La vision est celle d'une stabilité naturelle susceptible d'être perturbée par l'action des hommes. Il en découle une approche de la gestion qui consiste à rétablir cette stabilité, c'est-à-dire en terme d'objectif, à maintenir le stock de ressources à un niveau jugé souhaitable.

Parallèlement à l'approche biologique, la théorie économique standard a une représentation proche du fonctionnement des sociétés humaines. Lorsque les questions environnementales, à l'origine absentes du champ de l'analyse économique, sont prises en compte sous la forme d'« externalités », les mécanismes d'ajustement automatiques ne peuvent jouer leur rôle et des équilibres non efficaces s'établissent. Dès lors, la « gestion économique » consiste à annihiler les effets perturbateurs de l'environnement [Takforyan, 2001].

La plate-forme CORMAS que nous avons développée offre un cadre pour la modélisation et la simulation multi-agents des dynamiques écologiques et sociales, ainsi que des interactions entre celles-ci. Elle prend en compte la spatialisation des agents et intègre un module d'importation d'informations cartographiques, ce qui renforce la vraisemblance des modèles et rend plus pertinentes les observations. Les simulations peuvent être suivies à travers trois interfaces :

- la visualisation des dynamiques spatiales des entités du modèle ;
- la structuration des réseaux sociaux sur un graphe dynamique représentant les communications et les échanges entre les agents ;
- les statistiques montrant l'évolution de certaines variables du modèle.

La communauté des utilisateurs de CORMAS est très active. Le site de la plate-forme `cormas.cirad.fr` répertorie les divers travaux développés sous cet environnement. L'étude que nous présentons dans les chapitres suivants et qui traite de l'étude du phénomène de la chasse dans les forêts de Djemiong à l'Est-Cameroun est une autre illustration de l'utilité de tels outils.

Le principal reproche que l'on peut faire à la plate-forme CORMAS est la lenteur des simulations lorsque le modèle devient très complexe, lenteur due au fait que Smalltalk est un langage interprété. Une des perspectives les plus envisagées pour l'évolution de cette plate-forme est de la reprogrammer en Java.

La tâche de programmation pour l'utilisateur de CORMAS, bien que régulièrement allégée du fait de l'enrichissement de la plate-forme par de nouvelles entités types et de nouvelles méthodes modélisant des comportements courants, reste importante. Un enjeu futur est de faciliter cette tâche en proposant au modélisateur un langage graphique de description de modèles.

Enfin, pour que la plate-forme soit un véritable outil d'aide à la décision, il serait souhaitable que l'utilisateur ait une « explication » des résultats des simulations. On pourrait par exemple envisager de coupler la plate-forme CORMAS à un moteur d'inférence afin de pro-

offrir des facilités de génération de nouvelles connaissances et d'explication du raisonnement développées en Intelligence Artificielle.

Deuxième partie

MODELISATION DE LA CHASSE

Chapitre 3

La chasse à l'Est-Cameroun

3.1 Introduction

Depuis quelques décennies, les états africains et la communauté internationale s'alarment sur la disparition de la faune des forêts africaines. Pour répondre à cette question cruciale, les pouvoirs publics mettent en place des mesures drastiques de protection des espèces « menacées ». Cependant, l'expérience a montré qu'une procédure de mise sous cloche de portions entières d'écosystèmes, dans le but de les préserver contre les agressions des hommes, représente une solution imparfaite car elle engendre des conflits avec les communautés qui subissent de forts coûts sociaux et économiques. Un appel croissant se fait entendre pour une gestion réellement locale de la faune sauvage, mais se pose alors la question de la fonctionnalité de cette gestion locale [Takforyan, 1996]. Cette question a suscité un projet de recherche sur la viabilité de la gestion locale de la faune, et trois villages de l'Est-Cameroun ont servi de terrain d'étude [Green, 1999].

En se situant en milieu rural, c'est-à-dire hors aire protégée et périphérie d'aire protégée, et hors projet de gestion participative, le principe est d'analyser les modes locaux d'exploitation de la faune sauvage, dans une situation où l'absence d'interdictions et de restrictions importantes permet aux populations de continuer à s'adonner à leurs activités selon un mode dit « traditionnel ». Il s'agit d'analyser l'existence éventuelle de modes de gestion communautaires. Les questions que l'on se pose sont d'ordre à la fois méthodologique et politique : comment les systèmes locaux de gestion s'analysent-ils, et dans quelle mesure ces systèmes peuvent-ils servir à bâtir des politiques de gestion communautaire de la faune sauvage ? La description des systèmes de gestion est longuement faite par Takforyan [Green, 1997].

Des enquêtes ont permis de recueillir sur le terrain des données portant sur l'organisation de la chasse en milieu rural, l'évolution des pratiques dans le temps et la place de l'activité dans les économies villageoises. Nous reprenons dans ce chapitre les observations faites par Takforyan lors de ces enquêtes, éléments qui seront utiles dans l'élaboration du modèle

multi-agent que nous présentons dans le chapitre suivant.

3.2 La chasse dans l'espace et dans le temps

3.2.1 La zone d'étude

Le terrain principal de l'étude est constitué de trois villages situés dans le département de la Kadei, dans la province de l'Est du Cameroun. La zone forestière de l'Est-Cameroun est la plus enclavée et la moins peuplée du pays. La densité de la population y est de moins de 5 habitants au km². Plusieurs exploitants forestiers sont installés dans la région depuis les années 70 et ont procédé à diverses coupes sélectives dans la zone d'étude. La forêt se caractérise par une grande biodiversité et apporte de nombreuses ressources aux populations (nourriture, matériaux de construction, produits médicinaux etc.); toutefois les villageois parlent d'une diminution sensible de la faune depuis une vingtaine d'années.

Les populations enquêtées sont des sociétés d'agriculteurs qui pratiquent également la chasse, la pêche et la cueillette. Cette région accueille quatre principaux groupes linguistiques : les pygmées Baka, les Bantou, les Semi-Bantou et les Soudanais (Baya). Les Baka sont des chasseurs-cueilleurs nomades, les autres étant des agriculteurs-chasseurs sédentaires. L'étude de terrain s'est concentrée sur deux villages voisins, d'ethnie différente : Gouté qui est un village Boli (sous-groupe Baya) et Djémiong qui est Mézimé (Bantou).

La chasse est une activité secondaire, mais la plupart des hommes s'y adonnent ; quelques « grands » chasseurs s'y consacrent de façon plus poussée.

3.2.2 L'activité de chasse

L'activité de chasse se pratique essentiellement par piégeage ; les pièges sont des collets qui attrapent l'animal au cou ou à la patte. On trouve quelques rares fusils, pour la plupart artisanaux. La saison de chasse correspond en grande partie à la saison des pluies, soit environ six mois dans l'année, avec un pic entre septembre et novembre. Les principaux animaux chassés sont les antilopes et les rongeurs. Le céphalophe bleu (*Cephalophus monticola*) fait partie des espèces les plus capturées. Il représente environ 30% des prélèvements totaux, en unité de gibier.

Chaque village s'approprie un territoire, une « forêt villageoise » composée d'espaces agricoles et forestiers ; la chasse est potentiellement pratiquée sur l'ensemble de ce territoire. La forêt villageoise est une aire habituelle d'usage et d'influence, avec des limites reconnues par les autres villages. Le territoire est identifié par des points et lieux-dits qui subdivisent la forêt villageoise en plusieurs aires.

On observe plusieurs règles portant sur l'activité de chasse. Il ne s'agit pas toujours de « règles » en tant que telles, mais d'habitudes et de régularités de comportement, dont

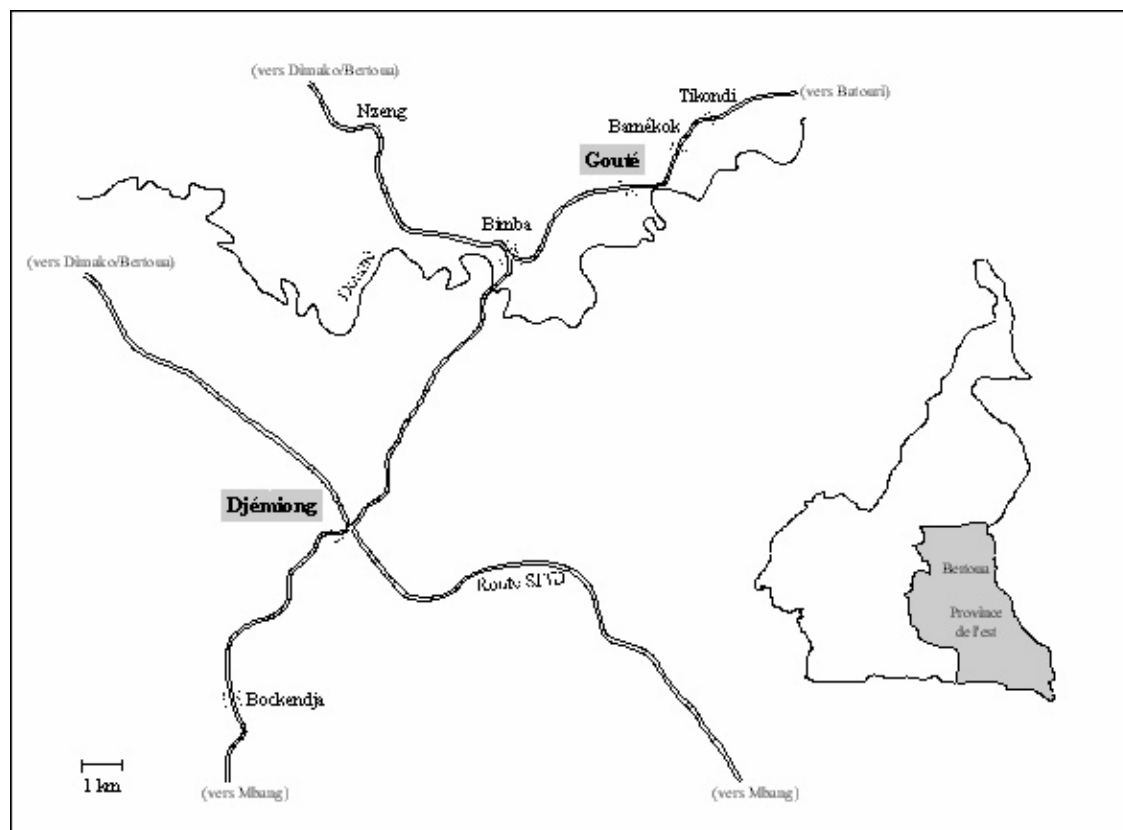


FIG. 3.1 – Carte des villages d'étude

l'examen à posteriori permet de déduire une certaine organisation des pratiques. Les règles ayant trait à la chasse dans le temps et dans l'espace sont les suivantes :

- Des règles d'accès : seuls les chasseurs du village et leurs parents / amis ont accès à cet espace ; tout étranger est en principe exclu.
- Des règles de prélèvement :
 - La saisonnalité : la chasse est pratiquée pendant environ 6 mois par an ; le reste de l'année, les chasseurs cessent en grande partie leur activité ;
 - La répartition des aires de chasses : chaque chasseur a des aires de prédilection (4 en moyenne) sur lesquelles il chasse durant toute sa vie ; ces aires sont choisies en fonction de la parenté, de l'abondance en gibier et des contraintes liées aux autres activités.
 - L'absence de compétition spatiale : les chasseurs tendent à s'éviter dans l'espace et une zone déjà occupée est interdite à tout nouveau venu
 - Le repos cynégétique : les chasseurs effectuent une rotation saisonnière entre leurs aires de prédilection, afin de laisser les zones abandonnées se repeupler. Les aires sur lesquelles les chasseurs ont pratiqué leur activité entre 1985 et 1995 ont été relevé par interview.

La question posée est celle de la fonctionnalité écologique des règles en usage.

3.2.3 Organisation sociale

La chasse est une activité qui se pratique souvent en groupe : dans les deux villages, 50 à 60% des sorties de chasse l'ont été en compagnie d'autres chasseurs. La chasse en groupe, outre qu'elle revêt une fonction sociale évidente, est également pratiquée pour des raisons purement utilitaires : il s'agit de ne pas rester seul compte tenu des dangers potentiels de la forêt.

La constitution des groupes est déterminée par ce que les villageois appellent des « affinités » : des chasseurs, qui présentent différents caractères jugés par eux-mêmes comme étant compatibles, vont chasser ensemble. Les liens familiaux sont les premiers déterminants, et de manière très naturelle, les groupes que l'on observe sont constitués du père et de ses fils. L'apprentissage de la chasse par les enfants et les adolescents se fait d'ailleurs aux côtés du père et des frères aînés. A ce lien familial nucléaire viennent, dans un second temps, s'ajouter les liens lignagers et de voisinage : le père et ses fils en bas âge sont accompagnés d'autres chasseurs avec lesquels ils sont « frères » au sens large, ou « amis ». Il est assez rare d'observer des groupes mixtes du point de vue ethnique lorsque les chasseurs ne cohabitent pas au sein d'un même village.

3.2.4 Le temps

La chasse s'effectue selon un rythme qui peut prendre deux formes : c'est ainsi que Takforyan [Takforyan, 2001] a qualifié la chasse de « courte durée » lorsqu'elle s'effectue au sein d'une même journée ; lorsque le chasseur part du village pour plusieurs jours consécutifs, elle parle de chasse de « longue durée ». La chasse de longue durée correspond à ce que les villageois désignent comme étant la chasse « en campement ». Le terme « campement » regroupe en fait plusieurs notions : celle, physique, d'une petite hutte en nattes de palmier raphia qui sert d'abri ; celle, géographique, de l'endroit où est situé cette hutte ; et enfin celle, générique, de la chasse de longue durée. Environ une fois par semaine, les chasseurs partent camper de trois à quatre jours, durant lesquels ils passent la majeure partie de leur temps à chasser. Il est important de noter que seule la chasse de longue durée est considérée comme une activité de chasse en tant que telle, pourvoyeuse de gibier. La chasse de courte durée, mis à part lorsque le moyen de capture est le fusil, est perçue comme une pratique sans importance, ayant pour but de rapporter occasionnellement une petite quantité de viande pour agrémenter les repas.

La chasse est pratiquée toute l'année, mais la grande période se situe surtout pendant la saison des pluies. On observe deux périodes dans l'année : la grande période de chasse de début Juin à début Novembre qui correspond à la saison des pluies, la période creuse jusqu'en Mai. La saisonnalité de la chasse est un des éléments potentiels de la régulation de la ressource par l'accès. Takforyan écrit dans [Takforyan, 2001] : « L'interruption des prélèvements durant une période donnée de l'année, qu'elle soit basée sur un savoir écologique traditionnel ou simplement sur des raisons pratiques, peut représenter une forme de gestion coutumière des ressources renouvelables. Il faut se demander s'il existe une adéquation entre la saisonnalité de la chasse et les périodes charnières pour la faune locale, ce qui permettrait une reconstitution des stocks durant la période de faible intensité de la chasse » Cette question fournit un des premiers sujets que nous aborderons avec le modèle.

3.2.5 L'espace

Les villages s'approprient un territoire au sein duquel ils exercent leurs différentes activités ; nous avons nommé ce territoire « forêt villageoise » car les villageois parlent de la « forêt de Gouté » et de la « forêt de Djémiong ». Cette forêt représente une aire habituelle d'activité et de rayonnement autour du village ; elle comprend le village, les terres à vocation agricole et la forêt en elle-même.

L'espace en tant que tel n'est appréhendé qu'en fonction des ressources qu'il contient et des usages qu'on leur accorde, points et réseaux. Les points où sont exercées les différentes activités sont identifiés dans l'espace, de manière plus ou moins précise d'un point de vue géographique, par des aires, quartiers et/ou lieux-dits. Le type d'identification est souvent lié à l'usage : les pistes de chasse, qui s'étalent dans l'espace, sont indiquées selon des aires,

alors que la cueillette l'est également par des points particuliers puisqu'elle s'adresse à des arbres et des tiges. Des particularités physiques du paysage permettent de séparer ces aires : rivières et leur source, anciennes routes forestières, collines, etc.

La physionomie de la forêt villageoise dépend des « habitudes » spatiales des villageois en matière d'usage, et sa périphérie est localisée là où les usages s'arrêtent « habituellement ». Ces habitudes se forment sur des critères relativement pragmatiques, tels que la distance à parcourir à partir des habitations, la connaissance que l'on a de la zone, et/ou la plus ou moins grande disponibilité, que celle-ci soit réelle ou anticipée, en ressources. La rencontre avec d'autres groupes humains, qui se sont déjà appropriés l'espace, est également un facteur décisif dans cette région où il n'existe pas de zone « vide » du type *no man's land* ; les habitudes spatiales des uns ont ainsi tendance à s'arrêter lorsque commencent celles des autres.

L'espace de chasse est inclus dans la forêt villageoise. La chasse se pratique essentiellement là où les captures sont probables, telles que les traces animales et lieux de nourriture ou d'abreuvement peuvent le suggérer. Le piégeage s'exerce soit le long des « pistes », terme utilisé localement pour désigner la ligne de pièges, soit autour des champs. Takforyan considère deux oppositions :

- L'espace proche / l'espace éloigné. A partir de 10 km on peut considérer que l'espace est éloigné des habitations et zones de cultures. Les zones de campement commencent à partir de cette distance. Près des trois quart des captures ont été effectuées dans l'espace proche.
- L'espace forestier / l'espace agricole. La chasse dans l'espace agricole apparaît comme faiblement développée relativement à la chasse effectuée en forêt : celle ci regroupe près de 75 % des captures. Dans l'espace agricole la chasse a pour objectif la protection des cultures.

Dans les villages d'étude on observe une rotation annuelle des sites de chasse : les chasseurs changent chaque année de zones. L'objectif affiché est de laisser les zones « se calmer » et d'abandonner les zones provisoirement dépeuplées en gibier pour retourner là où l'espace a eu le temps de se repeupler. Au sein de l'espace de chasse, on constate que, tout au long de sa carrière de chasseur chacun effectue ses prélèvements sur quelques zones seulement, sans qu'il y ait attribution ou répartition officielle de ces zones. Les choix spatiaux sont effectués en raisons de critères individuels (rentabilité, complémentarité avec d'autres activités, accessibilité) mais aussi collectifs. Ces choix individuels sont contraints par des choix collectifs, qui amènent les chasseurs à identifier leurs lieux de capture en fonction de critères dictés par la société.

Il est interdit de tendre des pièges aux abords d'une autre piste. Il y a volonté d'éviter la compétition spatiale. La parenté est le critère de base dans les choix spatiaux du chasseur « car le fils suit toujours les traces de son père ». Le fils rouvre les pistes de chasse de son père et utilise les campements de celui-ci. La répartition des chasseurs dans l'espace est

donc en première instance sociale. Takforyan écrit : « le système de rotation annuelle des lieux de chasse est une règle explicite de renouvellement de la faune et affiché en tant que telle ».

La fonctionnalité réelle de ce système de rotation est le deuxième sujet que nous aborderons avec le modèle.

3.3 Le Céphalophe bleu

3.3.1 Connaissances éthologiques

Grâce aux travaux de Gérard Dubost [Dubost, 1980], [Dubost, 1983a] et [Dubost, 1983b]; les connaissances éthologiques sur la vie du Céphalophe bleu (*Cephalophus monticola*) sont bien établies. Plus connu sous le nom de lièvre, c'est un ruminant vivant exclusivement en Afrique. Son activité est strictement diurne, avec un rythme bimodal. Il se déplace assez rapidement soit par bonds allongés, soit par bonds successifs exécutés à grande vitesse. Il est doté de bonnes facultés visuelles et son ouïe est excellente. Lorsque le groupe familial est face à un danger, il s'éclate en étoile et chaque individu s'isole rapidement, ce qui correspondrait à un comportement anti-prédateur efficace. L'animal présente des comportements de jeu variés : la course, les cabrioles, le saut en longueur ou les grattages désordonnés contre la végétation. C'est un animal qui n'aime pas l'eau mais qui est tout à fait capable de traverser des rivières.

Le céphalophe est un animal territorial qui vit sur un territoire d'environ trois hectares. Sur son territoire, il procède à divers types de marquages sur son chemin : tantôt en déposant des glandes préorbitaires sur des feuilles, des brindilles, des tiges d'herbe, des troncs ou des racines, tantôt en raclant avec ses cornes les troncs des arbustes. Les marquages par les glandes préorbitaires se font aussi entre les membres d'un même groupe familial.

Dans le système d'organisation sociale adopté par le Céphalophe bleu, le couple est l'unité de base autour de laquelle s'organise la cellule familiale. Il y a chez l'espèce une nette tendance au regroupement : les jeunes sevrés continuent de participer dans une certaine mesure aux activités du couple jusqu'à ce qu'ils soient en mesure de fonder un couple indépendant. Cette indépendance survient à l'âge de 18 mois pour la femelle et de 24 mois pour le mâle. La femelle a une gestation de 30 semaines. Le céphalophe a une durée de vie totale de 7 ans en moyenne. La mortalité des juvéniles est de 30% pour les 40 premières semaines de vie.

3.3.2 Etude du comportement social

Pour des raisons simplement matérielles (taille mémoire, durée des simulations...), il ne serait pas envisageable de mener une étude de la dynamique de la faune dans la région

en reproduisant les moindres détails de la vie quotidienne de chaque individu de la population. Des hypothèses seront faites pour extrapoler les observations des comportements quotidiens afin d'obtenir leurs équivalents à une échelle de temps plus grande. On envisagera plus tard de considérer que le pas de temps est la semaine. Mais pour valider cette hypothèse, nous allons mener une première modélisation en vue de montrer comment les comportements supposés du céphalophe à l'échelle hebdomadaire sont effectivement une émergence des comportements quotidiens observés sur le terrain. Considérons par exemple la tendance qu'ont les céphalophes à se regrouper en « familles ». La littérature sur la question [Dubost, 1980] nous apprend que cette attitude pourrait résulter des actions anodines comme le fait que les individus reconnaissent et suivent les traces de leurs parents. Comment ce phénomène résulte-t-il des comportements directement observables dans la réalité ? Nous avons utilisé les automates cellulaires pour modéliser le comportement social du céphalophe.

3.4 Conclusion

Les recherches menées dans les villages de l'Est Cameroun par Ani Takforyan portaient sur l'existence d'un système de gestion coutumière de la faune sauvage. Diverses pratiques ont été relevées qui semblent participer à ce système de gestion. Elle ont trait à la limitation de l'accès, au repos cynégétique, à la répartition spatiale des actes de chasse, à la limitation des quantités et au contrôle des règles.

Les principaux résultats dégagés par Ani Takforyan de ces enquêtes sont les suivants :

- la chasse est une activité spécialisée et adaptée aux conditions locales, et qui a connu de profonds changements suite à des évolutions techniques (introduction du fusil et du câble métallique), économiques (développement du commerce, désenclavement) et écologiques (diminution de la faune) ;
- il existe une forte interactivité et une complémentarité entre la chasse et les autres activités, en terme de pratiques et sur le plan monétaire ;
- les structures de parenté sont un déterminant majeur de l'accès et de la répartition des ressources.

Ces résultats illustraient principalement les modes d'appropriation, tout en complétant les informations concernant les processus de décision. Takforyan en conclut que « la gestion locale implique de trouver des solutions novatrices qui, du point de vue des objectifs, dépassent les seules logiques villageoises et, du point de vue des outils et des instances décisionnelles, combinent la flexibilité des structures locales avec des structures plus formelles et normatives ; de telles innovations nécessitent la mise en place d'un processus de négociation entre tous les acteurs concernés, négociation qui pourrait aider leur émergence et permettrait leur légitimation ».

Parmi les questions soulevées, nous en avons identifié deux qui mettent en jeu des

problèmes de coordinations (dans le temps, dans l'espace, entre les acteurs) des usages et pour lesquelles nous pensons que les simulations peuvent être utiles en couplant des dynamiques naturelles et des dynamiques sociales. Ces questions sont les suivantes :

- quel est le rôle du repos cynégétique, qui vient à la fois d'une saisonnalité de la chasse et des rotations des sites de chasse, dans la préservation de la ressource ?
- quel est le rôle des règles collectives de répartition spatiale dans la préservation de la ressource, qu'il s'agisse du partage familial des lieux de chasse ou de la compétition spatiale ?

Chapitre 4

Modélisation multi-agent de la chasse

4.1 Introduction

Les enquêtes menées dans le village de Djemiong à l'Est-Cameroun avaient pour but d'étudier la viabilité des pratiques locales de chasse. Nous avons présenté dans la chapitre précédent les principales observations faites sur le terrain, ainsi que les questions soulevées à la suite de ces enquêtes. Elles peuvent se résumer ainsi : en quoi les stratégies spatio-temporelles mises en oeuvre par les chasseurs concourent-elles à la viabilité de la faune dans cette région ?

Des modèles mathématiques exprimant les variations dans le temps de populations animales ont été proposées par divers chercheurs ([Barbault, 1992], [Le Page, 1996], [Volterra, 1926]).

Ces modèles présentent certaines insuffisances, notamment [Ferber, 1995] :

- L'étanchéité du niveau d'analyse : ces modèles relient des paramètres qui se situent tous au même niveau d'analyse, et il n'est pas possible de faire correspondre des comportements effectués au niveau « micro » à des variables globales mesurées au niveau « macro ».
- La complexité et le réalisme des paramètres : pour être utilisables et correspondre à la réalité, ces modèles comportent souvent un grand nombre de paramètres difficiles à estimer et manquent surtout de réalisme.
- Difficulté à modéliser l'action : la plupart des phénomènes collectifs sont le résultat d'un ensemble de prises de décisions individuelles qui tiennent compte des comportements des autres acteurs du système. En ne considérant les actions que par leurs conséquences mesurables au niveau global ou par leur probabilité d'apparition, il s'avère difficile d'expliquer les phénomènes émergents dus à l'interaction de ces comportements individuels.
- La carence qualitative : par leur nature même, les modèles numériques ne peuvent considérer que des paramètres quantitatifs et apparaissent désarmés devant la multitude d'informations qualitatives recueillies par les chercheurs de terrain.

Céphalophe
partenaire age dureeGestation sexe ageMiseBas
chercherPartenaire deplacement estAccessible: estAccueillant: evoluer: isAdulte mortaliteChasse mortaliteNaturelle rencontrerPartenaire

FIG. 4.1 – Un agent Céphalophe

Nous avons élaboré un modèle multi-agent de dynamique de la population de céphalophes bleus. Cette espèce est de loin la plus chassée, au regard des observations effectuées. La chasse au céphalophe s'effectue par piégeage, ce qui suppose une stratégie spatiale des chasseurs que nous voulons tester. Le modèle a été élaboré à partir des données collectées dans le village de Djémiong, village sur lequel nous disposons d'une cartographie des différents lieux dits (il en existe vingt-neuf, cartographiés par Takforyan en identifiant les limites avec les paysans), des routes et des rivières. Nous avons procédé à plusieurs séries de simulations visant à tester la viabilité du système de gestion locale de la faune.

4.2 Modélisation

4.2.1 Le céphalophe

Comme nous l'avons vu au chapitre précédent, les connaissances sur le céphalophe bleu sont bien établies. Un agent céphalophe a été créé (figure 4.1). Ses attributs sont l'âge, le sexe, la durée de gestation et le partenaire. Le comportement du céphalophe est implémenté par une méthode qui utilise les paramètres décrits ci-dessus pour simuler une fonction de mortalité, de croissance et de reproduction. Le pas de temps de la simulation est la semaine, et le rayon de déplacement pendant cette période d'un céphalophe non installé est de 600 m.

Nous reproduisons quelques méthodes qui sont implémentées par l'agent Céphalophe :

- chercherPartenaire est appelée lorsque l'animal doit rechercher un partenaire.

```

chercherPartenaire
| tmp |
tmp := patch extensiveNeighbourhood select:
    [:unP | (unP theOccupants at: self class name) notNil].
tmp do: [:unP | (unP theOccupants at: self class name)
    do: [:unC | (unC sexe = self sexe) not & unC partenaire
        isNil & unC dead not & unC isAdulte ifTrue: [^unC]]].
^nil

```

- isAdulte teste si l'animal a atteint l'âge adulte. Cet âge est fonction du sexe de l'animal.

```

isAdulte
    ^(self sexe = #male and: [self age > 95])
    or: [self sexe = #femelle and: [self age > 72]]

```

- déplacement est appelée lorsque l'animal doit se déplacer

```

deplacement
| autreAilleurs nouveauLieu |
autreAilleurs := self rencontrerPartenaire.
autreAilleurs isNil
    ifTrue:
        [nouveauLieu := (patch extensiveNeighbourhood select:
            [:unP |
                (unP theOccupants at: self class name) isEmpty
                & (self estAccessible: unP)]) asOrderedCollection.
nouveauLieu isEmpty ifFalse:
    [self leave.
    self moveTo: (nouveauLieu at:
        (Cormas random * (nouveauLieu size - 1)) rounded + 1)]]
    ifFalse:
        [partenaire := autreAilleurs at: 1.
        self leave.
        self moveTo: (autreAilleurs at: 2).
        partenaire partenaire: self.
        partenaire leave.
        partenaire moveTo: (autreAilleurs at: 2).
        (self estAccueillant: (autreAilleurs at: 2)) not ifTrue:
            [self halt]]

```

- `estAccessible` permet à l'animal de savoir si la cellule sur laquelle elle souhaite se déplacer ne contient pas de piège.

```
estAccessible: unPatch
    unPatch isNil ifTrue: [^false].
    ^( unPatch huntingLocality isNil) not
```

- `estAccueillant` permet à l'animal de savoir si la cellule sur laquelle elle souhaite se déplacer n'est ni sur la route, ni sur un cours d'eau.

```
estAccueillant: unPatch
    unPatch huntingLocality isNil ifTrue: [^false].
    ^unPatch road not & unPatch river not
```

- `mortaliteChasse` évalue si l'animal est pris dans le piège posé sur la cellule sur laquelle il se trouve.

```
mortaliteChasse
    (self patch huntingPressure > 0
    and: [Cormas random < (self patch huntingPressure / 1000)])
    ifTrue:
        [partenaire notNil ifTrue:
            [partenaire partenaire: nil.
             partenaire := nil].
         dead := true]
```

- `mortaliteNaturelle` évalue si l'animal est considéré comme mort suivant un pourcentage de mortalité naturelle.

```
mortaliteNaturelle
  (self age < 40 and: [Cormas random < (1 - (0.7 raisedTo: 1 / 40))])
  ifTrue:
    [self dead: true].
  (self age >= 40 and: [self age < 280 and: [Cormas random < (1 - (0.95
raisedTo: 1 / 240))]])
  ifTrue:
    [self partenaire notNil
  ifTrue:
    [self partenaire partenaire: nil.
    self partenaire: nil].
    self dead: true].
  (self age >= 281 and: [Cormas random < (1 - (0.01 raisedTo: 1 /
100))])
  ifTrue:
    [self partenaire notNil
  ifTrue:
    [self partenaire partenaire: nil.
    self partenaire: nil].
    self dead: true]
```

- `rencontrerPartenaire` est la méthode exécutée par l'animal lors d'un déplacement pour rechercher de préférence les cellules sur lesquelles il est susceptible de rencontrer un partenaire.


```

rencontrerPartenaire
  | lesTerritoiresPossibles partenairePotentiel lieuxPartenaire
nouvelHabitat |
  (partenairePotentiel := self chercherPartenaire) isNil ifTrue: [^nil].
  lieuxPartenaire := partenairePotentiel patch extensiveNeighbourhood
select: [:unP | self estAccueillant: unP].
  lesTerritoiresPossibles := OrderedCollection new.
  ((patch theOccupants at: self class name) size = 1 and: [self
estAccueillant: patch])
    ifTrue: [lesTerritoiresPossibles add: patch].
  ((partenairePotentiel patch theOccupants at: self class name) size = 1
and: [self estAccueillant: partenairePotentiel patch])
    ifTrue: [lesTerritoiresPossibles add: partenairePotentiel
patch].
  lieuxPartenaire do: [:unP | ((unP theOccupants at: self class name)
isNil and: [patch extensiveNeighbourhood includes: unP])
    ifTrue: [lesTerritoiresPossibles add: unP]].
  lesTerritoiresPossibles isEmpty
    ifTrue: [^nil]
    ifFalse:
      [nouvelHabitat := Cormas selectRandomlyFrom:
lesTerritoiresPossibles.
      (self estAccueillant: nouvelHabitat) not ifTrue: [self halt].
      ^Array with: partenairePotentiel with: nouvelHabitat]

```

- **evoluer** : est la méthode que l'animal exécute à chaque pas de temps de la simulation.

```

evoluer: unTemps
  | petit |
  petit := 0.
  age := age + 1.
  self mortaliteNaturelle.
  self dead
    ifTrue: [^0]
    ifFalse:
      [self mortaliteChasse.
       self dead ifTrue: [^1]].
  self isAdulte & partenaire isNil ifTrue: [self deplacement].
  self sexe = #femelle
    ifTrue:
      [dureeGestation > 0
       ifTrue: [dureeGestation := dureeGestation + 1].
       dureeGestation = 30
       ifTrue:
         [dureeGestation := 0.
          petit := self class new init0.
          petit patch: patch]
       ifFalse: [(partenaire notNil and: [dureeGestation = 0
and: [(patch theOccupants at: self class name)
       contains: [:a | a age < 21])not]])
         ifTrue: [dureeGestation := 1]].
    ^petit

```

4.2.2 Le sentier de chasse

D'après les observations de terrain et diverses données bibliographiques on peut estimer l'aire couverte par des sentiers de chasse d'un chasseur de 20 hectares à 100 hectares. Dans le modèle nous avons considéré qu'un sentier composé de 30 pièges couvre une surface de 24 hectares, ce qui représente huit cellules. Estimer le paramètre de capturabilité est extrêmement difficile. La seule donnée dont nous disposons est une étude faite dans la région du Dja [Ngandjui, 1994] qui montre qu'une ligne de piège vide sa surface de prise en deux mois. Mais, étant donnée que les animaux se déplacent comment estimer cette surface? Dans les simulations que nous présentons ci-dessous nous avons choisi une probabilité de 0.01 : tout céphalophe dans une zone chassée a 1% de chances de disparaître chaque semaine. Par la suite nous allons procéder par une démarche comparative. En maintenant fixés ces deux paramètres (surface et capturabilité), nous allons comparer différents scénarios d'actions

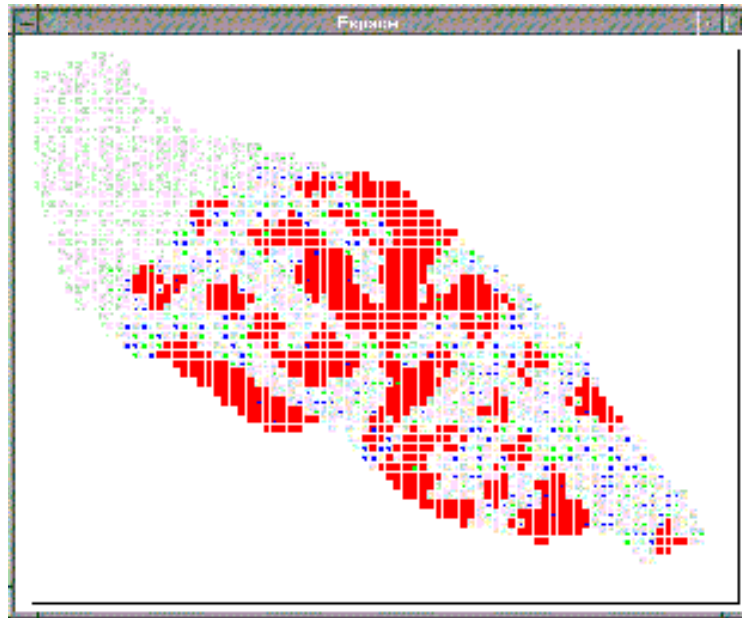


FIG. 4.2 – Carte de Djemiong avec les zones chassées en rouge

Chasseur
terrainsChasse dernierLieuDitChasse
choixLieuDit

FIG. 4.3 – Un agent Chasseur

de chasse.

D'un point de vue modélisation nous avons utilisé le concept d'agrégat pour modéliser un sentier de chasse. Un agrégat est composé de huit cellules pour les sentiers éloignés des villages et quatre cellules pour les sentiers plus proches du village où l'activité agricole occupe des portions de l'espace. Nous avons inclus une certaine complémentarité avec l'agriculture. Les lieux dits du milieu de la carte font l'objet d'une concurrence entre la chasse et l'agriculture. En analysant les données nous avons observé que la densité de pièges aux céphalophes est environ deux fois moindre dans ces endroits. De plus cette chasse est effectuée aux abords des champs. Pour cette raison, dans ces lieux-dits, nous avons fixé la taille du sentier à quatre cellules. La figure 4.2 donne un exemple de configuration de zones chassées.

4.2.3 Le chasseur et le groupe de chasseurs

Pour certains scénarios de simulation un agent chasseur a été créé. Son comportement est très simple. Au départ de la simulation chaque chasseur reçoit un lot de quatre sentiers de chasse.

Chaque saison au moment de commencer la chasse, il choisit parmi ces quatre sentiers un des sentiers qui n'a pas été chassé l'année précédente. Dans une variante nous introduisons un autre agent : le groupe de chasseurs. C'est le groupe de chasseurs, composé de trois chasseurs, qui contrôle quatre sentiers et qui va allouer les sentiers aux chasseurs.

- `choixLieuDit` est la méthode que le chasseur exécute lors du choix du lieu-dit où il posera ses pièges.

```
choixLieuDit
  | lieuxDitsPossibles |
  lieuxDitsPossibles := OrderedCollection new.
  self terrainsChasse keys do: [:ld | ld = self dernierLieuDitChasse
iffFalse: [lieuxDitsPossibles addLast: ld]].
  self dernierLieuDitChasse: (Cormas selectRandomlyFrom:
lieuxDitsPossibles).
  ^self dernierLieuDitChasse
```

- `choixTerrains` est la méthode que le groupe de chasseurs exécute lors du choix de la zone à chasser.

```

choixTerrains
"etat chasse = 0 -> chasse, 1 un an de repos, 2 deux ans de repos"
  | chasseurs sentiers sentiersTries r terrainsAdonner o tad |
chasseurs := self components.
chasseurs do: [:a | a terrainsChasse: Dictionary new].
sentiers := self terrainsChasse values.
sentiers
  do: [:s | s etatChasse > 0 ifTrue: [s etatChasse: s etatChasse + 1]].
sentiersTries := sentiers asSortedCollection: [:a1 :a2 | a1 etatChasse
< a2 etatChasse].
sentiersTries last etatChasse = 0
  ifTrue: [sentiersTries last etatChasse: 1].
sentiersTries last etatChasse = 4
  ifTrue:
    [sentiersTries last etatChasse: 0.
    r := Cormas randomFrom: 1 to: 3.
    (sentiersTries at: r) etatChasse: 1].
terrainsAdonner := self terrainsChasse select: [:a | a etatChasse = 0].
(self terrainsChasse select: [:a | a etatChasse > 0])
  do: [:t | t components do: [:tc | tc numero: 0]].
o := OrderedCollection new.
terrainsAdonner do: [:t | o addAll: t components].
o do: [:t | t numero: 1].
chasseurs
  do: [:c |
    tad := terrainsAdonner associations first.
    c terrainsChasse add: tad.
    terrainsAdonner removeKey: tad key].
~o

```

4.3 Simulations

4.3.1 Calibrage

Pour tester la qualité de notre modèle de dynamique des populations nous avons effectué plusieurs simulations en faisant varier les conditions initiales. On remarque en résultat (figure 4.4) que toutes les simulations convergent vers le même attracteur qui se situe autour d'une densité de population égale à 80 habitants / km². Or il s'avère que cette densité est bien celle relevée par Dubost dans les zones non chassées.

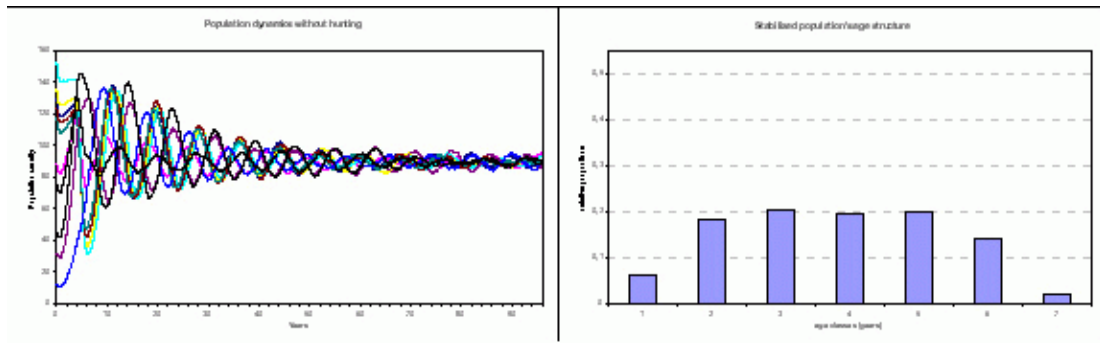


FIG. 4.4 – Convergence de la dynamique sans chasse. 10 simulations ont été effectuées en changeant le nombre initial d’animaux. La figure gauche présente la convergence vers un attracteur en densité d’animaux, la figure de droite la structure d’âge résultante après 5000 pas de temps.

Nous observons aussi d’autres paramètres sur la structure de la population, comme la structure en âge de la population. Après avoir simulé la dynamique de la population pendant 1000 ans, nous avons enregistré la configuration qui servira de conditions initiales pour tous les scénarios testés ensuite.

4.3.2 Scénarios

Nous avons défini divers scénarios de chasse :

Scénario 1. En utilisant les données de chasse de l’année 1995 (nombre de sentiers par lieux-dits) nous avons répété cette configuration initiale. Les sentiers sont chassés en continu et restent toujours au même endroit.

Scénario 2. Ce scénario est le même que le premier mais on fait intervenir la durée de chasse. Pendant 26 semaines, les pièges sont retirés. Mais les sentiers sont toujours repositionnés aux mêmes endroits.

Scénario 3. Les données issues du terrain se rapportent à une échelle spatiale qui est celle du lieu-dit. Ce lieu-dit couvre des surfaces importantes, bien plus larges que celle d’un sentier de chasse. Dans ce troisième scénario le nombre de sentiers par lieu-dit reste fixe mais à l’intérieur de ce lieu-dit leur place est choisie au hasard chaque année.

Scénario 4. Afin de prendre en compte le système de rotation décrit sur le terrain, les agents représentant des chasseurs sont introduits (au nombre de 90). Ils disposent chacun de quatre sentiers de chasse entre lesquels ils tournent chaque année.

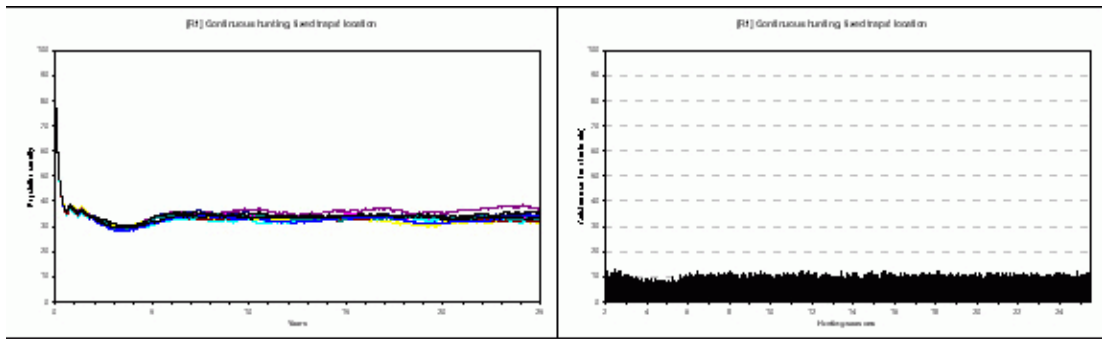


FIG. 4.5 – Scénario 1 : chasse en continu et sentiers fixes

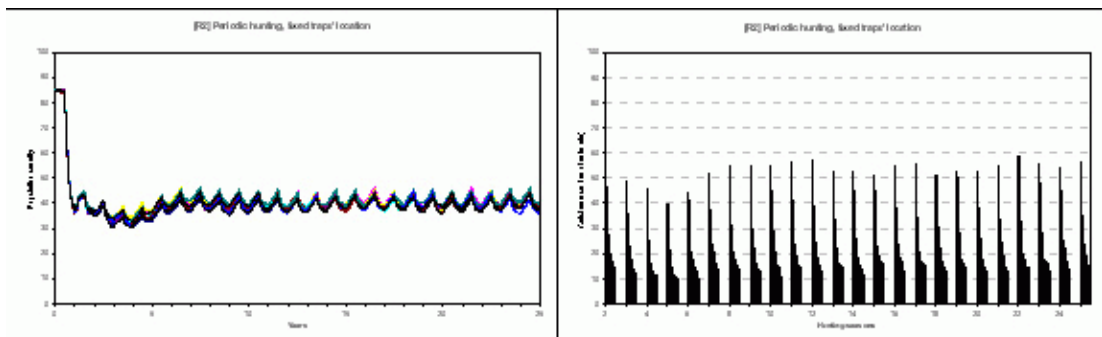


FIG. 4.6 – Scénario 2 : chasse périodique et sentiers fixes

Scénario 5. La gestion des sentiers est effectuée par les groupes de chasseurs (au nombre de 30). Parmi les quatre sentiers d'un groupe, un sentier est choisi puis protégé pendant trois ans avant de revenir dans le groupe des sentiers chassés.

4.3.3 Résultats des simulations

Nous avons choisi d'observer deux types de résultats : la dynamique de la population de céphalopes d'une part afin d'évaluer la persistance des scénarios envisagés et le nombre des captures d'autre part. Les simulations durent 1300 pas de temps, soit 25 ans. Nous observons la densité de population d'animaux et les captures au cours des 52 semaines de la saison de chasse (figures 4.5 à 4.9).

Les figures qui représentent la densité de population montrent dix réalisations de simulations, tandis que la courbe des captures représente la moyenne des valeurs pour les dix simulations. La première saison de chasse donne de très fortes captures. Pour permettre d'observer distinctement les données, nous avons omis de représenter les premières saisons de chasse sur les figures.

Les indicateurs sont résumés dans les tableaux 2 et 3. Le premier indicateur est la surface chassée, exprimée en nombre de cellules, qui donne une indication de l'effort de

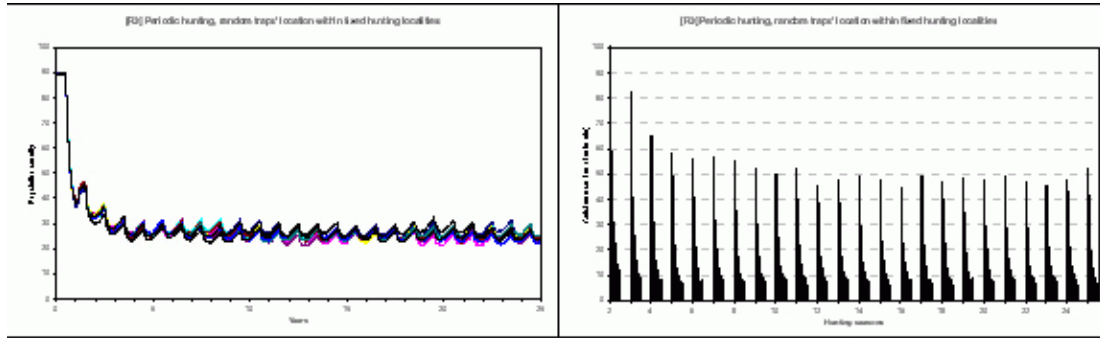


FIG. 4.7 – Scénario 3 : chasse périodique et sentiers redéfinis chaque année

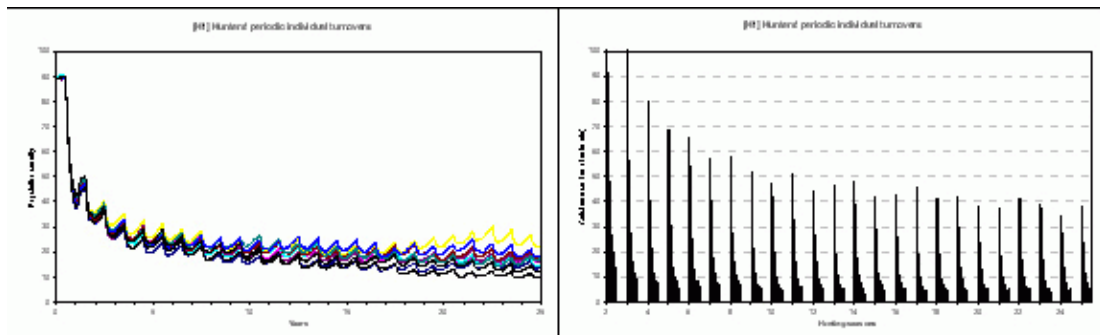


FIG. 4.8 – Scénario 4 : rotation individuelle des chasseurs

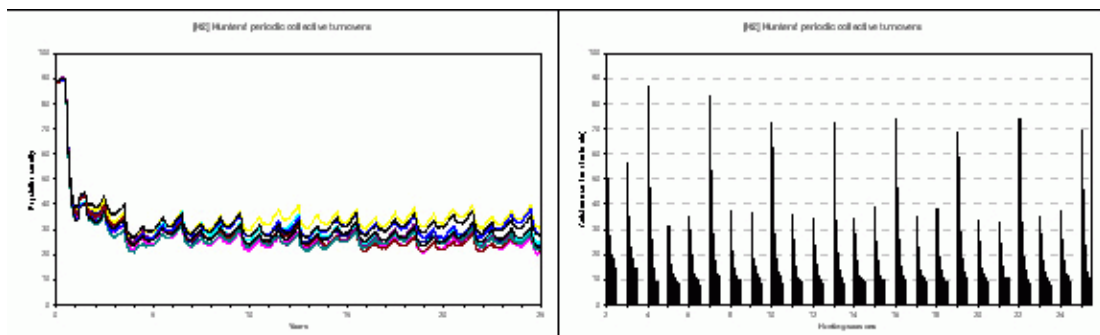


FIG. 4.9 – Scénario 5 : Chasseurs qui se coordonnent en groupes.

chasse. Le second est la densité de population après 25 ans et le dernier est la somme des captures pendant 25 ans.

Scénario	Surface (nombre de cellules)	chassée	Densité de population après 25 ans (nombre d'animaux / Km2)	Captures to- tales (nombre d'animaux)
1	722.9 (4.7)		33.64 (1.8)	14321.9 (405.7)
2	718.6 (7.4)		38.20 (1.1)	16036.8 (346.5)
3	717.3 (9.2)		23.62 (0.7)	13400.9 (219.6)

Tableau 2 : Synthèse des résultats pour les scénarios sans chasseurs

Scénario	Surface (nombre de cellules)	chassée	Densité de populaiaon après 25 ans (nombre d'animaux / Km2)	Captures totales (nombre d'ani- maux)
4	640.8 (28.4)		15.77 (3.4)	12073.7 (766.1)
5	747.2 (26.5)		24.21 (3.4)	13970.3 (824.3)

Tableau 3 : Synthèse des résultats pour les scénarios avec chasseurs

4.3.4 Interprétation

La première remarque concerne la comparaison des scénarii 1, 2 et 3. On peut discuter les effets comparés de la prise en compte d'une périodicité et d'une répartition spatiale. Si le scénario 2 qui présente à la fois une périodicité de la chasse et une stabilité spatiale donne les meilleurs résultats, on remarque aussi que le scénario 3 présente de meilleurs résultats que le scénario 1. Cela laisse penser que la gestion de la faune est plus sensible à l'allocation spatiale des pièges qu'à un arrêt de la chasse. Il semble ici que l'entité spatiale à considérer n'est pas le lieu-dit mais le sentier. Dans la réalité, il est bien présenté que ce sont les sentiers qui s'héritent de père en fils. Cette fixation des lieux préserve d'autres zones de prélèvements trop nombreux. Il semble bien que les six mois de repos accordés par les chasseurs à l'espace aient moins d'importance que le système de reconduction sur du long terme des sentiers. On a ainsi vu dans la littérature que deux mois de chasse suffisent à vider un espace de sa population. La recolonisation ne s'avère pas très rapide après ce « vidage ».

La deuxième remarque concerne la comparaison des simulations 4 et 5. Dans ces deux simulations les chasseurs interviennent. Dans la simulation 4, la rotation se fait au niveau de l'individu chasseur. Ce système de rotation n'a donc aucun sens, même pour des sentiers fixes, s'il ne s'accompagne d'une décision collective qui évite que l'espace laissé au repos par l'un soit chassé par l'autre. La différence entre les deux simulations porte sur deux facteurs. Dans les deux cas, chaque année ce sont 90 sentiers qui sont chassés (un par chasseur). Cependant dans le cas des simulations à décisions individuelles c'est un ensemble de 360 sentiers qui sont potentiellement chassés, tandis que dans le cas des simulations à décisions de groupe c'est un ensemble de $30 \times 4 = 120$ sentiers qui sont potentiellement chassés, parmi lesquels 30 sont protégés pour des durées de 3 ans.

Les résultats de ces simulations tendent donc à montrer la fonctionnalité des systèmes d'accès à l'espace des chasseurs. Par ailleurs, si l'arrêt de chasse pendant six mois soutient la durabilité du système de chasse, elle n'est pas le seul facteur qui joue dans la durabilité. La relation entre des lieux précis de l'espace, qui ne sont pas les lieux-dits mais plutôt les sentiers eux-mêmes, et les liens sociaux semble prendre aussi une importance dans la gestion de la faune sauvage. Ainsi, les décisions collectives de mise en protection mais aussi et surtout l'appropriation de lieux par des groupes de chasseurs d'une même famille ont, d'après les simulations des conséquences sur la pérennité de la chasse.

4.4 Discussion

Le travail qui a été effectué sur la chasse dans l'Est Cameroun amène bien des questions. Nous choisissons de discuter ici trois points : les résultats des simulations de la chasse, les problèmes de prise en compte de l'espace et les problèmes d'usage des outils de simulation.

Il s'agit tout d'abord d'une expérience méthodologique sur l'usage de systèmes multi-agents pour comprendre les interactions entre sociétés et ressources. D'autres études sur le sujet existent ([Bousquet and D'Aquino, 1999], [Barreteau and Bousquet, 2000]). Le processus mené ici a consisté à isoler quelques questions posées par le chercheur de terrain qui concernaient l'organisation des prélèvements dans l'espace et dans le temps. En l'absence de données sur les densités animales nous avons simulé une ressource virtuelle à partir de connaissances éthologiques pour tester des scénarios de chasse. Pour cela nous avons été amenés à développer les outils de simulation, ce qui s'est avéré par la suite très profitable pour des développements conceptuels en géographie ([Bousquet and Gauthier, 1999], [Le Page et al., 1999]) et pour de nombreuses autres applications. Les expérimentations portent sur la fonctionnalité de ce qui est présenté comme un système de gestion et tentent de faire la part entre l'influence des choix temporels et spatiaux. Les résultats montrent l'importance des choix spatiaux, ce qui permet de s'interroger sur les déterminants de ces choix sur le terrain. L'inscription des choix dans l'organisation sociale des chasseurs est soulignée.

Ce travail permet également de s'interroger sur la représentation de l'espace. La connaissance d'un espace passe par celle des lieux. Le rapport à ces lieux dépend de la relation entre l'individu et la communauté, de son statut dans le groupe qui seul a les moyens de contrôler cet espace. Au contraire ce qui fonde la connaissance d'un espace dans les représentations des gestionnaires des ressources, c'est la connaissance des limites de cet espace qui permet de le classer et éventuellement de l'affecter. C'est bien d'ailleurs cette confrontation que nous avons étudiée en partant de simulations de chasse sur des surfaces répertoriées à l'aide d'un Système d'Information Géographique pour aboutir à des simulations qui prennent en compte des lieux auxquels sont associés des sentiers de chasse, et qui sont l'inscription dans l'espace d'une relation dans des lignages. Les conventions topocentriques fondées sur une pratique des lieux, sont d'abord devenues une fois cartographiées, des limites géométriques. Nous nous sommes aperçus de l'inadéquation de cette représentation pour étudier et comprendre les interactions entre les paysans et leur ressource. De façon plus générale, la compréhension de la coordination passe plus par cet ensemble de lieux et de liens que par des portions d'espaces que l'on pourrait attribuer de manière différenciée à l'exploitation forestière, à l'agriculture ou à des réserves de faune par exemple.

4.5 Conclusion

Les simulations du modèle multi-agent de l'activité de chasse que nous avons présentées dans ce chapitre ont permis de tester la viabilité des scénarios de chasse observés sur le terrain. Nous avons validé le modèle de comportement du céphalophe mis en oeuvre en simulant la dynamique hors chasse de cette population. En choisissant diverses valeurs pour la taille de la population initiale, toutes les simulations convergent vers le même attracteur qui se situe autour d'une densité de population égale à 80 habitants / km². Cette densité de population correspond à celle qui est décrite dans la littérature, concernant les zones non chassées [Dubost, 1980]. Nous avons ainsi créé un monde virtuel sur lequel nous avons testé les divers scénarios de chasse.

Les résultats de ces simulations ont montré d'une part la fonctionnalité des systèmes d'accès à l'espace des chasseurs, et d'autre part que si l'arrêt de chasse pendant six mois soutient la durabilité du système de chasse, elle n'est pas le seul facteur qui influe sur cette durabilité. La relation entre des lieux précis de l'espace, qui ne sont pas les lieux-dits mais plutôt les sentiers eux-mêmes, et les liens sociaux semble prendre aussi une importance dans la gestion de la faune sauvage.

Comme dans toute expérience de modélisation, il se pose ici la question de la validation du modèle : quelle est l'influence des choix de simplification que nous avons fait sur les résultats issus des simulations et sur l'interprétation que l'on doit leur donner ? Comment peut-on extrapoler les résultats des simulations multi-agents ? Ces questions font l'objet de la dernière partie de cette thèse.

Troisième partie

**VALIDATION PAR LES RESEAUX
DE PETRI**

Chapitre 5

Modélisation par les Réseaux de Petri

5.1 Introduction

Un des intérêts majeurs de la modélisation multi-agent de la dynamique de la faune que nous avons menée est qu'elle s'inscrit dans un courant méthodologique très actuel en biologie des populations : la modélisation basée sur l'individu. Cependant, les simulations d'un système multi-agent correspondent à des évolutions particulières du modèle, c'est-à-dire à des chemins particuliers dans l'espace des états accessibles du système. A elles seules, elles ne renseignent pas sur la non existence de situations de blocage que nous cherchons à éviter, et ne permettent pas de répondre à certaines questions que l'on se pose sur la gestion de ressources renouvelables, notamment les questions sur la viabilité du système en fonction des conditions initiales et des hypothèses d'évolution. Il s'est donc avéré nécessaire de procéder à une étude formelle du système. L'équipe Systèmes Répartis Coopératifs (SRC) du LIP6 à l'Université Pierre et Marie Curie en France nous a accueilli pour mener à bien ce travail.

Bien que les équations différentielles nous offrent d'énormes possibilités d'analyse formelle, il est très difficile de traduire le modèle multi-agent dans ce formalisme sans le dénaturer totalement. C'est ainsi que nous avons eu recours aux méthodes formelles, techniques bâties à partir de deux ingrédients principaux : un langage de spécification et un système de vérification [Monin, 1996]. Les langages de spécification des méthodes formelles sont en général basés soit sur des langages ensemblistes (c'est le cas des méthodes Z, VDM et B), soit sur des algèbres (spécification algébrique).

Les réseaux de Petri sont à la fois un formalisme mathématique et un langage graphique pour modéliser les systèmes complexes. Ils nous semblent être un bon compromis entre les systèmes multi-agents qui offrent un riche pouvoir d'expression mais ne permettent pas de vérification formelle, et les modèles d'équations différentielles très adaptées à une analyse formelle, mais inefficaces lorsqu'il faut représenter des entités autonomes en interactions ou des comportements qualitatifs.

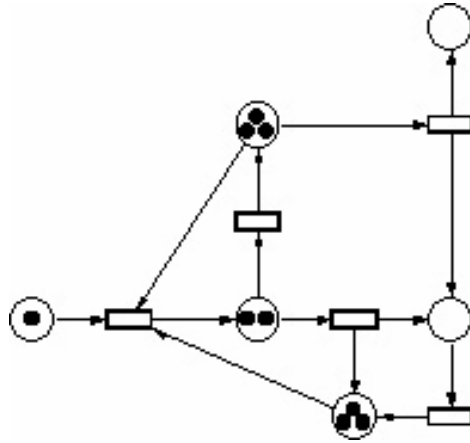


FIG. 5.1 – Exemple de réseau de Petri marqué

Comme nous allons le voir dans la suite de ce chapitre, les réseaux de Petri sont souvent utilisés pour modéliser les systèmes multi-agents ([Magnin and Ferber, 1993], [Vernadat et al., 1994], [El Fallah, 1994]). Cependant, ces travaux se sont généralement concentrés sur la représentation de modèles à agents cognitifs et communicants, en faisant l'hypothèse selon laquelle le nombre d'agents reste constant. Nous proposons une approche de modélisation par les réseaux de Petri colorés qui prend en compte l'espace et les interactions spatiales entre les différentes entités du système, de même que l'apparition ou la disparition d'agents.

Ce chapitre présente le modèle réseau de Petri de l'activité de chasse que nous avons élaboré, modèle qui sera analysé dans le chapitre suivant.

5.2 Les Réseaux de Petri

5.2.1 Définition informelle

Un réseau de Petri est un graphe biparti comprenant deux sortes de noeuds : les places et les transitions. Un graphe biparti se définit comme un graphe $G = (S, A)$ dans lequel S est l'ensemble des sommets (places et transitions) et A est l'ensemble des arcs. Dans un tel graphe, S peut être partitionné en deux ensembles S_1 et S_2 tels que $(u, v) \in A$ implique soit $u \in S_1$ et $v \in S_2$, soit $u \in S_2$ et $v \in S_1$. En d'autres termes, tous les arcs passent entre les deux ensembles S_1 et S_2 [Cormen et al., 1999]. On représente généralement les places par des cercles et les transitions par des barres ou des rectangles (figure 5.1). Les arcs d'un réseau de Petri relient donc les transitions aux places ou les places aux transitions. Les places contiennent des jetons ou marques qui se déplacent de place en place en franchissant les transitions suivant une règle dite de franchissement. Les réseaux de Petri permettent de modéliser les systèmes parallèles et offrent d'énormes possibilités d'analyse formelle du système, en s'appuyant sur l'algèbre linéaire.

Les réseaux de Petri ont des applications diverses. Ils permettent dans l'industrie de modéliser la supervision des systèmes discontinus [Champagnat, 1998]. Ils permettent également de modéliser des activités parallèles ou en concurrence, des protocoles de communication ou d'étudier des langages formels [Peterson, 1981].

5.2.2 Définition formelle

Un intérêt majeur des réseaux de Petri vient du fait qu'en plus d'être un outil graphique de modélisation, c'est aussi un formalisme mathématique. Un réseau de Petri est défini par $R = (P, T, W)$ [Naquet and Geniet, 1992] où

- P est un ensemble fini $\{p_1, \dots, p_m\}$ de places
- T est un ensemble fini $\{t_1, \dots, t_n\}$ de transitions
- $W : (P \times T) \cup (T \times P) \longrightarrow N$ est la fonction de valuation.

N représente l'ensemble des entiers naturels

Un marquage M est une fonction de P dans N qui définit la distribution des marques dans les places. A toute place p_i de P est associé le nombre de jetons qu'elle contient.

La fonction d'incidence avant (resp. arrière) notée Pre (resp. $Post$) est la restriction de W à $P \times T$ (resp. $T \times P$).

$M_0 : P \longrightarrow N$ est le marquage initial.

5.2.3 Règle de franchissement

La matrice de connexion est la matrice $C = C_{p,t}$ avec

$$c(p, t) = Post(p, t) - Pre(t, p) \quad (5.1)$$

Une transition t est franchissable pour un marquage M si et seulement si

$$\forall p \in P, M(p) \geq Pre(t, p) \quad (5.2)$$

A partir du marquage M , le franchissement de la transition t produit un marquage M' tel que

$$\forall p \in P, M'(p) = M(p) - Pre(t, p) + Post(p, t) \quad (5.3)$$

5.2.4 Théorème fondamental (Condition nécessaire d'accessibilité)

Si le marquage M_d est accessible à partir du marquage M_0 après le franchissement de la séquence de transitions $s = (s_1, s_2, \dots, s_d)$, alors

$$M_d = M_0 + CS \quad (5.4)$$

où S est un vecteur de taille $n \times 1$ dans lequel chaque S_j représente le nombre de fois où la transition t_j est franchie dans la séquence s .

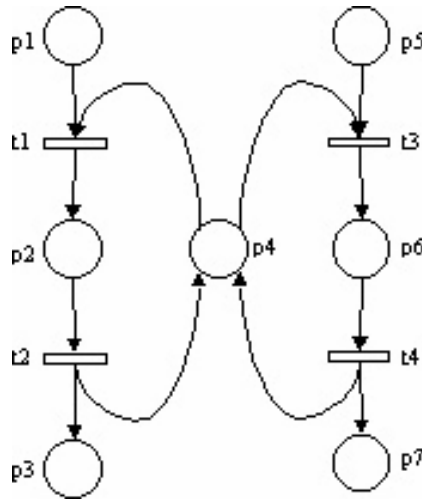


FIG. 5.2 – Modélisation du partage d’une ressource par un réseau de Petri

5.2.5 Exemple

Les réseaux de Petri sont utilisés dans la modélisation des systèmes pour exprimer une situation de partage d’une ressource commune. Lorsque la ressource est critique, l’exclusion mutuelle s’avère être un moyen efficace de gérer l’utilisation de celle-ci. La figure 5.2 représente un modèle de partage d’une ressource disponible dans la place p_4 par des processus localisés en p_1 et p_5 .

Ce réseau peut être défini mathématiquement par les ensembles

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\} \quad (5.5)$$

et

$$T = \{t_1, t_2, t_3, t_4\} \quad (5.6)$$

et la matrice de connexion

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.7)$$

Un élément $c_{i,j}$ de la matrice représente la variation du nombre de jetons de la place p_i après le franchissement de la transition t_j .

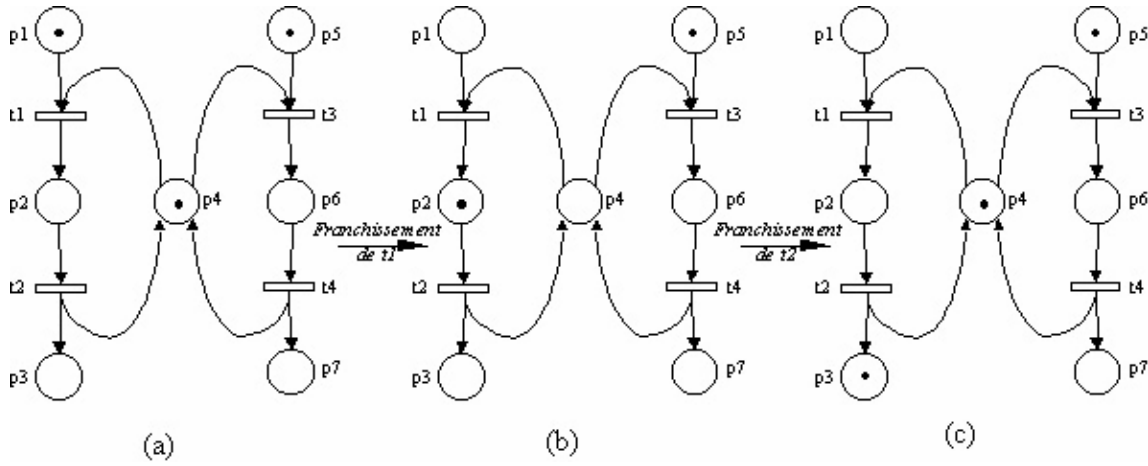


FIG. 5.3 – Franchissement de transitions dans un réseau de Petri

Considérons le marquage initial suivant

$$M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{5.8}$$

qui correspond à la présence d'un jeton dans les places p_1 , p_4 et p_5 pendant que les autres places restent vides (figure 5.3 a)). Les transitions t_1 et t_3 sont franchissables. L'une d'elles sera effectivement franchie et selon que t_1 ou t_3 a été franchie, on obtiendra un marquage pour lequel l'une des transitions t_2 ou t_4 sera franchissable.

La figure 5.3 illustre le franchissement de la séquence de transitions (t_1, t_2) .

Cette séquence est représentée par le vecteur

$$S = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{5.9}$$

dans lequel chaque s_j représente le nombre de fois où la transition t_j est franchie dans la séquence. En appliquant le théorème fondamental énoncé ci-dessus, on retrouve le marquage

M_2 de la figure 5.3 c).

$$M_2 = M_0 + CS = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad (5.10)$$

5.3 Systèmes multi-agents et réseaux de Petri

Les réseaux de Petri proposent des techniques algébriques de vérification de propriétés sur le comportement du système [Naquet and Geniet, 1992]. Cette vérification est structurale, c'est-à-dire indépendante de toute exécution et permet ainsi d'énoncer des propriétés générales sur le système. De nombreux travaux ont été faits sur la conception des systèmes multi-agents par les réseaux de Petri [Magnin and Ferber, 1993], [Vernadat et al., 1994] et [El Fallah, 1994].

5.3.1 Le formalisme BRIC

Magnin et Ferber [Magnin and Ferber, 1993] proposent la définition d'un système multi-agent par composants modulaires de type BRIC et réseaux de Petri. BRIC (Block-like Representation of Interactive Components) est un langage de haut niveau [Ferber, 1995] permettant de concevoir et de réaliser des systèmes multi-agents à partir d'une approche modulaire. Un système multi-agent est modélisé comme un ensemble de composants interconnectés. Chaque composant ou module est une structure logicielle qui est caractérisée extérieurement par des bornes d'entrée/sortie qui lui servent d'interface avec les autres composants. De manière interne, c'est un ensemble de composants interconnectés, dont le comportement est déjà spécifié ou directement un réseau de Petri. Le formalisme utilisé ici est celui des réseaux de Petri à arc inhibiteur. La liaison entre les composants se fait par leurs bornes d'entrée et de sortie grâce aux liens de communication. Ces liens correspondent à des arcs dans un réseau de Petri. La structure BRIC peut ainsi s'éclater en un réseau de Petri.

Ce formalisme peut être utilisé pour modéliser un système multi-agent communicant dans lequel les agents sont des composants dont les bornes d'entrée/sortie sont reliées aux bornes d'un module d'acheminement de message. Ce dernier se contente de récupérer tous les messages et à les envoyer aux agents destinataires à partir de l'adresse indiquée dans le message [Magnin and Ferber, 1993].

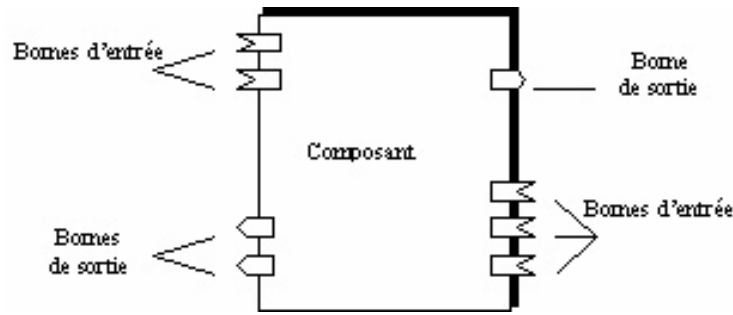


FIG. 5.4 – Structure d'un composant BRIC et de ses bornes

La modélisation de l'environnement [Ferber, 1995] en BRIC se fait sous la forme d'un composant qui reçoit les influences produites par les agents, les combine avec l'état courant de l'environnement et renvoie aux agents les informations sur le nouvel état (consommation, perceptions)

5.3.2 Le formalisme VAL

Vernadat propose une démarche de conception mixte basée sur deux approches complémentaires : le prototypage à l'aide d'outils formels (VAL) et la simulation dans l'environnement de programmation cible (PRAL).

Le modèle de description [Vernadat et al., 1994] est basé sur l'approche acteur, utilisant le langage acteur PRAL. Ce langage qui repose sur un modèle de calcul concurrent permet de concevoir une application comme un ensemble d'acteurs communiquant par envoi de messages asynchrones avec leurs accointances dans la topologie de communication. Un acteur au sens PRAL est un agent élémentaire autonome. Il lui est associé un automate qui décrit les états dans lesquels l'acteur est susceptible d'évoluer. Un acteur PRAL est alors défini par son interface ses accointances et l'ensemble de ses comportements admissibles.

Le formalisme VAL est une extension des réseaux de Petri Prédicat-Transition, dans lesquels les transitions portent des prédicats. Ce formalisme prend en compte de manière explicite la communication et le dynamisme. En plus des pré-conditions et des post-conditions des réseaux classiques, les transitions comprennent des clauses de synchronisation de rendez-vous, les messages attendus, les messages à émettre ainsi que les instances d'agent à créer.

Les acteurs PRAL sont traduits dans le formalisme VAL afin d'exploiter les potentialités de vérification de celui-ci. La vérification concerne les propriétés attendues du système. Elle se fait par construction du graphe d'accessibilité (l'espace d'états accessibles) du système. Ce graphe est comparé au graphe du comportement effectivement fourni par le système (équivalence observationnelle) [Vernadat et al., 1994].

5.4 Vers une approche basée sur les réseaux colorés

Les travaux de Ferber et Magnin d'une part et de Vernadat d'autre part ont montré l'intérêt de l'approche réseau de Petri dans la vérification des systèmes dynamiques. Cependant, les méthodes proposées laissent généralement de côté deux aspects fondamentaux des modèles de dynamique de ressources :

- Elles s'orientent principalement vers la représentation de modèles à agents cognitifs et communicants. La prise en compte de l'espace et des interactions spatiales entre les différentes entités du système n'est pas assez étudiée.
- La dynamique du système se limite aux changements d'états des instances d'agents existant au début de la simulation. Les agents peuvent apparaître et disparaître en cours de simulation, ce qui n'est pas exprimable en termes de création ou de suppression de modules ou de sous réseaux.

Nous proposons une approche de modélisation de systèmes multi-agents basée sur les réseaux de Petri colorés qui s'adapte mieux à la classe des modèles qui nous intéressent, c'est-à-dire l'étude des problèmes de gestion de ressources naturelles.

5.4.1 Les réseaux de Petri colorés

Afin d'enrichir le pouvoir d'expression des réseaux de Petri, le formalisme a été étendu et plusieurs variantes ont été proposées. Il s'agit notamment des réseaux Prédicats-Transitions dans lesquels les transitions portent des prédicats dont la valeur de vérité conditionne le franchissement, des réseaux de Petri colorés [Jensen, 1997], des réseaux de Petri à objets dans lesquels les marques sont des objets [Sibertin-Blanc, 1985], des réseaux synchrones et temporisés pour lesquels on associe une durée aux transitions ou aux places [Ramchandani, 1974]. Cependant, ces diverses classes de réseaux de Petri, en améliorant le pouvoir d'expression de ceux-ci, réduisent en même temps les facilités d'analyse formelles qu'ils offrent. Les réseaux colorés sont ceux qui offrent encore suffisamment de possibilités de vérification formelle de leurs propriétés et pour lesquels les outils associés sont assez disponibles.

Les réseaux de Petri colorés sont des réseaux de Petri dans lesquels les jetons portent des couleurs. Une couleur est une information attachée à un jeton. Cette information permet de distinguer des jetons entre eux et peut être de type quelconque. Ainsi, les arcs ne sont pas seulement étiquetés par le nombre de jetons mais par les couleurs de ces jetons. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires qui en plus satisfont les couleurs qui étiquettent les arcs. Après le franchissement d'une transition, les jetons qui étiquettent les arcs d'entrée sont retirés des places en entrée tandis que ceux qui étiquettent les arcs de sortie sont ajoutés aux places en sortie de cette transition.

Plus formellement, un réseau de Petri coloré est défini par un tuple $(\Sigma, P, T, A, N, C, G, E, I)$ où [Jensen, 1997] :

- Σ est un ensemble fini d'ensembles de couleurs. Chaque ensemble de couleurs étant fini et non vide
- P est un ensemble fini de places
- T est un ensemble fini de transitions
- A est un ensemble fini d'arcs tel que $P \cap T = P \cap A = T \cap A = \emptyset$
- $N : A \longrightarrow P \times T \cup T \times P$ est la fonction de connexité
- $C : P \longrightarrow \Sigma$ est la fonction de couleurs
- G est une fonction définie de T vers les expressions telles que

$$\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma] \quad (5.11)$$

- E est une fonction définie de A vers les expressions telles que

$$\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma] \quad (5.12)$$

où $p(a)$ est la place associée à $N(a)$ et C_{MS} dénote l'ensemble de tous les multi-ensembles sur C

- I est la fonction d'initialisation définie de P vers les expressions telles que

$$\forall p \in P : [Type(I(p)) = C(p)_{MS} \wedge Var(E(a)) = \emptyset]. \quad (5.13)$$

5.4.2 Modélisation par les réseaux de Petri colorés

La particularité de notre approche porte sur les points suivants :

- Nous proposons l'utilisation des modules (ou sous-réseaux) comme des classes d'agents et les jetons comme des agents, cela permet de modéliser l'apparition ou la disparition d'agents en termes de création ou suppression de jetons.
- Nous proposons de modéliser des systèmes multi-agents situés en représentant l'environnement comme un objet particulier du système, particulier par sa structure et ses liens avec les autres objets.

Cette approche, comme nous allons le voir par la suite, s'adapte bien à la représentation de systèmes multi-agents par les réseaux de Petri pour des problèmes de gestion de ressources renouvelables.

5.4.3 CPN-AMI

Une large gamme d'environnements de manipulation des réseaux de Petri coexistent dans la communauté scientifique. Selon les outils d'analyse qu'ils proposent, l'existence et la convivialité de leur éditeur graphique ainsi que le formalisme qui leur est associé, les

caractéristiques de ces environnements sont très diverses. L'un des environnements les plus utilisés est CPN-AMI.

Développé au Laboratoire d'Informatique de Paris VI (LIP6) de l'Université Pierre et Marie Curie, CPN-AMI est un environnement dédié à la théorie des réseaux de Petri colorés [Kordon and Paviot-Adet, 1999]. Il supporte la spécification, la validation, la vérification formelle et le prototypage conforme d'application coopératives. Ses outils permettent de vérifier des formules de logique temporelle (model checking), de calculer des propriétés structurelles (invariants, verrous, trappes...). La plate-forme permet également de simuler les réseaux de Petri colorés, de déboguer et d'engendrer du code. L'accès aux outils se fait à travers une interface utilisateur (Macao) qui est un éditeur de graphes générique.

AMI-Net, modèle mis en oeuvre par CPN-AMI, est un formalisme de spécification de réseaux de Petri colorés qui ajoute à la structure classique des réseaux places- transitions des informations textuelles sur les places, les transitions et les arcs. Ces informations permettent de définir la sémantique d'un réseau de Petri suivant ce formalisme.

Ainsi, à chaque place est associé un domaine qui définit l'ensemble des couleurs que peuvent prendre les jetons de cette place. La notion de domaine est ici équivalente à celle de type telle que définit dans les langages de programmation. Il n'est pas nécessaire de définir explicitement des domaines pour les transitions puisqu'ils se déduisent automatiquement de ceux des places qu'elles ont en entrée et des étiquettes portées par les arcs entrants. Par contre, AMI-Net offre la possibilité d'associer à une transition un prédicat qui doit être évalué à *VRAI* pour que la transition soit franchie.

5.5 Le modèle Réseau de Petri coloré

Le modèle réseau de Petri colorés de l'activité de chasse est fait de modules qui correspondent chacun à une hypothèse ou à un groupe d'hypothèses. Les hypothèses décrivent le modèle, et favorisent une approche modulaire. Dans ce modèle, les agents animaux sont représentés par des jetons de la place *Population*. A chaque animal correspondra un jeton regroupant les informations sur l'animal, notamment son âge, sa position dans l'espace et son état.

Nous utilisons une approche par composition de réseaux, dans laquelle les sous- systèmes ou modules sont d'abord conçus, en partageant entre elles les places de séquençement et des places globales du système. Chaque module comprend une place *Start_Hx* et une place *End_Hx* (non colorées et 1-bornées). Dès que *Start_Hx* contient un jeton, cela entraîne le démarrage d'une exécution du module *Hx*. Cette exécution s'achève avec la présence d'un jeton dans *End_Hx*. Ces places dites de séquençement permettront donc de "plonger" le module dans le système global, en servant d'interface entre les modules.

Les modules sont d'abord étudiés séparément et on peut vérifier certaines propriétés locales au niveau de ces modules. Ils sont par la suite assemblés en fusionnant les places

partagées pour obtenir le système global.

5.5.1 Les hypothèses du modèle

A partir du modèle multi-agent du système décrit précédemment, nous avons retenu les hypothèses ci-dessous :

- H1a : Les animaux se déplacent de façon aléatoire dans les 4 directions
- H1b : Les animaux qui se déplacent sur une cellule piégée sont pris dans le piège suivant une probabilité de prise
- H1c : Les animaux pris dans un piège sortent du système (mort)
- H1d : Le nombre d'animaux dans une cellule est borné
- H2 : La population d'animaux augmente de 20% par an
- H3a : Les animaux grandissent
- H3b : Les animaux ayant atteint l'âge *ageMax* sortent du système (mort)
- H4 : Le séquençement des différentes actions est bien défini
- H5 : Le déplacement des animaux spécifié par les hypothèses H1x concerne à chaque pas de temps tous les animaux du système.

Elles correspondent aux aspects du système que nous jugeons pertinents pour la modélisation par les réseaux de Petri.

5.5.2 Description des places principales

La place *Population* contient les jetons représentant les animaux du modèle. Un animal est représenté par un jeton $\langle \text{Age}, X, Y, \text{Etat} \rangle$ où *Age* est son âge en semestres, avec pour valeurs dans *INT_A*, *X* et *Y* sont les coordonnées de sa position dans l'espace, *X*, *Y* dans *INT_E*, *Etat* est son état et prend ses valeurs dans *ETAT_A*.

La place *Espace* contient des jetons représentant les cellules spatiales. Chaque cellule (jeton) est un triplet $\langle X, Y, \text{Etat} \rangle$, où *X* et *Y* sont ses coordonnées et *Etat* son état (valeurs dans *ETAT_E*).

La place *EspVide* contient des jetons $\langle X, Y \rangle$ représentant les positions libres dans les cellules spatiales. Elle permet de modéliser le nombre maximum d'animaux par cellule. A chaque cellule $\langle X, Y, \text{Etat} \rangle$ correspondent *m* cellules $\langle X, Y \rangle$ de la place *EspVide*, si cette cellule peut accueillir *m* animaux supplémentaires.

La place *Taille* contient des jetons non colorés, en même nombre que ceux de la place *Population*. Son marquage correspond à chaque instant à la taille de la population.

La place *TailleB* contient des jetons non colorés. C'est la place complémentaire de la place *Taille*. La somme des marquages des deux places est constante, et est égale à la borne supérieure de la taille de la population.

Les places *Tmax* et *TmaxB* non colorées sont complémentaires l'une de l'autre et sont utilisées dans le mécanisme de pompage de la place *Taille*.

L'utilisation des places « duales » `TailleB` et `TmaxB` est l'astuce que nous avons trouvée pour contrôler l'instant où une place est vide. En général, ces situations sont modélisées par l'utilisation d'un arc inhibiteur, ce qui n'est pas pris en compte dans le formalisme AMI-Net.

5.5.3 Module de déplacement

Les animaux se déplacent de façon aléatoire dans les 4 directions. Ils peuvent être pris dans les pièges et le nombre d'individus par cellule est borné.

Ce module représente le processus d'un "déplacement" d'un animal. Ce "déplacement" peut correspondre à un changement de position (vers une des quatre cellules voisines) ou au fait de rester sur place. Dans l'un et l'autre cas, si la cellule d'origine est piégée, il est évalué pour cet animal une probabilité de prise et il peut alors être sorti du système en cas de prise, passer à la cellule d'arrivée ou rester sur place. Le processus contrôle qu'en cas de déplacement, le nombre d'animaux dans la cellule d'arrivée ne dépasse pas le maximum fixé.

Pour représenter un déplacement dans une direction donnée, le jeton représentant l'animal (`<Age,X,Y,Etat>` de la place `Population`), celui représentant sa cellule de départ (`<X,Y,Attr>` de la place `Espace`) et celui de la disponibilité de la cellule d'arrivée (`<f(X),f(Y)>` de la place `EspVide`) sont requis en entrée de la transition, avec :

$$f(X) = \begin{cases} X \text{ ou} \\ X + 1 \text{ ou} \\ X - 1 \end{cases} \quad (5.14)$$

On associe aussi un jeton qui exprime la disponibilité de la cellule d'arrivée. Cette disponibilité est représentée dans la place `EspVide` qui contient à chaque instant, pour une cellule de coordonnées `<X,Y>`, autant de jetons `<X,Y>` qu'on peut ajouter d'animaux dans cette cellule. Cette information y est représentée pour toutes les cellules de l'espace.

Lorsque la place `Start_H1` est marquée, l'une des transitions `SurPlace`, `Haut`, `Bas`, `Gauche` et `Droite` peut être franchie. Après ce franchissement, dans le cas d'une transition de déplacement effectif, un jeton `<X,Y>` est retourné dans la place `EspVide` pour modéliser qu'une position dans cette cellule a été libérée. Le jeton `<X,Y,Attr>` est retourné dans la place `Espace` qui doit contenir à chaque instant toutes les cellules de l'espace. Les informations des jetons en entrée sont "factorisées" dans la place `P1` et au passage, les coordonnées des cellules de départ sont modifiées (la fonction f précédente) pour représenter celles de la cellule d'arrivée. On peut alors passer à la phase de l'évaluation du piégeage.

Si la cellule d'origine n'est pas piégée, la transition `NonPiege` sera franchie et le jeton `<Age,X,Y,Etat>` représentant l'animal est ajouté à la place `Population`. Si la cellule est piégée, l'une des transition `PiegeOK` ou `PiegePB` sera franchie suivant que la valeur du

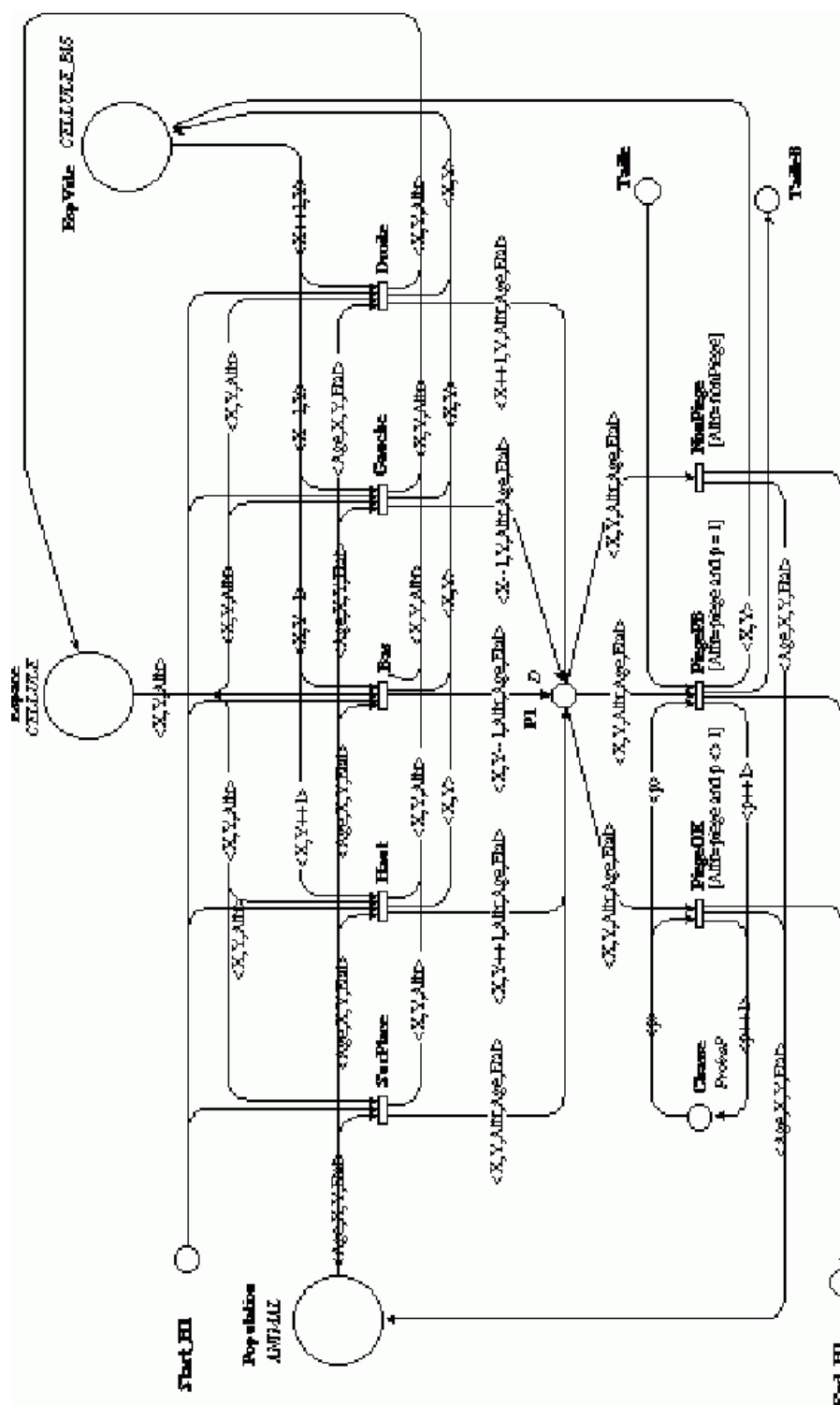


FIG. 5.5 – Module de déplacement et de prélèvement par piégeage

jeton $\langle p \rangle$ de la place **Chasse** est différente ou égale à 1. Ce choix correspond à la mise en oeuvre de la probabilité d'être pris dans le piège. Le jeton $\langle p \rangle$ est incrémentée et est remis dans la place **Chasse**, et dans le cas où la transition **PiegeOK** a été franchie, le jeton $\langle \text{Age}, X, Y, \text{Etat} \rangle$ est remis dans la place **Population**. Autrement (transition **PiegePB** franchie), il sort du système. On met à jour la place **Taille** contenant la taille de la population (en retirant un jeton) et sa place complémentaire **TailleB** (en lui ajoutant un jeton). Cette étape produit un jeton dans la place **End_H1**, ce qui marque la fin du processus.

5.5.4 Module de dynamique de la population

Croissance de la taille de la population

On considère que la taille de la population est bornée par B . A la place **Population** est associée une place **Taille** qui contient autant de jetons non colorés qu'il y a d'animaux. La place **Taille** possède une place "complémentaire" **TailleB** (non colorée elle aussi). Soit X le nombre d'animaux dans le système, **Taille** contient X jetons et **TailleB** en contient $B-X$. La somme des marquages de ces deux places est constante : B . De même, les places **Tmax** et **TmaxB** sont complémentaires. La somme de leurs marques est toujours égale à B .

Pour une croissance de la population de $x\%$, soit $n = 100/x$. Pour tout groupe de n individus, on produit un de plus.

Lorsque la place **Start_H2** est marquée, (les dispositions ayant été prises pour que **Taille** contienne X jetons et **TailleB** $B-X$ jetons), la transition **div** est franchie. Ce franchissement prend n jetons de **Taille** et en produit $n+1$ dans **Tmax**. Un jeton animal $\langle \text{Age}, X, Y, \text{Etat} \rangle$ est produit et ajouté à la place **Population**. La transition **div** sera franchie jusqu'à ce qu'il reste moins de n jetons dans la place **Taille**. Une des transitions **div_i** ($1 \leq i < n$) sera alors franchie.

Cette étape se termine avec B jetons dans la place **TailleB**. La transition **tr1** peut alors être franchie et elle produit un jeton dans la place **StartP2** qui déclenche l'étape de transfert des jetons de **Tmax** vers **Taille**. Cela se fait par les franchissements de **tr2**. La transition **tr3** sera franchie lorsque la place **TmaxB** contiendra B jetons (ie **Tmax** vide). Ce franchissement produit un jeton dans la place **End_H2** et marque la fin du processus de croissance de la population.

5.5.5 Module de croissance des individus

Les animaux grandissent et ceux ayant atteint ageMax sortent du système.

La place **Population** contient initialement tous les jetons représentant les animaux du modèle. Chaque jeton est retiré de **Population**, on incrémente l'âge et il est ensuite mis dans la place **Pop_Bis**. Lorsque la place **Population** est vide, on transfère tous les jetons

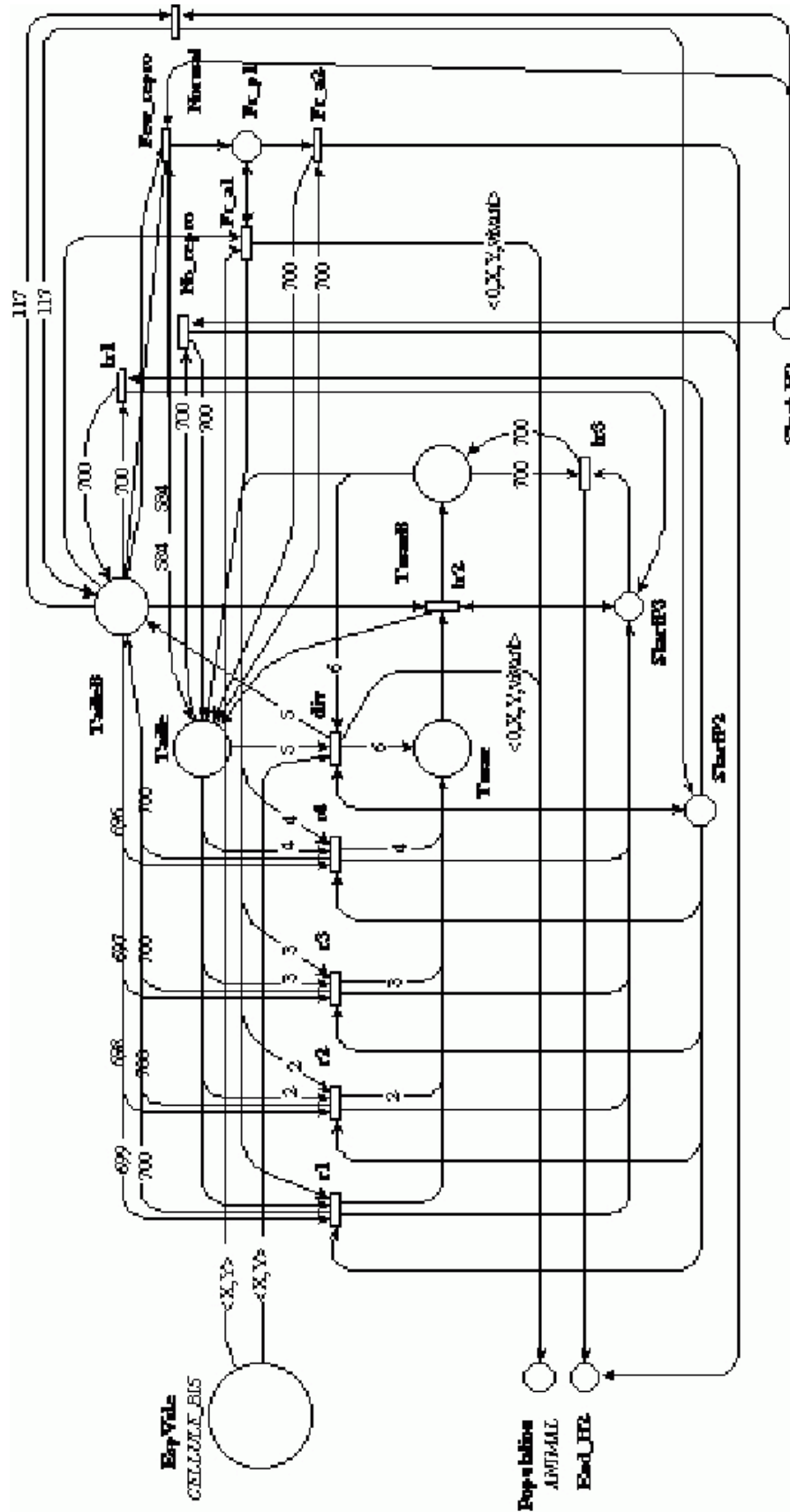


FIG. 5.6 – Module de croissance de la population

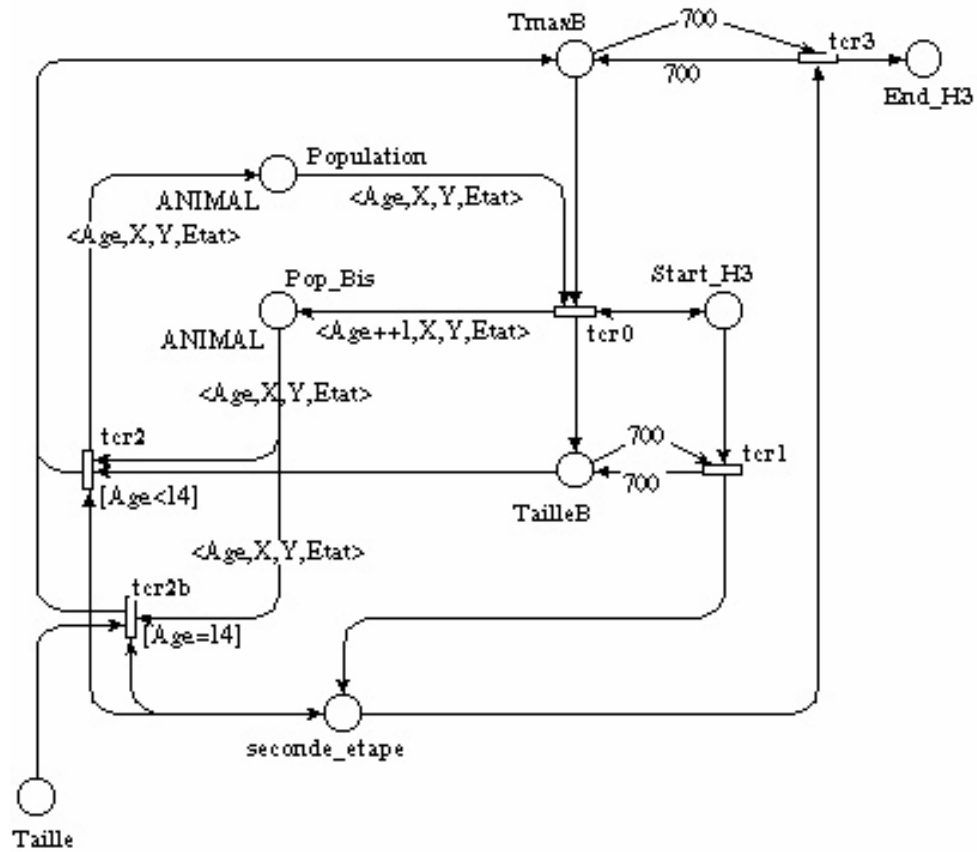


FIG. 5.7 – Module de croissance des individus

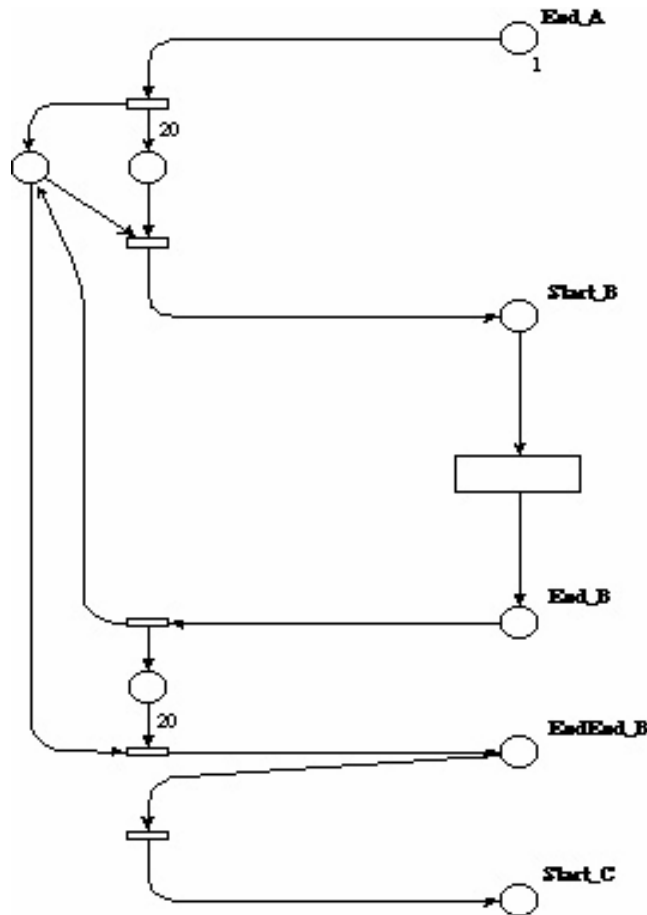


FIG. 5.8 – Module de séquençement

de `Pop_Bis` dans `Population`. Au moment de transférer `Population` dans `Pop_Bis`, les jetons pour lesquels l'âge est égal à `ageMax` sont retirés du système.

Lorsque la place `Start_H3` est marquée (début du processus), la transition `tcr0` est franchie jusqu'à ce que la place `Population` se vide. Ce franchissement retire chaque jeton de `Population`, incrémente sa variable `Age` et le met dans `Pop_Bis`. La place `Population` est vide lorsque la place `TailleB` contient `B` jetons. La transition `tcr1` est alors franchie et produit un jeton dans la place `seconde_etape` qui démarre l'étape de transfert des jetons de `Pop_Bis` vers `Population`. Pour chaque jeton de `Pop_Bis`, la transition `tcr2` ou `tcr2b` sera franchie selon que la variable `Age` a atteint `ageMax` ou non. Si le jeton doit sortir du système ($Age = ageMax$), un jeton doit être retiré de la place `Taille` (qui compte le nombre de jetons de la place `Population`). Cette étape se termine quand la place `Pop_Bis` se vide (ie la place `TmaxB` contient `B` jetons) et la transition `tcr3` est alors franchie. Elle produit un jeton dans la place `End_H3`, ce qui marque la fin du processus.

5.5.6 Module de séquençement

On représente ici le modèle de connexion des modules entre eux. Le séquençement des actions définit quel est l'ordre dans lequel les actions sont effectuées et combien de fois chaque action l'est. Nous présentons en exemple sur la figure 5.8 comment plonger un module B entre les modules A et C, en supposant que le module B s'exécute après A et qu'il s'exécute 20 fois avant qu'on passe au module C.

5.5.7 Les paramètres du modèle

5.5.7.1 La probabilité de prise dans un piège

Le fait qu'un animal se trouve dans une cellule piégée ne se traduit pas automatiquement par la mort de l'animal. On modélise la possibilité d'être pris dans le piège par une probabilité qui l'exprime. Dans le modèle multi-agent, elle vaut 0.01. Sachant que le pas de temps du modèle réseau de Petri vaut 26 pas de temps du modèle multi-agent (c'est-à-dire le nombre de semaines dans un semestre), nous avons calculé la probabilité pour un animal de mourir dans un pas de temps du modèle Réseau de Petri.

Soient les événements A et B suivants :

A : « mourir au cours d'un pas de temps »

B : « mourir au cours de k pas de temps »

On a,

$$p(A) = 0.01 \quad (5.15)$$

Pour calculer $p(B)$, il suffit de remarquer que l'événement \overline{B} qui est « ne pas mourir au cours de k pas de temps » a pour probabilité

$$p(\overline{B}) = p(\overline{A})^k \quad (5.16)$$

Donc

$$p(B) = 1 - p(\overline{B}) = 1 - (1 - p(A))^k \quad (5.17)$$

Ce qui donne en prenant $k = 26$,

$$p(B) = 0.23 \quad (5.18)$$

Nous utilisons la valeur 1/4 dans le modèle, par souci de simplification.

5.5.7.2 Le nombre maximum d'animaux dans une cellule m

Le nombre d'individus par cellule tient compte du fait que lorsque deux adultes fondent une famille, ils restent positionnés dans la cellule. Les plus jeunes se déplacent aléatoirement sur tout l'espace jusqu'à atteindre l'âge adulte pour s'installer à leur tour. On estime qu'en plus des 2 adultes sur une cellule, on n'aura pas plus de 5 jeunes de passage à la fois. C'est la raison pour laquelle m est fixé à 7.

5.5.7.3 Le taux d'accroissement de la population

Nous ne modélisons pas explicitement les naissances puisque le modèle ne prend pas en compte les différences de sexe entre les individus. Le modèle Réseau de Petri de la chasse ne simule pas pour le moment une dynamique hors chasse. Nous considérons un taux d'accroissement de 20% tous les 6 mois. La seule contrainte est que l'accroissement de la population sera régulée par la borne maximale qui est fixée à B , valeur elle-même déduite du nombre maximal d'individus par cellule.

5.5.7.4 La taille de la grille

Nous travaillons sur une grille virtuelle de taille 10×10 .

5.5.7.5 La taille maximale de la population B

Elle est déduite du nombre maximum d'individus par cellule et de la taille de la grille. Elle vaut donc, pour une grille de 10 cellules par côté, $10 \times 10 \times 7 = 700$.

5.5.7.6 Age maximum des individus

Le pas de temps vaut 6 mois, et on estime l'âge maximum réel à 7 ans. L'âge maximum des individus dans notre modèle sera donc fixé à 14 pas de temps de la simulation.

5.5.7.7 Initialisation du modèle

Soient N la taille de la population initiale, B la taille maximale de la population, x et y les nombres de lignes et de colonnes de l'espace, m le nombre maximum d'animaux par cellules. Le modèle est initialisé de la manière suivante :

- N jetons $\langle \text{Age}, X, Y, \text{Etat} \rangle$ sont créés dans la place **Population**
- N jetons non colorés sont créés dans la place **Taille**
- $B - N$ jetons non colorés sont créés dans la place **TailleB**
- B jetons non colorés dans la place **TmaxB**
- $x \times y$ jetons $\langle X, Y, \text{Attr} \rangle$ dans la place **Espace**
- $m \times x \times y - N$ jetons $\langle X, Y \rangle$ dans la place **EspVide**, correspondant aux positions libres dans les cellules
- un jeton $\langle 1 \rangle$ dans la place **Chasse**
- un jeton non coloré dans la place **Start_H3**
- Toutes les autres places sont vides.

Les ensembles des valeurs sont définies comme suit :

- $\text{INT_A} = [0..13]$
- $\text{INT_E} = [1..10]$
- $\text{ETAT_A} = \{\#\text{vivant}\}$

– ETAT_E = {#piege, #nonPiege}

5.6 Conclusion

Nous avons au cours de ce chapitre présenté le modèle de l'activité de chasse par un réseau de Petri coloré. Nous avons procédé par une approche de composition de réseaux, en concevant d'abord des modules correspondant à des groupes d'hypothèses. Ces modules sont ensuite assemblés pour former le modèle complet, par fusion des places communes.

Nous avons mis en oeuvre des mécanismes sur les réseaux de Petri colorés pour modéliser certaines dynamiques comme le choix de la direction de déplacement ou le comportement du système lorsque la cellule de destination de l'animal est déjà occupée, de même que la prise en compte des événements aléatoires comme la probabilité pour un animal d'être pris dans un piège. Il a aussi fallu maintenir la cohérence dans les places duales du système, ce que nous avons réalisé en proposant des mécanismes appropriés.

Nous avons par la suite procédé à l'analyse de ce modèle. C'est l'objet du chapitre suivant.

Chapitre 6

Validation formelle

6.1 Introduction

Il existe deux approches principales d'analyse d'un réseau de Petri : l'identification des propriétés structurelles et l'analyse par model checking du graphe des marquages accessibles [Jensen, 1997]. L'analyse structurelle est fondée sur les propriétés algébriques du réseau, tandis que le graphe des marquages accessibles est un graphe orienté dont les noeuds correspondent aux états accessibles du système et les arcs aux transitions franchissables entre ces états.

Les simulations de la dynamique du réseau de Petri nous ont permis de valider chaque module pendant son élaboration, en vérifiant s'il présente le comportement attendu sur de petits marquages initiaux. Après la phase de débogage, nous avons vérifié la cohérence du système grâce aux propriétés d'invariants structurels.

6.1.1 Analyse structurelle

Nous avons au cours de cette phase d'analyse calculé les invariants structurels du modèle réseau de Petri. On distingue deux types d'invariants : les invariants de places et les invariants de transitions. Dans notre modèle, il est particulièrement intéressant de s'assurer que les propriétés suivantes sont effectivement vérifiées :

$$\text{Taille} + \text{TailleB} = B$$

$$\text{Tmax} + \text{TmaxB} = B$$

CPN-AMI nous a effectivement fourni ces propriétés, et beaucoup d'autres qui ne sont pas pertinentes du point de vue de notre modèle. L'analyse structurelle a principalement servi à valider les mécanismes de contrôle des processus.

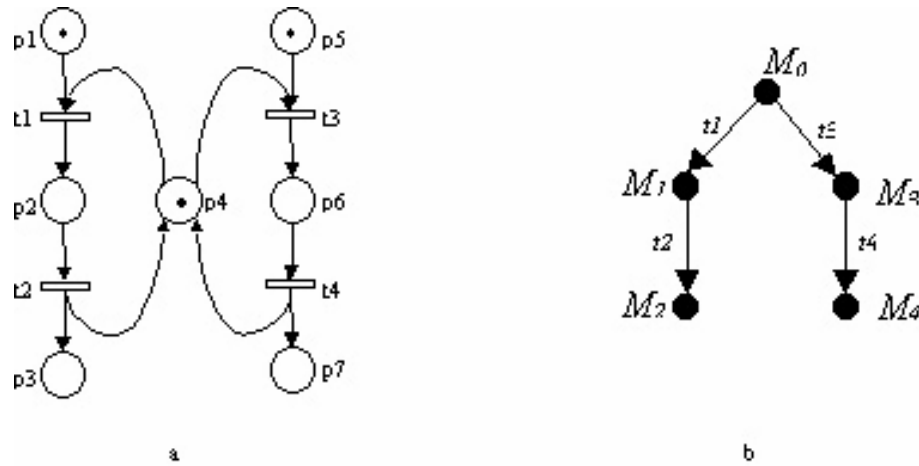


FIG. 6.1 – Un réseau de Petri marqué (a) et son GMA (b)

6.1.2 Le graphe des marquages accessibles

C'est un graphe orienté dont les noeuds correspondent aux états du modèle et les arcs aux franchissements des transitions. Reprenons le réseau de Petri marqué de la figure 5.3 a).

A partir du marquage initial M_0 , les transitions t_1 et t_3 sont franchissables. Le franchissement de t_1 produit le marquage M_1 tandis que le franchissement de t_3 produit le marquage M_3 . Lorsque le marquage M_1 est atteint, seule la transition t_2 est franchissable et son franchissement conduit au marquage M_2 . De même, à partir de M_3 , la seule transition franchissable est t_4 et conduit au marquage M_4 . Les marquages M_2 et M_4 sont des états du réseau à partir desquels aucune transition n'est franchissable. Ils correspondent à des situations de blocage.

$$\left\{ \begin{array}{l} M_0^T = (1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0) \\ M_1^T = (0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0) \\ M_2^T = (0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0) \\ M_3^T = (1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0) \\ M_4^T = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1) \end{array} \right. \quad (6.1)$$

Le GMA fournit donc de manière exhaustive l'ensemble des configurations possibles du système, et on peut ainsi rechercher la présence ou l'absence de configurations particulières. Bien que le graphe de l'exemple ci-dessus soit assez compact, on peut noter qu'à partir de chaque noeud partent autant d'arcs que de transitions franchissables. Ce qui conduit généralement à des graphes de très grande taille.

Pour les paramètres que nous nous sommes fixés dans le modèle réseau de Petri de la chasse, le GMA atteint très rapidement une taille énorme (plusieurs millions de noeuds) et est donc difficile à manipuler. Nous avons donc étudié le comportement du système sur des

exemples de taille réduite, ce qui a permis d'identifier les principales causes de l'explosion de la taille du GMA.

Nous avons ainsi noté que la taille du GMA est énorme à cause de plusieurs facteurs :

- les permutations des jetons de la place `Population` dans le module de déplacement
- les permutations des jetons de la place `Population` dans le module de croissance des animaux
- le choix des directions de déplacements
- le choix de la cellule qui accueille un nouvel agent dans le module de croissance de la population.

Ces situations sont toutes celles qui autorisent des alternatives dans l'évolution du modèle, soit parce que plusieurs transitions sont simultanément franchissables, soit parce que le franchissement de la même transition admet plusieurs possibilités de retrait des jetons dans les places en entrée.

6.2 Optimisation du modèle

Dans le but de réduire la taille du graphe des marquages accessibles, nous avons procédé à des simplifications pour supprimer des branches jugées inutiles. Le GMA a produit une explosion combinatoire dont une part importante est due aux permutations sur les jetons de la place `Population`. La simplification consiste donc à imposer un ordre dans ces jetons et séquentialiser ainsi les actions qui font intervenir ces jetons. Cette simplification nous a aussi permis de prendre en compte l'hypothèse H1e qui stipule que tous les animaux effectuent une action de déplacement.

Quant à la partie déclarative du modèle, nous ajoutons aux jetons $\langle \text{Age}, X, Y, \text{Etat} \rangle$ de la population un champ `Id`. Ces jetons deviennent donc $\langle \text{Id}, \text{Age}, X, Y, \text{Etat} \rangle$. Le champ `Id` est un identifiant entier, une sorte de numéro d'ordre unique pour chaque jeton, qui permet de les utiliser dans un certain ordre (ici décroissant). La valeur maximale courante du champ `Id` correspondant à la taille de la population, cette modification influe aussi sur la façon de gérer l'accroissement de la population ou le retrait d'un jeton.

6.2.1 Module de déplacement révisé H'1

Nous avons utilisé le mécanisme représenté à la figure 6.2 pour contraindre le choix de la direction de déplacement. On peut observer qu'à partir d'un marquage quelconque, on aura au plus une transition de déplacement qui est franchissable car ce franchissement est désormais conditionné par la présence d'un jeton non-coloré qui joue le rôle de bascule entre ces différentes transitions.

Ce module ne correspond plus à un déplacement d'un animal, mais plutôt au déplacement de tous les animaux du système. Il est donc équivalent à autant d'exécutions du

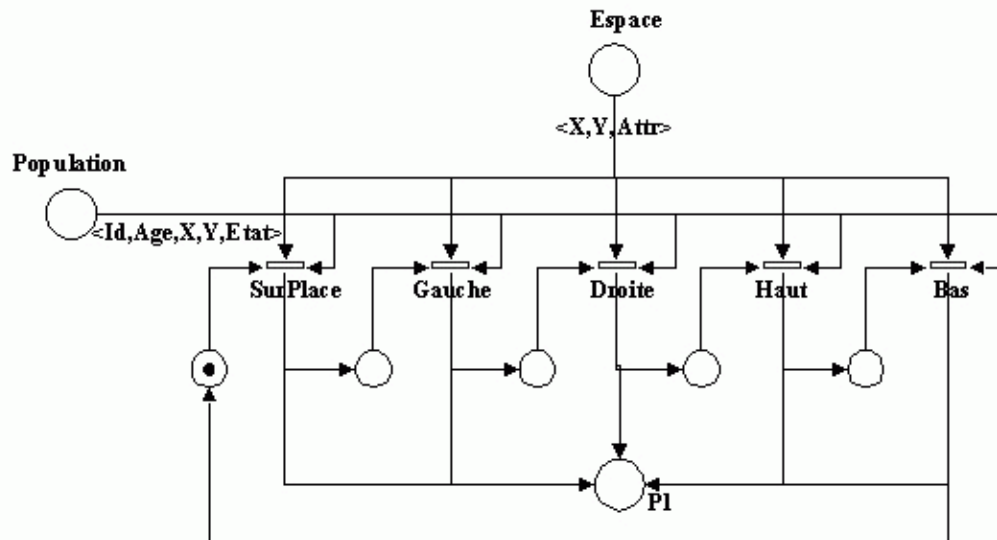


FIG. 6.2 – Le choix de la direction de déplacement

module H1 qu'il y a de jetons dans la place **Population**. Il correspond en fait à la prise en compte des hypothèses H1x et de H5.

Les places **Start_H1**, **Espace**, **EspVide**, **P1**, **Chasse**, **End_H1** sont les mêmes que dans H1. La place **Population** contient des jetons $\langle Id, Age, X, Y, Etat \rangle$. La place **Taille** contient un jeton $\langle IdT \rangle$, de même nature que **Id** (un entier), et représentant à chaque instant la plus grande valeur de **Id** dans les jetons $\langle Id, Age, X, Y, Etat \rangle$ de la place **Population**. La place **TailleBis** contient un jeton $\langle Id \rangle$ qui à chaque instant identifie le jeton $\langle Id, Age, X, Y, Etat \rangle$ en cours de traitement.

Lorsque la place **Start_H1** est marquée, le franchissement de la transition **copie** permet de "copier" la taille de la population $\langle IdT \rangle$ de **Taille** vers **TailleBis**. C'est le contenu de **TailleBis** qui sera décrémenté au fur et à mesure que le processus évolue. Ensuite, une des transitions **SurPlace**, **Haut**, **Bas**, **Gauche** ou **Droite** sera tirée selon la direction de déplacement (ou le fait de ne pas se déplacer) qui est ici un choix aléatoire. Ce tir prend dans la place **Population** le jeton qui a dans son champ **Id** la même valeur que celle de **Id** dans la place **TailleBis**. La valeur de $\langle Id \rangle$ dans la place **TailleBis** détermine donc le choix du jeton $\langle Id, Age, X, Y, Etat \rangle$ dans **Population**, et ce dernier détermine le choix d'un jeton $\langle X, Y, Attr \rangle$ dans la place **Espace**. La transition à franchir conditionne le retrait du jeton $\langle f(X), f(Y) \rangle$ de la place **EspVide** correspondant à la destination de déplacement.

Les informations ainsi collectées sont factorisées dans la place **P1** comme dans le module H1 et en fonction des valeurs de **Attr** et $\langle p \rangle$ de la place **Chasse**, une des transitions **PiegeOK**, **PiegePb** ou **NonPiege** sera tirée. Dans le cas où **NonPiege** ou **PiegePb** est tirée (ie que l'animal n'est pas pris dans le piège), le jeton $\langle Id, Age, X, Y, Etat \rangle$ est remis dans la place **Population** et le jeton $\langle Id-1 \rangle$ est mis dans la place **TailleBis** pour permettre de traiter

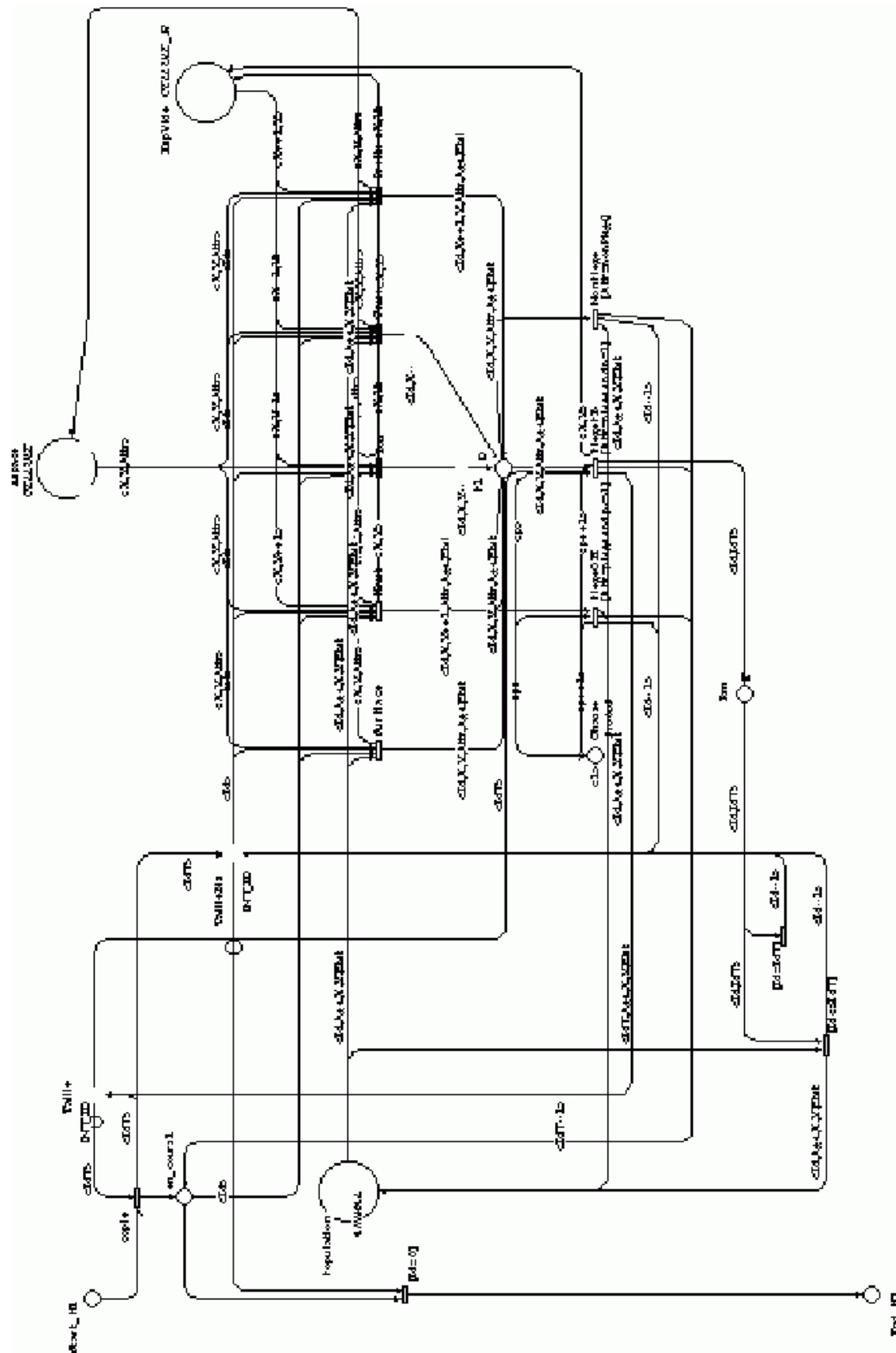


FIG. 6.3 – Module de déplacement et de prélèvement par piégeage, sans les permutations

le jeton suivant de `Population` dont l'identifiant est `<Id-1>`. Par contre si la transition `PiegeOK` est tirée (l'animal est pris dans le piège), sa position dans l'espace est libérée et le jeton `<X,Y>` est donc ajoutée à la place `EspVide`, le jeton `<IdT-1>` remplace le jeton `<IdT>` dans la place `Taille` (notons que cette transition retire le jeton `<IdT>` de la place `Taille`) et les jetons `Id` et `IdT` sont mis dans la place `Rm`, ce qui enclenche un mécanisme permettant d'allouer au jeton de la place `Population` qui a pour identifiant `IdT` l'identifiant `Id` qui vient d'être libéré. Ce mécanisme garantit qu'à chaque instant, `IdT` représente bien la taille de la population et que les identifiants utilisés dans la place `Population` sont bien toutes les valeurs entières comprises entre 1 et `IdT`.

Ce mécanisme consiste à tirer l'une des transitions `Iegal` ou `Idiff`, suivant les valeurs de `Id` et `IdT`. Si elles sont égales, `Iegal` est tirée et on se contente de décrémenter la valeur de `Id` avant de le remettre dans la place `TailleBis`, car le jeton qui a été retiré du système était déjà celui avec la plus grande valeur de `Id`. Si par contre ces valeurs sont différentes, `Idiff` est tirée et en plus de décrémenter la valeur de `<Id>` pour le remettre dans `TailleBis`, le jeton de la place `Population` qui a pour identifiant `IdT` se voit attribuer l'identifiant `Id`.

La présence d'un jeton `<Id>` dans la place `TailleBis` enclenche à nouveau le processus de déplacement de l'animal représenté par le jeton de la place `Population` qui a cet identifiant. Ce processus se poursuit jusqu'à ce que `<Id>` prenne la valeur 0, et la transition `findepl` sera alors tirée. Cette transition produit un jeton dans la place `End_H1`, ce qui marque la fin du module.

6.2.2 Les autres modules

Nous avons modifié les modules de croissance des individus et de la population pour tenir compte de cette contrainte. De même, le séquençement des modules prend en compte le fait que le module `H1` correspond au déplacement de tous les individus de la population, et non plus d'un seul individu.

6.3 Analyse du modèle optimisé

Malgré la réduction de la complexité du GMA par élimination des permutations, il reste toujours difficile à maîtriser. Cependant, en choisissant de petites valeurs pour les paramètres, nous avons pu apprécier le gain dû à la suppression des permutations dans le modèle Réseau de Petri. En calculant les GMA pour des espaces 3×3 , avec 4 individus maximum par cellule et un âge maximum de 4 saisons, nous observons que, le système converge vers un état terminal avec extinction de la population quand celle-ci est fixée au départ à une taille inférieure ou égale à 5. Ce phénomène est certainement dû au fait que la population est trop faible pour s'accroître et l'effondrement du système est alors inévitable.

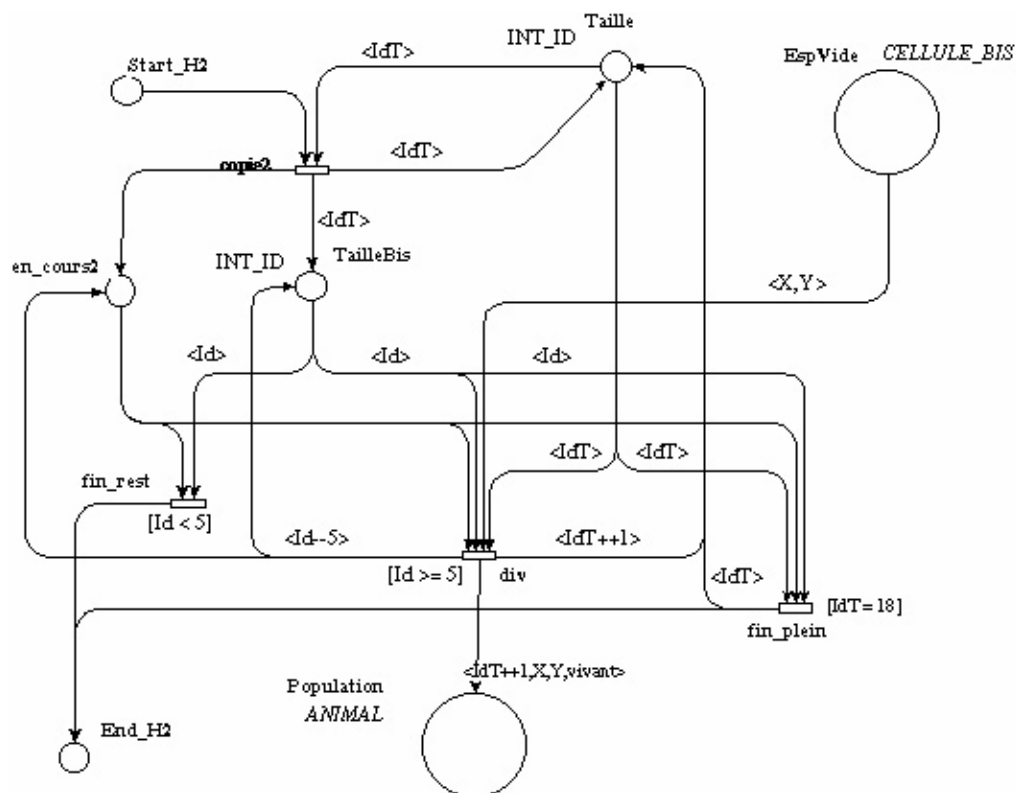


FIG. 6.4 – Module révisé de croissance de la population

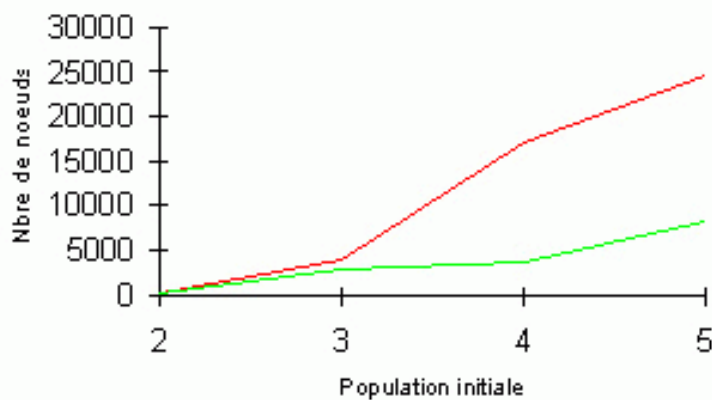


FIG. 6.5 – Evolutions comparatives des tailles du GMA : Le gain dû à la suppression des permutations est significatif



configurations	chasse permanente						alternance de saisons de chasse et de pause (2)		
	pièges maintenus (1a)		alternance des pièges (1b)			nœuds	arcs	nœuds terminaux	
	nœuds	arcs	nœuds terminaux	nœuds	arcs				nœuds terminaux
	389584	412081	2	776396	820866	12	565206	583922	0
	443260	469479	6	884760	936715	0	724103	748391	0

FIG. 6.6 – Statistiques des graphes de marquages accessibles correspondant aux trois scénarios étudiés : le système est viable lorsqu'il n'y a aucun noeud terminal

Par contre, sur les tailles initiales de plus de 5 individus, nous n'avons pas pu générer de GMA jusqu'au bout pour des raisons matérielles, mais nous avons observé lors de la génération de ces graphes qu'on parvenait à plusieurs millions de noeuds sans trouver un seul état terminal, ce qui laisse croire que dans ces conditions, il est fort possible que le GMA ne possède pas d'état terminal avec population nulle, dont que le système est viable.

Cette étude ayant davantage un intérêt qualitatif, nous avons utilisé les paramètres minimaux de manière à pouvoir obtenir des graphes d'accessibilité de taille raisonnable, tout en reflétant sur le plan comportemental le modèle multi-agent et donc la réalité. Nous sommes donc partis d'un espace 2×2 où chaque cellule peut contenir au plus deux animaux et 50% des cellules sont piégées. Avec une population initiale de 4 animaux, nous avons étudié plusieurs scénarios résumés dans le tableau de la figure 6.6.

La non existence de noeuds terminaux correspond à la vivacité du réseau de Petri, c'est toujours le cas lorsque les saisons de chasse et de pause sont alternées (2). Par contre, lorsque la chasse se fait de façon permanente (1a), le graphe comporte toujours des noeuds terminaux, ce qui est équivalent à l'existence dans ces conditions de schémas d'évolution conduisant à l'extinction de la population. Le scénario (1b) dans lequel on maintient les pièges au cours des saisons en changeant leurs positions fait apparaître une différence de résultats selon les configurations spatiales.

6.4 Conclusion

La formalisation de systèmes à agents répond généralement à un des deux objectifs suivants [Singh et al., 1999] : soit comme langage de spécification interne utilisé par les agents dans leurs processus d'action et de raisonnement, soit comme métalangage utilisé par les modélisateurs pour spécifier et vérifier des propriétés comportementales d'agents situés dans un environnement dynamique. Nous situons notre approche dans cette dernière perspective. C'est ainsi que nous discutons ici l'intérêt de cette démarche au regard de la modélisation du système et la vérification des propriétés.

La représentation des comportements spatialisés par les réseaux de Petri colorés né-

cessite la mise en oeuvre de mécanismes pour modéliser certaines dynamiques comme le choix de la direction de déplacement, le comportement du système lorsque la cellule de destination de l'animal est déjà occupée ou la prise en compte des événements aléatoires. Il a aussi fallu maintenir la cohérence dans les places « duales » du système. Du point de vue de la modélisation, ces difficultés liées intrinsèquement aux réseaux de Petri peuvent contribuer à empêcher la focalisation sur le cas d'étude. Cependant, la démarche présente un intérêt indéniable dans la mesure où la sémantique formelle des réseaux de Petri oblige à lever toute ambiguïté lors de la description du système. La modularité de la conception facilite la maintenance et l'effet spaghetti inhérent aux modèles complexes est évité.

La modélisation par les réseaux de Petri ne peut être menée qu'au prix de nombreuses simplifications sur le modèle initial. Mais il est indéniable que la richesse de l'analyse que nous procurent ces méthodes formelles renseigne mieux que les simulations multi-agents, à condition de résoudre au passage les problèmes de complexité dans l'exploration du GMA.

Le véritable challenge méthodologique a été de mener à bien l'étude des propriétés du système, à cause de sa complexité. Nous avons analysé les graphes de marquages accessibles afin de déterminer l'existence ou non d'un état pour lequel la taille de la population est nulle. Le *model checker* nous a permis d'exhiber des résultats qualitatifs très significatifs dans le domaine de la gestion des ressources spatialisées, notamment l'influence de la distribution spatiale et des comportements associés sur la dynamique à long terme de la population. Ces résultats ont pour la plupart confirmé les simulations multi-agents, à la différence que dans ce cas, le modèle nous a renseigné sur l'ensemble exhaustif des évolutions du système dans un temps infini.

Avec l'accroissement des performances des *models checkers*, il est à espérer plus tard d'utiliser pour les paramètres du modèle des valeurs plus proches de la réalité. Nous pourrions aussi envisager d'utiliser de nouvelles techniques d'analyse symbolique, telles que l'approche par les diagrammes de décision [Couvreur et al., 2002].

Conclusion générale

L'objet de cette thèse était d'évaluer les pratiques « villageoises » de chasse dans la forêt avoisinante du village de Djémiong à l'Est-Cameroun. La modélisation multi-agent est utilisée comme une alternative aux modèles mathématiques classiques, afin de prendre en compte les variables qualitatives et les stratégies développées par les chasseurs pour préserver la ressource.

Pour mener à bien ce travail – et pour répondre à un besoin d'outils de modélisation et de simulation de systèmes similaires –, nous avons commencé par mettre sur pied une plate-forme multi-agent. CORMAS est à la fois un outil d'aide à la modélisation des problèmes de gestion de ressources naturelles renouvelables et un environnement de simulation de modèles multi-agents. En intégrant la prise en compte des dynamiques spatiales et des dynamiques sociales, la plate-forme CORMAS répond à un courant méthodologique très actuel en gestion de ressources communes.

En effet, pendant de nombreuses années cette question fut abordée soit sous le seul angle du « système écologique qui subit des perturbations anthropiques » soit sous le seul angle du « système social soumis à des contraintes naturelles » [Bousquet et al., 1996]. La tendance actuelle est de considérer que l'enjeu est au niveau de la compréhension des interactions entre dynamiques écologiques et dynamiques sociales. Notre plate-forme est un bon support pour la mise en oeuvre de telles études.

La modélisation multi-agent de l'activité de chasse à Djémiong a consisté, à partir d'enquêtes de terrain, à construire des agents représentant les céphalophes (espèce la plus chassée), des agents représentant des chasseurs et à simuler divers scénarios de chasse. L'environnement des agents du système est la carte de la forêt avoisinante de Djémiong que nous avons intégrée dans la plate-forme CORMAS après quelques transformations appropriées dans le SIG IDRISI. Les simulations sur cet espace « proche » du réel nous ont permis de montrer l'influence des choix spatiaux et temporels des chasseurs sur la dynamique de la ressource. Nous avons testé des règles d'organisation sociale pour illustrer leur impact sur les stratégies des chasseurs.

Pour valider les résultats des simulations multi-agents, nous avons procédé à une formalisation du modèle par les réseaux de Petri. Les réseaux de Petri colorés ont été choisis pour le fait qu'ils offrent un pouvoir d'expression très riche tout en gardant la possibilité de

vérifier un grand nombre de propriétés. Cette caractéristique est principalement due aux opérations de « dépliage » qui permettent d'associer à tout réseau de Petri coloré un réseau de Petri ordinaire qui lui est équivalent. Nous avons ainsi, à partir d'hypothèses extraites du modèle multi-agent, élaboré un modèle réseaux de Petri colorés. L'analyse structurelle nous a permis de vérifier la consistance du modèle. L'étude du graphe des marquages accessibles nous a renseigné sur l'existence éventuelle des situations de blocage dans notre modèle, suivant les divers scénarios de chasse que nous avons testés. Cependant, la croissance exponentielle de la taille de ce graphe nous a contraint à utiliser de petites valeurs pour les paramètres du modèle, rendant ainsi difficile toute extrapolation des résultats obtenus à des modèles plus réalistes.

C'est un des points sur lesquels nous espérons améliorer notre travail. Plusieurs pistes peuvent être envisagées : l'élaboration de modèles réseaux de Petri structurellement moins complexes et représentant le même phénomène, l'utilisation de modèles checkers plus performants que celui qui a servi pour cette étude ou l'usage d'un autre formalisme de spécification de systèmes offrant de meilleures possibilités d'analyse formelle. Il est concevable après une telle étude d'élaborer une méthode reproductible de formalisation et de vérification des modèles multi-agents.

Les résultats des simulations multi-agents étant validés par cette étude formelle, nous pouvons considérer qu'ils permettent effectivement de montrer que les stratégies de gestion locale de la faune que nous avons identifiées conduisent bien à assurer la pérennité de l'espèce. Cette conclusion appelle cependant d'autres questions : comment amener les pouvoirs publics à intégrer cette approche dans la gestion des ressources naturelles renouvelables ? Comment amener les populations à se réappropriier les ressources qu'elles n'ont plus le droit d'exploiter ?

Bibliographie

- [Abrami, 1999] Abrami, G. (1999). Modélisation multi-agents pour l'aménagement forestier : Usages multiples dans l'écosystème forestier de didy. Technical Report 50, Université Louis Pasteur, Strasbourg.
- [Antona, 2000] Antona, M. (2000). Régulation d'une filière d'exploitation d'une ressource renouvelable : le bois énergie au niger. Technical Report 90, Université Montpellier 1 - ENSAM.
- [Baeijs et al., 1995] Baeijs, C., Demazeau, Y., and Alvares, L. (1995). Application des systèmes multi-agents à la généralisation cartographique. *Actes des troisièmes journées francophones sur l'IAD et les systèmes multi-agents, Hermes, Paris.*
- [Bah, 2000] Bah, A. (2000). Les systèmes multi-agents génétiques : vers une société d'agents génétiques. Technical report, Université Cheikh Anta Diop de Dakar.
- [Bah et al., 1998] Bah, A., Canal, R., d'Aquino, P., and Bousquet, F. (1998). Les sma génétiques pour l'étude de la mobilité pastorale en zone inter-tropicale sèche. *Actes du colloque SMAGET. Clermont-Ferrand.*
- [Bakam and Bousquet, 1997] Bakam, I. and Bousquet, F. (1997). Cormas : Un environnement de simulation de modèles multi-agents pour la gestion de ressources renouvelables. Technical Report 15, United Nations University - Dépt. d'Informatique, Univ. De Yaoundé 1.
- [Bakam and Bousquet, 1998] Bakam, I. and Bousquet, F. (1998). Modèles multi-agents pour la gestion de l'environnement : vers un couplage simulations et approches formelles de modélisation. *Actes du colloque SMAGET. Clermont-Ferrand.*
- [Bakam et al., 2000a] Bakam, I., Kordon, F., Le Page, C., and Bousquet, F. (2000a). Formalisation de modèles multi-agents par les réseaux de petri : Application à la gestion de la chasse à l'est-cameroun. *Actes du CARI'2000 : 5e Colloque Africain sur la recherche en Informatique, Antananarivo, Madagascar.*
- [Bakam et al., 2000b] Bakam, I., Kordon, F., Le Page, C., and Bousquet, F. (2000b). Formalization of a multiagent model using coloured petri nets for the study of an hunting management system. *Lecture Notes in Artificial Intelligence*, 1871 :123–132.

- [Bakam et al., 1997] Bakam, I., Takforyan, A., Proton, H., Weber, J., and Bousquet, F. (1997). Simulations spatio-temporelles d'interactions entre chasseurs et ressources par les systèmes multi-agents : illustration par le cas de la chasse dans lest-cameroun. *Actes du 13ème Congrès de l'Union Internationale des Biologistes du Gibier. Lyon.*
- [Barbault, 1992] Barbault, R. (1992). *Ecologie générale, Structure et fonctionnement de la biosphère.* Masson.
- [Barreteau and Bousquet, 1999] Barreteau, O. and Bousquet, F. (1999). Jeux de rôles et validation des systèmes multi-agents. *Ingénierie des systèmes multi-agents.*
- [Barreteau and Bousquet, 2000] Barreteau, O. and Bousquet, F. (2000). a multi-agent model to tackle viability of irrigated systems. *Annals of Operations Research.*
- [Bauer, 1998] Bauer, H. (1998). *Proceedings of the seminar «People and Parks»*, chapter People and Parks : the stakes of Co-management. CEDC Maroua.
- [Berkes and Farvar, 1999] Berkes, F. and Farvar, M. T. (1999). *Common property resources. Ecology and community-based sustainable development*, chapter Introduction and Overview, pages 1–17. Belhaven press.
- [Bonin and Le Page, 2000] Bonin, M. and Le Page, C. (2000). Sig, sma, connaissances et gestion de l'espace. le cas du massif du tanargue. *Revue de Géomatique.*
- [Boulanger et al., 1995] Boulanger, D., Colloc, J., Dubois, G., and Wintergerst, C. (1995). Objets-agents : Continuum ou différences. *Actes des troisièmes journées francophones sur l'IAD et les systèmes multi-agents.*
- [Bousquet, 1994] Bousquet, F. (1994). *Des milieux, des poissons, des hommes : Etude par simulations multi-agents.* PhD thesis, Université Claude Bernard, Lyon I.
- [Bousquet et al., 1998] Bousquet, F., Bakam, I., Proton, H., and Le Page, C. (1998). Cor-mas : Common-pool resources and multi-agents systems. *Lecture Notes in Artificial Intelligence*, 1416 :826–837.
- [Bousquet et al., 1996] Bousquet, F., Barreteau, O., Mullon, C., and Weber, J. (1996). Modélisation d'accompagnement : Systèmes multi-agents et gestion des ressources renouvelables. *Quel environnement au XXIème siècle ? Environnement, maîtrise du long terme et démocratie. Actes du colloque international, Abbaye de Fontevraud.*
- [Bousquet and D'Aquino, 1999] Bousquet, F. and D'Aquino, P. (1999). Rangeland herd and herder mobility in dry intertropical zones : multi-agent systems and adaptation. *Vith International Rangeland Congress, Townsville, Australy.*
- [Bousquet and Gauthier, 1999] Bousquet, F. and Gauthier, D. (1999). Comparaison de deux approches de modélisation des dynamiques spatiales par simulation multi-agents : les approches spatiales et acteurs. *Revue Européenne de géographie, Cybergeog.*

- [Bousquet and Le Page, 2000] Bousquet, F. and Le Page, C. (2000). Ecological scales and use rights : the use of multiagent systems. forest and society. *the role of research Sub- ple-nary session XXI IUFRO World Congress 2000*. Baskaran Krishnapillay, E. Soepadmo, Najib Lotfy Arshadet al. Kuala Lumpur, pages 730–742.
- [Bousquet et al., 2001] Bousquet, F., Le Page, C., Bakam, I., and Takforyan, A. (2001). Multiagent simulations of hunting wild meat in a village in eastern cameroon. *Ecological Modeling*, 138 :331 – 346.
- [Brooks, 1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intel-ligence*, 47 :139–160.
- [Carles and Dignum, 2001] Carles, S. and Dignum, F. (2001). Agent-mediated electronic commerce : Scientific and technological roadmap. *Lecture Notes in Artificial Intelligence*, 1991 :1–18.
- [Champagnat, 1998] Champagnat, R. (1998). *Supervision des systèmes discontinus : Dé-finition d'un modèle hybride et pilotage en temps réel*. PhD thesis, Université Paul Sabatier, Toulouse.
- [Connell, 1990] Connell, J. H. (1990). *Minimalist Mobile Robotics*. Academic Press, Londres.
- [Cormen et al., 1999] Cormen, T., Leiserson, C., and Rivest, R. (1999). *Introduction à l'Algorithmique*. Dunod.
- [Couvreur et al., 2002] Couvreur, J.-M., Encrenaz, E., Paviot-Adet, E., Poitrenaud, D., and Wacrenier, P.-A. (2002). Data decision diagrams for petri nets analysis. *Actes de Petri Nets 2002*.
- [de Mongolfier and Natali, 1987] de Mongolfier, J. and Natali, J. M. (1987). *Le Patrimoine du futur, Economia*. Economia, Paris.
- [Demazeau and Müller, 1991] Demazeau, Y. and Müller, J. P. (1991). *Decentralized Artifi-cial Intelligence*, chapter From reactive to intentional agents. Elsevier Science Publishers B. V.
- [Drogoul, 1993] Drogoul, A. (1993). *De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents*. PhD thesis, Université Paris VI.
- [Dubost, 1980] Dubost, G. (1980). L'écologie et la vie sociale du céphalophe bleu (cephalophus monticola thunberg), petit ruminant forestier africain. *Journal of Comparative Ethology*.
- [Dubost, 1983a] Dubost, G. (1983a). Le comportement de cephalophus monticola thun-berg et c. dorsalis gray, et la place des céphalophes au sein des ruminants, 1ère partie. *Mammalia*, 2(47) :141–177.

- [Dubost, 1983b] Dubost, G. (1983b). Le comportement de *cephalophus monticola thunberg* et *c. dorsalis gray*, et la place des céphalophes au sein des ruminants, 2e partie. *Mammalia*, 2(47) :281–310.
- [El Fallah, 1994] El Fallah, S. (1994). Représentation et manipulation de plans à l’aide des réseaux de petri. *Actes des 2èmes Journées Francophones IAD-SMA. Voiron.*
- [El Fallah et al., 1997] El Fallah, S. A., Haddad, S., and Taghelit, M. (1997). *Introduction aux systèmes de planification*. Support de cours du DEA 127 de l’Université Paris-Dauphine.
- [Ferber, 1994] Ferber, J. (1994). *Many-agents simulation and Artificial Life*, chapter Simulating with reactive agents. IOS Press.
- [Ferber, 1995] Ferber, J. (1995). *Les systèmes multi-agents. Vers une intelligence collective*. InterEditions.
- [Ferber and Gutknecht, 1998] Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. *Proceedings of ICMAS’98*.
- [Ferrand et al., 1998] Ferrand, N., Demazeau, Y., and Beijs, C. (1998). Systèmes multi-agents réactifs et résolution de problèmes spatialisés. *Revue d’Intelligence Artificielle*, (1) :37–72.
- [Feuillette and Bousquet, 2000] Feuillette, S. and Bousquet, F. (2000). Sinuse, un outil d’interactions distribuées entre une nappe et ses usagers. Technical report, GIRN-ZIT, Bamako.
- [Feuillette and Garin, 2000] Feuillette, S. and Garin, P. (2000). Un modèle d’exploration des dynamiques ressource-usages pour la gestion intégrée d’une nappe surexploitée. application à la nappe de kairouan, tunisie. *Hydrologie des Régions Méditerranéennes, Montpellier.*
- [Gauvin, 1995] Gauvin, D. (1995). Un environnement de programmation orienté agent. *Troisièmes journées francophones sur l’IAD et les systèmes multi-agents.*
- [Green, 1997] Green (1997). Etude économique d’un écosystème forestier tropical, volets forêt et faune. Technical report, Ministère de la coopération.
- [Green, 1999] Green (1999). La coordination des usages des ressources renouvelables dans un écosystème forestier tropical. le rôle des simulations et des systèmes d’information. Technical report, SEAH.
- [Guessoum, 1998] Guessoum, Z. (1998). Dima : Une plate-forme multi-agents en smalltalk. *Revue Objet*, (4) :393–410.
- [Gutknecht et al., 2000] Gutknecht, O., Ferber, J., and Michel, F. (2000). Madkit : une architecture de plate-forme multi-agent générique. Technical Report 0061, LIRMM.
- [Hadadi, 1996] Hadadi, A. (1996). Communication and cooperation in agent systems. *Lecture Notes in Artificial Intelligence.*

- [Hardin, 1968] Hardin, G. (1968). The tragedy of the commons. *Science*, (162) :1243–1248.
- [Hardin and Badin, 1977] Hardin, G. and Badin, J. (1977). Managing the commons. *W. H. Freeman and Company, New York*.
- [Herimandimby et al., 1998] Herimandimby, V., Randriajaona, E., Barreteau, O., Bousquet, F., and Weber, J. (1998). Systèmes multi-agents et démarche patrimoniale : utilisation de jeux de rôle. *Actes du colloque SMAGET. Clermont-Ferrand*.
- [Hideyuki and Itsuki, 1998] Hideyuki, N. and Itsuki, N. (1998). Dynamic subsumption architecture for programming intelligent agents. *Proceedings of ICMAS'98*, pages 190–197.
- [Hiebeler, 1994] Hiebeler, D. (1994). The swarm simulation system and individual-based modeling. *Proceedings of Decision Support 2001. Advanced Technology for Natural Resource Management, Toronto*.
- [Holland, 1995] Holland, J. (1995). *Hidden Order. How Adapation Builds Complexity*. Addison-Wesley.
- [Hraber et al., 1997] Hraber, P., Jones, T., and Forest, S. (1997). The ecology of echo. *Artificial life*, 3 :165–190.
- [Jensen, 1997] Jensen, K. (1997). *Coloured Petri Nets : Basic Concepts, Analysis Methods and Pratical Use*, volume 1. Springer.
- [Jieh Hsiang, 1999] Jieh Hsiang, H.-C. T. (1999). Personalized web retrieval : Three agents for retrieving web information. *PRIMA '98, LNAI*, 1599 :118–132.
- [Kam and Castella,] Kam, S.-P. and Castella, J.-C. Methodological integration : lessons from the ecoregional initiative for the humid and sub-humid tropics of asia. *Conservation Ecology, submitted*.
- [Kordon and Paviot-Adet, 1999] Kordon, F. and Paviot-Adet, E. (1999). Using cpn-ami to validate a soft channel protocol. *The toolset Proceedings of the International Conference on Theory and Applications of Petri Nets, Williamsburg, USA*.
- [Labidi and Lejouad, 1993] Labidi, S. and Lejouad, W. (1993). De l'intelligence artificielle distribuée aux systèmes multi-agents. Technical Report 2004, INRIA.
- [Le Page, 1996] Le Page, C. (1996). *Biologie des populations et simulations individus-centrées*. PhD thesis, Université Paris VI.
- [Le Page et al., 2000] Le Page, C., Bousquet, F., Bakam, I., Bah, A., and Baron, C. (2000). Cormas : A multi-agent simulation toolkit to model natural and social dynamics at multiple scales. *Ecological Modeling*.
- [Le Page et al., 1999] Le Page, C., Lardon, S., Bommel, P., Baron, C., and Bousquet, F. (1999). Entités spatiales génériques et modèles de simulation multi-agent. *JFIAD-SMA '99, La Réunion, France*.

- [Lifran et al., 1998] Lifran, R., Bakam, C. L. I., and Bousquet, F. (1998). Externalités de voisinage, communication entre agents et pareto optimalités. étude théorique et par un sma. *Actes du colloque SMAGET. Clermont-Ferrand*.
- [Magnin, 1996] Magnin, L. (1996). *Modélisation et simulation de l'environnement dans les systèmes multi-agents*. PhD thesis, Université Paris VI, Paris.
- [Magnin and Ferber, 1993] Magnin, L. and Ferber, J. (1993). Conception de systèmes multi-agents par composants modulaires et réseaux de petri. *Actes des Journées du PRC-IA, Montpellier*, pages 129–140.
- [Mathevet, 1998] Mathevet, R. (1998). L'approche multi-agents comme outil de simulation de la gestion des marais de chasse dans les exploitations agricoles camarguaises. *Actes du colloque SMAGET. Clermont-Ferrand*.
- [Mathevet, 2000] Mathevet, R. (2000). *Usages des zones humides camarguaises : enjeux et dynamique des interactions Environnement/Usagers/Territoire*. PhD thesis, Université Jean Moulin, Lyon 3.
- [Merlat and Seyrat, 1997] Merlat, W. and Seyrat, C. (1997). Javanetagents : une plateforme d'exécution d'agents mobiles pour le développement de systèmes multi-agents sur internet. *Actes des 5e journées francophones sur l'Intelligence Artificielle et les Systèmes Multi-Agents*.
- [Monin, 1996] Monin, J. F. (1996). *Comprendre les méthodes formelles, Panorama et outils logiques*. Ed Manon et CNET-ENST. Paris.
- [Murata, 1989] Murata, T. (1989). Petri nets : Properties, analysis and applications. *Proceedings of IEEE*, 77.
- [Naquet and Geniet, 1992] Naquet, G. V. and Geniet, A. C. (1992). *Réseaux de Petri et systèmes parallèles*. Ed Armand Colin.
- [Ngandjui, 1994] Ngandjui, G. (1994). Etude de la chasse en vue de sa gestion durable en collaboration avec les populations résidentes. cas de mekas (ouest de la réserve de faune du dja). Technical report, ECOFAC Cameroun, Yaoundé.
- [Peterson, 1981] Peterson, J. L. (1981). *Petri Nets Theory and the Modeling of Systems*. Englewood Cliffs, Nj. Prentice-Hall Inc.
- [Ramchandani, 1974] Ramchandani, C. (1974). *Analysis of asynchronous concurrent systems by timed Petri nets*. PhD thesis, MIT, Cambridge (Etats-Unis).
- [Sibertin-Blanc, 1985] Sibertin-Blanc, C. (1985). High level petri nets with data structure. *6th European Workshop on Petri Nets and Applications. Espoo, Finlande*.
- [Singh et al., 1999] Singh, M., Rao, A. S., and Georgeff, M. P. (1999). *Multiagent Systems : A Modern Approach To Distributed Artificial Intelligence*, chapter Formal Methods in DAI : Logic Based Representation and Reasoning. MIT Press.

- [Takforyan, 1995] Takforyan, A. (1995). Gestion de la faune sauvage : les nouveaux communs comme alternative aux aires protégées? le cas de la zone forestière de l'est-cameroun. *5th common property conference : reinventing the commons, Bodo, Norvège*.
- [Takforyan, 1996] Takforyan, A. (1996). Vers une gestion locale de la faune sauvage en afrique? le cas de l'est-cameroun. *Colloque Gestion communautaire des ressources naturelles renouvelables et développement durable, Harare, Zimbabwe*.
- [Takforyan, 2001] Takforyan, A. (2001). *Chasse villageoise et gestion locale de la faune sauvage en Afrique Une étude de cas dans une forêt de l'Est-Cameroun*. PhD thesis, EHESS, Paris.
- [Vernadat et al., 1994] Vernadat, F., Lanusse, A., and Azéma, P. (1994). Modélisation par réseaux de petri d'un langage acteur : Application à la vérification de systèmes multi-agents. *Actes des 2èmes Journées Francophones IAD-SMA. Voiron*.
- [Volterra, 1926] Volterra, V. (1926). Variations and fluctuations of the number of animal speices living together. *Animal Ecology*.
- [Weber, 1992] Weber, J. (1992). L'occupation humaine des aires protégées à madagascar : diagnostic et éléments pour une gestion viable. *Natures-Sciences- Sociétés*, 2(3) :157–164.
- [Weber, 1996] Weber, J. (1996). Conservation, développement et coordination : Peut-on gérer biologiquement le social? *Colloque Gestion communautaire des ressources naturelles renouvelables et développement durable, Harare, Zimbabwe*.
- [Weber, 1997] Weber, J. (1997). Economie des ressources renouvelables. *Notes de cours, CIRAD Montpellier*.
- [Wooldridge, 1999] Wooldridge, M. (1999). *Multiagent Systems : A Modern Approach To Distributed Artificial Intelligence*, chapter Intelligent Agents. MIT Press.
- [Zunga et al., 1998] Zunga, Q., Vagnini, A., Le Page, C., Touré, I., Lieurain, E., and Bousquet, F. (1998). Coupler systèmes d'information géographique et systèmes multi-agents pour modéliser les dynamiques de transformation des paysages. le cas des dynamiques foncières de la moyenne vallée du zambèze (zimbabwe). *Actes du colloque SMAGET. Clermont-Ferrand*.