

Multi-modélisation & Simulation de Systèmes Complexes : de la théorie à l'application

MÉMOIRE

présenté et soutenu publiquement le 15 Octobre 2012

pour l'obtention de l'

**Habilitation à Diriger les Recherches
de l'Université du Littoral Côte d'Opale
(Spécialité Informatique)**

par

Jean-Christophe Soulié

Composition du jury

Président : Éric Ramat – Professeur – Université du Littoral Côte d'Opale

Rapporteurs : David Hill – Professeur – Université Blaise Pascal, LIMOS
Philippe Mathieu – Professeur – Université Lille 1, LIFL
Jean-Pierre Muller – Cadre de Recherche, HDR – CIRAD, UPR Green

Examineurs : Michael Chelle – Directeur de Recherche – INRA, UMR EGC
Michael Dingkuhn – Cadre de Recherche, HDR – CIRAD, IRRI, UMR AGAP

Mis en page avec la classe thloria.

*La théorie, c'est quand on sait tout et que rien ne fonctionne.
La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi.
(A. Einstein)*

Table des matières

Introduction & Positionnement	1
--	----------

1 Approche Mono-paradigme
--

1.1	Introduction	7
1.2	Systèmes multi-agents	8
1.2.1	Contexte	8
1.2.2	Travaux initiaux	8
1.2.3	Problématique	11
1.2.4	Approche à environnements multiples dans DEVS	11
1.3	Couplage d'équations aux différences	15
1.3.1	Présentation du problème	15
1.3.2	TREQ : TRanslator EQuation	17
1.4	Conclusion	25

2 Approche multi-paradigmes
--

2.1	Introduction	27
2.2	Couplage de modèles discrets et d'équations différentielles	27
2.2.1	Résolution d'équations différentielles spatialisées	28
2.3	Les méthodes d'intégration QSS1 et QSS2	28
2.3.1	Présentation de QSS	29
2.3.2	Mise en œuvre et validation	31
2.4	Application aux équations différentielles spatialisées	35
2.4.1	Application aux équations aux dérivées partielles	36
2.4.2	Test de la méthode	39
2.4.3	Gestion des perturbations	41
2.5	Conclusion	42

3**Applications**

3.1	Introduction	46
3.2	Un modèle de dynamique spatiale : MUFINS (MULTI Fish INdian ocean Simulator)	46
3.2.1	Le cahier des charges du modèle	46
3.2.2	Gestion des données	47
3.2.3	Le système conatif, la mise en œuvre des comportements	47
3.2.4	Quelques simulations avec MUFINS	53
3.3	Un modèle exploratoire : une pêcherie théorique à métiers multiples – Modèle DYNFISH 1	58
3.3.1	Problématique	58
3.3.2	D’un modèle théorique	58
3.3.3	... À une modélisation multi-agents	61
3.3.4	... À la simulation	63
3.3.5	Application à la pêcherie d’Algue Française – Modèle SEAWEED	73
3.4	Une extension du modèle : DYNFISH 2	77
3.4.1	Présentation du modèle	77
3.4.2	Implémentation dans VLE	79
3.4.3	Les résultats de la simulation	80
3.5	Le modèle CHALOUPE	80
3.5.1	Présentation du projet Chaloupe (Projet Biodiversité ANR-IFB 2005) et objectifs	80
3.5.2	Positionnement de DYNFISH 3 dans CHALOUPE	84
3.5.3	Présentation du modèle bio-économique DynFish 3	85
3.6	Le modèle ECOMERISTEM	93
3.6.1	Présentation du modèle	93
3.6.2	Exemple : impact du stress hydrique sur la taille des organes	94
3.7	Conclusion	96

4**Conclusion & Perspectives****A****Curriculum Vitæ détaillé****B****Liste de Publications**

B.1	Articles dans des revues internationales ou nationales avec comité de lecture – ACL	115
B.2	Ouvrages scientifiques – OS	115
B.3	Conférences invitées dans un congrès national ou international – INV	116
B.4	Communications avec actes dans un congrès International – ACTI	116
B.5	Communications avec actes dans un congrès National – ACTN	120
B.6	Autres productions – AP	120
Bibliographie		123

Table des figures

1.1	Le modèle d'agent mono-environnemental	9
1.2	Le modèle d'agent multi-environnemental	12
1.3	Schéma simplifié du graphe d'état du modèle atomique	22
2.1	Quantification avec hystérésis d'une fonction, tiré de [Kofman et Junco, 2001]	30
2.2	Modèle DEVS de QSS1, tiré de [Kofman et Junco, 2001]	30
2.3	Modèle DEVS de QSS2, tiré de [Kofman, 2002], F.O.Q. est pour « <i>First Order Quantizer</i> »	30
2.4	Quantification d'ordre 1, tiré de [Kofman, 2002]	31
2.5	Sortie de QSS1 pour le volterra_test	33
2.6	Sortie de QSS2 pour le volterra_test	33
2.7	Erreur sur le cos_test dans QSS2	34
2.8	Sortie du RLC_test dans QSS1 comparée avec $y(t)$	34
2.9	Le modèle Cell-DEVS	37
2.10	Gestion des perturbations sous QSS1	41
2.11	Gestion d'une fonction g externe au système. Q est le filtre qui quantifie la fonction g , pour prévenir la case x qu'en cas de changement de palier	42
3.1	Les niveaux d'appartenance des termes flous associés à la vitesse (abscisses vitesse en km.h^{-1} , ordonnées niveau d'appartenance)	50
3.2	Les niveaux d'appartenance des valeurs floues associées aux changements de direction (abscisses variation de direction en degrés, ordonnées : niveau d'appartenance)	51
3.3	Schéma de la sélection des animats « capturables » par les lignes de pêche	52
3.4	Schéma du processus de capture statique	53
3.5	Schéma du processus de capture dynamique	54
3.6	Trajectoires de l'animat ayant atteint la latitude la plus australe (a) et trajectoires « extrêmes » (b) pour la même simulation	55
3.7	Évolution de la distribution des distances à la position d'origine des animats en cours de simulation (abscisses : pas de temps, ordonnées : km)	56
3.8	Distribution des animats aux pas de temps 0 (positions initiales régulièrement distribuées), 186 (positions intermédiaires) et 372 (positions finales) ; on a aussi représenté (en gras) les trajectoires « extrêmes »	57
3.9	Arbre de décision du domaine des possibles à chaque pas de temps	61
3.10	Évolution de la rente pour différent niveaux d'effort données	64
3.11	Dynamique de la réponse à une fermeture de zone	64
3.12	Coûts par zone et globaux de la fermeture de zone	65

3.13 Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort bas	66
3.14 Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort à l'optimum	66
3.15 Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort à l'optimum	67
3.16 Évolution des captures totales	69
3.17 Évolution, par espèces, de l'effort et des captures	70
3.18 Répartition de l'effort final en fonction de l'effort initial et du prix	70
3.19 Évolution de la biomasse totale en fonction de l'effort et des différentiels de prix .	71
3.20 Évolution de la biomasse en fonction des prix et de l'effort	72
3.21 Évolution de la rente en fonction des prix et de l'effort	72
3.22 Représentation de l'environnement dans Cormas	73
3.23 Captures totales	76
3.24 Captures individuelles	77
3.25 Les carrés bleus sont les bateaux, le grand carré noir en haut à gauche est le port et enfin la couleur des cases la concentration en poissons. On peut constater une concentration de poisson en bas à droite, résultant de l'interdiction de pêche dans ce secteur pour une période de la simulation	81
3.26 Description UML d'une pêcherie « générique »	86
3.27 Évolution de la biomasse pêchée de langoustines	91
3.28 Évolution du RTBS pour les chalutiers 12-16 du nord	92
3.29 Évolution de la biomasse restant en mer de langoustines	92
3.30 Illustration du fonctionnement d'Ecomeristem	94
3.31 Adaptation d'Ecomeristem au stress hydrique – Issu de [Luquet <i>et al.</i> , 2006] . . .	95
3.32 Comparaison Stress/Sans Stress hydrique pour une parcelle	96
3.33 Comparaison Stress/Sans Stress hydrique pour deux vitesses foliaires	97
4.1 Un modèle atomique	101
4.2 Automate de K. Lorenz	102
4.3 L'ensemble de modèles	103
4.4 Schéma d'intégration des modèles de l'UPR 74	105

Introduction & Positionnement

Introduction

Ce document constitue le mémoire de synthèse d'habilitation à diriger des recherches, il vise à répondre aux trois objectifs requis pour l'obtention de ce diplôme et est caractérisé par :

- la présentation d'une démarche originale dans un domaine scientifique ;
- la maîtrise d'une stratégie autonome de recherche scientifique ;
- la capacité d'encadrement de jeunes chercheurs.

Pour ce faire, ce document s'appuie sur une organisation en 4 chapitres et 2 annexes.

Le premier chapitre présente l'approche mono-paradigme que j'ai développée dans le cadre de mes travaux de recherche et de l'encadrement doctoral de G. Quesnel [Quesnel, 2006]. Ce chapitre est composé de deux sections principales : la première section présente une extension de mon travail de thèse [Soulié, 2001] en utilisant le formalisme DEVS (Discrete Event Specification)[Zeigler, 1976] qui nous assure de pouvoir disposer d'un cadre de travail validé par la théorie de la modélisation et de la simulation. Ce travail consiste, essentiellement, en une formalisation des concepts définis dans mes travaux de thèse. La deuxième section présente une extension du formalisme DEVS classique qui permet de gérer les équations aux différences (des équations de la forme $y = F(t)$ où t représente le pas de temps et F peut être elle même un ensemble de fonctions définies par le temps, voir 1.3 pour une présentation plus complète). Ce travail a été réalisé dans le cadre de mon encadrement doctoral de D. Versmisse [Versmisse, 2008]. L'extension permettant d'utiliser des équations aux différences est une extension extrêmement importante et était très attendues. En effet, ce formalisme est très souvent utilisé dans les modèles naturels qui représentent la grande partie de nos applications (par exemple : [Versmisse *et al.*, 2007], [Dingkuhn *et al.*, 2006] ou bien encore [Jeuffroy et Recous, 1999], ...).

Le deuxième chapitre présente une approche multi paradigmes. Là encore ces travaux ont été réalisés lors de l'encadrement doctoral de D. Versmisse. Cette partie montre comment, grâce à DEVS et à l'environnement VLE [Quesnel *et al.*, 2008], on peut coupler, au delà des méthodes d'intégration numériques classiques (par exemple la méthode d'Euler ou de Runge-Kutta [Gear, 1971]) des méthodes plus sophistiquées et, surtout, parfaitement adaptées à DEVS : les méthodes d'intégration QSS1 [Kofman et Junco, 2001] et QSS2 [Kofman, 2002] de E. Kofman. Dans ce chapitre, également, nous montrons comment ces méthodes sont utilisées dans le cadre d'équations différentielles spatialisées. Ces techniques seront mises en œuvre dans l'application DYNFISH 2 (voir 3.4).

Le troisième chapitre présente des applications qui ont été réalisées dans le cadre des travaux plus formels présentés dans les 2 chapitres précédents. La première de ces applications est une

mise en œuvre directe de l'approche à environnements multiples décrite dans ma thèse. Le but de ce projet était de simuler des trajectométries de bancs d'espadons dans l'Océan Indien. La deuxième application est une application de l'approche multi-agents plus « classique ». Ce travail rentrait dans le cadre de mon stage post-doctoral au sein du Département d'Économie Maritime de l'IFREMER¹. Ces travaux visaient à représenter la dynamique des pêcheries et surtout les phases transitoires lorsqu'une perturbation se déclarait selon une approche bio-économique. Ces perturbations pouvaient être, par exemple : création d'une aire marine protégée, pollution, mise en place de mesure de restriction (par exemple mise en place de quotas sur une espèce donnée ou une pêcherie donnée). La troisième application présente une extension du modèle précédent, mais en utilisant DEVS et les approches développées dans le chapitre 2. Ce travail n'entre pas dans le cadre d'un projet particulier, mais permettait de valider des hypothèses. La quatrième application consiste en une mise en œuvre des concepts présentés dans le chapitre 1 et plus particulièrement la section 1.3 concernant les équations aux différences. Ce travail entre dans le cadre du projet ANR Chaloupe piloté par l'IFREMER. Enfin, la dernière application présente le modèle EcoMeristem. Ce modèle est un des modèles sur lequel je travaille depuis mon arrivée au CIRAD. EcoMeristem est un modèle de plante selon l'approche « structure-fonction » : nous représentons explicitement la structure de la plante (feuille, bourgeon, axe, entrenœud, ...) à laquelle nous attachons des processus. Chaque partie de la plante (structure) joue un rôle au sein du système global.

Le quatrième chapitre, classiquement, consiste en une conclusion des travaux réalisés lors de ces 11 dernières années depuis la fin de ma thèse. Enfin, les perspectives et pistes de recherches sont présentées.

La première annexe est un Curriculum Vitæ détaillé présentant de manière synthétique, mais exhaustive, les aspects propres à mon activité dans le domaine de la recherche scientifique. Cette section présente l'ensemble de mes activités en qualité de membre de conseils, comités scientifiques ou groupes de travail ou bien mon implication au niveau administratif. Le lecteur trouvera également dans cette section l'ensemble des projets et programmes de recherche dans lesquels je me suis impliqué pour mener à bien mon activité scientifique. Enfin, les aspects relatifs à l'animation de manifestations à caractère scientifique sont présentés. Par son niveau de synthèse cette section offre un angle de vue global sur mon activité et constitue un premier niveau de réponse aux trois objectifs fixés par ce document.

La deuxième annexe présente la liste exhaustive de mes publications à ce jour.

Positionnement

Les travaux présentés dans ce document ont été réalisés dans le cadre de mes différentes affectations et localisations, à savoir, chronologiquement : l'équipe Systèmes Multi-Agents de l'Institut de REcherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, le Département d'Économie Maritime (DEM) de l'IFREMER, le Laboratoire d'Informatique Signal et Image (LISIC) de la Côte d'Opale de l'Université du Littoral Côte d'Opale et, maintenant, l'Unité Mixte de Recherche Amélioration Génétique et Amélioration des Plantes (AGAP) du CIRAD.

Au delà de cette diversité géographique et institutionnelle, les travaux réalisés se situent dans

1. <http://www.ifremer.fr/drvsem>

un continuum bien précis : celui de la modélisation et de la simulation des systèmes complexes et plus particulièrement les systèmes complexes naturels.

Si aujourd'hui le terme de système complexe est très à la mode et répandu (pour s'en convaincre, il suffit de faire une recherche sur Google et on obtient en 2012, environ 99 200 000 réponses ; avec une augmentation d'environ 8 000 000 par an depuis 5 ans), il est souvent employé mal à propos et on peut constater une tendance forte à confondre compliqué et complexe. De nombreux systèmes se prétendent complexes alors qu'ils sont souvent seulement compliqués car ils ne remplissent pas un certain nombre de critères définissant un système complexe. Si de nombreuses définitions pour les systèmes complexes existent, on peut synthétiser ces dernières et proposer la liste suivante comme étant des critères essentiels pour des systèmes complexes [The Center for Study of Complex System, 2009] :

On considère un phénomène dans les sciences sociales, de la vie, physiques et de la décision un système complexe s'il possède un nombre significatifs des caractéristiques suivantes :

- Si les briques de base du système étudié sont les caractéristiques et les activités de processus individuels ;
- Ces processus sont différents en terme de caractéristiques et sont hétérogènes ;
- Les caractéristiques peuvent changer dans le temps. La dynamique qui décrit comment le système change est classiquement non linéaire et parfois chaotique. Le système est rarement dans une situation d'équilibre ;
- Les boucles de rétro-actions viennent modifier, perturber les caractéristiques des processus, augmentant encore la non linéarité et amenant la non-dérivabilité du système.

La science des systèmes complexes est, par définition, une science interdisciplinaire où les mathématiques (équations différentielles, équations aux différences, automates cellulaires, théorie des jeux, processus Markovien, algorithmes génétiques, théorie des graphes, analyse de séries temporelles, etc.) et la simulation informatique (dans la plupart des cas, les simulations proviennent de modèles mathématiques) sont des sciences à l'interface des autres.

Une grande partie de mes travaux a été réalisée dans le cadre du projet VLE² initié par E. Ramat [Ramat et Preux, 2003] et finalisé par Q. Quesnel [Quesnel *et al.*, 2008], [Quesnel, 2006] dans le cadre de son travail doctoral. Le projet VLE est une implémentation du formalisme DEVS ; il propose ainsi un simulateur et un ensemble d'outils qui permettent de simuler et coupler des modèles hétérogènes.

Pour ce faire, VLE s'appuie, bien évidemment, sur le formalisme DEVS et plus particulièrement sur une extension de DEVS : PDEVS (Parallel DEVS) [Chow et Zeigler, 1994]. Dans le formalisme PDEVS, on définit un modèle atomique par le n-uplet suivant :

$$DEVS = \langle X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$$

Où :

- X est l'ensemble des ports et des valeurs d'entrée ;
- Y est l'ensemble des ports et des valeurs de sortie ;
- S est l'ensemble des états du système ;
- $ta : S \rightarrow \mathbb{R}_0^+$ est la fonction d'avancement du temps ;

2. <http://www.vle-project.org>

- $Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$, avec :
 - Q : l'ensemble des états totaux ;
 - e : la durée écoulée depuis la dernière transition.
- $\delta_{int} : S \rightarrow S$ est la fonction de transition interne ;
- $\delta_{ext} : Q \times X^b \rightarrow S$ est la fonction de transition externe où X^b est un « bag » d'éléments pris dans X ;
- $\delta_{con} : S \times X^b \rightarrow S$ est la fonction de conflit, sachant que $\delta_{con}(S, \emptyset) = \delta_{int}(s)$;
- $\lambda : S \rightarrow Y$ la fonction de sortie.

Et un modèle couplé est défini par le n-uplet suivant :

$$N = \langle X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC \rangle$$

- Les définitions de X et Y sont identiques à celle d'un modèle atomique ;
- D est l'ensemble des identifiants des modèles intervenant dans le modèle couplé. M_d est un modèle PDEVS appartenant au réseau ;
- EIC est la liste de ports d'entrée du modèle couplé ;
- EOC est la liste de ports de sortie du modèle couplé ;
- IC est la liste des connexions internes au modèle couplé.

Pour finir, grâce à la propriété de fermeture sous couplage, un modèle couplé possède les mêmes propriétés et peut être vu comme un modèle atomique. Cette propriété permet construire des modèles hiérarchiques.

Les définitions *stricto sensu* d'un modèle atomique et d'un modèle couplé PDEVS sont la base même de tous les modèles. Néanmoins, ces définitions restent rustiques et pauvres dès que l'on commence à vouloir représenter des systèmes plus compliqués. C'est pourquoi des extensions à DEVS ont été développées. Un grand nombre sont disponibles aujourd'hui, on peut citer : Cell-DEVS [Wainer et Giambiasi, 2001], [Zeigler *et al.*, 2000] pour un automate cellulaire, DS-DEVS [Barros, 1995] qui permet de pallier l'impossibilité à changer dynamiquement de structure (par exemple, instancier des modèles dynamiquement, ce qui est un inconvénient rédhibitoire dans les systèmes dynamiques). Le lecteur intéressé pourra trouver une description complète et précise de DEVS, PDEVS dans [Quesnel, 2006].

L'ensemble des travaux théoriques que j'ai menés et encadrés consiste donc à ajouter de nouvelles briques à la plate-forme VLE afin d'étendre ses fonctionnalités. Nous appelons ces briques des extensions. Pourquoi des extensions ? Premièrement, l'utilisation de PDEVS implique de penser les modèles (principalement des équations différentielles ou des équations aux différences dans notre cas) avec la composante événementielle et leur reformulation dans cette approche n'est pas toujours triviale. Deuxièmement, PDEVS ne propose pas d'algorithmique puissante qui vienne en aide aux modélisateurs. Troisièmement, les modèles développés à l'heure actuelle sont de plus en plus compliqués et complexes. Tout ceci fait que l'on doit pouvoir proposer des outils de haut niveau et les plus simples possibles permettant de mettre en œuvre des modèles PDEVS tout en masquant leur complexité au maximum. C'est donc le rôle des extensions développées au sein de VLE.

Ces extensions sont vues de deux manières : les approches mono-paradigmes qui n'utilisent qu'une seule sorte de formulation (équations aux différences ou système multi-agents) et les approches multi-paradigmes qui combinent plusieurs formulations (intégration numérique, perturbation, temps discret, temps continu, etc.) – approches rendues possibles par PDEVS et

VLE.

Chapitre 1

Approche Mono-paradigme

Sommaire

1.1	Introduction	7
1.2	Systèmes multi-agents	8
1.2.1	Contexte	8
1.2.2	Travaux initiaux	8
1.2.3	Problématique	11
1.2.4	Approche à environnements multiples dans DEVS	11
1.3	Couplage d'équations aux différences	15
1.3.1	Présentation du problème	15
1.3.2	TREQ : TRanslator EQUation	17
1.4	Conclusion	25

1.1 Introduction

Bien que PDEVS et VLE permettent de, et soient fait pour, coupler des modèles qui sont hétérogènes, il faut, pour pouvoir réaliser cette hétérogénéité, réaliser les briques de base c'est à dire construire des extensions qui ne reposent sur un seul paradigme.

Dans ce chapitre, nous allons présenter deux contributions mono-paradigmes qui ont été réalisées pendant les thèses de G. Quesnel [Quesnel, 2006] pour la partie multi-agents et de D. Versmisse [Versmisse *et al.*, 2007]. La première section présente la contribution relative à la formalisation sous DEVS de l'approche à environnement multiples et la deuxième section présente la contribution relative à l'ajout dans VLE du formalisme basé sur les équations aux différences.

1.2 Systèmes multi-agents

1.2.1 Contexte

Cette section est relative à l'approche Système Multi-Agents [Ferber, 1995] et plus particulièrement à la modélisation et à la simulation orientée agents (la littérature à ce sujet est extrêmement fournie, mais on peut citer, par exemple : [Weiss, 1999], [Huhngs et Singh, 1998], [Wooldridge *et al.*, 2000], [Drogoul *et al.*, 2008], [Bousquet et Le Page, 2004] ou bien encore, et pour finir [Uhrmacher et Kullick, 2000]) que j'ai développée dans ma thèse [Soulié, 2001] et [Soulié et Marcenac, 2000]. Ce travail a été initié en 1998 dans le cadre du développement de la plate-forme GEAMAS ([Marcenac et Giroux, 1998], [Soulié *et al.*, 1998]) à l'Université de la Réunion.

1.2.2 Travaux initiaux

Dans ce travail doctoral je me suis concentré, essentiellement, sur la définition et la modélisation de l'environnement dans les systèmes multi-agents. Entité qui restait, à l'époque le parent pauvre de cette approche [Ferber, 1995].

A ce titre, j'ai proposé une classification des environnements dans les systèmes multi-agents :

- L'environnement centralisé est une structure unique qui définit l'ensemble des éléments de l'environnement. Les agents y accèdent par l'intermédiaire de question. L'environnement réagit à cette demande par une réponse relative à la question. Le principal défaut de cette technique provient du fait qu'elle s'adapte mal à une augmentation de sa complexité, par exemple, lorsque le nombre d'agents ou d'interactions devient important ;
- L'environnement distribué est défini par un ensemble de cellules formant une grille ou une construction plus complexe. Chaque cellule peut être vue comme un environnement centralisé sur lequel les agents peuvent dialoguer et se déplacer de cellule en cellule. Le voisinage entre les cellules est alors la partie la plus importante pour l'action des agents ;
- La définition d'environnement sous la forme d'agent a été initiée en 1996 sur la plate-forme SWARM [Minar *et al.*, 1996] par C. Langton et en 1998 par la plate-forme Madkit [Ferber et Gutknecht, 1998]. Tout comme son nom le définit, les environnements sont ici décrits par des agents. Tous les objets de l'environnement sont alors des agents.

Découpage de l'agent en plusieurs entités

Tout d'abord, en s'appuyant sur la définition de J. Ferber, il a été nécessaire d'isoler la composante « raisonnement » de la composante « environnementale » de l'agent afin de pouvoir isoler les types de messages échangés. Dans notre modèle d'agent, celui-ci est décomposé en trois entités principales : le **système conatif**, l'**instance dans l'environnement** et le **lien bidirectionnel de dépendance**.

- Le système conatif est en quelque sorte « l'âme » de l'agent. C'est lui qui s'occupe (1) de collecter les perceptions de l'agent, (2) de les intégrer dans son processus de raisonnement, de prendre des décisions en fonction des perceptions, des tendances propres à

- l'agent et de son état courant et (3) de demander à ce que les commandes résultant de sa prise de décision soient exécutées sur l'environnement où évolue l'agent. Enfin le système conatif contient toutes caractéristiques nécessaires à la définition d'un agent (autonomie, indépendance, ...);
- L'instance dans l'environnement peut être vue comme « le corps » de l'agent. Cette instance est directement plongée dans l'environnement. Elle est équipée de capteurs qui lui permettent d'obtenir des informations provenant de l'environnement. De manière analogue, l'instance dans l'environnement est équipée d'effecteurs qui vont lui permettre d'exécuter les commandes décidées par le système conatif sur l'environnement;
 - Le lien bidirectionnel de dépendance, si on continue à faire une analogie avec l'être humain, est, en quelque sorte, la colonne vertébrale de l'agent. C'est le long de ce lien que vont transiter les informations entre le système conatif et son instance dans l'environnement et inversement.

La Figure 1.1 illustre cette décomposition.

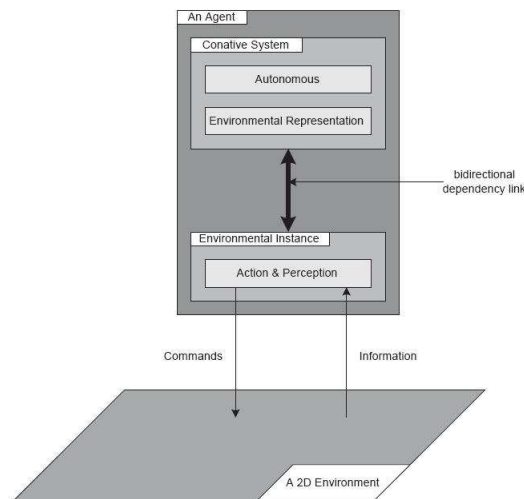


FIGURE 1.1 – Le modèle d'agent mono-environnemental

Passage aux environnements multiples

La motivation principale de ces travaux est issue d'une constatation simple : on peut très facilement avoir besoin d'utiliser **plusieurs environnements simultanément** pour créer une simulation multi-agents. Pour s'en convaincre, il suffit de prendre pour exemple le cas où un agent évolue dans un environnement réel quelconque et il possède aussi un réseau d'accointances (représentant l'environnement social de l'agent). Dans ce cas, il y a bien deux environnements et l'agent évolue en même temps dans ces deux environnements. Or, à la lecture de la littérature liée à ce thème, il apparaît que la plupart des plates-formes multi-agents actuelles ne permettent pas, ou tout du moins pas simplement, de prendre en compte ce cas de figure. C'est pourquoi

cette partie de mes travaux consiste à mettre en place un modèle qui permette de gérer ces cas de figure.

Afin de construire notre modèle à environnements multiples, je me suis appuyé sur le modèle construit précédemment. Le passage d'un environnement simple à des environnements multiples s'est fait en trois étapes : le **découpage des environnements**, le **maintien de l'intégrité des données sur les environnements** et la **gestion du temps**.

- Le découpage des environnements est l'opération la plus délicate. Elle consiste à trouver les informations qui peuvent être mises en commun dans un seul environnement. Le moyen le plus simple est de regrouper sur un seul environnement toutes les données d'un seul et unique type. Ce découpage implique qu'il soit créé autant d'instances dans l'environnement que d'environnements. Chaque instance dans l'environnement possède donc le même type que l'environnement auquel elle est rattachée. De la même manière, comme il a été créé autant d'instances dans l'environnement que d'environnements, il faut donc créer autant de liens bidirectionnels que d'instances dans l'environnement afin que celles-ci soient reliées au système conatif de l'agent. Les types attachés à ces liens sont les mêmes que ceux de l'environnement. De cette manière, on réalise bien une décomposition de l'environnement en plusieurs environnements qui ne vont posséder des informations relatives à un seul type. Enfin, les capteurs et effecteurs des instances environnementales sont eux aussi dédiés aux types de l'instance dans l'environnement qui les contient.
- Comme les données ont été découpées dans des environnements séparés, on peut se retrouver dans la cas de figure suivant : le système conatif prend une décision pour un environnement donné qui est en contradiction avec une autre décision qui a été prise pour un autre environnement. Pour résoudre ce problème, nous avons créé quatre nouvelles entités dans notre modèle : **l'environnement virtuel**, **l'instance virtuelle dans l'environnement**, **le lien bidirectionnel virtuel de dépendance** et **le lien d'intégrité**.
 - **L'environnement virtuel** est une entité qui n'est pas rattachée à des données réelles, d'où son nom. Cette entité va contenir tous les effecteurs et les capteurs qui vont être présents dans au moins deux environnements. Par extension, il va donc contenir tous les effecteurs et tous les capteurs qui sont présents dans tous les environnements. De cette manière, on réduit le nombre d'effecteurs et de capteurs présents dans le système et on peut donc éviter les conflits au niveau des commandes prises par le système conatif.
 - **L'instance virtuelle dans l'environnement** est l'entité qui va être plongée dans l'environnement virtuel. Elle va donc être la représentation physique du système conatif dans l'environnement virtuel.
 - **Le lien bidirectionnel virtuel de dépendance** est le lien qui va relier l'instance virtuelle dans l'environnement et le système conatif de l'agent. C'est par ce lien que vont transiter les perceptions et les commandes échangées entre l'environnement virtuel et le système conatif.
 - **Le lien d'intégrité** est un lien qui va relier l'environnement virtuel et les autres environnements. Ce lien va permettre d'échanger des données entre les environnements. Dans le cas où la modification d'une valeur sur un environnement entraîne la modification d'une autre valeur sur un autre environnement c'est par ce lien que l'information va transiter. Dans ce cas, l'environnement concerné envoie un message spécial à l'environnement virtuel (ce message spécial va notamment contenir à qui il est destiné) et celui-ci va automatiquement rediriger l'information sur l'environnement (ou les environnements) concerné(s). Enfin, dans le cas où la modification d'une valeur dans l'environnement virtuel entraîne des modifications dans tous les autres environnements, c'est

- toujours par ce lien que l'environnement virtuel va prévenir les autres environnements.
- La gestion du temps représente un aspect fondamental des systèmes multi-agents. Dans notre approche, on peut s'apercevoir que tous les environnements sont découpés et autonomes. Se pose donc la question dont on va pouvoir gérer le temps sur l'ensemble des ces environnements. De plus, il se peut que ce temps ne soit pas le même sur l'ensemble des environnements que l'on utilise dans le modèle. Dans notre modèle nous avons choisi de développer une approche qui permette de prendre en compte ces deux cas de figure. Tout d'abord, chaque environnement réel est équipé d'un gestionnaire de temps. Celui-ci va donc donner le pas de temps sur l'environnement réel considéré. De ce fait, chaque environnement réel va pouvoir posséder son propre temps et le gérer de manière autonome. Enfin, l'environnement virtuel va lui aussi posséder un gestionnaire de temps. Ce gestionnaire de temps va, comme pour les autres environnements réels, être chargé de gérer le temps sur l'environnement virtuel. Mais, de plus, c'est lui qui va être chargé de gérer les synchronisations entre les différents environnements réels. Pour ce faire, lors de l'initialisation du système, chaque environnement réel va aller s'enregistrer auprès de l'environnement virtuel en lui précisant quel est son pas de temps (cette opération est rendu possible par les liens d'intégrités qui relient les environnements réels et l'environnement virtuel). Lorsque tous les environnements réels se sont enregistrés auprès de l'environnement virtuel, celui-ci va déterminer quel va être le point de synchronisation pour chaque environnement réel et il va renvoyer ce résultat à tous les environnements réels. Lorsque la simulation va se dérouler, chaque environnement réel va faire dérouler le temps de manière autonome et lorsque son point de synchronisation va être atteint, le gestionnaire de temps de l'environnement concerné va mettre en sommeil l'activité de celui-ci. Cet environnement sera donc arrêté jusqu'à ce qu'il reçoive un message provenant de l'environnement virtuel le prévenant qu'il peut reprendre son activité. De cette manière, on est assuré que tous les environnements réels vont être synchronisés et qu'il n'y aura pas de problème de temps sur l'ensemble des environnements.

La Figure 1.2 présente le passe aux environnements multiples.

1.2.3 Problématique

Or, cette approche, bien que validée et opérationnelle (voir [Soulié *et al.*, 2001] et la section 3.2), souffrait d'une grosse lacune : c'est qu'elle ne reposait sur aucun modèle formel comme, par exemple DEVS ou PDEVS.

Or, lors de mon arrivée au sein du Laboratoire d'Informatique de l'Université du Littoral, R. Duboz dans [Duboz, 2004] finissait ses travaux de doctorat où il avait formalisé l'agent en utilisant le paradigme DEVS (voir [Duboz *et al.*, 2002] et [Duboz *et al.*, 2004]). Dans ce continuum, G. Quesnel dans son travail de thèse [Quesnel, 2006] a repris les éléments de l'approche à environnement multiple et les a formalisé en utilisant DEVS.

1.2.4 Approche à environnements multiples dans DEVS

Nous définissons ici le modèle environnement comme une structure simple de type questions-réponses dont le graphe d'état est très simple afin de permettre la construction facile de modèles

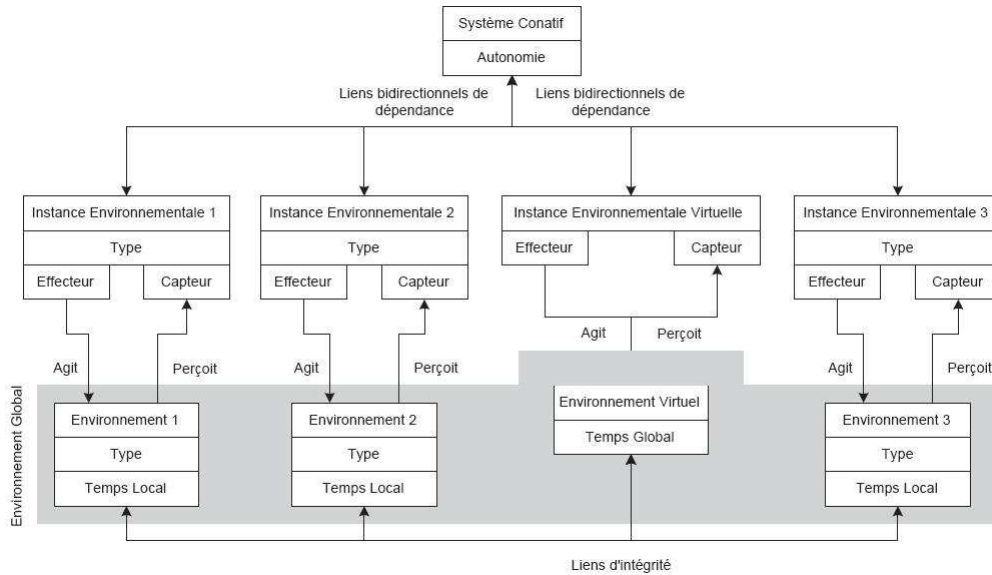


FIGURE 1.2 – Le modèle d’agent multi-environnemental

environnements. Cette coupure déporte les comportements complexes comme la détection des intersections, le déplacement d’agent dans d’autres modèles spécifiques à ces actions.

Les environnements, dans cette spécification liée au paradigme DEVS, sont donc des environnements centralisés. Toutes les informations sont regroupées au sein d’une seule entité ; mais aussi distribuées, puisque le modèle environnement peut être attaché, par exemple, à un automate cellulaire de type Cell-DEVS [Wainer et Giambiasi, 2001]. Ainsi, nous avons décidé de définir deux classes d’environnements : physique et social. La première pourra être dissociée en plusieurs environnements, continus, discrets et sous forme de graphe. Le deuxième représente le groupe du concept Agent-Groupe-Rôle (AGR) de J. Ferber [Ferber *et al.*, 2003].

Formalisation générique

Les environnements physiques contiendront les informations de types géographiques comme les positions et vitesses des agents, alors que l’environnement social contient lui, des informations comme les rôles disponibles et les comportements associés.

Les agents disposent d’informations que l’on nomme « publiques », dans le sens où ces informations ne sont disponibles que depuis l’environnement pour tous les agents. Ces informations sont utiles pour proposer des informations globales à l’environnement et permettent de minimiser les échanges d’informations entre les agents.

Les environnements sont des structures dont le rôle est de stocker les informations des agents

et de fournir un mécanisme de type questions-réponses aux requêtes des autres modèles. Ces modèles sont représentés sous forme de modèles couplés ou modèles atomiques. Leurs rôles sont simplement de stocker les informations des agents et des informations publiques de l'environnement lui-même. Nous formalisons donc l'environnement sous forme d'un modèle atomique :

$$M_{env} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \delta_{inst}, \delta_{con}, \lambda, ta \rangle \quad (1.1)$$

Environnements physiques

L'environnement physique est une entité dont le rôle est de contenir toutes les informations des entités sous forme de vecteurs positions et vitesses. Les informations que les entités laissent libres aux autres entités, ainsi que les informations relatives à l'environnement lui-même, comme la taille de l'espace et son nombre de dimension. L'environnement physique se représente comme un modèle atomique :

$$DEVS_{env.physique} = \langle X, Y, X_{init}, Y_{state}, S, \delta_{ext}, \delta_{int}, \delta_{init}, \delta_{state}, \delta_{inst}, \delta_{con}, \lambda, ta \rangle \quad (1.2)$$

L'état de l'environnement physique est défini par :

$$S = \{(phase, E, M)\} \quad (1.3)$$

Où *phase* possède la même définition que celle de l'environnement générique. La variable *M* définit les informations de l'environnement :

$$M = \{C_M, P_M\} \quad (1.4)$$

Où : C_M est un ensemble d'informations relatives à la représentation spatiale et P_M est l'ensemble de valeurs publiques de l'environnement.

À ce niveau, la formalisation n'est pas complète. En effet, la gestion des espaces physiques n'est décrite que via les variables C_M et P_M . Cependant, aucune information n'est fournie sur le type d'environnement et les spécificités de celui-ci. On distingue trois types d'environnements physiques suivant la qualification de l'espace de l'environnement, discret, continu ou graphe dont les variables E et C_M dépendent.

- **Environnements discrets** : dans ce cas, l'extension Cell-DEVS de DEVS spécialisée dans la modélisation d'automate cellulaire a été utilisée.

E définit les informations des agents par rapport à l'environnement physique, c'est-à-dire, les coordonnées, directions, vitesses et les informations publiques :

$$E_{env.discret} = \left\{ \left(C_E, \overrightarrow{D_E}, V_E, P_E \right) \right\} \quad (1.5)$$

Avec :

- $C_{E_i} \in \{C_E\}$ et $C_{E_i} \in \mathbb{Z}^n$ avec $n = \{1, 2, 3\}$, l'ensemble des coordonnées des entités dans un espace à 1, 2 ou 3 dimensions ;
- $\overrightarrow{D_{E_i}} \in \{\overrightarrow{D_E}\}$ et $\overrightarrow{D_{E_i}} \in \mathbb{Z}^n$ avec $n = \{1, 2, 3\}$, l'ensemble des vecteurs directions des entités ;

- $V_{E_i} \in \{VE\}$ et $V_{E_i} \in \mathbb{Z}$ l'ensemble des vitesses des entités ;
- $\{P_{E_i}\} \in \{PE\}$ et $\{P_{E_i}\} \in \mathbb{R}, \mathbb{N}, \mathbb{C}, \dots$, les valeurs publiques des entités proviennent de divers domaines.

L'espace C_M est formalisé, pour un exemple simple, par la taille, le nombre de cellules et leurs indices pour chaque axe de l'espace.

L'environnement discret possède une légère différence due à l'utilisation de l'extension Cell-DEVS pour représenter l'environnement. Certaines parties du corps de l'agent sont connectées directement aux cellules de l'automate cellulaire, comme l'effecteur de perturbation de la cellule. Cependant, l'autre partie de l'agent reste connectée à l'environnement pour la gestion des informations publiques comme, par exemple, l'effecteur de déplacement.

- **Environnements continus** : Dans les environnements continus, les informations de position, le vecteur vitesse se trouvent dans \mathbb{R}_n avec $n \in \{1, 2, 3\}$ suivant la dimension de l'espace. Les caractéristiques des agents, publiées dans l'environnement, définissent la variable E :

$$E_{env.continu} = \left\{ \left(C_E, \overrightarrow{D_E}, V_E, P_E \right) \right\} \quad (1.6)$$

Avec :

- $C_{E_i} \in \{C_E\}$ et $C_{E_i} \in \mathbb{Z}^n$ avec $n = \{1, 2, 3\}$, l'ensemble des coordonnées des entités dans un espace à 1, 2 ou 3 dimensions ;
- $\overrightarrow{D_{E_i}} \in \{\overrightarrow{D_E}\}$ et $\overrightarrow{D_{E_i}} \in \mathbb{Z}^n$ avec $n = \{1, 2, 3\}$, l'ensemble des vecteurs directions des entités ;
- $V_{E_i} \in \{VE\}$ et $V_{E_i} \in \mathbb{Z}$ l'ensemble des vitesses des entités ;
- $\{P_{E_i}\} \in \{PE\}$ et $\{P_{E_i}\} \in \mathbb{R}, \mathbb{N}, \mathbb{C}, \dots$, les valeurs publiques des entités proviennent de divers domaines.

L'espace C_M est formalisé, dans un exemple très simple, par deux points définissant un cube, un carré ou une ligne suivant le nombre de dimension de l'espace. Cet exemple est le plus simple, mais il permet de simplifier les calculs attachés aux modèles effecteurs et capteurs.

- **Environnements graphes** : Dans un environnement de relation nœuds/connexions, les caractéristiques des entités sont en relations avec les nœuds. Ainsi, E est défini comme :

$$C_{M_{env.graphes}} = \langle M, C \rangle \quad (1.7)$$

Où :

- M l'ensemble des places de l'environnement physique,
- $C = \{(i, j) \mid i, j \in M, i \neq j\}$ l'ensemble des connexions reliant deux places.

Et :

$$E_{env.graphes} = \{(C_E, D_E, P_E)\} \quad (1.8)$$

Où :

- $C_{E_i} \in \{C_E\}$ et $C_{E_i} \in C_M$, l'ensemble des places sur lesquelles se trouve l'entité,
- $D_{E_i} \in \{D_E\}$ et $D_{E_i} \in C_M$, l'ensemble des directions vers les places distances,
- $\{P_{E_i}\} \in \{PE\}$ et $\{P_{E_i}\} \in \mathbb{R}, \mathbb{N}, \mathbb{C}, \dots$, les valeurs publiques peuvent prendre des valeurs dans divers domaines.

Environnements sociaux

Cette spécification agent propose d'associer les groupes et les environnements sociaux. Avant d'aborder les caractéristiques des environnements sociaux, nous développons la notion de groupe dans les systèmes multi-agents et leur relation avec les rôles [Gutknecht *et al.*, 2001] des agents.

Spécification

L'environnement social ou le groupe possède les mêmes caractéristiques que les environnements physiques en localisant, dans une seule entité, les informations des agents ainsi que les informations de rôles disponibles et les interactions entre ces rôles. Les informations stockées dans cet environnement sont les informations relatives aux graphes d'accointances entre les agents. La structure DEVS se représente sous la forme d'un modèle atomique qui communique avec un modèle exécutif pour gérer les communications entre agents. Le modèle environnement social est défini comme un modèle atomique dont la structure est :

$$M_{env.social} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \delta_{inst}, \Lambda, ta \rangle \quad (1.9)$$

Organisation

Dans les systèmes multi-agents, l'organisation se réfère souvent au niveau social, où, comme pour les groupes, l'organisation régule, contraint et guide le comportement des agents (voir, par exemple : [Odell *et al.*, 2003] [Da Silva et De Lucena, 2004]). Comme pour la définition des groupes, nous n'attribuons pas de comportement autonome à l'organisation.

Une organisation suit alors le même comportement de question-réponse que les environnements, elle est définie par :

$$M_{organisation} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \delta_{inst}, \lambda, ta \rangle \quad (1.10)$$

Le modèle d'organisation repose exactement sur le comportement des groupes en ajoutant la possibilité aux agents, via leurs effecteurs, de créer ou de retirer des groupes en posant des événements sur son port de sortie Y_χ connecté au modèle exécutif du système multi-agents, en considérant celui-ci comme un réseau de modèle dynamique DSDEVS [Barros, 1995].

Lors d'une demande d'ajout d'un agent dans un groupe, deux niveaux de contrôles sont alors établis. Le premier par le groupe, celui-ci accepte la création de l'ajout de la part de l'effecteur et envoie, via son déclencheur d'ajout d'agent, un message à l'organisation pour l'ajout de l'agent. Celui-ci peut alors accepter ou refuser la connexion, suivant le perturbateur qui lui est connecté. Le deuxième niveau de contrôle est donc réalisé par l'organisation. Si les deux entités, l'organisation et le groupe, ont des choix différents, c'est au modélisateur de faire le choix.

1.3 Couplage d'équations aux différences

1.3.1 Présentation du problème

Cette section est relative à la problématique des équations aux différences. Ces approches étant très classiques et depuis de nombreuses années, il existe une littérature très abondante à ce sujet, mais on peut citer, par exemple : [Lakshmikantham et Trigianta, 1988]. Ces équations

permettent de créer des modèles basés sur un temps discret qui calculent la valeur d'une variable au temps t en fonction de sa propre valeur à $t - \Delta t, t - 2\Delta t, \dots$ et en fonction des valeurs d'autres variables à $t - \Delta t, t - 2\Delta t, \dots$. La construction de cette extension revêt une importance tout particulière car ces équations définissent une classe de modèles que l'on retrouve très fréquemment que ce soit dans le domaine de l'agronomie ou de l'halieutique (voir par exemple la section 3.5). Elles permettent, par exemple, d'exprimer très facilement la relation entre différentes fonctions en fonction du temps.

Plus formellement, si on considère une fonction X qui est exprimée en un pas de temps t , on peut la définir comme ceci :

$$\forall i \in \{1, \dots, n\}, X_i(t) = f_i(\underline{Z}_i(t), \underline{W}_i(t)) \quad (1.11)$$

Où :

$$\underline{Z}_i(t) = \begin{bmatrix} X_i(t - \Delta t) \\ \dots \\ X_i(0) \end{bmatrix} \quad (1.12)$$

Et :

$$\underline{W}_i(t) = [\dots, \underline{W}_j(t), \dots] \text{ pour } j \in \{1, \dots, n\} \text{ et } j \neq i \quad (1.13)$$

Et :

$$\underline{W}_i(t) = \begin{bmatrix} X_j(t) \\ X_j(t - \Delta t) \\ \dots \\ X_j(0) \end{bmatrix} \quad (1.14)$$

Certaines équations doivent être évaluées avec les variables d'entrée à leurs valeurs à t_{i-1} d'autres avec les valeurs calculées à t_i . Pour ces dernières, l'évaluation de l'équation doit impérativement attendre l'arrivée des bonnes valeurs en entrée pour pouvoir être évaluées ; l'ordre est ici important.

Considérons le graphe orienté dont :

- l'ensemble des sommets sont les équations ;
- les arêtes (orientées) sont les liens reliant les sorties des équations (les variables calculées par l'équation) aux entrées des autres (quand ces variables sont nécessaires, bien sûr).

Ce graphe doit être parcouru dans un ordre précis afin de pouvoir satisfaire les contraintes.

Dans le même registre, lors de l'initialisation, certaines données sont manquantes. On peut considérer trois cas pour la valeur d'une variable à l'initialisation :

1. la valeur est donnée, auquel cas, on peut penser que cette donnée est nécessaire pour calculer les autres à $t = 0$;
2. il est possible que le modélisateur précise la valeur pour réaliser un forçage sur la variable ;

3. la valeur n'est pas donnée, simplement car le modèle a toute la connaissance nécessaire pour la calculer puisque ce calcul est identique au calcul normal lors de la simulation.

Dans le dernier cas, le modèle doit évaluer les équations une à une en prenant garde de parcourir le graphe défini plus haut correctement, et ceci en fonction des données présentes ou absentes. L'ordre est primordial, ou sinon nous pouvons arriver à des situations de blocage alors que le système peut être entièrement calculé. Ce dernier point est important car le modèle devenant complexe, il est très difficile de savoir si le nombre de variables initialisées à une valeur injectées dans le modèle est suffisant pour pouvoir amorcer le calcul dans sa globalité. Il serait donc intéressant que l'outil de modélisation puisse, non seulement trouver un ordre d'évaluation juste, mais qu'en plus il signale une impossibilité logique dans le parcours du graphe.

Enfin pour terminer la synthèse des problèmes soulevés par le modèle on peut encore se poser cette série de questions ou de constatations :

- comment gérer, décrire toutes ces équations ?
- quelle méthode ou quel langage choisir pour la description des formules complexes ?
- comment expliciter facilement les dépendances entre toutes les équations ?
- comment décrire de façon compacte des équations qui sont très similaires ?
- et, peut-être le plus important, comment réaliser toutes ces opérations de façon efficace, rapide et sûre ?

1.3.2 TREQ : TRanslator EQUation

Dans un premier temps nous allons vous présenter la syntaxe choisie pour le XML qui va décrire le modèle, puis nous allons voir la proposition d'implémentation pour le modèle DEVS supportant la syntaxe et la logique développée.

Description de la syntaxe XML choisie

Nous avons commencé par scinder le XML en deux parties : une partie **PARAMETERS** pour régler les paramètres globaux de la simulation et une partie **ROOT** qui contient la description du modèle en tant que tel.

La partie PARAMETERS C'est sous cette balise que l'on va régler les différents pas de temps, ainsi que les entrées/sorties qui vont servir à paramétrer la simulation. En centralisant ces données, il est ainsi possible de factoriser les informations les concernant et épargner la redondance. Par exemple, il est possible d'associer un nom à un flux entrant qui peut être une base de données. Les paramètres configurant l'accès à cette base sont centralisés ici et toutes les futures requêtes se servent uniquement de la référence.

C'est dans cette optique que nous commençons cette partie par la gestion des pas de temps. Nous avons fait le choix d'obliger le modélisateur à fixer un pas de temps global, puis à définir les autres pas de temps éventuels comme des multiples de celui-ci. Cela n'est pas trop restrictif et cela va simplifier la mise au point du modèle atomique DEVS. La description se fait simplement par la donnée d'une série de balises et d'attributs comme dans l'exemple suivant :

```
<CLOCK STEP="0.5">
  <TOP NAME="top2" MULT="2"/>
</CLOCK>
```

Pour les entrées, le simulateur accepte pour l'instant de lire des fichiers CSV et de faire des requêtes dans des bases de données. Comme pour les pas de temps, nous pouvons associer un nom à une entrée précise, qui permet de récupérer la ou les valeurs désirées, en faisant suivre le nom d'une requête.

la partie ROOT Cette partie du XML va nous permettre de décrire le modèle proprement dit. Le problème qui à nos yeux était le plus important était de trouver un agencement des équations de telle sorte que les dépendances soient faciles à décrire, c'est pourquoi il a été décidé de placer les équations dans une structure hiérarchique.

Une équation est donc une boîte noire avec des entrées et des sorties, et la problématique est de décrire les connexions entre toutes les boîtes le plus facilement possible. Aussi, pour chaque équation, nous sommes obligés d'explicitier où il est possible de trouver dans le système toutes ses entrées. Ces entrées sont soit des sorties d'autres boîtes, soit des constantes. En plaçant toutes ces boîtes, que nous avons baptisées « modules de calcul » dans un arbre, il est possible de regrouper sur les différentes branches les modules qui ont un sens commun et donc des liaisons fortes. Ainsi, par un système de recherche de nom d'entrée, en parcourant l'arbre des feuilles vers la racine, nous n'avons, presque toujours, qu'à mettre le nom de la variable et l'algorithme est capable de trouver soit la constante soit la sortie du module qu'il faut utiliser. Reste les cas particuliers à traiter qui sont inévitables, auxquels nous avons répondu par l'ajout d'un système de description de chemins dans l'arbre semblable à ce que l'on utilise dans un système de fichiers.

Prenons un exemple, considérons l'extrait suivant :

```
<NODE NAME="NIVEAU 1">
  <MODULE NAME="module 1" .../>
  <CONSTANT NAME="A" VALUE="1"/>
  <NODE NAME="NIVEAU 2">
    <CONSTANT NAME="B" VALUE="1"/>
    <MODULE NAME="module 2" ...>
      <CONSTANT NAME="C" VALUE="1"/>
    </MODULE>
    <MODULE NAME="module 3" .../>
  </NODE>
</NODE>
```

Les balises <NODE> permettent la hiérarchisation des modules. Le module de calcul **module 1** peut accéder directement à la constante **A** puisqu'elle est à un niveau inférieur dans l'arbre, mais va devoir définir un chemin plus complet pour lire le contenu de **B**. Par contre **module 2**, par héritage peut accéder directement **A** et **B**. Enfin chaque entité possède deux noms : un nom et

un nom long, par exemple la constante **A** a comme nom long `ROOT:NIVEAU 1:A`. Une référence à une sortie de modèle ou à une constante peut se faire de trois façons :

1. par héritage, en tapant simplement le nom de la variable ;
2. avec le nom long ;
3. de façon relative en utilisant la notation « .. ». Par exemple si le module `module 3` veut lire la constante `C`, il peut utiliser le chemin « `..:module 2:C` ». le « .. » sert à sortir de `module 3`, on revient par conséquent un cran en arrière dans l'arbre, puis ensuite il faut rentrer dans `module 2` et lire `C`.

Le système étant, notamment, caractérisé par une taille conséquente, nous avons ajouté une gestion de traitement par lots via une structure répétitive `FOR` ainsi qu'une structure alternative `IF`. Ces deux constructions ayant la possibilité de tirer leurs expressions des flux entrants, donc des fichiers CSV et des bases de données. Ainsi il est possible de créer une boucle sur toutes les espèces présentes dans une colonne du fichier CSV `species`.

Enfin, nous avons incorporé un interpréteur d'expressions mathématiques pour évaluer durant la lecture des données de petits calculs qui peuvent être utiles au paramétrage du système. Il est capable de calculer des expressions contenant : `+` `-` `*` `/` `%`, les opérateurs logiques `.not`, `.and`, `.or` et `.xor`, les opérateurs de comparaisons `.eq`, `.neq`, `.gt`, `.lt`, `.geq` et `.leq` ainsi que les parenthèses. Son utilisation consiste à placer le calcul désiré entre `$ $` dans une chaîne de caractères du XML, et le résultat sera substitué au calcul. Bien sûr pour que cela puisse avoir un intérêt, en plus des constantes on peut manipuler dans ces expressions les variables lues dans les flux d'entrées ou définies dans le corps du fichier XML.

Description des équations Une fois les modules mis en place et le graphe des dépendances défini, il faut décrire la ou les équations (au sens large, elles peuvent prendre la forme d'un morceau de code) qui vont calculer les sorties en fonction des entrées. Concrètement l'utilisateur doit écrire une classe Python, ou C++, héritée de la classe `Treq` puis il a la possibilité de surcharger un certain nombre de fonctions si leurs comportements par défaut ne le satisfont pas. Enfin, nous avons adjoint au module un mode qui peut être (1) `PREEVALUATION` auquel cas `compute` est appelé avec des valeurs d'entrée du pas de temps précédent, (2) `POSTEVALUATION` auquel cas `compute` est appelé avec les valeurs courantes du pas de temps considéré et (3) un mode `LAZY` réservé au calcul des constantes.

Le « *translator* » Une fois la syntaxe choisie pour le XML, il fallu écrire un « *translator* » pour la plate-forme VLE. Le but est de parser le fichier XML et de produire une description du modèle DEVS qui va traduire le modèle. L'architecture DEVS que nous avons choisie repose sur l'idée d'un modèle atomique par `MODULE`. Le but de ce « *translator* » consiste essentiellement à :

- lire la partie XML lui étant réservée, collecter les informations, lancer l'interpréteur sur les petits calculs à effectuer, interpréter les requêtes, lire dans les fichiers CSV d'entrée, traduire les `FOR`, les `SET` et les `IF` et enfin construire en mémoire la hiérarchie déduite ;
- parcourir l'arbre obtenu en mémoire, vérifier la description des chemins pour les entrées (d'un module) et ainsi trouver pour chacune d'elle soit une valeur (si c'était une constante), soit une référence (à savoir son nom long) si c'est la variable de sortie d'un autre module ;

- enfin, parcourir à nouveau l’arbre une dernière fois pour cette fois-ci écrire le XML à transmettre à VLE grâce à toutes les informations collectées précédemment. Entre autres, pour chaque `MODULE`, il y a création d’un modèle atomique DEVS ainsi que ses connexions. Il faut en effet relier un modèle atomique à tous les modèles qui vont utiliser ses variables de sortie.

Description du modèle DEVS

L’écriture du modèle DEVS est certainement la partie la plus délicate de tout le processus. C’est un programme assez court qui traduit directement les algorithmes du modèle DEVS mais son écriture est souvent pénible. Avec l’expérience acquise sur les autres modèles DEVS déjà réalisés, il est souvent pratique de penser un modèle DEVS comme une machine à états, puis de faire réagir les fonctions DEVS appelées en fonction de l’état courant du modèle. C’est sur ce modèle que nous sommes parti, en déclarant 6 états possibles pour le modèle :

1. `INIT_WAITING` : uniquement pendant la phase d’initialisation, le modèle attend des valeurs pour ses variables d’entrées ;
2. `INIT_SENDING` : uniquement pendant la phase d’initialisation, pour que le modèle envoie les valeurs de ses variables de sortie aux modèles connectées à ses ports de sorties ;
3. `INIT_OK` : état transitoire à la fin de l’initialisation avant de commencer le processus normal de calcul, envoi, etc. ;
4. `WAITING` : en attente des valeurs d’entrées ;
5. `SENDING` : envoi des sorties ;
6. `IDLE` : inactif.

Pendant sa phase d’initialisation, le modèle atomique reçoit les informations envoyées par le « *translator* », à savoir :

- la liste de ses entrées et leurs types : sortie d’un modèle avec éventuellement sa valeur d’initialisation ou constante avec sa valeur ;
- la liste des sorties ;
- la localisation du code de l’équation ;
- et enfin une série d’informations supplémentaires, comme le `MODE` (préévaluation, postévaluation, ...), etc.

Nous allons présenter l’algorithme sous-tendant le modèle atomique mais avant cela signalons que le modélisateur n’aura, bien sûr, pas à se soucier de ces détails, même s’il veut redévelopper une partie du modèle en C++. Le modèle atomique définit, pour cela, une interface fonctionnelle afin d’appeler du code C++ spécifique à l’équation. La démarche sera la même que pour le modélisateur désirant travailler en Python. Il suffira, dans les deux cas, d’hériter d’une classe particulière (`Treq_module` pour le C++ et `Treq` pour Python) et d’accepter, ou redévelopper, le comportement des fonctions suivantes :

- `initialize()` est appelée avant le lancement de la simulation, par défaut cette fonction ne fait rien ;
- `auto_init()` est appelée pendant la phase d’initialisation. Par défaut, cette fonction attend l’arrivée de toutes les entrées du module puis appelle la fonction `compute(0)` ;

- `compute(t)` est appelée exactement au bon moment par le modèle DEVS afin de réclamer une évaluation de l'équation avec les valeurs courantes des entrées au pas de temps `t`. Par défaut, cette fonction ne fait rien ;
- `finalize()` est appelée à la fin de la simulation.

Pour interroger l'état en cours des variables, le modélisateur a accès aux fonctions suivantes :

- `has_in(name)` pendant la phase d'initialisation, pour savoir si la valeur de la variable `name` est disponible ;
- `has_out(name)` pendant la phase d'initialisation, pour savoir si la variable de sortie `name` a reçu une valeur d'initialisation ou s'il faut la calculer ;
- `get_in(name)` pour connaître la valeur d'une variable d'entrée du nom de `name` ;
- `get_out(name)` idem pour une variable de sortie ;
- `set_out(name, value)` afin de fixer à la valeur `value` la variable de sortie `name`.

Pour toutes ces fonctions, le nom `name` peut-être le nom long, le nom court ou un mélange entre les deux. Par exemple, pour une variable d'entrée de nom long `ROOT:BURO:K`, on peut indiquer le nom `BURO:K`, l'application cherche alors dans sa liste de variables d'entrée (propre au module) et s'il n'y a pas d'ambiguïté satisfait à la demande de l'utilisateur. Cette facilité n'est pas uniquement là pour raccourcir le code, elle permet l'utilisation de la même portion de code Python ou C++ pour plusieurs modules. En effet la liste des entrées étant spécifiée dans le module, on peut alors, sans ambiguïté, indiquer de façon naturelle les variables incriminées dans l'équation et l'application va charger les bonnes valeurs.

Par exemple, le calcul suivant nous permet d'évaluer l'équation de capture utilisée dans le projet ANR CHALOUPE (cf. 3.5) :

$$C_{s,c} = \frac{N_{s,c} \times (1 - e^{-Z_{s,c}})}{Z_{s,c}} \times F_{s,c}$$

```

1 class Catch(Treq):
2     def compute(self, t):
3         self.set_out("C",
4             self.get_in("N")*(1-exp(-self.get_in("Z"))))
5         /self.get_in("Z")
6         *self.get_in("F"))

```

La déclaration du module correspondant est :

```

<MODULE NAME="CATCH" PYTHON="chaloupe" CLASS="Catch">
  <IN PATH="..:BIOLOGY:N"/>
  <IN PATH="..:TOTAL_MORTALITY:Z"/>
  <IN PATH="..:FISHING_MORTALITY:F"/>
  <OUT NAME="C" CSV_NAME="Catch:SPECIES/CLASS:$species$/$class$"
    SCRIBE="catch" />
</MODULE>

```

Voyons maintenant la description du modèle DEVS proprement dite. Il faut donc expliciter l'ensemble des fonctions DEVS : ta , λ , δ_{ext} , etc. Afin de faciliter la compréhension de l'ensemble, nous avons réalisé un schéma simplifié de l'état du modèle atomique (cf. Figure 1.3). Les simplifications portent essentiellement sur la gestion des erreurs et l'omission volontaire des transitions

externes pour la mise à jour des valeurs des variables, elles apparaissent sur seulement deux états (INIT_WAITING et WAITING) alors qu'elles sont susceptibles d'être appelées pour tous les états.

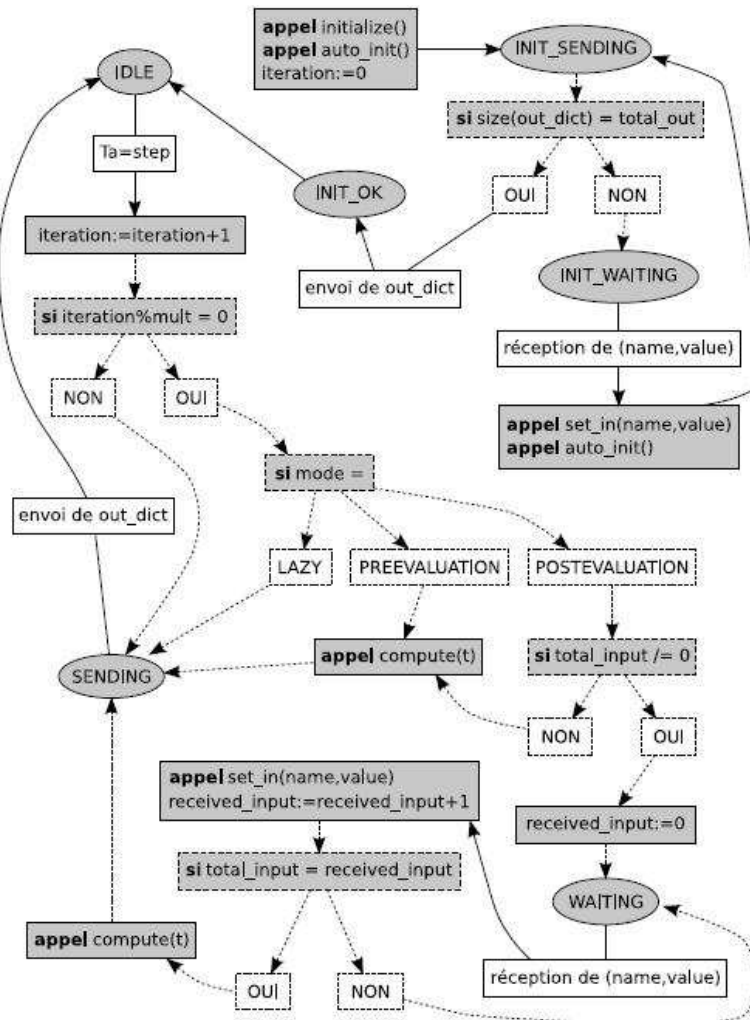


FIGURE 1.3 – Schéma simplifié du graphe d'état du modèle atomique

Commençons, tout d'abord, par les fonctions *init* et *finalize* qui sont appelées respectivement juste avant et après la simulation.

```

1 appel initialize ()
2 appel auto_init ()
3 iteration ← 0 // iteration compte le nombre de « ~top ~ »
4 state ← INIT_SENDING

```

Listing 1.1 – La fonction *init*

Comme on peut le voir la fonction met l'état sur INIT_SENDING. Les modèles atomiques vont donc tous commencer par envoyer les sorties qu'ils ont en leur possession. Cet ensemble

peut être vide à ce stade.

```
1 appel finalize ()
```

Listing 1.2 – La fonction *finalize*

Puis vient la fonction *ta* qui retourne le temps qui doit s'écouler avant le prochain appel à la fonction δ_{int} . Il n'y a que deux possibilités pour notre modèle 0 ou **step** qui correspond au pas de temps général du système (l'attribut **STEP** de la balise **<CLOCK>**).

```
1 si state=INIT_WAITING alors
2   retourne step
3 sinon si state=INIT_SENDING alors
4   retourne 0
5 sinon si state=INIT_OK alors
6   retourne 0
7 sinon si state=WAITING alors
8   retourne step
9 sinon si state=SENDING alors
10  retourne 0
11 sinon si state=IDLE alors
12  retourne step
13 fin si
```

Listing 1.3 – La fonction *ta*

La fonction qui suit s'occupe d'envoyer des événements aux autres modules. Les seuls messages envoyés serviront à indiquer les nouvelles valeurs des sorties aux autres modules, ils seront par conséquent sous la forme du nom long de la variable suivi de sa valeur (un nombre réel). Le modèle atomique possède un port par variable calculée. Ainsi, en plaçant ces événements sur les bons ports, seuls les modules nécessitant la valeur de cette sortie vont la recevoir.

```
1 si state=INIT_WAITING alors
2   retourne «Rien»
3 sinon si state=INIT_SENDING alors
4   si size(out_dict)=total_out
5     retourne «Toutes les sorties sur leurs ports»
6     state←INIT_OK
7   sinon
8     retourne «Rien»
9   fin si
10 sinon si state=IDLE alors
11  retourne No event
12 sinon si state=WAITING alors
13  retourne No event
14 sinon si state=SENDING alors
15  retourne «Toutes les sorties sur leurs ports»
16 fin si
```

Listing 1.4 – La fonction λ

La fonction δ_{int} sert, dans la logique « machine à états », à changer d'état. C'est ici qu'il est possible de vérifier si tout est correct dans le graphe de connexions des équations. En effet, le module se met en état d'attente des valeurs nécessaires à son propre calcul pendant la phase

d'initialisation (état : `INIT_WAITING`). Comme cet état est associé à un *ta* de `step`, si nous arrivons dans la fonction δ_{int} avec cet état, c'est que nous sommes arrivés au pas de temps $1 \times \text{step}$ et que la phase d'initialisation n'est pas terminée. Une erreur s'est donc produite, elle ne peut venir que d'un manque de valeurs d'entrée pour ce module, donc d'une erreur dans le graphe. Le modélisateur doit revoir son modèle, en particulier, il doit ajouter des valeurs d'initialisation à ses variables, car le modèle ne peut pas les trouver seul.

```

1 si state=INIT_WAITING alors
2   erreur «Ce module n'a pas reçu suffisamment de
3     donnée pour faire son auto-initialisation»
4 sinon si state=INIT_SENDING alors
5   state←INIT_WAITING
6 sinon si state=INIT_OK alors
7   state←IDLE
8 sinon si state=IDLE alors
9   iteration←iteration+1
10  si iteration modulo mult=0 alors
11    si mode=PREEVALUATION alors
12      appel compute(t)
13      state←SENDING
14    sinon si mode=POSTEVALUATION alors
15      si total_input ≠ 0 alors
16        received_input←0
17        state←WAITING
18      sinon
19        appel compute(t)
20        state←SENDING
21      fin si
22    sinon si mode=LAZY alors
23      state←SENDING
24    fin si
25  fin si
26 sinon si state=WAITING alors nop
27 sinon si state=SENDING alors
28   state←IDLE
29 fin si

```

Listing 1.5 – La fonction δ_{int}

Enfin, terminons cette description des fonctions DEVS du modèle par la fonction δ_{ext} qui se doit de réagir aux perturbations du modèle atomique. Ici les perturbations ne sont que l'arrivée de nouvelles valeurs pour les variables en entrée du module.

```

1 appel set_in(name, value)
2 si state=INIT_WAITING alors
3   appel auto_init()
4   state←INIT_SENDING
5 sinon si state=INIT_SENDING alors
6   appel auto_init()
7 sinon si state=INIT_OK alors
8   nop
9 sinon si state=IDLE alors
10  nop
11 sinon si state=WAITING alors

```

```

12 received_input←received_input+1
13 si total_input=received_input alors
14     appel compute(t)
15     state←SENDING
16 sinon si state=SENDING alors
17     nop
18 fin si
19 fin si

```

Listing 1.6 – La fonction δ_{ext} . L'événement reçu par la fonction comporte deux arguments, le nom (name) de la variable, ainsi que sa valeur (value)

1.4 Conclusion

Dans ce chapitre, nous montrons comment nous avons ajouté de nouvelles approches mono-paradigme dans l'environnement VLE.

Dans une première section, nous avons présenté un travail qui nous a permis de formaliser l'environnement et les environnements multiples dans les systèmes multi-agents en utilisant la théorie de la modélisation et la simulation et en exploitant, au mieux, les concepts de la systémique et de DEVS avec la composition et la hiérarchisation de modèles.

Cette spécification offre de multiples avantages de modélisation de systèmes multi-agents avec le formalisme DEVS. L'un d'eux est la simplification des environnements, permettant d'exploiter au mieux les concepts de DEVS. De plus, si la charge d'information est trop importante pour une structure informatique unique, les modèles environnements peuvent être développés sous forme de modèles couplés afin de répartir la charge sur plusieurs unités de calcul. Enfin, la réutilisabilité des modèles est aussi un point important de notre spécification.

Cette approche a été partiellement mise en œuvre dans la section 3.4. Modèle qui est juste un test de faisabilité et de validation.

Dans une deuxième section, nous avons présenté un travail qui nous a permis d'ajouter les équations aux différences dans la cadre de travail VLE tout en s'appuyant sur DEVS.

Cette extension était extrêmement attendue par la communauté des développeurs VLE car l'utilisation des équations aux différences est massivement répandue dans le domaine d'application que nous explorons. Cette extension permet, assez simplement, de développer des modèles à base d'équations aux différences et bien évidemment de les coupler avec des modèles s'appuyant sur d'autres paradigmes.

Cette approche a été totalement mise en œuvre dans la section 3.5 relative au projet Chaloupe. (projet ANR dont j'ai assuré la responsabilité de la tâche de modélisation) et le projet EcoMeristem (un des modèles phare de notre unité dont j'assure le développement et qui a été financé par de nombreux projets nationaux et internationaux).

Chapitre 2

Approche multi-paradigmes

Sommaire

2.1	Introduction	27
2.2	Couplage de modèles discrets et d'équations différentielles	27
2.2.1	Résolution d'équations différentielles spatialisées	28
2.3	Les méthodes d'intégration QSS1 et QSS2	28
2.3.1	Présentation de QSS	29
2.3.2	Mise en œuvre et validation	31
2.4	Application aux équations différentielles spatialisées	35
2.4.1	Application aux équations aux dérivées partielles	36
2.4.2	Test de la méthode	39
2.4.3	Gestion des perturbations	41
2.5	Conclusion	42

2.1 Introduction

Ce deuxième chapitre montre comment, grâce à la construction d'approches basées sur un seul paradigme (voir chapitre 1), on peut construire des modèles qui combinent des approches provenant de plusieurs paradigmes. La encore, nous nous situons dans la démarche qui est la notre, à savoir de proposer des extensions permettant de proposer des formalismes puissants tout en masquant la complexité liée à l'utilisation de DEVS.

Nous allons présenter les schémas d'intégration numérique QSS1 [Kofman et Junco, 2001] et QSS2 [Kofman, 2002]. Ces schémas d'intégration sont particulièrement bien adaptés à DEVS et donc à VLE. De plus, nous allons montrer comment on peut prendre en charge des perturbations ponctuelles (dont à pas de temps discrets) sur le schéma d'intégration.

2.2 Couplage de modèles discrets et d'équations différentielles

Dans cette section, nous allons nous attacher à montrer comment grâce à la plate-forme VLE [Quesnel *et al.*, 2008] et au formalisme DEVS [Zeigler, 1976], nous pouvons coupler des

modèles discrets et des modèles à base d'équations différentielles. Par la même, nous montrons que l'approche de couplage est valide et opérationnelle. Elle est d'autant plus opérationnelle qu'elle sera concrètement mise en œuvre dans la section 3.4.

2.2.1 Résolution d'équations différentielles spatialisées

Il existe également de nombreuses méthodes de résolution d'équations différentielles spatialisées. Nous allons présenter la méthode dite explicite (voir [Fletcher, 2000]). Pour ce faire, prenons l'exemple simple de l'équation de la chaleur :

$$\frac{\partial T(t, x)}{\partial t} = K \frac{\partial^2 T(t, x)}{\partial x^2} \quad (2.1)$$

Cette équation traduit la diffusion de la chaleur le long d'une barre, T est bien sûr la température qui dépend du temps t et de la position sur barre x et K est le coefficient de diffusion.

Afin d'appliquer la méthode, nous allons devoir cette fois discrétiser le temps mais aussi l'espace. Commençons par cette dernière dimension : on approxime $T(t, x)$ par ses valeurs en différents points de la barre après avoir choisi un pas spatial Δx . En notant $T_i(t)$ ces valeurs, l'équation 2.1 devient :

$$\frac{\partial T_i(t)}{\partial t} = K \frac{T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)}{(\Delta x)^2}, \text{ pour } i \text{ allant de } 1 \text{ à } N \quad (2.2)$$

Que l'on peut réécrire comme suit :

$$T_i(t + \Delta t) = T_i(t) + \Delta t \times K \frac{T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)}{(\Delta x)^2} \quad (2.3)$$

Connaissant l'ensemble des $T_i(t)$ a une date donnée, cette dernière formule nous permet d'obtenir une approximation de ce même ensemble mais à la date $t + \Delta t$.

C'est la méthode la plus simple, elle nous servira de comparatif dans les tests des section suivantes.

2.3 Les méthodes d'intégration QSS1 et QSS2

Les méthodes de résolution basées sur les événements discrets sont issues principalement du principe de quantification. L'idée générale est de définir une discrétisation de l'espace des valeurs au lieu de discrétiser le temps. La quantification introduit alors une base pour définir les dates de changement de quantum, ces dates permettent de rejoindre la notion d'événement. Nous allons voir qu'avec QSS1 et QSS2 le calcul des quantums est varié (voir [Zeigler, 1998], [Zeigler et Lee, 1998] et [Zeigler *et al.*, 2001]).

2.3.1 Présentation de QSS

E. Kofman propose, à travers QSS1 et QSS2, deux algorithmes pour la résolution numérique d'équations différentielles du premier ordre. Pour les deux méthodes, considérons le système général suivant :

$$\begin{cases} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{cases} \quad (2.4)$$

x , u et y sont des fonctions d'une variable (en t) mais peuvent être des vecteurs d'une dimension quelconque. On peut remarquer la dépendance en t de f qui se fait via une fonction u , ce qui est une façon d'intégrer la gestion du temps au niveau de f , mais aussi surtout d'une éventuelle fonction externe au système. La fonction y est la sortie désirée du système, c'est-à-dire la fonction que l'on désire observer. Enfin f et g permettent respectivement le calcul \dot{x} et de y . Ce cas d'étude va nous donner une bonne idée du fonctionnement d'un graphe de modules DEVS connectés dont la fonction est ici de résoudre un système possédant une équation différentielle. Le système dans sa globalité peut être vu comme une boîte possédant des entrées (u) et des sorties (y). La difficulté est bien sûr de trouver x , et une version simplifiée de la première équation aurait été suffisante pour expliquer le fonctionnement des QSS, mais avec cette façon de faire E. Kofman place ainsi directement QSS1 et QSS2 en situation concrète de fonctionnement.

QSS1

QSS1 repose sur deux principes :

- La quantification avec hystérésis de la fonction x ;
- Un calcul du pas de temps en cohérence avec $\dot{x}(t)$ et la quantification.

Ceci explique le sigle QSS choisi par E. Kofman pour : « *Quantized State Systems* ».

Détaillons davantage la méthode en commençant par la quantification avec hystérésis : celle-ci consiste à remplacer une fonction à valeurs réelles par une fonction constante par morceaux en utilisant une discrétisation de \mathbb{R} (qui peut ne pas être à pas constant) mais en incorporant de l'hystérésis dans le processus. Par exemple : soit $\dots, d_{-1}, d_0, d_1, \dots$ une discrétisation de \mathbb{R} (donc $d_i < d_{i+1}$), supposons que $x(t_0)$ soit légèrement supérieur à d_4 et que x soit croissante. Alors l'approximation de x sera d_4 et dès que x atteindra d_5 , l'approximation sera d_5 , etc. Mais si x est décroissante, alors il faut attendre que x atteigne la valeur $d_4 - \epsilon$ pour que l'approximation devienne d_3 . E. Kofman appelle ϵ la fenêtre d'hystérésis. Tout ceci se résume sur la figure 2.1 où q est l'approximation de x et où les flèches représentent le sens de variation de x .

Le modèle DEVS Le schéma de la figure 2.2 représente le système dans sa globalité. La machine DEVS M_F s'occupe de calculer les \dot{x}_j qu'elle envoie aux modèles DEVS M_j . L'approximation est alors calculée et renvoyée à M_F ainsi qu'à M_G pour le calcul de la sortie globale.

QSS2

L'idée est de garder les concepts présentés dans QSS1, mais en passant à un degré supérieur (voir [Kofman, 2002]) : on n'approxime plus la sortie par une constante mais par une fonction

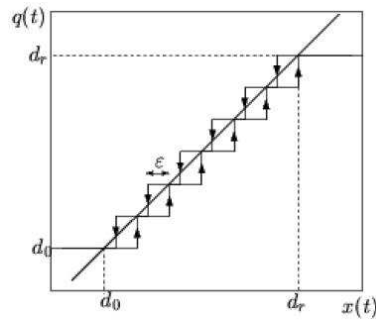


FIGURE 2.1 – Quantification avec hystérésis d'une fonction, tiré de [Kofman et Junco, 2001]

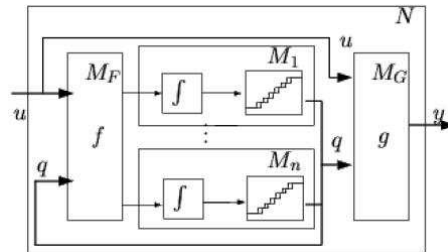
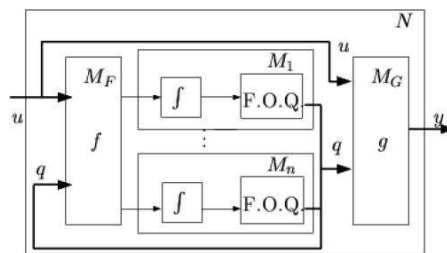


FIGURE 2.2 – Modèle DEVS de QSS1, tiré de [Kofman et Junco, 2001]

affine et pour réaliser cela, f nous fournit la dérivée de x_j et également sa dérivée seconde, le modèle conservant la même architecture, cf. Figure 2.3.

FIGURE 2.3 – Modèle DEVS de QSS2, tiré de [Kofman, 2002], F.O.Q. est pour « *First Order Quantizer* »

La machine DEVS M_j réalise donc une approximation de la variable $x_j(t)$ à l'aide de x_j , \dot{x}_j et \ddot{x}_j via la formule de Taylor. Ainsi la représentation de $x_j(t)$ est une parabole. Elle calcule ensuite σ de sorte que la différence entre la parabole et son approximation par une fonction affine soit inférieure à un certain Δq . On retrouve de l'hystérésis à travers le calcul de σ . (voir Figure 2.4).

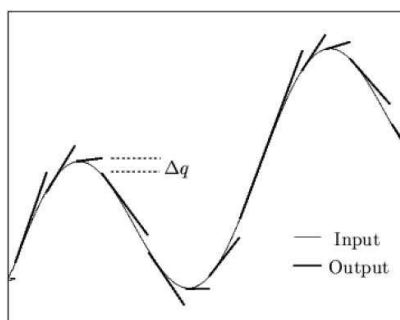


FIGURE 2.4 – Quantification d'ordre 1, tiré de [Kofman, 2002]

2.3.2 Mise en œuvre et validation

Pour tester les deux algorithmes, nous avons réalisé un prototype d'implémentation en C++ auquel nous avons soumis quatre tests différents.

Présentation des programmes

Dans les deux cas, nous avons simplifié le superviseur par rapport à un modèle DEVS classique : en effet, on peut remarquer dans les deux programmes que, dès qu'une machine M_j génère un événement, toutes les autres machines M_j activent leur fonction de transition externe. Elles recalculent donc toutes leurs ta et ainsi, pour connaître le prochain événement, il suffit de trouver le plus petit ta .

Présentation des quatre tests employés

cos_test Pour ce premier test, considérons le système suivant :

$$\begin{cases} \dot{x}_1(t) = x_2(t) & \text{avec } x_1(0) = 0 \\ \dot{x}_2(t) = -x_1(t) & \text{avec } x_2(0) = 1 \\ y(t) = x_2(t) \end{cases}$$

avec $t \in [0; 10]$. La solution de ce système est bien sûr :

$$\begin{cases} x_1(t) = \sin(t) \\ x_2(t) = \cos(t) \\ y(t) = \cos(t) \end{cases}$$

$y(t)$ étant la sortie, nous allons pouvoir comparer les solutions obtenues avec la fonction $\cos(t)$.

volterra_test Pour ce deuxième test, utilisons les équations de Lotka-Volterra [Volterra, 1926], [Volterra, 1931] et [Lotka, 1925] :

$$\begin{cases} \dot{x}_1(t) = a \times x_1(t) - x_1(t) \times x_2(t) & \text{avec } x_1(0) = 2 \\ \dot{x}_2(t) = -b \times x_2(t) + x_1(t) \times x_2(t) & \text{avec } x_2(0) = 2 \\ y(x_1) = x_2 \end{cases}$$

avec $a = 0.4$, $b = 1.2$ et $t \in [0; 15]$.

Puisque nous ne connaissons pas la solution exacte, nous allons juger de la qualité de l'approximation par l'obtention d'une courbe ($y(x_1) = x_2$) qui doit se refermer sur elle-même.

degre4_test Pour ce test, nous avons voulu mettre en pratique la transformation « équation d'ordre p - équation d'ordre 1 ». L'équation choisie est la suivante :

$$y^{(4)} = -y - 2y''$$

avec $y(0) = 0, y'(0) = 1, y''(0) = 0$ et $y^{(3)} = -3$. La solution exacte de cette équation est $y(x) = x \times \cos(x)$. Après transformation en un système de degré 1, nous obtenons les équations suivantes :

$$\begin{cases} \dot{x}_0(t) = x_1(t) & \text{avec } x_0(0) = 0 \\ \dot{x}_1(t) = x_2(t) & \text{avec } x_1(0) = 1 \\ \dot{x}_2(t) = x_3(t) & \text{avec } x_2(0) = 0 \\ \dot{x}_3(t) = -x_0(t) - 2x_2(t) & \text{avec } x_3(0) = -3 \\ y(t) = x_0(t) \end{cases}$$

avec $t \in [0; 10]$. De nouveau, nous allons pouvoir comparer les solutions obtenues avec $t \times \cos(t)$.

RLC_test E. Kofman propose lui-même un exemple dans [Kofman et Junco, 2001]. Nous l'avons repris pour tester les deux programmes.

$$\begin{cases} \dot{x}_1(t) = \frac{1}{L}x_2(t) & \text{avec } x_1(0) = 0 \\ \dot{x}_2(t) = U - \frac{1}{C}x_1(t) - \frac{R}{L}x_2(t) & \text{avec } x_2(0) = 0 \\ y(t) = \frac{1}{L}x_2(t) \end{cases}$$

où les paramètres sont : $R = 100.01, L = 0.01, C = 0.01, U = 100$ et $t \in [0; 10]$. La solution exacte de ce système est :

$$y(t) = \frac{10000}{9999}(e^{-t} - e^{-10000t})$$

Les résultats

Dans tous les tests, nous avons fixé pour QSS1 le pas de quantification, ainsi que ϵ , à 10^{-4} . Et pour QSS2, nous avons fixé Δq à 10^{-4} . La seule exception est le test RLC où nous avons repris les choix de E. Kofman à savoir, pour QSS1, un pas de quantification et un ϵ à 10^{-2} pour x_1 et à 10^{-4} pour x_2 . En effet, les machines DEVS M_j étant indépendantes, nous pouvons les régler sur des précisions différentes.

On peut remarquer l'égalité systématique du pas de quantification avec ϵ . Ce choix, qu'E. Kofman souligne, permet de mettre en cohérence la sortie de d_{i-1} dans QSS1 avec la valeur réelle de x_j lorsque la variable est décroissante.

Sortie des Volterra

Nous obtenons les tableaux 2.1 et 2.2. La colonne « précision » correspond à la plus grande erreur relevée pendant le calcul en comparaison avec la solution exacte.

TABLE 2.1 – Résultats des tests pour QSS1

Tests	QSS1		
	itérations	précision	temps
cos_test	127079	0.0008	4.8 s
volterra_test	154724	cf. figure 2.5	5.75 s
degre4_test	1484549	0.0012	54.49 s
RLC_test	301	0.01	0.03 s

TABLE 2.2 – Résultats des tests pour QSS2

Tests	QSS2		
	itérations	précision	temps
cos_test	1073	0.0005	0.05 s
volterra_test	1366	cf. figure 2.6	0.04 s
degre4_test	5357	0.0007	0.24 s
RLC_test	130931	0.02	5.54 s

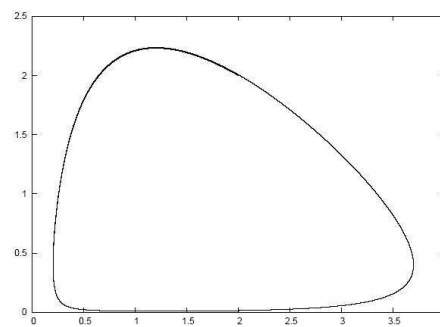


FIGURE 2.5 – Sortie de QSS1 pour le volterra_test

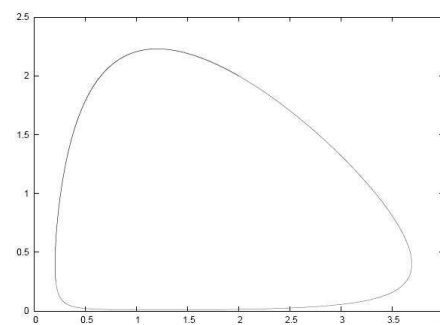
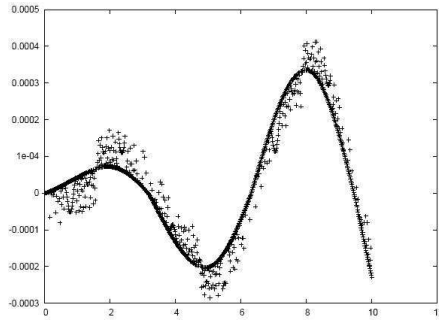
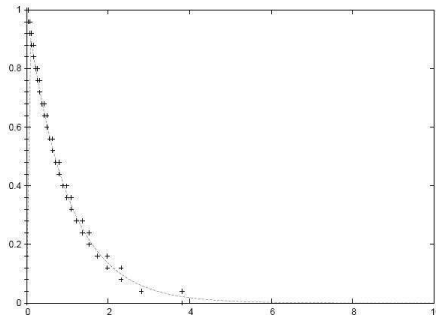


FIGURE 2.6 – Sortie de QSS2 pour le volterra_test

FIGURE 2.7 – Erreur sur le `cos_test` dans QSS2FIGURE 2.8 – Sortie du `RLC_test` dans QSS1 comparée avec $y(t)$

Les Figures 2.5 et 2.6 nous donnent les sorties de QSS1 et QSS2 pour le `volterra_test`. La Figure 2.7 nous montre l'évolution de l'erreur dans QSS2 pour le `cos_test` et enfin la Figure 2.8 fait une comparaison entre la sortie de QSS1 et la fonction réelle pour le `RLC_test`.

Interprétations

Pour l'instant, mettons à part le `RLC_test` qui nous a posé des problèmes de précision machine. Nous y reviendrons à la fin du paragraphe.

Dans un premier temps, nous pouvons remarquer la nette diminution du nombre d'itérations entre QSS1 et QSS2, et donc du temps de calcul. Ceci s'explique certainement par l'emploi de fonctions affines dans QSS2 qui permettent de rester près de x_j un temps plus long que si l'on utilise des constantes comme dans QSS1. Soulignons également le fait que l'on ne maîtrise pas le temps d'exécution du programme.

On peut noter que cette diminution d'itérations n'a pas été à l'encontre de la précision, au contraire celle-ci est toujours meilleure dans QSS2.

Enfin, on peut constater les très bons résultats de QSS2 : une résolution d'une équation d'ordre 4, pour $t \in [0; 10]$ en seulement 5357 itérations et une erreur plus petite que 7.10^{-4} en valeur absolue !

Le `RLC_test` est le seul test qui contredit l'amélioration lors du passage à QSS2. Nous pen-

TABLE 2.3 – Comparaison de différentes méthodes sur le RLC_test

Méthodes	Nombre de pas
Euler	150000
Runge-Kutta	90000
Runge-Kutta 4-5	30000
Adams-Bashforth-Moulton	60000
Matlab's ode15s	81

sons que ceci est dû à un problème de précision machine. En effet, si on regarde la solution :

$$y(t) = \frac{10000}{9999}(e^{-t} - e^{-10000t})$$

on constate la présence de $e^{-10000t}$. Et en posant $g(t) = e^{-10000t}$, on a $g''(t) = 10^8 e^{-10000t}$. La dérivée seconde va donc prendre des valeurs énormes comparées aux autres valeurs présentes ce qui va poser des problèmes lors de l'addition de ces nombres. Pour nous convaincre, nous avons augmenté progressivement la précision dans le cos_test et nous avons obtenu un décrochement de la méthode à partir de 10^{-12} , or toutes les dérivées et dérivées secondes du cos_test sont bornées en valeur absolue par 1. On a donc une différence entre précision et valeurs prises du même ordre de grandeur que pour le RLC_test, ce qui peut expliquer cette imprécision. Les résultats fournis par QSS1 sont parfaitement similaires aux résultats de E. Kofman. QSS1 est moins sensible à ces problèmes que QSS2, du fait qu'il n'utilise pas la dérivée seconde de x_j .

Comparaison avec d'autres méthodes

E. Kofman a réalisé dans [Kofman et Junco, 2001] une comparaison de QSS1 avec d'autres méthodes connues. Il est parti du principe que pour le test RLC, QSS1 a obtenu une précision de 10^{-2} , puis il a cherché comment obtenir la même précision avec les autres méthodes. Il a ensuite comparé le nombre de pas de calculs nécessaires. Il a obtenu les résultats du tableau 2.3. Nous pouvons voir que pour toutes les méthodes le nombre de pas de calcul est bien supérieur à celui de QSS1 (qui était de 301 itérations). Seule la méthode Matlab's ode15s réalise une meilleure performance. Mais selon E. Kofman, c'est au prix d'un calcul par pas très complexe (méthode d'ordre 5, inversion de matrice, etc) sans comparaison avec la simplicité de QSS1. B. P. Zeigler trouve dans [Zeigler, 1998] des résultats tout aussi encourageants en soulignant en plus que, pour QSS1, les messages qui transitent au sein du système sont +1 ou -1 (pour d_{i+1} ou d_{i-1}) et tiennent donc sur 1 bit.

2.4 Application aux équations différentielles spatialisées

Comme annoncé dans la section précédente, nous allons proposer une méthode pour utiliser QSS1 et QSS2 sur un plus grand panel d'équations. Premièrement, nous allons expliquer la démarche et la méthode, puis nous passerons à une série de tests et nous terminerons par une discussion sur les performances ainsi que sur des pistes possibles pour les améliorer.

2.4.1 Application aux équations aux dérivées partielles

Principe

Si maintenant nous essayons d'appliquer directement les méthodes QSS1 et QSS2 aux équations différentielles manipulant des fonctions qui ne dépendent plus seulement du temps, alors nous nous heurtons à une impossibilité logique. En effet, QSS1 et QSS2 reposent sur la notion d'événements discrets ce qui implique une dépendance des processus vis à vis du temps. Par contre, il est possible de réaliser une méthode hybride où la gestion du temps se ferait par un intégrateur type QSS et la gestion des dépendances spatiales par des méthodes classiques d'approximation de dérivée.

Un exemple d'application Afin d'illustrer notre propos, reprenons l'exemple de la diffusion de la température sur une barre. Celle-ci est gérée [Fletcher, 2000] par l'équation suivante :

$$\frac{\partial T(t, x)}{\partial t} = K \frac{\partial^2 T(t, x)}{\partial x^2} \quad (2.5)$$

où K est le coefficient de diffusion.

Repartons du calcul présenté dans la méthode explicite. On approxime $T(t, x)$ par ses valeurs en différents points de la barre après avoir choisi un pas spatial Δx . Si on note $T_i(t)$ ces valeurs, l'équation 2.5 devient :

$$\frac{\partial T_i(t)}{\partial t} = K \frac{T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)}{(\Delta x)^2}, \text{ pour } i \text{ allant de } 1 \text{ à } N \quad (2.6)$$

en utilisant, par exemple, une approximation de la dérivée seconde en trois points.

La méthode explicite consisterait maintenant à approximer $\frac{\partial T_i(t)}{\partial t}$ par $\frac{T_i(t+\Delta t) - T_i(t)}{\Delta t}$, mais à ce niveau les équations forment un système d'équations différentielles du premier ordre en $T_i(t)$ (cf. 2.7). Il est donc possible de les résoudre à l'aide d'une méthode type QSS.

$$\begin{pmatrix} \dot{T}_1(t) \\ \dot{T}_2(t) \\ \vdots \\ \dot{T}_{N-1}(t) \\ \dot{T}_N(t) \end{pmatrix} = \begin{pmatrix} \frac{K}{(\Delta x)^2} & \frac{-2K}{(\Delta x)^2} & \frac{K}{(\Delta x)^2} & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & \frac{K}{(\Delta x)^2} & \frac{-2K}{(\Delta x)^2} & \frac{K}{(\Delta x)^2} & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \frac{K}{(\Delta x)^2} & \frac{-2K}{(\Delta x)^2} & \frac{K}{(\Delta x)^2} \end{pmatrix} \begin{pmatrix} \text{Limite droite} \\ T_1(t) \\ \vdots \\ T_N(t) \\ \text{Limite gauche} \end{pmatrix} \quad (2.7)$$

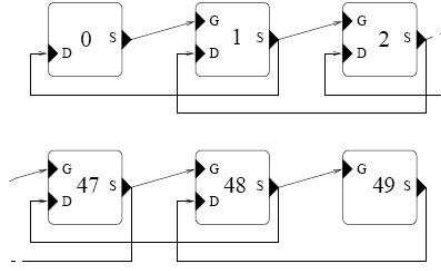


FIGURE 2.9 – Le modèle Cell-DEVS

La formalisation DEVS Supposons que l'on découpe la barre en 50 morceaux, on peut modéliser le problème par 50 cellules Cell-DEVS interconnectées (cf. figure 2.9). Une cellule possède un port de sortie S qui envoie à ses voisins la nouvelle température avant l'évaluation de sa fonction de transition interne. Elle possède également deux ports d'entrée G et D pour recevoir la température des voisins par activation de sa fonction de transition externe. Dans notre cas, le délai d'envoi de l'état de la cellule est nul. Les voisins sont donc immédiatement informés du changement d'état.

Type d'équations intégrables

En appliquant directement le principe de discrétisation de l'espace et la transformation en système linéaire d'ordre 1, il est déjà possible d'intégrer des équations du type :

$$\frac{\partial^n T}{\partial t^n} = \varphi \left(\frac{\partial^{n-1} T}{\partial t^{n-1}}, \dots, \frac{\partial T}{\partial t}, T, \frac{\partial T}{\partial x_1}, \frac{\partial^2 T}{\partial x_1 \partial x_2}, \dots, \frac{\partial^p T}{\partial x_{i_1} \dots \partial x_{i_p}}, \dots \right) \quad (2.8)$$

dès l'instant où une approximation de $\frac{\partial^p T}{\partial x_{i_1} \dots \partial x_{i_p}}$ est possible, par exemple pour la dérivée, la dérivée seconde, etc.

Démontrons cette affirmation : notons $f_i(t)$ la discrétisation de $\frac{\partial^p T}{\partial x_{i_1} \dots \partial x_{i_p}}$. $f_i(t)$ va dépendre de $T(t, x)$, $T(t, x - h)$, etc, mais ne possède plus de différentielle. Notons également $T_1(t) = \frac{\partial T}{\partial t}, \dots, T_{n-1}(t) = \frac{\partial^{n-1} T}{\partial t^{n-1}}$. Alors l'équation 2.8 devient :

$$\frac{\partial^n T}{\partial t^n} = \varphi (T_{n-1}(t), \dots, T_1(t), T(t), f_1(t), f_2(t), \dots, f_p(t)) \quad (2.9)$$

et se transforme en le système suivant :

$$\begin{cases} \frac{\partial T}{\partial t} & = T_1 \\ \frac{\partial T_1}{\partial t} & = T_2 \\ \frac{\partial T_2}{\partial t} & = T_3 \\ & \vdots \\ \frac{\partial T_{n-2}}{\partial t} & = T_{n-1} \\ \frac{\partial T_{n-1}}{\partial t} & = \varphi (T_{n-1}(t), \dots, T_1(t), T(t), f_1(t), f_2(t), \dots, f_p(t)) \end{cases} \quad (2.10)$$

qu'il est possible d'intégrer localement au sein d'une cellule avant d'envoyer la valeur de T aux cellules voisines. La variété des équations différentielles intégrables est donc très importante.

Initialisation des données

Position du problème Replaçons nous dans le cas simple de la diffusion sur la barre. Il est clair que s'il faut gérer une fonction $T(t, x)$ qui varie beaucoup selon x , il faudra davantage de cases pour supporter cette richesse que pour une fonction constante en x . Le même problème se posera lors des perturbations du système (cf. paragraphe 2.4.3).

Comment quantifier le nombre de cases nécessaires afin d'obtenir une précision acceptable? Voici une piste possible : localement, dans l'équation de diffusion (2.5) la discrétisation spatiale intervient uniquement dans le calcul de l'approximation de : $\frac{\partial^2 T}{\partial x^2}$. Une condition «nécessaire» au bon fonctionnement de la simulation est donc que cette approximation soit suffisamment précise. Supposons pour la suite que l'écart entre deux cases soit constant, et notons cette valeur h . Alors d'après la formule de Taylor-Lagrange (pour $f \in \mathcal{D}^{n+1}$) il existe un réel $c \in]a; a+h[$ tel que :

$$f(a+h) = f(a) + f'(a)h + \frac{f''(a)h^2}{2!} + \frac{f^{(3)}(a)h^3}{3!} + \frac{f^{(4)}(c)h^4}{4!} \quad (2.11)$$

et de même il existe $d \in]a-h; a[$ tel que :

$$f(a-h) = f(a) - f'(a)h + \frac{f''(a)h^2}{2!} - \frac{f^{(3)}(a)h^3}{3!} + \frac{f^{(4)}(d)h^4}{4!} \quad (2.12)$$

En additionnant 2.11 et 2.12, on obtient :

$$f(a+h) + f(a-h) = 2f(a) + f''(a)h^2 + \frac{f^{(4)}(c)h^4}{4!} + \frac{f^{(4)}(d)h^4}{4!} \quad (2.13)$$

Notons $\hat{f}''(a)$ l'approximation de la dérivée seconde utilisée dans nos calculs, on a alors :

$$\hat{f}''(a) = \frac{f(a+h) + f(a-h) - 2f(a)}{h^2} \quad (2.14)$$

Ainsi d'après 2.13 et 2.14, il apparaît que :

$$|\hat{f}''(a) - f''(a)| = \left| \frac{f^{(4)}(c)h^2}{4!} + \frac{f^{(4)}(d)h^2}{4!} \right| \leq \frac{2h^2}{4!} \sup_{]a-h; a+h[} |f^{(4)}| \quad (2.15)$$

d'où :

$$|\hat{f}''(a) - f''(a)| \leq \frac{h^2}{12} \sup_{[0;1]} |f^{(4)}| \quad (2.16)$$

Nous ne connaissons pas la fonction $T(t, x)$, mais on peut déjà appliquer ce raisonnement sur la fonction d'initialisation.

Un exemple de calcul Considérons l'exemple **heat1** qui sera présenté dans le paragraphe suivant 2.4.2.0. On a $T(0, x) = \sin(\pi x)$. Appliquons la formule 2.16 à $f(x) = T(0, x) : f^{(4)}(x) = \pi^4 \sin(\pi x)$, donc $\sup_{[0;1]} |f^{(4)}| = \pi^4$ (atteint en $\frac{\pi}{2}$). D'après l'équation 2.5 et l'inégalité 2.16, pour avoir une précision de 10^{-4} (par exemple) sur $\frac{\partial T(t, x)}{\partial t}$, il suffit d'avoir $K \times \frac{h^2}{12} \times \pi^4 \leq 10^{-4}$. Avec $K = \frac{1}{\pi^2}$, on obtient $h \leq \sqrt{\frac{12 \times 10^{-4}}{\pi^2}}$. Ce qui donne un total d'au moins 92 cases pour assurer cette précision.

On peut faire le même genre de raisonnement pour des dérivées autres que la dérivée seconde et ainsi, si l'équation différentielle le permet, obtenir une minoration du nombre de cases en fonction de la précision voulue.

TABLE 2.4 – Test **heat1** avec la méthode explicite

Time step	$T = 10s$			$T = 30s$		
	temps	précision	itérations	temps	précision	itérations
0.001	< 1s	0.00089	480 000	2s	0.00089	1 440 000
0.0005	1s	0.00046	960 000	4s	0.00046	2 880 000
0.0001	7s	0.00012	4 800 000	20s	0.00012	14 400 000
0.00005	13s	0.00013	9 600 000	40s	0.00013	28 800 000

TABLE 2.5 – Test **heat1** avec QSS1

Précision	$T = 10s$			$T = 30s$		
	temps	précision	itérations	temps	précision	itérations
0.1	< 1s	0.095	6213	< 1s	0.095	6212
0.05	< 1s	0.05	9233	< 1s	0.050	9232
0.01	< 1s	0.0098	16 567	< 1s	0.0098	16 566
0.005	< 1s	0.0053	20 463	< 1s	0.0053	20 462
0.001	1s	0.0010	45 565	1s	0.0010	45 564
0.0005	1s	0.00052	76 771	1s	0.00052	76 770
0.0001	4s	0.00019	326 147	4s	0.00019	326 146
0.00005	7s	0.00016	637 452	8s	0.00016	637 884

2.4.2 Test de la méthode

heat1

Nous avons réalisé ce premier test avec l'équation de la diffusion (cf. 2.5) et les paramètres suivants :

- une barre graduée de 0 à 1 ;
- un coefficient de diffusion de : $K = \frac{1}{\pi^2}$;
- des valeurs limites aux bords fixées à 0 : $T(t, 0) = T(t, 1) = 0 \forall t$;
- une fonction d'initialisation à $t = 0$ égale à $T(0, x) = \sin(\pi x)$.

Ainsi initialisé, la solution exacte du système est : $T(t, x) = e^{-t} \sin(\pi x)$. Pour la discrétisation spatiale, la barre est découpée en 50 cases, les cases aux extrémités étant fixées à une température de 0. La méthode explicite (en faisant varier le pas de temps) et les méthodes QSS1 et QSS2 (en faisant varier la précision de quantification) ont été testées. Les résultats obtenus sont présentés dans les tableaux 2.4, 2.5 et 2.6. T correspond au temps simulé, le temps est la durée d'exécution, la précision donne la plus grande erreur obtenue sur la barre et sur la durée de l'expérience (les relevés étant réalisés toutes les 0.1 secondes) et enfin les itérations comptent le nombre d'itérations pour toutes les cases, celles-ci correspondant à un cycle complet de calcul pour une case. On peut préciser ici que la méthode explicite calcule à chaque itération les valeurs pour l'ensemble des cases de la barre. Afin de comparer les résultats, le nombre d'itérations de la méthode explicite est multiplié par le nombre de cases. Ceci explique le nombre important d'itérations par rapport à QSS1 avec, malgré tout, un temps d'exécution relativement comparable.

TABLE 2.6 – Test **heat1** avec QSS2

Précision	$T = 10s$			$T = 30s$		
	temps	précision	itérations	temps	précision	itérations
0.1	2s	0.13	201 080	5s	0.13	621 627
0.05	2s	0.061	202 560	5s	0.061	623 578
0.01	2s	0.013	206 961	5s	0.013	627 057
0.005	2s	0.006	207 933	5s	0.006	628 504
0.001	2s	0.0012	209 107	5s	0.0012	630 084
0.0005	2s	0.00064	209 478	5s	0.00064	629 039
0.0001	2s	0.00023	209 362	5s	0.00023	628 794
0.00005	2s	0.00017	209 926	5s	0.00017	630 505

TABLE 2.7 – Test **heat2** avec QSS1

Précision	$T = 10s$			$T = 30s$		
	temps	précision	itérations	temps	précision	itérations
0.1	< 1s	0.095	6213	< 1s	0.095	6212
0.05	< 1s	0.05	9233	< 1s	0.050	9232
0.01	< 1s	0.0098	16 567	< 1s	0.0098	16 566
0.005	< 1s	0.0053	20 463	< 1s	0.0053	20 462
0.001	1s	0.0010	45 565	1s	0.0010	45 564
0.0005	1s	0.00052	76 771	1s	0.00052	76 770
0.0001	4s	0.00019	326 147	4s	0.00019	326 146
0.00005	7s	0.00016	637 452	8s	0.00016	637 884

heat2

Pour réaliser ce second test, nous avons repris exactement les mêmes paramètres excepté :

- des valeurs limites aux bords fixées à 0 et 1 : $T(t, 0) = 0$ et $T(t, 1) = 1 \forall t$;
- une fonction d'initialisation à $t = 0$ égale à $T(0, x) = x + \sin(\pi x)$.

Ainsi initialisé, la solution exacte du système est : $T(t, x) = x + e^{-t}\sin(\pi x)$.

Conclusion

Retenons d'abord qu'il a été nécessaire de démarrer les tests sur la méthode explicite à partir d'un pas de temps de 0.001. En effet, la méthode est stable dès que l'on vérifie $K \frac{\Delta t}{(\Delta x)^2} < \frac{1}{2}$ (ce qui donne ici $\Delta t < 0.002$). Par contre QSS1 et QSS2 semblent stables pour toutes les précisions. Nous avons notamment testé QSS1 sur d'autres problèmes avec toujours le même succès en ce qui concerne la stabilité.

Ensuite, on peut remarquer que QSS1 et QSS2 génèrent nettement moins d'itérations que la méthode classique, et la différence devient écrasante pour $T = 30s$. Ceci s'explique par le maximum de la fonction $T(x)$ qui vaut $Max(t) = e^{-t}$ et pour, par exemple, $t = 15s$ $Max(15) \simeq 3.06 \times 10^{-7}$; les méthodes QSS produisant une itération après chaque franchissement d'une barrière de quantification, elles ne sont plus nécessaires à partir d'un certain t donné.

Remarquons également que la précision des méthodes QSS1 et QSS2 est de l'ordre de grandeur de la précision choisie, et enfin, que QSS2 semble avoir davantage de difficultés vis à vis de QSS1 pour adapter sa charge de calcul, son nombre d'itérations étant presque constant.

2.4.3 Gestion des perturbations

Contexte

Supposons que l'on désire réaliser une simulation de feu de forêt avec intervention des pompiers. La propagation du feu peut être modélisée [Muzy *et al.*, 2002] par une équation différentielle spatialisée gérant la température (cf équations 2.17). La solution peut donc être calculée par une méthode type QSS. Mais comme on le constate sur les équations deux types de perturbations peuvent survenir :

- une perturbation ponctuelle, qui serait pour notre exemple l'intervention d'un pompier qui fait brutalement diminuer la température d'une case ;
- une perturbation par l'ajout d'une fonction (ici $Q \frac{\partial \sigma_v}{\partial t}$) au sein même de l'équation différentielle.

$$\left\{ \begin{array}{l} \frac{\partial T}{\partial t} = -k(T - T_a) + K\Delta T - Q \frac{\partial \sigma_v}{\partial t} \\ \text{avec :} \\ \frac{\partial \sigma_v}{\partial t} = 0 \text{ pour une cellule inerte} \\ \frac{\partial \sigma_v}{\partial t} = -\alpha \sigma_v \text{ pour une cellule en feu} \end{array} \right. \quad (2.17)$$

où ΔT note le Laplacien de T .

Les perturbations ponctuelles

Plaçons-nous dans le cas de QSS1 : la perturbation peut prendre la forme d'un événement externe à une case, lui spécifiant de prendre telle valeur ou encore de diminuer de telle autre, etc. Tous les cas envisageables sont pour la case une seule opération : elle doit modifier sa valeur. La différence avec une fonction de transition interne ou externe est que la valeur calculée change arbitrairement et non via l'incrément ou la décrémentation de i . Il convient donc dans l'ordre de :

- recalculer la pente avec la nouvelle valeur ;
- mettre à jour le moteur QSS1 par l'appel de la fonction `reset` (cf 2.1) qui consiste essentiellement à recalculer i , puis σ en accord avec la nouvelle pente ;
- générer une fonction de transition externe chez les voisins, pour leur signaler la nouvelle valeur.

La figure 2.10 est un exemple basé sur la diffusion, perturbée aléatoirement toutes les secondes par un ajout sur une case.

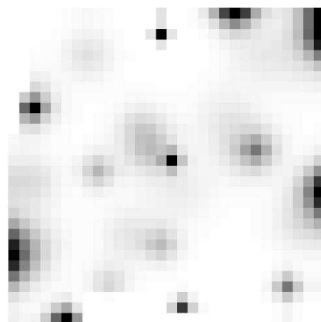


FIGURE 2.10 – Gestion des perturbations sous QSS1

Pour terminer cette partie, nous signalerons également qu'il ne faut pas oublier de respecter le modèle et la discrétisation spatiale lors des perturbations. En effet, sur notre exemple, la température est modélisée par une fonction T continue en t , x et y , dérivable par rapport au temps, etc. Il convient donc de la modifier en une autre fonction \hat{T} (résultante de l'addition de T avec la perturbation) possédant les mêmes propriétés. Modifier une seule case peut donc conduire à des aberrations. Une solution à ce problème peut être que le pompier réalise des perturbations gaussiennes centrées au point d'action et propager la perturbation sur les cases environnantes. En effet, si la perturbation est d'amplitude $f(d) = Ke^{-\frac{d^2}{\sigma^2}}$ avec d la distance à l'épicentre de la perturbation et que l'on utilise QSS1 pour l'intégration, pour trouver le nombre de cases à perturber autour de l'épicentre, il faut trouver toutes les cases à une distance telle que $|f(d)| \leq \frac{\Delta_q}{2}$, ce qui nous donne toutes les cases à une distance de $d \leq \sigma\sqrt{\frac{2K}{\Delta_q}}$. La discrétisation spatiale doit être également en cohérence avec l'écart-type de la gaussienne. Un écart-type petit impliquera la nécessité d'avoir réalisé un découpage spatial en petites cases. Nous avons en fait les mêmes problèmes que pour la fonction d'initialisation (2.4.1.0.0) mais en plus grave car nous ne connaissons pas \hat{T} . Signalons simplement que $\max|f^{(4)}| = \frac{12K}{\sigma^4}$ peut prendre rapidement des valeurs très significatives avec des petites valeurs de σ .

Cas des fonctions au sein de l'équation différentielle

Considérons le cas à une dimension pour simplifier notre propos et appelons g la fonction externe de perturbation. La fonction fait donc partie intégrante de l'équation différentielle. Après la discrétisation spatiale le système devient :

$$\frac{\partial T_i(t)}{\partial t} = f(T_i(t), g(x_i, t)) \text{ , pour } i \text{ allant de } 1 \text{ à } N \quad (2.18)$$

Et on se retrouve avec une fonction $g(x_i, t)$ qui dépend du temps. Il faut donc la quantifier en accord avec la méthode choisie (QSS1 ou QSS2) puis prévenir l'intégrateur QSS d'un changement de palier, comme pour les autres cases, cf figure 2.11.

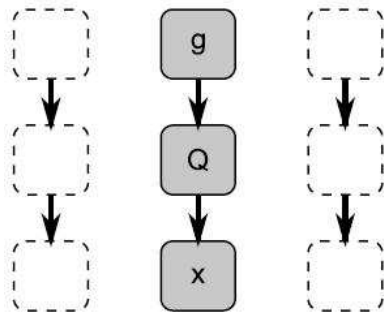


FIGURE 2.11 – Gestion d'une fonction g externe au système. Q est le filtre qui quantifie la fonction g , pour prévenir la case x qu'en cas de changement de palier

2.5 Conclusion

Dans ce chapitre, nous avons présenté deux nouveaux schémas d'intégration : QSS1 et QSS2. Ces méthodes, outre le fait d'être complètement adaptées au formalisme DEVS, sont

particulièrement performantes et, en tout cas, plus performantes que les approches classiques : Euler ou Runge-Kutta. Enfin, nous avons montré comment on pouvait venir perturber ce schéma d'intégration de manière discrète.

Ces méthodes ont été complètement appliquées dans la partie DYNFISH 2 (voir la section 3.4).

```
1 void Moteur_Qss1::reset(double xb, double penteb, double heureb)
2 {
3     x=xb;
4     pente=penteb;
5     heure=heureb;
6
7     /* Calcul de i */
8     i=(Zint) floor(x/precision);
9
10    /* calcul de sigma */
11    if(pente==0)
12        {
13            sigma=infinity;
14            return;
15        }
16
17    if(pente>0)
18        sigma=(d(i+1)-x)/pente;
19    else
20        sigma=((d(i)-x)-precision)/pente;
21 }
```

Listing 2.1 – Fonction gérant les perturbations au sein de QSS1

Chapitre 3

Applications

Sommaire

3.1	Introduction	46
3.2	Un modèle de dynamique spatiale : MUFINS (MULTI Fish INdian ocean Simulator)	46
3.2.1	Le cahier des charges du modèle	46
3.2.2	Gestion des données	47
3.2.3	Le système conatif, la mise en œuvre des comportements	47
3.2.4	Quelques simulations avec MUFINS	53
3.3	Un modèle exploratoire : une pêcherie théorique à métiers multiples – Modèle DYNFISH 1	58
3.3.1	Problématique	58
3.3.2	D’un modèle théorique	58
3.3.3	... À une modélisation multi-agents	61
3.3.4	... À la simulation	63
3.3.5	Application à la pêcherie d’Algue Française – Modèle SEAWEED	73
3.4	Une extension du modèle : DYNFISH 2	77
3.4.1	Présentation du modèle	77
3.4.2	Implémentation dans VLE	79
3.4.3	Les résultats de la simulation	80
3.5	Le modèle CHALOUPE	80
3.5.1	Présentation du projet Chaloupe (Projet Biodiversité ANR-IFB 2005) et objectifs	80
3.5.2	Positionnement de DYNFISH 3 dans CHALOUPE	84
3.5.3	Présentation du modèle bio-économique DynFish 3	85
3.6	Le modèle ECOMERISTEM	93
3.6.1	Présentation du modèle	93
3.6.2	Exemple : impact du stress hydrique sur la taille des organes	94
3.7	Conclusion	96

3.1 Introduction

Dans ce chapitre nous allons présenter l'application des techniques définies et présentées dans les chapitres 1 et 2. Cette partie applicative me paraît tout particulièrement importante car elle montre le bien fondé et l'opérationnalité de nos travaux. Enfin, elle prend tout son sens dans la mesure où elle permet de faire avancer la recherche dans d'autres thématiques que la nôtre.

Ce chapitre va être composé de 5 sections : la première section présente une application multi-agents qui vise à simuler la trajectoire des bancs d'espadons dans la zone Sud de l'Océan Indien à base d'image satellite ; ce projet a été financé par la Région Réunion, l'IRD Réunion et l'IFREMER Réunion. La deuxième concerne le projet DYNFISH 1 dont le but est la modélisation et la simulation multi-agents de la dynamique de pêcherie à l'aide de la plate-forme CORMAS [CORMAS, 2003] ; ce projet a été réalisé dans le cadre de mon stage post-doctoral réalisé à l'IFREMER. Ce modèle théorique est appliqué à la gestion de la pêcherie d'algues en Bretagne. La troisième section reprend le modèle DYNFISH 1 tout en utilisant la méthode QSS avec les agents. La quatrième section présente le travail réalisé dans le cadre du projet ANR Biodiversité CHALOUPE coordonné par l'IFREMER et dont j'avais la responsabilité de la tâche modélisation. Ce projet vise à représenter la dynamique des pêcheries dans le Golfe de Gascogne. Enfin, je présente le modèle EcoMeristem qui est un modèle structure fonction de croissance de riz, sorgho et canne à sucre.

3.2 Un modèle de dynamique spatiale : MUFINS (Multi Fish INdian ocean Simulator)

Cette partie est issue des travaux de thèse de D.Guyomard [Guyomard, 2005]. Ceux-ci ont été financés par le Conseil Régional de La Réunion, l'IFREMER Réunion et l'IRD Réunion dans le cadre du Programme Palangre Réunion (PPR). D. Guyomard a utilisé et mis en application la représentation des agents proposée dans ma thèse [Soulié, 2001].

3.2.1 Le cahier des charges du modèle

Le modèle proposé repose sur quatre hypothèses fondamentales :

- L'information environnementale est perçue de manière directe par les poissons, au niveau individuel local ;
- C'est cette information qui détermine leurs déplacements à méso-échelle, sous différentes contraintes éco-physiologiques (système de contrôle du comportement) ;
- La fonctionnalité de l'environnement (tel que perçu par les pêcheurs) « émerge » de l'ensemble de ces comportements individuels ;
- La pêche apporte les éléments de validation et la connaissance globale de l'environnement permet de caractériser le paysage environnemental comme habitat privilégié de l'espèce.

Ce modèle repose sur une plate-forme de simulation de type multi-agents pour la simulation du comportement de grands pélagiques (ici l'espadon) et de l'exploitation halieutique (ici la pêche palangrière réunionnaise) au sein d'un environnement multi-paramètres (SIG dynamique

ici appliqué aux paramètres de l'environnement océanique de surface captés par satellite). Cette approche se situe donc résolument à la croisée des disciplines suivantes

- La modélisation individus-centrée : des animats espadons se déplacent dans un espace virtuel selon des hypothèses éthologiques et écologiques ;
- L'écologie du paysage : l'environnement est représenté par différents paramètres dont on cherche à comprendre les relations horizontales à travers le comportement du poisson ;
- Les systèmes multi-agents : des agents « espadons » et des agents « lignes de pêche » ;
- La Vie Artificielle : modélisation de phénomènes « vivants » et émergence d'objets océaniques fonctionnellement cohérents à partir d'hypothèses sur le comportement d'agents poissons.

3.2.2 Gestion des données

Les données satellites

La gestion des données satellitales dans MUFINS a été établie selon l'approche développée par M.Desruisseaux [Desruisseaux *et al.*, 2001]. La base de données **seas** contient les informations sur les images satellitales. Elle est structurée de manière à pouvoir manipuler facilement les cartes disponibles dans le cadre du programme PPR : elle permet de disposer des informations sur chaque image (type de données, résolution spatiale, système de coordonnées, unités, plage temporelle, type de codage pour l'affichage) et de faciliter la sélection des images correspondant à une date et un type donnés. Au sein de MUFINS, des requêtes de type SQL sont passées à la base **seas**, qui renvoie les informations nécessaires pour manipuler une carte satellitale (date, chemin d'accès sur le disque de l'ordinateur ou par le réseau, système d'affichage de l'image notamment).

Les données « pêche »

De la même manière, les données de pêche, notamment les positions des filages de la pêche-réunionnaise de 1998 à 2000, ont été stockées sous MySQL dans une base appelée **fishery**. MUFINS peut y accéder à chaque fois que le système a besoin de créer des agents lignes de pêche aux bonnes positions pour la date spécifiée.

3.2.3 Le système conatif, la mise en œuvre des comportements

Animats et comportements

Le système conatif reçoit des informations typées, encapsulant la valeur au pixel où se situe l'agent et ses coordonnées géographiques en longitude/latitude. Certains comportements peuvent s'en accommoder comme source d'information avant de mettre en œuvre un comportement et envoyer une commande. Mais surtout, le transfert de cette information « marque » le début du processus décisionnel du système conatif dans le système de gestion du temps. Après réception de l'information, le système conatif délibère et renvoie la position de destination aux instances des environnements réels par l'intermédiaire de l'environnement virtuel.

Les comportements peuvent récupérer l'ensemble des valeurs dans une zone située autour du point de départ, d'autres des valeurs captées par l'agent précédemment (mémoire de l'agent).

A l'instar de R. Humston [Humston *et al.*, 2000], nous avons choisi de coder les comportements des espadons selon des principes simples de *kinésie*. *L'hypothèse principale est que chaque individu modifie les paramètres de son trajet en fonction des stimuli du milieu*. En reprenant les conclusions de S. Benhamou et P. Bovet [Benhamou et Bovet, 1991] et les règles de comportement des poissons proposées par D.L. De Angelis et G.T. Yeh [De Angelis et Yeh, 1984], on peut très simplement considérer que **les poissons diminuent leur vitesse de nage et augmentent la sinuosité de leur parcours** (ils effectuent des changements de direction plus fréquents) **lorsque les conditions du milieu leur sont favorables**. Inversement, ils auront tendance à nager plus vite et « droit devant » pour quitter une zone moins favorable.

Lors des simulations avec MUFINS, les valeurs d'*entrée* du système de contrôle de chaque animat sont les valeurs des *variables environnementales* captées par cet animat au pas de temps t . Les valeurs de *sortie* sont les valeurs des *variables de trajectométrie* (vitesse et/ou changement de direction) appliquée au mouvement de l'animat au pas de temps suivant $t + 1$.

Dans le cas de notre application à l'espadon, les conditions favorables sont représentées par les valeurs des différentes variables environnementales qui influencent positivement les captures dans les résultats des GAM. Toutefois, les valeurs pour lesquelles les kinésies se mettent en œuvre et les modifications des paramètres de trajectoire (vitesse et orientation) ne sont probablement pas nettement définies à chaque instant.

Afin de prendre en compte ces incertitudes et limiter l'influence des nombreux paramètres d'entrée du modèle, nous avons opté pour un codage des comportements à partir des règles de la *logique floue* [Zadeh *et al.*, 1996]. Le flou concerne en effet les informations pour lesquelles les limites ne sont pas clairement définies. Contrairement à la théorie classique des séries où l'on a affaire à des objets dont l'appartenance peut être clairement décrite, dans la théorie de la logique floue, l'appartenance d'un objet à une série peut être partielle, c'est à dire qu'un élément appartient à une série selon un certain niveau (probabilité) d'appartenance.

Le système de contrôle du comportement des animats a été codé entièrement selon les principes de la logique floue. On a ainsi défini les entrées du système : les niveaux d'appartenance des variables environnementales sont codés sous forme de séries floues, correspondant aux intervalles de valeurs entre lesquelles les conditions du milieu seront considérées comme plus ou moins favorables pour les animats à chaque pas de temps. Les limites sont établies à partir des points d'inflexion des courbes issues des résultats graphiques des modèles GAM présentés dans la première partie de ce travail : dans chaque intervalle ainsi défini, une règle différente sera appliquée. La définition des limites est donc importante, mais en même temps, la logique floue autorise une certaine imprécision autour de ces limites sans que le comportement général du modèle en soit *a priori* affecté. Les sorties du système de contrôle du comportement (vitesse et orientation des animats à chaque pas de temps) doivent aussi être codées sous forme de séries floues.

Spécification des valeurs floues associées aux variables conclusion de trajectométrie

Les variables conclusion de trajectométrie (vitesse et variations de direction) sont modifiées par le système de contrôle du comportement, selon que le comportement soit de type orthokinétique ou klinokinétique.

Vitesse (orthokinésie)

Les vitesses de déplacements des animats ont été spécifiées à partir de la distribution des vitesses horaires observées par Carey et Robison [Carey et Robinson, 1981] et Carey [Carey, 1990]. De la même manière que pour les variables environnementales, la vitesse doit être représentée sous forme de termes flous et niveaux d'appartenance associés. Quatre termes ont été définis, dont les noms (**Low**, **Med**, **Hig** et **VLow**) ne correspondent pas aux valeurs de vitesse mais au contraire, à l'attraction supposée du poisson pour la zone considérée (vitesse faible = attraction forte)

1. **Hig** : vitesse faible, correspondant à une phase d'alimentation et de faible activité de nage liée à la satisfaction de besoins physiologiques (variable floue centrée sur 1 km/h) ;
2. **Med** : vitesse moyenne, correspondant à une phase de recherche de nourriture dans une zone plutôt favorable (centrée sur 3 km/h) ;
3. **Low** : vitesse élevée, correspondant à une phase active de déplacement vers une zone plus favorable, notamment au sein de courants marins (vitesse centrée sur 6 km/h) ;
4. **VLow** : vitesse très élevée, correspondant à une phase de migration (centrée sur 9 km/h), beaucoup plus rarement atteinte (seules quelques règles de déclenchement permettent de déclencher cette conclusion).

La distribution des séries floues des 4 termes a été établie de manière relativement arbitraire : il s'agit de distributions centrées sur les valeurs remarquables des vitesses observées, avec un étalement identique pour chaque série, sans forcément de recoupement.

La figure 3.1 illustre les niveaux d'appartenance des termes flous associés à la vitesse.

Changements de direction (klinokinesie)

Les changements de direction des animats sont définis par rapport au déplacement immédiatement précédent : une valeur de -10° correspond à une faible ré-orientation de sa nage vers la droite, une valeur de 90° correspond à un quart de tour vers la gauche. L'univers du discours de la variable s'étend donc en théorie de -180° à $+180^\circ$ mais par le principe même de défuzzification par pondération des aires, il est nécessaire de proposer des niveaux d'appartenance de la variable conclusion dont la distribution ne soit pas symétrique centrée sur 0. Pour tenir compte des changements de direction vers la droite ou vers la gauche, un facteur multiplicatif aléatoirement égal à 1 ou -1 a été appliqué à la sortie défuzzifiée. On a ainsi distingué 3 niveaux, dont les noms suivent la même terminologie que pour les vitesses

1. **Hig** : brusque changement de direction, correspondant à une forte sinuosité de parcours dans une zone plutôt favorable, avec retours successifs autour d'un point focal (centré sur 180°) ;
2. **Med** : changements de direction correspondant à une recherche active de nourriture ou de conditions favorables (centré sur 90°) ;

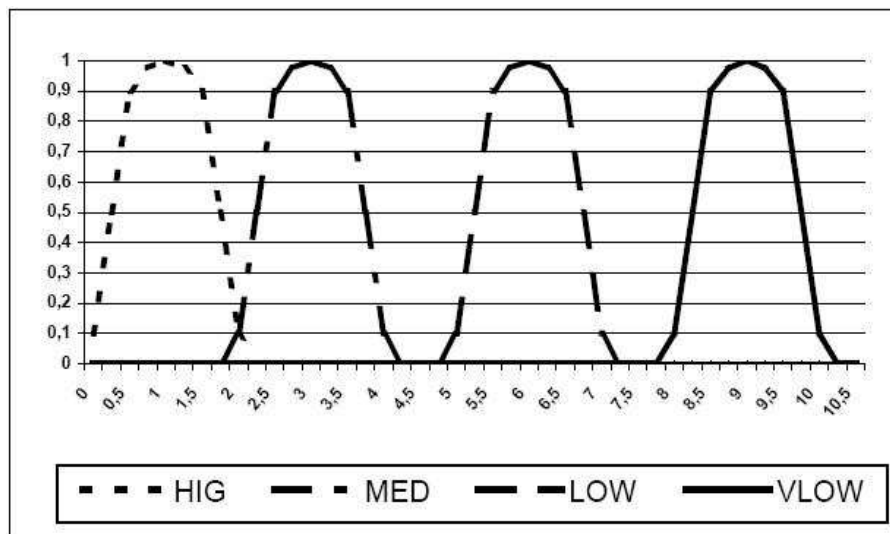


FIGURE 3.1 – Les niveaux d'appartenance des termes flous associés à la vitesse (abscisses vitesse en km.h⁻¹, ordonnées niveau d'appartenance)

3. **Low** : faibles changements de direction, l'animal a tendance à nager en ligne droite lors de déplacements orientés de type migratoire (centré sur 0°).

La figure 3.2 illustre les niveaux d'appartenance des valeurs floues associées aux changements de direction

Dans le cas de la klinokinésie, la vitesse des animats est fixée à 3 km.h⁻¹, ce qui correspond à la vitesse moyenne de déplacement d'un espadon d'après les observations de Carey et Robison [Carey et Robinson, 1981] et Carey [Carey, 1990].

La base des règles de comportement

Les règles de comportement ont été codées selon des principes les plus simples possibles. Le principe général est que la conclusion est « proportionnelle » aux antécédents. Pour un seul antécédent, les règles sont évidentes (Tableau 3.1). Pour plusieurs antécédents, on établit des « moyennes » (ex : **Low** et **Hig** donnent **Med**, voir Tableau 3.2) ou on privilégie les niveaux les plus extrêmes (ex : **Low** et **Med** donnent **Low** et non **Med**; **Med** et **Hig** donnent **Hig**, et non **Med**). Dans tous les cas, il est considéré que les influences respectives de tous les paramètres sont strictement égales.

Les règles de comportement sont codées selon le nombre de valeurs captées par l'animat et les variables environnementales spécifiées comme kinétiques dans le fichier de configuration de MUFINS. Les tableaux suivants illustrent les règles de comportement.

Dans le cas de l'orthokinésie, ces règles sont appliquées à la vitesse de déplacement des animats. Dans le cas de la klinokinésie, elles sont appliquées aux changements de direction. On a

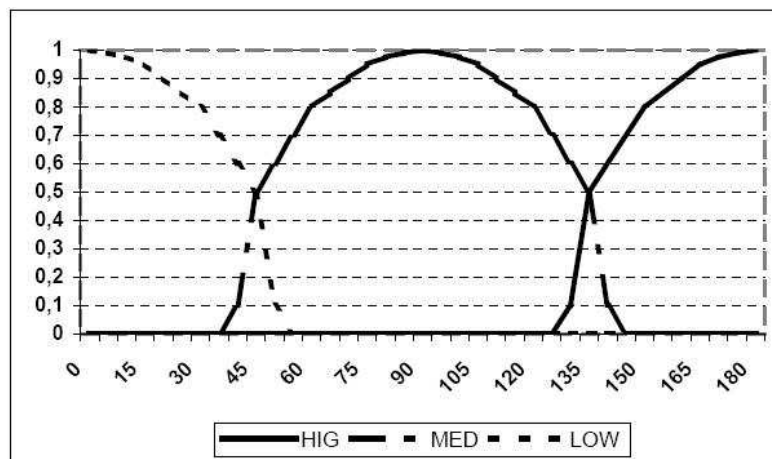


FIGURE 3.2 – Les niveaux d'appartenance des valeurs floues associées aux changements de direction (abscisses variation de direction en degrés, ordonnées : niveau d'appartenance)

TABLE 3.1 – Kinésie à 1 antécédent

Antécédent	Sortie
Low	Low
Med	Med
Hig	Hig

TABLE 3.2 – Kinésie à 2 antécédents

Antécédent 1	Antécédent 2	Sortie
Low	Low	Low/VLow
Med	Med	Med
Hig	Hig	Hig
Low	Med	Low
Low	Hig	Med
Med	Low	Low
Med	Hig	Hig
Hig	Low	Med
Hig	Med	Hig

aussi introduit un type de comportement mixte appelé orthoklinokinésie, qui combine ces règles de déclenchement à la fois sur la vitesse et les changements de direction.

Lignes de pêche et capture virtuelle

Le comportement des lignes de pêche est différent du comportement des animats : les lignes ne se déplacent pas en cours de simulation, mais elles interagissent avec les animats par le processus de *capture virtuelle*.

A tout instant de la simulation, les lignes de pêche peuvent percevoir les positions des animats au moyen de l'interface dédiée (connexion à la table `Positions` de la base `simulations`). Le processus de capture virtuelle est basé sur les distances géographiques entre les lignes et les animats : si un animat est proche d'une ligne de pêche, il est considéré comme « capturable ».

Toutefois, pour optimiser les performances du simulateur, plusieurs options méthodologiques ont été choisies. Ainsi, chaque ligne de pêche ne récupère pas la totalité des distances de tous les animats présents dans la société des agents : un périmètre de capture est défini, en considérant le rectangle défini par la ligne de pêche, élargi d'une distance correspondant grossièrement au déplacement possible des animats au cours de la durée de simulation pendant laquelle la capture est mise en œuvre (soit une demi-douzaine d'heures à environ 3.5 km/h). Ainsi, tous les animats présents dans ce périmètre élargi sont susceptibles d'être capturés par cette ligne.

La figure 3.3 illustre le schéma de sélection des animats.

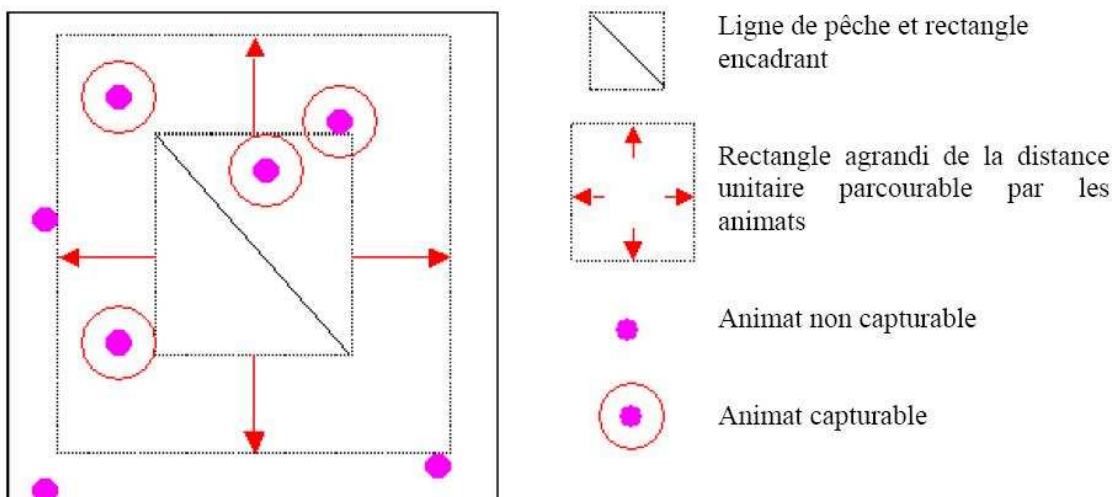


FIGURE 3.3 – Schéma de la sélection des animats « capturables » par les lignes de pêche

Une fois la liste des animats « capturables » disponible pour chaque ligne, deux types de processus de capture ont été codés.

Le premier type correspond à la capture caractérisée de statique. Dans ce cas, seule la distance de l'animat à la ligne conditionne la capture au pas de temps considéré. La distance de capture est spécifiée par l'utilisateur, en milles nautiques. Si un animat se trouve à une distance euclidienne inférieure à cette distance, il est alors capturé.

La figure 3.4 illustre le processus de capture statique.

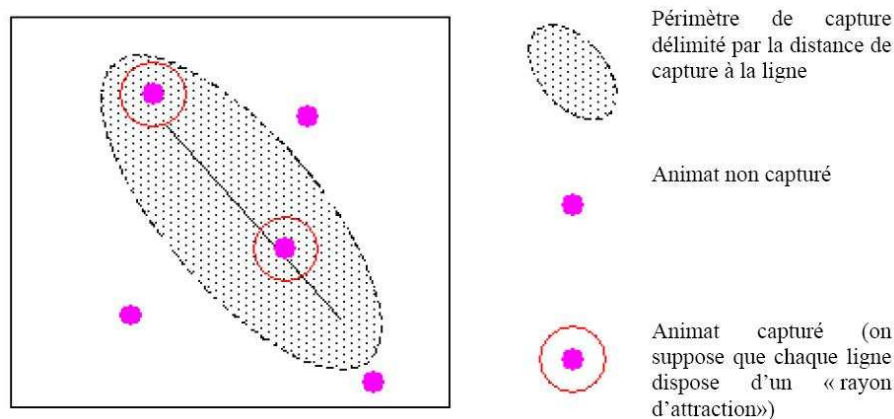


FIGURE 3.4 – Schéma du processus de capture statique

Le deuxième type de capture est appelé capture dynamique. Cette fois, ce n'est pas la distance de l'animat à la ligne à un instant donné qui détermine la capture. Il est ici nécessaire de disposer de la trajectoire de chaque agent entre l'instant précédent et l'instant considéré si cette trajectoire croise la ligne de pêche (intersection non nulle), L'animat est capturé. Il peut ensuite être enlevé du système selon la spécification de l'utilisateur.

La figure 3.5 illustre le processus de capture dynamique.

3.2.4 Quelques simulations avec MUFINS

Hypothèses comportementales et spécification des simulations

L'utilisation de MUFINS implique de spécifier un grand nombre de paramètres, que l'on peut regrouper en cinq catégories principales

- La spécification des comportements (ici codés selon les principes de la logique floue) ;
- Le choix des types de comportements mis en œuvre par les animats (parmi ceux codés) ;
- Les combinaisons de variables environnementales introduites ;
- Les zones et périodes de simulation ;

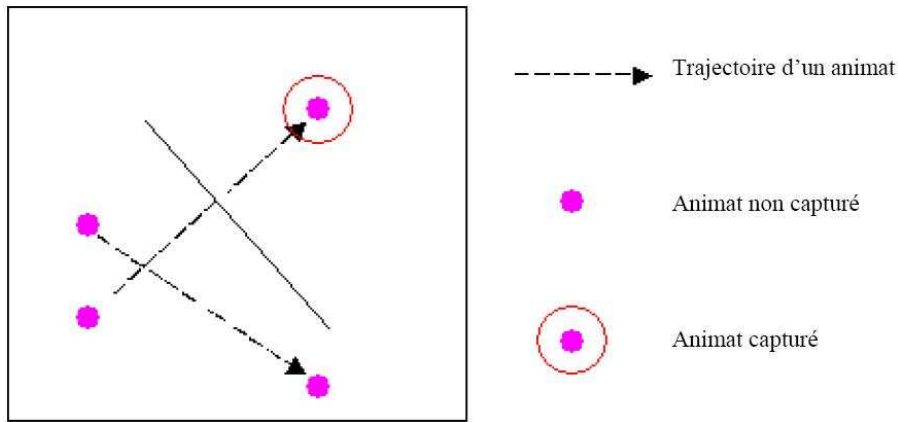


FIGURE 3.5 – Schéma du processus de capture dynamique

– La distribution initiale des animats.

Le choix du codage des comportements par la logique floue, que nous pouvons considérer comme une approche robuste du système de contrôle des comportements, nous permet de focaliser les analyses sur les grandes questions posées par les hypothèses issues de nos précédentes analyses. Toute la spécification des comportements (y compris les bases de règles de comportement) est basée sur ces hypothèses et sur les choix posés concernant le système de contrôle des comportements : elle sera conservée pour toutes nos simulations.

Concernant le choix des types de comportement, il nous a semblé important de tester séparément les deux types de kinésies : orthokinésie et klinokinésie. On a de plus défini un type mixte, appelé orthoklinokinésie, qui combine les effets des deux précédentes, à savoir une modification à la fois de la vitesse et de la fréquence de changements de direction des animats à chaque pas de temps de simulation. Ce type de comportement est censé apporter plus de réalisme aux déplacements des animats.

Parmi les facteurs limitant le choix des périodes de simulations figure dans un premier temps la disponibilité des cartes satellitaires et dans un deuxième temps, leur qualité etc. Certaines cartes satellitaires (notamment celles de SST) présentent en effet de nombreux artefacts, ce qui est rédhibitoire pour la simulation de comportements individuels basés sur cette information satellitaire. Du fait de la répartition très irrégulière de ces cartes de « qualité médiocre », seules des périodes relativement courtes ont pu être conservées.

Afin de tester notre hypothèse de déplacement des individus des zones équatoriales aux zones tropicales au moment de la saison de reproduction, basée sur la progression des champs thermiques de surface (variable SST), il a tout de même fallu sélectionner une période suffisamment longue pour pouvoir simuler des déplacements à échelle océanique. Nous avons ainsi sélectionné une plage temporelle de 93 jours, du 30 octobre 1998 au 30 janvier 1999, correspondant au début de l'été austral. Durant cette période, les températures moyennes de la zone seychelloise évoluent

de 27.8°C à 28.2°C, tandis que celles de la zone Réunion évoluent de 26.1°C à 28.2°C. L'évolution des champs thermiques de surface n'est toutefois pas régulière et la qualité des cartes n'est pas non plus idéale.

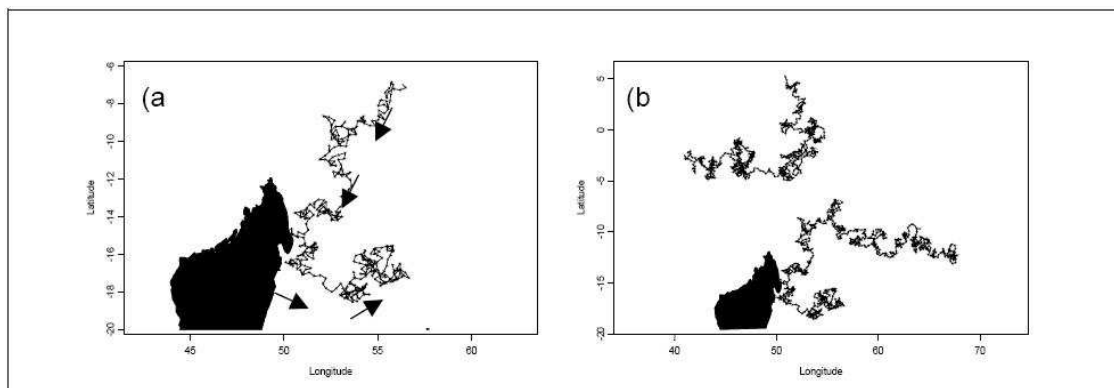


FIGURE 3.6 – Trajectoires de l'animat ayant atteint la latitude la plus australe (a) et trajectoires « extrêmes » (b) pour la même simulation

On remarque ainsi d'abord que très rares sont les animats ayant atteint les latitudes de la zone Réunion (au sud de 18°S) avant la fin de la simulation : seulement 4 animats (sur un total de 9000 animats) ont atteint la latitude 18°S, au cours de deux simulations. Ce sont les déplacements de type orthokinétiques et les simulations à 4 pas de temps journaliers qui ont permis à ces animats de tout juste atteindre les latitudes où nos hypothèses les prédisposaient à se retrouver au coeur de l'été austral. Ainsi, l'animat qui est arrivé le plus au sud a parcouru 11.6° de latitude vers le sud depuis sa position initiale (6.96°S). Toutefois, si on observe les autres trajectoires « extrêmes » pour cette simulation (trajectoires des animats ayant atteint les latitudes et longitudes maximales et minimales), on observe que ces déplacements sont très semblables à ceux de l'individu le plus austral (Figure 3.6). La latitude la plus boréale atteinte a d'ailleurs été observée pendant la même simulation.

Si on observe l'évolution des distances des animats par rapport à leur position d'origine, on remarque que en moyenne, les individus s'éloignent (Figure 3.7). La Figure 3.8 représente l'évolution de la distribution des 500 animats en cours de simulation. Les animats, initialement régulièrement distribués, s'écartent petit à petit de la zone originelle et s'agrègent en petits groupes, puis s'écartent de nouveau, sans qu'une dynamique particulière apparaissent véritablement. Aucune direction privilégiée ne semble se dégager des simulations, pour aucun des types de comportements testés.

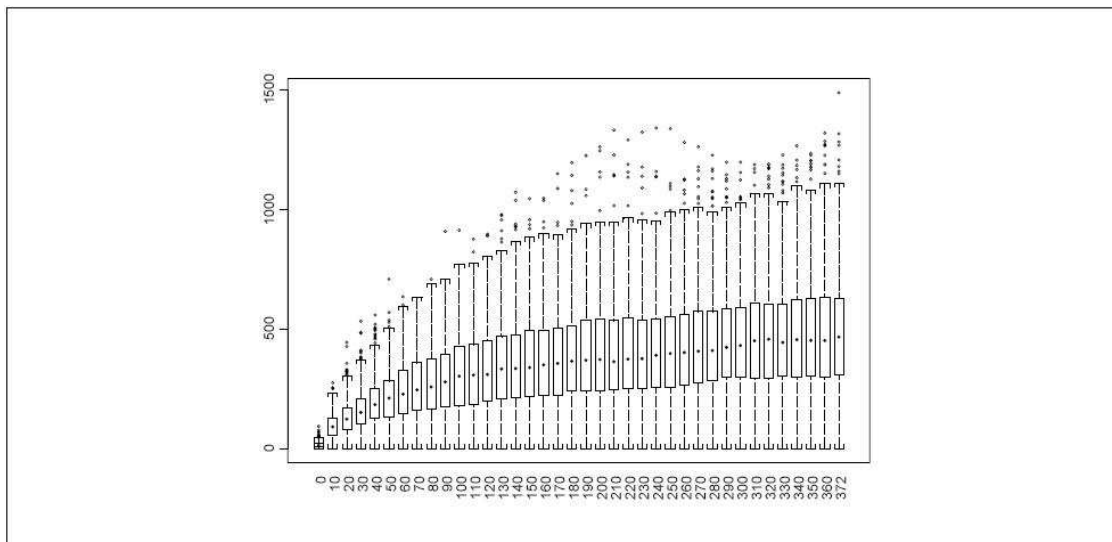


FIGURE 3.7 – Évolution de la distribution des distances à la position d'origine des animaux en cours de simulation (abscisses : pas de temps, ordonnées : km)

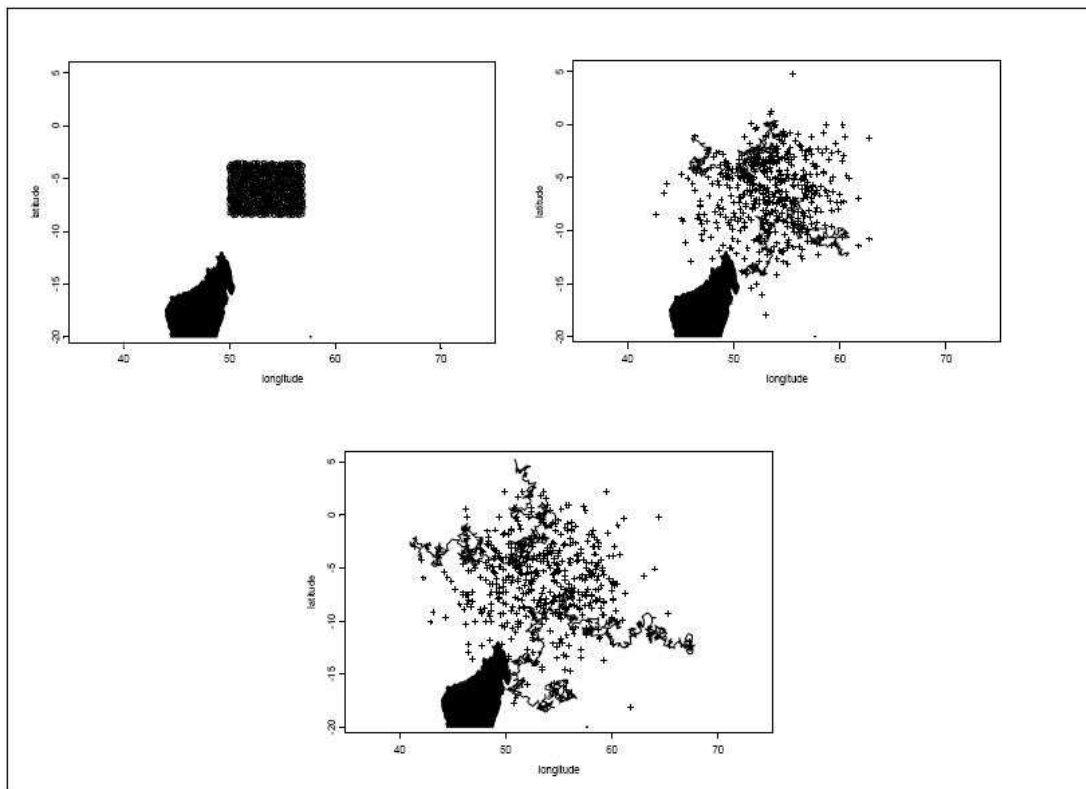


FIGURE 3.8 – Distribution des animaux aux pas de temps 0 (positions initiales régulièrement distribuées), 186 (positions intermédiaires) et 372 (positions finales); on a aussi représenté (en gras) les trajectoires « extrêmes »

Si on observe plus en détails la forme des trajectoires des animats, on remarque que chaque trajectoire est caractérisée par des patrons identiques : les animats se déplacent de manière relativement erratique, ils ont tendance à explorer certaines zones intensément sans parcourir de grandes distances, puis se mettent à franchir des distances plus grandes en ligne plus ou moins droite. Cette forme en « collier de perles » des trajectoires indique que les animats rencontrent des conditions favorables à certaines périodes des simulations puis s'en éloignent (en effectuant des brusques déplacements apparemment orientés, appelés « spurts » ou « emballements » d'après De Angelis et Yeh [De Angelis et Yeh, 1984]) avant d'y retourner. Quelque soit le type de comportement et le facteur d'échelle du système de contrôle du comportement, on observe ces mêmes schémas chez tous les animats.

3.3 Un modèle exploratoire : une pêcherie théorique à métiers multiples – Modèle DYNFISH 1

3.3.1 Problématique

La capacité des mesures de régulation des activités halieutiques à influencer favorablement sur les pratiques des entreprises constitue un enjeu majeur des politiques publiques mises en œuvre dans ce domaine. Or les effets économiques et écologiques à attendre de différentes mesures dépendent étroitement de la manière dont les entreprises sont susceptibles de réagir à ces mesures. Ainsi, dans le domaine des pêches, les effets concrets de mesures telles que les quotas de captures et de rejets, les mesures techniques visant à modifier les maillages et la sélectivité des engins ou à diminuer l'impact des engins sur les habitats, les interdictions d'activités de pêche dans certaines zones et/ou durant des périodes déterminées, ou le recours aux instruments économiques de gestion, dépendront notamment de phénomènes de reports d'effort de pêche, de changements de métiers, d'espèces cibles, . . . Il en va de même en cas de fluctuations de court terme dans les conditions environnementales (événements climatiques, pollutions). L'évaluation des conséquences économiques de ces processus suppose de mieux prendre en compte les possibilités d'adaptation des entreprises à de telles évolutions de court terme.

Une contribution importante de l'analyse économique est de développer des méthodes d'analyse et de représentation des comportements des entreprises et de leurs déterminants. Ces méthodes permettent d'évaluer les effets dynamiques des contraintes économiques, environnementales et réglementaires sur l'exploitation, et apportent ainsi des éléments pour l'évaluation des effets concrets à attendre de politiques publiques mises en œuvre dans ce domaine.

L'objectif de ce travail est de développer un modèle économique de court terme de la dynamique d'activités halieutiques, en s'appuyant sur les développements récents de la simulation multi-agents appliquée aux problèmes de gestion des ressources renouvelables.

3.3.2 D'un modèle théorique . . .

La pêcherie est composée d'une flotte de bateaux ciblant des espèces différentes et qui opèrent sur une aire A divisée en j zones. La ressource exploitée est composée d'un ensemble de i espèces, sans interactions biologiques. Chaque espèce est distribuée sur A en des stocks locaux et caractérisée par :

- Leur biomasse $X_{i,j}$;
- Un taux de croissance naturel intrinsèque $r_{i,j}$;
- Un coefficient de mobilité spatiale $d_{i,j,j'}$;
- Une capacité de charge $X_{i,j}^{max}$ pour chaque espèce dans chaque zone (*e.g* chaque stock).

Le concept de « métier » est introduit pour décrire l'ensemble des options de pêches disponibles pour la flotte. Un métier est défini comme le choix de cibler une espèce particulière dans une zone particulière, avec un niveau de productivité donné. Les notations suivantes sont utilisées pour décrire l'effort de pêche nominal affecté à chaque métier, à chaque pas de temps et ainsi que les paramètres techniques, de coûts et de prix :

- $E_{i,j}$: l'effort de pêche ciblant l'espèce i dans la zone j ;
- $q_{i,j}$: le coefficient de capturabilité pour l'espèce i dans la zone j ;
- $c_{i,j}$: le coût associé à l'espèce i dans la zone j ;
- p_i : le prix de chaque espèce i .

Dynamique de Stock

La dynamique du stock est modélisée comme suit :

$$X_{i,j}(t + \Delta t) = X_{i,j}(t) + \left(f_{i,j}(X_{i,j}(t)) X_{i,j}(t) + \sum_{j' \neq j} S_{i,j,j'}(t) - Y_{i,j}(t) \right) \Delta t \quad (3.1)$$

Où :

- $f_{i,j}(X_{i,j}(t))$ mesure la croissance instantanée par unité de biomasse pour l'espèce i dans la zone j ;
- $S_{i,j,j'}$ mesure la migration de la biomasse de l'espèce i de la zone j vers la zone j' ($j \neq j'$) ;
- $Y_{i,j}(t)$ mesure les captures d'une espèce i dans une zone j .

Pour les stocks de biomasse, une fonction logistique est utilisée :

$$f_{i,j}(X_{i,j}(t)) = r_{i,j} \left(1 - \frac{X_{i,j}(t)}{X_{i,j}^{max}} \right) \quad (3.2)$$

Les captures par unité d'effort dans chaque zone sont considérées comme directement proportionnelles entre l'effort de pêche $E_{i,j}$ et l'abondance de l'espèce ciblée i dans la zone j :

$$Y_{i,j}(t) = q_{i,j} E_{i,j}(t) X_{i,j}(t) \quad (3.3)$$

Le transfert net de biomasse entre la zone j et une zone adjacente j' (fonction de la connectivité considérée sur la grille A) est densité-dépendante :

$$S_{i,j,j'} = d_{i,j,j'} \left(\frac{X_{i,j'}(t)}{X_{i,j'}^{max}} - \frac{X_{i,j}(t)}{X_{i,j}^{max}} \right) \quad (3.4)$$

Où $d_{i,j,j'}$ représente le coefficient de migration de l'espèce i entre la zone j et la zone j' .

Il n'y a pas de interaction biologique (compétition, prédation, ...) entre les espèces dans ce modèle. Donc, chaque séquence de mortalité observée dans celui-ci est forcément dû une action de pêche.

Dynamique de l'Effort de Pêche

La représentation de la dynamique de l'effort de pêche est basée sur l'hypothèse que la flotte alloue son activité entre les métiers en fonction des marges sur coût variables anticipées associées à chaque métier. On peut noter que dans notre cas, nous avons choisi de modéliser l'allocation de l'effort dans le système au niveau du métier et sur la base d'une représentation agent des zones. Dans notre cas, par conséquent, le choix individuel des bateaux n'est pas représenté dans ce modèle.

La capacité de l'effort de pêche à se déplacer vers une autre zone et/ou espèce est décrite par les coefficients suivants :

- n : un coefficient de polyvalence décrivant la capacité d'une unité de pêche à se réallouer vers une autre espèce ;
- m : une coefficient de mobilité décrivant la capacité d'une unité de pêche à se réallouer vers une autre zone de pêche.

Un modèle d'anticipation adaptative avec information parfaite est utilisé pour calculer les marges anticipées et l'effort alloué. On considère ici que la flotte a un comportement myope : les marges anticipées par métier sont supposées être les mêmes que les marges observées au pas de temps précédent, corrigées par des coûts supplémentaires liés à la sélection d'une nouvelle espèce cible et/ou d'une nouvelle zone de pêche.

On définit donc les marges anticipées par métier comme suit :

$$M_{i,j,i',j'}^{exp}(t) = p_{i'} q_{i',j} X_{i',j'}(t) - [1 + \gamma_{j,j'}] c_{i'} \quad (3.5)$$

Où :

- $p_{i'}$ est le prix de l'espèce i' ;
- $q_{i',j}$ est la capturabilité de l'espèce i' dans la zone j ;
- $c_{i'}$ est le coût unitaire lié à la pêche de l'espèce i' ;
- γ est le surcoût lié au changement de zone et fonction de la distance entre les zones j et j' ($0 \leq \gamma_{j,j'} \leq 1$).

L'allocation de l'effort de pêche est basée sur les marges par unité d'effort et calculée comme suit :

$$g_{i,j}(t) = \frac{M_{i,j}^{exp}(t)}{E_{i,j}(t)} \quad (3.6)$$

Le changement d'un métier vers un autre de l'effort de pêche (*i.e* d'une combinaison espèce/zone à une autre) est calculé comme suit :

$$E_{i,j}(t + \Delta t) = E_{i,j}(t) + \left[\sum_{i \neq i'} n T_{i,j,i',j}(t) + \sum_{j \neq j'} m T_{i,j,i,j'}(t) + \sum_{i \neq i', j \neq j'} nm T_{i,j,i',j'}(t) \right] \Delta t \quad (3.7)$$

Avec :

$$T_{i,j,i',j'} = \begin{cases} -(g_{i,j}(t) - g_{i',j'}(t)) & \text{si } > 0 ; \\ 0 & \text{sinon.} \end{cases} \quad (3.8)$$

La figure 3.9 illustre ce mécanisme grâce à l'utilisation d'un arbre de décision.

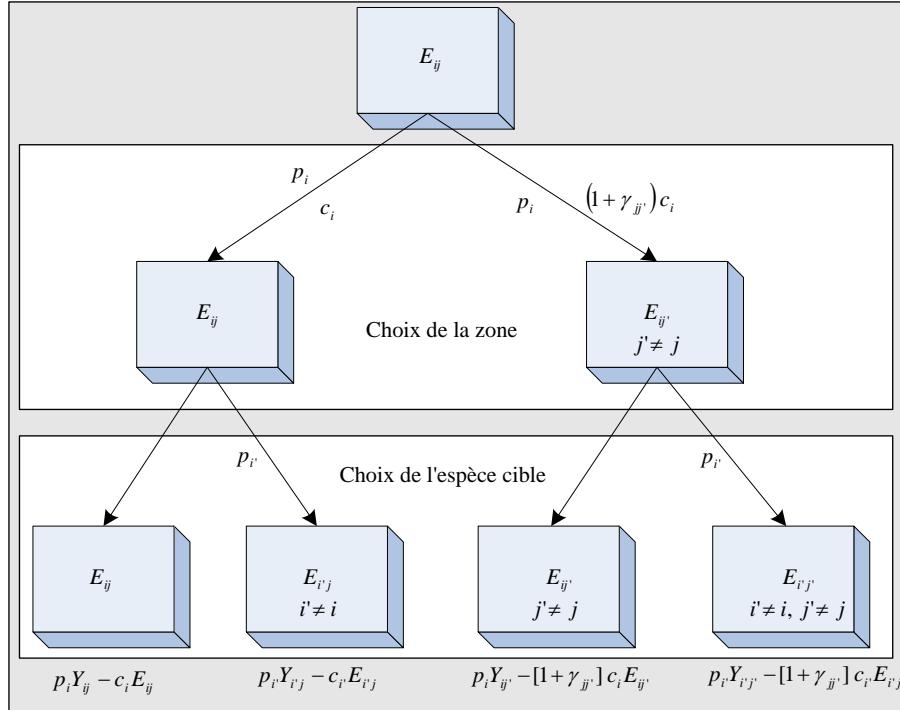


FIGURE 3.9 – Arbre de décision du domaine des possibles à chaque pas de temps

3.3.3 ... À une modélisation multi-agents ...

Afin de passer à la simulation multi-agents, nous avons utilisé l'environnement de programmation Cormas [Bousquet *et al.*, 1998] [CORMAS, 2003]. Il offre un cadre de développement de modèles de simulation des modes de coordination entre des individus et des groupes qui exploitent ces ressources en commun. Ce cadre se structure en trois modules :

- Un premier module permet de définir les entités du système à modéliser, que l'on appelle des agents informatiques, et leurs interactions. Ces interactions s'expriment par des procédures de communication directe (envois de messages), et/ou par le fait plus indirect de partager le même support spatial. Dans ce module, on utilise trois entités prédéfinies :
 - Les entités spatiales qui vont représenter l'environnement spatial de la simulation ;
 - Les entités sociales qui vont représenter les agents de la simulation ;
 - Les entités passives qui vont représenter les objets dont nous avons besoin pour mettre en place notre simulation, mais qui ne relèvent pas du domaine spatial et de l'agent.
- Le second module concerne le contrôle de la dynamique globale (ordonnancement des différents événements d'un pas de temps du modèle) ;
- Un troisième module permet de définir une observation de la simulation selon des points de vue. Cette fonctionnalité autorise l'intégration, dans le processus de modélisation, des

modes de représentation.

Nous allons nous attacher à présenter comment nous avons utilisé Cormas pour modéliser DYNFISH1. Afin de restituer le plus fidèlement possible le découpage des entités proposé par Cormas, notre approche de modélisation peut être décomposée en trois sous parties distinctes : modélisation spatiale, sociale et passive.

Modélisation spatiale

La modélisation spatiale consiste à définir les entités qui vont représenter l'espace dans lequel la simulation va se dérouler. Dans le cas que nous considérons, nous n'avons que deux entités spatiales :

- L'entité *Sea*. Cette entité va nous permettre de décomposer l'océan en petites cellules. Celles-ci vont donc être la plus petite unité élémentaire de définition de l'environnement. Elles héritent de l'entité *SpatialEntityCell* de Cormas. De part cet héritage, elles sont donc considérées comme des cellules d'un automate cellulaire. Cette caractéristique nous permet donc de simuler les différents flux entre les cellules. Enfin, l'entité *Sea* sait si un port lui est rattaché et lequel ;
- L'entité *Zone*. Cette entité permet de regrouper un ensemble de cellules du type *Sea* dans des partitions clairement définies. De cette manière, nous pouvons représenter le fait que l'océan est découpé en n zones distinctes. L'entité *Zone* hérite de l'entité *SpatialEntityAggregate* de Cormas. En outre, une entité de type *Zone* connaît les poissons qui sont dans sa zone et le port qui lui est attaché (par le biais des cellules de type *Sea* qui la composent). De plus, elle connaît le flux instantané de migration des poissons entre les zones, les mortalités instantanées liées à la pêche et la croissance instantanée des populations de poisson. Enfin, comme ces entités *Zone* vont être soumises à des pollutions, elles sont donc capables, à tout instant, de déterminer si elles sont dans un état pollué ou non ;

Modélisation sociale

La modélisation sociale consiste à définir quelles vont être les entités actives (*e.g.* les agents) du système. Dans notre cas, et du fait que notre approche est, pour l'instant, très globale, nous avons décidé de considérer que l'entité active du système n'est pas le bateau, mais le port. En fait, c'est ce dernier qui gère une flottille de bateau et qui prend les décisions pour eux. C'est donc lui qui va prendre les décisions quant aux déplacements éventuels d'un port à un autre d'un ensemble de bateaux. Enfin, cette entité est située et va donc être rattachée à une cellule. Cette entité est appelée *Harbor* et elle hérite de l'entité *AgentLocation* de Cormas. Elle connaît le prix des poissons, ce qui lui permet de calculer les différentiels d'efforts qui vont lui permettre d'adapter la répartition de sa flottille en fonction de la biomasse, des pêches et donc de la rentabilité de l'activité de pêche.

Modélisation passive

A la différence de la modélisation sociale, la modélisation passive, comme son nom l'indique, consiste à énumérer les entités qui vont être présentes dans le modèle et qui vont servir de support

pour les agents ou pour les entités spatiales. Dans notre modèle, nous avons identifié deux entités passives :

- L’entité *Fish*. Cette entité représente la biomasse simulée dans le modèle. Elle hérite de l’entité *ObjectLocation*. Les espèces sont localisées. Il y aura donc autant d’entités de type *Fish* que d’espèces présentes dans la simulation. L’entité *Fish* contient des informations sur la population, le taux de croissance naturel et le coefficient de mobilité propre à son espèce. De plus, elle connaît la zone à laquelle elle est rattachée. Enfin, cette entité est capable de prendre en compte la mortalité liée à la pêche, de migrer vers d’autres zones et d’augmenter grâce au taux de croissance naturel.
- L’entité *Boat*. Cette entité représente l’effort de pêche simulé. Elle hérite de l’entité *ObjectLocation*. Elle n’est pas localisée directement, mais elle connaît son port de rattachement. Il n’y a qu’une seule entité de type *Boat* par zone. Une entité *Boat* connaît le prix de poisson qu’elle pêche et ceci en fonction des espèces, l’effort total disponible par espèces, la capturabilité des espèces qu’elle cible, ses captures, ses gains et ses coûts. Enfin, chaque entité de type *Boat* est munie d’un taux de polyvalence qui exprime sa faculté à changer de métier et d’un coefficient de diffusion spatial qui exprime les coûts liés à la pêche en dehors de sa zone de rattachement.

3.3.4 ... À la simulation

Impact d’un arrêt temporaire de pêche

Plan d’expérimentation

Dans un premier temps, une analyse de sensibilité du modèle sur le niveau d’effort de pêche a été réalisée. Nous avons donc pu déterminer le niveau d’équilibre du modèle dans le cas où il n’y a pas de fermeture de zone et dans le cas où il y a une fermeture de zone. Les simulations sont réalisées sur 300 pas de temps. Une première simulation est réalisée sans fermeture de zone, une deuxième est réalisée avec une fermeture de zone portant sur une espèce et une troisième est réalisée avec une fermeture de zone portant sur les deux espèces présentes dans la zone. Dans les deux cas où on effectue une fermeture de zone, celle-ci intervient entre les pas de temps 10 et 100.

Dans un deuxième temps, l’effort de pêche initial est fixé à un niveau qui permet de dégager des marges positives au sein de la pêcherie. Les simulations sont réalisées sur 300 pas de temps et la fermeture de zone sur une espèce dans une zone donnée intervient entre les pas de temps 10 et 100. Deux cas sont considérés : dans une première simulation, les unités de pêche concernées par la fermeture de zone ne sont pas autorisées à se reporter spatialement (*i.e.* elles ne peuvent aller pêcher ailleurs) ; dans une seconde simulation, les unités de pêche concernées par la fermeture de zone sont autorisées à se réallouer spatialement ou à changer d’espèce cible. L’analyse de ces résultats porte sur la dynamique de la réponse de l’effort de pêche et les conséquences économiques de cette fermeture de zone pour toutes les combinaisons espèce/zone. Ces comparaisons se font en valeurs actualisées.

Dans un troisième temps, une analyse de sensibilité portant sur la mobilité et la polyvalence des bateaux a été réalisée. De nouveau, les simulations sont réalisées sur 300 pas de temps et une fermeture de zone portant sur une espèce dans une zone précise intervient entre les pas de temps 10 et 100. Ces analyses de sensibilité sont réalisées à des niveaux d’effort de pêche initial

différents afin de pouvoir étudier l'impact de la pression de pêche. L'analyse de ces résultats porte sur les coûts économiques globaux de la fermeture de zone en comparaison de la situation lorsqu'il n'y a pas de fermeture de zone.

Jeu de simulation 1 : Les rentes à court terme à l'équilibre comme une fonction du niveau d'effort

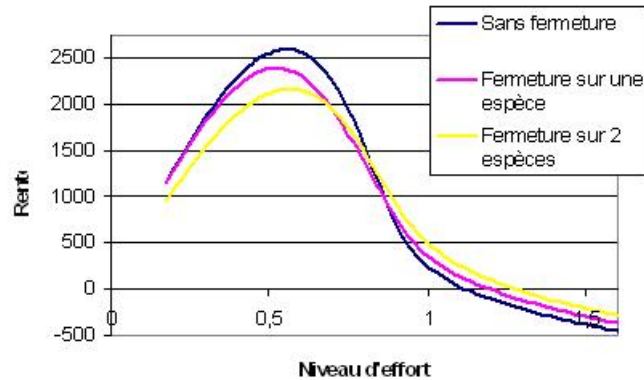


FIGURE 3.10 – Évolution de la rente pour différents niveaux d'effort donnés

La fermeture de zone implique une diminution de la rente globale de la pêcherie. Cette diminution est plus importante si on ferme complètement une zone que si on la ferme partiellement. Dans les deux cas (fermeture de zone partielle et totale), la rente optimale se situe avant celle qui correspond au cas où il n'y a pas de fermeture de zone. De plus, un des effets de la fermeture partielle de zone totale ou partielle est que la pêcherie demeure rentable malgré un niveau d'exploitation supérieur. Ceci est dû à un effet «réserve». En effet, pendant la fermeture de zone, la biomasse concernée croît rapidement à sa capacité de charge maximum. Quand ce niveau est atteint, la biomasse exportée dans son voisinage direct. Par conséquent, le voisinage reçoit un flot de biomasse et supporte l'augmentation de l'effort

Jeu de simulation 2 : Dynamique de la réponse à une fermeture de zone

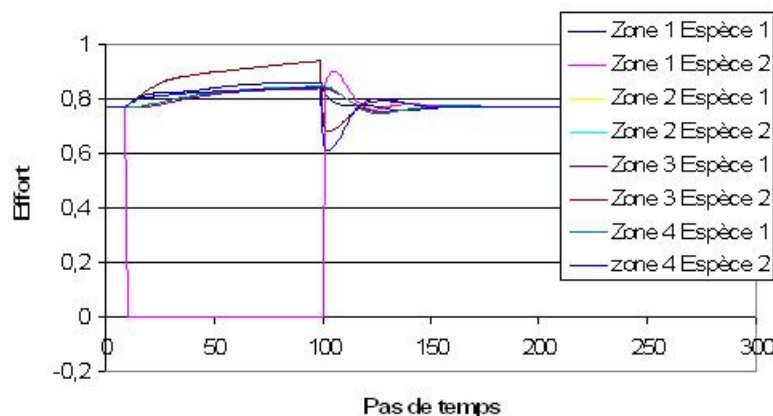


FIGURE 3.11 – Dynamique de la réponse à une fermeture de zone

Pendant la fermeture de zone dans la zone 1 sur l'espèce 2, l'effort de pêche sur cette espèce

tombe à 0. Cet effort est transféré aux autres zones et/ou métiers. L'effort sur les espèces 2 dans les zones 2, 3 et 4 augmente grâce à ces transferts. L'effort sur l'espèce 1 dans les zones 1, 2, 3 et 4 augmente pendant la fermeture de zone, mais moins rapidement que sur l'espèce 2.

A la fin de la fermeture de zone, la zone 1 devient très attractive il y a des transferts massifs vers celle-ci. Mais ces transferts engendrent le fait que la zone 1 devient rapidement moins attractive et il apparaît des transferts négatifs. 50 pas de temps après la réouverture de la zone, le système revient à une situation d'équilibre.

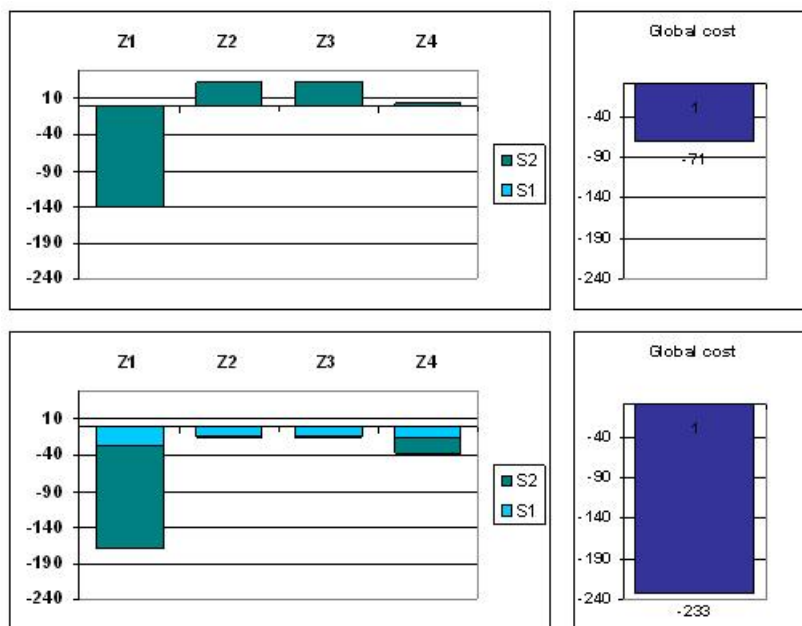


FIGURE 3.12 – Coûts par zone et globaux de la fermeture de zone

La première partie de la Figure 3.12 montre le coût global et le coût par zone de la fermeture de zone quand les bateaux ne sont pas autorisés à se réallouer spatialement (*i.e.* le bateaux restent au port). Dans ce cas, le coût est entièrement supporté par la zone où la fermeture est effectuée. Les autres, au contraire, dégagent une rente supérieure grâce aux migrations qu'elles ont reçu de la zone 1. La zone 4 est moins touchée par la fermeture car elle n'est pas située dans le voisinage direct de la zone 1. Elle reçoit des migrations provenant de la zone 2 et 3 par un effet de cascade.

La seconde partie de la Figure 3.12 montre le coût global et le coût par zone de la fermeture de zone quand les bateaux sont autorisés à se réallouer spatialement et/ou à changer de métier. Dans ce cas, le coût de la fermeture est entièrement supporté par toutes les zones. De plus, le coût est aussi supporté par les pêcheries ciblant l'autre espèce. Ceci est dû au taux de polyvalence : un certain nombre d'effort change de métier et donc les coûts sont aussi supportés par les bateaux ciblant l'autre espèce.

Jeu de simulation 3 : Analyse de sensibilité

La Figure 3.13 montre le cas d'une pêcherie sous exploitée. Dans ce cas, le coût de la fermeture de zone est plus important si les bateaux restent au port ou s'ils changent de métier. En effet, du fait que la biomasse est sous exploitée, il reste au sein de la pêcherie un fort potentiel

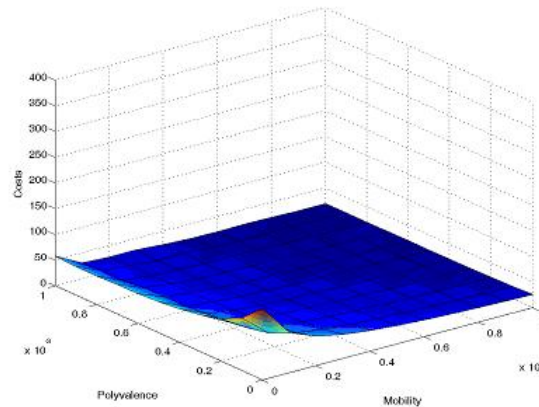


FIGURE 3.13 – Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d’effort bas

d’exploitation. Donc, même avec des pénalités de réallocation spatiale, les niveaux de capture sont tels qu’ils annulent pratiquement les surcoûts. Dans le cas où on change seulement de métier, on augmente la pression de pêche sur une espèce et donc le niveau de capture baisse et par conséquence, la rente aussi.

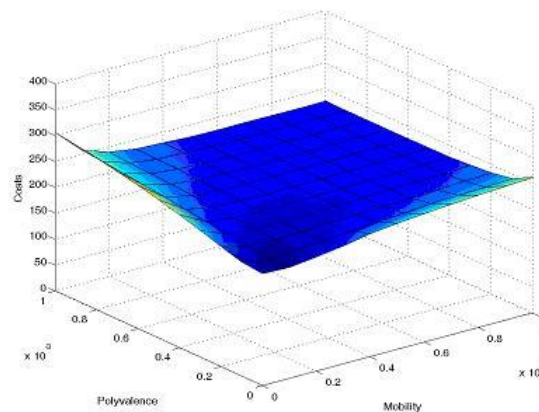


FIGURE 3.14 – Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d’effort à l’optimum

La Figure 3.14 montre le cas d’une pêcherie exploitée à son optimum. Dans ce cas, le coût de la fermeture de zone est plus haut que précédemment (environ 4 fois plus). Les bateaux n’ont aucun intérêt à être seulement polyvalent ou seulement mobiles spatialement. Dans ces deux cas, le coût de la fermeture augmente. Dans tous les autres cas, le coût de la fermeture reste assez similaire. Lorsqu’on se situe à l’optimum, le fait d’augmenter, même légèrement, la pression de pêche se traduit immédiatement par le fait que l’on passe dans une situation sur exploitée et donc par une baisse des captures et par conséquence par une baisse de la rente.

La Figure 3.15 montre le cas d’une pêcherie sur exploitée. Dans ce cas, la meilleure solution pour les bateaux est de rester au port. Dans tous les autres cas, le coût de la fermeture est

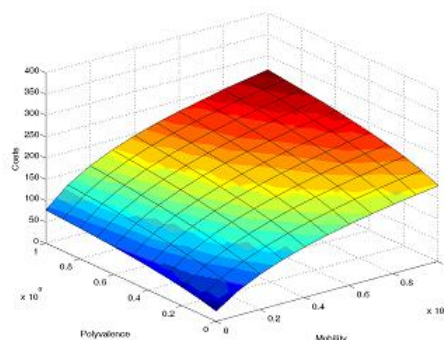


FIGURE 3.15 – Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d’effort à l’optimum

supérieur. On peut noter que l’option de polyvalence est moins coûteuse que la mobilité. Ceci est dû aux pénalités spatiales qui sont incluses dans le modèle.

Nouvelles simulations utilisant DYNFISH 1

Dans cette partie, nous utilisons le modèle pour tester de nouvelles hypothèses et des possibilités différentes de DYNFISH 1. Ces hypothèses viennent du fait que des travaux récents dans le domaine de l’écologie des pêcheries ont fait états de grands changements dans la composition de débarquements des pêcheries commerciales ([Pauly *et al.*, 1998], [Essington *et al.*, 2006]).

Les dimensions économiques des séquences d’exploitation ont conduit au développement de recherches empiriques qui étudient les changements dans les débarquements des pêcheries que ce soit en terme de volume ou de valeur. [Sumaila, 1998] a présenté une analyse basée sur des données de la FAO collectée entre 1950 et 1996 sur plus de 1000 espèces de poisson. [Pinnegar *et al.*, 2002] a analysé les modifications dans la composition des débarquements et leurs prix relatifs sur 26 espèces de la mer Celtique de 1970 à 2000. [Pinnegar *et al.*, 2006] propose une autre analyse des données italiennes recensant 33 espèces sur la même période. Enfin, [Steinmetz *et al.*, 2006] a analysé la composition des débarquement de la flotte française opérant dans l’atlantique nord-est sur la même période. Là encore, les mêmes modifications ont été observées.

Cette partie vise donc à reproduire ces séquences d’exploitations avec le simple modèle bio-économique DYNFISH 1. Ces simulations ont été réalisées dans le cadre du projet Chaloupe (voir la section 3.5 pour une description complète du projet).

Plan d’expérimentation

Le modèle est appliqué dans le cas deux espèces et seulement une zone car la spatialisation n’est pas importante dans cette configuration

Dans ce cadre, nous avons effectué deux types de simulations :

1. Premièrement, les simulations sont réalisées avec différents jeux de paramètres dans un cadre dynamique afin de montrer des séquences d’exploitations des espèces et de tester la sensibilité de ces séquences aux paramètres biologiques des deux espèces.

La pêche est dans une situation d'accès libre et les bateaux peuvent entrer et sortir de la pêche en fonction de sa rentabilité. Dans le but d'observer l'impact sur la dynamique de la pêche des paramètres biologiques et économiques des deux espèces, nous avons testé deux scénarios : (i) le cas avec deux espèces ayant des paramètres biologiques identiques, mais avec des paramètres économiques différents (le prix d'une espèce et dix fois plus petite que l'autre) et (ii) le cas où les espèces ont des paramètres biologiques et économiques différents (le taux de croissance intrinsèque de l'espèce la plus chère et trois fois plus petite que l'autre espèce). Dans les cas, la capacité de l'effort de pêche à réallouer entre les deux espèces est la même (polyvalence identique). Au début de la simulation, les deux espèces ne sont pas exploitées avec une introduction limitée d'effort au pas de temps 1.

2. Deuxièmement, nous avons réalisé une analyse de l'état d'équilibre de la pêche et la sensibilité de ces états d'équilibre aux prix relatifs de ces deux espèces.

Dans ce jeu de simulation, nous considérons que les différents niveaux d'effort dans la pêche peuvent être contrôlés. Par contre, il n'est pas possible de contrôler les niveaux d'effort qui sont dédiés à chaque espèce sur la zone. De ce fait, comme le niveau total d'effort est fixé, la réallocation entre les espèces se fait librement. Les simulations sont réalisées dans le but d'identifier les états d'équilibre de la pêche pour différents niveaux d'efforts et en fonction des prix relatifs des deux espèces : des analyses de sensibilité sont réalisées en fixant $p_1 = \alpha \times p_2$ avec $\alpha \in [0, \dots, 1]$ (p_i signifiant prix de l'espèce i). Enfin, des simulations sont aussi réalisées en fonction des deux scénarios pré-cités.

Résultats de simulation

1. La Figure 3.16 illustre la dynamique globale de la pêche à partir d'une situation in-exploitée, dans le cas de conditions d'accès libre et avec des paramètres biologiques et économiques différents. D'une manière générale, les captures totales augmentent continuellement tout au long de la simulation, alors que l'effort total se caractérise par trois phases distinctes : (i) augmentation rapide, (ii) stagnation et légère diminution, et (iii) nouvelle augmentation et enfin, ajustement à l'équilibre d'accès libre.

Cette évolution est le résultat de la dynamique combinée de l'effort et des captures, comme illustré dans la Figure 3.17.

Le développement de la pêche, dans un premier temps, se fait sur l'espèce 2 qui a un prix de vente plus haut. Ce développement conduit à une diminution de la productivité qui encourage l'exploitation de l'espèce moins rémunératrice et à une réduction de l'effort total dans la pêche (du fait que les revenus par unités d'effort deviennent négatifs). Le développement progressif de l'espèce 1 continue de manière monotone alors que l'effort sur l'espèce 2 oscille jusqu'à ce que le système s'ajuste à un équilibre d'accès libre.

Dans le cas où les espèces ont les mêmes paramètres biologiques mais des paramètres économiques différents, les simulations ont une dynamique assez similaire. La seule différence notable consiste en une oscillation plus forte.

2. Les résultats concernant l'état d'équilibre de la pêche sont présentés en terme d'effort, de biomasse et de rente.

Quand l'effort augmente dans la pêche, sa distribution entre les deux espèces est modifiée : alors que pour les bas niveaux d'efforts, la pêche se concentre sur les espèces chères (espèce 2), la pêche d'espèces peu chères se développe pour des niveaux d'efforts intermédiaires (Figure 3.18).

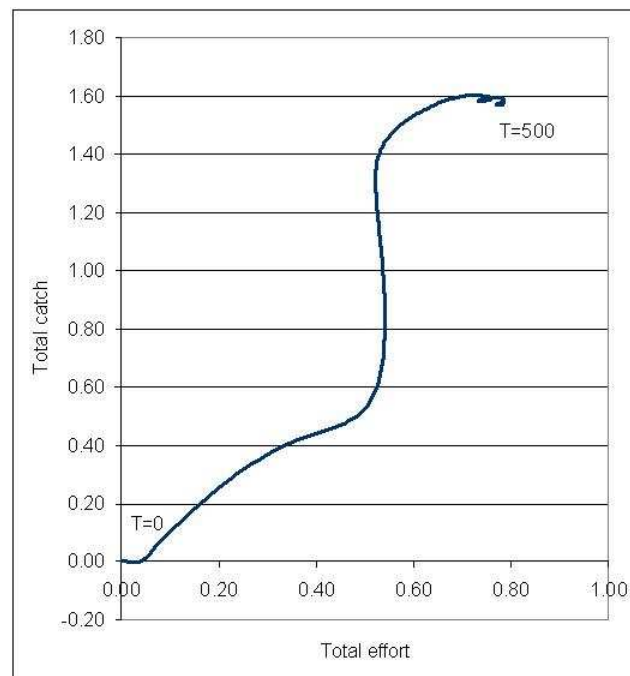


FIGURE 3.16 – Évolution des captures totales

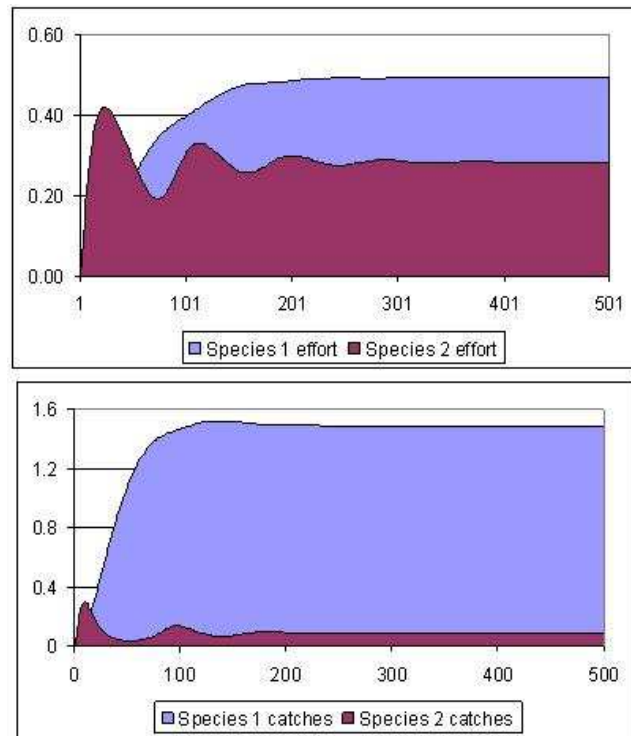


FIGURE 3.17 – Évolution, par espèces, de l'effort et des captures

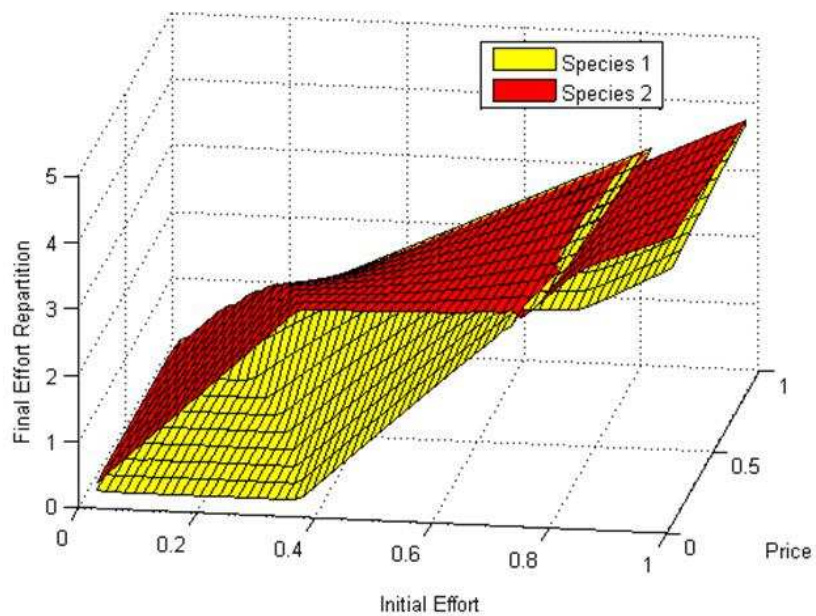


FIGURE 3.18 – Répartition de l'effort final en fonction de l'effort initial et du prix

Le statut d'équilibre de la pêcherie en accès libre varie en fonction de l'intensité de la différence de prix entre les deux espèces et l'effort total dans la pêcherie (Figure 3.19).

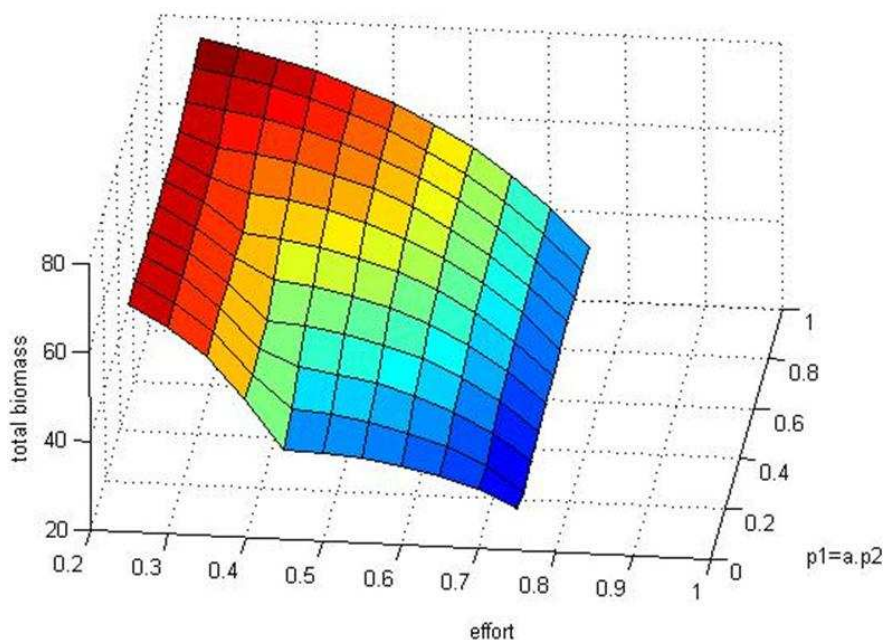


FIGURE 3.19 – Évolution de la biomasse totale en fonction de l'effort et des différentiels de prix

On peut mettre en exergue deux aspects :

- (a) Plus le différentiel de prix entre les espèces est fort, moins le niveau de biomasses à l'équilibre est haut pour un niveau d'effort donné. Ceci est dû au fait qu'une grande différence entre les prix entraîne une plus grande concentration de l'effort de pêche et donc un niveau de biomasse plus faible pour cette espèce ; particulièrement pour des niveaux d'efforts intermédiaires (toutes choses égales par ailleurs) ;
- (b) La diminution de la biomasse à l'équilibre quand l'effort augmente est plus aiguë avec des différentiels de prix qui augmentent. De plus, on peut constater une discontinuité. Ceci résulte de la combinaison de différentes réponses des deux espèces à différentes pressions de pêche. Comme illustré dans la Figure 3.20, avec de forts différentiels de prix et de faibles niveaux d'efforts, la pêcherie cible principalement les espèces chères (espèce 2) et, par conséquent, la biomasse de celle-ci est réduite. Pour des niveaux d'efforts intermédiaires, la pêcherie cible des espèces peu chères et donc sa biomasse et aussi réduite, alors que la biomasse à l'équilibre de l'espèce 2 continue de diminuer.

La Figure 3.21 montre la rente à l'équilibre dans la pêcherie pour différents niveaux d'efforts. Les plus hauts niveaux de rentes sont obtenus pour des niveaux d'efforts intermédiaires et pour des prix similaires. Par contre, quand les différentiels de prix deviennent importants, les plus hauts niveaux de rentes sont atteints pour des niveaux d'efforts plus bas. Dans ce cas, l'exploitation se fera essentiellement sur le prix élevés.

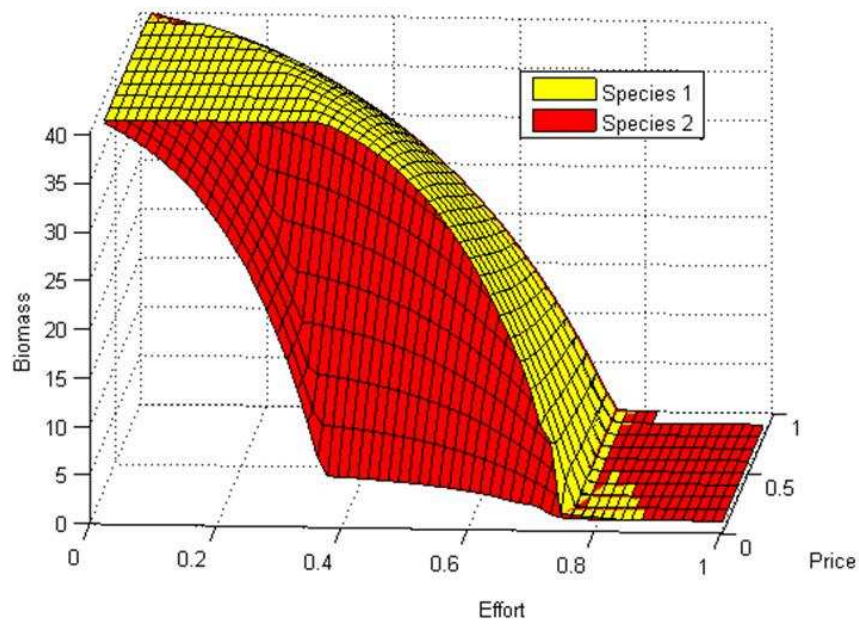


FIGURE 3.20 – Évolution de la biomasse en fonction des prix et de l'effort

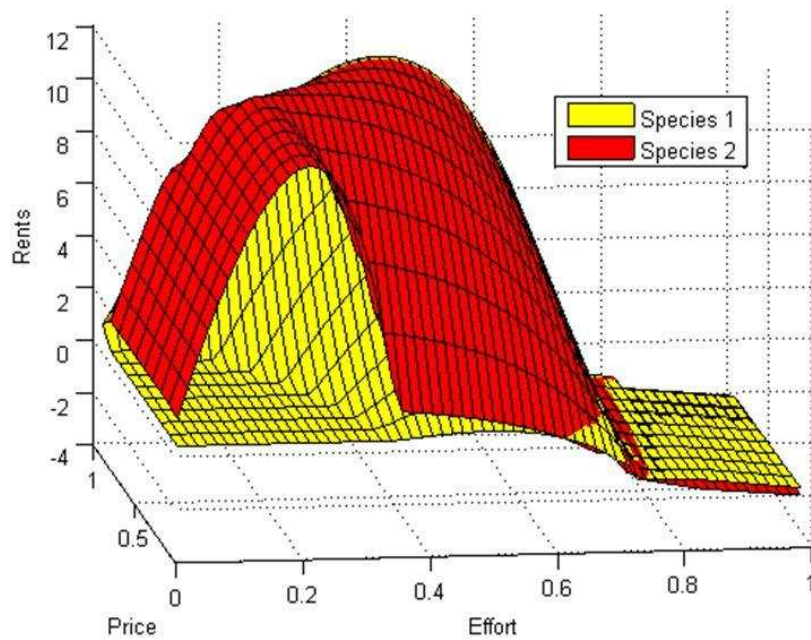


FIGURE 3.21 – Évolution de la rente en fonction des prix et de l'effort

3.3.5 Application à la pêcherie d'Algue Française – Modèle SEAWEED

Le modèle SEAWEED est une extension du modèle théorique DYNFISH 1. Nous allons présenter ici les extensions liées aux spécificités de ce modèle.

Modélisation spatiale

Dans ce modèle, nous sommes confrontés à une véritable spatialisation. Les zones de pêche des Algues sont stockées dans un fichier Arc Gis et les données liées à l'activité de pêche sont stockées dans une base de données. Cette dernière contient des données journalières depuis les 15 dernières années [Arzel, 1998].

Concernant la spatialisation de l'environnement, nous avons exporté les informations de Arc Gis vers un fichier texte pouvant être lu et interprété par Cormas. Suite à ce couplage « faible » ou bien encore par les données, nous obtenons l'environnement de simulation suivant :

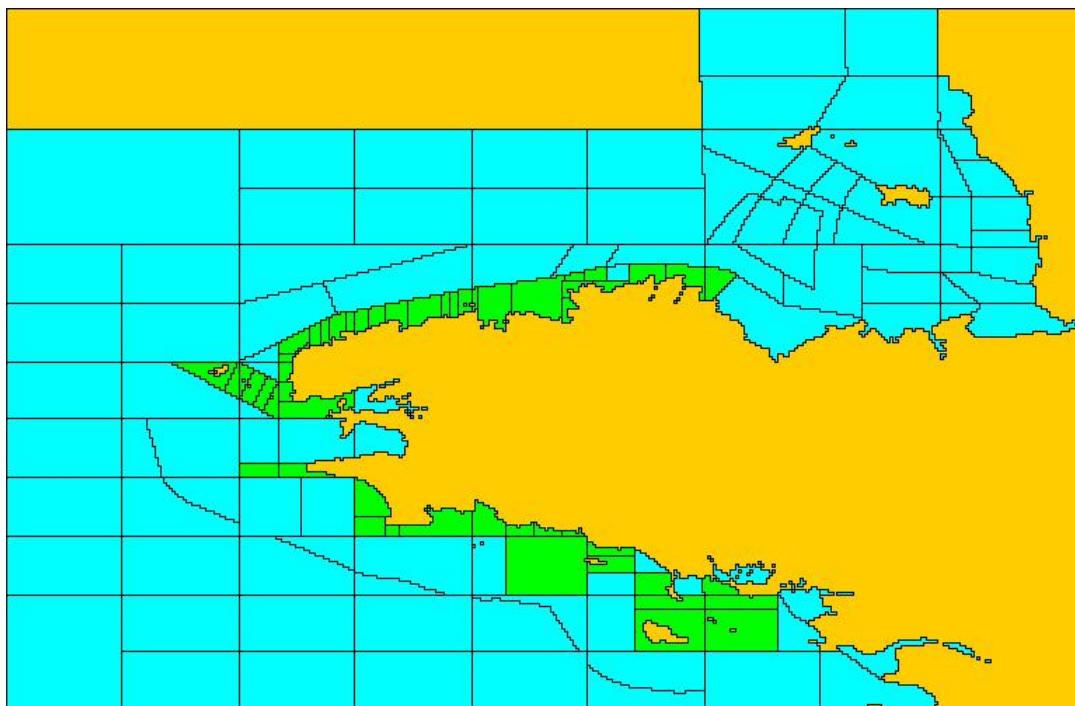


FIGURE 3.22 – Représentation de l'environnement dans Cormas

Les entités spatiales utilisées dans le modèle sont les suivantes :

- L'entité *Sea*. Cette entité va nous permettre de décomposer l'océan en petites cellules. Celles-ci vont donc être la plus petite unité élémentaire de définition de l'environnement. Elles héritent de l'entité *SpatialEntityCell* de Cormas. De part cet héritage, elles sont donc considérées comme des cellules d'un automate cellulaire. Cette caractéristique nous permet donc de simuler les différents flux entre les cellules. Enfin, l'entité *Sea* sait si un port lui est rattaché et lequel ;
- L'entité *Zone*. Cette entité permet de regrouper un ensemble de cellules du type *Sea* dans des partitions clairement définies. De cette manière, nous pouvons représenter le fait que

l'océan est découpé en n zones distinctes. L'entité *Zone* hérite de l'entité *SpatialEntityAggregate* de Cormas. En outre, une entité de type *Zone* connaît les poissons qui sont dans sa zone et le port qui lui est attaché (par le biais des cellules de type *Sea* qui la composent). De plus, elle connaît le flux instantané de migration des poissons entre les zones, les mortalités instantanées liées à la pêche et la croissance instantanée des populations de poissons. Enfin, comme ces entités *Zone* vont être soumises à des pollutions, elles sont donc capables, à tout instant, de déterminer si elles sont dans un état pollué ou non ;

- L'entité *StatsZone*. Elle hérite de l'entité *Zone*. Elle représente un morceau de l'océan identifié par un carré statistique. L'identifiant est un nombre qui est extrait du SIG et on stocke aussi son nom (par exemple : « 25E4 »). Ces zones sont représentées par les rectangles bleus dans la Figure 3.22 ;
- L'entité *SeaweedZone*. Elle hérite de l'entité *StatsZone*. Elle représente une zone de l'océan qui contient un champ d'algues qui est exploité par les pêcheurs d'algues. Comme pour l'entité *StatsZone*, cette entité contient un identifiant généré par le SIG et son nom associé (par exemple : « A3 »). De plus, cette zone garde en mémoire les bateaux qui y sont venus et le tonnage des algues qui y ont été pêchés. Enfin, cette entité contient un booléen qui définit si la zone peut-être exploitée ou non. Ces zones sont représentées par les rectangles verts dans la Figure 3.22.

Modélisation passive

Les entités passives utilisées dans le modèle sont les suivantes :

- L'entité *Seaweed*. Cette entité représente la biomasse d'algues dans une *SeaweedZone* donnée. Elle est caractérisée par une aire, une densité d'algues, une capacité de charge, un taux de croissance naturelle, une accessibilité (plus la zone est accessible, plus on pêche d'algues) et un taux de rejet (ce coefficient représente le volume de biomasse inexploitable dans la zone considérée). Cette entité est, en fait, l'entité *Fish* du modèle théorique auquel on a rajouté les éléments spécifiques au modèle SEEWEEED ;
- L'entité *Harbor*. À la différence du modèle théorique, l'entité *Harbor* n'est plus un agent car nous avons individualisé notre flotte virtuelle. L'entité *Harbor* est définie par un ID, un nom et une zone. La zone est l'endroit de la côte où est localisé le port ;
- L'entité *SeaweedHarbor*. Cette entité hérite de l'entité *Harbor*. Cette entité représente un port qui va servir de lieu de déchargement des algues pêchées. Cette entité va, en outre, être utilisée dans notre algorithme du choix de la zone de pêche ;
- L'entité *Manager*. Cette entité est une entité abstraite qui va servir de racine d'héritage pour les gestionnaires du temps et des saisons ;
- L'entité *SeasonManager*. Cette entité sera responsable de faire passer la saison courante entre `#summer` et `#winter` ; et inversement. Le passage de `#winter` à `#summer` se fait aléatoirement entre Avril et Mai. Le passage de `#summer` à `#winter` se fait aléatoirement entre Septembre et Octobre ;
- L'entité *DateManager*. L'unité des données étant en heure et comme nous devons aussi gérer les saisons, nous avons utilisé un gestionnaire de temps classique. Il gère les heures, les jours et les mois. De cette manière, nous pouvons aussi produire des indicateurs du type rendement horaire.

Modélisation sociale

L'entité *Boat* contient les attributs suivants :

- *ID* ;
- *Power* : la puissance du bateau en CV ;
- *Length* : la longueur du bateau ;
- *Width* : la largeur du bateau ;
- *NumberOFGear* : le nombre d'engins de pêche (dans notre cas, 1 ou 2) ;
- *Crew* : le nombre de personnes composant l'équipage.
- *StartFishingTime* : l'heure à laquelle le bateau commence à pêcher ;
- *EndFishingTime* : l'heure à laquelle le bateau termine à pêcher.

De plus, cette entité accède à la base de données liées au SIG pour connaître les distances exactes entre son port de rattachement et les zones de pêches potentielles. Ces données vont être utilisées dans la fonction de décision du choix de zone de pêche.

Enfin, cette entité va surcharger la fonction *evolve* de Cormas afin de définir, à chaque pas de temps, dans quelle zone le bateau va aller pêcher. Soient les notations suivantes :

- Le port : p ;
- La classe de bateau : c ;
- La zone de pêche : i ;

On va définir les variables suivantes :

- Le temps de pêche maximum : MFT ;
- Le temps de parcours aller/retour entre le port et la zone de pêche : $ST_{c,p,i}$;
- Le temps de pêche :

$$FT_{c,p,i} = MFT - ST_{c,p,i} \quad (3.9)$$

- La productivité : $Produc_c$;
- La charge maximum : $MaxLoad_c$;
- La production :

$$C_{c,p,i} = \min(Produc_c \times FT_{c,p,i}, MaxLoad_c) \quad (3.10)$$

- Le nombre de bateaux : $NbBoats_{c,p,i}$;
- L'index d'effort :

$$EI_{c,p,i} = \frac{(NbBoats_{c,p,i} \times Produc_c)}{\sum_{c,p} (NbBoats_{c,p,i} \times Produc_c)} \quad (3.11)$$

- La biomasse disponible : B_i ;
- La production potentielle par zone et par port :

$$PP_{c,p,i} = NbBoats_{c,p,i} \times C_{c,p,i} \quad (3.12)$$

- La production effective par zone et par port :

$$EP_{c,p,i} = \begin{cases} EI_{c,p,i} \times B_i & \text{si } \sum_i PP_{c,p,i} > B_i \\ PP_{c,p,i} & \text{sinon.} \end{cases} \quad (3.13)$$

– Production effective par vaisseau :

$$EPV_{c,p,i} = \frac{EP_{c,p,i}}{NbBoats_{c,p,i}} \quad (3.14)$$

– Temps de pêche effectif :

$$EFT_{c,p,i} = \frac{NbBoats_{c,p,i} \times EPV_{c,p,i}}{Produc_c} \quad (3.15)$$

– Revenus moyens :

$$AR_i = \frac{\sum_i EP_{c,p,i}}{\sum_i EFT_{c,p,i}} \quad (3.16)$$

– Choix de la zone de pêche à un pas de temps t :

$$SelectedZone(t) = \begin{cases} \max_i (AR_i) & \text{si } AR_i = 0 \\ SelectedZone(t-1) & \text{sinon.} \end{cases} \quad (3.17)$$

Simulations & Résultats

Un premier jeu de simulations a été réalisé sur les bases de données et du modèle présenté ci-dessus. Les résultats sont préliminaires et encore assez peu réalistes. Malheureusement le décès de P. Arzel qui était un des experts français de la pêche à l'algue et la référence à l'IFREMER nous a empêché d'aller plus loin dans ce travail.

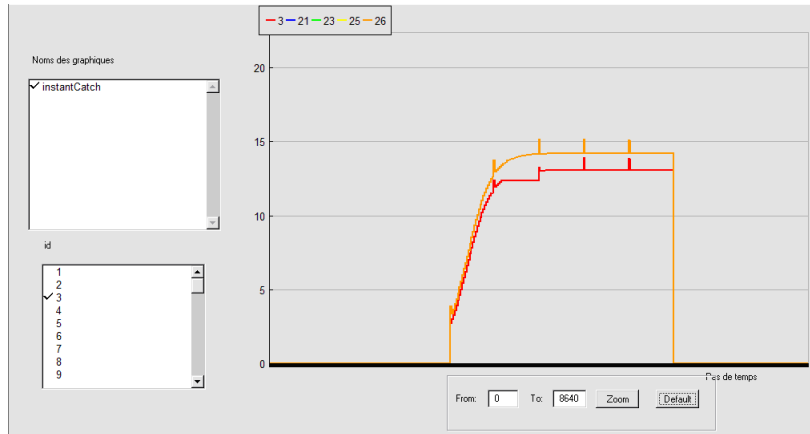


FIGURE 3.23 – Captures totales

Les captures totales (Figure 3.23) augmentent régulièrement pendant la simulation. Cette augmentation commence lorsque la saison passe de `#winter` à `#summer` et cesse lorsque la saison change de `#summer` à `#winter`. Le volume d'algues pêché est le même que celui observé dans les bases de données. Au niveau global nous pouvons conclure que le modèle semble se comporter de manière correcte.

L'activité de pêche au niveau individuel (Figure 3.24) se déroule lorsque la saison est à `#winter`. Dans cet exemple, les bateaux sélectionnés pêchent avec la même intensité même s'ils ne sont pas dans la même zone de pêche. Les changements abruptes sont liés aux changements de

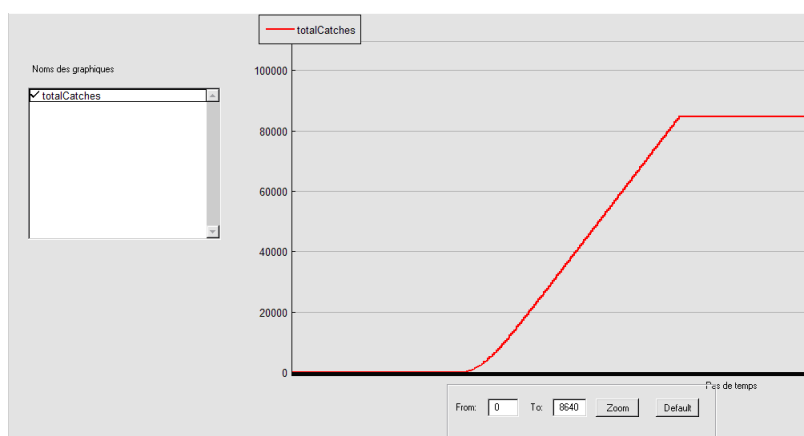


FIGURE 3.24 – Captures individuelles

zones. A ce stade, et au niveau individuel, le modèle ne semble pas se comporter correctement. Il restait donc un gros travail de validation à faire avec P. Arzel, malheureusement impossible à ce jour.

3.4 Une extension du modèle : DYNFISH 2

3.4.1 Présentation du modèle

Le modèle DYNFISH 1 (voir section 3.3) nous a permis de valider un certain nombre d'hypothèses et d'explorer les nombreuses capacités du modèle.

Néanmoins, du fait de l'approche mono-paradigme liée à la plate-forme CORMAS, nous avons souhaité tester la faisabilité et la mise en œuvre de l'extension concernant le couplage de modèles discrets et d'équations différentielles (voir section 2.2). Nous en avons profité pour modifier les spécifications du modèle DYNFISH 1 initial afin de lui rajouter un modèle de marché simplifié afin de pouvoir simuler un micro économie en considérant le prix actuel du poisson ainsi que la consommation des bateaux en carburant. Par contre, comme dans le modèle DYNFISH 1 initial, l'espace est discrétisé en cellules qui vont contenir de la biomasse (une seule espèce) et nous avons un seul port. Enfin, nous sommes passés à une échelle inférieure en considérant que les agents ne sont plus les ports, mais les bateaux. Cette option avait été considérée au début de la mise en œuvre du modèle DYNFISH 1, mais pour des problèmes de performance, cette approche a été abandonnée.

Ce modèle DYNFISH 2 repose sur trois entités : les bateaux, les poissons et un port.

- **Les bateaux** : un bateau est défini comme une entité autonome caractérisée par trois variables :
 - la vitesse de déplacement ;
 - la consommation en carburant ;
 - l'effort de pêche qui traduit la capacité d'un navire d'aller opérer dans une zone donnée en utilisant un engin donné.

Après être allé au port chercher un scénario de pêche, le bateau part en mer effectuer sa

campagne. Il va noter sa consommation en carburant, ainsi que la quantité de poisson capturée pour chaque zone.

La campagne de pêche terminée, il rentre au port pour vendre ses poissons et indiquer sa consommation. Le port est alors capable de calculer le bénéfice associé à cette campagne.

Il demande ensuite un nouveau scénario et repart en mer.

- **Les poissons** : à l'inverse des pêcheurs, le choix pour la dynamique des poissons s'est porté sur un modèle analytique plutôt que sur une simulation comportementale. Et ceci pour plusieurs raisons : les équations gérant la biomasse de populations de poissons sont classiques et éprouvées (on peut trouver dans la littérature un certain nombre d'exemples. Parmi celles-ci, nous pouvons signaler : [Gordon, 1954], [Shaefer, 1957] et [Sanchirico et Wilen, 1999]. Ces dernières ayant aussi été à la base des travaux sur DYN-FISH 1 (voir 3.3). Enfin, et surtout ce choix permet de tester l'intégration dans le système complexe d'équations différentielles. Le modèle repose sur le découpage de la mer en zone où l'on va suivre une biomasse de poissons.

Les interactions et l'évolution de la biomasse sont gouvernées dans notre modèle par la série d'équations suivantes :

$$x_i = f_i(x_i) + d_{ii}x_i + \sum_{j=1, j \neq i} d_{ij}x_j \text{ pour } i = 1, \dots, n \quad (3.18)$$

Où :

- x_i est la biomasse de poisson dans la zone i (la case i de l'automate cellulaire représentant l'espace considéré) ;
- $f_i(x_i)$ traduit la dynamique de croissance de la population dans la cellule i ;
- $d_{ii}x_i + \sum_{j=1, j \neq i} d_{ij}x_j$ traduit l'interaction entre les cellules : dispersion de l'espèce considérée, source, puits, etc.

Différents choix sont possibles pour fixer ces fonctions, nous avons fait les suivants :

1. $f_i(x_i) = r_i \left(1 - \frac{x_i}{x_i^{\max}}\right)$ avec r_i le taux de croissance naturelle et x_i^{\max} la capacité de charge de la cellule. Cette équation traduit une croissance logistique de la population de poissons au sein d'une cellule ;
2. $d_{ii}x_i + \sum_{j=1, j \neq i} d_{ij}x_j = \sum_j \left(\frac{x_j}{x_j^{\max}} - \frac{x_i}{x_i^{\max}}\right)$ avec j dans le voisinage de i (les 4 cellules autour de i dans le cas d'un voisinage de connexité 4, les 8 cellules autour de i dans le cas d'un voisinage de connexité 8, etc.) qui décrit une dispersion naturelle des poissons en fonction de la concentration dans la cellule (ou bien encore appelée « relation de densité dépendance ») ;
3. Lorsqu'un bateau prélève des poissons dans une cellule, la biomasse se voit soustraite de la quantité suivante :

$$perte = x \times (1 - e^{c \times T \times E})$$

Où

- x est la biomasse présente ;
 - c est la capturabilité du poisson ;
 - T est la durée de la pêche ;
 - E est l'effort de pêche du bateau.
- **Le port** : à l'instar des bateaux, le port est une entité autonome. Après chaque campagne, les bateaux lui font part de leurs pêches et, grâce à ces renseignements, il va tenter

de créer une représentation approximative des réserves de poissons disponibles. Les bateaux peuvent effectuer une requête de scénario suivant une politique prédéfinie. Le port possède donc une liste de politiques possibles pour la création de scénarios. Puis, suivant sa représentation interne des réserves disponibles ainsi que du choix politique, il va créer le scénario. Ce choix a été fait afin de simuler les discussions que pourraient avoir les pêcheurs une fois au port.

Le port est également responsable du calcul des bénéfices associés à une campagne de pêche. Il connaît le prix du poisson pour chaque cellule ainsi que le prix du carburant. Les prix peuvent osciller durant la simulation suivant un schéma prédéfini dans le fichier de configuration.

Le port gère également des interdictions de pêche dans certaines zones. De la même façon que les prix, ces interdictions sont définies sous forme d'actions à une date précise dans le fichier de configuration. La conséquence d'une interdiction de cellule à la pêche est que celle-ci ne peut apparaître dans un scénario tant que l'interdiction n'est pas levée.

3.4.2 Implémentation dans VLE

D'un point de vue purement conceptuel, nous allons reprendre le schéma développé pour le modèle *fireman*³ de la section précédente, seule la complexité des acteurs en présence va le différencier. Ainsi une approche orientée agents a été choisie sur l'ensemble du projet pour faire suite au modèle CORMAS et les pompiers/pyromanes sont ici remplacés par le port et les bateaux. L'ensemble des cellules contenant le poisson représente l'environnement physique où évoluent les agents.

- **Le port** : l'intérêt principal d'une conception orientée multi-agents est la possibilité de définir précisément le comportement des entités présentes dans la simulation et ce de façon naturelle. C'est en quelque sorte une définition locale à l'inverse d'une définition globale plus classique (par exemple à travers des équations synthétisant un comportement quelconque des entités). Cette façon de faire offre, bien sûr, plus de possibilités et est beaucoup plus paramétrable. Son point noir reste, dans certain cas, la vitesse d'exécution des simulations. Il est par ailleurs intéressant de constater qu'il est possible de mélanger ces deux approches. En effet les poissons sont gérés par des équations différentielles et les pêcheurs/ports sont orientés agents afin de tirer le meilleur parti des deux formalismes. La définition à proprement parler de l'agent port reste simple : le port est un agent réactif qui communique avec les pêcheurs.
- **Les bateaux** : tout comme le port, les pêcheurs sont de simples agents réactifs qui vont échanger des données avec le port et interagir avec les cellules de poissons. Mais la conception orientée multi-agents permet d'envisager des comportements très complexes : on peut imaginer d'ajouter la possibilité d'échange de données en mer pour effectuer des campagnes groupées ou, au contraire, pour se partager le plus efficacement possible les ressources à un instant donné, mettant ainsi en place un réseau social représentant les accointances entre les pêcheurs. Ce réseau n'étant pas forcément figé, l'intensité de la relation entre les pêcheurs pouvant évoluer au cours de la simulation. On peut également envisager des comportements sur le choix des lieux et cibles de pêche très sophistiqués très proches de la réalité, etc. Tant de comportements impossibles à faire ressortir avec de simples modèles analytiques. La méthode employée est de nouveau la même que pour les pompiers et les pyromanes. Les bateaux sont des agents

3. voir : <http://vle.univ-littoral.fr/fr/index.php/Fireman>

encapsulés dans un modèle atomique DEVS. Lors de leur déplacement ces modèles se retrouvent connectés tantôt avec une cellule de poisson ou tantôt avec le port. Une fois la connexion établie, par le biais des événements externes propres à DEVS, les agents communiquent avec la case en faisant des requêtes de scénarios au port, en pêchant dans les cellules de poissons, etc.

- **Les poissons** : C'est ici que les méthodes d'intégration développées dans les sections précédentes vont entrer en jeu. Nous avons ici à faire à une série d'équations différentielles interconnectées. Ce n'est pas une équation discrétisée sur un ensemble de cases comme pour la diffusion de la chaleur sur une barre, mais une série de zones, où vivent des poissons. Dans chaque zone, il y a une équation différentielle à résoudre pour obtenir la biomasse de poisson présente. Par contre, chaque zone est connectée à ses 4 voisines et elles s'influencent toutes mutuellement. Nous avons aussi suivi la méthode développée dans les sections précédentes :
 - la grille est gérée par un ensemble de cellules type CellDEVS/QSS1, implémentée directement dans VLE ;
 - chaque cellule ainsi définie est connectée avec ses 4 voisines. Le formalisme CellDEVS rend automatique la gestion des perturbations entre cellules : quand une cellule change de valeur, cette nouvelle valeur est automatiquement envoyée aux voisines.
 - QSS1 prend en charge dans la cellule la résolution de l'équation. Nous n'avons pas testé QSS2 sur ce modèle ;
 - enfin nous avons opté pour une politique de perturbation, lorsque les bateaux viennent pêcher dans une cellule. Puisque chaque cellule résout une équation différente, nous avons choisi une perturbation ponctuelle simple (sans gaussienne) comme pour le modèle *fireman*.

Le code pour CellDEVS/QSS1 étant préexistant au sein de VLE, il a suffi d'hériter de cette classe afin d'implémenter une case de poisson,

3.4.3 Les résultats de la simulation

Ce modèle restant purement théorique, les résultats attendus sont uniquement un comportement correct de l'ensemble des acteurs en présence. Nous pouvons constater cela sur la Figure 3.25 où nous avons testé un scénario d'Aire Marine Protégée. Les pêcheurs n'ont pas le droit d'aller pêcher dans une zone de la grille (dans le sud-est) pendant un laps de temps durant la simulation. On peut constater une production des poissons dans cette zone avec une diffusion dans les cases voisines à celles interdites. Lors de la levée de l'interdiction, les pêcheurs se précipitent sur les cases non exploitées.

3.5 Le modèle CHALOUPE

3.5.1 Présentation du projet Chaloupe (Projet Biodiversité ANR-IFB 2005) et objectifs

Objectifs & Contexte

Contexte

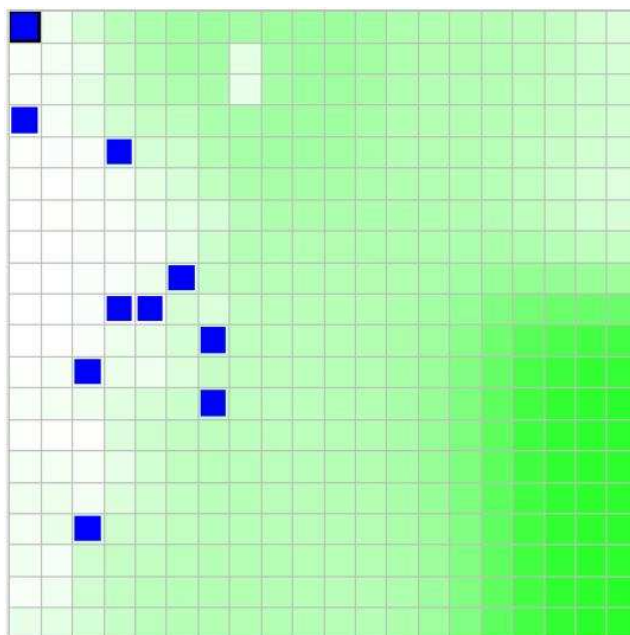


FIGURE 3.25 – Les carrés bleus sont les bateaux, le grand carré noir en haut à gauche est le port et enfin la couleur des cases la concentration en poissons. On peut constater une concentration de poisson en bas à droite, résultant de l’interdiction de pêche dans ce secteur pour une période de la simulation

Des changements majeurs sont aujourd’hui observés dans les écosystèmes marins et les activités humaines qui en dépendent, en particulier la pêche. Si les écosystèmes sont naturellement variables, les changements observés soulèvent des inquiétudes croissantes, à la fois en terme de pertes de biodiversité des peuplements marins et de leur capacité à absorber ces changements sans être radicalement transformés. Ces inquiétudes portent également sur la viabilité des pêcheries, et plus largement des modes actuels d’utilisation des ressources marines vivantes.

Objectifs

L’objectif de la recherche est de décrire et expliquer, à l’échelle régionale, les changements au sein des systèmes peuplement marins / pêcheries, et d’en déterminer les conditions de viabilité. Ces changements doivent être conçus en termes de co-évolution des composantes écologique et anthropique de ces systèmes, en intégrant les sources exogènes de variation : climat, marchés des facteurs de production et des produits de la mer, et gouvernance. Afin de mieux identifier le rôle respectif des différents facteurs d’évolution, cette analyse s’appuiera en particulier sur une approche comparative de trois systèmes choisis pour illustrer la diversité des écosystèmes marins et de leurs modes d’exploitation : le plateau continental tempéré du golfe de Gascogne, la zone à upwelling du Maroc, et le plateau continental amazonien de Guyane française.

Contenu

L’analyse s’appuiera sur les travaux suivants :

1. identifier les changements au cours des dernières décennies des caractéristiques des peuplements de poissons et d’invertébrés et des facteurs potentiels d’évolution en cause (tendances environnementales, effets de la pêche) ;

2. identifier aux mêmes échelles, les changements des caractéristiques de la production halieutique soutenue par ces peuplements et les facteurs potentiels d'évolution en cause (changements des peuplements, évolutions du contexte économique, des marchés des produits de la mer, et de la gouvernance) ;
3. à partir de ces résultats, et d'une réflexion pluri-disciplinaire sur les conditions de viabilité des systèmes étudiés, élaborer des outils de diagnostic ;
4. en s'appuyant sur les trois tâches qui précèdent, proposer des hypothèses concernant les processus écologiques et économiques à l'œuvre, à tester dans des modèles de simulation bio-économique, et explorer les évolutions possibles des systèmes peuplements-pêcheries suivant différents scénarios de gouvernance et d'évolutions de l'environnement marin.

La problématique Scientifique

Viabilité des systèmes peuplements marins / pêcheries

La viabilité des systèmes peuplements marins / pêcheries dépend du maintien de caractéristiques de biodiversité des peuplements marins (diversité, tailles moyennes, dominance, types de stratégie vitale, . . .) et de caractéristiques économiques de l'exploitation et de la commercialisation des productions.

Les pêcheries : interfaces entre systèmes écologiques et systèmes socio-économiques

L'hypothèse sur laquelle s'appuie le projet est la suivante : les évolutions de structure spécifique des débarquements issus de la pêche professionnelle sont le résultat des changements qui se produisent dans le peuplement, et des changements qui s'opèrent dans les stratégies d'exploitation. Ces évolutions déterminent en particulier l'évolution de la valeur de la production halieutique, des espèces marines capturées, et de sa répartition entre les différents agents économiques de la filière halieutique.

Analyse des évolutions : la nécessité d'une approche au niveau régional

Les facteurs clés en cause dans l'évolution des peuplements marins et des pêcheries sont bien identifiés : changement climatique et surexploitation d'une part ; défaillances de la régulation de l'accès aux ressources conduisant à une surcapacité de pêche, dans un contexte d'accroissement de la demande de produits de la mer et d'ouverture des marchés, à l'échelle internationale, d'autre part. L'amplitude des changements observés, et le poids relatifs de ces différents facteurs, à l'échelle régionale restent cependant à quantifier.

Organisation

Pour réaliser les objectifs fixés, quatre tâches ont été définies et seront à réaliser pour trois cas d'étude. Chaque tâche et cas d'étude est sous la responsabilité d'un ou deux chercheurs. Chaque responsable de tâche est choisi pour ses compétences thématique et veille à la réalisation des travaux pour atteindre les objectifs fixés. Chaque responsable de cas d'étude est choisi pour son expertise régionale et veille à la cohérence des travaux des différentes disciplines qui interviennent dans le cas d'étude.

- **Tâche 1. Réponse des peuplements, variations temporelles environnementales et impact de la pêche**

Les variations et tendances temporelles de la structure des peuplements de poissons et invertébrés et de l'environnement hydro-climatique seront analysées aux mêmes échelles spatiales (écosystème) et temporelles (1 à plusieurs décennies). Cette analyse sera conduite en intégrant les connaissances acquises dans la tâche 2 sur l'exploitation pour mieux comprendre le rôle des facteurs en causes (pêche et environnement) dans les variations des peuplements. La réalisation de cette tâche s'appuie sur l'utilisation des données issues de campagnes d'échantillonnage scientifiques et de réseaux de mesures environnementales. Des échantillonnages supplémentaires seront nécessaires dans certains cas d'étude. Les analyses statistiques classiques seront utilisées (régression simples, multiples, ordinations, ...).

La nature des facteurs environnementaux d'intérêt est à identifier par cas d'étude. Il peut s'agir des températures (surface, fond, moyennes mensuelles, annuelles, ...), salinité, débit des fleuves, intensité d'upwelling, régimes de vent, pluviométrie, SOI, ...

Les variations temporelles des caractéristiques des peuplements seront décrites à partir des indicateurs à choisir et définir pour leur réponse à l'impact de la pêche (spectres de taille individuelle, de prix, ...) et au changements environnementaux (affinité thermique, origine biogéographique, habitat, ...).

– **Tâche 2. Réponses des pêcheries aux changements écologiques, économiques et institutionnels**

La tâche 2 vise à mieux comprendre les réponses des pêcheries aux changements écologiques, économiques et institutionnels observés dans les différents écosystèmes étudiés à l'échelle d'une à plusieurs décennies selon les cas.

Dans un premier temps, les évolutions de la production des pêcheries seront décrites : en particulier les changements observés dans la structure des captures et des débarquements. L'analyse des tendances s'appuiera sur des référentiels d'espèces et de groupes d'espèces communs à ceux utilisés pour l'analyse des évolutions des peuplements exploités. Les changements dans la composition de la production et dans les prix à la première vente seront identifiés, ainsi que leurs conséquences en terme d'évolution de la production vendue. L'évolution résultante de la valeur économique de cette production sera analysée à partir des données concernant l'évolution de la structure des coûts d'exploitation et de la demande de produits de la mer : l'évolution de la contribution des différentes espèces exploitées à cette valeur sera identifiée.

Dans un deuxième temps, les principaux facteurs de ces évolutions seront analysés en cherchant à distinguer les effets des changements dans la biodiversité des peuplements mis en évidence par la tâche 1, des autres facteurs. L'analyse de ces autres facteurs s'appuiera sur des informations concernant d'une part l'évolution des capacités de pêche et des stratégies d'exploitation, et d'autre part l'évolution de l'environnement économique (prix des facteurs de production, marché des produits de la mer) et des politiques publiques d'encadrement du secteur (mesures de conservation, conditions d'accès aux ressources, aides publiques).

Cette recherche pourra être réalisée à différentes échelles dans chacun des écosystèmes étudiés, du niveau régional au niveau de flottilles particulières. Elle nécessitera la compilation de séries statistiques existantes à ces différentes échelles auprès de différentes sources : statistiques officielles de production et d'échanges des produits de la mer, réseaux d'observation scientifiques, systèmes de suivi comptable des entreprises, ... L'analyse nécessitera également l'acquisition d'informations complémentaires concernant l'économie des activités de pêche et les stratégies d'exploitation, notamment par voie d'enquête.

– **Tâche 3. Diagnostic sur le statut écologique et économique et conditions de**

viabilité des systèmes peuplements exploités/pêcheries

Les connaissances acquises dans les tâches 1 et 2 doivent permettre d'établir des diagnostics sur le statut écologique et économique des systèmes étudiés. Cela suppose d'identifier un ensemble d'indicateurs décrivant l'état des différentes composantes du système étudié : biodiversité des peuplements exploités, conditions environnementales, exploitation halieutique, environnement économique et politiques publiques.

La synthèse de l'ensemble des tendances observées sur les indicateurs passe par l'élaboration d'une grille d'interprétation a priori : Les combinaisons de tendances indésirables peuvent alors être interprétées au regard des objectifs de gestion et de la dynamique récente.

Cela suppose de conduire une réflexion sur les limites de viabilité des peuplements et des pêcheries.

– Tâche 4. Modèles de simulation des évolutions possibles du système bio-économique

L'objectif est d'aborder de front par le biais de la modélisation la représentation conjointe des processus biologiques et économiques. Il s'agira de reformuler des modèles bio-économiques conventionnels en fonction des spécificités de la question de la biodiversité et de celle des changements d'exploitation :

- biodiversité des peuplements exploités, et son évolution telle que caractérisée dans la tâche 1 ;
- valorisation de la biodiversité telle que caractérisée dans la tâche 2 ;
- représentation des modes de gestion tenant compte de la biodiversité telle que caractérisée dans la tâche 3.

Les modèles envisagés permettront de représenter dans un cadre unique les composantes du système. Ils seront développés d'abord dans le cas du golfe de Gascogne et ensuite transposés aux autres cas d'étude.

A long terme, l'utilité des modèles est double : d'une part mieux discriminer le rôle des facteurs forçant dans les trajectoires observées des différents systèmes et d'autre part simuler l'évolution des systèmes à long terme (décennies) pour des scénarios de gestion et de tendances environnementales. Le second point est un objectif à atteindre au delà du programme proposé ici, car il nécessite d'abord une validation des modèles ; de plus ces modèles devront être alimentés par des scénarios environnementaux vraisemblablement non disponibles dans les 3 ans qui viennent. Les travaux réalisés viseront donc principalement à explorer les différentes hypothèses concernant les facteurs d'évolution des systèmes étudiés émergeant des études empiriques conduites dans le cadre du projet.

3.5.2 Positionnement de DYNFISH 3 dans CHALOUPE

Nous nous sommes positionnés dans la tâche 4 qui a trait à la modélisation et à la simulation. Nous nous sommes donc orientés vers le développement d'un modèle bio-économique des pêcheries du Golfe de Gascogne.

Ce modèle a pour vocation de grossir en complexité jusqu'à la fin de l'ANR, mais aussi de s'étoffer de scénarios tests dont le but est de fournir un outil capable de répondre, au moins en partie, aux questionnements qui ont été soulevés. Il repose aujourd'hui sur le suivi biologique et économique de la pêche mixte merlu-langoustine. Pour cela un modèle biologique par classes d'âge pour les merlus et les langoustines a été développé ainsi qu'un modèle économique basé

sur les flottilles dont la granularité s'étend jusqu'à la flottille/métier/espèce/classe. Une notion de marché a également été mise en place.

Enfin, notre démarche a été guidée par un principe de « généralité » afin de pouvoir ajouter facilement des espèces et des flottilles à ce modèle de manière à ce qu'on puisse étudier de plus nombreuses interactions entre les pêcheries.

Pour finir, ce modèle est une application directe des travaux sur les équations aux différences présentées dans la Section 1.3

3.5.3 Présentation du modèle bio-économique DynFish 3

Dans un premier temps, le modèle s'est attaché à décrire la pêcherie merlu/langoustine du golfe de Gascogne. Les données sont structurées en flottille, métier, classe d'âge, marché et catégorie commerciale. Le pas de temps est l'année. A cet effet, nous avons élaboré un diagramme UML général décrivant une pêcherie. Ce diagramme est présenté dans la Figure 3.26

Les données

Physiquement les données sont regroupées dans un fichier « tableur » dispatchées dans les feuilles suivantes :

- **species** qui regroupe la liste des espèces présentes dans la simulation ainsi que certaines de leurs caractéristiques, par exemple : nombre de classes d'âge, débarquements initiaux totaux, etc ;
- **species_class** qui regroupe cette fois-ci des données comme le poids moyen aux âges, la catégorie commerciale de la classe, etc ;
- **market_species_com_cat** recense les modes de calcul des prix au kilo et les données s'y rattachant ;
- **flotilla** contenant l'effort total initial ainsi que les données économiques nécessaires pour calculer le revenu rattaché à une flottille ;
- **flotilla_metier** suite des données précédentes spécifiques aux métiers ;
- **flotilla_metier_species** dans cette feuille nous avons entre autres les débarquements initiaux pour chaque espèce en fonction de la flottilles et du métier ;
- **flotilla_metier_species_class** et enfin celle-ci précise la répartition de ces débarquements à travers les différentes classes d'âge en fonction donc de la flottille, le métier et l'espèce.

L'ensemble de ces données sont tirées du SIH, le Système d'Information Halieutique⁴.

Le modèle théorique

Conventions

1. Les noms en italique *XYZ* sont les variables du système et les noms normaux XYZ sont des données tirées du fichier de données ;
2. Les notations utilisées pour les indices sont les suivantes :

4. http://www.ifremer.fr/sismer/FR/banquesnat/sih_FR.htm

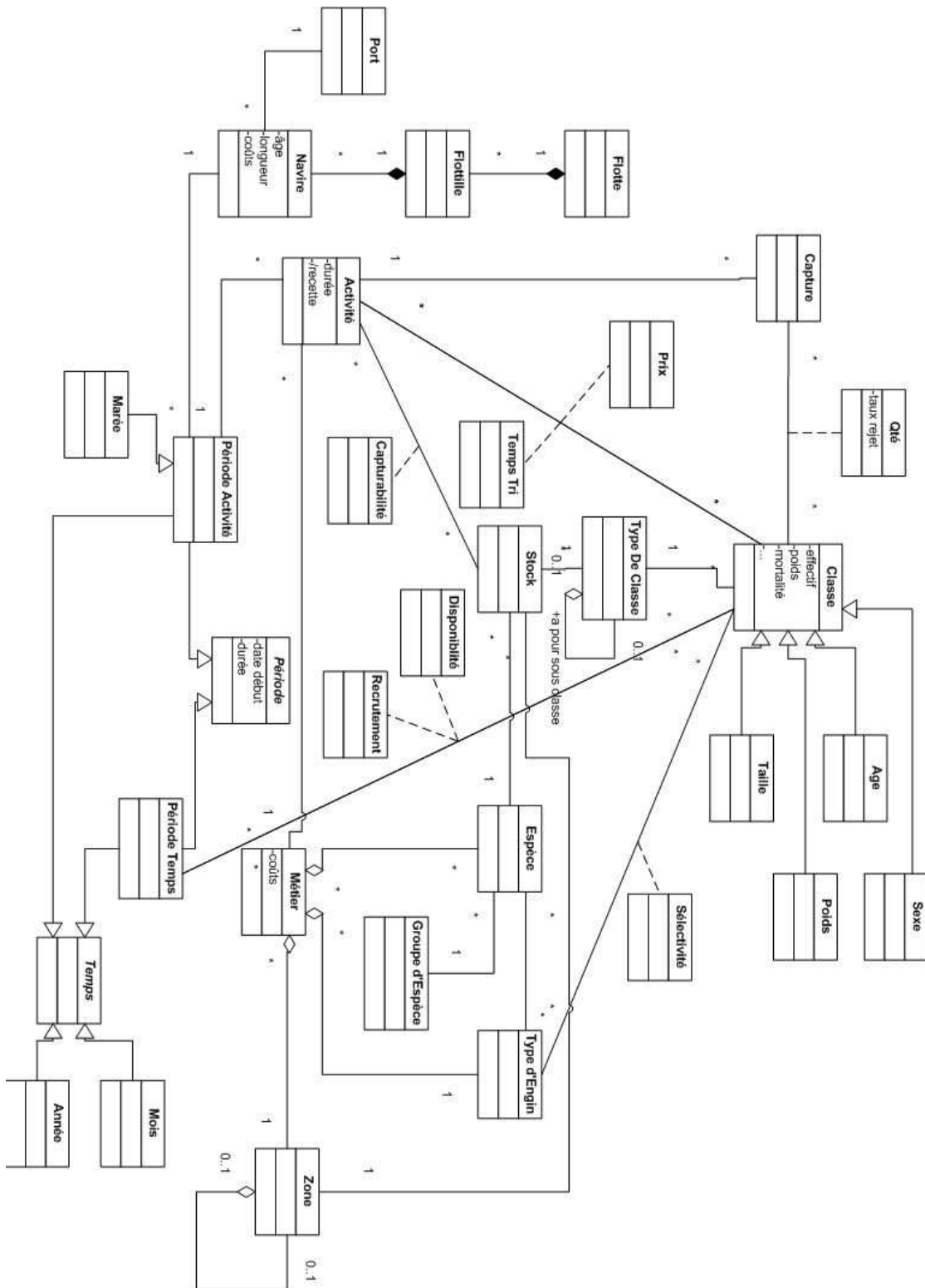


FIGURE 3.26 – Description UML d’une pêcherie « générique »

- f pour flottille ;
 - m pour métier ;
 - s pour espèce ;
 - c pour classe ;
 - M pour marché ;
 - C pour catégorie commerciale.
3. Les variables, possédant un zéro en exposant XYZ⁰, représentent des calculs réalisés au début de la simulation pour initialiser le modèle. Leurs valeurs ne va plus changer durant la simulation.

Modèle par classes d'entités

– Par flottille

- Calcul du nombre de vaisseaux par flottille. Pour l'instant c'est une constante.

$$NbVessels_f = \text{INITIAL NUMBER}_f \quad (3.19)$$

- Évaluation du reste à partager, RTBS « *Return To Be Shared* » pour les métiers autres que ceux simulés au sein d'une flottille. Comme on peut le voir, on récupère le RTBS des métiers calculés, on fait la différence avec le revenu réellement observé et pour la suite on fait l'hypothèse que cette différence restera proportionnelle à l'effort attribué à ORTBS « *Other Return To Be Shared* ». Cet effort peut changer, par exemple, dans le scénario de réallocation dynamique de l'effort (où l'effort est réalloué en fonction des marges réalisées lors de la précédente campagne).

$$OtherEffort_f = \left(1 - \sum_m \text{INITIAL EFFORT}_{f,m} \right) \times \text{INITIAL EFFORT}_f \quad (3.20)$$

$$OMC_f^0 = \frac{\text{INITIAL RTBS}_f \times \text{INITIAL EFFORT}_f \times NbVessels_f - \sum_m \text{RTBS}_{f,m}}{OtherEffort_f} \quad (3.21)$$

$$ORTBS_f = OMC_f^0 \times OtherEffort_f \quad (3.22)$$

- Calcul du reste à partager total pour une flottille.

$$RTBS_f = ORTBS_f + \sum_m \text{RTBS}_{f,m} \quad (3.23)$$

– Par flottille/métier

- Calcul de l'effort alloué à un métier. Ce calcul produit une constante, mais il peut être remplacé par une évolution dynamique de la répartition de l'effort total (initialisé avec la constante INITIAL EFFORT_f qui est un pourcentage) au sein de l'ensemble des métiers disponible pour une flottille.

$$E_{f,m} = \text{INITIAL EFFORT}_f \times \text{INITIAL EFFORT}_{f,m} \quad (3.24)$$

- Simulation d'un revenu provenant des autres espèces pour un métier donné. Le calcul est similaire au calcul de ORTBS, on évalue le revenu donné par les espèces suivies, on calcule la différence avec le revenu initial observé et on suppose la différence proportionnelle à l'effort attribué au métier ($IGTO = \text{Initial Gross Turn Over}$).

$$IGT_{f,m} = IGTO_f \times IGTO_{f,m} \times NbVessel_f \quad (3.25)$$

$$OGRC_{f,m}^0 = \frac{IGT_{f,m} - G_{f,m}}{NbVessels_f \times E_{f,m}} \quad (3.26)$$

$$GO_{f,m} = NbVessels_f \times E_{f,m} \times OGRC_{f,m}^0 \quad (3.27)$$

- Calcul du revenu pour ce métier.

$$G_{f,m} = GO_{f,m} + \sum_s G_{f,m,s} \quad (3.28)$$

- On peut ensuite en déduire une série d'indicateurs économiques qui aboutissent au RTBS.

$$NetRevenue_{f,m} = (1 - \text{LANDING COST}_{f,m}) \times G_{f,m} \quad (3.29)$$

$$SharedCost_{f,m} = (\text{FUELC}_{f,m} + \text{BAITC}_{f,m} + \text{ICEC}_{f,m} + \text{FOODC}_{f,m}) \times E_{f,m} \times NbVessels_f \quad (3.30)$$

$$RTBS_{f,m} = NetRevenue_{f,m} - SharedCost_{f,m} \quad (3.31)$$

– **Par flottille/métier/espèce**

- Calcul des débarquements totaux pour une espèce donnée au sein d'une flottille/métier.

$$L_{f,m,s} = \sum_c L_{f,m,s,c} \quad (3.32)$$

- Évaluation du revenu brut pour une espèce donnée au sein d'une flottille/métier.

$$G_{f,m,s} = \sum_C G_{f,m,s,C} \quad (3.33)$$

– **Par flottille/métier/espèce/classe**

- Calcul de $PL_{f,m,s,c}$ (*PERCENT LANDINGS*) qui permet de répartir les INITIAL LANDINGS $_{f,m,s}$ qui sont donnés pour une espèce donnée à travers les classes d'âge. Il y a deux possibilités soit cette donnée est connue et utilisée, soit on suppose une répartition homogène à travers les classes :

1. Donnée connue :

$$PL_{f,m,s,c}^0 = \text{PERCENT LANDINGS}_{f,m,s,c} \quad (3.34)$$

2. Hypothèse d'une répartition homogène :

$$PL_{f,m,s,c}^0 = \frac{\text{INITIAL LANDINGS}_{f,m,s}}{\text{TOTAL LANDINGS}_s} \quad (3.35)$$

- Ceci permet d'évaluer un coefficient de capturabilité globale Q qui permet de passer de l'effort de pêche (en heure) à une mortalité par pêche F injectée dans le modèle biologique pour calculer les captures réalisées dans cette flottille/métier/espèce/classe.

$$Q_{f,m,s,c}^0 = \frac{\text{INITIAL } F_{s,c} \times PL_{f,m,s,c}}{NbVessels_f \times E_{f,m}} \quad (3.36)$$

$$F_{f,m,s,c} = Q_{f,m,s,c}^0 \times NbVessels_f \times E_{f,m} \quad (3.37)$$

- Après un passage dans le modèle biologique, on récupère les captures que l'on suppose réparties au prorata des F respectifs des flottille/métier. Le calcul intermédiaire est utilisé dans les scénarios de quota.

$$RealCaughtWeight_{f,m,s,c} = \frac{C_{s,c} \times F_{f,m,s,c}}{F_{s,c}} \times W_AGE_{s,c} \quad (3.38)$$

$$C_{f,m,s,c} = RealCaughtWeight_{f,m,s,c} \quad (3.39)$$

$$L_{f,m,s,c} = (1 - \text{DISCARD}_{f,m,s,c}) \times C_{f,m,s,c} \quad (3.40)$$

On peut constater une gestion des rejets qui permet le passage des captures C aux débarquements L .

– **Par flottille/métier/catégorie commerciale**

- Calcul des débarquements pour une espèce/catégorie commerciale donnée au sein d'une flottille/métier.

$$L_{f,m,s,C} = \sum_{c \in C} L_{f,m,s,c} \quad (3.41)$$

- Et on en déduit le gain pour cette pêche à travers le marché/catégorie commerciale et de son prix.

$$M_{f,m,s} = \text{MARKET}_{f,m,s} \quad (3.42)$$

$$G_{f,m,s,C} = P_{s,M,C} \times L_{f,m,s,C} \quad (3.43)$$

– **Par espèce/marché/catégorie commerciale**

- Calcul du prix pour une espèce/marché/catégorie commerciale, il y a deux choix possibles au sein du modèle :

1. Prix variable

$$L_{s,M,C} = \sum_{f,m} L_{f,m,s,C} \quad (3.44)$$

$$P_{s,M,C} = \text{ALPHA}_{M,s,C} \times L_{s,M,C}^{\text{BETA}_{M,s,C}} \quad (3.45)$$

2. Prix moyen fixe

$$P_{s,M,C} = \text{MEAN PRICE}_{M,s,C} \quad (3.46)$$

– **Par espèce/classe**

- Calcul de la mortalité par pêche pour une espèce/classe donnée. On commence par simuler cette mortalité pour les flottilles non calculées OF en faisant la différence à $t = 0$ entre la somme des mortalités par pêche de flottille/métier qui pêchent l'espèce donnée et la mortalité initiale observée, on suppose pour la suite de la simulation que cette pression de pêche est constante.

$$OF_{s,c}^0 = \text{INITIAL } F_{s,c} - \sum_{f,m} F_{f,m,s,c} \quad (3.47)$$

$$F_{s,c} = OF_{s,c}^0 + \sum_{f,m} F_{f,m,s,c} \quad (3.48)$$

– Calcul de la mortalité totale.

$$Z_{s,c} = \text{NATURAL_MORTALITY}_{s,c} + F_{s,c} \quad (3.49)$$

– Et enfin, pour la partie pêche on calcule les captures.

$$C_{s,c} = \frac{N_{s,c} \times (1 - e^{-Z_{s,c}})}{Z_{s,c}} \times F_{s,c} \quad (3.50)$$

– Biologie de l'espèce. Il y a trois cas : la première classe (recrutement ou naissance), les classes normales et la dernière classe qui regroupe tous les animaux âgés n'entrant pas dans les autres cas. La dynamique de la population est basée sur les équations de Von Bertalanffy [Von Bertalanffy, 1938].

1. Première classe.

$$N_{s,c} = \text{INITIAL } N_{s,c} \quad (3.51)$$

2. Classe « normale ».

$$N_{s,c} = N_{s,(c-1)} \times e^{-Z_{s,(c-1)}} \quad (3.52)$$

3. Dernière classe.

$$N_{s,c} = N_{s,(c-1)} \times e^{-Z_{s,(c-1)}} + N_{s,c} \times e^{-Z_{s,c}} \quad (3.53)$$

Les scénarios

Différents scénarios ont déjà été programmés et testés, on peut citer par exemple :

1. test d'un quota (de 3000t) uniquement sur les langoustines appliqué à l'ensemble des flottilles ;
2. mise en place également d'un quota de merlu mais uniquement sur les chaluts ;
3. étude des impacts sur le système d'une modification de la sélectivité des métiers sur les deux espèces : la pêche sur les trois premières classes d'âge est interdite ;
4. répercussions d'une réallocation de l'effort en faveur d'un métier casier sur une flottille. L'exemple choisi est la flottille des chalutiers 12-16 mètres du nord du golfe de Gascogne. Cette réallocation est figée dans un premier temps à 50% pour le métier « casier » et le reste de l'effort est partagé à hauteur de 40% pour le métier TTB, 8% pour OTB et 2% pour un métier virtuel représentant les « autres » métiers de la flottille. Dans un second temps, nous avons également testé 100% sur le casier ;
5. fort du scénario précédent, nous avons élaboré un scénario de réallocation dynamique de l'effort en fonction du revenu au pas précédent. Chaque métier calcule un coefficient d'attraction (les revenus divisés par l'effort à fournir) et la flottille réalloue ses efforts au pas de temps suivant tout en tenant compte d'une certaine inertie.

Validation

Nous avons, pour l'instant, programmé, au fil de nos rencontres avec les biologistes et les économistes, l'ensemble des équations présentées dans la partie 3.2 ainsi que l'ensemble des scénarios présentés ci-dessus. Nous avons également implémenté une première version du modèle bar. En guise d'exemple, nous allons présenter quelques sorties du scénario de modification de sélectivité dans le cas du modèle merlu-langoustine. Rappelons que ce scénario consiste à annuler l'effort de pêche sur les trois premières classes d'âge des deux espèces à partir d'un certain t par exemple $t = 30$ ans.

Commençons par la fin avec les figures 3.27 et 3.28 qui représentent respectivement l'évolution de la biomasse pêchée de langoustines et l'évolution du RTBS (« Return To Be Shared », Reste à Partager) pour la flottille formée des chalutiers 12-16m du nord. Le scénario consistant à une privation, les flottilles ne pouvant plus pêcher de petits poissons/crustacés, on pourrait s'attendre à une diminution de la masse pêchée avec pour conséquence une diminution des revenus, mais c'est le contraire qui se produit. Et, non seulement les pêcheurs font des bénéfices plus importants mais en plus la biomasse totale des poissons restant en mer est plus importante ! Voir figure 3.29. En laissant les poissons se développer, les pêches réalisées sur les poissons âgés sont bien plus volumineuses et puisque les pêcheurs font leurs revenus sur le poids et non sur le nombre et que les plus gros ont un prix/kilo plus important, ils sortent gagnant de l'interdiction. On constate bien ces deux phénomènes sur les graphiques, après une petite période de flottement en début de simulation, on voit les indicateurs se fixer à leurs valeurs de référence (obtenues en laissant tourner la simulation 50 pas de temps). Puis au moment de l'application de la mesure, une baisse qui dure 3/4 années suivie par une augmentation significative de la biomasse pêchée ainsi que du revenu.

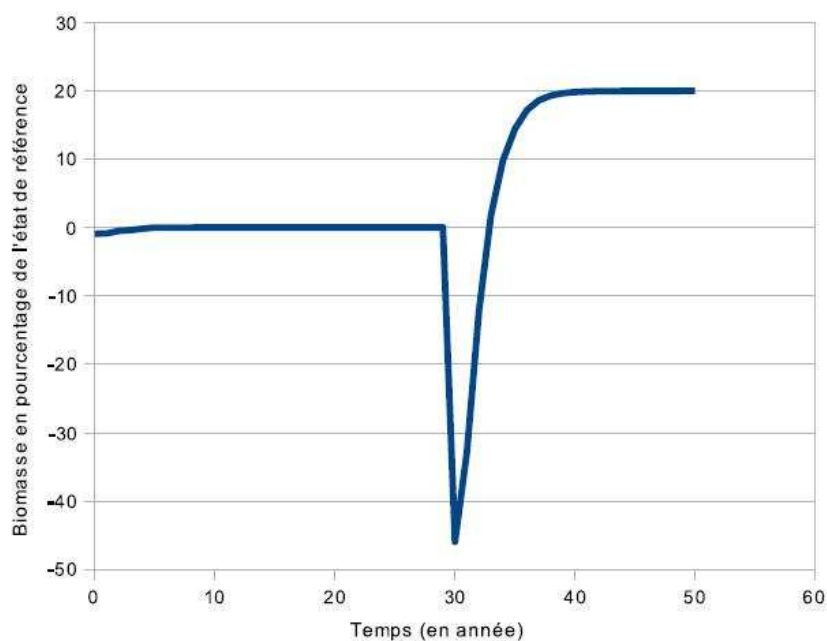


FIGURE 3.27 – Évolution de la biomasse pêchée de langoustines

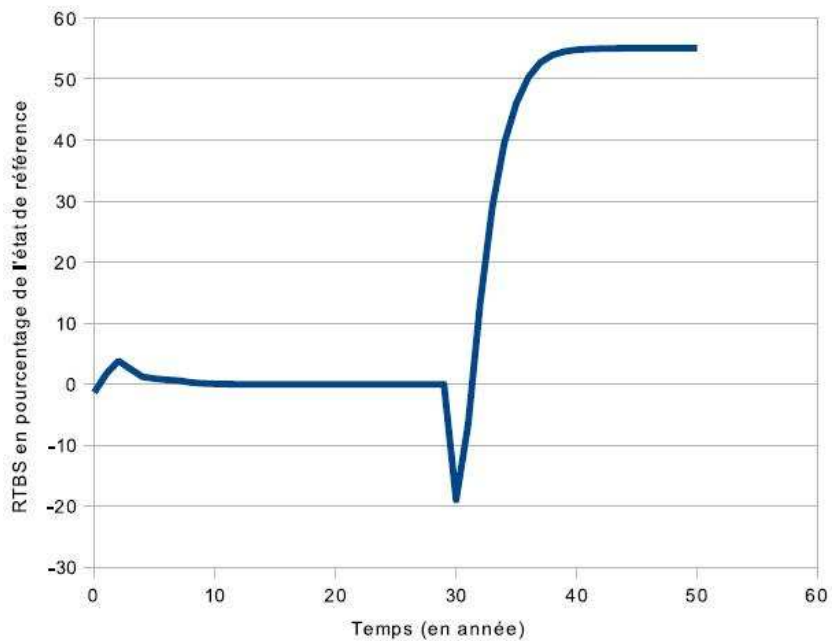


FIGURE 3.28 – Évolution du RTBS pour les chalutiers 12-16 du nord

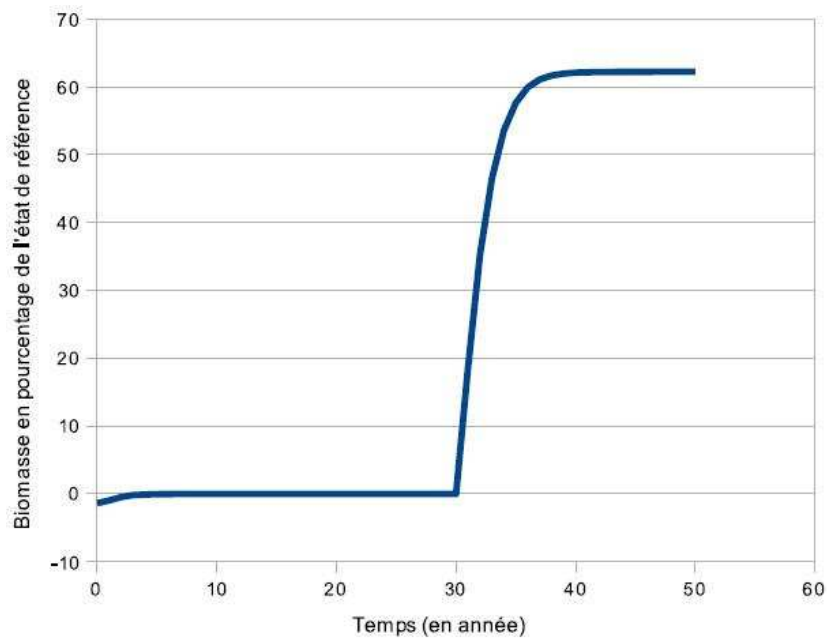


FIGURE 3.29 – Évolution de la biomasse restant en mer de langoustines

3.6 Le modèle ECOMERISTEM

La modélisation des plantes selon l'approche structure-fonction (FSPM) offre une opportunité unique de formaliser, de manière intégrative, les bases élémentaires des interactions *génotype* \times *environnement* ($G \times E$) et leurs relations [Dingkuhn *et al.*, 2005] [Vos *et al.*, 2010]. Les paramètres génotypiques du modèle peuvent donc être utilisés comme des traits élémentaires pour le phénotypage [Reymond *et al.*, 2004] [Yin *et al.*, 1999] et la qualité intégrative du modèle au niveau des processus permet l'exploration d'idéotypes [Dingkuhn *et al.*, 2005].

3.6.1 Présentation du modèle

Ecomeristem est un modèle de culture déterministe de type FSPM qui simule la morphogénèse de la plante au niveau de l'organe et de manière journalière en utilisant la météorologie et l'hydrologie du sol comme paramètre d'entrée. La plante est simulée comme un individu moyen d'une population formant une canopée.

Les processus organogénétiques et morphogénétiques sont pilotés par la relation entre l'assimilation carbonée (photosynthèse brute vue comme une fonction de l'interception lumineuse et d'un coefficient de conversion génotypique, moins la respiration de maintenance) et un puits (demande des organes en train de grandir dépendant de paramètres génotypiques et de conditions environnementales).

La vitesse d'initiation des organes est rythmée par un plastochrone potentiel (*plasto* : un paramètre génotypique). Il est égal au phyllochrone et à la durée de la phase d'expansion d'une feuille lorsqu'elle est apparue. Une fois initiée, une feuille n est pré-dimensionnée en fonction : de la taille finale (longueur, largeur) de la feuille $n - 1$, d'un paramètre génotypique *MGR* (Meristem Growth Rate) et d'un coefficient additif appliqué à la taille finale de la feuille $n - 1$. Ce pré-dimensionnement peut être régulé par une variable *Ic* qui mesure l'état de la plante en terme de ressource (le fonctionnement de la variable *Ic* est expliqué ci-après). Une fois pré-dimensionnée, la feuille grandit selon une vitesse d'expansion foliaire *LER* (Leaf Expansion Rate) égale au ratio entre la longueur finale potentielle de la feuille et de son temps d'expansion (égal à *plasto*, ou plus long si la ressource est limitée).

L'index de compétition interne à la plante *Ic* est calculé chaque jour comme le ratio entre ce que la plante peut fournir en carbone (calculé en utilisant une approche « big leaf » [Luquet *et al.*, 2006]) et la demande de la plante (la somme des demandes des organes de la plante). En fonction de la valeur de l'*Ic*, le nombre d'organes potentiels ainsi que leurs tailles est régulé [Luquet *et al.*, 2007] de la manière suivante :

- Si $Ic < 1$, la taille potentielle de la nouvelle feuille qui va être initiée est réduite, les réserves sont mobilisées pour assurer l'expansion des organes. Si le déficit persiste, la plus vieille feuille de chaque talle est détruite et une part de leur biomasse est réallouée. Si le déficit persiste toujours l'expansion des organes est retardée ; le phyllochrone et le plastochrone sont augmentés ;
- Si $Ic < Ict$ (un paramètre génotypique, c'est la valeur à partir de laquelle on va déclencher le tallage), alors une nouvelle talle n'est pas créée. Dans le cas contraire, on crée une nouvelle talle ;
- Si $Ic \geq 1$, l'excès d'assimilat est stocké temporairement dans un compartiment réserve. Si la réserve est saturée, la vitesse d'assimilation est réduite.

La figure 3.30 illustre le fonctionnement général d'Ecomeristem.

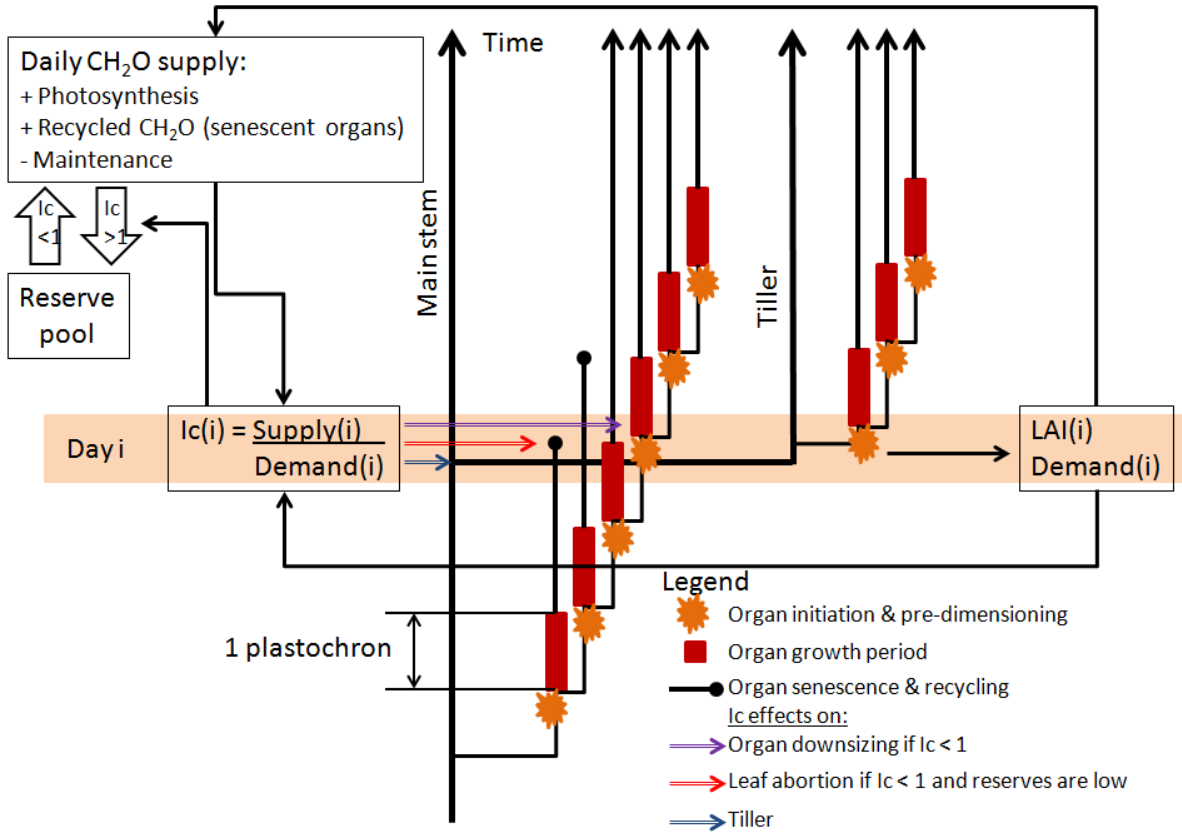


FIGURE 3.30 – Illustration du fonctionnement d'Ecomeristem

Le lecteur intéressé pourra trouver une description beaucoup plus détaillée d'Ecomeristem dans [Luquet *et al.*, 2006] et [Luquet *et al.*, 2007].

3.6.2 Exemple : impact du stress hydrique sur la taille des organes

Afin que la plante puisse prendre en compte le stress hydrique, et tout en conservant le fonctionnement décrit dans la Figure 3.30, les variables et équations suivantes ont été implémentées au sein d'Ecomeristem (pour un organe n donné, à un temps t donné) :

$predim(n)$: le pré-dimensionnement en cm d'un organe est calculé à chaque fois qu'on en crée un nouveau. On a $predim(n) = predim(n - 1) + MGR \times testIc(t) \times fctr(t)$

Où :

- MGR est la paramètre défini en 3.6.1 ;
- $testIc(t) = \min(1, \sqrt{Ic(t)})$ et $Ic(t) = \min(5, \sum_i \frac{Offre_i}{Demande_i})$ pour $\forall i \in \{t - j\}$ avec $\forall j \in [0, 4]$;

- $fcstr(t) = \sqrt{FTSW(t) \times \frac{1}{ThresTransp}}$ avec :
 - $ThresTransp$: un paramètre qui exprime un seuil à partir duquel la transpiration de la plante diminue ;
 - $FTSW(t) = \frac{(SWC(t-1) - PlantTranspiration) + WaterSupply}{RU1}$ où SWC représente le contenu du sol en eau, $FTSW$ la fraction d'eau transpirable du sol pour la plante, $RU1$ la réserve entre la capacité de rétention maximale et la quantité d'eau minimale et $WaterSupply$ un apport en eau extérieur lié à l'irrigation.

La figure 3.31 illustre le principe de fonctionnement d'Ecomeristem sous stress hydrique.

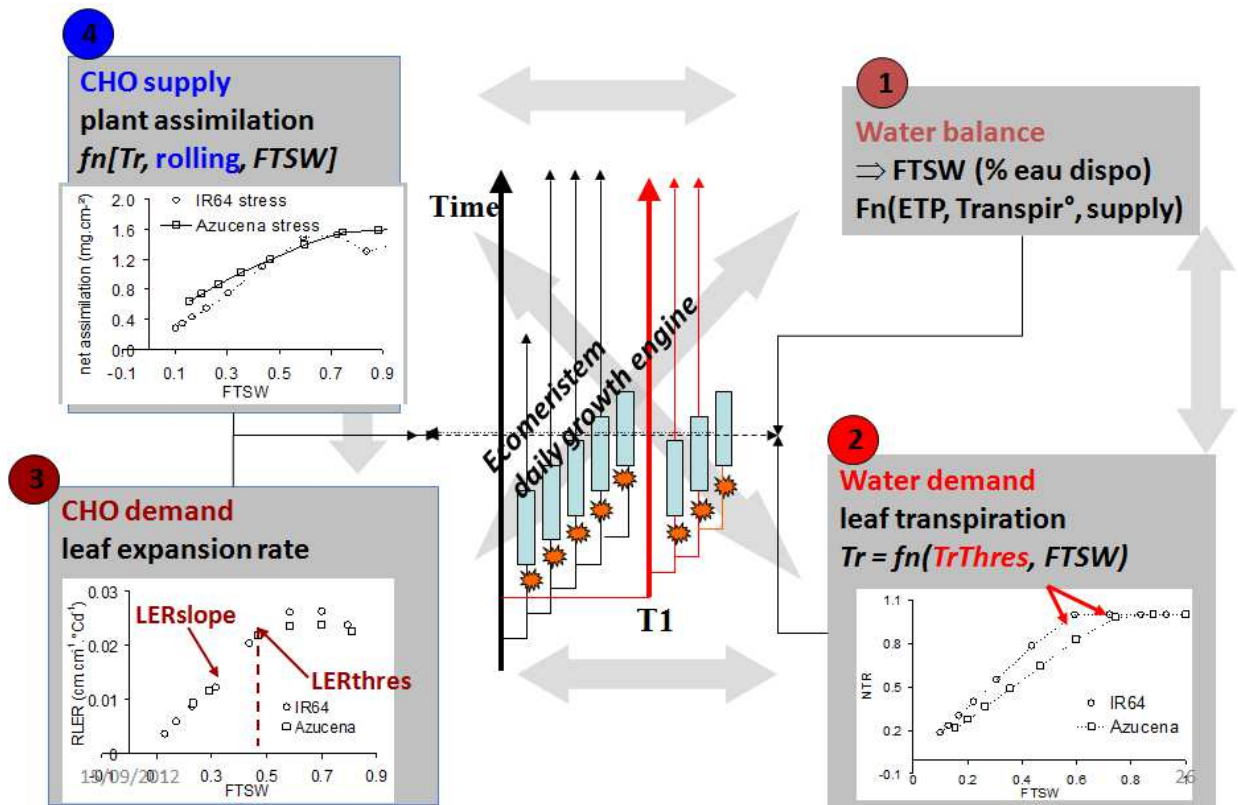


FIGURE 3.31 – Adaptation d'Ecomeristem au stress hydrique – Issu de [Luquet *et al.*, 2006]

On voit donc que le pré-dimensionnement de chaque organe créé est donc fonction de la taille de l'organe précédent et de l'état de la plante par rapport à un stress hydrique. Dans le cas favorable $testIc(t)$ et $fcstr(t)$ sont égaux à 1 et donc, le pré-dimensionnement n'est pas affecté. Si $testIc(t) < 1$ et/ou $fcstr(t) < 1$ (à un instant t donné) c'est donc que la plante subit un stress hydrique et il faut réduire la taille de l'organe créé. Chaque organe grandissant de manière indépendante pendant le reste de sa vie, il représente donc une image de l'état de stress de la plante au moment de sa création.

Les Figures 3.32 et 3.33 illustrent la mise en œuvre de ce mécanisme dans le cadre de

simulation de riz (IR64).

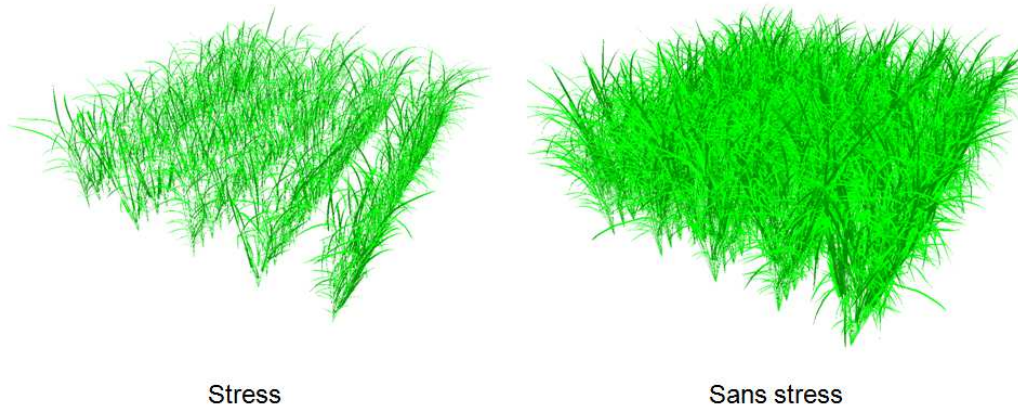


FIGURE 3.32 – Comparaison Stress/Sans Stress hydrique pour une parcelle

On voit donc que l'effet du stress hydrique s'applique bien sur les plantes. L'effet visuel est plus fort au niveau de la parcelle (le paysage est plus fermé). Dans le cas d'un simple plan de riz, on peut constater que si la vitesse foliaire de la plante est forte, le stress est plus atténué et dans le cas où la vitesse foliaire est faible, le stress est beaucoup plus marqué.

3.7 Conclusion

Dans ce chapitre, nous avons présenté des applications directes des approches présentées dans les chapitres précédents. MUFINS est une application directe de l'approche à environnement multiples pour les agents. DYNFISH 1 est un modèle développé sous une autre plate-forme que VLE : CORMAS. DYNFISH 2 quant à lui est une application directe des agents et de la méthode d'intégration QSS. Quant au projet CHALOUPE, il utilise directement les résultats issus de l'extension équations aux différences. Le modèle Ecomeristem, quant à lui existe en deux versions : une version développée sous matlab et une autre sous une plateforme dédiée appelée Ecophen. Néanmoins, le paradigme utilisé est très proche de DEVS.

Ces applications permettent de simuler des systèmes très complexes et de mettre en avant des résultats qu'il n'avait pas toujours été possible de montrer avant. L'avancée la plus notable concerne DYNFISH 1, car il nous a permis de montrer un grand nombre de résultats à partir d'un simple modèle. De plus, le modèle CHALOUPE, de part sa généricité et le travail réalisé

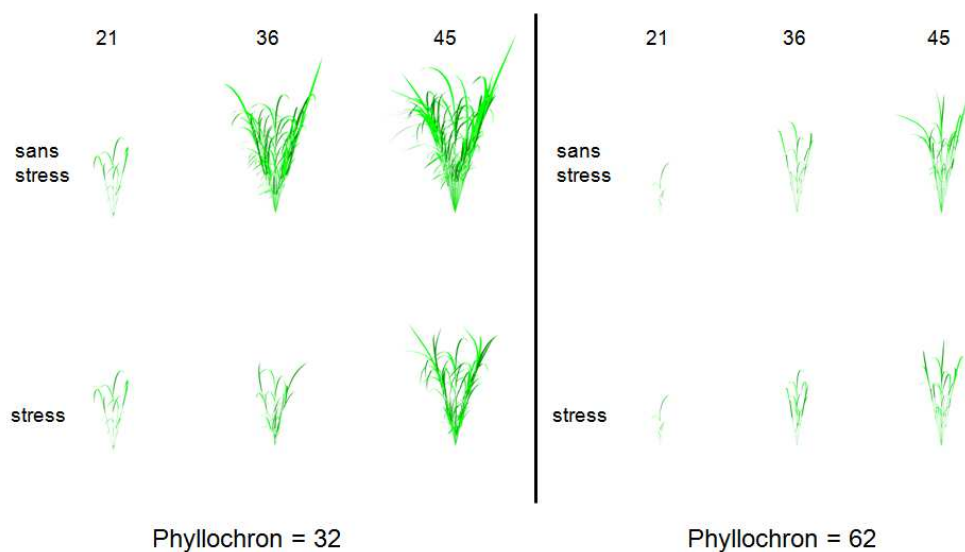


FIGURE 3.33 – Comparaison Stress/Sans Stress hydrique pour deux vitesses foliaires

sur la standardisation des données, va permettre de simuler les pêcheries des 10 espèces qui représentent 80% de l'activité commerciale du Golfe de Gascogne ; déjà, une étude réalisée par B. Drouot sur le bar utilisant ce modèle donne des résultats très prometteurs.

Ces applications montrent la pertinence et le bien fondé de nos choix. De plus, elles ont été le support de 4 thèses : D. Guyomard, G. Quesnel et D. Versmisse et M. C. Rebolledo. Enfin, elles ont été une source non négligeable de financement pour notre équipe.

Chapitre 4

Conclusion & Perspectives

Dans ce document, j'ai présenté mes travaux réalisés dans le cadre de la modélisation et la simulation des systèmes complexes. Ces travaux sont le fruit d'encadrement de doctorants, de travail personnel et ceci, dans la grande majorité des cas, par le biais de projets financés par des organismes de recherche ou bien encore par l'agence nationale pour la recherche par exemple.

Dans un premier chapitre, j'ai montré comment on pouvait construire une extension DEVS pour les systèmes multi-agents avec une approche à environnement multiples. Les approches à base d'agents, ou individus centrés étant de plus en plus utilisées, le fait de pouvoir ajouter cette approche est une réelle plus-value à VLE. Enfin, nous avons ajouté les équations aux différences à VLE. Cet ajout était, également, très attendu car bon nombres de modèles en agronomie ou en halieutique utilisent ce genre d'approches. Dans le domaine de l'agronomie, on peut citer, par exemple STICS ([Brisson *et al.*, 1998], [Brisson *et al.*, 2002]) qui est certainement le plus célèbre d'entre eux au niveau français. Le fait de pouvoir définir des modèles à base d'équations aux différences est un atout non négligeable pour faire adopter VLE dans la communauté des thématiciens.

Dans un deuxième chapitre, nous avons montré comment on pouvait intégrer des schémas d'intégration numérique à VLE. Ces schémas, QSS1 et QSS2 sont basés sur les travaux de E. Kofman. Enfin, nous avons montré comment on pouvait perturber ponctuellement ces méthodes, prouvant que cette approche continu/discret était viable. Là encore, pouvoir coupler des méthodes continues et des méthodes discrètes présente de nombreux avantages et permet de pouvoir développer des modèles plus complets et plus complexes.

Dans un troisième chapitre, nous avons appliqué les éléments présentés ci-dessus dans des cas concrets. La thématique générale de ces applications est la bio-économie maritime. Nous avons étudiés l'apport des systèmes multi-agents dans le cadre de la simulation de la dynamique de pêche. De plus, en tant qu'exercice de style nous avons repris les travaux sur QSS1 et QSS2 pour montrer qu'on pouvait mettre en œuvre la notion d'aire marine protégée à l'aide de ces approches. De plus, nous avons repris l'extension sur les équations aux différences pour simuler la dynamique des pêcheries dans le cadre du projet ANR CHALOUPE. Enfin, le modèle Ecomeristem a été présenté. Ce modèle est un modèle de plante s'appuyant sur un système dynamique. Toutes ces applications montrent que le travail précédent est opérationnel et peut être proposé dans la cadre de collaborations extérieures avec d'autres instituts de recherche.

Ces travaux montrent comment on a pu intégrer des nouveaux formalismes à DEVS et par là même à la plate-forme VLE. Mais au delà de ces apports, on a permis de coupler des modèles hétérogènes ce qui ouvre de nouvelles voies dans le domaine. En effet, les premiers couplages réalisés étaient, dans la plupart des cas, des couplages dits « légers », c'est à dire par les fichiers, puis il a été réalisé des couplages dits « techniques » en imposant un langage de programmation unique comme langage de développement des modèles (on peut citer, par exemple OpenAlea [Pradal *et al.*, 2008] qui utilise Python ou bien encore AnyLogic [ANY, 2008] qui utilise Java). Grâce à VLE on dispose d'un cadre formel et opérationnel qui permet de faire des couplages de modèles hétérogènes que ce soit par le paradigme sur lequel ils s'appuient ou le langage qui les implémente. Les apports réalisés dans ce document ouvrent donc la voie au couplage de modèles à base d'agents, d'équations aux différences, etc. ; permettant ainsi de proposer une nouvelle gamme d'outils aux modélisateurs et thématiciens des différents domaines.

Au delà de ces aspects, ces travaux conduisent à une réflexion beaucoup plus profonde de l'utilisation des modèles dans nos simulations. En effet, notre approche, dans l'utilisation des modèles a toujours été guidée par l'existant que l'on peut trouver dans la littérature. Par exemple, pour simuler de la dynamique de population, on peut utiliser les équations aux différences de Von Bertalanffy, ou bien encore, une simple logistique ; ou bien dans le cadre d'une relation proie/prédateur, on peut utiliser les équations de Lokta-Volterra. Ces nouvelles approches nous permettent de remettre en cause ces approches classiques et d'ouvrir de nouveaux champs de recherche. Pourquoi ne pas envisager de nouvelles définitions dans de nouveaux paradigmes de ces relations ?

Perspectives

Les perspectives de recherches sont nombreuses et variées. En effet, ce domaine est actuellement en plein essor et les problèmes ouverts sont légions. Je présente ci-dessous une liste, pas forcément exhaustive, mais qui se veut la plus complète possible.

Spatialisation et couplage avec les systèmes d'informations géographiques

Dans l'exercice de la simulation de systèmes naturels, on est toujours tenté (voire même obligé dans certains cas) de passer par une spatialisation afin de se rapprocher de la réalité et de prendre en compte les phénomènes spatio-temporels. A ce titre, l'utilisation d'un système d'informations géographiques (SIG) [Jones, 1997] se justifie naturellement. Néanmoins, si les SIG proposent souvent des langages de scripts pour développer des modules à l'intérieur de leurs outils, ce n'est que très récemment qu'ils proposent des facultés de couplage avec le monde extérieur⁵ et encore, le plus souvent avec de solutions propriétaires.

Pour sortir de cette ornière, une première approche, là encore classique a été de passer par les fichiers et donc de réaliser un couplage « léger » par les données. Plusieurs travaux ont déjà été réalisés dans ce sens, par exemple : le mémoire de S. Schneider [Schneider, 1998], la thèse de D. Urbani [Urbani, 2006] et les travaux du groupe Cormas [CORMAS, 2003].

Si cette approche a le mérite indéniable de marcher et de rendre opérationnel le couplage, elle ne permet pas de pouvoir construire des modèles de manière formelle. L'idée, ici, serait donc de pouvoir créer une extension à DEVS qui permette de définir les modèles élémentaires à la construction d'une spatialisation. Celle-ci nous permettrait de pouvoir tirer parti des caractéristiques des SIG tout en restant dans le cadre DEVS, c'est à dire, tout en pouvant les coupler avec d'autres modèles venant d'autres paradigmes.

5. voir <http://www.esrifrance.fr/> par exemple

Au delà de ces aspects de couplage, il conviendrait de développer un langage de haut niveau permettant de décrire ces aspects spatiaux. Ce langage qui pourrait être sous la forme un XML permettrait de décrire des phénomènes spatio-temporels et leur dynamique. Cette dynamique pourrait soit être embarquée dans un modèle DEVS soit dans un modèle, certes extension DEVS, mais provenant du monde des SIGs. Je pense que pour mener à bien ce travail, il faut aller voir du côté des travaux de nos collègues physiciens qui ont déjà énormément travaillé sur les problèmes de spatialisation.

Développement d'outils autour du graphe de connexion

Considérons le modèle ci-dessous :

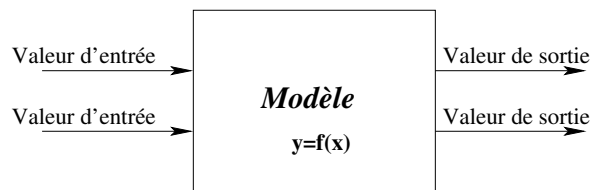


FIGURE 4.1 – Un modèle atomique

Ce modèle, classiquement, prend un certain nombre de données en entrée et produit un certain nombre de valeurs en sortie. Ces valeurs d'entrée et de sortie peuvent être de type différents : des entiers, des réels, des booléens, des fractions, des types définis par l'utilisateur, etc.

Lorsque l'on connecte ce modèle avec d'autres modèles du même type, on crée donc, toujours classiquement, un graphe de connexion de modèles. Ce graphe de connexion représente donc un flux d'informations qui peuvent transiter par les ports d'entrée et les ports de sortie.

La problématique ici vient du fait que le contenu des messages échangés entre les modèles manque, jusqu'à présent, de sémantique. En effet, qui n'a jamais cherché pendant des heures pourquoi une simulation ne marchait pas alors que, formellement, tout était correct, mais un modèle en entrée prenait une valeur en kilo, par exemple, alors qu'il recevait des grammes, faussant ainsi toute la simulation.

Je pense qu'il serait intéressant et profitable de pouvoir augmenter la sémantique de ces messages échangés de manière à pouvoir faire une vérification de l'intégrité au niveau du graphe de modèles. Cette vérification ne serait faite que lors de la phase de conception et non pas pendant la simulation car ce mécanisme risque d'être assez coûteux en temps. On peut, par contre, imaginer une vérification « à la volée » lors que la création dynamique d'un modèle.

Cet exemple marche au niveau des unités, mais on peut tout à fait imaginer ajouter d'autres propriétés pour aider à la conception de modèles. Par exemple : les pré-requis permettant le calcul, la famille, le type, etc.

Unification de la modélisation

Les plate-formes actuelles utilisent bien souvent des descriptions des modèles, de la simulation et des plans d'expérience basées sur une ontologie transcrite en XML. Par exemple le fichier VPZ pour VLE (voir [Quesnel, 2006] page 191) ou bien encore EcoTrop⁶ (voir [Luquet *et al.*, 2006]) dans sa version avec son fichier XML ou bien encore APSIM⁷.

6. <http://ecotrop.cirad.fr>

7. <http://www.apsim.info/apsim/>

Toutes ces plate-formes proposent des interfaces graphiques permettant de construire ces modèles et ces simulations. Or, l'expérience montre que ce type d'interface graphique montre assez rapidement ses limites dans le cas de modèle complexes. Ce problème étant un des facteurs le plus limitant concernant la prise en main des plate-formes de simulation par les thématiciens.

Continuer dans cette voie paraissant illusoire, il me semble intéressant de pouvoir proposer un langage de plus haut niveau qui abstrait cette complexité. En utilisant ce langage et un compilateur qui générerait le fichier adéquat pour la plate-forme concernée on pourrait ainsi avoir une augmentation significative de l'utilisation des plate-formes. Enfin, cette approche permettrait de pouvoir échanger des modèles entre plate-forme beaucoup facilement qu'à l'heure actuelle. En effet, dans la mesure où ce langage de haut niveau sera « plate-forme indépendant », il suffit d'écrire un compilateur pour chaque plate-forme de simulation cible. Ceci permettrait d'aller plus loin que les travaux réalisés dans le cadre du réseau MEXICO [MEXICO, 2009] [Richard *et al.*, 2013].

Vers une nouvelle approche de la modélisation des systèmes naturels

Dans les modèles que nous avons déjà mis en œuvre, les comportements, ou les réactions, étaient activés, classiquement par des fonctions avec des seuils qui permettaient de déclencher des comportements avec des niveaux plus ou moins importants. Ces approches, très classiques, sont à rapprocher de l'automate de K. Lorenz, comme illustré en Figure 4.2.

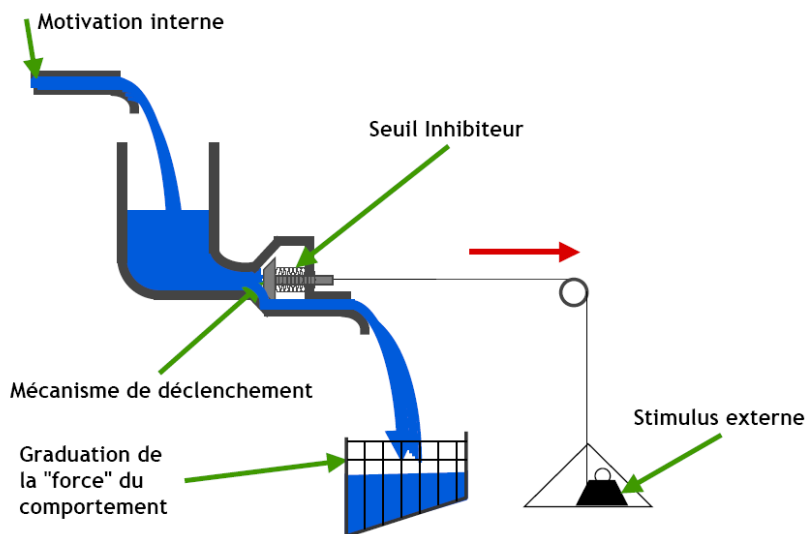


FIGURE 4.2 – Automate de K. Lorenz

Or, il apparaît que, dans les systèmes naturels, cette analogie n'est pas toujours vraie. En effet, ces systèmes, tels une plante, sont en permanence dans un équilibre instable tout en essayant de réaliser leur but final. Par des observations, on peut se rendre compte qu'il y a en permanence des ajustements entre les différents organes de la plante. Pour se faire, on peut imaginer que l'approche individus-centrés est un bon candidat pour représenter ces phénomènes. Mais il faut pouvoir activer des comportements à différent niveaux d'intensité. Cela pourrait être fait en utilisant les ensembles flous.

Un sujet de stage de master recherche pour étudier ce qui a déjà été fait dans ce domaine a été rédigé. En fonction des résultats de ce stage, nous envisageons de proposer un sujet de thèse au CIRAD sur ce qui pourrait être une nouvelle approche de la modélisation et la simulations de systèmes naturels.

Applications

Je finirai ce document par la partie applicative car les applications que nous pouvons faire de nos concepts et développements informatiques sont une part importante de notre travail qui se situe à l'interface de plusieurs champs disciplinaire. Au delà d'être une source de financement non négligeable, ces applications nous permettent de confirmer (ou, malheureusement parfois, d'infirmer) des hypothèses prises au niveau plus théorique. Enfin, elles permettent de faire un travail d'abstraction qui conduit à identifier des mécanismes ou des processus plus généraux. Ce travail étant, par retour, profitable à tous les utilisateurs.

La Figure 4.3 présente tous les modèles sur lesquels j'ai travaillé depuis 1998.

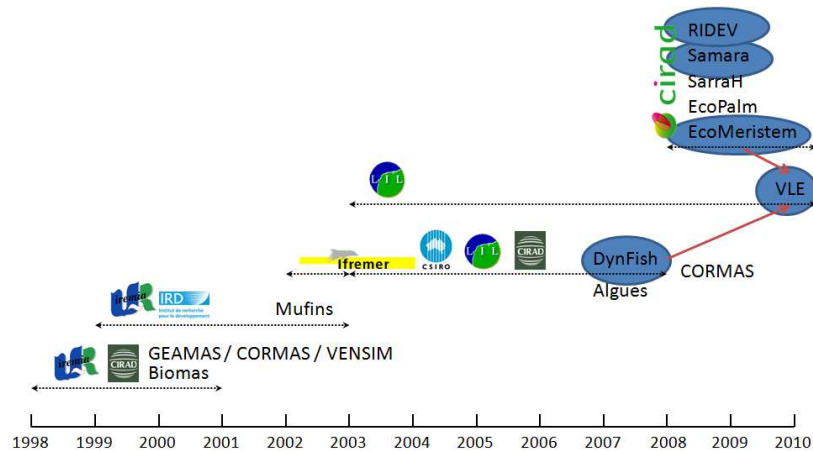


FIGURE 4.3 – L'ensemble de modèles

Les modèles entourés de bleu sont ceux qui, au jour d'aujourd'hui, ont une existence au sein de VLE. RIDEV et SAMARA ne sont pas complètement réécrits sous VLE. À la différence d'Ecomeristem qui a été entièrement porté dans sa version « phase végétative » au sein de VLE.

Cette réécriture de modèle a été profitable à deux niveaux :

- Pour VLE : Ecomeristem a nécessité l'utilisation de l'extension équations aux différences pour gérer les processus de la plante, l'extension automates à états finis pour gérer les différents états de la plante (MORPHOGENESIS, ELONGATION, PANICLE INITIATION, FLOWERING, ...) et de DS-DEVS [Barros, 1996] [Barros, 1997] pour gérer l'apparition dynamique des organes, ces derniers étant vu comme des modèles à instancier. Il a donc été possible de tester, séparément la correction et la complétude de ces extensions et,

enfin, de tester que leur couplage était opérationnel.

- Pour Ecomeristem : on dispose maintenant d'un modèle qui s'exécute dans un environnement moderne et formellement défini. Les temps de calculs sont en très net amélioration et l'utilisation de RVLE permet de piloter les simulations depuis R et, par conséquent, de profiter de cet environnement pour réaliser des études statistiques des sorties, des analyses de sensibilité et des estimations de paramètres.

Néanmoins, le modèle n'est pas encore complet et nous devons développer maintenant le cycle complet et vérifier la généralité du modèle ainsi terminé. Le développement du cycle complet sera réalisé dans le cadre du projet Grand Emprunt Biomass For the Future qui va commencer d'ici la fin de l'année. La généralité du modèle sera éprouvée dans le cadre du projet ANR Delicas dans lequel Ecomeristem simule de la canne à sucre et dans le cadre du FP7, nous pourrions tester l'adaptation d'Ecomeristem au sorgho.

De plus, l'utilisation de VLE et RVLE, nous permettra d'utiliser de manière systématique, via les outils MEXICO, l'analyse de sensibilité des modèles afin de vérifier la surface de réponse du modèle en fonction de la variation des valeurs des paramètres d'entrée. Cette approche ne doit pas être vue comme une sanction sur le fait que le modèle ne marche pas, mais plutôt comme une aide à la modélisation en mettant en exergue les zones du modèle qui semblent ne pas se comporter correctement.

Enfin, nous souhaitons pouvoir utiliser des outils de calcul 3D de balance d'énergie (interception lumineuse, micro-climat, ...) du type Caribu [Chelle *et al.*, 1998]. Une première étude de faisabilité a été réalisée en utilisant la plateforme OpenAlea [Soulie *et al.*, 2010] et bien que cette étude montre que ce couplage fonctionne, elle reste très primaire et décevante. En effet, ce couplage est un couplage « léger » et nous avons utilisé de manière massive les fichiers. Or, Ecomeristem étant un système dynamique, la taille de ces derniers augmente de manière considérable au cours de la simulation. Nous sommes donc passés de 300ms pour 40 jours de simulations à 30mn pour la version 3D (avec au moins 90% du temps passé à lire et écrire les fichiers intermédiaires). Ce qui est inacceptable si on veut réaliser des optimisations avec ce couplage. Grâce à Ecomeristem version VLE, nous pourrions « wrapper » Caribu afin de pouvoir utiliser les résultats de ce dernier en s'affranchissant des problèmes liés aux fichiers.

Dans le cadre de la collaboration avec l'UPR 74 « Recyclage et risque », nous souhaitons développer un ensemble de modèles selon le schéma présenté dans la figure 4.4.

La finalité des recherches menées par l'unité est de proposer des solutions, des méthodes ou des modèles pour recycler les matières organiques et les déchets ? effluents d'élevage, boues d'épuration, effluents de l'agro-industrie, déchets verts et ordures ménagères ? par des transformations et des pratiques agricoles à risques agro-environnementaux contrôlés, en utilisant au mieux le pouvoir épurateur du sol et de la plante, et en raisonnant les complémentarités / oppositions au niveau des territoires.

Dans les pays du Sud où s'exerce le mandat du Cirad, les activités de l'unité contribuent ainsi à inventer une agriculture écologiquement intensive et à mieux comprendre les relations entre l'agriculture et l'environnement pour gérer durablement les espaces ruraux.

Ce schéma représente les deux niveaux d'organisation considérés par l'unité en tant qu'objets d'études :

- Le territoire et l'exploitation sur lequel sont indiqués les flux de matières organiques (MO) entre les principaux acteurs des filières de production, transformation et gestion

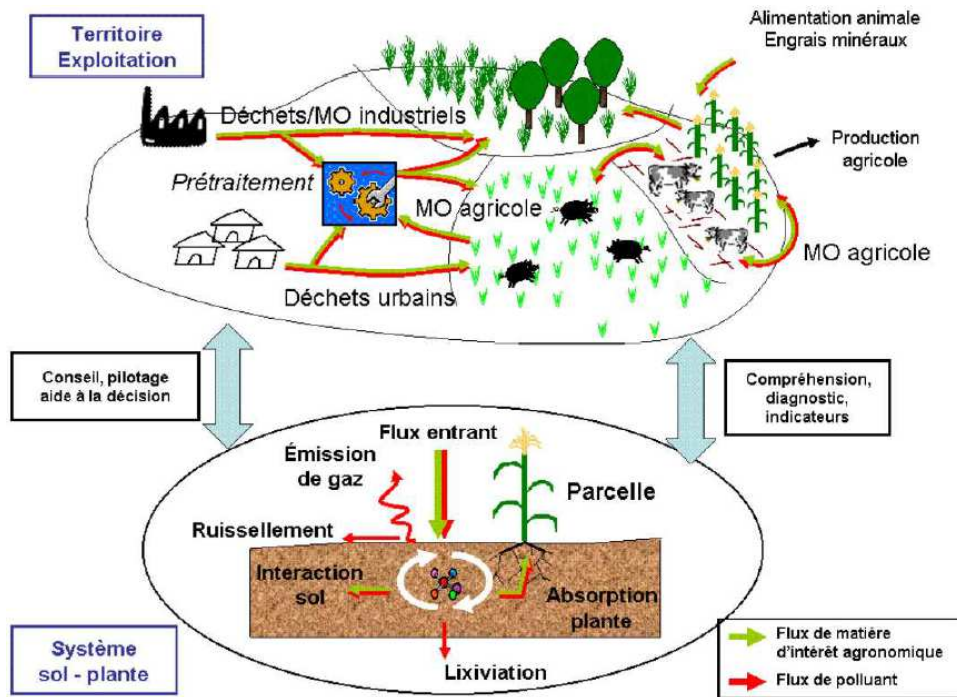


FIGURE 4.4 – Schéma d'intégration des modèles de l'UPR 74

des déchets ;

- Le système sol-plante, niveau privilégié pour étudier les flux susceptibles d'induire des risques pour l'environnement.

L'un des défis scientifiques que l'unité s'est proposé de relever est de développer des méthodes de caractérisation des flux à ces différents niveaux d'organisation et d'intégrer les connaissances acquises pour mieux comprendre, représenter et finalement conseiller les différents acteurs : agriculteurs, conseillers, groupements, industriels, associations, autorité publique.

Cette collaboration nous semble tout à fait intéressante et pertinente car elle va nécessiter de mettre en œuvre des modèles à l'échelle de l'exploitation, à l'échelle de la plante aussi bien qu'à l'échelle biophysique. Toutes ces approches ont déjà été réalisées dans de précédents modèles, mais l'étude des interactions entre ces différentes échelles n'a pas été effectuée.

Annexe A

Curriculum Vitæ détaillé

Adresse et état civil

UMR 1334 AGAP – Amélioration Génétique et Adaptation des Plantes
Département Systèmes Biologiques
CIRAD
Avenue Agropolis Lavalette – TA-A 108/01
34398 Montpellier Cedex 5
Email : jean-christophe.soulie@cirad.fr

41 ans, divorcé, 1 enfant
Nationalité Française

Expériences professionnelles

- 2008–...** : Détaché en tant que Cadre de Recherche au Centre de Coopération Internationale pour la Recherche en Agronomie et le Développement (CIRAD), Département : *Systèmes Biologiques (BIOS)*, Unité Mixte de Recherche 1334 : *Amélioration Génétique et Adaptation des Plantes*
- 2003–...** : Maître de Conférences au Laboratoire d'Informatique Signal et Image de la Côte d'Opale (EA 4491), Université du Littoral Côte d'Opale.
- 2002–2003** : Cadre de Recherche post-doctoral en informatique au Service d'Économie Maritime à l'Institut Français de REcherche pour la MER (IFREMER) à Brest.
- 2000–2002** : Attaché Temporaire d'Enseignement et de Recherche (ATER) à mi-temps à l'Institut de Recherche en Mathématiques et Informatique Appliquées (EA 2525), Université de La Réunion.
- 1997–2000** : Moniteur de l'Enseignement Supérieur à l'Institut de Recherche en Mathématiques et Informatique Appliquée (EA 2525), Université de La Réunion.

Formation

- 2001** : Doctorat de l'Université de la Réunion, spécialité Informatique.

1997 : DESS Informatique, spécialité Communication Homme/Machine. Université d'Avignon et des Pays de Vaucluse.

Recherche

Thèse de doctorat

Titre : Vers une approche multi-environnement pour les agents

Mention : Très Honorable

Soutenance : le 3 décembre 2001 à l'Université de La Réunion devant le jury composé de Messieurs :

- Henri Ralambondrainy, Professeur, Institut de Recherche et Mathématiques et Informatique Appliquées, Université de La Réunion, *Président*.
- Jean-Pierre Müller, Cadre de Recherche – HDR, CIRAD, UPR GREEN, Montpellier, *Rapporteur*.
- Joël Quinqueton, Directeur de Recherche, INRIA Montpellier, *Rapporteur*.
- Christophe Le Page, Cadre de Recherche, CIRAD, UPR GREEN, Montpellier, *Examineur*.
- Jean-Michel Stretta, Docteur d'Etat, Représentant IRD La Réunion, *Examineur*.
- Pierre Marcenac, Professeur, Institut de Recherche et Mathématiques et Informatique Appliquées, Université de La Réunion, *Directeur de Thèse*.

Publications⁸

- ACL : 5
- OS : 6
- INV : 5
- ACTI : 35
- ACTN : 6
- AP : 6

Encadrement Doctoral

- Gauthier Quesnel
 - Titre : Approche formelle et opérationnelle de la multi-modélisation et de la simulation des systèmes complexes. Apports pour la simulation de Systèmes Multi-Agents
 - Début : Novembre 2003
 - Fin : 1^{er} Décembre 2006
 - Co-encadrement : 50% avec E. Ramat (Professeur, LISIC - ULCO)
 - Emploi : Chargé de Recherche, Institut National pour la Recherche en Agronomie, Département Mathématique et Informatique Appliquées, Unité Biométrie et Intelligence Artificielle

8. Selon la classification AERES

- David Versmisse
 - Titre : Intégration de modèles hétérogènes pour la modélisation et la simulation de systèmes complexes : application aux anthroposystèmes marins
 - Début : Septembre 2004
 - Fin : 17 Janvier 2008
 - Co-encadrement : 50% avec E. Ramat (Professeur, LISIC - ULCO)
 - Emploi : Ingénieur Informatique, Allemagne
- Katell Hamon
 - Titre : Evaluation bio-économique des impacts de changements dans les modes de régulation de l'accès aux ressources halieutiques : le cas du passage aux quotas individuels transférables dans les pêcheries côtières australiennes
 - Début : Novembre 2007
 - Fin : Mai 2011
 - Co-encadrement : 25% avec O. Thebaud (IFREMER Brest) et S. Frusher (University of Tasmania, Tasmanian Aquaculture and Fisheries Institute, Hobart, Australie) et R. Little (CSIRO Marine Research, Hobart, Tasmanie, Australie) - Thèse en co-tutelle franco-australienne
 - Emploi : Contrat post-doctoral Institute for Marine Resources & Ecosystem Studies, Wageningen UR, Netherlands
- M. Marzloff
 - Dynamics of rocky reefs and fishers in eastern Tasmania : Modelling using a complex systems approach
 - Début : 2008
 - Fin : 2012
 - Co-encadrement : 20% avec C. Johnson (University of Tasmania, Tasmanian Aquaculture and Fisheries Institute, Hobart, Australie), S. Frusher (University of Tasmania, Tasmanian Aquaculture and Fisheries Institute, Hobart, Australie) et R. Little (CSIRO Marine Research, Hobart, Tasmanie, Australie) - Thèse financée à 100% par l'University of Tasmania

Master Recherche

1. O. Sall
 - Titre : *Gestion du temps et modélisation de l'environnement dans les systèmes multi-agents*
 - Date début : Février 2004 - Date fin : Juin 2004
 - % encadrement : 100
 - Situation actuelle : Thèse Co-tutelle ULCO - Sénégal (Sous la direction de H. Basson)
2. D. Schneider
 - Titre : *Intégration de la théorie des hiérarchies dans un modèle d'agent pour Virtual Laboratory Environment*
 - Date début : Février 2005 - Date fin : Juin 2005
 - % encadrement : 100
 - Situation actuelle : Ingénieur R & D Cap Gémini Lille
3. Romain Hennuyer
 - Titre : *Théorie des jeux, théories des votes pour la modélisation et la simulation de comportements d'agents*

- Date début : Février 2007 – Date fin : Juin 2007
- % encadrement : 50 avec E. Ramat (Professeur, LIL - ULCO)

Bénéficiaire de la Prime d'Encadrement Doctoral et de Recherche (PEDR) du 1^{er} Octobre 2007 à Juillet 2008. Arrêt de la PEDR suite à mon détachement au CIRAD.

Master Professionnel

Encadrement de 18 étudiants en projet de Master Professionnel 2^{ème} année et de 20 étudiants en projet de Master Professionnel 1^{ère} année.

Responsabilités administratives

- Chef de l'équipe « Mathématiques et Informatique ». (5 personnes : 2 chercheurs + 2 chercheurs associés + 1 technicien) UPR 104. Département BIOS. CIRAD
Année : De Décembre 2008 à Juin 2009
- Directeur du Département d'Informatique (20 permanents + 10 doctorants – Filière de 300 étudiants environ)
Année : De 2005 à 2008
- Directeur d'Etude : Master Professionnel 1^{ère} Année « Ingénierie des Systèmes Informatiques Distribués »
Année : 2003, 2004
- Membre suppléant de la CSE 27^{ème} Section – Laboratoire d'Informatique du Littoral (EA 4029), Université du Littoral Côte d'Opale
Année : De 2004 à 2008
- Membre extérieur de la CSE 27^{ème} Section – Systèmes Physique de l'Environnement (UMR CNRS 6134), Université de Corse
Année : De 2007 à 2008
- Membre du Jury « Validation des Acquis Extérieurs »
Année : De 2005 à 2008
- Membre du Comité d'Évaluation de l'Université
Année : De 2005 à 2008

Responsabilités Scientifique et Contrats de Recherche

1. **Convention N°03/2 210 426. Contrat de Recherche IFREMER - Département d'Economie Maritime et Laboratoire d'Informatique du Littoral - Université du Littoral Côte d'Opale**
2. **Contrat d'Incitation Universitaire N° 953 R321. Contrat de Recherche IFREMER - Département d'Economie Maritime et Laboratoire d'Informatique du Littoral - Université du Littoral Côte d'Opale**
3. **Contrat d'expertise. IFREMER - Département d'Economie Maritime et Laboratoire d'Informatique du Littoral - Université du Littoral Côte d'Opale**

4. **Contrat de Partenariat - Partenaire n° 4. IFREMER - Département d'Economie Maritime, Halieutique, Ecologie et Modèles, Halieutique Tropicale ; Institut National de la Recherche Halieutique (Maroc) ; Université de Bretagne Occidentale ; CNRS Muséum ; World Fish Center ; IRD (Ecosystèmes d'upwelling) ; Ecole Nationale d'Ingénieur de Brest et Laboratoire d'Informatique du Littoral - Université du Littoral Côte d'Opale**
5. **Projet Agence Nationale de la Recherche : Entomochick. Contrat ANR avec le CIRAD Ile de la Réunion – UMR 53 Peuplements Végétaux et Bio-agresseurs en Milieu Tropical, Intitut de Recherche en Mathématiques et Informatique Appliquées - Université de La Réunion et Laboratoire d'Informatique du Littoral - Université du Littoral Côte d'Opale**

Rayonnement

National

Comité Scientifique

- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 07) – Carcassonne – Octobre 2007
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 08) – Brest – Octobre 2008

Relecture d'Articles

- Revue Technique des Sciences Informatiques (TSI) – Hermès. Numéro spécial ASTI – Thèse jeunes-chercheurs de TSI
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 05) – Calais – Novembre 2005
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 06) – Annecy – Novembre 2006
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 07) – Carcassonne – Octobre 2007
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 08) – Brest – Octobre 2008

Organisation de conférences

- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 98) – Saint Denis – Novembre 1998
- Journées Francophones pour les Systèmes Multi-Agents (JFSMA 05) – Calais – Novembre 2005
- STIC et Environnement – Calais – 2009

Réseaux nationaux

- Représentant CIRAD au sein du réseau « Méthodes pour l'Exploration Informatique des modèles Complexes » (<http://www.reseau-mexico.fr>)

Cours

- Initiation à la modélisation et à la simulation multi-agents de systèmes naturels. École Nationale Supérieure d'Agronomie. DAA Halieutique. Rennes, Janvier 2003
- Modélisation et simulation multi-agents. Application à la bio-économie maritime. Ecole Nationale Supérieure d'Agronomie. Master Recherche. Rennes, Décembre 2005
- Systèmes Naturels, Modélisation et simulation multi-agents. Master 2 Politique des ressources de l'agriculture, de la mer et de l'environnement Ecole Nationale Supérieure d'Agronomie. Master Recherche. Rennes, Décembre 2009
- Modélisation et Simulation. Définitions et exemples. Master 2 Amélioration des Plantes et Ingénierie végétale Méditerranéennes et Tropicales, SupAgro Montpellier, Décembre 2011

Participation à des programmes de Recherche

- Programme Palangre Réunion 2002
- Défi Golfe de Gascogne 2002/2005
- Programme Seine Aval II 2002/2003
- Projet ANR DELICAS 2008/2011 (Programme de Génomique Axe thématique 2 - Génomique des Plantes - Edition 2008) : Association mapping and model phenotyping for the characterization of molecular markers associated with sugarcane yield formation and limitation
- ATPd ORYTAGE 2008/2011 (Étude d'association pour l'adaptation aux contraintes agro-écologiques)
- Convention PTSmart⁹ 2008/2011

International

Participation à des programmes de Recherche

- Projet Generation Challenge Program Rice Productivity
- Projet Developing Rice and Sorghum Crop Adaptation Strategies for climate change in vulnerable environments in Africa
- Projet Africa Rice
- Amélioration des prévisions climatiques pour l'agriculture (PICREVAT)

Relecture d'Articles

- Annals of Software Engineering
- Journal of Mathematics & Computer Modelling
- Ecological Modelling

9. <http://www.smart-tbk.com/>

- Ecological Economics
- Environmental Modelling & Software
- International Congress on Modelling and Simulation (2003, 2005, 2007)

Organisation de sessions dans des conférences

- Social, Economic and Ecological Modelling of Fisheries and Marine Resources. R. Little (CSIRO - Hobart, Tas, AUS) et J.-C. Soulié (LIL - Université du Littoral Côte d'Opale). Session Spéciale à International Congress on Modelling and Simulation 2007, Auckland, NZ. 20 papiers sélectionnés.

Organisation de conférences

- XVth European Association of Fisheries Economics Conference (EAFE 03) - Brest - Mai 2003

Réseaux Internationaux

- Human Exploited Ecosystem Modelling with Agent (HEMA network) : programme de collaboration Franco-Australien (CSIRO - Australian National University - CIRAD - IFREMER - INRA - CEMAGREF - Laboratoire d'Informatique du Littoral, ULCO)

Séminaires et Cours

- Short Term Fisheries Dynamics Modelling With Agent. Tasmanian Aquaculture and Fisheries Institute – Tasmalian University – Hobart, Tasmanie, 12 Juillet 2005
- Multi-Agent Based Model and Simulation. Tasmanian Aquaculture and Fisheries Institute – Tasmalian University – Hobart, Tasmanie, 13-14 Juillet, 2005
- RISOCAS Crop Modeling Workshop – Cotonou, Bénin, 20-27 Avril 2009

Annexe B

Liste de Publications

B.1 Articles dans des revues internationales ou nationales avec comité de lecture – ACL

- Simulation of Inflorescence Dynamics in Oil Palm and Heuristic Estimation of Environment-Sensitive Phenological Phases. A Model Based Analysis. J.-C. Combres, B. Pallas, L. Rouan, J.-P. Caliman, J.-C Soulié and M. Dingkuhn. *Functional Plant Biology*. 2012. Accepté, Sous presse
- Modelling rice plant vegetative morphogenesis plasticity as a result of carbon and water metabolism regulation under drought. D. Luquet, J.-C. Soulié, A. Clément-Vidal , M. Dingkuhn. *Journal of Agronomy and Crop Science*. 2011. Accepté, Sous presse
- Virtual Laboratory Environment : un environnement de multi-modélisation et de simulation de systèmes complexes. G. Quesnel, E. Ramat, J.-C. Soulié, D. Duvivier and R. Duboz. *Studia Universalis*, 10 (1) : 205-234.
- Modelling fleet response in regulated fisheries : an agent-based approach. J.-C. Soulié and O. Thébaud. *Journal of Mathematical and Computer Modelling*. Elsevier, Vol. 44, 5-6 :553-564, 2006
- JAAFAAR : A Web-Based Multiagent Toolkit for Collective Research. S. Calderoni and J.-C. Soulié. *Annals of Software Engineering, Special Issue on Object-Oriented Web-Based Software Engineering*. Vol. 13, pp. 265-283, Kluwer Academic Publisher, 2002

B.2 Ouvrages scientifiques – OS

- Développement d'un environnement générique pour piloter l'exploration numérique de modèles : la boîte à outils MEXICO. H. Richard, H. Monod, J. Wang, J. Couteau, N. Dumoulin, B. Poussin, J.-C. Soulié et E. Ramat. Quae Éditions, sous presse
- Modeling and Simulation of Living Systems as Systems of Systems. R. Duboz and J.-C. Soulié. B. P. Zeigler eds. Springer, sous presse
- L'approche par simulation de la dynamique de la capturabilité de l'espadon par la palangre de surface : 1ère Partie - La modélisation dans la compréhension des déplacements des grands pélagiques exploités par la pêche. D. Guyomard, J.-C. Soulié, M. Desruisseaux, J.-M. Stretta et M. Petit. *Halieutique et environnement océanique*. M. Petit et F. Huynh Eds. pp. 210-230. IRD Editions, 2006

- L’approche par simulation de la dynamique de la capturabilité de l’espadon par la palangre de surface : 2ème Partie - Un applicatif de modélisation individus centrée basé sur une architecture de type multi agents (Mufins). J.-C. Soulié, D. Guyomard, M. Desruisseaux, J.-M. Stretta et M. Petit. *Halieutique et environnement océanique*. M. Petit et F. Huynh Eds. pp. 230-282. IRD Editions, 2006

B.3 Conférences invitées dans un congrès national ou international – INV

- Formal model coupling using DEVS. J.-C. Soulié, G. Quesnel, E. Ramat. Mathematical Modeling of Plant Development. Mathematical Bioscience Institute. 27 Septembre 2010-1 Octobre 2010. Columbus, Ohio, USA
- Modelling approach enabling integrative studies of rice morphogenesis regulation. J.-C. Soulié, and D. Luquet. Mathematical Modeling of Plant Development. Mathematical Bioscience Institute. 27 Septembre 2010-1 Octobre 2010. Columbus, Ohio, USA
- 1st HEMA Workshop. Melbourne, Australie, Juin 2003
- 2nd HEMA Workshop. Canberra, Australie, Juin 2004
- InVitro Workshop. CSIRO Marine Research, Hobart, Australie, Décembre 2005

B.4 Communications avec actes dans un congrès International – ACTI

- Functional-structural plant modeling to support complex trait phenotyping : case of rice early vigour and drought tolerance using Ecomeristem model. D. Luquet, M.-C. Rebolledo, and J.-C. Soulié. In *Proceedings of the fourth international symposium on plant growth modelling, simulation, visualization and application*, Shangai, China, 31 October-3 Novembre 2012. IEEE Computer Society Press (to be printed)
- A management-support framework for subtidal rocky-reef communities on the east coast of Tasmania. M.Marzloff, C. R Johnson., R. L. Little, S. D. Ling, J.-C. Soulié, and S. D. Frusher. In *Proceedings of the MODSIM Congress*, Perth, Australia 12-16 December 2011
- Sensitivity analysis of light interception to geometrical traits of apple trees : an in silico study based on MAppleT model. L. Han, J.-C. Soulié, F. Boudon, D. Da Silva, T. Cokelaer, C. Pradal, L. Rouan, and E. Costes. In *Proceedings of Acta Horticulturae. Montreal*, Canada. July 2011
- Modelling phase shifts in Tasmanian rocky-reef communities. Marzloff Martin P., Johnson Craig R., Little L. Richard, Ling Scott D., Soulié J.-C., Frusher Stewart D. In *Proceedings of the 2010 ICES Annual Science Meeting*, September 2010, Nantes, France
- ECOPALM. A model to understand the complex phenology of mature oil palm. Combres J.C., Soulie J.C., Rouan L., Braconnier S., Dingkuhn M. In Wery Jacques (ed.), Shili-Touzi I. (ed.), Perrin A. (ed.). *Proceedings of Agro 2010 : the XIth ESA Congress*, August 29th - September 3rd, 2010, Montpellier, France. Montpellier : Agropolis international, p. 559-560. ESA Congress. 11, 2010-08-29/2010-09-03, Montpellier, France.
- Building modular FSPM under OpenAlea : concepts and applications. Fournier C., Pradal C., Louarn G., Combes D., Soulie J.C., Luquet D., Boudon F., Chelle M. In : Dejong Theodore M. (ed.), Da Silva David (ed.). In *Proceedings of the 6th International Workshop*

- on *Functional-Structural Plant Models, FSPM 2010*, September 12-17, 2010, University of California, Davis (USA). Davis : University of California, p. 109-112. International Workshop on Functional-Structural Plant Models (FSPM). 6, 2010-09-12/2010-09-17, Davis, Etats-Unis.
- Predicting crop productivity and adapting the rice plant to changing climates : The importance of modeling. Lafarge T., Luquet D., Baron C., Heinemann A., Rebolledo, Julia C., Muller B., Rouan L., Soulie J.C., Dingkuhn M. In : *Proceedings of the 28th International Rice Research Conference, Climate Change and Rice Agriculture*, November 08-12, 2010, Hanoi, Vietnam. s.l. : s.n., 1 p. International Rice Research Conference. 28, 2010-11-08/2010-11-12, Hanoi, Vietnam.
 - Does development rate drive early growth vigour in rice ? Implications for modelling and crop improvement. Luquet D., Rebolledo, Soulie J.C., Rouan L., Dingkuhn M. In : Wery Jacques (ed.), Shili-Touzi I. (ed.), Perrin A. (ed.). In *Proceedings of Agro 2010 : the XIth ESA Congress*, August 29th - September 3rd, 2010, Montpellier, France. Montpellier : Agropolis international, p. 547-548. ESA Congress. 11, 2010-08-29/2010-09-03, Montpellier, France.
 - Modelling plant morphogenesis and source-sink processes to support crop performance phenotyping : application of Ecomeristem model to sugar cane. Luquet D., Soulie J.C., Huet C., Giraud A., Martiné J.F. In : Wery Jacques (ed.), Shili-Touzi I. (ed.), Perrin A. (ed.). In *Proceedings of Agro 2010 : the XIth ESA Congress*, August 29th - September 3rd, 2010, Montpellier, France. Montpellier : Agropolis international, p. 579-580. ESA Congress. 11, 2010-08-29/2010-09-03, Montpellier, France.
 - Exploring the feasibility of sugarcane phenotyping using crop models with contrasted climatic conditions in Réunion Island. Martiné J.F., Gozé E., Luquet D., Thong-Chane A., Houles, Soulie J.C., Rouan L., Nibouche S. In : Wery Jacques (ed.), Shili-Touzi I. (ed.), Perrin A. (ed.). In *Proceedings of Agro 2010 : the XIth ESA Congress*, August 29th - September 3rd, 2010, Montpellier, France. Montpellier : Agropolis international, p. 577-578. ESA Congress. 11, 2010-08-29/2010-09-03, Montpellier, France.
 - The DELICAS project : model assisted phenotyping in sugarcane for the identification of marker-trait associations. Nibouche S., Martiné J.F., Luquet D., Gozé E., Rouan L., Costet L., D'Hont A., Soulie J.C., Thong-Chane A. In : Wery Jacques (ed.), Shili-Touzi I. (ed.), Perrin A. (ed.). In *Proceedings of Agro 2010 : the XIth ESA Congress*, August 29th - September 3rd, 2010, Montpellier, France. Montpellier : Agropolis international, p. 541-542. ESA Congress. 11, 2010-08-29/2010-09-03, Montpellier, France.
 - Le taux de développement entraîne-t-il la vigueur précoce chez le riz ? Implications pour la modélisation et l'amélioration des cultures. Rebolledo, Dingkuhn M., Soulie J.C., Rouan L., Luquet D. In : Kiepe Paul (ed.), Diatta M. (ed.), Millar D. (ed.). *Innovation and partnerships to realize Africa's rice potential : Abstracts of Africa Rice Congress*, 22 to 26 March 2010, Bamako, Mali. Bamako : IER, p. 190-191. Africa Rice Congress, 2010-03-22/2010-03-26, Bamako, Mali.
 - Feedbacks between plant microclimate and morphogenesis in fluctuating environment : analysis for rice using Ecomeristem model coupled with 3D plant and energy balance computation tools in OpenAlea platform. Soulie J.C., Pradal C., Fournier C., Luquet D. In : Dejong Theodore M. (ed.), Da Silva David (ed.). *Proceedings of the 6th International Workshop on Functional-Structural Plant Models, FSPM 2010*, September 12-17, 2010, University of California, Davis (USA). Davis : University of California, p. 138-140. International Workshop on Functional-Structural Plant Models (FSPM). 6, 2010-09-12/2010-09-17, Davis, Etats-Unis.

- EcoPalm : A new model simulating seasonal variation of palm yield : I. Day length sensitivity of flowering and competition among sinks explain bunch number. Combres J.C., Mialet-Serra I., Bonnal V., Soulie J.C., Rouan L., Braconnier S., Dingkuhn M. *In : Proceedings of the 16th International Oil Palm Conference and Expopalma. Challenges in sustainable oil palm development*, 22 to 25 September 2005, Cartagena de Indias, Colombia. s.l. : s.n., [14] p. International Oil Palm Conference and Expopalma. 16, 2009-09-22/2009-09-25, Carthagène des Indes, Colombie.
- Modelling rice phenotypic plasticity in diverse climates using EcoMeristem : Model evolution and applications to rice improvement. Luquet D., Soulie J.C., Lafarge T., Dingkuhn M., Wassmann R. *In : Hasegawa Toshihiro (ed.), Sakai Hidemitsu (ed.). Crop production under heat stress : monitoring, impact assessment and adaptation. Proceedings of the MARCO Symposium*, Tsukuba, Japan, 5-9 October 2009. Tsukuba : NIAES, p. 74. Marco Symposium "Challenges for Agro-Environmental Research in Monsoon Asia". 1, 2009-10-05/2009-10-09, Tsukuba, Japon.
- Model assisted phenotyping of processes involved in rice response to drought : case study of a tropical japonica population during vegetative phase. Rebolledo, Forest M., Seranuch C., Soulie J.C., Rouan L., Dingkuhn M., Fabre D., Luquet D. *In : InterDrought-III : Proceedings of the 3rd International Conference on Integrated Approaches to Improve Crop Production Under Drought Prone Environments*. [S.l.] : s.n., p. 224-225. InterDrought. 3, 2009-10-11/2009-10-16, Shanghai, Chine.
- Model assisted phenotyping of rice response to vegetative drought : case study of a collection of 203 tropical japonica genotypes. Rebolledo, Forest M., Seranuch C., Soulie J.C., Rouan L., Dingkuhn M., Fabre D., Luquet D. *In : Proceedings of the 3rd International Conference on Integrated Approaches to Improve Crop Production Under Drought Prone Environments : InterDrought*, October 11-16, 2009, Shanghai, China. s.l. : s.n., 1 p. InterDrought. 3, 2009-10-11/2009-10-16, Shanghai, Chine.
- Developing a bioeconomic simulation tool of fisheries dynamics : a case study. D. Vermisse, C. Macher, E. Ramat, J.-C. Soulié, and O. Thebaud. *In Proceedings of the International Congress on Modelling and Simulation*. Christchurch. New Zealand. Decembre 2007. International Society for Computer Simulation.
- Fishing through fish communities : a simple bio-economic model. O. Thebaud, and J.-C. Soulié. *In Proceedings of the International Congress on Modelling and Simulation*. Christchurch. New Zealand. Decembre 2007. International Society for Computer Simulation.
- Coupling DEVS modelling and supervised learning for optimization of a bioeconomic system. R. Hennuyer, E Ramat, and J.-C. Soulié. *In Proceedings of the International Modeling and Simulation Multiconference*. Bueno Aires, Argentina, 8 - 10 February 2007. International Society for Computer Simulation
- Simulation of renewable resources exploitation : between heuristic and decision making models. J.-C. Soulié, and O. Thébaud. *In Proceedings of Joint Conference on Multi-Agent Modeling and Environmental Management*. Bourg-Saint-Maurice, France, 2005
- The economic consequences of producer responses to accidental pollution : a fishery example. O. Thebaud, O. Guyader, J.-C. Soulié, and A. Tétard. *In Proceedings of the 8th Biennial Conference of the International Society for Ecological Economics*. Montreal, Canada, 2004
- The bio-economic implications of fisheries responses to spatial and temporal regulatory controls. O. Thébaud and J.-C. Soulié. *In Proceedings of the World Fisheries Congress*. Vancouver, Canada, 2004
- Towards a multiagent simulation model for the analysis of short term fisheries dynamics :

- a case study. O. Thébaud and J.-C. Soulié. In *Proceedings of the 15th Annual Conference of the European Association of Fisheries Economists*. Brest, France, 14 - 16 May 2003
- Short term analysis of fisheries dynamics : a multi-agent simulation approach. O. Thébaud and J.-C. Soulié. In *Proceedings of the International Congress on Modelling and Simulation*. Theme : Integrative Modelling of Biophysical Social and Economic Systems for Resource Management Solutions. Townsville, Queensland, Australia, 14 - 17 July 2003. International Society for Computer Simulation
 - A multi-agent simulation model for short term analysis of fisheries dynamics. J.-C. Soulié and O. Thébaud. In *Proceeding of the 4th International Workshop on Agent Based Simulation*, J.-P. Müller Eds., pp. 113-119, Montpellier, France, 28-30 April 2003. Society for Computer Simulation - Bvba Publisher
 - A Multiagent System for Modelling Individual Catchability of Swordfish (*Xiphias Gladius*) targeted by Longliners in the South-western Indian Ocean. D. Guyomard and J.-C. Soulié. In *Proceedings of the International Congress on Modelling and Simulation*. Theme : Integrative Modelling of Biophysical Social and Economic Systems for Resource Management Solutions. Townsville, Queensland, Australia, 14 - 17 July 2003. International Society for Computer Simulation
 - Proposition of a Multiagent Framework for Simulating Individual Fishery Agents Behavior in a Dynamic GIS-like Software. D. Guyomard, J.-C. Soulié and M. Desruisseaux. In *Proceedings of the 2002 World Conference on Natural Resource Modelling*. Theme : Modelling Natural and Biotic Resources in a Changing Planet, Lesvos, Greece, June 23 - 26, 2002
 - MUFINS : a Multiagent Toolkit for Simulating Pelagic Fish Behavior in Distributed Multiple Environments. JC. Soulié, D. Guyomard and M. Desruisseaux. In *Proceedings of the 13th European Simulation Symposium*, pp. 885-890, Marseille, France, 18-20 October 2001. SCS Europe Bvba Publisher
 - A Framework to Model Multiple Environments in Multiagent Systems. J.-C. Soulié and P. Marcenac. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, pp. 383-393, Melbourne, Australia, August 28-September 1, 2000. Lecture Notes in Artificial Intelligence, Vol. 1886, Springer Verlag
 - Coupling Multiagent Systems and Environmental Data. J.-C. Soulié, S. Barbin and P. Marcenac. In *Proceedings of the 33rd International Conference on Technology of Object-Oriented Languages and Systems*, pp. 265-273, Melbourne, Australia, 22-25 November 1999. IEEE Computer Society Press
 - Environmental Simulations using Multiagent Systems. J.-C. Soulié and P. Marcenac. In *Proceedings of the IEEE International Conference on Information, Intelligence and Systems*, pp. 668-673, Washington, DC, USA, November 1999. IEEE Computer Society Press
 - Modeling the Environment for Multiagent Simulations. J.-C. Soulié and P. Marcenac. In *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis*, Orlando, USA, August 1999. International Institute of Informatics and Systemic
 - GEAMAS v2.0 : an Object Oriented Platform for Complex Systems Simulations. J.-C. Soulié, P. Marcenac, S. Calderoni and R. Courdier. In *Proceedings of the 26th International Conference on Technology of Object-Oriented Languages and Systems*, pp. 230-242, Eds. M. Singh, B. Meyer, J. Gil and R. Mitchell, Santa Barbara, USA, August 1998. IEEE Computer Society Press
 - Towards an Emergence Machine for Complex Systems Simulations. P. Marcenac, R. Courdier, S. Calderoni and J.-C. Soulié. In *Proceedings of the 11th International Conference*

on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 785-794, Benicàssim, Spain, June 1998. Lecture Note in Artificial Intelligence, Vol. 1416, Springer Conférences Nationales avec Actes et Comité de Lecture

B.5 Communications avec actes dans un congrès National – ACTN

- Une étude de cas dans le domaine des pêcheries pour la simulation de systèmes complexes. D. Versmisse, J.-C. Soulié, et G. Quesnel. Dans *les actes de 6^{ème} Conférence Francophone de Modélisation et Simulation*. Rabat, Maroc, Avril 2006.
- Jusqu’où peut-on aller dans la caractérisation de la dynamique à court terme du système d’exploitation par analyse des calendriers d’activité ? S. Mahévas, D. Pelletier, O. Thébaud, J.C. Soulié, O. Guyader, P. Berthou, A. Biseau. Dans *les Actes 6^{ème} Forum Halieumétrique*. Montpellier, 24 - 26 juin 2003
- Etude par Simulation Multi-Agents du Comportement de l’Espadon (*Xiphias Gladius*) dans un Paysage Océanique Tropical. D. Guyomard, J.-C. Soulié et P. Marcenac. Dans *les actes du 32^{ème} Colloque de la Société Française pour l’Etude du Comportement Animal*, Lyon, Mai 2000
- Mise en Place d’un Système Multi-Agents Destiné à la Simulation de Dynamiques Comportementales Spatiales Environnement/Ressource, Appliqué à l’Espadon (*Xiphias Gladius*) dans le Sud-ouest de l’Océan Indien. D. Guyomard, J.-C. Soulié, P. Marcenac et M. Larue. Dans *les actes du 4^{ème} Forum de l’Association Française d’Halieumétrie, les Espaces de l’Halieutique*, Rennes, Juin 1999. Editions IRD
- Conception d’un Modèle Multi-Agents pour la Gestion des Effluents d’Elevage à l’Echelle d’une Localité Rurale. F. Guerrin, R. Courdier, S. Calderoni, J.-M. Paillat, J.-C. Soulié et J.-D. Vally. Dans *les actes des 6^{ème} Journées Francophones de l’Intelligence Artificielle Distribuée et des Systèmes Multi-Agents*, pp. 25-37, Pont-à-Mousson, France, Novembre 1998. Hermes
- BIOMAS : un Modèle Multi-Agents pour aider à la Gestion Négociée d’Effluents d’Elevage. F. Guerrin, R. Courdier, S. Calderoni, J.-M. Paillat, J.-C. Soulié et J.-D. Vally. Dans *les actes du 1^{er} Colloque sur les Modèles et Systèmes Multi-Agents pour la Gestion de l’Environnement et des Territoires*, Nils Ferrand Eds., pp 359-378, CEMAGREF Editions, Clermont-Ferrand, France, Octobre 1998. Actes des Colloques du CEMAGREF

B.6 Autres productions – AP

- Rapport d’Activité - Contrat d’Incitation Universitaire N° 953 R321. Jean-Christophe Soulié. 80 pages. 2005
- Rapport de Stage Postdoctoral - Modélisation économique des dynamiques de court terme des activités halieutiques : approche par simulation multi-agents. Jean-Christophe Soulié. Direction des Ressources Vivantes, Service d’Economie Maritime. 150 pages. 2004
- Rapport d’Activité - Convention N°03/2 210 426. Jean-Christophe Soulié. 35 pages. 2004
- Conséquences économiques de l’évolution de l’environnement dans l’estuaire de la Seine sur la pêche professionnelle. O. Thébaud, J. Boncoeur, O. Curtil, O. Guyader, M. Jezequel, A. Martin, J.-C. Soulié, A. Tétard. Rapport Final du Projet. Programme Seine Aval II. 2003

- Des satellites, des poissons et des palangres : MUFINS, un simulateur multi-agents modélisant les dynamiques de comportement et de capturabilité de l’espadon (*Xiphias Gladius*) ciblé par la pêche palangrière dans le sud-ouest de l’océan Indien. D. Guyomard, J.-C. Soulié, I. Bandama et D. K/Bidi. Table ronde sur la simulation. Association Française d’Halieumétrie. Nantes, France, Octobre 2002
- Analyse économique des impacts de pollutions sur le secteur de la pêche professionnelle. O. Thébaud, J. Boncoeur, O. Guyader, A. Martin, J.-C. Soulié et A. Tétard. Programme Seine Aval. Séminaire de Travail du 2 et 3 Octobre 2002.

Bibliographie

- [ANY, 2008] (2008). *How to Build a Combined Agent Based / System Dynamics Model in Any-Logic*. System Dynamics Conference.
- [Arzel, 1998] ARZEL, P. (1998). *Les laminaires sur les côtes bretonnes, évolution de l'exploitation et de la flottille de pêche, état actuel et perspectives*. Edition de l'Ifremer.
- [Barros, 1995] BARROS, F. J. (1995). Dynamic structure discrete event system specification : A new modelling and simulation formalism for dynamic structure systems. *Dans Proceedings of the 1995 Winter Simulation Conference*, pages 781–785.
- [Barros, 1996] BARROS, F. J. (1996). Dynamics structure discrete event system specification : Formalism, abstract simulators and applications. *Transaction of the Society of Computer Simulation*, 13(1):35–46.
- [Barros, 1997] BARROS, F. J. (1997). Modeling formalisms for dynamic structure systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 7:501–515.
- [Benhamou et Bovet, 1991] BENHAMOU, S. et BOVET, P. (1991). Modeling and simulation of animals' movements. *Dans MEYER, J. et WILSON, S., éditeurs : Proceedings of the first international conference on simulation of adaptive behavior, From animals to animats*, pages 135–139. MIT Press-Bradford.
- [Bousquet *et al.*, 1998] BOUSQUET, F., BAKAM, I., PROTON, H. et LE PAGE, C. (1998). Cormas : Common pool resources and multiagent system. *Dans POBIL, A. D. et ALI, M., éditeurs : Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 1416 de *Lecture Notes in Artificial Intelligence*, pages 826–838, Benicassim, Espana. Kluwer.
- [Bousquet et Le Page, 2004] BOUSQUET, F. et LE PAGE, C. (2004). Multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176:313–332.
- [Brisson *et al.*, 1998] BRISSON, N., MARY, B., RIPOCHE, D., JEUFFROY, M., RUGET, F., GATE, P., DEVIENNE-BARRET, F., ANTONIOLETTI, R., DURR, C., NICOUILLAUD, B., RICHARD, G., BEAUDOIN, N., RECOUS, S., TAYOT, X., PLENET, D., CELLIER, P., MACHET, J., MEYNARD, J. et DELÉCOLLE, R. (1998). Stics : a generic model for the simulation of crops and their water and nitrogen balance. I. theory and parametrization applied to wheat and corn. *Agronomie*, 18:311–346.
- [Brisson *et al.*, 2002] BRISSON, N., RUGET, F., GATE, P., LORGEOU, J., NICOUILLAUD, B., TAYOT, X., PLENET, D., JEUFFROY, M., BOUTHIER, A., RIPOCHE, D., MARY, B. et JUSTES, E. (2002). Stics : a generic model for the simulation of crops and their water and nitrogen balance. II. assessment by comparing with experimental reality for wheat and corn. *Agronomie*, 22:69–93.

- [Carey, 1990] CAREY, F. (1990). Further acoustic telemetry observations of swordfish. *Dans* STROUD, R., éditeur : *Proceedings of the 2nd International Billfish Symposium, Planning the Future of Billfishes, Research and Management in the 90s and Beyond*, pages 103–122, Kalia-Kona, Hawaii, USA.
- [Carey et Robinson, 1981] CAREY, F. et ROBINSON, B. (1981). Daily patterns in the activities of swordfish *xiphias gladius* observed by acoustic telemetry. *Fishery Bulletin*, 79(2):277–292.
- [Chelle *et al.*, 1998] CHELLE, M., ANDRIEU, B. et BOUATOUCH, K. (1998). Nested radiosity for plant canopies. *The Visual Computer*, 14:109–125.
- [Chelle *et al.*, 2004] CHELLE, M., HANAN, J. et AUTRET, H. (2004). Lighting virtual crops : the caribu solution for open l-systems. *Dans* GODIN, C., HANAN, J., WINFRIED, K., LACOINTE, A., TAKENAKA, A., PRZEMYSŁAW, P., DEJONG THEODORE, M. et BEVERIDGE, C., éditeurs : *Proceedings of the 4th International Workshop on Functional-Structural Plant Models*, Montpellier, France.
- [Chow et Zeigler, 1994] CHOW, A. et ZEIGLER, B. (1994). Parallel devs : a parallel, hierarchical, modular, modeling formalism. *Dans Proceedings of the 26th conference on Winter simulation*, pages 716–722, Orlando, Florida, United States.
- [CORMAS, 2003] CORMAS (2003). <http://cormas.cirad.fr>.
- [Da Silva et De Lucena, 2004] DA SILVA, V. T. et DE LUCENA, C. J. P. (2004). From a conceptual framework for agents and objects to a multi-agent system modeling language. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):145–189.
- [De Angelis et Yeh, 1984] DE ANGELIS, D. et YEH, G. (1984). *An introduction to modeling migratory behavior of Fishes*, pages 445–469. *Mechanisms of Migration in Fishes*. Plenum Press, New-York and London.
- [Desruisseaux *et al.*, 2001] DESRUISSEAUX, M., PETIT, M. et GARDEL, L. (2001). Seasview : un logiciel d'extraction des données environnementales. Rapport final du projet palangre réunion (PPR), IRD-Unité ESPACE/SEASnet, Montpellier, France.
- [Dingkuhn *et al.*, 2006] DINGKUHN, M., LUQUET, D., KIM, TAMBOUR, L. et CLÉMENT-VIDAL, A. (2006). Ecomeristem, a model of morphogenesis and competition among sinks in rice : 2. simulating genotype responses to phosphorus deficiency. *Functional plant biology*, 33(4):325–337.
- [Dingkuhn *et al.*, 2005] DINGKUHN, M., LUQUET, D., QUILOT, B. et REFFYE, P. D. (2005). Environmental and genetic control of morphogenesis in crops : Towards models simulating phenotypic plasticity. *Australian Journal of Agricultural Research*, 56:1–14.
- [Drogoul *et al.*, 2008] DROGOUL, A., TREUIL, J.-P. et ZUCKER, J.-D. (2008). *Modélisation et simulation à base d'agents : Exemples commentés, outils informatiques et questions théoriques - Cours et exemples*. Dunod.
- [Duboz, 2004] DUBOZ, R. (2004). *Intégration de modèles hétérogènes pour la modélisation et la simulation de systèmes complexes : application au transfert d'échelles en écologie marine*. Thèse de doctorat, Université du Littoral Côte d'Opale, Calais, France.
- [Duboz *et al.*, 2002] DUBOZ, R., RAMAT, E. et GIAMBIASI, N. (2002). Utilisation du formalisme DEVS pour la spécification de systèmes d'agents réactifs. *Dans Actes des Journées Francophones en Intelligence Artificielle et Systèmes Multi Agents (JFIADSMA 2002)*, pages 99–102, Lille, France. Hermès.

- [Duboz *et al.*, 2004] DUBOZ, R., RAMAT, E. et QUESNEL, G. (2004). Systèmes multiagents et théorie de la modélisation et de la simulation : une analogie opérationnelle. Dans GUESSOUM, Z. et BOISSIER, O., éditeurs : *Actes des douzièmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2004)*, Paris, France. Hermes.
- [Essington *et al.*, 2006] ESSINGTON, T., BEAUDREAU, A. et WIEDENMANN, J. (2006). Fishing through marine food webs. *Proceedings of the National Academy of Science*, 103(9):3171–3175.
- [Ferber, 1995] FERBER, J. (1995). *Les systèmes multi-agents : vers une intelligence collective*. InterÉditions, informatique, intelligence artificielle édition.
- [Ferber et Gutknecht, 1998] FERBER, J. et GUTKNECHT, O. (1998). A meta-model for the analysis and design of organisations in multi-agent systems. Dans *Proceedings of the 3rd International Conference on Multi-Agent Systems*, pages 128–135, Paris, France. ICS Press.
- [Ferber *et al.*, 2003] FERBER, J., GUTKNECHT, O. et MICHEL, F. (2003). Agent/group/roles : Simulating with organizations. Dans MULLER, J.-P., éditeur : *Proceedings of the 4th Conference on Agent Based Simulation*, Montpellier, France. European Society for Computer Simulation.
- [Fletcher, 2000] FLETCHER, C. (2000). *Computational Techniques for Fluid Dynamics*, volume 1 and 2. Springer, 2 édition.
- [Gear, 1971] GEAR, C. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*, volume 2, chapitre 2. Prentice-Hall, Englewood Cliffs, NJ, USA.
- [Gordon, 1954] GORDON, H. (1954). The economic theory of common-property resource : The fishery. *Journal of Political Economy*, 62:124–143.
- [Gutknecht *et al.*, 2001] GUTKNECHT, O., FERBER, J. et MICHEL, F. (2001). Integrating tools and infrastructures for generic multi-agent systems. Dans MÜLLER, J. P., ANDRE, E., SEN, S. et FRASSON, C., éditeurs : *Processings of the Fifth International Conference on Autonomous Agent*, pages 441–448, Montreal, Canada.
- [Guyomard, 2005] GUYOMARD, D. (2005). *Approche empirique de la pêche de l'espadon (*Xiphias gladius*) dans l'océan Indien et modélisation des déplacements des poissons dans le paysage océanique tropical*. Thèse de doctorat, École Nationale Supérieure Agronomique de Rennes, Rennes, France.
- [Huhngs et Singh, 1998] HUHNGS, M. N. et SINGH, M. P. (1998). Agents and multiagent systems : themes, approaches and challenges. *Reading in agents*, pages 1–23.
- [Humston *et al.*, 2000] HUMSTON, R., AULT, J., LUTCAVAGE, M. et OLSON, D. (2000). Schooling and migration of large pelagic fishes relative to environmental cues. *Fisheries Oceanography*, 9(2):136–146.
- [Jeuffroy et Recous, 1999] JEUFFROY, M. et RECOUS, S. (1999). Azodyn : a simple model simulating the date of nitrogen deficiency for decision support in whet fertilisation. *European Journal of Agronomy*, 10:129–144.
- [Jones, 1997] JONES, C. (1997). *Geographical Information System and Computer Cartography*. Addison Wesley, 1 édition.
- [Kofman, 2002] KOFMAN, E. (2002). A second order approximation for devs simulation of continuous systems. *Transactions of the Society Computer Simulation International*, 78(2):76–89.
- [Kofman et Junco, 2001] KOFMAN, E. et JUNCO, S. (2001). Quantized state systems. a devs approach for continuous systems simulation. *Transactions of the Society Computer Simulation International*, 18(3):123–132.

- [Lakshmikantham et Trigiante, 1988] LAKSHMIKANTHAM, V. et TRIGIANTE, D. (1988). *Theory of difference equations*. Academic press.
- [Lotka, 1925] LOTKA, A. (1925). Elements of physical biology. *unknown*.
- [Luquet *et al.*, 2006] LUQUET, D., DINGKUHN, M., KIM, H. K., TAMBOUR, L. et CLÉMENT-VIDAL, A. (2006). Ecomeristem, a model of morphogenesis and competition among sinks in rice. 1. concept, validation and sensitivity analysis. *Functional Plant Biology*, 33:309–323.
- [Luquet *et al.*, 2007] LUQUET, D., SONG, Y. H., ELBELT, S., THIS, D., CLÉMENT-VIDAL, A., PÉRIN, C., FABRE, D., et DINGKUHN, M. (2007). Model-assisted physiological analysis of phyllo, a rice architectural mutant. *Functional Plant Biology*, 34:11–23.
- [Marcenac et Giroux, 1998] MARCENAC, P. et GIROUX, S. (1998). Geamas : A generic architecture for agent-oriented simulations of complex processes. *Applied Intelligence*.
- [MEXICO, 2009] MEXICO (2009). Méthodes pour l’exploration informatique des modèles complexes. <http://w3.avignon.inra.fr/mexico/index.php/Accueil>.
- [Minar *et al.*, 1996] MINAR, N., BURKHART, R., LANGTON, C. et ASEKANAZI, M. (1996). The swam simulation system : a toolkit for building multi-agent simulations. Rapport technique, Santa Fe Institute.
- [Muzy *et al.*, 2002] MUZY, A., WAINER, G., INNOCENTI, E., AIELLO, A. et SANTUCCI, J. (2002). Comparing simulation methods for fire spreading across a fuel bed. *AIS*.
- [Odell *et al.*, 2003] ODELL, J. J., PARUNAK, H. V. D. et FLEISCHER, M. (2003). *The Role of Roles in Designing Effective Agent Organizations*, volume 2603/2003 de *Lecture Notes in Computer Science*, chapitre Software Engineering for Large-Scale Multi-Agent Systems. Springer Berlin / Heidelberg.
- [Pauly *et al.*, 1998] PAULY, D., CHRISTENSEN, V., DALSGAARD, J., FROESE, R. et TORRES, F. (1998). Fishing down marine food webs. *Science*, 279:860–863.
- [Pinnegar *et al.*, 2006] PINNEGAR, J., HUTTON, T. et PLACENTI, V. (2006). What relative sea-food prices can tell us about the status of stocks. *Fish and Fisheries*, 7(219–226).
- [Pinnegar *et al.*, 2002] PINNEGAR, J., JENNINGS, S., O’BRIEN, C. et POLUNIN, N. (2002). Long-term changes in the trophic level of the celtic sea fish community and fish market price distribution. *Journal of Applied Ecology*, 39:377–390.
- [Pradal *et al.*, 2008] PRADAL, C., DUFOUR-KOWALSKI, S., BOUDON, F., FOURNIER, C. et GODIN, C. (2008). Openalea : a visual programming and component-based software platform for plant modelling. *Functional Plant Biology*, 35:751–760.
- [Quesnel, 2006] QUESNEL, G. (2006). *Approche formelle et opérationnelle de la multimodélisation et de la simulation des systèmes complexes - Apports pour la simulation de Systèmes Multi-Agents*. Thèse de doctorat, Université du Littoral Côte d’Opale, Calais, France.
- [Quesnel *et al.*, 2008] QUESNEL, G., DUBOZ, R. et RAMAT, E. (2008). The virtual laboratory environment-an operational framework for multi-modelling, simulation and analysis of complex dynamical systems. *Simulation Modelling Practice and Theory*, (in press).
- [Ramat et Preux, 2003] RAMAT, E. et PREUX, P. (2003). Virtual laboratory environment (vle) : A software environment oriented agent and object for modeling and simulation of complex systems. *Journal of Simulation Practice and Theory*, 11:45–55.
- [Reymond *et al.*, 2004] REYMOND, M., MULLER, B. et TARDIEU, F. (2004). Dealing with the genotypexenvironment interaction via a modelling approach : a comparison of qtls of maize leaf length or width with qtls of model parameters. *Journal of Experimental Botany*, 55:2461–2472.

- [Richard *et al.*, 2013] RICHARD, H., MONOD, H., WANG, J., COUTEAU, J., DUMOULIN, N., POUSSIN, B., SOULIÉ, J.-C. et RAMAT, E. (2013). *Développement d'un environnement générique pour piloter l'exploration numérique de modèles : la boîte à outils MEXICO*, chapitre 9. Quae.
- [Sanchirico et Wilen, 1999] SANCHIRICO, J. et WILEN, J. (1999). Bioeconomics of spatial exploitation in a patchy environment. *Environmental Economics and Management*, 37(2):129–150.
- [Schneider, 1998] SCHNEIDER, S. (1998). Couplage sig/geamas. Mémoire de D.E.A., Université de la Réunion, Saint Denis, Île de la Réunion, France.
- [Shaefer, 1957] SHAEFER, M. (1957). Some considerations of population dynamics and economics relation to the management of marine fisheries. *Journal of the Fisheries Resources Board of Canada*, 14:669–681.
- [Soulié, 2001] SOULIÉ, J.-C. (2001). *Vers une approche multi-environnements pour les agents*. Thèse de doctorat, Université de la Réunion, Saint Denis, Île de la Réunion, France.
- [Soulié *et al.*, 2001] SOULIÉ, J.-C., GUYOMARD, D. et DESRUISSEAU, M. (2001). Mufins : a multiagent toolkit for simulating pelagic fish behavior in distributed multiple environments. *Dans Proceedings of the 13rd European Simulation Symposium*, pages 885–890, Marseille, France. SCS Europe Bvba Publisher.
- [Soulié et Marcenac, 2000] SOULIÉ, J.-C. et MARCENAC, P. (2000). A framework to model multiple environments in multiagent systems. *Dans Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, volume 1886 de *Lecture Notes in Artificial Intelligence*, pages 383–393, Melbourne, Australia. Springer Verlag.
- [Soulié *et al.*, 1998] SOULIÉ, J.-C., MARCENAC, P., CALDERONI, S. et COURDIÉ, R. (1998). Geamas v2.0 : an object oriented platform for complex systems simulations. *Dans SINGH, M., MEYER, B. et MITCHELL, R., éditeurs : Proceedings of the 26th International Conference on Technology of Object-Oriented Languages and Systems*, pages 230–242, Santa Barbara, USA. IEEE Computer Society Press.
- [Soulié *et al.*, 2010] SOULIÉ, J.-C., PRADAL, C., FOURNIER, C. et LUQUET, D. (2010). Feedbacks between plant microclimate and morphogenesis in fluctuating environment : analysis for rice using ecomeristem model coupled with 3d plant and energy balance computation tools in openalea platform. *Dans DEJONG, T. M. et SILVA, D. D., éditeurs : Proceedings of the 6th International Workshop on Functional-Structural Plant Models, FSPM 2010*, pages 138–140, University of California, Davis, United States.
- [Steinmetz *et al.*, 2006] STEINMETZ, F., THÉBAUD, O., GUYADER, O. et BLANCHARD, F. (2006). A preliminary analysis of long-term changes in the value of landings by french fishing fleets operating in the north-east atlantic. *Dans Proceedings of the 13rd biennial conference of the IIFET*, Portsmouth, UK.
- [Sumaila, 1998] SUMAILA, U. (1998). Markets and the fishing down marine food webs phenomenon. *European Community Fisheries Cooperation Bulletin*, 11(3–4):25–26.
- [The Center for Study of Complex System, 2009] THE CENTER FOR STUDY OF COMPLEX SYSTEM (2009). What is the study of complex systems? <http://www.cscs.umich.edu/old/complexity.html>.
- [Uhrmacher et Kullick, 2000] UHRMACHER, A. et KULLICK, B. (2000). Plug and test - software agents in virtual environments. *Dans JOINES, J., BARTON, R., KANG, K. et FISHWICK, P., éditeurs : Proceedings of the 2000 Winter Simulation Conference*, pages 1722–1729.

- [Urbani, 2006] URBANI, D. (2006). *Élaboration d'une approche hybride SMA-SIG pour la définition d'un système d'aide à la décision. Application à la gestion de l'eau*. Thèse de doctorat, Université de Corse, Corte, Corse, France.
- [Versmisse, 2008] VERSMISSE, D. (2008). *Gestion de la complexité formelle et opérationnelle des systèmes complexes - Application aux anthroposystèmes marins*. Thèse de doctorat, Université du Littoral Côte d'Opale, Calais, France.
- [Versmisse et al., 2007] VERSMISSE, D., MACHER, C., RAMAT, E., SOULIÉ, J.-C. et THEBAUD, O. (2007). Developing a bioeconomic simulation tool of fisheries dynamics : a case study. *Dans Proceedings of the International Congress on Modelling and Simulation*, Christchurch, New Zealand. International Society for Computer Simulation.
- [Volterra, 1926] VOLTERRA, V. (1926). Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz. dei Lincei. Ser. VI, 2*.
- [Volterra, 1931] VOLTERRA, V. (1931). Variations and fluctuations of the number of individuals in animal species living together. *Animal Ecology*.
- [Von Bertalanffy, 1938] VON BERTALANFFY, L. (1938). A quantitative theory of organic growth. *Human Biology*, 10(181–213).
- [Vos et al., 2010] VOS, J., EVERS, J. B., BUCK-SORLIN, G. H., ANDRIEU, B., CHELLE, M. et de VISSER, P. H. B. (2010). Functional structural plant modelling : a new versatile tool in crop science. *Journal of Experimental Botany*, 61:2101–2115.
- [Wainer et Giambiasi, 2001] WAINER, G. et GIAMBIASI, N. (2001). Application of the cell-devs paradigm for cell spaces modelling and simulation. *Simulation*, 76(22–39).
- [Weiss, 1999] WEISS, G. (1999). *Multiagent Systems. A modern approach to distributed artificial intelligence*. MIT Press.
- [Wooldridge et al., 2000] WOOLDRIDGE, M., JENNINGS, N. R. et KINNY, D. (2000). The GAIA methodology for agent-oriented analysis and design. *Autonomous Agent Multi-Agent Systems*, 3(3):285–312.
- [Yin et al., 1999] YIN, X., KROPFF, M. et STAM, P. (1999). The role of ecophysiological models in qtl analysis : the example of specific leaf area in barley. *Heredity*, 82:412–421.
- [Zadeh et al., 1996] ZADEH, L. A., KLIR, G. J. et YUAN, B. (1996). *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*. World Scientific.
- [Zeigler, 1976] ZEIGLER, B. (1976). *Theory Of Modeling and Simulation*. Wiley Interscience.
- [Zeigler, 1998] ZEIGLER, B. (1998). Devs theory of quantization. *Dans DARPA Contract N6133997K-0007 : ECE Dept.*, Tucson, AZ, USA.
- [Zeigler et al., 2001] ZEIGLER, B., CHO, H., KIM, J., SARJOUGHIAN, H. et LEE, J. (2001). Quantization-based filtering in distributed simulation : Experiments and analysis. *Journal of Parallel and Distributed Computing*.
- [Zeigler et Lee, 1998] ZEIGLER, B. et LEE, J. (1998). Theory of quantized systems : Formal basis for DEVS/HLA distributed simulation environment. *Dans Enabling Technology for Simulation Science (II)*, SPIE AeoroSense, Orlando, FL, USA.
- [Zeigler et al., 2000] ZEIGLER, B. P., KIM, D. et PRAEHOFER, H. (2000). *Theory of Modeling and Simulation : Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press.