

# UNE ETUDE DE CAS DANS LE DOMAINE DES PECHERIES POUR LA SIMULATION DE SYSTEMES COMPLEXES

D. VERSMISSE, J.C. SOULIE,  
G. QUESNEL

Laboratoire d'Informatique du Littoral  
Université du Littoral Côte d'Opale  
62100 Calais, France

[versmisse@lil.univ-littoral.fr](mailto:versmisse@lil.univ-littoral.fr), [soulie@lil.univ-littoral.fr](mailto:soulie@lil.univ-littoral.fr), [quesnel@lil.univ-littoral.fr](mailto:quesnel@lil.univ-littoral.fr)

**RESUME :** Cet article est l'occasion de montrer l'avancée du laboratoire dans le domaine de la simulation de systèmes complexes à travers une étude de cas : la simulation d'une pêcherie. On va notamment y présenter VLE la plate-forme développée au laboratoire mais surtout les choix techniques sur lesquels elle repose.

**MOTS-CLES :** écosystèmes, DEVS, systèmes complexes, plate-forme de simulation, SMA.

## 1 INTRODUCTION

La simulation de systèmes complexes, mais également la modélisation à travers des outils simples, intuitifs et rapides sont des enjeux actuels majeurs. Ils permettront la création et surtout l'utilisation de modèles complexes donnant ainsi des outils de décisions modernes et efficaces.

Nous allons présenter à travers une étude de cas les approches choisies par le laboratoire pour répondre à ces préoccupations. Ce projet a débuté en 2001 en partenariat avec l'IFREMER, sa problématique vise à étudier la faisabilité d'une simulation de la dynamique à court terme de pêcherie avec notamment une prise en compte des fermetures de zones pour cause de pollution ou de régulation : les AMP (Aires Marines Protégées).

Il est à noter que la plupart des logiciels développés dans ce domaine reposent sur des solutions maison pour gérer les problèmes posés par les systèmes complexes.

La plate-forme choisie à l'époque fut CORMAS (Bouquet et al., 1998) et elle a permis de valider une série de choix en particulier l'intégration des SMA comme support de calcul pour les pêcheries. Une description du projet nommé **Dynfish** est présentée en partie 3.

Parallèlement le laboratoire a développé sa propre plate-forme : VLE pour *Virtual Laboratory Environment*. (Ramat et Preux, 2003). Cette plate-forme se veut, entre autres, généraliste, elle repose sur le formalisme DEVS et intègre un *plugin* « agent ». Il est donc possible via VLE, après « transcription » en DEVS, de faire interagir des variables gérées par des équations différentielles et des agents. Afin de valider une partie de la plate-forme et de faire évoluer le projet, il a été décidé en mai 2005 de basculer **Dynfish** sur VLE. Ce fut également l'occasion de modifier le modèle en accentuant encore davantage le

côté agent. Nous allons présenter cette approche dans la partie 4.

## 2 ETAT DE L'ART

Nous allons présenter dans cette partie l'état de l'art des différents formalismes et technologies utilisées par nos modèles.

### 2.1 Systèmes multi-agents (SMA)

Bien qu'il y ait un grand nombre de chercheurs et de scientifiques qui travaillent dans le domaine des systèmes multi-agents, il n'y a malheureusement pas de définition commune pour un agent et par conséquent pour un système multi-agents.

En effet, en fonction du domaine de recherche considéré, on peut trouver des définitions pour les agents logiciels (Hermans, 1997), les agents réactifs (Ferber, 1994), les agents cognitifs (Wooldridge, 1995), ... C'est pourquoi, dans cette section, nous proposons d'utiliser les définitions de J. Ferber (Ferber, 1995). Elles peuvent être résumées comme présenté ci-dessous :

Un agent est une entité physique ou virtuelle qui :

- Est capable d'agir sur son environnement,
- Est capable de communiquer directement avec d'autres agents,
- Possède des ressources,
- Est capable de percevoir son environnement local,
- Possède une représentation de son environnement,
- Peut éventuellement se reproduire.

Un système multi-agents contient :

- Un environnement  $E$ ,
- Un ensemble  $O$  d'objets,

- Un ensemble  $A$  d'agents qui sont des objets particuliers ( $A \subseteq O$ ),
- Un ensemble de relations  $R$  qui relient les objets (et les agents),
- Un ensemble de processus  $P$  qui permet aux agents de percevoir, créer, détruire, altérer et manipuler des objets de  $O$ .

Enfin, on peut isoler deux types de systèmes multi-agents bien distincts :

- *Les systèmes multi-agents purement communiquant* : quand  $A = O$  et  $E = \emptyset$ , alors les relations  $R$  définissent un réseau de communication. Les agents sont en relation directe avec les autres agents,
- *Les systèmes multi-agents situés* : quand les agents sont situés dans l'environnement. Dans ce cas,  $E$  est généralement défini avec une métrique (e.g. une distance euclidienne).

Mais dans la plupart des cas, on utilise plutôt des systèmes multi-agents qui appartiennent en partie à chacun de ces types.

L'utilisation des systèmes multi-agents pour modéliser et simuler des écosystèmes et des écosystèmes exploités est maintenant largement répandue. De plus, ils ont fait la preuve de leur robustesse et de leur facilité d'expression dans le monde des systèmes complexes. Les lecteurs intéressés par une revue complète de l'utilisation des systèmes multi-agents pour la simulation de la gestion des écosystèmes pourront lire (Bousquet et Le Page 2004).

## 2.2 DEVS

Afin de garantir une homogénéité au sein des échanges entre les différents composants d'un système complexe, il est nécessaire de disposer d'un langage commun. Le formalisme DEVS permet dans une certaine mesure de répondre à cette préoccupation. DEVS définit le « dialogue » entre les différents composants par l'échange d'événements discrets estampillés. La spécification d'un modèle atomique DEVS se fait via la définition d'une structure  $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, Ta \rangle$ , où :

$X$  est l'ensemble des types d'événements externes et  $Y$  l'ensemble des types d'événements de sortie ;

$S$  est l'ensemble des états séquentiels ;

$\delta_{int}$  (resp.  $\delta_{ext}$ ) est la fonction de transition interne (resp. externe) définissant les changements d'état dus à des événements internes (resp. externes) ;

$\lambda$  est la fonction de sortie ;

$Ta$  est la fonction définissant la durée de vie des états.

Typiquement, les événements émis en sortie d'un modèle perturbent les modèles qui sont connectés à cette sortie. La perturbation est spécifiée par la fonction de transition externe. La fonction de transition interne est,

quant à elle, évaluée si la durée de vie d'un état notée  $Ta$  est atteinte. La fonction de sortie est activée lors des transitions internes.

Le formalisme DEVS prévoit également la possibilité de réaliser des modèles couplés. Un modèle couplé définit comment est couplé un ensemble de modèles entre eux pour former un nouveau modèle. Il peut lui-même faire parti d'un modèle couplé. On définit alors une construction hiérarchique des modèles.

Un modèle couplé comprend les informations suivantes :

- l'ensemble des modèles qui le compose,
- l'ensemble des ports d'entrée qui recevront les événements externes,
- l'ensemble des ports de sortie qui émettront les événements,
- les couplages en ports d'entrée et ports de sortie des modèles composant le modèle couplé.

On trouvera dans la littérature en plus d'une description plus détaillée de DEVS les simulateurs abstraits correspondant (Zeigler, 1976).

## 2.3 Cell-DEVS

L'extension Cell-DEVS est née de la constatation suivante : de nombreux modèles font intervenir des espaces discrets et utilisent des formalismes tels que les automates cellulaires. Wainer et Giambiasi dans (Wainer et Giambiasi, 2001) développent l'extension Cell-DEVS. Cette extension doit pouvoir décrire et simuler des modèles à base d'automates cellulaires multidimensionnels et à événements discrets. La dynamique des cellules est temporisée c'est à dire que l'état d'une cellule sera modifié en fonction de l'état de son voisinage mais il sera connu des cellules voisines qu'après un certain délai. L'idée de base est de fournir un mécanisme simple de définition de la synchronisation des cellules. Comme toutes propositions d'extensions, les auteurs offrent à la fois l'extension du formalisme qui se résume à l'ajout de variables supplémentaires et de leurs sémantiques et le simulateur abstrait.

Un modèle Cell-DEVS est basé sur la même structure qu'un modèle DEVS classique. Nous allons retrouver les modèles atomiques pour les cellules, les éléments de base d'un réseau, et les modèles couplés pour les réseaux eux-mêmes. Néanmoins, la structure proposée par Zeigler (Zeigler et al., 2000) se voit augmentée d'attributs. Le modèle DEVS d'une cellule est défini par la structure suivante :

$$TDC = \langle X, Y, I, S, N, \text{délai}, d, \delta_{int}, \delta_{ext}, \tau, \lambda, Ta \rangle$$

où :

$X$  et  $Y$  est l'ensemble des ports et des valeurs d'entrée et de sortie,

$I$  l'interface de la cellule

$S$  l'ensemble des états de la cellule,

$N$  l'ensemble des états des cellules voisines,

*délai* le type de délai utilisé

$d$  la durée du délai

$\delta_{int}$  et  $\delta_{ext}$  la fonction de transition interne et externe,

$\tau$  la fonction locale de sortie,

$\lambda$  la fonction de sortie,

$Ta$  la fonction d'avancement du temps.

L'interface  $I$  de la cellule définit le voisinage de la cellule ainsi que les connexions en terme de ports d'entrée et de sortie entre la cellule et ses voisins. Il y a autant de ports d'entrée que de voisins. La fonction de calcul  $\tau$  modélise la fonction de calcul de l'état de la cellule en fonction de l'état du voisinage. L'état de la cellule est effectif pour les cellules voisines qu'au bout du délai d'attente  $d$ .

La définition des cellules est complétée par la définition d'un modèle couplé. Le modèle d'une cellule définit l'interface qu'elle offre à l'extérieur mais ne précise pas la forme des connexions. Le rôle du modèle couplé est donc de définir les connexions entre les cellules (à l'aide d'un *pattern* -  $N$ ,  $C$  et  $B$ ), la taille  $\{t_1, \dots, t_n\}$  et la dimension  $n$  du réseau et l'interface  $I$  avec d'éventuels modèles DEVS (de type Cell-DEVS ou non).

$$GCC = \langle X_{list}, Y_{list}, I, X, Y, n, \{t_1, \dots, t_n\}, N, C, B, Z \rangle$$

$X_{list}$  et  $Y_{list}$  définissent la liste des cellules du réseau possédant des ports d'entrée et des ports de sortie non connectés en interne et par conséquent disponibles pour une connexion avec un autre modèle. En terminologie DEVS, cet ensemble est l'ensemble des connexions entre les ports d'entrée et des ports de sortie avec les ports du modèle couplé. L'interface  $I$  du réseau complète la définition en réunissant au sein d'une même structure les éléments de définition de l'interface du réseau vers l'extérieur. Cette définition intègre l'ensemble  $Z$  qui met en relation les ports de sortie appartenant à  $Y_{list}$  d'un réseau et les ports d'entrée appartenant à  $X_{list}$  d'un autre réseau.  $X$  et  $Y$  représentent l'ensemble des événements d'entrée et de sortie. De manière à simplifier la définition des connexions entre les cellules du réseau (communément notés connexions internes dans les modèles couplés) un *pattern* de voisinage, noté  $N$ , est défini. Ce *pattern* spécifie pour chaque cellule n'appartenant pas à la bordure  $B$  la position relative de ses voisins.

Avec Cell-DEVS, nous disposons d'un outil de spécification DEVS pour les automates cellulaires. Il faut noter que Cell-DEVS ne s'arrête pas à un langage de spécification mais offre aussi le simulateur abstrait afin de préciser le comportement d'un modèle.

## 2.4 QSS

Parallèlement aux méthodes classiques sont apparues de nouvelles techniques de résolution numérique d'équations différentielles basées sur la quantification des valeurs de sortie plutôt que sur la discrétisation du temps. Ainsi par exemple, Kofman propose à travers QSS1 et QSS2 deux méthodes de résolution numérique d'équations différentielles du premier ordre. Pour la

présentation de ces deux méthodes, considérons le système suivant :

$$\dot{x}(t) = f(x(t), u(t))$$

$$y(t) = g(x(t), u(t))$$

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 (ou d'une fonction extérieure) dans le formalisme DEVS. La fonction  $y$  est, quant à elle, la sortie désirée. Voyons maintenant ces deux techniques de résolution plus en détails.

### 2.4.1 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

La quantification avec hystérésis consiste à remplacer la fonction  $x$  par une fonction constante par morceaux  $\hat{x}$ . Cette fonction va prendre ses valeurs dans  $\dots, d_{i-1}, d_i, d_{i+1}, \dots$  en accord avec une discrétisation de  $\mathcal{R}$ . La fonction  $\hat{x}$  prend la valeur  $d_i$  quand  $x$  augmente et  $d_i \leq x < d_{i+1}$  ou quand  $x$  décroît et que  $d_{i-\varepsilon} \leq x < d_{i+1} - \varepsilon$ .  $\varepsilon$  est la largeur de la fenêtre d'hystérésis. Finalement le calcul du pas de temps est réalisé en calculant quand  $x(t)$  (approximé par  $x(t_0) + \dot{x}(t_0)(t - t_0)$ ) va atteindre  $d_{i+1}$  ou  $d_i - \varepsilon$ . Voyons maintenant le modèle DEVS de QSS1 :

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, Ta \rangle$$

$$X = Y = \mathcal{R}$$

$$S = \mathcal{R} \times \mathcal{R} \times Z \times \mathcal{R}^+ \cup \infty$$

$$\delta_{int}(x, u, i, \sigma) = (x + \sigma \cdot u, u, i + \text{sgn}(u), \sigma')$$

$$\delta_{ext}(x, u, i, \sigma, e, v) = (x + e \cdot u, v, i, \sigma'')$$

$$\lambda(x, u, i, \sigma) = d_{i + \text{sgn}(u)}$$

$$Ta(x, u, i, \sigma) = \sigma$$

où :

$\sigma'$  prend la valeur :

$$\frac{d_{i+2} - (x + \sigma \cdot u)}{u} \quad \text{si } u > 0$$

$$\frac{x + \sigma \cdot u - d_{i-1} + \varepsilon}{|u|} \quad \text{si } u < 0$$

$\sigma''$  prend la valeur :

$$\frac{d_{i+1} - (x + e \cdot u)}{v} \quad \text{si } v > 0$$

$$\frac{x + e \cdot u - d_i + \varepsilon}{|v|} \quad \text{si } v < 0$$

$$\infty \quad \text{si } v = 0$$

avec :

- $i$  (via  $d_i$ ) l'approximation courante de  $x$  pour le reste du système,
- $x$  et  $u$  l'approximation de  $x(t)$  et  $\dot{x}(t)$ .  $u$  est donné par  $f$ .

Si  $x(t)$  est multi-variables, il suffit d'appliquer la même méthode sur toutes ses parties  $x_i(t)$ .

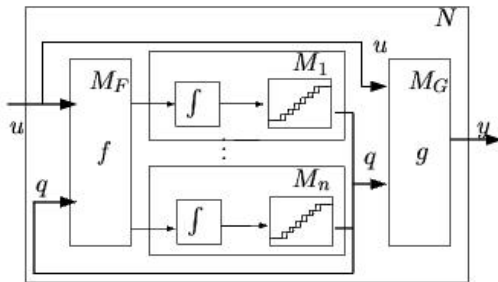


Fig 1 : Modèle DEVS d'un intégrateur QSS. Chaque boîte est un modèle DEVS atomique. (tiré de Kofman et Junco, 2001)

### 2.4.2 QSS2

QSS2 repose sur les mêmes idées que QSS1 mais à la place d'approximer  $x(t)$  par une fonction constante par morceau, on approxime  $x(t)$  par une fonction affine par morceaux. Pour réaliser ce calcul, nous devons connaître  $\ddot{x}(t)$ , Kofman propose par conséquent un modèle DEVS de  $f$  permettant de réaliser une approximation de  $\ddot{x}(t)$ .

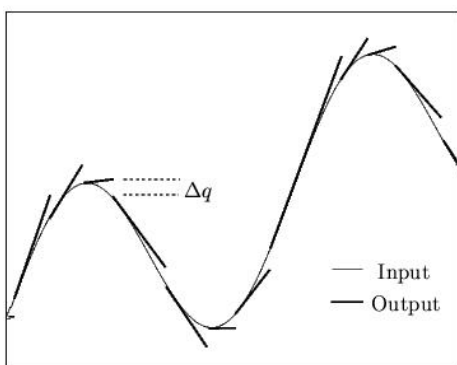


Fig 2 : Approximation par une fonction affine par morceaux de la fonction à intégrer (tiré de Kofman 2002)

## 3 PROJET CORMAS

Ce premier projet fut développé en collaboration avec l'IFREMER. L'accent a été mis sur les technologies actuellement disponibles pour la programmation et la maintenance d'un système complexe type pêche.

Cette première phase a permis de mettre en avant l'intérêt d'une modélisation du problème à l'aide du formalisme SMA (Système Multi-Agents.) La modélisation de comportements complexes, ainsi que l'interaction entre les différentes entités s'en trouvent fortement simplifiées. Nous avons également fait un choix pour une plate-forme qui s'est porté sur CORMAS (CORMAS 2005). Beaucoup d'arguments justifient ce choix : CORMAS est une plate-forme éprouvée, utilisée par une grande communauté et enfin elle est dédiée à la modélisation.

### 3.1 Modélisation sous CORMAS

Afin de restituer le plus fidèlement possible le découpage des entités proposé par le logiciel CORMAS, notre approche de modélisation peut être décomposée en trois sous parties distinctes : modélisation spatiale, sociale et passive.

Une présentation complète et détaillée du modèle et de ses résultats peut-être lue dans (Soulié et Thébaud, 2005).

#### 3.1.1 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 *SpatialEntity-Cell* de CORMAS. De part cet héritage, elles sont donc considérées comme des cellules d'un automate cellulaire. Cette caractéristique peut nous permettre de simuler les différents flux entre les cellules.

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 soit découpé en quatre 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 compose). 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.

#### 3.1.2 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 au déplacement éventuel d'un port à un autre d'un ensemble de bateau. 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.

### 3.1.3 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'il cible, ses captures, ses gains et ses coûts. De plus, 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.

### 3.2 Application à la simulation d'un arrêt temporaire de pêche

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.

| Paramètres             | Espèce 1 | Espèce 2 |
|------------------------|----------|----------|
| Paramètres biologiques |          |          |

|  |        |        |
|--|--------|--------|
| Capacité de charge   | 10     | 10     |
| Taux intrinsèque de croissance   | 0,8    | 0,8    |
| Coefficients de diffusion de bio-masse   | 0.2    | 0.2    |
| <b>Paramètres économiques</b>  |        |        |
| Prix des débarquements   | 10     | 10     |
| Coût unitaire de l'effort de pêche   | 3      | 3      |
| Polyvalence  | 0,0004 | 0,0004 |
| Mobilité   | 0,0004 | 0,0004 |
| Taux de pénalité lié au changement de zone : 0,1 pour un changement horizontal/vertical et 0,2 pour un changement en diagonale |        |        |

Tab 1 : Valeur des paramètres utilisés dans la simulation

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 pêches concernées par la fermeture de zone ne sont pas autorisées à se reporter spatialement ; 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.

#### 3.2.1 Résultat des simulations

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

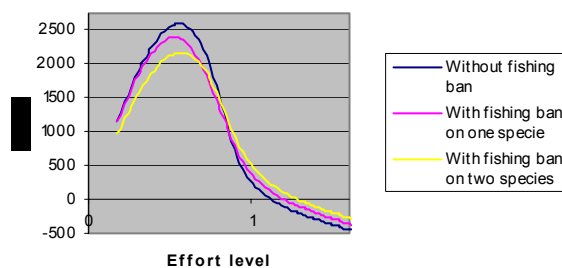


Fig 3 : Evolution de la rente pour différents niveaux d'efforts donnés

La fermeture de zone implique une diminution de la rente globale de la pêche. Cette diminution est plus importante si on ferme complètement une zone que si on la ferme partiellement. Dans les deux cas, 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 pêche 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 exporte 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*

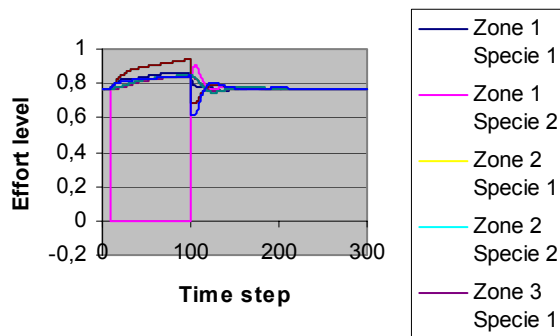


Fig 4 : Dynamique de la réponse à une fermeture

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.

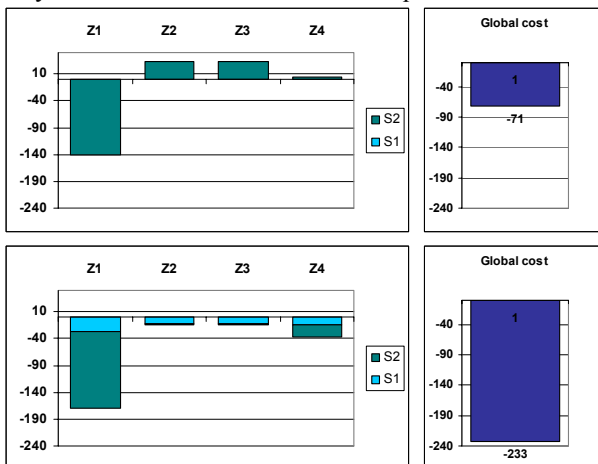


Fig 5 : Coûts par zone et globaux de la fermeture de zone

La première figure ci-dessus 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. 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 reçues 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 figure ci-dessus 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*

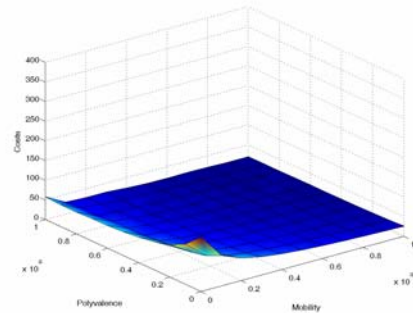


Fig 6 : Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort bas

Cette figure montre le cas d'une pêche 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 pêche un fort potentiel 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 l'on change seulement de métier, on augmente la pression de pêche sur une espèce et donc le niveau de capture baisse par conséquent, la rente aussi.

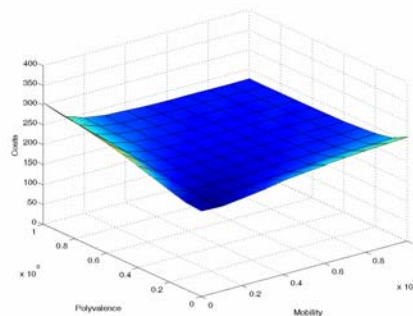


Fig 7 : Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort à l'optimum

Cette figure 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 par conséquent par une baisse de la rente.

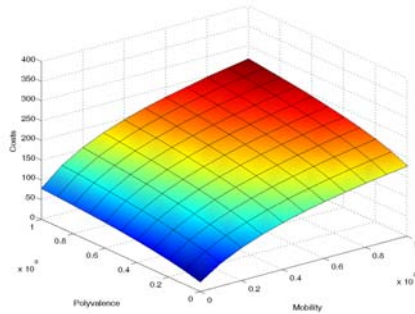


Fig 8 : Aire couverte par les coûts quand la mobilité et la polyvalence augment pour un niveau d'effort haut

Cette figure 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 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.

## 4 PROJET VLE

Ce second projet explore une autre approche pour répondre à la même problématique de simulation de systèmes complexes engendrés par une simulation de pêcherie. Le contre-pied principal par rapport à la précédente étude est l'utilisation de la nouvelle plate-forme VLE développée par l'équipe du Laboratoire d'Informatique du Littoral. Nous allons commencer par lister les contraintes liées au modèle qui sont légèrement différentes du premier modèle sous CORMAS puis nous verrons les réponses que nous avons apportées lors de la modélisation. Les objectifs généraux sont les suivants : simuler une pêcherie avec un seul port et plusieurs bateaux. Les bateaux vont évoluer sur une surface découpée en cellules où vivent une seule espèce de poissons. Une microéconomie est mise en place en considérant le prix actuel du poisson ainsi que la consommation des bateaux en carburant. Il sera ainsi possible de dégager le bénéfice global de la pêcherie.

### 4.1 Description du modèle bio-économique

Ce modèle repose sur trois entités : les bateaux, les poissons et un port. Voyons leurs définitions et contraintes plus en détails.

#### 4.1.1 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 leur capacité à capturer des poissons.

Après être allé au port chercher un scénario de pêche (une liste de taille variable de cellules en mer associées à un temps de pêche, cf. 3.3 pour une description plus précise des attributions de scénarii), le bateau part en mer effectuer sa campagne. Il va alors noter sa consommation en carburant, ainsi que la quantité de poisson capturé 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.

#### 4.1.2 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. Ceci pour plusieurs raisons : les équations gérant la biomasse de populations de poissons sont classiques et éprouvées 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. Voir pour cela les références (Gordon, 1954), (Shaefer, 1957) et (Sanchirico et Wilen, 1999).

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

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

où :

- $x_i$  est la biomasse de poisson dans la case  $i$  ;
- $f_i(x_i)$  traduit l'évolution propre de la biomasse dans la cellule ;
- $d_{ii}x_i + \sum_{j=1, j \neq i}^n d_{ij}x_j$  traduit l'interaction entre les cellules : dispersion de l'espèce, source, puits, ...

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

$$f_i(x_i) = r_i \left(1 - \frac{x_i}{x_i^{\max}}\right) x_i \text{ avec } r_i \text{ le taux de croissance naturelle et } x_i^{\max} \text{ la capacité de charge de la cellule.}$$

Cette équation traduit une croissance logistique de la population de poissons au sein d'une cellule.

$$d_{ii}x_i + \sum_{j=1, i \neq j}^n d_{ij}x_j = \sum_j \left( \frac{x_j}{x_j^{\max}} - \frac{x_i}{x_i^{\max}} \right) \text{ avec } j \text{ dans le}$$

voisinage de  $i$  (les 4 cellules autour de la case  $i$ ) qui dépeint une dispersion naturelle des poissons en fonction de la concentration dans la cellule. Lorsqu'un bateau prélève des poissons dans une cellule, la biomasse se voit soustraite de la quantité suivante :

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

- $x$  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.

#### 4.1.3 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 scénarii. 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.

#### 4.2 Implémentation dans VLE

L'implémentation a été effectuée sur la plate-forme VLE, *Virtual Laboratory Environment*, développée au Laboratoire d'Informatique du Littoral. L'orientation de cette plate-forme est modulaire, le but étant de développer ou de réutiliser de petits modules puis de les assembler en décrivant leurs interactions. La dynamique choisie est une gestion événementielle du temps qui permet aussi bien de faire de l'événementiel que du « temps discret classique », les événements étant alors générés de façon régulière. Ces deux raisons ont conduit l'équipe à adopter le formalisme DEVS développé par Zeigler pour l'ensemble du simulateur (Zeigler, 1976). VLE est donc un moteur DEVS et par conséquent les modules sont,

dans la terminologie DEVS, des modèles atomiques ou des modèles couplés. La définition des interactions entre ces modèles se fait par une description des connexions reliant ces modèles. Une fois ce choix fait, il convient maintenant de détailler les différents modèles afin de voir comment développer les contraintes imposées en DEVS.

Une approche orientée agents a été choisie sur l'ensemble du projet. Le port et les bateaux sont considérés comme des agents et l'ensemble des cellules contenant le poisson représente l'environnement physique où évoluent les agents. Un des problèmes actuels de l'équipe est d'automatiser la gestion des agents, faisant alors de VLE l'une des premières plate-formes intégrant simultanément des outils de bas niveau (modèles DEVS) et de haut niveau (formalisme agent).

#### 4.2.1 Intégration des agents dans un contexte DEVS

Les premiers travaux d'une formalisation en DEVS des systèmes multi-agents sont réalisés par A.M. Uhrmacher au milieu des années 90 (Uhrmacher et Schattenberg, 1998.) Ces travaux mettent en relation l'agent et le modèle DEVS par une description des fonctions de transitions internes et externes formalisant, respectivement, le comportement autonome de l'agent et la perception de l'agent. Les fonctions de sorties sont alors vues comme des actions des agents sur l'environnement. Depuis, d'autres travaux (Duboz et al., 2004) ont formalisé les agents d'une manière plus complète basés sur l'extension DS-DEVS pour Dynamics Structure DEVS (Barros, 1996.) Cette extension a pour but la manipulation des connexions d'un modèle couplé et la création de modèles pendant l'exécution d'une simulation. DS-DEVS définit l'ensemble des connexions d'un modèle couplé comme étant un état du système. Les connexions peuvent alors changer sous l'influence de fonctions qui viennent modifier cet état. Ces changements de structures permettent de mieux refléter les dynamiques réelles des SMA où les relations entre agents sont susceptibles de changer dynamiquement. La plate-forme VLE intègre une définition complète des systèmes multi-agents principalement réactifs et sociaux en se basant sur le concept des environnements multiples (Soulié, 2001) et de DS-DEVS.

#### 4.2.2 Application au 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. Le point noir de cette approche 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 formalismes, en effet les poissons sont gérés par des équations différentielles et les pê-



cheurs/port sont orientés agents afin de tirer le meilleur parti des deux formalismes.

La définition a proprement parlé de l'agent port reste simple : Le port est un agent réactif qui communique avec les pêcheurs.

#### 4.2.3 Application aux 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 de simples modèles analytiques.

#### 4.2.4 Application aux poissons

Différents travaux sur l'intégration d'équations différentielles en DEVS ont déjà été effectués. On peut citer par exemple les travaux de Kofman qui propose QSS1 et QSS2 (Kofman et Junco, 2001 et 2002), deux méthodes d'intégration de systèmes différentiels d'ordre 1. Nous avons opté pour l'implémentation de QSS1 en créant un modèle atomique DEVS hérité d'un modèle de case Cell-DEVS. Cell-DEVS est une extension de DEVS prenant en compte les problèmes où la division en cases est naturelle (par exemple le jeu de la vie de John Conway (Gardner, 1970), simulation de feu de forêt, ...).

Le modèle CellQSS1 étant préexistant au sein de VLE, il a suffi d'hériter de cette classe afin d'implémenter une case de poisson. Les seules fonctions à surcharger étant celle évaluant  $\dot{x}$  et celle permettant la configuration des constantes en fonction des paramètres rentrés dans le fichier de configuration. Vient ensuite le problème des connexions. En effet ces cases sont toutes interconnectées et il convient de créer ce réseau. Le *plugin* agent en cours de réalisation devra dans l'avenir remplir ce rôle automatiquement, mais pour l'instant il est possible au sein de VLE de définir dans le fichier de configuration des parties non compatibles VLE et traduites à la volé par des *translators* en code VLE. Il a suffi d'écrire un *translator* spécifique aux cases de poissons afin de créer le réseau de connexions automatiquement. Il aurait été possible d'utiliser le codage VLE directement ou encore le modeleur graphique GVLE, mais les connexions sont trop nombreuses pour être écrites à la main (sauf sur les bords, la sortie d'une cellule est connectée à ses 4 cellules voisines).

## 5 SIMULATION

La plate-forme VLE est développée conjointement sous Unix et Windows, c'est un projet écrit en C++ s'appuyant sur un ensemble de bibliothèques classiques (glib, gtkmm, libxml++...) afin de faciliter le portage sur toutes les plates-formes. La version Linux est aujourd'hui la plus aboutie. On lance le simulateur avec comme paramètre un fichier XML décrivant l'expérience à réaliser. Parallèlement au simulateur, l'équipe développe un outil graphique GVLE permettant l'écriture automatique du fichier d'expérience.

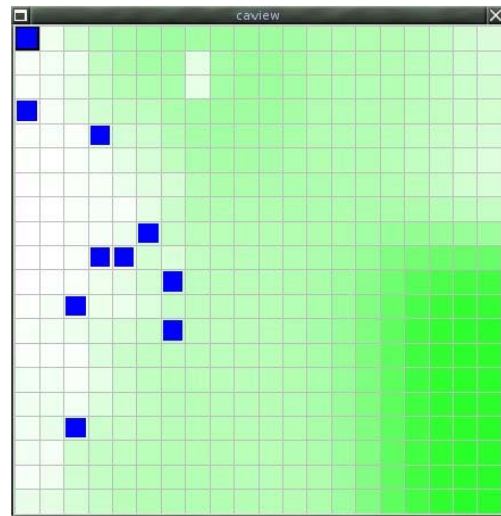


Fig 6 : Exécution de la simulation sous VLE

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.

## 6 CONCLUSION

Nous avons présenté dans cet article deux approches différentes pour répondre aux préoccupations de la modélisation et la simulation de systèmes complexes. Ces deux approches traduisent l'avancé chronologique du laboratoire dans ce domaine.

La première version développée sous CORMAS, logiciel développé par le CIRAD bénéficie d'un grand nombre d'utilisateurs (modélisateurs et thématiciens) formant une communauté avec ses avantages évidents. Le développement de modèles est très facile et très rapide. Un autre point, que partage d'ailleurs CORMAS et VLE, est qu'il est facile de modifier un élément quelconque de l'expérience en cours, puisque la description de celle-ci est faite au sein d'un fichier XML.

La seconde version présentée sous VLE nous paraît également intéressante pour plusieurs raisons : l'utilisation de DEVS comme formalisme pour gérer la modularité, la réutilisation des modèles. Une montée en charge du système est même possible dans le sens où un modèle peut-

être remplacé par un autre s'il possède au moins les mêmes ports. L'utilisation de DEVS nous donne également un cadre formel sûr pour assurer le déroulement logique de la simulation (par exemple la chronologie des événements est respectée au sein du simulateur). Enfin la popularité de DEVS nous permet de profiter de tous les travaux dans ce domaine. Un point qui nous paraît maintenant important est de trouver et d'implémenter une logique pour simplifier encore l'écriture des modèles atomiques, nous pensons par exemple à une description via les *Statecharts* (Harel, 1987) ou encore un langage à base de règles.

Dans sa version actuelle, le modèle DynFish sous VLE permet déjà d'appréhender un large nombre de scénarii (mobilité, espèce avec des caractéristiques distinctes selon les zones, etc.). Néanmoins, nous envisageons de lui rajouter un certain nombre de fonctionnalités :

Complexifier le modèle spatial, en intégrant des cartes avec des zones de pêche, des zones de terre, etc.

Dans le même ordre d'esprit, il serait intéressant de pouvoir définir plusieurs espèces afin de pouvoir simuler des réallocations de l'effort de pêche sous forme spatiale mais également avec changement d'espèces cibles, avec, pourquoi pas, une relation proie-prédateur entre les espèces de poissons.

## REFERENCES

- Barros F., 1996, Dynamics Structure Discret Event System Specification: Formalism, Abstract Simulators and Applications, *Transactions of the Society for Computer Simulation*, 1(13), p. 35-46.
- Bousquet F., I. Bakam, H. Proton et C. Le Page, 1998, CORMAS: Common-pool ressources and multi-agents system. In A.P del Pobil and M. Ali, editors, Proceedings of IEA-AIE'98, volume 1416 of LNAI, pages 826-838. Springer Verlag, June 1998.
- Bousquet, F. et C. Le Page, 2004. Multi-agent simulations and ecosystem management: a review. *Ecological Modelling*, 176 (3-4): 313-332
- CORMAS 2005; Site web. <http://cormas.cirad.fr>.
- Duboz R. et E. Ramat et G. Quesnel, 2004, Systèmes Multi-Agents et théorie de la modélisation et de la simulation : une analogie opérationnelle, *Actes des douzièmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA) – Système multi-agents défis scientifiques et nouveaux usages*, Olivier Boissier et Zahia Guessoum, Paris.
- Ferber J., 1994 Reactive distributed artificial intelligence: Principles and applications. In *G.M.P O'Hare and N.R. Jennings, editors, Sixth Generation Computer Technology*, pages 287-314, New York, USA. Waley-Interscience Publication
- Ferber J., 1995. *Les Systèmes Multi-Agents – Vers une intelligence collective*. iia – InterEditions.
- Gardner M., 1970, Mathematical Games. The fantastic combinations of John Conway's new solitaire game « Life ». *Scientific American*, 223, p. 120-123.
- Gordon H.S., 1954, The economic theory of common-property resource: The fishery. In *J. Political Econom*, 62, p. 124-142.
- Harel D., 1987, Statecharts: A Visual Formalism For Complex Systems, *Science of Computer Programming*, 8, p.231-274.
- Hermans B., 1997. Intelligent Software Agents on The Internet: An Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments. *First Monday – Peer-Reviewed Journal On The Internet*.
- Kofman E. and S. Junco, 2001, Quantized State Systems. A devs approach for continuous systems simulation. *In Transactions of SCS*, 18, p. 123-132.
- Kofman E., 2002, A second order approximation for devs simulation of continuous systems. *In transaction of SCS*, 78, p. 76-89
- Ramat E. and P. Preux, 2003, Virtual Laboratory Environment VLE: a software environment oriented agent and object for modelling and simulation of complex systems. *In Simulation Modelling Practice and Theory*, 11, p. 45-55.
- Shaeffer M.B., 1957, Some considerations of population dynamics and economics relation to the management of marine fisheries. *In J. Fisheries Res. Board Canada*, 14, p. 669-681.
- Sanchirico J.N. and J.E. Wilen, 1999, Bioeconomics of spatial exploitation in a patchy environment. *In Journal of Environmental Economics and Management*, 37, p. 129-150.
- Soulié J.C., 2001, Vers une approche multi-environnement pour les agents. *Université de la Réunion, Thèse de doctorat*.
- Soulié J.C. and O ; Thébaud, 2005, Modelling Fleet Response in Regulated Fisheries: An Agent-Based Approach, *Journal of Mathematical and Computer Modelling*, In Press.
- Uhrmacher A. M. And B. Schattenberg, 1998, Agents in Discrete Event Simulation, *10TH European Simulation Symposium „Simulation in Industry – Simulation Technology: Science and Art” (ESS'98)*, Nottingham, UK, p. 129-136.
- Wainer G.A. and N. Giambiasi, 2001, Application of the cell-devs paradigm for cell spaces modeling and simulation. *In simulation*, 18, p. 22-39.
- Wooldridge M. and N.R Jennings, 1995 Intelligent Agents: *Theory and Practice*. Knowledge Engineering Review.
- Zeigler B.P., 1976, Theory Of Modelling and Simulation. *Wiley Interscience*.
- Zeigler B.P. and H. Praehofer and T.G. Kim, 2000, Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, *second edition, Academic Press*.