



RAPPORT BIBLIOGRAPHIQUE

DEVS et ses extensions pour des simulations multi-modèles en interaction multi-échelles

Auteur :

Nicolas DUHAIL

Master Recherche Informatique – Université de Rennes 1

Élève Ingénieur – Telecom Bretagne

Encadrants :

Marc PARENTHOEN

Pascal REDOU

Lab-STICC – IHSEV

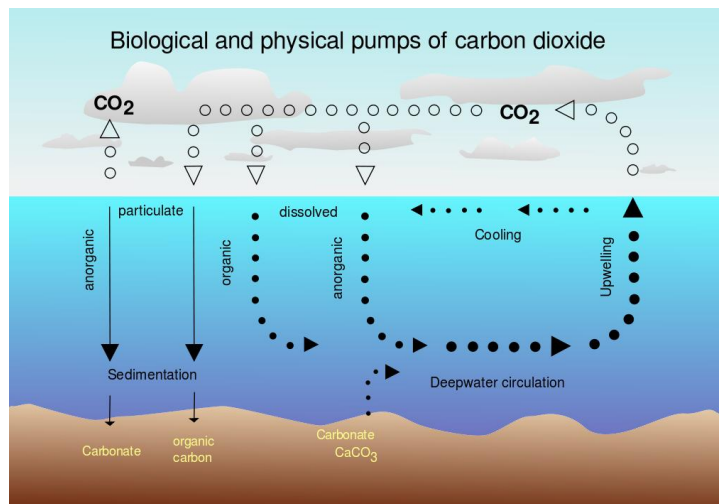
Centre Européen de Réalité Virtuelle

Table des matières

Introduction.....	3
I/ Formalisme DEVS.....	4
1- Présentation du Framework.....	4
2- Atomic DEVS	4
3- Coupled DEVS.....	5
II/ Extensions de DEVS	6
1- Aspect multi-modèles.....	6
a) Modélisation dynamique	6
b) Modélisation basée agent.....	7
c) Modélisation hybride.....	7
2- Aspect multi-échelles	7
a) Multi-Resolution DEVS	8
b) Multi-Level DEVS	8
3- Optimisation.....	9
III/ Outils basés sur DEVS pour la simulation de systèmes complexes	10
1- Expérimentations en réalité virtuelle	10
2- Plateformes de simulation et laboratoires virtuels	11
a) VLE	11
b) NetLogo.....	12
c) AnyLogic.....	13
Conclusion	13
Références.....	14

Introduction

Dans le but d'étudier la pompe biologique (cycle marin du carbone) dans le domaine mésopélagique (zone de l'océan comprise entre 100m et 1000m de profondeur), ce qui sera le sujet de mon stage réalisé au CERV (Centre Européen de Réalité Virtuel), nous serons amenés à développer un laboratoire virtuel pour expérimenter des modèles des phénomènes impliqués dans cette colonne d'eau. Notre objectif est de pouvoir vérifier différents modèles fournis par des spécialistes du cycle marin du carbone. Ce laboratoire virtuel devra permettre de simuler un système multi-modèles interchangeable et modifiables facilement en tenant compte des interactions multi-échelles lors de leur simulation.



En effet, la pompe biologique est un système complexe qui fait intervenir plusieurs phénomènes : à la surface, le phytoplancton fixe le carbone, puis sédimente vers le fond de l'océan, en s'agrégeant et se dissolvant en fonction de la microbiologie associée à ces particules. Elles sont de plus en interaction avec le zooplancton et le système trophique de la zone mésopélagique. Les phénomènes physico-chimiques de l'océan sont également à considérer : turbulence, température, concentrations chimiques. Il y aura donc un aspect continu d'une part pour les phénomènes physiques et un aspect agent d'autre part pour le comportement des individus du système trophique [26].

Cela nous amène à étudier les différentes échelles de ce système. Au niveau spatial, cela va donc de quelques micromètres (taille du phytoplancton) jusqu'à un kilomètre environ (colonne d'eau). Au niveau temporel, il y a un aspect continu (sédimentation, dissolution) et un aspect discret (désagrégation/agrégation du sédiment, broutage par le plancton). Aussi les échelles varient de quelques années pour une expérience virtuelle où l'on veut mesurer ce qui arrive au fond de l'océan en fonction du choix des modèles, à quelques millisecondes pour le broutage.

Depuis son introduction en 1976 [38], le formalisme DEVS a donné lieu à de nombreux travaux pour formaliser de tels systèmes complexes. Nous examinerons parmi ces travaux ceux permettant la modélisation à la fois dynamique, basée agent, multi-échelles mais aussi permettant leur expérimentation. DEVS peut-il rendre compatible les aspects multi-modèles ? Notre problématique est de trouver quel formalisme DEVS permettrait la réalisation d'un laboratoire virtuel pour la pompe biologique dans le domaine mésopélagique.

Le présent document est organisé de la façon suivante : dans un premier temps nous étudierons un formalisme pour la modélisation, la simulation et l'analyse de systèmes complexes, appelé DEVS. La plupart des travaux de modélisation de systèmes complexes sont basés sur ce Framework. Ensuite nous analyserons les extensions de DEVS qui pourraient nous être utiles pour construire un laboratoire virtuel de la pompe biologique. Enfin nous aborderons quelques plateformes basées sur DEVS permettant la simulation de systèmes complexes.

I/ Formalisme DEVS

1- Présentation du Framework

Le formalisme DEVS (de l'anglais Discrete Event system Specification) a été introduit par B.P. Zeigler en 1976 [38]. C'est un formalisme à événements discrets reconnu pour son expressivité, abstraction, universalité et son indépendance de toute implémentation. Il intègre intrinsèquement la dimension temporelle (elle est donc explicite) et permet de manipuler conjointement les deux concepts d'état et d'évènement. Il introduit également la possibilité d'évolution autonome du modèle grâce à la durée de vie des états.

C'est un formalisme modulaire et hiérarchique pour l'analyse, la modélisation et la simulation de systèmes complexes. Depuis sa création, il a été utilisé à de nombreuses reprises en ingénierie (systèmes de communications [21], ...), en biologie [10] et en sociologie [19] par exemple. Sur la base de ce formalisme, de nombreux travaux de recherche ont donné lieu à des extensions, méthodes et approches [4 ; 22 ; 32].

Il existe deux objets élémentaires dans DEVS, un modèle atomique et un modèle couplé, qui échangent des informations sous la forme d'événements. Leurs caractéristiques sont décrites dans les deux parties suivantes.

2- Atomic DEVS

Un modèle atomique est défini comme un ensemble de ports d'entrée et de sortie, et un ensemble de fonctions :

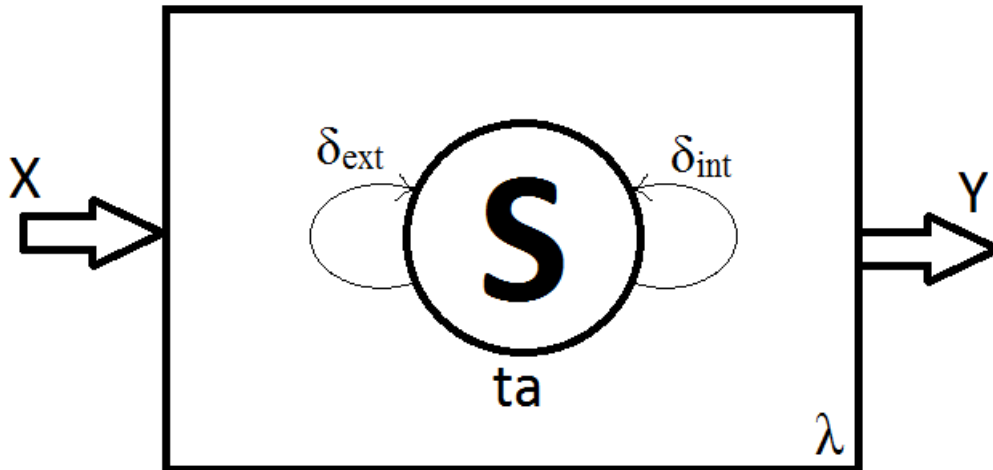
- la transition d'état interne ;
- la transition d'état externe ;
- une fonction qui peut générer des événements destinés à d'autres modèles ;
- une fonction qui gère la durée des différents états.

Ainsi, on peut définir un modèle atomique par un 7-uplet de la forme :

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

où X est l'ensemble des événements entrants, Y celui des événements sortants, S est l'ensemble des états du système, ta est la fonction *time advance* qui gère la durée de vie d'un état, δ_{ext} est la transition d'état externe qui définit comment un événement d'entrée modifie l'état du système, δ_{int} est la transition d'état interne qui définit comment un état du système change sans action externe (quand le temps écoulé atteint la durée de vie de l'état), et λ est la fonction de sortie (elle définit comment un état du système génère un événement).

Voici un schéma représentant un modèle atomique DEVS symétrique :



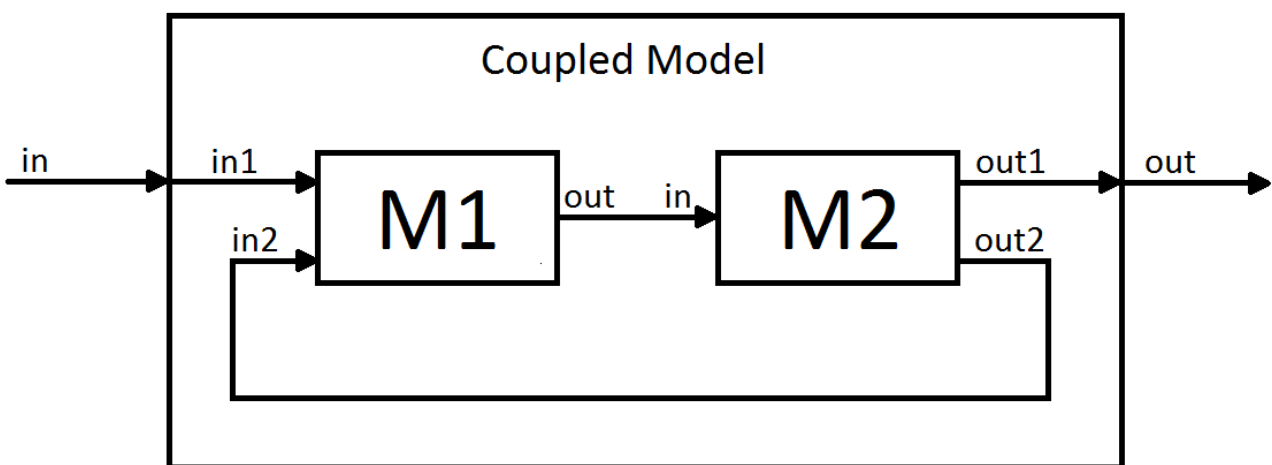
3- Coupled DEVS

Ce modèle décrit un système à événements discrets sous la forme d'un réseau de composants couplés. Les connexions entre différents modèles atomiques peuvent être faites par un modèle couplé. De la même manière que pour le modèle atomique, un modèle couplé est défini par un 8-uplet de la forme :

$$N = \langle X, Y, D, \{M_i\}, C_{xx}, C_{yx}, C_{yy}, Select \rangle$$

où D est l'ensemble des noms des sous-composants, M_i est l'ensemble des sous-composants (pour chaque i , M_i est soit un modèle atomique soit un modèle couplé), C_{xx} est l'ensemble des couplages externes en entrée, C_{yx} est l'ensemble des couplages internes, C_{yy} est la fonction de couplage externe en sortie, et $Select$ est une fonction qui définit quel événement choisir dans le cas d'événements simultanés (voir partie II-3 concernant l'optimisation pour la description d'un formalisme qui s'affranchit de cette fonction, Parallel-DEVS).

De plus, il est possible de rendre un modèle couplé équivalent à un modèle atomique quel que soit le système étudié [33]. Ci-dessous est présenté un schéma d'un modèle couplé DEVS :



Le formalisme DEVS de base est donc très utile pour formaliser des systèmes de toutes sortes grâce à l'apport de cohérence entre les modèles. Mais il possède des limites car il est trop générique et n'est pas suffisant pour la spécification de certains systèmes complexes [22 ; 32]. Ainsi, pour avancer

dans leurs travaux, des scientifiques ont proposé des variations de DEVS. Nous allons en aborder certaines dans la partie suivante.

II/ Extensions de DEVS

Depuis sa création en 1976, de nombreuses extensions du formalisme classique DEVS ont été développées, avec pour chacune des caractéristiques propres et des applications diverses. Nous verrons tout d'abord celles qui traitent de l'aspect multi-modèles, puis celles qui abordent l'aspect multi-échelles, et enfin celles qui considèrent l'optimisation.

1- Aspect multi-modèles

Certains formalismes basés sur DEVS traitent de la dynamique des systèmes, tandis que d'autres se concentrent plus particulièrement sur les agents. Quelques-uns essaient quant à eux de rassembler tout cela sous un seul et même formalisme, pour les systèmes hybrides.

a) Modélisation dynamique

La modélisation dynamique est la modélisation d'un système de manière continue. Elle est basée sur l'écriture d'équations différentielles qui régissent le système. Pour les résoudre à tout instant, il s'agit de discrétiser le système. La simulation de ce genre de modèles continus peut mener à des temps d'exécution très longs. On peut les réduire en se focalisant sur les sous-systèmes ayant les plus grands niveaux d'activité [24]. En utilisant DEVS, il a été montré qu'on peut réduire un programme complexe en une suite de fonctions simples [10].

Voici quelques formalismes basés sur DEVS pour la modélisation dynamique :

- dynDEVS [34] : un modèle dynDEVS est un ensemble de modèles DEVS plus une fonction de transition pour définir l'ordre des modèles [22]. Il est adapté aux modèles à structure dynamique.
- rho-DEVS [31] : cette extension introduit les ports dynamiques et les couplages multiples. Leur réunion autorise les modèles à activer ou désactiver certaines interactions en même temps.
- RT-DEVS (*Real Time DEVS*) [15] : propose une sémantique pour la spécification de systèmes en temps réel d'une façon modulaire.

Afin de modéliser un système complexe tel que l'océan (ou une partie de l'océan), les modèles continus ne vont plus être adaptés. En effet, dans la communauté océanographique, il a été montré que ces modèles ne sont pas suffisants pour représenter les divers flux de particules au fond de l'océan [26]. De plus, les modèles développés jusqu'alors ne sont pas capables de relater la complexité de l'écosystème. Chaque population d'individus est représentée comme un seul groupe et non comme un ensemble de plusieurs individus, donc de nombreuses interactions entre individus d'un même groupe sont négligées [26]. Pour surmonter ces difficultés, plusieurs approches sont actuellement développées, basées sur les agents.

b) Modélisation basée agent

Un système multi-agent (SMA) est un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Largement utilisés en intelligence artificielle, les SMA permettent de modéliser de manière intéressante des sociétés diverses et variées [1 ; 11 ; 28].

Tout d'abord, les agents sont dotés d'un système de décisions et éventuellement de planification à plusieurs. On les dote ensuite d'un modèle cognitif : on peut par exemple citer le modèle BDI [3]. Enfin, les agents doivent posséder un système de communication.

Voici quelques formalismes basés sur DEVS pour la modélisation basée agent :

- *Cell-DEVS* [35] : cette extension permet la modélisation de systèmes physiques complexes en tant qu'espaces cellulaires.
- *Symbolic DEVS* [39] : ce formalisme se concentre sur le temps discret.

c) Modélisation hybride

Afin de prendre en compte à la fois des aspects système dynamique et des aspects agent, certains scientifiques se sont penchés sur les modèles hybrides. Lors de la modélisation de systèmes complexes, la complexité est généralement non seulement due au grand nombre de composants mais aussi à la diversité de ces composants et à leurs interactions.

Voici quelques formalismes basés sur DEVS pour la modélisation hybride :

- *FD-DEVS (Finite & Deterministic DEVS)* [16] : ce formalisme a été conçu pour l'analyse et la vérification de ses réseaux.
- *SP-DEVS (Schedule Preserving DEVS)* [18] : c'est une sous-classe de DEVS et FD-DEVS.
- *G-DEVS (Generalized DEVS)* [8] : spécialement imaginé pour les modèles hybrides.
- *GK-DEVS (Geometric & Kinematic DEVS)* [17] : cette extension permet la modélisation et la simulation de systèmes multi-composants 3D.

Après avoir vu l'utilisation de DEVS à travers l'aspect multi-modèles, intéressons-nous à présent à son utilisation dans la considération de l'aspect multi-échelles.

2- Aspect multi-échelles

En fonction des types de modèles impliqués dans les simulations, nous devons déterminer si ces systèmes sont capables de représenter des interactions simultanément à différentes échelles, ou concevoir des algorithmes efficaces pour implémenter de telles simulations multi-échelles sur des architectures multicores.

La plupart des systèmes réels incluent des interactions parmi une grande variété de phénomènes physiques. De plus, l'échelle de temps et de taille de chaque processus peut varier de manière importante. Ainsi, les simulations numériques de ces problèmes multi-échelles nécessitent des modèles

sophistiqués et des méthodes pour leur intégration, mais aussi des algorithmes efficaces et des techniques avancées de calcul [25].

La conception de simulations multi-agents appliquées aux systèmes complexes pose entre autres le problème de la modélisation de comportements intervenant à des échelles spatiales, temporelles, et comportementales différentes, chacune pertinente pour représenter un des aspects du phénomène étudié. La simulation de systèmes complexes, qu'ils soient composés principalement d'agents réactifs comme en biologie cellulaire, ou plutôt d'agents cognitifs comme dans les sociétés artificielles, se doit de prendre en compte les divers niveaux d'organisation qui rendent compte des phénomènes étudiés. Aussi a-t-on vu apparaître des architectures destinées à représenter des emboîtements de systèmes et sous-systèmes (encapsulation d'agents ou d'environnements). Plus récemment, ce sont des problématiques d'appartenance multiple (à des groupes, à des espaces différents) qui ont pris le pas sur le simple problème d'encapsulation.

Il existe actuellement plusieurs plateformes de gestion d'échelles comme SWARM [23], AGRE [7], MASQ [29], IODA/JEDI [25]... Elles sont basées sur des formalismes comme DEVS, et nous allons voir dans cette partie deux exemples d'extensions de DEVS qui prennent en compte l'aspect multi-échelles.

a) Multi-Resolution DEVS

Le terme anglais *resolution* représente le degré de détail et de précision utilisé dans la représentation d'aspects du monde réel dans un modèle ou une simulation. Le principal but de la modélisation *multi-resolution* est d'extraire les détails de modèles complexes qui sont nécessaires pour restituer une partie du système étudié, et de se débarrasser des détails inutiles.

Le formalisme MR-DEVS, basé sur la structure dynamique de DEVS, a été créé pour les problèmes de modélisation *multi-resolution* [22]. Auparavant, il n'y avait rien qui permettait d'avoir des modèles cohérents entre eux pour ce genre de modélisation. MR-DEVS est un méta-modèle permettant de décrire des entités *multi-resolution*, c'est-à-dire des entités qui peuvent interagir avec des objets à une autre échelle que la leur.

Cette extension de DEVS a été introduite afin de modéliser la biologie au niveau système. Elle peut s'adapter pour contrôler les relations entre les niveaux micro et macro dans un système biologique. De plus, MR-DEVS peut modéliser un processus biologique à plusieurs niveaux hiérarchiquement et en terme de modularité [9].

Par rapport à un modèle classique DEVS (atomique ou couplé), un modèle MR-DEVS est défini par un 9-uplet comprenant des variables qui prennent en compte les changements de précision. Ces dernières peuvent impacter d'autres modèles MR-DEVS grâce à des fonctions de diffusion qui s'assurent de la cohérence du système modélisé.

b) Multi-Level DEVS

La base de la modélisation est l'abstraction. Ainsi, la réunion de différents niveaux d'abstraction est d'un intérêt primordial pour la communauté de modélisation et simulation depuis de

nombreuses années [32]. En biologie, nous allons donc avoir besoin d'un moyen de décrire les niveaux micro et macro et les liens qui existent entre ces niveaux.

Une nouvelle fois, une variation de DEVS va nous être utile, Multi-Level DEVS (ML-DEVS) [32]. Dans cette approche, le niveau macro peut accéder aux informations du niveau micro et vice-versa, les micro-modèles peuvent être activés simultanément par le macro-modèle, et les micro-modèles peuvent modifier la dynamique du niveau macro. Pour lier tout cela, on combine différentes méthodes, parmi lesquelles le couplage de valeurs et les activations synchrones [32].

Dans ML-DEVS, le modèle couplé est équipé d'un état et d'un comportement propre, afin que le niveau macro n'apparaisse pas en tant qu'unité à part. D'autre part, il faut définir comment chaque niveau affecte l'autre. Ces deux tâches sont liées. Pour propager l'information, l'idée de couplage de valeurs semble satisfaisante. Il s'agit de diffuser l'information du niveau macro sur des ports spécifiques, et les micro-modèles peuvent accéder aux macro-variables en définissant les bons noms de ports.

Au niveau micro et macro, les modèles sont toujours déclenchés par le temps et l'arrivée d'événements. Mais ici, on ajoute le fait que le niveau macro peut directement activer ses composants en leur envoyant un événement, et la dynamique du niveau macro peut être modifiée au niveau micro, si le nombre de composants dans un certain état dépasse un seuil fixé.

3- Optimisation

Afin de réduire les temps de calcul lors de simulation de systèmes complexes, des extensions de DEVS permettant la parallélisation des calculs ont été mises en place. Elles s'avèrent particulièrement efficaces pour la résolution d'équations différentielles [20] (modèle dynamique).

En se basant sur l'extension de DEVS nommée Parallel-DEVS (P-DEVS) [4], on va pouvoir utiliser des méthodes de simulations en parallèle. Les extensions traditionnelles utilisent des techniques où des branches de la hiérarchie de modèles sont réparties sur des calculateurs. Un point de synchronisation est alors nécessaire pour effectuer un test sur la causalité des événements. La méthode P-DEVS utilise une méthode simple de parallélisation des calculs en utilisant une parallélisation des modèles lorsqu'au dépilement d'une date dans l'échéancier, toutes les fonctions de transition internes ou externes des modèles sont effectuées en parallèle. La différence entre P-DEVS et DEVS classique est l'utilisation de sac d'événements (*bags*) en entrée sur la fonction de transition externe [4]. De plus, la fonction Select de DEVS classique est remplacée par une fonction de conflit, qui donne la possibilité au modélisateur de choisir, pour un modèle, entre ses événements externes et interne s'ils ont lieu à la même date.

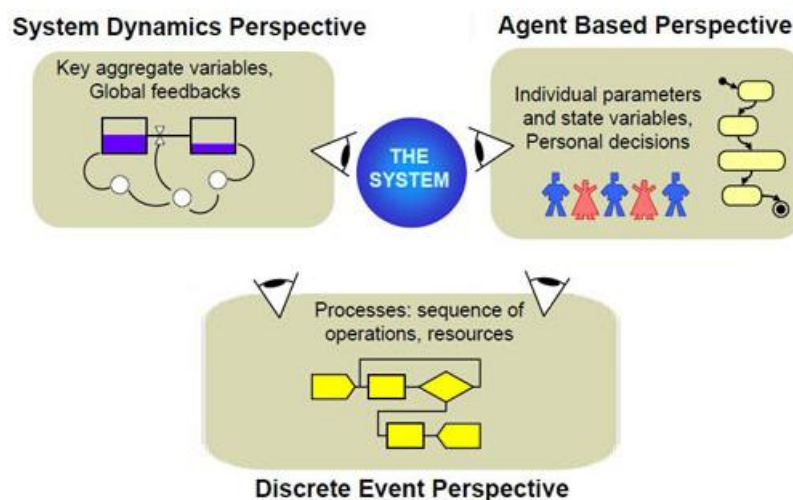
Les simulations parallèles se servent des algorithmes conservatifs pour garantir la cohérence et l'ordre causal des messages échangées entre des processus logiques. Elles utilisent des modèles modifiés de DEVS (comme P-DEVS) pour gérer la synchronisation [12 ; 40]. La simulation parallèle se fait en deux phases. Tout d'abord, tous les composants sont en lecture seule et ils lisent les informations sur les autres composants. À la fin de cette étape on synchronise pour vérifier la cohérence. Ensuite, tous les composants sont mis à jour selon le temps minimum d'un événement, puis resynchronise pour garantir la réussite de la mise à jour et la transmission des messages.

La performance de cet algorithme parallèle est principalement affectée par la répartition de la charge sur les composants, la communication entre composants, et la synchronisation entre processus logiques. Si le nombre de composants est largement supérieur au nombre de processeurs utilisés (ce qui est très souvent le cas en pratique), on peut s'attendre à obtenir des performances satisfaisantes [12].

Nous venons de passer en revue des formalismes divers permettant la spécification de modèles de systèmes complexes dans des cas variés. Il est maintenant nécessaire d'utiliser des outils pour la simulation de ces modèles, afin de pouvoir les valider ou non. C'est ce que nous allons aborder dans la partie suivante.

III/ Outils basés sur DEVS pour la simulation de systèmes complexes

Afin de simuler des systèmes complexes, plusieurs types de méthodes existent, à commencer par la simulation d'une société composée d'agents [1]. On peut également étudier la dynamique du système [24]. Notre objectif est de pouvoir facilement simuler plusieurs modèles d'un même système complexe grâce à des plateformes basées sur DEVS.



1- Expérimentations en réalité virtuelle

La réalité virtuelle est une simulation informatique interactive immersive, visuelle, sonore et/ou haptique, d'environnements réels ou imaginaires. Dans le cadre d'un laboratoire virtuel, elle permet aux scientifiques de réaliser des expériences *in virtuo* de modèles numériques de systèmes complexes [5]. Ce type d'expérimentations (dynamiques et interactives) est en effet nécessaire quand on modélise des systèmes comportant une structure et une dynamique complexes. Elles exploitent l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités qui interagissent en temps réel (entre elles et avec l'utilisateur).

En plus de pouvoir observer l'activité du modèle numérique en cours d'exécution sur un ordinateur, l'utilisateur peut tester la réactivité et l'adaptabilité du modèle en fonctionnement, profitant ainsi du caractère comportemental des modèles numériques. Une expérimentation *in virtuo* est ainsi

une expérimentation conduite dans un univers virtuel de modèles numériques en interaction et auquel l'homme participe. Une telle simulation participative en réalité virtuelle met en œuvre des modèles de types différents (multi-modèles) à des échelles spatio-temporelles différentes (multi-échelles) issus de domaines d'expertise différents. Elle est souvent complexe car son comportement global dépend autant du comportement des modèles eux-mêmes que des interactions entre modèles [30].

Un outil d'expérimentation conçu par le CERV pour étudier de tels modèles est le *virtuoscope* [30]. Ce néologisme désigne un laboratoire virtuel pour l'étude des systèmes complexes en s'appuyant sur les concepts, les modèles et les outils de la réalité virtuelle. Ce paradigme aide à formuler des modèles mixtes dans des échelles de temps et d'espaces différentes, ce qui est un prérequis pour l'étude de systèmes biologiques complexes [22 ; 26 ; 32].

2- Plateformes de simulation et laboratoires virtuels

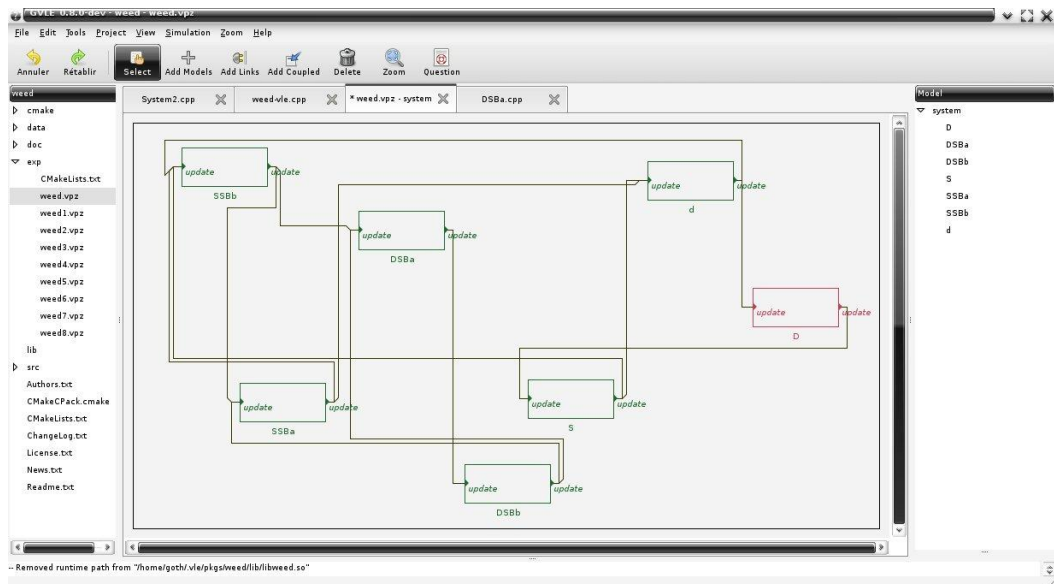
Le but d'un laboratoire virtuel est de pouvoir expérimenter un modèle plutôt qu'un autre. On rencontre le plus souvent dans le domaine de la simulation deux types principaux de laboratoires virtuels :

- Un laboratoire virtuel basé agent est utilisé pour la simulation de modèles de sociétés composés d'agents et utilise pour cela des technologies d'intelligence artificielle [6]. Ainsi, dans ce type d'environnement, chaque élément est modélisé par un agent, et on les place dans un environnement virtuel (sur ordinateur) pour observer l'évolution du système. On peut ensuite transposer les résultats de la simulation obtenus à des agents réels. Un laboratoire virtuel permet d'observer des événements qu'on ne pourrait pas voir en réalité [6]. Des laboratoires virtuels sont également utilisés pour acquérir des connaissances [6 ; 14] (destinés notamment à des élèves).
- Une autre manière de simuler des systèmes complexes est d'utiliser un logiciel de dynamique des systèmes. En effet, la dynamique des systèmes prend en compte les boucles de rétroaction internes et les effets retard qui affectent le comportement global du système. Elle est fondée sur des modèles qui sont une formalisation de nos suppositions à propos d'un système [13]. En dynamique des systèmes, faire tourner une simulation consiste à résoudre les équations mathématiques pour obtenir la valeur de chaque variable au cours du temps.

a) VLE

VLE, pour *Virtual Laboratory Environment*, est une plate-forme de multi-modélisation et de simulation de systèmes dynamiques basée sur le formalisme à événements discrets DEVS [27]. VLE fournit de nombreux programmes comme un simulateur, une interface utilisateur pour modéliser et développer des systèmes, et des outils pour analyser et visualiser les résultats de simulations. Les bibliothèques fournies permettent également le développement de programmes personnalisés.

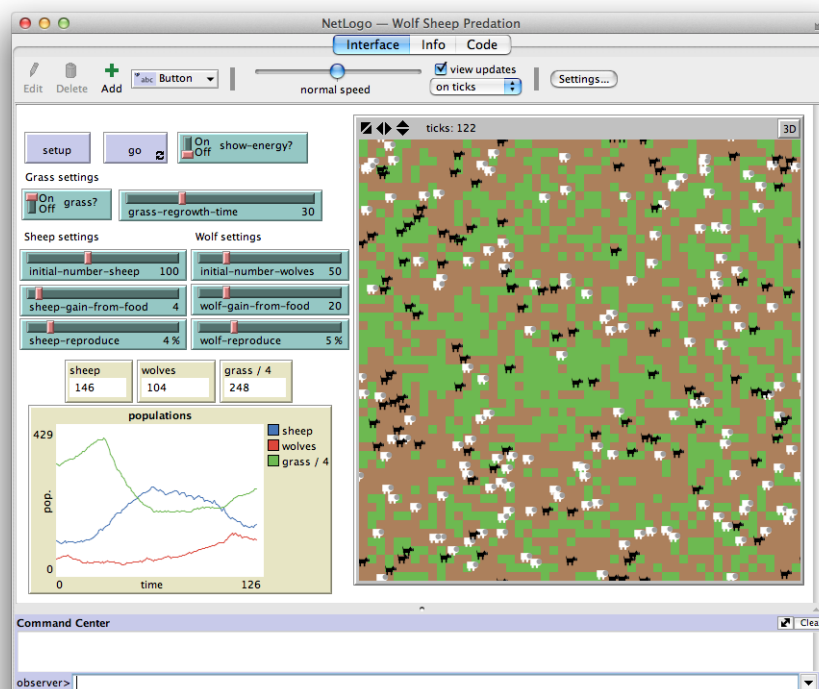
VLE rend possible la spécification de systèmes complexes spatiaux en terme d'objets et d'agents réactifs [27]. On peut citer le projet RECORD qui utilise VLE pour modéliser et simuler des agroécosystèmes [2].



b) NetLogo

NetLogo est un environnement de modélisation et de simulation multi-agents permettant d'observer la relation entre le comportement de chaque individu (niveau micro) et la structure générale qui s'en dégage [36] (niveau macro).

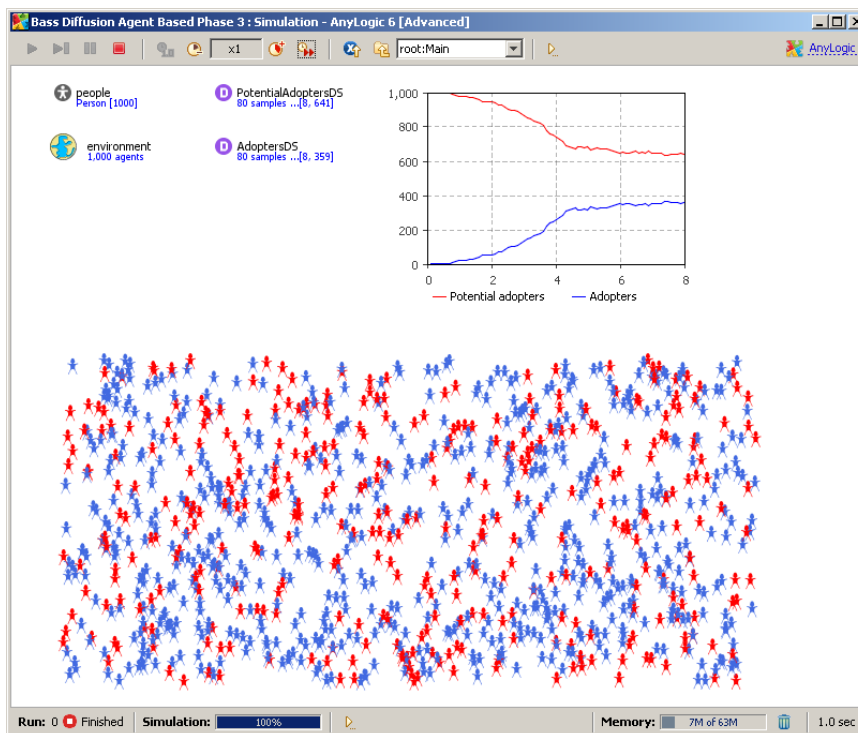
Il est bien adapté à la modélisation de systèmes complexes composés de milliers d'agents agissant en parallèle. Les « modélisateurs » peuvent donner des instructions à un grand nombre d'agents opérant indépendamment les uns des autres, ce qui permet d'explorer les liens entre les comportements des individus à leur niveau et les schémas généraux (comportements de groupe ou de masse) qui émergent des interactions entre de nombreux individus.



c) AnyLogic

AnyLogic permet la modélisation et la simulation de quasiment tous les phénomènes réels grâce à une diversité exceptionnelle [37]. Les modèles AnyLogic peuvent être basés sur n'importe quel paradigme de simulation : événement discret, dynamique de systèmes, et système multi-agents.

Les modélisateurs doivent s'intéresser à des approches combinées en vue d'obtenir une vision plus profonde des processus interdépendants complexes et de nature très différente. AnyLogic permet au modélisateur de combiner ces approches de simulation dans le même modèle. Il n'y a aucune hiérarchie fixe. Cette approche de langage combinée est directement applicable à une large variété de problèmes complexes qui peuvent être modélisés via chaque approche indépendamment ou bien avec leurs combinaisons [37].



Il existe d'ores et déjà un grand nombre de plateformes de modélisation et simulation basées sur DEVS, qui semblent (tout du moins celles étudiées) très complètes. La possibilité de vérifier plusieurs modèles est souvent présente, mais l'aspect multi-échelles est moins fidèle.

Conclusion

Cette étude bibliographique nous a permis de constater que le formalisme DEVS de base se révèle très utile pour formaliser des systèmes de toutes sortes grâce à l'apport de cohérence entre les modèles. Cependant, il reste très générique et n'est donc pas adéquat pour la spécification de certains systèmes complexes. C'est pourquoi des extensions variées de ce formalisme ont vu le jour, permettant la modélisation et la simulation de modèles complexes dans des cas précis, ou au contraire s'appliquant à un grand nombre d'applications. Plusieurs extensions de DEVS s'avèrent utiles pour notre

problématique, néanmoins, nous avons pu constater qu'il n'existe pas une extension unique de DEVS qui nous conviendrait pour un laboratoire virtuel de la pompe biologique.

Toujours en utilisant le formalisme DEVS, des outils de modélisation et de simulation ont été développés avec pour objectif l'analyse de systèmes complexes. Ces plateformes, et surtout les laboratoires virtuels, fournissent la possibilité de vérifier plusieurs modèles, comme les modèles dynamiques ou basés agents.

L'objectif du stage va être la conception d'un laboratoire virtuel pour l'étude de la pompe biologique dans le domaine mésopélagique. La modélisation des événements qui régissent ce phénomène va donc être à la base du travail demandé, puis la validation de ces modèles (ou non). Nous aurons besoin d'un formalisme adapté de DEVS et qui répondra à la problématique de l'individu au sein d'une population (niveau micro et macro). La question de l'outil que nous allons utiliser reste ouverte : une plateforme qui existe déjà (VLE, AnyLogic, ...), ou bien nous devons développer notre propre outil générique qui répondrait à nos exigences. Nous pourrions comparer les performances de ces plateformes par la simulation d'un modèle simple de la pompe biologique.

Nous pourrions également nous intéresser aux systèmes multi-interactions (SMI) pour la modélisation de la pompe biologique. En effet, la nature biologique des systèmes modélisés dans un environnement *in virtuo* a conduit à explorer la notion de réification des interactions entre les composants de systèmes biologiques complexes. On est ainsi passé d'une approche de la modélisation initialement centrée individu vers une approche centrée interaction. Les SMA sont adaptés à l'approche individu-centrée, mais l'approche SMI permet de modéliser et simuler certains couplages de phénomènes de façon plus adaptée. Enfin, les SMI permettent à l'utilisateur d'avoir accès à la sémantique du modèle au cours de l'exécution, ce qui en facilite l'observation, la modification et ainsi l'expérience [5]. Cependant, nous n'avons pas trouvé d'extension de DEVS adaptée aux SMI, mais le CERV possède ses propres outils détachés du formalisme DEVS qui pourront s'avérer utiles.

Références

- [1] Banas, N. S., Wang, D.-P., & Yen, J. (2003). *Experimental validation of an individual-based model for zooplankton swarming*.
- [2] Bergez, J.-E., Raynal, H., & Garcia, F. (2012). *RECORD: an open platform to build, evaluate and simulate integrated models of farming and agroecosystems*. Toulouse, France.
- [3] Bratman, M. E. (1999). *Intention, Plans, and Practical Reason*.
- [4] Chow, A. C.-H. (1996). *Parallel-DEVS: A parallel, hierarchical, modular modeling formalism and its distributed simulator*. Austin, TX.
- [5] Desmeulles, G., Bonneaud, S., Redou, P., Rodin, V., & Tisseau, J. (2009). In virtuo Experiments Based on the Multi-Interaction System Framework: the RéISCOP Meta-Model. *CMES Volume 47 Numéro 3*, 299-329.
- [6] Duarte, M., Butz, B. P., Miller, S. M., & Mahalingam, A. (2008). An Intelligent Universal Virtual Laboratory (UVL). *TRANSACTIONS ON EDUCATION volume 51 numéro 1*, 2-9.
- [7] Ferber, J., Michel, F., & Baez, J. (2004). AGRE: Integrating Environments with Organizations. *E4MAS*, (pp. 48-56).
- [8] Gambiasi, N., Escude, B., & Ghosh, S. (2001). GDEVS: A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems. *International Symposium on Autonomous Decentralized Systems*, (pp. 464-469). Washington, DC.
- [9] Gao, J., Li, Y., Wang, Y., & Chen, G. (2012). Micro-Macro Modeling for Systems Biology with MR-DEVS. *Applied Mechanics and Materials*, 220-223, 2975-2982.

- [10] Goldstein, R., & Wainer, G. (2009). DEVS-Based design of spatial simulations of biological systems. *Winter Simulation Conference*, (pp. 743-754). Austin, TX.
- [11] Grimm, V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling* 115, 129-148.
- [12] Guo, G., Chen, B., Qiu, X. G., & Li, Z. (2012). Parallel simulation of large-scale artificial society on CPU/GPU mixed architecture. *Workshop on Principles of Advanced and Distributed Simulation*, (pp. 174-177).
- [13] Hall, C., & Day, J. W. (1977). *Ecosystem Modeling in Theory and Practice*.
- [14] Herga, N. R., & Dinevski, D. (2012). Using a Virtual Laboratory to Better Understand Chemistry - an Experimental Study on Acquiring Knowledge. *International Conference on Information Technology Interfaces*, (pp. 237-242). Cavtat, Croatia.
- [15] Hong, J. S., Song, H.-S., & Kim, T. G. (1997). A Real-Time Discrete Event System Specification Formalism for Seamless Real-Time Software Development. *Discrete Event Dynamic Systems Volume 7 Issue 4*, 355-375.
- [16] Hwang, M.-H. (2005). Generating Finite-State Global Behavior of Reconfigurable Automation Systems: DEVS Approach. *International Conference on Automation Science and Engineering*, (pp. 254-260). Edmonton, Canada.
- [17] Hwang, M.-H., & Choi, B.-K. (2001). GK-DEVS: Geometric and Kinematic DEVS Formalism for Simulation Modeling of 3-Dimensional Multi-Component Systems. *TRANSACTIONS Volume 18 Number 3*, 159-173.
- [18] Hwang, M.-H., Cho, S. K., Zeigler, B. P., & Lin, F. (2007). *Processing Time Bounds of Schedule-Preserving DEVS*. Tucson, AZ.
- [19] Indraprastha, A. (2011). *Computational Model of Social Interaction in Multi-agent Simulation based on Personality Traits*. Bandung, Indonesia.
- [20] Jammalamadaka, R., & Zeigler, B. P. (2007). A Generic Pattern for Modifying Traditional PDE Solvers to Exploit Heterogeneity in Asynchronous Behavior. *International Workshop on Principles of Advanced and Distributed Simulation*, (pp. 45-52).
- [21] Kim, T. G. (1995). *DEVS Research at KAIST CORE LAB : From Theory To Practice*. Taejon, Korea.
- [22] Li, Y., Li, B. H., Hu, X., & Chai, X. (2011). Formalization of Multi-Resolution Modeling Based on Dynamic Structure DEVS. *International Conference on Information Science and Technology*, (pp. 855-864). Jiangsu, China.
- [23] Minar, N., Burkhart, R., Langton, V., & Askenazi, M. (1996). *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*.
- [24] Muzy, A., & Zeigler, B. P. (2005). *Discrete event simulation of large-scale spatial continuous systems*.
- [25] Picault, S., Mathieu, P., & Kubera, Y. (2010). *PADAWAN, un modèle multi-échelles pour la simulation orientée interactions*. Lille, France.
- [26] Ragueneau, O. (2010). *Complex Ecosystems: a Lagrangian, Virtual approach of the ocean carbon biological pump*. Plouzané, France.
- [27] Ramat, E., & Preux, P. (2003). "Virtual laboratory environment" (VLE): a software environment oriented agent and object for modeling and simulation of complex systems. *Simulation Modelling Practice and Theory* 11, 45-55.
- [28] Ramat, E., Preux, P., Seuront, L., & Lagadeuc, Y. (1998). *Modélisation multi-agents de systèmes naturels - Réflexions générales et application en biologie marine*.
- [29] Stratulat, T., Ferber, J., & Tranier, J. (2009). MASQ: towards an integral approach to interaction. *AAMAS*, (pp. 813-820).
- [30] Tisseau, J. (2005). *Le virtuoscope - Propositions pour les pôles de compétitivité*. Plouzané, France.
- [31] Uhrmacher, A., & Himmelpach, J. (2006). Introducing variable ports and multi-couplings for cell biological modeling in DEVS. *Winter Simulation Conference*, (pp. 832-840). Monterey, CA.
- [32] Uhrmacher, A., Ewald, R., John, M., Maus, C., Jeschke, M., & Biermann, S. (2007). Combining micro and macro-modeling in DEVS for computational biology. *Winter Simulation Conference*, (pp. 871-880). Washington, DC.
- [33] Vangheluwe, H. (2001). *The Discrete Event System specification (DEVS) formalism*.
- [34] Wainer, G. (2009). *Discrete-Event Modeling and Simulation: A Practitioner's Approach*. Ottawa, Canada.
- [35] Wainer, G., & Giambiasi, N. (2001). *Timed Cell-DEVS: modelling and simulation of cell spaces*.
- [36] Wilensky, U. (2012). *NetLogo User Manual*.
- [37] XJ Technologies. (2005). *AnyLogic Enterprise Library Reference Guide*.
- [38] Zeigler, B. P. (1976). *Theory of Modelling and Simulation*. New York, NY.
- [39] Zeigler, B. P., & Chi, S. (1991). Symbolic discrete event system specification. *AIHAS*.
- [40] Zhang, B., Chan, W. K., & Ukkusuri, S. V. (2011). Agent-Based Discrete-Event Hybrid Space Modeling Approach for Transportation Evacuation Simulation. *Winter Simulation Conference*, (pp. 199-209).