

# Systemes multi-agents et DEVS, une petite histoire commune

Romain Franceschini<sup>1</sup>

Raphaël Duboz<sup>2,3</sup>

<sup>1</sup> Université de Corse Pasquale Paoli

<sup>2</sup> ASTRE, CIRAD, INRA, Univ Montpellier, Montpellier, France

<sup>3</sup> Sorbonne Université, IRD, UMMISCO, F-93143, Bondy

Campus Grimaldi, bât Alfonsi, BP52, 20250 Corte, franceschini\_r@univ-corse.fr

32 Avenue Henri Varagnat 93143 Bondy Cedex France, raphael.duboz@cirad.fr

## Résumé :

Depuis le début des années 2000, des propositions de formalisation avec DEVS et ses extensions des systèmes multi-agents ont été proposées. Cet exposé se propose de visiter les plus emblématiques d'entre-elles en les restituant dans le contexte de leur développement, c'est-à-dire en essayant d'identifier à quels types de problèmes elles se proposent de répondre et quels sont les apports et les limites de ces propositions. Nous essayerons ensuite de faire une synthèse de ces propositions afin de discuter des développements à venir, notamment dans les domaines de la vérification et de la communication des SMAs.

## Mots-clés :

DEVS, Agents, SMA, MAS, ABM, formalisation, reproductibilité.

## Abstract:

Since the early 2000s, multi-agent systems formalisation attempts using DEVS and its extensions have been proposed. Among them, this paper proposes to review the most emblematic by restoring them in their development context. To this end, we try to identify the problems addressed by each proposal, their contributions as well as their limits. Finally, we try to synthesise the reviewed works in order to discern what in the agent-based simulations can be formalised using DEVS, and what cannot be formalised.

## 1 Introduction

Le paradigme agent est particulièrement bien adapté à la modélisation et la simulation de systèmes dynamiques complexes. Il permet de décrire des systèmes en utilisant des concepts abstraits comme les agents, leurs environnements physiques ou sociaux, les interactions, l'autonomie, la communication, *etc.* Ces concepts peuvent s'incarner dans un grand nombre de cas d'applications, ce qui permet au paradigme d'être utilisé dans de nombreux domaines scientifiques et techniques. Les Systèmes Multi-Agents (SMAs) consti-

tuent l'ensemble des systèmes physiques et computationnels qui adoptent ce paradigme. Les SMAs permettent la modélisation d'un système au niveau des éléments le constituant et de leurs interactions. Ils encouragent donc l'adoption d'un niveau d'analyse fin et permettent la simulation de phénomènes émergents. De plus, les éléments comme les interactions peuvent être hétérogènes, c'est-à-dire de différentes natures, augmentant d'autant plus le potentiel descriptif des SMAs.

Une conséquence de ce grand potentiel de description est que la spécification et l'implémentation des SMAs est difficile, et les comportements simulés parfois aussi complexes que les phénomènes qu'ils modélisent [1], ce qui peut poser des problèmes de crédibilité lorsque les SMAs sont utilisés comme support d'aide à la décision. De nombreuses propositions ont été faites pour formaliser ou décrire les SMAs, mais aucune n'est suffisamment générale pour permettre la spécification des SMAs dans leur ensemble. La variété et parfois l'absence de spécification formelle et opérationnelle des SMAs, ainsi que les difficultés liées à leur communication, leur vérification et leur reproductibilité a été soulevé à plusieurs reprises [2, 3, 4, 1, 5, 6]. La communication des méthodes et des outils ainsi que la vérification et la validation des résultats obtenus sont à la base de l'activité scientifique. Ces étapes augmentent la crédibilité des modèles et devraient être systématiques dans le cadre d'une utilisation scientifique des SMAs. Même si le protocole ODD proposé

par V. Grimm [7] représente une avancée importante, il est largement perfectible [1]. Ainsi en 2020, la communication et la vérification des SMAs est toujours un frein important à leur utilisation, notamment pour l'aide à la décision [8].

La Théorie de la Modélisation et de la Simulation (TM&S), initialement proposée par B.P. Zeigler dans les années 70 et largement développée depuis [9], semble prometteuse pour répondre au défi de la crédibilité scientifique des SMAs. Néanmoins, malgré les nombreux travaux rapprochant la TM&S et les SMAs, force est de constater que la TM&S n'est que très peu utilisée pour les SMAs. Dans cet article, nous proposons d'initier une discussion pour identifier et comprendre les raisons pour lesquelles DEVS est peu utilisé pour les SMAs. Notre objectif est didactique, avec l'ambition d'ouvrir un débat et potentiellement de nouvelles pistes de recherche communes aux deux domaines.

Nous commençons par présenter de manière informelle en 2 certaines notions de la TM&S qui sont importantes pour comprendre comment cette théorie s'applique dans le cadre des SMAs. Nous présentons ensuite une revue de la littérature s'intéressant au rapprochement de la TM&S et des SMAs 3. Nous terminons par une discussion en section 4 avant de conclure section 5.

## 2 Théorie de la modélisation et de la simulation (TM&S)

### 2.1 Éléments de la TM&S pouvant intéresser les SMAs

Nous introduisons ici des éléments de la TM&S intéressants dans le cadre des SMAs. Nous invitons le lecteur intéressé par une introduction complète à la TM&S à se référer à la dernière édition du livre qui lui est consacrée [9].

La TM&S propose des outils formels et opérationnels pour la spécification et la simulation des systèmes dynamiques discrets, continus et hybrides. Les aspects formels reposent sur les mathématiques des systèmes. La notion de couplage de modèles est centrale dans la TM&S qui propose une composition modulaire et hiérarchique des modèles qualifiés d'atomiques, au sens d'indécomposables, et des simulateurs associés. Les aspects formels garantissent des propriétés comme le respect du principe de causalité et de l'avancement du temps, ou encore la fermeture sous couplage qui assure par formellement par construction d'équivalence que le résultat du couplage de plusieurs modèles (composition) est un modèle (composé) qui conserve les propriétés dynamiques et structurelles des modèles composants. Ces propriétés sont fondamentales car elle assurent que tout modèle couplé peut être vu comme un modèle atomique, permettant la composition modulaire et hiérarchique des modèles en utilisant une même interface pour l'échange de données entre modèles et une également identique pour leur synchronisation. Le formalisme DEVS (*Discrete Event Systems Specification*) classique fut le premier formalisme développé dans le cadre de la TM&S. DEVS sépare explicitement le modèle de son simulateur. Le premier spécifie formellement le système étudié, le second correspond aux algorithmes de simulations, appelés simulateurs abstraits pour marquer leur non dépendance à un langage ou une infrastructure matérielle particulière. DEVS possède en ce sens une sémantique opérationnelle, qui permet d'exécuter les simulateurs (la simulation) de manière non-ambiguë et conforme au modèle. Les modèles DEVS atomiques sont formellement proches des automates temporisés, la principale différence est l'ajout d'une fonction de transition interne au modèle qui lui confère une dynamique autonome en plus d'une dynamique réactive spécifiée par une fonction de transition externe (déclenchée par l'arrivée d'événements extérieurs au modèle). Une fonction de sortie permet d'envoyer des données vers l'extérieur

Tableau 1 – Notions de base du paradigme Agent en DEVS

Aspects des SMAs	Formalisation DEVS	Commentaires
Agent	Modèle atomique ou couplé	Dépend de la complexité interne de l'agent
Environnement	Modèle atomique ou couplé	Dépend de la complexité interne de l'environnement
Interaction	Évènements	La structure des évènements sont à définir selon les besoins du modèle
Autonomie	Fonctions de transition interne	Fonctions à définir. Toute dynamique calculable est possible. Fonctions pouvant ne pas être formalisées mathématiquement
Proaction	Fonctions de sortie	Idem
Réaction	Fonctions de transition externe	Idem
Connaissances, croyances, intentions	Formalisés comme des états des modèles	Pas de restriction quand à la spécification des états
Ordonnancement	Fonctions d'avancement du temps	Avancement à pas de temps variables, équivalent au temps discret si les fonctions d'avancement du temps se réduisent à une constante

du modèle. Un modèle DEVS couplé correspond à un réseau dont les noeuds sont des modèles atomiques et les liens les connexions entre modèles qui s'échangent des données. Depuis les années 70, de nombreuses extensions au formalisme DEVS ont été proposées. Ces extensions enrichissent le formalisme original et permettent de prendre en compte des aspects majeurs de la M&S contemporaine. Les extensions pouvant être utiles pour les SMAs sont :

- PDEVS [10] – ajoute la gestion explicite de la simultanéité des évènements,
- DSDEVS [11] et DSDE [12] – ajoute la dynamique structurelle (changement de la composition et de la topologie du graphe des connexions entre modèles),
- Cell-DEVS – pour la représentation discrète de l'espace [13],
- STDEVS [14] – Formalise les aspects stochastiques en DEVS.

Toute extension de DEVS est formellement équivalente au DEVS classique. En d'autres termes, tout modèle formalisé dans une extension a un équivalent en DEVS classique. Les extensions apportent une spécialisation syntaxique et algorithmique sans altérer les propriétés de DEVS citées plus haut. En plus des extensions, H. Vangheluwe a montré que DEVS est suffisamment général pour spécifier

des modèles écrits dans une grande variété de formalismes [15]. Il permet donc de coupler des modèles hétérogènes, au sens des formalismes et de leur simulateur. Ainsi, DEVS apparaît comme une base formelle et opérationnelle pouvant être utilisée pour la spécification des SMAs, permettant ainsi leur communication et leur vérification algorithmique, ainsi que formelle quand c'est possible.

## 2.2 Analogie entre les SMAs et DEVS

Sans entrer dans les détails formels, qui sont donnés par la littérature citée dans la section 3, nous exposons dans le tableau 1 quelques éléments de l'analogie entre DEVS et les SMAs qui illustrent comment le premier peut formaliser le deuxième. Pour l'ensemble des concepts SMAs cités dans le tableau 1, tous ont un équivalent possible en DEVS. Il est donc possible d'utiliser ce formalisme comme base de représentation sans sacrifier les concepts inhérents aux SMAs. Bien que le domaine des SMAs soit assez hétéroclite, il est possible de capturer bon nombre de ses variantes, e.g. AC-MAS (SMA centrés agents), OCMAS (SMAs centrés organisation) et ICMAS (SMAs centrés

interaction) puisqu'il est possible de décrire des agents réactifs, cognitifs et hybrides; des environnements spatialisés et organisationnels et de représenter l'interaction comme abstraction de première classe.

### 3 Historique des SMAs en DEVS

Cette section présente d'abord les liens existants entre la communauté SMA et la communauté DEVS à travers une analyse bibliométrique de la littérature montrant l'évolution de l'intérêt d'un tel rapprochement. Nous présentons ensuite divers travaux théorisant l'approche.

#### 3.1 Évolution de l'intérêt SMA-DEVS

Dès 1994, la communauté de la TM&S a exprimé un intérêt pour la spécification de modèles agent à l'aide du formalisme DEVS. La figure 1 montre l'évolution du nombre de publications concernant DEVS et les SMAs entre 1994 et 2019. Ces résultats sont issus d'une recherche dans les bases de données d'articles *Scopus* et *Web of Science* dans les champs *titre*, *résumé* et *mots-clés* des articles avec la requête suivante: `devs AND (mas OR agent OR abm) AND simulation`.

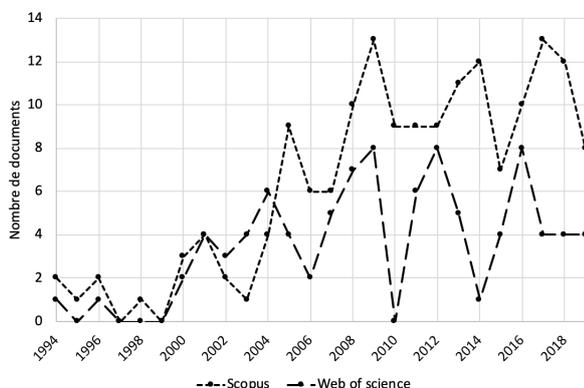


Figure 1 – Évolution du nombre de papiers traitant de DEVS et des SMAs en fonction des années pour chaque banque de publications.

Nous avons ensuite analysé les références obtenues depuis les différentes sources afin d'éliminer les doublons et les faux positifs. Sur un total de 255 références trouvées, 61 sont des doublons et 53 sont des faux positifs principalement issus de résumés d'actes de conférences dans le domaine de la simulation. Les 144 références restantes ont été triées en trois catégories : *application* (100), *théorie* (37) et *outil* (7). Les papiers comprenant une formalisation des SMAs ou faisant des liens entre concepts issus des différentes communautés sont dans la catégorie *théorie*, y compris ceux incluant un modèle illustratif ou mentionnant un outil associé à l'approche. Les papiers dans la catégorie *application*, qui est la plus représentée, présentent des modèles orientés agent spécifiés à l'aide du formalisme DEVS ou de ses variantes, de façon *ad-hoc*, ou basé sur la théorie proposée par un autre article. Les modèles mixtes (SMA + DEVS) font également partie de cette catégorie. Enfin, la catégorie *outil* comprend les papiers présentant un outil logiciel basé sur le formalisme DEVS et faisant mention explicite des SMAs comme application possible. La synthèse de cette analyse est donnée par la figure 2.

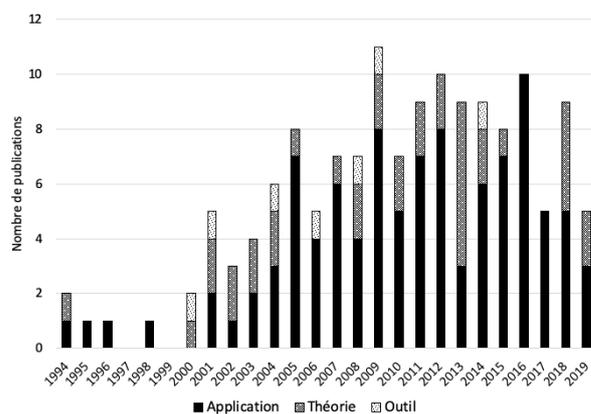


Figure 2 – Nombre et catégories de papiers traitant de DEVS et des SMAs par année.

Les deux seules bases de données de publication utilisées et la requête formulée ne permettent pas une découverte exhaustive des publications disponibles à ce jour. Néanmoins, nous pensons que ce parcours de la littérature

Tableau 2 – Types d’approches utilisant DEVS pour les SMAs.

Approche	Avantages	Difficultés
Co-simulation	Réutilisation des simulateurs existants	Sémantique ambiguë, simulateurs “boîtes noires”
Analogies	Adaptabilité	Maîtrise de DEVS + extensions requise
Extensions	Formalisation unique	Définition d’une extension

illustre bien l’évolution de l’intérêt de la communauté de la TM&S pour les SMAs. Dans la section suivante, nous complétons ces résultats en présentant des travaux qui nous semblent emblématiques de l’utilisation de DEVS pour les SMA.

### 3.2 Travaux emblématiques

La Figure 3 donne une vue d’ensemble des relations entre les différents travaux connus des auteurs à ce jour et qui proposent une formalisation des SMA en DEVS. Le tableau 2 classe les différents approches présentées en 3 catégories : *analogies*, *extensions*, et *co-simulation*.

L’ensemble des contributions sont détaillées selon ces aspects.

**Co-simulation.** La co-simulation consiste à coupler des simulateurs existants. DEVS permet l’encapsulation de simulateurs en utilisant la notion de bus logiciel [16]. Il est donc possible de coupler différents simulateurs SMA à l’aide de cette approche. C’est ce qui a été fait dans la plateforme MECSYCO [17], qui permet d’encapsuler des simulateurs SMA tels que NetLogo dans un composant DEVS, et ainsi de le coupler à d’autres simulateurs SMA et modèles DEVS [18]. Si cette approche facilite la réutilisabilité des modèles et évite leur réécriture et reformalisation, elle rend difficile la maîtrise de l’ordonnancement des agents au

sein de chaque simulateur. À notre connaissance ce travail n’a pas d’équivalent, comme illustré par la Figure 3.

**Analogies.** Le principe général est de faire une analogie entre les concepts propres au SMAs et le formalisme DEVS. Cette analogie suppose de faire des choix dans les définitions des concepts propres aux SMAs, comme celles de l’agent, de l’action et la réaction, de l’autonomie et de la proactivité, de la communication, de l’environnement etc. Ces choix sont arbitraires. Ils ont donc donné lieu à des propositions différentes.

À notre connaissance, les premiers travaux de ce type ont été proposés par A. Uhrmacher [19, 20]. Ils posent les bases de l’analogie entre les propriétés d’un agent et la structure d’un modèle DEVS, par exemple entre la perception d’un agent et la fonction de transition externe d’un modèle atomique DEVS. Les travaux utilisant l’analogie guident le modélisateur dans la représentation des entités constituant un SMA. Ils vont parfois plus loin en proposant des modèles génériques permettant à la fois de standardiser la représentation du SMA et de faciliter la réutilisation de modèles existants.

La spécification DSDEMAS [21] est une proposition basée sur l’extension DSDEVS [11] permettant la dynamique structurelle. Une interface particulière entre les composants du SMA est donnée, avec des gabarits de modèles permettant de faciliter la spécification de ces entités (agents, environnements). L’utilisation d’extensions DEVS supplémentaires est suggérée pour représenter certains types d’environnements, tels que Cell-DEVS pour représenter des environnements physiques discrets. Ces aspects sont élaborés dans [22], qui se base sur DSDE plutôt que DSDEVS. Un exemple d’architecture d’agent complexe se basant sur DSDEMAS et sur l’architecture Beliefs-Desire-Intentions [23] est également proposé dans [24].

Par la suite, ces travaux, avec la théorie *influence/réaction* [25] et le formalisme

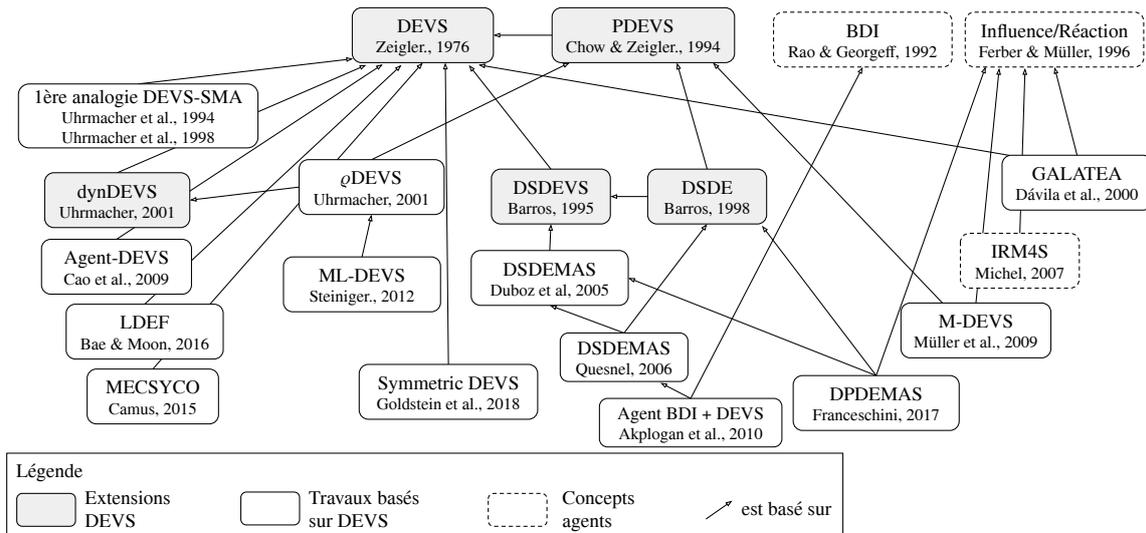


Figure 3 – Vue d'ensemble des contributions visant à formaliser les SMA en DEVS.

IRM4S [3] ont fortement influencé la spécification DPDEMAS [26]. DPDEMAS est basé sur PDEVS pour la gestion de la simultanéité des actions (influences) et rendre explicite leur effet (réaction) sur l'environnement. DSDE est utilisé pour la dynamique structurelle. Afin de favoriser la réutilisabilité des modèles, DPDEMAS formalise de façon générique la structure et le comportement autant que possible des entités actives et passives du SMA : agents, dynamique structurelle, environnement, interactions, perception automatisée et structuration de la gestion des influences. Cette approche permet de proposer une bibliothèque de modèles réutilisables (types d'environnements, influences). Dès lors, le modélisateur n'a que le comportement des agents et la gestion de la réaction de l'environnement aux influences à spécifier.

**Formalismes.** Plusieurs contributions définissent un formalisme permettant de représenter des SMA. Tous s'inspirent de DEVS, certains sont des extensions DEVS, d'autres se basent dessus sans montrer la propriété de fermeture sous couplage, indispensable pour considérer un formalisme comme extension DEVS. Néanmoins, il s'agit là encore de faire une analogie entre les SMA et DEVS.

Par exemple, pour intégrer les changements dynamiques de la topologie et de la composition du graphe de modèles (i.e. d'agents) couplés, propriétés centrale dans les SMA, A. Uhrmacher définit dynDEVS [27] et par la suite  $\rho$ -DEVS [28]. La notion d'environnement apparaît avec ML-DEVS [29] qui associe un comportement à un modèle couplé qui joue le rôle d'environnement.

Le modèle d'action *influence/réaction* développé par J. Ferber et J.P. Müller [25] au sein de la communauté SMA a eu une influence importante sur les travaux qui ont suivi. Ce modèle qui décrit les actions des agents (influences) et leur effet sur l'environnement (réaction) est repris dans la plateforme GALATEA [30] qui offre un langage de modélisation permettant de simuler les modèles DEVS qui en résultent. La gestion de la simultanéité rappelle la façon dont PDEVS [10] traite les évènements. F. Michel [3] s'inspire également de cette théorie, à laquelle il ajoute la temporalité, indispensable à la simulation, et propose le formalisme IRM4S. Ce n'est pas du DEVS mais un lien avec la sémantique de PDEVS est établi, c'est pourquoi nous la citons ici. J.P. Müller fait également le lien avec la sémantique de PDEVS et propose M-DEVS [31] qui rassemble certains des concepts

du paradigme agent au sein d'un même formalisme en incluant la gestion de la simultanéité et de la dynamique structurelle. Une restriction sur la représentation du temps empêche la réaction instantanée d'un composant.

Le formalisme LDEF [32], fortement inspiré de DEVS, favorise la réutilisabilité des modèles à travers une définition modulaire et hiérarchique des SMAs : les agents sont par exemple décomposés en différentes structures, y compris leur comportement. Plutôt que de spécifier les composants d'un SMA en DEVS, les auteurs définissent leurs propres structures puis montrent dans un second temps que LDEF est homomorphe à DEVS.

D'autres travaux tentent également de définir une extension permettant de réunir les qualités nécessaires à la définition des SMAs, tels que Agent-DEVS [33]. Plus récemment, Symmetric DEVS [34] propose une approche où les paramètres d'initialisation sont passés via *data flow*, la simulation repose sur DEVS et les agents sont représentés par des structures identifiables au sein d'un composant de type collection, qui permet la dynamique structurelle.

## 4 Discussion

### 4.1 Évolution du rapprochement entre la TM&S et les SMAs

En l'état, la Figure 1 suggère un intérêt grandissant pour la problématique. Les catégories sont données à titre indicatif et ne sont pas toujours représentatives puisque les contributions de certains articles sont à cheval sur 2 d'entre elles, voire les 3. Néanmoins, ce parcours de la littérature illustre l'intérêt croissant de la communauté de la M&S pour l'utilisation du formalisme DEVS dans le cadre des SMAs. Globalement, la moyenne du nombre de publications durant les trois dernières décennies a augmenté : 0.83 articles par an entre 1994 et 1999, 5.8 pour la période 2000-2009 et 8.1 entre 2010 et 2019. La majorité des articles étant liée à des applications semble montrer qu'il n'y a pas

besoin de définir une spécification ou une extension particulière de DEVS pour les SMAs. Pourtant, il nous semble intéressant de continuer les efforts de formalisation pour améliorer la reproductibilité des SMAs, étant donné que la plupart des modèles agents spécifiés en DEVS l'ont été de façon ad-hoc. Une telle formalisation permettrait à terme le développement de bibliothèques de modèles agents où DEVS n'apparaîtrait pas puisque présent uniquement au niveau du simulateur, assurant ainsi au modélisateur utilisant les SMAs de ne pas avoir à connaître DEVS tout en bénéficiant des apports d'une sémantique opérationnelle rigoureuse.

### 4.2 Difficultés de DEVS pour les SMAs

**Algorithmes et interface fonctionnelle.** L'interface fonctionnelle de DEVS peut être difficile à appréhender. Des simplifications ont été proposées, comme par exemple l'utilisation de la notion physique de temps super dense par J. J. Nutaro [35] pour supprimer la difficulté liée aux états de durée nulle. Une retombée possible de son travail est une interface de programmation DEVS similaire à celles des plateformes SMAs existantes. Les travaux de [36] proposent un outil DEVS ayant pour contrainte de réduire la courbe d'apprentissage du formalisme. Ces idées pourraient être également appliquées pour des modèles DEVS ciblant le domaine des SMAs.

**Représentation de l'espace et utilisation de variables globales.** Si l'espace est continu, les difficultés de modélisation ne sont pas inhérentes à DEVS. Par contre si l'espace est discret, l'intérêt de DEVS est d'adopter une approche visant à coupler des modèles de « zones » entretenant des relations de voisinages. Dès lors, les dynamiques liées à la contrainte de localité sont plus compliquées à modéliser. Par exemple, la perception d'un agent implique une requête de la zone sur laquelle il est situé aux zones voisines (selon la portée de perception de l'agent) et l'attente de leurs réponses. Notons toutefois

que cette complexité une fois modélisée au niveau du modèle de zone peut être masquée par un modèle couplé. Cette difficulté est inhérente aux couplages entre modèles DEVS, qui sont unidirectionnels. Pour ces mêmes raisons, l'accès en lecture et écriture à une variable globale (commune à tous les modèles-agents) nécessite plus d'opérations en DEVS que dans les outils SMA. Des solutions comme l'approche multi-composant DEVS permettent de faciliter ces dynamiques. Le formalisme multiPDEVS [37] mentionne explicitement les environnements SMAs comme application.

**La perception passive.** Il est souvent nécessaire de s'assurer que suite à l'action d'un agent, les agents inactifs puissent tout de même percevoir les changements d'états de l'environnement qui les concernent. L'architecture proposée par DPDEMAs [26] apporte une solution à cette problématique en situant l'ensemble des actions effectuées par les agents. Le calcul des perceptions peut alors être automatisé en tenant compte de la portée des agents.

**Simulation interactive.** La possibilité pour l'utilisateur d'interagir avec les simulations est importante pour les SMAs, particulièrement dans le cadre du prototypage et de la modélisation participative. Cette possibilité est peu discutée dans la communauté de la TM&S, et à notre connaissance jamais abordée quand la TM&S s'intéresse aux SMAs. Dans le domaine des systèmes dynamiques, par définition, les valeurs initiales des variables et des paramètres ne sont jamais affectées par la dynamique d'un modèle. DEVS ne formalise donc pas l'initialisation des modèles, ni leur observation, qui par définition ne modifie pas l'état du système. Dans le cadre de simulations interactives, ces deux caractéristiques de DEVS obligent à modéliser explicitement l'initialisation et l'observation. Une extension DEVS a été proposée récemment pour ajouter explicitement l'état initial à l'état total du modèle [38].

### 4.3 Apports de DEVS aux SMAs

**La gestion du temps.** Comme nous l'avons déjà dit, DEVS permet la modélisation et la simulation hybride (couplage de dynamiques à événements discrets, temps discret et temps continu). La plupart des plateformes SMAs considèrent le temps comme discret. Par construction, DEVS permet la modélisation d'interactions asynchrones.

Parmi les facteurs pouvant influencer les résultats d'une simulation orientée agent, l'ordonnement des agents est important. Le fait qu'une multitude d'agents réalisent leurs actions de manière simultanée dans un environnement commun peut entraîner des trajectoires d'états différentes en fonction de la représentation du temps et de l'ordre d'activation des agents. Les plateformes SMAs effectuent des choix différents vis-à-vis de cette activation (séquentielle, aléatoire, ordonnée) et du changement d'état de l'environnement (modification en place ou non). DEVS permet de gérer de façon explicite les actions simultanées des agents au sein du modèle d'environnement. Il apporte donc une solution à la problématique de façon similaire au principe *influence/réaction*. Cependant, cette approche ne semble pas avoir été adoptée par la communauté SMA. En effet, la modélisation du comportement de l'agent et de l'effet de ses actions est différente des approches proposées par les plateformes SMAs et requiert un apprentissage. La mécanique étant similaire à DEVS, l'adoption de DEVS pour décrire les SMAs permettra peut-être de démocratiser cette approche.

**Bibliothèque de modèles.** Parmi les avantages des plateformes SMAs, les fonctionnalités à disposition du modélisateur sont nombreuses : modèle d'environnement existant ou encore fonctions facilitant l'interaction entre agents participent grandement à l'attrait et l'utilisabilité des outils. DEVS étant par définition modulaire, les simulateurs DEVS offrent également la possibilité de construire des bibliothèques

de modèles, comme VLE [39] par exemple. Ces bibliothèques peuvent par exemple suivre la sémantique *System Entity Structure* (SES) proposée par B.P. Zeigler [40] pour sélectionner un modèle parmi plusieurs variantes représentant un même système.

**Formalisation, communication et vérification.** L'utilisation de DEVS constitue un pas vers la reproductibilité des expériences numériques, puisqu'il est doté d'une sémantique précise. Le partage de modèles SMAs spécifiés en DEVS permet de se rapprocher du degré de "recherche vérifiable" décrit par [41], où les mêmes conclusions peuvent être atteintes indépendamment du code source fourni par les auteurs d'une publication.

## 5 Conclusion

Dans ce travail, nous avons exploré les travaux au carrefour de deux communautés : celle des SMAs et de la TM&S. Convaincus des bénéfices que le rapprochement de ces communautés peut apporter à l'une et à l'autre, nous avons dédié une première section permettant de détailler ces derniers : spécification formelle et sémantique précise pour les SMAs ; élargissement des classes de systèmes représentables pour la TM&S. Une analogie des concepts issus des SMAs avec ceux du formalisme DEVS a été brièvement exposée. Un nombre important de chercheurs ayant déjà établi ce lien, nous avons réalisé une synthèse des travaux existants dans une seconde partie. Une analyse quantitative nous a permis de montrer : (1) l'évolution de l'intérêt pour cette problématique et (2), une part importante de travaux réalisant ce lien entre les communautés de façon *ad-hoc*. L'analyse de travaux de formalisation des SMAs en DEVS a été proposée et nous a permis d'établir les liens d'influence entre ces travaux ainsi que leurs objectifs, leurs avantages et leurs inconvénients. DEVS, bien qu'il présente de nombreux avantages, a également sa part de problématiques que nous abordons dans une discussion et aux-

quelles nous apportons notre éclairage. Nous espérons que cette vue d'ensemble suscitera chez le lecteur un intérêt pour la spécification de SMAs en DEVS.

Au-delà de fournir une bibliothèque de modèles réutilisable au sein d'un outil de simulation DEVS, nous pensons que se tourner vers l'Ingénierie Dirigée par les Modèles (IDM) constituerait un pas important vers une plus grande adoption de DEVS par la communauté SMA. Parmi les différents travaux permettant de spécifier des SMAs en DEVS, aucun ne tire parti de l'IDM à notre connaissance. Cette approche présente pourtant de nombreux avantages, puisqu'elle permet de proposer un langage spécifique au domaine (DSL) pour les SMAs, et donc au modélisateur de manipuler directement les concepts du domaine agent sans connaissance de DEVS. L'IDM permet d'automatiser plusieurs étapes du processus de M&S, telles que la vérification de modèles via un langage de contrainte (e.g. OCL) ou encore la génération automatique d'un modèle via des règles de transformation, réduisant ainsi les biais potentiels associés à une traduction manuelle d'un modèle. À ce titre, nous pensons que l'IDM représente une suite logique des travaux de rapprochement DEVS-SMA déjà entamés.

Si les difficultés propres à DEVS font parti des raisons pour lesquelles la TM&S est peu représentée dans la communauté des SMAs, nous espérons que cet article permettra un dialogue constructif entre les deux communautés avec comme perspective un enrichissement mutuel.

## Références

- [1] R. DUBOZ, B. BONTÉ et G. QUESNEL, « Vers une spécification des modèles de simulation de systèmes complexes », *Stud.*

- Inform. Univ.*, vol. 10, no. 1, p. 7–37, 2012.
- [2] J. ROUCHIER, « Re-implementation of a multi-agent model aimed at sustaining experimental economic research : The case of simulations with emerging speculation », *Journal of Artificial Societies and Social Simulation*, vol. 6, Oct 2003.
- [3] F. MICHEL, « Le modèle IRM4S. De l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents », *Revue d'Intelligence Artificielle*, vol. 21, no. 5-6, p. 757–779, 2007.
- [4] U. WILENSKY et W. RAND, « Making models match : Replicating an agent-based model », *Journal of Artificial Societies and Social Simulation*, vol. 10, Oct 2007.
- [5] K. BAJRACHARYA et R. DUBOZ, « Comparison of three agent-based platforms with a simple epidemiological model (wip) », in *Theory of Modeling & Simulation Symposium, SpringSim Multi-Conference*, p. 362–367, April 2013.
- [6] E. DONKIN, P. DENNIS, A. USTALAKOV, J. WARREN et A. CLARE, « Replicating complex agent based models, a formidable task », *Environmental Modelling & Software*, vol. 92, p. 142–151, juin 2017.
- [7] V. GRIMM, U. BERGER, D. L. DEANGELIS, J. G. POLHILL, J. GISKE et S. F. RAILSBACK, « The odd protocol : A review and first update », *Ecological Modelling*, vol. 221, p. 2760–2768, 2010.
- [8] L. AN, V. GRIMM et B. L. TURNER II, « Editorial : Meeting grand challenges in agent-based models », *Journal of Artificial Societies and Social Simulation*, vol. 23, no. 1, p. 13, 2020.
- [9] B. P. ZEIGLER, A. MUZY et E. KOFMAN, *Theory of Modeling and Simulation. Discrete Event & Iterative System Computational Foundations*, Academic Press, 3rd edition éd., 2019.
- [10] A. C. H. CHOW et B. P. ZEIGLER, « Parallel devs : A parallel, hierarchical, modular, modeling formalism », in *Proceedings of the 26th Conference on Winter Simulation*, p. 716–722, Society for Computer Simulation International, 1994.
- [11] F. J. BARROS, « Dynamic Structure Discrete Event System Specification : A New Formalism for Dynamic Structure Modeling and Simulation », in *Proceedings of the 27th Conference on Winter Simulation*, (Arlington, Virginia, USA), p. 781–785, IEEE Computer Society, 1995.
- [12] F. J. BARROS, « Modeling formalisms for dynamic structure systems », *ACM Trans. Model. Comput. Simul.*, vol. 7, p. 501–515, oct. 1997.
- [13] G. A. WAINER et N. GIAMBIASI, « Application of the Cell-DEVS Paradigm for Cell Spaces Modelling and Simulation », *SIMULATION*, vol. 76, p. 22–39, jan. 2001.
- [14] R. CASTRO, E. KOFMAN et G. WAINER, « A formal framework for stochastic discrete event system specification modeling and simulation », *SIMULATION*, vol. 86, no. 10, p. 587–611, 2010.
- [15] H. VANGHELUWE, « Devs as a common denominator for multi-formalism hybrid systems modelling », in *IEEE International Symposium on Computer-Aided Control System Design, 2000. CACSD 2000*, p. 129–134, IEEE, 2000.
- [16] Y. J. KIM, J. H. KIM et T. G. KIM, « Heterogeneous simulation framework using devs bus », *Simulation*, vol. 79, p. 3–18, 2003.
- [17] B. CAMUS, *Environnement Multi-agent pour la Multi-modélisation et Simulation des Systèmes Complexes*. Thèse doctorat, Université de Lorraine, Nov 2015.
- [18] T. PARIS, L. CIARLETTA et V. CHEVRIER, *Designing Co-simulation with Multi-agent Tools : A Case Study with Net-Logo*, vol. 10767, p. 253–267. Springer International Publishing, 2018.

- [19] A. M. UHRMACHER et R. ARNOLD, « Distributing and maintaining knowledge : agents in variable structure environments », in *Fifth Annual Conference on AI, and Planning in High Autonomy Systems*, p. 178–184, IEEE Comput. Soc. Press, déc. 1994.
- [20] A. M. UHRMACHER et B. SCHATTENBERG, « Agents in discrete event simulation », in *European Simulation Symposium*, 1998.
- [21] R. DUBOZ, D. VESMISSE, G. QUESNEL, A. MUZY et E. RAMAT, « Specification of Dynamic Structure Discret event Multiagent Systems », in *Agent-directed simulation, part of the 2006 Spring Simulation Multiconference*, (Huntsville, AL, USA.), p. 23–30, SCS, avril 2005.
- [22] G. QUESNEL, *Approche formelle et opérationnelle de la multi-modélisation et de la simulation des systèmes complexes*. Thèse doctorat, Université du Littoral - Côte d’Opale, déc. 2006.
- [23] A. S. RAO et M. P. GEORGEFF, « An abstract architecture for rational agents », in *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, p. 439–449, Oct 1992.
- [24] M. AKPLOGAN, G. QUESNEL, F. GARCIA, A. JOANNON et R. MARTIN-CLOUAIRE, « Towards a deliberative agent system based on devts for application in agriculture », in *Proceedings of the 2010 Summer Computer Simulation Conference*, p. 250–257, Jul 2010.
- [25] J. FERBER et J.-P. MÜLLER, « Influences and Reaction : a Model of Situated Multiagent Systems », in *Proceedings of the Second International Conference on Multiagent Systems*, 1996.
- [26] R. FRANCESCHINI, *Approche Formelle Pour La Modélisation et La Simulation à Évènements Discrets de Systèmes Multi-Agents*. PhD Thesis, Université de Corse Pasquale Paoli, déc. 2017.
- [27] A. M. UHRMACHER, « Dynamic Structures in Modeling and Simulation : A Reflective Approach », *ACM Transactions on Modeling and Computer Simulation*, vol. 11, p. 206–232, avril 2001.
- [28] A. M. UHRMACHER, J. HIMMELSPACH, M. RÖHL et R. EWALD, « Introducing Variable Ports and Multi-Couplings for Cell Biological Modeling in DEVS », in *2006 Winter Simulation Conference*, p. 832–840, IEEE, déc. 2006.
- [29] A. STEINIGER, F. KRUGER et A. M. UHRMACHER, « Modeling agents and their environment in Multi-Level-DEVS », in *2012 Winter Simulation Conference*, p. 1–12, IEEE, déc. 2012.
- [30] J. DÁVILA et M. UZCÁTEGUI, « GALATEA : A multi-agent, simulation platform », in *International Conference on Modeling, Simulation and Neural Networks*, (Mérida, Venezuela), p. 187–198, 2000.
- [31] J.-P. MÜLLER, « Towards a Formal Semantics of Event-Based Multi-agent Simulations », in *Multi-Agent-Based Simulation IX* (N. DAVID et J. S. SICHTMAN, édés), p. 110–126, Berlin, Heidelberg : Springer, 2009.
- [32] J. W. BAE et I.-C. MOON, « LDEF Formalism for Agent-Based Model Development », *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, vol. 46, p. 793–808, juin 2016.
- [33] Q. CAO, Z.-s. HE et L. YU, « Modeling and simulation for material supply in emergent disaster based on agent-devts/hla », in *2010 Third International Conference on Information and Computing Science (ICIC)*, p. 269–272, IEEE, Jun 2010.
- [34] R. GOLDSTEIN, S. BRESLAV et A. KHAN, « A symmetric formalism for discrete event simulation with agents », in *Proceedings of the 2018 Winter Simulation Conference*, WSC ’18, p. 584–595, IEEE Press, 2018.

- [35] J. J. NUTARO, « A new modeling interface for simulators implementing the discrete event system specification », in *Spring Simulation Conference (SpringSim)*, p. 1–12, The Society for Modeling and Simulation International (SCS), 2019.
- [36] R. GOLDSTEIN, S. BRESLAV et A. KHAN, « Practical aspects of the designdevs simulation environment : », *SIMULATION*, vol. 890, p. 1–39, Aug 2017.
- [37] D. FOURES, R. FRANCESCHINI, P.-A. BISGAMBIGLIA et B. P. ZEIGLER, « multipdevs : A parallel multicomponent system specification formalism », *Complexity*, vol. 2018, p. 1–19, 2018.
- [38] Y. VAN TENDELOO et H. VANGHELUWE, « Extending the devs formalism with initialization information », Feb 2018. arXiv : 1802.04527.
- [39] G. QUESNEL, R. DUBOZ et E. RAMAT, « The Virtual Laboratory Environment – An operational framework for multi-modelling, simulation and analysis of complex dynamical systems », *Simulation Modelling Practice and Theory*, vol. 17, p. 641–653, April 2009.
- [40] B. P. ZEIGLER, C. SEO, R. COOP et D. H. KIM, « Creating suites of models with system entity structure : global warming example », in *DEVS 13 : Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium*, p. 32, Society for Computer Simulation International, Apr 2013.
- [41] V. STODDEN, D. H. BAILEY, J. BORWEIN, R. J. LEVEQUE, W. RIDER et W. STEIN, « Setting the default to reproducible : reproducibility in computational and experimental mathematics », rap. tech., 2013.