

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/337950907>

# Approche DEVS/SMA pour la Modélisation et la Simulation des Systèmes Complexes

Poster · May 2016

DOI: 10.13140/RG.2.2.15934.64329

---

CITATIONS

0

READS

12

1 author:



Noureddine Seddari

Université Constantine 2

12 PUBLICATIONS 24 CITATIONS

SEE PROFILE

# Approche DEVS/SMA pour la Modélisation et la Simulation des Systèmes Complexes

Noureddine Seddari

Université Abdelhamid Mehri, Constantine 2, Algeria  
noureddine.seddari@univ-constantine2.dz

**Résumé.** Un système complexe est un système composé d'un grand nombre d'entités en interaction. Dans un tel système, le comportement global émerge de l'interaction des entités qui le composent. Cette notion d'émergence mérite d'être bien définie et complètement déterminée, car, " le tout n'est pas forcément l'ensemble des parties ". Pour modéliser tels systèmes, nous avons besoin plus d'un formalisme. Ce papier propose une approche de transformation de modèle DEVS (Discrete Event System Specification) vers le modèle AGR (Agent/Groupe/Rôle) qui est un modèle conceptuel sous-jacent du modèle organisationnel AALAADIN, Il s'agit d'un mécanisme qui peut systématiquement transformer les modèles couplés et atomiques de DEVS vers le modèle AGR. La transformation de ces modèles produit un ensemble d'agents et groupes d'agents capables de l'exécuter au sein de plateformes de Systèmes Multi-Agents (SMA) basées sur AGR. En particulier, la plate-forme MAD-KIT. L'avantage de cette démarche est son adaptabilité, sa souplesse ainsi que ses possibilités d'extension.

**Mots-clés:** Systèmes Multi-Agents (SMA), Systèmes complexes, Discrete Event systems Specification (DEVS), Modèles atomiques et couplés, AGR (Agent/Groupe/Rôle). MAD-KIT, AALAADIN.

## 1 Introduction

La modélisation et la simulation [1-8] des systèmes complexe permettent de manipuler, d'observer et d'améliorer la compréhension des phénomènes mis en jeu. Ces derniers sont étudiés à différentes échelles comme pour la simulation des changements climatiques, des phénomènes physiques, des systèmes de trafic urbain.... La plupart des phénomènes modélisés atteignent aujourd'hui des complexités et des degrés de finesse élevés qui imposent l'utilisation de modèles et d'outils informatiques de plus en plus performants, flexibles et d'une complexité généralement assez élevée.

A ce stade, nous pouvons considérer, pour certains systèmes, des approches hybrides dans lesquelles les méthodes de la modélisation/simulation seraient combinées.

Notre travail se situe dans ce cadre, nous avons opté pour une méthode hybride qui combine le formalisme DEVS (*Discrete Event System Specification*) [9] et le modèle AGR (Agent/Groupe/Rôle).

Dans un premier temps, le formalisme DEVS est un concept introduit par le professeur Zeigler dans les années 1970 [9], est un formalisme qui permet de définir mathématiquement un modèle. Autrement dit, DEVS détermine les structures et les évolutions dynamiques (comportement) des entités qui composent le système à l'aide de deux modèles atomiques et couplés. La représentation DEVS encapsule la structure et le comportement du modèle. Ainsi, les modèles DEVS fournissent des représentations mathématiques qui peuvent être formellement vérifiées et validées.

Dans un deuxième temps, Les approches de Systèmes Multi-Agents (SMA) présentent des outils théoriques et pratiques solides dans le domaine de simulation. Un système multi-agents comprend des outils pour gérer les agents (création, destruction, interactions entre agents ...). A ce niveau, l'implémentation du système peut être facilement réalisée.

L'idée développée ici est basée principalement sur la transformation d'un modèle vérifié et validé en DEVS vers le modèle AGR de la plateforme MadKit, cette plateforme a été mise au point par l'équipe du professeur J. Ferber au LIRMM [10] et repose sur les principes d'agent, groupe et rôle. Un ensemble de procédures et fonctions permettant d'effectuer des passages entre le modèle DEVS et AGR sont détaillées dans la suite de cet article.

Le reste de cet article est organisé comme suit: La section 2 présente le formalisme DEVS. La troisième section est consacrée aux SMA. L'approche de transformation proposée est présentée dans la quatrième section. Finalement, nous terminons avec une conclusion et des perspectives.

## **2 Formalism DEVS**

Le formalisme DEVS [1,11] est une approche de modélisation basée sur la théorie générale des systèmes. Il s'agit, plus précisément, d'un formalisme modulaire et hiérarchique pour la modélisation basée sur le concept d'état.

DEVS a été élaboré et adopté depuis par une communauté internationale de chercheurs [11-18]. Ces travaux sont basés sur le développement d'architectures logicielles permettant d'une part de faciliter les étapes de modélisation, de simulation et de validation et d'utiliser, d'autre part, le même environnement de multi modélisation pour analyser les systèmes résultant des différents champs afin de générer automatiquement les algorithmes de simulation.

DEVS peut être considéré comme un environnement de multi-modélisation qui permet de rassembler (voire de fédérer) de manière cohérente d'autres formalismes de modélisation qui se basent eux-mêmes sur la théorie générale des systèmes. C'est, en effet, un formalisme adapté à un grand nombre de champs d'application [19-22].

## 2.1 Spécification formelle d'un modèle atomique DEVS classique

Un modèle atomique DEVS est décrit par l'équation suivante:

$$AtomicDEVS = (X, Y, S, \delta_{int}, \delta_{ext}, t_a, \lambda) \quad (1)$$

$X$  est l'ensemble des entrées externes.

$Y$  est l'ensemble des sorties du modèle.

$S$  représente l'ensemble des états.

$\delta_{int}: S \rightarrow S$  est la fonction de transition interne qui fait évoluer le système d'un état à un autre de manière autonome.

$\delta_{ext}: Q \times X \rightarrow S$  est la fonction de transition externe qui se produit lorsque le modèle reçoit un événement externe. Elle renvoie le nouvel état du système basé sur l'état actuel.

$Q = \{(s, e) \mid s \in S, 0 \leq e \leq t_a(s)\}$  ensemble total des états.  $e$  : est le temps écoulé depuis la dernière transition.

$\lambda: S \rightarrow Y$ : fonction de sortie du modèle. Elle est activée lorsque le temps écoulé dans un état donné est égal à sa durée de vie.

$t_a(s)$  indique la durée de vie d'un Etat "s" du système. C'est le temps pendant lequel le modèle demeurera dans cet état si aucun événement externe ne se produit.

## 2.2 Spécification formelle d'un modèle DEVS couplé

Un modèle DEVS couplé (équation 2) décrit un système comme étant un réseau de composants connectés. Ces composants consistent en modèles qu'ils soient atomiques ou couplés.

$$CoupledDEVS = (X_{self}, Y_{self}, D, \{M_d/d \in D\}, EIC, EOC, IC) \quad (2)$$

$self$ : est le modèle lui-même.

$X_{self}$ : est l'ensemble des entrées du modèle couplé.

$Y_{self}$ : est l'ensemble des sorties du modèle couplé.

$D$  est l'ensemble des noms associés à des éléments du modèle,  $self$  n'est pas en  $D$ .

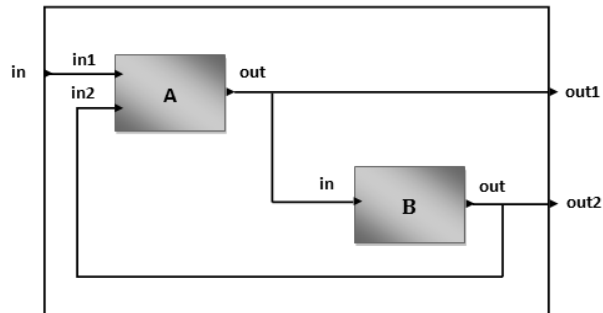
$\{M_d / j \in D\}$  est l'ensemble des composants du modèle couplé.

EIC, EOC et IC définissent la structure de couplage dans le modèle couplé.

EIC est l'ensemble des couplages externes en entrée. Ils relient les entrées du modèle couplé à celles de ses propres composants.

EOC est l'ensemble des couplages externes en sortie. Ils relient les sorties des composants à celles des modèles couplés.

IC définit le couplage interne. Il relie les sorties des composants aux entrées provenant d'autres composants dans le même modèle couplé. Toutefois, aucune rétroaction directe des boucles n'est autorisée. Ce qui signifie qu'un port de sortie d'un composant (modèle) ne peut pas être connecté à un port d'entrée du même composant. La figure 1 présente un modèle couplé se composant de deux modèles atomiques.



**Fig. 1.** Représentation graphique d'un modèle couplé (C) se composant de deux modèles atomiques (A et B).

### 3 Les systèmes Multi-Agents (SMA)

L'approche multi-agent est à l'origine de la connexion de plusieurs domaines spécifiques de l'intelligence artificielle, des systèmes informatiques distribués et des technologies du génie logiciel. C'est une discipline qui se concentre sur les comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents. Un système multi-agent est un système distribué composé d'un ensemble d'agents. Dans un SMA, l'agent peut être réactif, en ce sens qu'il dispose de capacités d'actions limitées, cognitives ; c'est-à-dire qu'il peut élaborer des plans d'actions, prendre des décisions et autres tâches 'intelligentes' ou hybride dans le cas où l'agent est à la fois réactif et cognitif.

Les interactions entre agents peuvent se faire de trois manières différentes :

- Grâce à une mémoire partagée ou tableau noir (Black Board).
- Par l'intermédiaire de messages.
- Par l'intermédiaire de l'environnement (ou monde) perceptible par l'agent.

Un système multi-agent présente, généralement, les caractéristiques suivantes :

- Il n'y a aucun contrôle global.
- Les données sont décentralisées;
- Le calcul est asynchrone.

Il existe plusieurs façons de représenter et de formaliser un agent, ainsi J.Ferber [23-25] représente un système multi-agent par le couple  $\langle A, W \rangle$  où A est un agent et W un environnement:

$$A = (P_a, \text{Percept}_a, F_a, \text{Infl}_a, S_a) \quad (3)$$

Avec

- $P_a$  : fonction de perception de l'agent.
- $\text{Percept}_a$  : ensemble des stimuli et sensations qu'un agent peut percevoir.
- $F_a$  : fonction de comportement de l'agent.
- $\text{Infl}_a$  : fonction d'action de l'agent.
- $S_a$  : ensemble des états internes de l'agent.

$$W = (E, \Gamma, \Sigma, R) \quad (4)$$

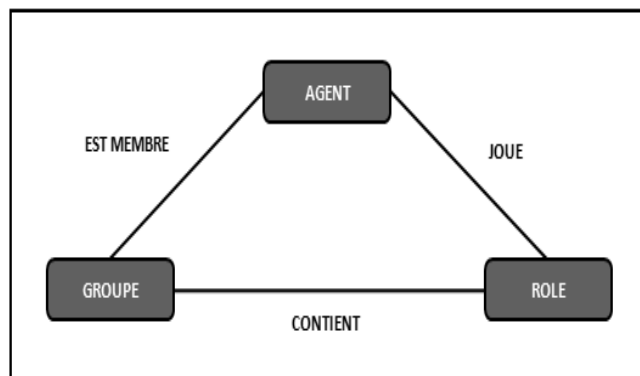
Avec

- E : espace dans lequel évolue l'agent.
- $\Gamma$  : espace des influences produites par l'agent.
- $\Sigma$  : état de l'environnement.
- R : loi d'évolution.

### 3.1 Le modèle AALAADIN

Le modèle organisationnel AALAADIN est un projet qui porte sur l'analyse, la conception, la formalisation et la réalisation de systèmes multi-agents dans une perspective organisationnelle. Le modèle conceptuel sous-jacent (AGR) est fondé sur les notions d'agents, de groupes et de rôles. Ce modèle a été développé par l'équipe du Professeur J.Ferber au LIRMM [10, 26]. La figure 2 présente brièvement ces trois concepts.

- **Agent:** Un agent est spécifié comme une entité communicante qui joue un ou plusieurs rôles dans un ou plusieurs groupes. Cette définition très générale laisse une grande liberté aux concepteurs.
- **Groupe:** Un agent fait partie d'un ou plusieurs groupes. Un groupe est un ensemble d'agents pouvant jouer un ensemble déterminé de rôles.
- **Rôle:** Le rôle est l'abstraction de l'ensemble des fonctions de l'agent (ou bien les services qu'il fournit ou son identification) au sein d'un groupe. Un agent peut jouer plusieurs rôles. Chacun des rôles est local à un groupe. Un agent peut demander à jouer un rôle dans un groupe, mais celui-ci ne lui sera pas forcément attribué. [27-31].



**Fig. 2.** Le modèle Agent/Groupe/rôle (AGR)

## 4 Approche DEVS/SMA proposée

DEVS est un formalisme qui permet de définir mathématiquement un modèle. Autrement dit, DEVS détermine les structures et les évolutions dynamiques (comportement) des entités qui composent le système à l'aide de deux modèles atomiques et couplés. La représentation DEVS encapsule la structure et le comportement du modèle. Ainsi, les modèles DEVS fournissent des représentations mathématiques qui peuvent être formellement vérifiées et validées.

Les approches SMA présentent des outils théoriques et pratiques solides dans le domaine de la modélisation. Un système multi-agent comprend des outils pour gérer les agents (création, destruction, interactions entre agents ...). A ce niveau, l'approche de conception globale et l'implémentation du système peuvent être facilement réalisées.

L'algorithme de transformation suivant détaille le passage d'un modèle DEVS couplé (CDEVSi) au modèle AGR.

### 4.1 Algorithme de transformation

L'algorithme présenté ci-dessous (Algorithme 1) permet de réaliser le passage du formalisme DEVS vers le modèle AGR. En premier lieu, la fonction CDEVSi crée les groupes  $G_i$ , puis les procédures CDEVSi\_EIC, CDEVSi\_IC et CDEVSi\_EOC définissent les différentes interconnexions entre les agents (AGA $_j$  « Agent correspondant au modèle atomique ADEVSi $_j$  » créés par la fonction ADEVSi\_AGA (ADEVSi $_j$ ), AGCi\_R « Agent représentant les ports d'entrée du modèle couplé CDEVSi » créés par la fonction et AGCi\_E « Agent représentant les ports de sortie du modèle couplé CDEVSi ») et les groupes  $G_i$  selon le type de couplage du modèle CDEVSi comme montré ci-dessous.

La fonction ADEVSi\_AGA (ADEVSi $_j$ ) : Elle permet la création des agents AGA $_j$  pour chaque modèle atomique ADEVSi $_j$ . Ainsi ; les récepteurs et effecteurs de l'agent AGA $_j$  représentent respectivement les ports d'entrée (ADEVSi $_j$ .Inports $_i$ ) et de sortie (ADEVSi $_j$ .Outports $_i$ ) du modèle atomique. La fonction de perception de l'agent AGA $_j$ .Pa correspond à la fonction de transition externe ADEVSi $_j$ . $\delta_{ext}$ . La fonction de comportement de l'agent AGA $_j$ .Fa représente la fonction de transition interne ADEVSi $_j$ . $\delta_{int}$ . La fonction d'action de l'agent AGA $_j$ .Infla correspond à la fonction de sortie ADEVSi $_j$ . $\lambda$ . L'évolution dynamique d'un agent représente la fonction d'avancement du temps ADEVSi $_j$ .ta et l'état interne d'un agent AGA $_j$ . Cela correspond à l'ensemble des états ADEVSi $_j$ .S.

La fonction CDEVSi\_AGC ( CDEVSi $_j$ ) :Elle crée deux agents coordinateurs AGC pour chaque modèle couplé CDEVSi $_i$ , le premier est un agent coordinateur récepteur AGCi\_R et le deuxième est un agent coordinateur émetteur AGCi\_E. L'objectif principal de ces deux coordinateurs est de bien conserver l'encapsulation du formalisme DEVS. Le modèle est représenté comme étant une boîte noire qui n'agit avec l'environnement que via les ports d'entrée et de sortie. Cependant les agents ont une liberté d'interaction avec leur environnement. Dans cette approche ; les agents voulant communiquer avec des agents internes d'un groupe correspondant à un modèle

DEVS couplé doivent impérativement passer par les agents coordinateurs de ce groupe.

La fonction CDEVS\_G ( CDEVS $i$  ) : Elle crée des groupes  $G_i$  pour chaque modèle couplé CDEVS $i$ , puis elle crée des agents AGA $j$  et elle va les regrouper dans ce groupe  $G_i$ . Ensuite elle crée les deux agents de coordination AGC $i$ \_R et AGC $i$ \_E grâce à la fonction CDEVS\_AGC.

---

**Algorithme 1:** Transformation CDEVS\_AGR

Input CDEVS Output AGR 1: <b>Pour</b> tous les modèles couplés CDEVS $i$ <b>faire</b> 2: Appeler CDEVS_G (CDEVS $i$ ) // Création des groupes et des agents 3: Appeler CDEVS_EIC (CDEVS $i$ ) // Définition des interconnexions // correspondant à CDEVS $i$ .EIC 4: Appeler CDEVS_IC (CDEVS $i$ ) // Définition des interconnexions correspondant // à CDEVS $i$ .IC 5: Appeler CDEVS_EOC (CDEVS $i$ ) // Définition des interconnexions // correspondant à CDEVS $i$ .EOC 6: <b>Fin pour</b>
--

---

**Fonction 1:** ADEVS\_AGA ( ADEVS : DEVS Atomique ) : AGA

1: Créer AGA $j$ // Modèle agent correspondant à ADEVS $j$ 2: Créer une boîte aux lettres Br de taille = ADEVS $j$ .Inports.Taille 3: Créer une boîte d'envoi Be de taille = ADEVS $j$ .Outports.Taille 4: <b>Si</b> un événement externe est aperçu dans un port : ADEVS $j$ .Inports <b>alors</b> 5: Exécuter AGA $j$ .Pa // Correspondant à ADEVS $j$ . $\delta_{ext}$ 6: <b>Fin si</b> 7: <b>Si</b> la durée de vie $t_a$ de l'état S de Pa est écoulé <b>alors</b> 8: Exécuter AGA $j$ .Fa // Correspondant à ADEVS $j$ . $\delta_{int}$ 9: Exécuter AGA $j$ .Infla // Correspondant à ADEVS $j$ . $\lambda$ 10: <b>Fin si</b>
--

---

**Fonction 2:** CDEVS\_G ( CDEVS : DEVS Couplé ) : G

1: Créer un groupe $G_i$ // Correspondant à CDEVS $i$ 2: <b>Pour</b> tous les ADEVS $j$ <b>faire</b> 3: ADEVS_AGA (ADEVS $j$ ) // Création des agents AGA $j$ 4: Ajouter AGA $j$ dans $G_i$ // Regroupement des agents 5: <b>Fin pour</b> 6: CDEVS_AGC (CDEVS $i$ ) 7: Ajouter AGC $i$ _E et AGC $i$ _R dans $G_i$
--



---

**Fonction 3: CDEVS\_AGC ( CDEVS : DEVS Couplé ) : AGC**

1 : Créer AGCi\_R // Agent récepteur pour CDEVS*i*.Inports  
2 : Créer une boîte aux lettres Br de taille = CDEVS*i*.Inports.Taille  
3 : Créer une boîte d'envoi Be de taille = CDEVS*i*.EIC.Taille  
4 : **Si** un événement externe est aperçu dans un port : CDEVS*i*.Inports **alors**  
5 :     Exécuter AGCi\_R.Pa  
6 :     **Fin si**  
7 : **Si** la boîte aux lettres Br non vide **alors**  
8 :     Exécuter AGCi\_R.Fa  
9 :     **Fin si**  
10 : **Si** la boîte d'envoi Be non vide **alors**  
11 :     Exécuter AGCi\_R.Infla  
12 :     **Fin si**  
13 : Créer AGCi\_E // Agent émetteur pour CDEVS*i*.Outports  
14 : Créer une boîte aux lettres Br de taille = CDEVS*i*.EOC.Taille  
15 : Créer une boîte d'envoi Be de taille = CDEVS*i*.Outports.Taille  
16 : **Si** un événement externe est aperçu dans un port : CDEVS*i*.Outports **alors**  
17 :     Exécuter AGCi\_E.Pa  
18 :     **Fin si**  
19 : **Si** la boîte aux lettres Br non vide **alors**  
20 :     Exécuter AGCi\_E.Fa  
21 :     **Fin si**  
22 : **Si** la boîte d'envoi Be non vide **alors**  
23 :     Exécuter AGCi\_E.Infla  
24 :     **Fin si**

---

**Procédure 1: CDEVS\_EIC( CDEVS : DEVS Couplé )**

1 : **Pour**  $m = 1$  jusqu'au nombre des ports d'entrée du CDEVS*i* **faire**  
2 :     **Pour**  $j = 1$  jusqu'au nombre des modèles internes M **faire**  
3 :         **Pour**  $k = 1$  jusqu'au nombre des ports d'entrée du Mj **faire**  
4 :             **Si** (CDEVS*i*.Mj = ADEVSj) **alors**  
5 :                 **Si** (CDEVS*i*.Xm = ADEVSj.Xk) **alors** // EIC entre modèle couplé  
                       //et modèle atomique  
6 :                 Définir l'ensemble des messages entre Gi.AGCi\_R et Gi.AGAj  
7 :                 **Fin si**  
8 :             **Fin si**  
9 :             **Si** (CDEVS*i*.Mj = CDEVSj) **alors**  
10 :                 **Si** (CDEVS*i*.Xm = CDEVSj.Xk) **alors** // EIC entre deux  
                       // modèles couplés  
11 :                 Définir l'ensemble des messages entre Gi.AGCi\_R et Gj.AGCj\_R  
12 :                 **Fin si**  
13 :             **Fin si**  
14 :     **Fin pour**  
15 :     **Fin pour**  
16 : **Fin pour**

---

**Procédure 2: CDEVS\_IC( CDEVS : DEVS Couplé )**

```
1 : Pour  $j = 1$  jusqu'au nombre des modèles internes  $M$  faire
2 :   Pour  $k = 1$  jusqu'au nombre des modèles internes  $M$  faire
3 :     Si  $j \neq k$  alors // un port de sortie ne doit pas être couplé avec un port d'entrée du
        // même modèle.
4 :       Pour  $n = 1$  jusqu'au nombre des ports de sortie du  $M_j$  faire
5 :         Pour  $m = 1$  jusqu'au nombre des ports d'entrée du  $M_k$  faire
6 :           Si  $(CDEVS_i.M_j = CDEVS_j)$  et  $(CDEVS_i.M_k = CDEVS_k)$  alors
7 :             Si  $(CDEVS_j.Y_n = CDEVS_k.X_m)$  alors //IC entre deux modèles couplés
8 :               Définir l'ensemble des messages entre  $G_j.AGC_j\_E$  et  $G_k.AGCK\_R$ 
9 :             Fin si
10 :          Fin si
11 :         Si  $(CDEVS_i.M_j = ADEVS_j)$  et  $(CDEVS_i.M_k = ADEVS_k)$  alors
12 :           Si  $(ADEVS_j.Y_n = ADEVS_k.X_m)$  alors // IC entre deux modèles
        //atomique
13 :             Définir l'ensemble des messages entre  $G_i.AGA_j$  et  $G_i.AGAK$ 
14 :           Fin si
15 :         Fin si
16 :         Si  $(CDEVS_i.M_j = ADEVS_j)$  et  $(CDEVS_i.M_k = CDEVS_k)$  alors
17 :           Si  $(ADEVS_j.Y_n = CDEVS_k.X_m)$  alors // IC entre un modèle
        //atomique et un modèle couplé
18 :             Définir l'ensemble des messages entre  $G_i.AGA_j$  et  $G_k.AGCK\_R$ 
19 :           Fin si
20 :         Fin si
21 :         Si  $(CDEVS_i.M_j = CDEVS_j)$  et  $(CDEVS_i.M_k = ADEVS_k)$  alors
22 :           Si  $(CDEVS_j.Y_n = ADEVS_k.X_m)$  alors // IC entre un modèle couplé
        //et un modèle atomique
23 :             Définir l'ensemble des messages entre  $G_j.AGC_j\_E$  et  $G_i.AGAK$ 
24 :           Fin si
25 :         Fin si
26 :       Fin pour
27 :     Fin pour
28 :   Fin si
29 : Fin pour
30 : Fin pour
```

---

**Procédure 3: CDEVS\_EOC (CDEVS : DEVS Couplé)**

```
1 : Pour  $j = 1$  jusqu'au nombre des modèles internes  $M$  faire
2 :   Pour  $p = 1$  jusqu'au nombre des ports de sortie du  $M_j$  faire
3 :     Pour  $q = 1$  jusqu'au nombre des ports de sortie du  $CDEVS_i$  faire
4 :       Si ( $CDEVS_i.M_j = ADEVS_j$ ) alors
5 :         Si ( $ADEVS_j.Y_p = CDEVS_i.Y_q$ ) alors // EOC entre modèle
           //atomique et modèle couplé
6 :         Définir l'ensemble des messages entre  $G_i.AGA_j$  et  $G_i.AGC_i_E$ 
7 :         Fin si
8 :       Fin si
9 :     Si ( $CDEVS_i.M_j = CDEVS_j$ ) alors
10 :      Si ( $CDEVS_j.Y_p = CDEVS_i.Y_q$ ) alors // EOC entre deux
           // modèles couplés
11 :      Définir l'ensemble des messages entre  $G_j.AGC_j_E$  et  $G_i.AGC_i_E$ 
12 :      Fin si
13 :    Fin si
14 :  Fin pour
15 : Fin pour
16 : Fin pour
```

Nous obtenons table 1 ci-dessous un ensemble de règles de passage entre DEVS et le modèle AGR conformément aux équations (1) et (2) concernant DEVS et les équations (3) et (4) concernant le modèle AGR ainsi que les algorithmes vus ci-dessus.

## 5 Conclusion et perspectives

Dans ce papier, nous avons proposé une approche basée DEVS/SMA pour la modélisation et la simulation des systèmes complexes. Pour cela, nous avons proposé une transformation de modèle DEVS vers un modèle AGR. Cette démarche est caractérisée par un ensemble de procédures et fonctions permettant la transformation systématique de ces modèles. L'avantage de notre approche est de profiter de la puissance formelle du formalisme DEVS pour la vérification et la validation d'une part et la puissance des outils évolués offerts par les plateformes SMA pour la mise en œuvre, le développement et l'implémentation d'autre part.

La combinaison de ces deux approches, DEVS qui est à la fois un outil de modélisation et de simulation, et l'approche méthodique AGR telle que définie dans AALAADIN et validée par la plateforme MadKit aboutit aussi à offrir un environnement pour la conception et le développement de cadre de modélisation et de simulation très souple et extensible et qui nous permettra d'ajouter le formalisme DEVS aux formalismes de modélisation déjà existants (Réseaux de pétri, UML, diagrammes d'interaction,...).

Nos perspectives portent sur la généralisation de l'approche proposée dans cet article et nous visons à proposer une méthode générale pour générer des transforma-

tions capables de fonctionner sur plusieurs méthodologies de modélisation de systèmes multi-agents et nous envisageons aussi à la mise en œuvre des extensions pour telles transformations sur d'autres environnements et plateformes open source basés sur les SMA.

## References

1. P. A. Fishwick, *Simulation Model Design and Execution. Building Digital Worlds*, Prentice Hall. 1995.
2. M. Sonnessa, *Modelling and simulation of complex systems* (doctoral dissertation, PhD Thesis in "Cultura e impresa", University of Torino, Italy). 2004.
3. R. E. Shannon, *Simulation modeling and methodology*. Proceedings of the 76 bicentennial conference on Winter simulation.9-15.
4. C. Oussalah, *Modèles hiérarchisés multi-vues pour le support de raisonnement dans les domaines techniques*. Technical report. 1998.
5. R. G. Ingalls, *Introduction to simulation*. Proceedings of the 33rd conference on Winter simulation. IEEE Computer Society, 7- 16. 2001.
6. R.E. Shannon, *Introduction to the art and science of simulation*. Proceedings of the 30th conference on Winter simulation. IEEE Computer Society Press, 7-14. 1998.
7. P. A. Fishwick, *Computer simulation: growth through extension*. Transactions of the Society for Computer Simulation International, 14(1), 13–23. 1997.
8. H. Vangheluwe, *Foundations of modelling and simulation of complex systems*. Electronic communication of the EASST, 10: Graph Transformation and Visual Modeling Techniques. <http://eceasst.cs.tuBerlin.de./index.php/eceasst/issue/view/19>. 2008
9. B.P. Zeigler, H. Praehofer, and T.G. Kim, *Theory of Modeling and Simulation*, Second edition, Academic Press, 2000.
10. O. Gutknecht, and J. Ferber, *The MadKit agent plateforme architecture*. Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier, 2000.
11. H. Vangheluwe, *The Discrete Event System specification DEVS Formalism*. Technical report, 2001. <http://moncs.cs.mcgill.ca/>.
12. S. H. Sarjoughian, and B. P. Zeigler, *The role of collaborative DevsModeler in federation development*. In Proceedings of the 99 System Interoperability Workshop, 1999.
13. E. Kofman, N. Giambiasi, and S. Junco, *Fdevs : A general DEVS based Formalism for fault modeling and simulation*. European Simulation Symposium, volume 1 Hamburg (2000), pages 77–82, 2000
14. D. R. Hild, *Discrete Event System Specification Distributed Object Computing Modeling and Simulation*. PhD thesis, 2000.
15. A. Anglani, P. Caricato, A. Grieco, F. Nucci, A. Matta, G. Semeraro, and T. Tolio, *Evaluation of capacity expansion by means of fuzzy-devs*. [citeseer.ist.psu.edu/499458.html](http://citeseer.ist.psu.edu/499458.html), May 2000. 14th European Simulation MultiConference. Gent, Belgium, 2000.
16. J.-B. Filippi, F. Bernardi, and M. Delhom, *The JDEVS environmental modeling and simulation environment*. IEMSS, Integrated Assessment and Decision Support, Lugano Suisse, pages 283–288, 2002.
17. C. Jacques, and G.A. Wainer, *Using the cd++ DEVS toolkit to develop petrinets*. In SCS, editor, Proceedings of the SCS Conference, 2002.
18. A. Hamri, N. Giambiasi, and C. Frydman, *Min-Max DEVS modeling and simulation*. *Simulation Modelling Practice and Theory (SIMPAT)*, 14(7) :909–929, October . Ed. Elsevier, ISSN 1569–190X, 2006.

19. F. Barros, Dynamic structure discrete event system specification : a new Formalism for dynamic structure modelling and simulation. In Proceedings of Winter Simulation Conference, 1995.
20. A. Uhrmacher, Dynamic Structures in Modeling and Simulation : A Reflective Approach. ACM Transactions on Modeling and Computer Simulation vol. 11 2001, pages 206–232, 2001.
21. L. Ntamo, and B. P. Zeigler, Expressing a forest cell model in parallelDEVS and timed cell-DEVS formalisms. Proceedings of the 2004 Summer Computer Simulation Conference, 2004.
22. A. Troccoli, and G. Wainer, Implementing parallel cell-DEVS. In IEEE, Proceedings of the 36th Annual Simulation Symposium, 2003.
23. J. Ferber, Les systèmes multi-agents : vers une intelligence collective. Informatique, intelligence Artificielle, InterEditions, Paris, 1995.
24. J. Ferber, Les Systèmes Multi-Agents : Un Aperçu Général, Revue Technique et Science Informatiques, Hermes-Lavoisier, 1997.
25. F. Michel, J. Ferber, A. Drogoul, : Multi-Agent Systems and Simulation : A survey from the agent's community perspective. Multi-Agent systems : simulation and application edited by A. M. Uhrmacher, D. Weyns– CRC Press- Taylor and Francis Group , pp. 3-52, 2009.
26. J. Ferber, and O. Gutknecht, Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems, ICMAS (International Conference on Multi-Agent Systems), Paris, Y. Demazeau (ed), IEEE Press, pp. 128-135, 1998.
27. Seddari, Noureddine, Mohammed Redjimi, and Sofiane Boukelkoul. "Using of DEVS and MAS tools for modeling and simulation of an industrial steam generator." *Journal of computing and information technology* 22.3 (2014): 171-189.
28. Seddari, Noureddine, and Mohammed Redjimi. "Multi-agent modeling of a complex system." *2013 3rd International Conference on Information Technology and e-Services (ICITeS)*. IEEE, 2013.
29. Seddari, Noureddine, Mohammed Redjimi, and Lazhar Benoudina. "Operational approach for modeling and simulation of an industrial process." *2013 International Conference on Computer Applications Technology (ICCAT)*. IEEE, 2013.
30. Noureddine, Seddari, and Redjimi Mohammed. "Modélisation et Simulation Multi-Agents d'un Processus Industriel (Processus de Génération de Vapeur avec le Poste d'Eau)." ,
31. Noureddine Seddari. Outils formels et opérationnels pour la modélisation et la simulation des systèmes complexes. Modélisation et simulation. Université de Skikda, 2015. Français. (tel-02055182)