

A DEVS/MAS-based Framework for Modeling/Simulation of Complex Systems

Noureddine Seddari^{1,2}, Mohammed Redjimi¹, and Mohamed Belaoued²

¹*Department of Computer Science, 20 August 1955 University of Skikda, Algeria.*

²*LIRE Laboratory, Software Technologies and Information Systems Department, University of Abdelhamid Mehri, Constantine, Algeria.*

¹n.seddari@univ-skikda.dz

¹medredjimi@gmail.com

²belaoued.mohamed@gmail.com

Abstract. This paper presents an approach for modeling and simulation of complex systems. This approach is based on the decomposition of the considered systems into sub-systems which appear on two levels: On the lower level; the decomposition concerns the division of a global system into atomic and coupled models based on DEVS formalism (Discrete Event Systems Specification). The system components are then represented using the DEVS mathematical equations. This step allows the formal system checking. At the higher lever, the implementation of the obtained DEVS models is realized using Multi-Agents Systems (MAS) based on Agent/Role/Group (AGR). Moreover, a set of procedures and functions permitting to transform DEVS to MAS models are defined and implemented. The main advantages of this approach are its adaptability for various domains, its flexibility (easy to implement), its extensibility (adding new components). A version of this work will be implemented using a functional extension of the Multi Agent Development KIT platform (MAD-KIT).

Keywords: Complex Systems, Discrete Event Systems Specification (DEVS), Modeling and Simulation, Multi-Agent Systems (MAS), AGR Model (Agent/Group/Role), MAD-KIT.

1 Introduction

Nowadays, complex systems are made of a large number of interacting entities into interaction, and the overall behavior of these systems emerges from these interactions [1, 2]. The increasing complexity of the studied systems and the requirements of the designers have led to the introduction of many categories of formalisms into the modelling and simulation process. Indeed, there is no single model, or formalism, to model all aspects or components of a system. Thus, different levels of explanation, different theories, and different formalisms or dedicated languages are necessary to properly describe the different aspects or components of the system.

The modeling and simulation [3-10] of complex systems can be used for several purposes. For instance, in ca be used in order to handle the modeled system without risks, to observe and analyze the phenomena emerging when we change the system inputs, and also to improve the understanding and to prevent eventual damages of the system. Therefore, the challenge of the modeling and simulation process of complex

systems is to provide adequate formalism to model these systems on one hand, and to use the appropriate tools for implementation, on the other hand.

This work presents a hybrid method that combines DEVS (Discrete Event System Specification) formalism [11] and AGR model (Agent/Group/Role) for modeling and simulating complex systems.

First, DEVS formalism [11] is used to define the structures of entities that make up the system by using both atomic and coupled models. The DEVS representation encapsulates the structure and the behavior of the model. Thus, DEVS models provide mathematical representations that can be formally checked and validated.

Second, multi-agent systems (MAS) present a strong theoretical and practical tool to implement the solution and provide several advantages such as autonomy, pro-activity, distribution, self-organization and adaptation. The agents cooperate for collective problem solving, since an isolated agent cannot solve the entire problem because it has not large perception and capabilities. Therefore, the system of agents provides a collective solution that emerges from collective actions within the population of the agents.

The idea developed here is focused on the transformation of models checked and validated in DEVS to agent model representation. The Agent/Role/Group (AGR) concept presents the most convenient paradigm in this scope. Moreover, a set of procedures allowing transition between DEVS model and AGR model has been determined and integrated in MAD-KIT platform. The latter has been developed by the team of Professor J. Ferber in LIRMM [12] and based on the principles of AGR.

This paper is organized as follows: The second and third sections present the Multi-Agent Systems and DEVS formalism. The fourth section describes related works, where we categorize combinations of DEVS and MAS from which our approach was inspired. Both approach and algorithms transformations that we propose are shown in the fifth section. Finally, section 6 concludes the work and underlines our perspectives.

2 The Multi-Agent Systems (MAS)

J. Ferber [13] [14] [15] represents a multi-agent system by the couple $\langle A, W \rangle$ where A is an agent and W an environment, and considers an agent in interaction with the world as a system composed of dynamic coupled two subsystems, the coupling taking place through perceptions that the agent has for environment and actions that modify this environment.

$$A = (P_a, \text{Percept}_a, F_a, \text{Infl}_a, S_a) \quad (1)$$

$$W = (E, \Gamma, \Sigma, R) \quad (2)$$

With

- P_a represents the function of perception of the agent,
- Percept_a the set of stimuli and sensations,
- F_a the function of behavior of the agent that determines the agent's state,
- Infl_a the function of action of the agent,
- S_a the set of the agent's internal states,

- E the space in which the agent evolves,
- Γ the space of influences produced by the agent and having like consequences to modify the evolution of the environment,
- Σ the set of states of the environment,
- R the law of evolution of the environment,

2.1 AGR model

The model (AGR) is based on the concepts of agents, groups and roles [12] [16].

Agent. An agent is only specified as an active communicating entity which plays roles within groups.

Group. Groups are defined as atomic sets of agent aggregation representing any usual multi-agent system. Each agent is part of one or more group.

Role. The role is abstract representation of an agent function, service or identification within a group. Each role handled by an agent is local to a group [17-20]. Figure 1 represents the diagram of AGR model.

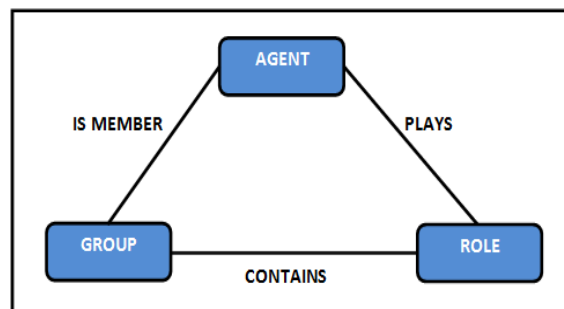


Fig. 1. AGR model

3 The DEVS Formalism

The Discrete Event System Specification (DEVS) [3] [21] is an environment for systems' modeling and simulation. DEVS provides a means of specifying a mathematical object called a system. In addition, DEVS allows gathering, in coherent way, other formalisms of modeling. It is, in fact, a formalism fitted to significant number of scopes of application [22-25].

3.1 Formal Specification of a DEVS Atomic Model

A Discrete Event System Specification is a structure

$$ADEVS = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, t_a, \lambda) \quad (3)$$

Where

X is the set of inputs

S is a set of states

Y is the set of outputs

δ_{int} : $S \rightarrow S$: is the internal transition function

$\delta_{ext}: Q \times X \rightarrow S$: is the external transition function, where

$Q = \{(s,e) \mid s \in S, 0 \leq e \leq t_a(s)\}$ is the total state set

e is the time elapsed since last transition.

$\lambda: S \rightarrow Y$ is the output function

$t_a(s)$ is the time advance function

3.2 Formal Specification of a DEVS Coupled Model

A DEVS coupled model is described by the following equation:

$$CDEVS = (X_{self}, Y_{self}, D, \{M_d / d \in D\}, EIC, EOC, IC) \quad (4)$$

$Self$: is the model itself.

X_{self} is the set of inputs of coupled model.

Y_{self} is the set of outputs of coupled model.

D is the set of names associated with the components of the model, self is not in D .

$\{M_d / d \in D\}$ is the set of components of coupled model.

EIC, EOC and IC define the coupling structure in coupled model.

EIC is the set of external input couplings.

EOC is the external output couplings.

IC defines internal coupling.

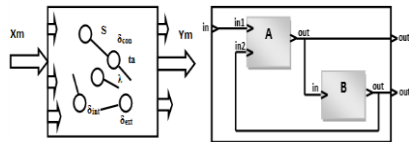


Fig. 2. Graphical representation of an atomic model and a coupled model consisting of two atomic models (A) and (B)

4 Related works

In this section, we present the most relevant tools and platforms for systems modeling and simulation frameworks that combined both DEVS and MAS.

The GALATEA platform [26] [27] provides a high level simulation concepts for specification and design of MAS systems. Thus, one can modeling roles and action plans and establishing communications, negotiations and dialogues between agents to allowing the emergence of efficiency distributed solutions. GALATEA is based on the Zeigler theory of simulation and the logical agents. This platform uses a set of logical programming oriented languages and allows the modeling by using DEVS formalisms. The major disadvantage of this platform is its difficulty of implementation [28].

The JAMES platform [29-31] allows dynamic simulations of MAS systems. An agent is represented by atomic DEVS model, a group of agents is represented by coupled DEVS model and the communication between models is determined by the

inputs and outputs events. The agent environment plays an important role and it is perceived as a shared framework for the agents interactions [32-35].

The VLE platform [36-37] is a high level simulation framework based on DEVS formalism. This platform allows the simulation of the MAS systems by using DEVS. An agent is represented by a DEVS atomic model, an external stimulus by a message and the environment is characterized by CELL-DEVS [38-39].

The DEVSImPy platform [34] considers the advantages and the disadvantages of these latter platforms and provides a framework for agents modeling and simulation by using DEVS formalism. Thus, both the agents and the environments are represented by DEVS atomic models. A group of agents is represented by a DEVS coupled model [40].

By analyzing the aforementioned works, one can conclude that in these approaches; the agents, the environments, the interactions between agents and the groups of agents are represented by atomic and coupled DEVS models.

Multi-agent systems (MAS), for their part, present a strong theoretical and practical tool in the field of simulation. MAS include tools for managing the agents. At this level, the implementation of the system can be easily realized. For these reasons, we opted in our work for the combination

5 Proposed approach

In this work, we propose a new approach for modeling and simulation of complex systems and precisely industrial ones. After studying the most relevant existing approaches we distinguished the following correspondence between DEVS and AGR models :

- A DEVS atomic model can be represented by agent model. The inputs and outputs of the atomic models are represented by a set of stimuli and action of the agent. The internal transition function and the external transition function are represented by the function of behaviour and the function of perception of the agent respectively. The output function represented by the function of action of the agent. The sequences of states (S) of the DEVS atomic model are by the set of the agent's internal states.
- A coupled model materializes a group. The inputs and outputs of coupled model are represented by transmitter coordinator agents. The external input couplings, the external output couplings and the internal coupling are represented by different interconnections between agents and groups of agents.

According to the aforementioned points, we provide three major contributions:

- Industrial systems are classified as discrete event dynamic systems (DEDS) and are characterized by a complex and hierarchical structure [41]. In our approach, the structure of industrial system's elements is formally defined and represented in DEVS, and according to B.P. Zeigler [11] we can prove that this formalism is adequate to model of this type of systems.
- Providing a mechanism which can systematically transform the DEVS to AGR model.
- Integrating this mechanism in MAD-KIT platform in order to create a new extension for DEVS formalism. Choosing MAD-KIT platform for implementation can be justified by its advantages such as, its genericity, its independence to a specific field of application, and finally, its simplicity of implementation.

5.1 Mechanisms of DEVS to AGR transformation

The proposed Algorithm below transforms atomic and coupled model of DEVS model to agent and groups of AGR model [41].

The CDEVS_G (CDEVS) function creates the groups G_i , then, the procedures CDEVS_EIC, CDEVS_IC and CDEVS_EOC define the various interconnections between the agents (AGA $_j$ “agent corresponding to atomic model ADEVS $_j$ ”) created by ADEVS_AGA (ADEVS $_j$) function, AGC $_i$ _R “Agent representing the input ports of coupled model CDEVS $_i$ ” and AGC $_i$ _E “Agent representing the output ports of coupled model CDEVS $_i$ ”) and the groups G_i according to coupling type CDEVS $_i$ as shown below.

The ADEVS_AGA (ADEVS) function allows the creation of AGA $_j$ agents for each atomic model ADEVS $_j$. Thus, both receptors and effectors of AGA $_j$ agent represent respectively the input ports (ADEVS $_j$.Inports $_i$) and output ports (ADEVS $_j$.Outports $_i$) of atomic model. The function of perception AGA $_j$.Pa corresponds to the external transition function ADEVS $_j$. δ_{ext} .

The behavior function AGA $_j$.Fa represents the internal transition function ADEVS $_j$. δ_{int} . The action function of AGA $_j$.Infla agent corresponds to the output function ADEVS $_j$. λ . The dynamic evolution of an agent represents time advance function ADEVS $_j$.ta and the internal states of AGA $_j$ agent. This corresponds to the set of states ADEVS $_j$.S.

The CDEVS_AGC (CDEVS $_j$) function creates two coordinators agents AGC for each coupled model CDEVS $_i$, the first one is a receiver coordinator agent (AGC $_i$ _R) and the second one is a transmitter coordinator agent (AGC $_i$ _E). The main objective of those both coordinators is well to conserve the encapsulation of DEVS formalism. The model is represented as a black box that acts with environment only via the input and output ports. However, the agents have free interaction with their environment. In this approach; the agents want to communicate with internal agents of a group corresponding to a DEVS coupled model should necessary go through coordinators agents of this group.

The CDEVS_G (CDEVS $_i$) function creates groups G_i for each CDEVS $_i$ coupled model; then it creates AGA $_j$ agents that it will gather them in this group G_i . Then; it creates both coordination agents AGC $_i$ _R and AGC $_i$ _E through CDEVS_AGC function. The figure 3 below illustrate a graphical representation of elementary transformation of our approach DEVS/AGR.

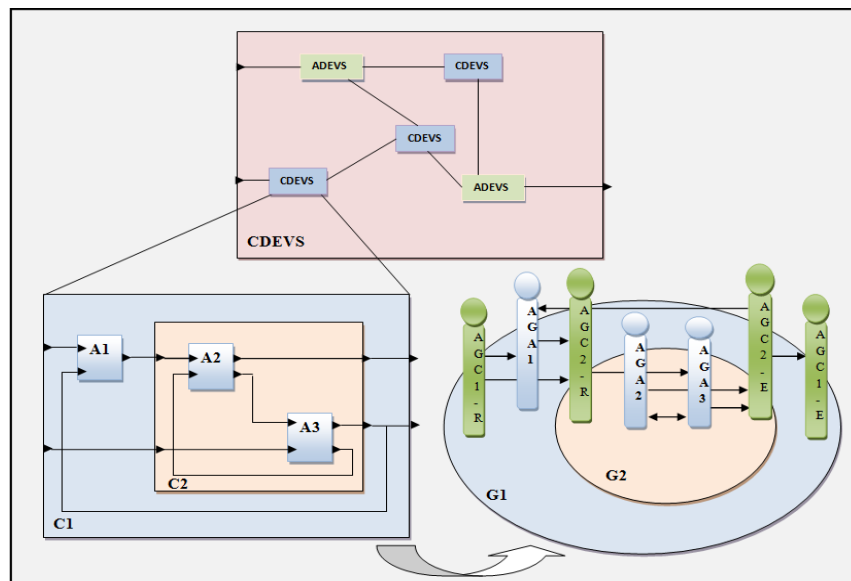


Fig. 3. DEVS/AGR transformation

Algorithm 1: Transformation CDEVS_AGR

Input CDEVS

Output AGR

```
1: for all CDEVS $i$  do
2:   call CDEVS_G (CDEVS $i$ ) // Creation of groups and agents
3:   call CDEVS_EIC (CDEVS $i$ ) // Definition of interconnections corresponding to
   // DEVS $i$ .EIC
4:   call CDEVS_IC (CDEVS $i$ ) // Definition of interconnections corresponding to
   CDEVS $i$ .IC
5:   call CDEVS_EOC (CDEVS $i$ ) // Definition of interconnections corresponding to
   //CDEVS $i$ .EOC
6: end for
```

Function 1: ADEVS_AGA (ADEVS : AtomicDEVS) : AGA

```
1: Create AGA $j$  // Agent model corresponding to ADEVS $j$ 
2: Create a mailbox Mb of size =ADEVS $j$ .Inports.Size
3: Create a sending box Sb of size =ADEVS $j$ .Outports.Size
4: if an external event is appeared in a port : ADEVS $j$ .Inports then
5:   Run AGA $j$ .P $a$  //corresponding to ADEVS $j$ . $\delta_{ext}$ 
6: end if
7: if the life time  $t_a$  of state S is elapsed then
8:   Run AGA $j$ .F $a$  // corresponding to ADEVS $j$ . $\delta_{int}$ 
9:   Run AGA $j$ .Infl $a$  // corresponding to ADEVS $j$ . $\lambda$ 
10: end if
```

Function 2: CDEVS_G (CDEVS : CoupledDEVS) : G

```
1: Create a group  $G_i$  // corresponding to CDEVS $i$ 
2: for all ADEVS $j$  do
3:   call ADEVS_AGA (ADEVS $j$ ) // creation of agents AGA $j$ 
4:   add AGA $j$  within  $G_i$  // regrouping of agents
5: end for
6: call CDEVS_AGC (CDEVS $i$ )
7: add AGC $i$ _E and AGC $i$ _R within  $G_i$ 
```

Function 3: CDEVS_AGC (CDEVS : CoupledDEVS): AGC

```
1: Create AGC $i$ _R // Receiver agent for CDEVS $i$ .Inports
2: Create a mailbox Mb of size = CDEVS $i$ .Inports.Size
3: Create a sending box Sb of size = CDEVS $i$ .EIC.Size
4: if an external event is appeared in a port CDEVS $i$ .Inports then
5:   Run AGC $i$ _R.P $a$ 
6: end if
7: if the mailbox Mb non empty then
8:   Run AGC $i$ _R.F $a$ 
9: end if
10: if the sending box non empty then
11:   Run AGC $i$ _R.Infl $a$ 
12: end if
```

```

13 : Create AGCi_E // emitter agent for CDEVSi.Outports
14 : Create a mailbox Mb of size = CDEVSi.EOC.Size
15 : Create a sending box Sb of size = CDEVSi.Outports.Size
16 : if an external event is appeared in a port : CDEVSi.Outports then
17 :   Run AGCi_E.Pa
18 : end if
19 : if the mailbox Mb non empty then
20 :   Run AGCi_E.Fa
21 : end if
22 : if the sending box non empty then
23 :   Run AGCi_E.Infla
24 : end if

```

Procedure 1: CDEVS_EIC(CDEVS : CoupledDEVS)

```

1 : for  $m = 1$  until the number of CDEVSi.Inports do
2 :   for  $j = 1$  until the number of internal models M do
3 :     for  $k = 1$  until the number of Mj.Inports do
4 :       if (CDEVSi.Mj = ADEVSj) then
5 :         if (CDEVSi.Xm = ADEVSj.Xk) then // EIC between coupled model and
           atomic model
6 :           Define the set of messages between Gi.AGCi_R and Gi.AGAj
7 :         end if
8 :       end if
9 :     if (CDEVSi.Mj = CDEVSj) then
10 :      if (CDEVSi.Xm = CDEVSj.Xk) then // EIC between two coupled models
11 :        Define the set of messages between Gi.AGCi_R and Gj.AGCj_R
12 :      end if
13 :    end if
14 :  end for
15 : end for
16 : end for

```

Procedure 2: CDEVS_IC(CDEVS : CoupledDEVS)

```

1 : for  $j = 1$  until the number of internal models
   M do
2 :   for  $k = 1$  until the number of internal Models M do
3 :     if  $j \neq k$  then // an outports should not be coupled with an inports of the same
       model.
4 :       for  $n = 1$  until the number of Mj.Outports do
5 :         for  $m = 1$  until the number of Mk.Inports do
6 :           if (CDEVSi.Mj = CDEVSj) and (CDEVSi.Mk = CDEVSk) then
7 :             if (CDEVSj.Yn = CDEVSk.Xm) then // IC between two coupled models
8 :               Define the set of messages between Gj.AGCj_E and Gk.AGck_R
9 :             end if
10 :          end if
11 :         if (CDEVSi.Mj = ADEVSj) and (CDEVSi.Mk = ADEVSk) then
12 :           if (ADEVSj.Yn = ADEVSk.Xm) then // IC between two atomic models
13 :             Define the set of messages between Gi.AGAj and Gi.AGAk
14 :           end if

```



```

15 :   end if
16 :   if (CDEVSi.Mj = ADEVsj) and (CDEVSi.Mk = CDEVSk) then
17 :     if (ADEVsj.Yn = CDEVSk.Xm) then // IC between atomic model and
// coupled model
18 :       Define the set of messages between Gi.AGAj and Gk.AGck_R
19 :     end if
20 :   end if
21 :   if (CDEVSi.Mj = CDEVsj) and (CDEVSi.Mk = ADEVSk) then
22 :     if (CDEVsj.Yn = ADEVSk.Xm) then // IC between coupled model and
// atomic model
23 :       Define the set of messages between Gj.AGCj_E and Gi.AGak
24 :     end if
25 :   end if
26 : end for
27 : end for
28 : end if
29 : end for
30 :end for

```

Procedure 3: CDEVs_EOC (CDEVs: CoupledDEVs)

```

1 : for j = 1 until the number of internal models M do
2 :   for p = 1 until the number of Mj.Outputs do
3 :     for q = 1 until the number of CDEVSi.Outputs do
4 :       if (CDEVSi.Mj = ADEVsj) then
5 :         if (ADEVsj.Yp = CDEVSi.Yq) then // EOC between atomic model and
// coupled model
6 :           Define the set of messages between Gi.AGAj and Gi.AGci_E
7 :         end if
8 :       end if
9 :       if (CDEVSi.Mj = CDEVsj) then
10 :        if (CDEVsj.Yp = CDEVSi.Yq) then // EOC between two coupled models
11 :          Define the set of messages between Gj.AGCj_E and Gi.AGci_E
12 :        end if
13 :      end if
14 :    end for
15 :  end for
16 :end for

```

The table 1 below shows a set of passage rules between AGR and DEVS model according to equations (1) and (2) regarding AGR model and equations (3) related to DEVS as well as the algorithms shown above.

Table 1. Correspondence between DEVS and AGR.

DEVS		AGR	
Atomic	X	Agent_a	Percept _a
	Y		AGC_E
	δ_{ext}		P _a
	δ_{int}		F _a
	λ		Infl _a
	S		Agent behaviour (Role)
Coupled	X	Group	AGC_R
	Y		AGC_E
	M		Group / $M \in CDEVs$ Agent / $M \in ADEVs$
	EIC		Interconnection between AGC _i _R & AGC _j _R / $M_j \in CDEVs, (i,j) \in N$ AGC _i _R & AGA _j / $M_j \in ADEVs, (i,j) \in N$
	EOC		Interconnection between AGC _j _E & AGC _i _E / $M_j \in CDEVs, (i,j) \in N$ AGA _j & AGC _i _E / $M_j \in$ ADEVs, $(i,j) \in N$
	IC		Interconnection between AGC _j _E & AGC _k _R / $M_j \in CDEVs, M_k \in CDEVs, (j,k) \in N \& j \neq k$ AGA _j & AGA _k / $M_j \in ADEVs, M_k \in ADEVs,$ $(j,k) \in N \& j \neq k$ AGA _j & AGC _k _R / $M_j \in$ ADEVs, $M_k \in CDEVs, (j,k) \in N \& j \neq k$ AGC _j _E & AGA _k / $M_j \in CDEVs, M_k \in$ ADEVs, $(j,k) \in N \& j \neq k$

5.2 Architecture for Simulation System

The figure 4 presents a global system architecture based on the proposed approach. The interaction module (1) allows the system agent to determinate all coupled and atomic models of system to be simulated through a set of pre-defined models in base of models (2). This base contains a set of re-used models (The system agent can grow it progressively).

The capture of models is done through a set of files code of system simulation. Thus, the system agent may follow the simulation evolutions on the display system (7) by having the possibility to interact with the system thanks to interaction module (1). The parameters are the components of equation of atomic and coupled models. An analysis of data integration is carried out in this level.

The files are saved in XML models (3). Then, the system transforms them thanks to CDEVs/AGR extension (4). The management module of agents (5) allows obtaining a set of agents and groups of agents according to AGR model. Those agents are integrated into MAD-KIT multi-agent platform (8) to obtain the executable code of system simulation.

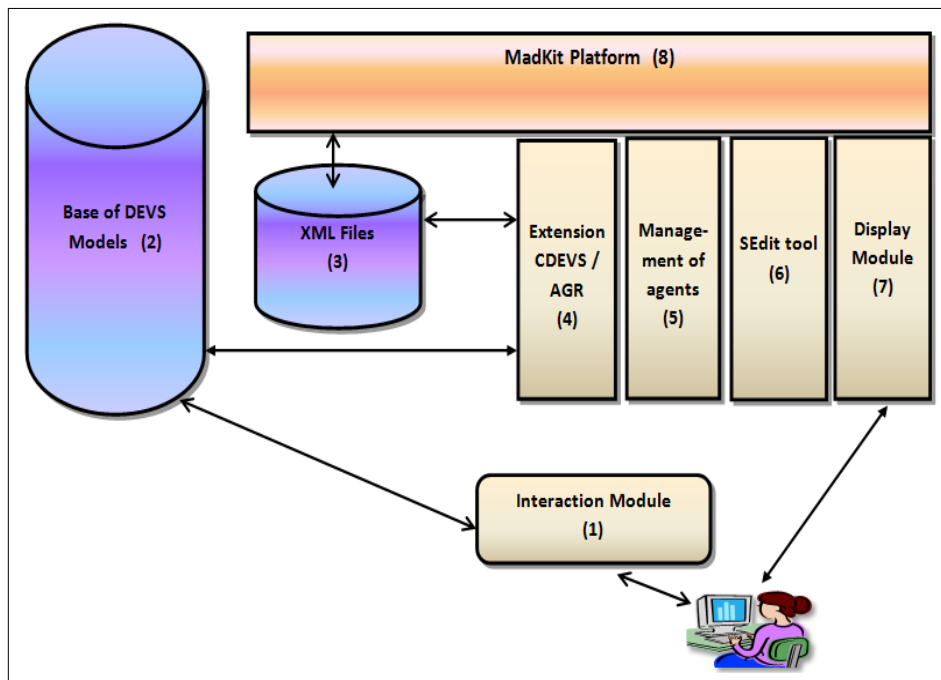


Fig. 4. Overall architecture of the simulation system.

6 Conclusion

In this paper we have proposed a new framework for modeling and simulation of complex systems using multi-agent platform MAD-KIT. The proposed approach is based on the DEVS formalism for system modeling and the AGR of MAD-KIT for their simulation. The main idea behind our approach is to take advantage of the formal power of DEVS formalism for verification and validation and the power of the MAD-KIT platform advanced simulation tools.

The advantage of our approach is that it offers an environment for the design and development of a very flexible and extensible modeling and simulation framework. The later will allow adding DEVS formalism to the modeling formalisms already existing in the platform such as Petri Nets, UML, interaction diagrams, etc. As a perspective, we will focus on testing our extension by modeling and simulating real complex systems such as industrial petroleum systems.

References

1. Simon, H.A. (1969) The sciences of the artificial, *Cambridge (MA). MIT Press.*
2. Bertalanffy, L.V. (1968) Théorie générale des systèmes. *Dunod.*
3. Fishwick, P.A. (1995) Simulation Model Design and Execution. *Building Digital Worlds, Prentice Hall.*
4. Sonnessa, M. (2004) Modelling and simulation of complex systems (*doctoral dissertation, PhD Thesis in "Cultura e impresa", University of Torino, Italy.*)
5. Shannon, R.E. (1976) Simulation modeling and methodology. *Proceedings of the 76 bicentennial conference on Winter simulation.*9-15.
6. Oussalah, C. (1998) Modèles hiérarchisés multi-vues pour le support de raisonnement dans les domaines techniques. Technical report.

7. Ingalls, R.G. (2001) Introduction to simulation. *Proceedings of the 33rd conference on Winter simulation. IEEE Computer Society*, 7- 16.
8. Shannon, R.E. (1998) Introduction to the art and science of simulation. *Proceedings of the 30th conference on Winter simulation. IEEE Computer Society Press*, 7-14.
9. Fishwick, P.A. (1997) Computer simulation: growth through extension. *Transactions of the Society for Computer Simulation International*, 14(1), 13–23.
10. Vangheluwe, H. (2008) 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>.
11. Zeigler, B.P., Praehofer, H. and Kim, T.G. (2000) Theory of Modeling and Simulation, Second edition, *Academic Press*.
12. Gutknecht, O. and Ferber, J. (2000) The MadKit agent plateforme architecture. *Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier*.
13. Ferber, J. (1995) Les systèmes multi-agents : vers une intelligence collective. *Informatique, intelligence Artificielle*, Intereditions, Paris.
14. Ferber, J. (1997) Les Systèmes Multi-Agents : Un Aperçu Général, *Revue Technique et Science Informatiques*, Hermes-Lavoisier.
15. Michel, F., Ferber, J. and Drogoul, A. (2009) 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.
16. Ferber, J. and Gutknecht, O. (1998) 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.
17. Seddari, N., Redjimi, M., & Benoudina, L. (2013, January). Operational approach for modeling and simulation of an industrial process. *In 2013 International Conference on Computer Applications Technology (ICCAT)* (pp. 1-6). IEEE.DOI: 10.1109/ICCAT.2013.6522031.
18. Seddari, N., & Redjimi, M. (2013, March). Multi-agent modeling of a complex system. *In 2013 3rd International Conference on Information Technology and e-Services (ICITeS)* (pp. 1-6). IEEE.
19. Seddari, N., Belaoued, M., & Bougueroua, S. (2017). Agent/Group/Role Organizational Model to Simulate an Industrial Control System. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 11(10), 2376-2385.
20. Seddari, N. and Redjimi, M. (2013). Multi-Agent Modeling of a Complex System . *In DLINe Journal - Transactions on Machine Design (TMD)*, Vol:1 Issue: 2, pp 61-75.
21. Vangheluwe, H. (2001) The Discrete EVent System specification DEVS Formalism. *Technical report*, 2001.<http://moncs.cs.mcgill.ca/>.
22. Barros, F. (1995) Dynamic structure discrete event system specification : a new Formalism for dynamic structure modelling and simulation. *In Proceedings of Winter Simulation Conference*.
23. Uhrmacher, A. (2001) Dynamic Structures in Modeling and Simulation : A Reflective Approach. *ACM Transactions on Modeling and Computer Simulation* vol. 11 2001, pages 206–232.
24. Ntaimo, L. and Zeigler, B.P. (2004) Expressing a forest cell model in parallel DEVS and timed cell-DEVS formalisms. *Proceedings of the 2004 Summer Computer Simulation Conference*.
25. Troccoli, A. and Wainer, G. (2003) Implementing parallel cell-DEVS. *In IEEE, Proceedings of the 36th Annual Simulation Symposium*.
26. Davilla, J. and Uzcategui, M. (2000) GALATEA: A multi_agent simulation platform, *in international conference on modelling, simulation and neural networks MSNN'2000*, Mérida Venezuela.
27. Davilla, J., Gomez, E., Lafaille, K., Tucci, K. and Uzcategui, M. (2005) Multi Agent Distributed Simulation with GALATEA ; *Proceedings of the 2005 Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'05)*.
28. Mattei, S., Bisgambiglia, P.A., Delhom, M. and Vittori, E. (2012) Towards Discrete Event Multi Agent Platform Specification, *Computation Tools 2012 : The third International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking*, pp. 14-21.
29. Uhrmacher, A.M. and Shattenberg, B. (1998) Agents in Discrete Event Simulation, *in Proceedings of ESS98*.

30. Himmelspach, J., Röhl, M. and Uhrmacher, A. M. (2010) Component based modelling and simulation for valid multi-agent system simulations. *International journal for Applied Artificial Intelligence*, 24(5):414–442.
31. Himmelspach, J. (2012) JAMES II: Extending, Using, and Experiments, in: *Proceedings of the 5th Simutools Conference ICST*, pp 208- 210.
32. Weyns, D., Van Dyke Panurak, H., Michel, F., Holvoet, T. and Ferber, J. (2005) Environments for Multi agent Systems State of the art and Research Challenges. *In Environments for Multi-agent Systems, edited by D .weyns, H. Dyke Panurak, and F. Michel , Vol 3374 of lecture Notes in Computer Science, 1-47. Berlin, Heidelberg : Springer Berlin Heidelberg.*
33. Weyns, D., Schumacher, M., Ricci, A., Viroli, M. and Holvoet, T. (2005) Environments in Multi-agent Systems. *The Knowledge Engineering Review*, 20 (02).
34. Wooldridge, M. (2009) An introduction to Multi-agent Systems. 2nded. Chistester, UK Jhon Wiley & Sons.
35. Steiniger, A., Krüger, F. and Uhrmacher, A.M. (2012) Modeling Agents and Their Environment in Multi-level-DEVS, *in Proceedings of the Winter Simulation Conference*.
36. Ramat, E. and Preux, P. (2003) Virtual Laboratory Environment (VLE): a software environment oriented agent and object for modeling and simulation of complex systems, *Simulation and modeling practice and theory, vol 11, N°01* pp. 45-55.
37. Quesnel, G., Duboz, R. and Ramat, E. (2009) The Virtual Laboratory Environment – An operational framework for multi-modelling and analysis of complex dynamical systems, *Simulation and modeling practice and theory, vol 17, N°4* pp. 641- 653.
38. Ilachinski, A. (2001) Cellular Automata, a Discrete Universe, *World Scientific Publishing Co*, ISBN 981-02-4623-4.
39. Wainer, G. A. et Giambiasi, N. (2001) Application of the Cell-DEVS Paradigm for Cell Spaces Modelling and Simulation , *Simulation* , vol. 76, N° 01, p 22-39.
40. Seddari N., Redjimi, M. and Boukelkoul, S. (2014). Using of DEVS and MAS Tools for Modeling and Simulation of an Industrial Steam Generator, *CIT 22*, pp.171-189.
41. Ben-Naoum, L., Boel, R., Bongaerts, L., De Schutter B., Peng, Y., Valckenaers, P., Vandewalle, J. and Wertz V (1995). Methodologies for discrete event dynamic systems: A survey, *Journal A*, 36(4):3–14.
42. Seddari, N. (2015). Outils formels et opérationnels pour la modélisation et la simulation des systèmes complexes (*Doctoral dissertation*, Université de Skikda).