

# FwkDEVS

Simulateur DEVS/GDEVS

Amine HAMRI

Laboratoire des Sciences de l'Information et des Systèmes (LSIS UMR 7296)  
Aix-Marseille University  
France

JDF'16

Les Journées DEVS Francophones : Théorie et Applications /  
Workshop

11-15 avril 2016, Cargèse, France

- Constat
- Objectifs
- Introduction
- Patrons de conception (design patterns)
- FwkDEVS
- Conclusion et Perspectives

- Nombreux simulateurs DEVS développés par des équipes de recherche.  
Problème : documentation technique succincte ou pléthorique.
- Zeigler et al. (2000) proposent d'implémenter les processus du simulateur DEVS (Coordinateur-Racine, Coordinateurs, Simulateurs atomiques) par des Threads.  
Problème : la nécessité d'un ordonnanceur pour lancer, synchroniser et supprimer les objets

- 1 Développer un simulateur basé sur les expériences de programmation proposées par la littérature du paradigme orienté objet.
- 2 Obtenir une documentation technique du simulateur facile à lire et à reproduire.

- Les patrons de conception offre une bibliothèque de solutions squelettes (abstraites) d'objets mises à la disposition des développeurs de logiciels (Gamma et al., 1995).
- Adaptation des patrons de conception à la Modélisation et la Simulation DEVS (Hamri et Baati, SpringSim2010).
- Les développeurs de simulation DEVS et ses extensions doivent documenter et partager leur expérience de développement.

- Une expérience de développement approuvé pour des contextes particuliers par des développeurs différents.
- Un langage de documentation (d'archivage) mise à la disposition des concepteurs et des programmeurs.
- Le paradigme orienté objet est l'essence des patrons de conception.

- La séparation entre les besoins de modélisation et l'architecture de simulation doit être le moteur de développement de toute simulation.
- Les extensions DEVS emploient les mêmes concepts définis en DEVS : Simulateur hiérarchique, modèle, ports, événement, état, fonctions de transitions, etc.

- Un cadre de développement de simulation DEVS/GDEVS destiné aux programmeurs (expérimentés et novices).
- Un noyau de simulation basé sur les patrons de conception (Composite et Observer).
- Un modèle DEVS ou GDEVS est structuré par le patron Composite.
- La structure du simulateur hiérarchique est déduite à partir du modèle à simuler.



Un modèle atomique doit :

- déclarer les variables d'état du modèle.
- déclarer si besoin et instancier les ports d'entrée et de sortie.
- implémenter les fonction de transition  $\delta_{int}()$  et  $\delta_{ext}()$ .
- définir la fonction de sortie  $\lambda()$ .
- préciser les conditions initiales (méthode d'initialisation).

Un modèle Couplé doit :

- déclarer si besoin et instancier les ports d'entrée et de sortie.
- préciser les relations de couplage *EIC*, *EOC* et *IC*.
- implémenter la méthode `setPriority()`.

# Conception d'un modèle

Une classe commune aux classes DEVSCoupled et DEVSAtomique peut être définie afin de factoriser les attributs et méthodes communes.

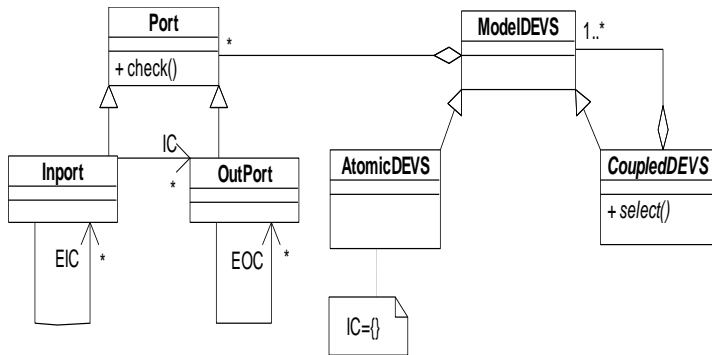


Figure : La structure d'un modèle DEVS à base du patron composite

Un simulateur hiérarchique DEVS doit :

- respecter la structure ordonnée par le modèle à simuler.
- comporter un seul coordinateur racine, des coordinateurs intermédiaires (modèles couplés) et des simulateurs basiques (modèles atomiques).
- échanger des messages descendants et ascendants selon le protocole du simulateur abstrait DEVS.

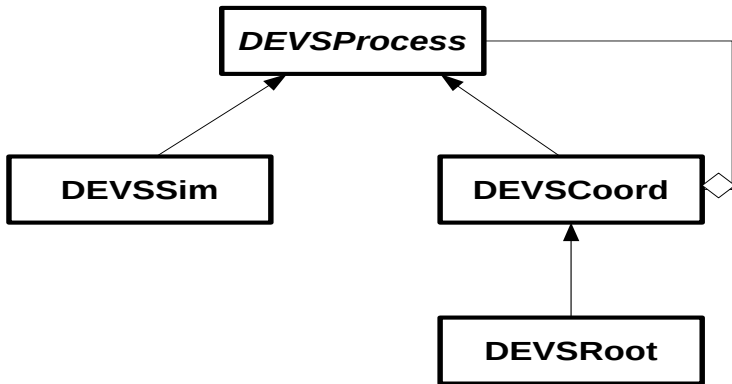


Figure : La structure du simulateur DEVS à base du patron composite

- Refonte de LSIS\_DME.
- GDEVSLogic : Un environnement de M&S de circuits électroniques.

Les expériences de développement de simulation DEVS et ses extensions : les architectures proposées (simulateur DEVS abstrait), séparation de la modélisation de la simulation, etc. peuvent être bénéfiques à d'autres communautés scientifiques (informatique, mathématiques, etc.).

Fédérer les forces et les développements existant au sein du groupe RED pour proposer un framework DEVS extensible à toute extension en favorisant la réutilisation de l'existant et apportant le moins de modification.