



UNIVERSITE DE CORSE-PASCAL PAOLI
ECOLE DOCTORALE ENVIRONNEMENT ET SOCIETE
UMR CNRS 6134 SPE



Thèse présentée pour l'obtention du grade de
DOCTEUR EN INFORMATIQUE

Soutenue publiquement par
EMILIE BROUTIN

le 14 Septembre 2012

Contribution à l'élaboration de concepts et outils pour la modélisation multicouches en vue de la simulation de systèmes hétérogènes

Directeur :

M. Paul Antoine Bisgambiglia, Professeur, Université de Corse

Rapporteurs :

Mme Claudia Frydman, Professeur, Université d'Aix-Marseille III

M. Gabriel Wainer, Professeur, Carleton University

Jury :

Mme Claudia Frydman, Professeur, Université d'Aix-Marseille III

M. Gabriel Wainer, Professeur, Carleton University

M. Jean-Jacques Chabrier, Professeur, Université de Bourgogne

M. Dominique Federici, Professeur, Université de Corse

M. Paul Antoine Bisgambiglia, Professeur, Université de Corse

M. Jean-François Santucci, Professeur, Université de Corse

Résumé

Modéliser un système complexe est une procédure longue et fastidieuse dans laquelle un grand nombre de spécialistes d'origines diverses coopèrent. Il en ressort des modèles très précis représentant de manière quasi parfaite un phénomène. Dans un souci de coût et de temps de développement, il est utile de pouvoir réutiliser des modèles existants. Des problèmes lors de la communication entre les modèles vont se poser.

Le but de cette thèse est la création d'un environnement de modélisation, basé sur le formalisme DEVS mis au point par le Professeur BP Zeigler, permettant de simuler ces modèles sans avoir recours à de lourdes réécritures. L'architecture mise au point permet de coupler des modèles ayant des unités de temps et de données différentes.

La modélisation de systèmes complexes étant trans-disciplinaire, un même phénomène sera vu de plusieurs manières en fonction des spécialistes. Chacun utilisera des unités qui lui sont propres. Or si il est nécessaire de coupler ces deux modèles un problème de précision de données va survenir. Les modèles n'utilisant pas les mêmes unités de données, ils ne pourront communiquer entre eux qu'au prix de lourdes modifications. Il faudra donc définir une personne centrale qui en fonction des besoins devra réécrire une partie des modèles pour les mettre en adéquation. Cette réécriture devra avoir lieu à chaque changement d'unité dans un modèle. On voit donc clairement ici que la connexion ad-hoc de modèles pose problème et nuit à leur réutilisabilité.

De plus, un problème temporel va également apparaître, de la même manière que les unités de données peuvent être différentes, les unités de temps peuvent l'être également.

Pour pallier à ces problèmes, nous avons mis au point une architecture de modélisation qui permet d'interconnecter n'importe quels modèles entre eux. Les problèmes présentés ci-dessus doivent être résolus afin de garantir l'intégrité du résultat de la simulation, en effet si une erreur, même minime, se produit lors d'un échange de données, le résultat final de la simulation peut être faussé de manière importante.

L'architecture proposée se base sur la notion de couche. Nous avons défini deux types de couches :

- Les couches de comportement : Elles définissent la dynamique du système, elles contiennent les modèles mis au point par les différents spécialistes.
- La couche d'agencement : elle est l'élément central de l'architecture de modélisation multicouche. Elle permet aux différentes couches de comportement de communiquer entre elles. Elle leur permet d'échanger et de partager des données.

Un exemple simple a été pris pour tester cette architecture : le bassin versant. L'étude du comportement hydraulique d'un bassin versant implique l'intervention de spécialistes d'horizon distincts. En particulier des spécialistes en hydrogéologie et des physiciens spécialisés en thermodynamique. Ce type d'étude va reposer sur l'élaboration de deux catégories de modèles :

- des modèles d'écoulement
- des modèles de prise en compte des températures pour gérer le phénomène de neige

Cet exemple transdisciplinaire est donc parfait pour tester notre architecture. A partir de modèles existant couplés via l'architecture multicouche, les modèles ont pu échanger des données sans réécriture préalable.

Cet architecture a été intégrée à l'environnement de modélisation DEVSIMPY mis au point au sein du projet TIC. Cet environnement permet de modéliser un phénomène de manière graphique. Il propose une méthode simple de création de modèles à partir des modèles de base induit par le formalisme DEVS.

Abstract

Nowadays, computer modeling is an important tool for studying complex systems. Modeling a complex system is a collaborative work between specialists from various expertise areas which results in the integration of a set of detailed models. These models have to deal with a great number of data. As a result, we obtain several accurate models. Each model is a part of the studied system, consequently, they must be connected to each other to properly model the system. These connexions could raise some problems. During data exchange between models two kind of problems could arise :

- data from a model may not fit to another model
- temporal problems may also arise because of data's time units.

These problems must be solved since they can produce erroneous data that would invalidate the entire simulation.

The main goal of these work is to propose a framework allowing to efficiently connect models with different data level of abstraction and different time units.

We use DEVS (Discrete EVent Specification) formalism created by BP Zeigler. This formalism is used since the 90's for modeling complex systems.

We differentiate two kinds of models :

- Behavioural layers, which describe the dynamic of the systems required in order to solve a given problem i.e. the models defined by the specialists.
- Assembly layer : it is the central point of the proposed framework. Each behavioral layer is linked to the assembly model. All the shared data are processed by this layer in order to performed data exchange between behavioral layer.

The assembly layer contains conversion and temporal components allowing to solve the two previously introduced problems.

The conversion components use a file in which model's creator declare the type and the format of the data they use. They also use a conversion table. These components are generics.

Temporal components allow to use data with different time units. This solution use the classical DEVS simulator. The other solution use graph dependencies in order to determine the model's order of execution. We must redefine the DEVS simulator for this solution.

We choose a simple example for the validation of the architecture : catchement bassin. This example needs the cooperation between several specialists and perfectly fits to our needs.

The multilayered architecture was integrated to the DEVSIMPY environment that has been developped within the TIC project.