L'état interne des composants, grand oublié des formalismes DEVS à structure dynamique ?

Clément Foucher

L'état interne des composants, grand oublié des formalismes DEVS à structure dynamique ?

Clément Foucher
LAAS–CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
7, avenue du Colonel Roche – BP 54200 – 31031 Toulouse cedex 4, France
Clement.Foucher@laas.fr

1 Introduction

De nombreux formalismes étendant DEVS ont été proposés par le passé, pour lui apporter différentes capacités et pour mieux répondre à des problèmes particuliers. En ce qui concerne la possibilité de modifier la structure même du modèle au cours du temps, on note plusieurs propositions dont deux au moins semblent majeures. Que l'on parle de modèle à structure dynamique, à structure variable ou encore réflexif, le cœur de la problématique reste le même : l'architecture interne du modèle n'est pas figée, mais peut évoluer au cours du temps.

Dans ces formalismes toutefois, l'état d'un composant continue à évoluer uniquement en fonction des transitions δ définies par le formalisme DEVS. Pourtant, il pourrait être intéressant d'envisager une évolution de cet état par un mécanisme différent, afin de pouvoir « forcer » l'état d'un composant de manière extérieure, et de prévoir cette possibilité directement au sein du formalisme.

Ce résumé propose de mettre en lumière l'intérêt de la notion de forçage de l'état d'un composant par un mécanisme externe afin de susciter la discussion autour de ce thème lors workshop.

2 DEVS et les formalismes dynamiques

Les modèles DEVS « standards » [3], notamment dans le formalisme DEVS original (Classic DEVS, CDEVS) et son extension parallèle (Parallel DEVS, PDEVS), sont statiques.

La notion d'état est en général portée par les composants atomiques et leur variable s, qui peut évoluer au sein de l'espace de définition de l'état S. On peut toutefois parler de l'état d'un composant couplé, que ce soit par l'agrégation des états de tous les composants atomiques qu'il contient récursivement, ou par la notion d'état du composant DEVS équivalent via la notion de « closure under coupling ».

Au moins deux formalismes principaux ont proposé l'extension de DEVS pour décrire des modèles à structure dynamique : DSDEVS/DSDE et dynDEVS/ρ-DEVS.

DSDEVS, et sa version parallèle DSDE [1], sont les premiers formalismes basés sur DEVS à définir formellement la notion de structure dynamique.

Dans DSDE la notion de structure dynamique s'applique au niveau du couplé, auquel on ajoute une couche d'abstraction en le représentant par un composant χ. χ est un composant spécifique dont le modèle M_χ représente la structure du couplé par le biais de son état. Ainsi, pour chaque état s_χ du composant χ, on peut extraire par une fonction γ appliquée à celui-ci une structure du couplé. Les transitions entre les états de M_χ représentent donc des changements de structure du composant couplé.

Les notions d'état entre les composants des structures successives sont définies comme suit : un composant qui était déjà présent avant un changement de structure conserve son état, tandis qu'un composant qui a été ajouté par un changement de structure prend son état initial.

Dans dynDEVS [2], la notion de changement de structure est étendue aux composants atomiques, en autorisant la modification de sa structure par le biais de la modification de ses variables internes et donc de son espace d'états. On nomme « modèles successifs » le couple formé des modèles d'un composant avant et après son changement de structure. Dans ce formalisme, la valeur des variables communes à deux modèles successifs est préservée, tandis que la valeur initiale par défaut est assignée aux variables introduites par le changement.

Il est également possible de modifier la structure d'un modèle couplé, l'état des composants du modèle suivant alors la même logique que dans DSDE.

3 Pourquoi permettre le forçage de l'état d'un composant ?

Dans ces deux formalismes à structure dynamique, on constate que les auteurs ne s'attardent pas sur la notion d'état lors d'un changement structurel : selon si un composant ou une variable était ou non déjà présent dans le modèle avant le changement structurel, sont état sera préservé ou initialisé . Ce point de vue est tout à fait défendable, car dans DEVS, l'état d'un composant est supposé évoluer par le biais des fonctions δ propres à un modèle.

Néanmoins, dans certains cas, se reposer uniquement sur ces fonctions donne des contraintes fortes au modèle, là où une intégration d'une mécanique dédiée dans le formalisme permettrait plus de flexibilité.

Un exemple typique concerne les systèmes dans lesquels un modèle peut être amené à être supprimé puis remis en place. Par exemple, la notion de time slincing consiste à partager le temps de calcul d'une ressource d'exécution entre plusieurs tâches. Lors de la préemption d'une tâche, son état est sauvegardé, puis on remplace la tâche. Lorsque la tâche initiale disposera à nouveau de temps d'exécution, il faudra alors *restaurer son contexte*, c'est-à-dire la replacer dans son état précédent.

Ces concepts peuvent être intégrés au sein même du modèle. Il faut alors disposer d'un composant de stockage pour conserver l'état passé du composant. Il faut également, pour chaque composant représentant une tâche, disposer de ports dédiés permettant d'en faire sortir son état courant ou de récupérer l'état depuis l'extérieur du composant. Chacun des ces composant doit également prévoir, dans sa fonction δ_{ext}, une transition permettant de réaliser le chargement de l'état ainsi que, dans sa fonction δ_{int} le mécanisme permettant d'émettre l'état complet du composant.

Mais pour un modèle à structure dynamique, les composants sont par nature amenés à être ajoutés, supprimés ou déplacés. La capacité à lire l'état d'un composant ou à forcer son état, par exemple pour recharger à nouveau un état préalablement sauvegardé peuvent ainsi être intéressantes. Dans ce cas, prévoir un mécanisme dédié pour ces opérations sur l'état d'un composant au sein même du formalisme pourrait permettre d'étendre les capacités de modélisation et de simulation de celui-ci.

Références

[1] Fernando J. BARROS. « Modeling formalisms for dynamic structure systems ». In : *ACM Transactions on Modeling and Computer Simulation* 7.4 (oct. 1997), p. 501–515. DOI : 10.1145/268403.268423.

[2] Adelinde M. UHRMACHER. « Dynamic Structures in Modeling and Simulation : A Reflective Approach ». In : *ACM Transactions on Modeling and Computer Simulation* 11.2 (avr. 2001), p. 206–232. DOI : 10.1145/384169.384173.

[3] Bernard P. ZEIGLER, Herbert PRAEHOFER et Tag Gon KIM. *Theory of modeling and simulation*. 2nd. Orlando, FL, USA : Academic press, 2000. ISBN : 0127784551.