

Support pour l'analyse dimensionnelle en SystemC-AMS

Torsten Mähne, Alain Vachoux et Yusuf Leblebici

Laboratoire de Systèmes Microélectroniques (LSM), École Polytechnique Fédérale de Lausanne (EPFL), Suisse

Résumé—La modélisation de systèmes hétérogènes conservatifs multi-domaines (électrique, mécanique, fluide, etc.) requiert un moyen de vérifier que les quantités manipulées par les modèles aient des unités cohérentes. Nous présentons ici les fondations pour le support de l'analyse dimensionnelle statique (effectuée à la compilation) de modèles développés à l'aide des extensions AMS (Analogue and Mixed-Signal) de SystemC. L'approche présentée dans ce papier sera incorporée dans un nouveau modèle de calcul Bond Graph qui complétera les modèles de calcul déjà prévus dans les extensions AMS de SystemC.

Les systèmes sur puces deviennent de plus en plus hétérogènes et complexes en incorporant des composants analogiques/RF, logiques/numériques, logiciels et non électroniques (capteurs, actuateurs). Leurs interactions fonctionnelles doivent être comprises et maîtrisées suffisamment tôt dans le processus de conception, ce qui requiert à la fois la modélisation et la simulation aux niveaux fonctionnel et architectural et la possibilité de raffiner les modèles jusqu'à un niveau permettant la réalisation des composants modélisés.

SystemC [1] offre un tel environnement basé sur C++ pour les systèmes matériels discrets avec logiciel embarqué. SystemC supporte le Modèle de Calcul (MoC) dirigé par les événements (DE – Discrete Event). Pour étendre les capacités de SystemC aux systèmes hétérogènes considérés ici, le groupe de travail AMS (AMSWG) du consortium OSCI (Open SystemC Initiative) développe une bibliothèque de classes C++ additionnelles supportant des modèles de calcul appropriés [2]. Il s'agit des modèles de calculs *Timed Data Flow* (TDF), *Linear Signal Flow* (LSF) et *Electrical Linear Network* (ELN). Ces modèles de calculs interagissent entre eux et avec le MoC de base de SystemC par l'intermédiaire d'une interface de synchronisation. Ce travail est une continuation des efforts du groupe de travail *SystemC-AMS* [3] qui a réalisé un premier prototype de l'environnement de modélisation et de simulation.

Dans le but d'améliorer les capacités de modélisation et de simulation de SystemC-AMS pour les systèmes conservatifs multi-domaines (électrique, mécanique, optique, thermique, etc.) et leurs interactions avec des composants de contrôle digitaux ou logiciels, nous avons commencé le développement d'un nouveau modèle de calcul basé sur les Bond Graphs [4]. Ce formalisme se prête très bien à une modélisation unifiée de composants conservatifs multi-domaines au moyen de graphes décrivant le flot d'énergie entre primitives généralisées. Ces primitives modélisent des sources d'énergie, des comportements résistifs/capacitifs/i-

nertiels, des transformations et des distributions d'énergie au travers de jonctions. Les équations constitutives des primitives et les règles d'interconnexion de celles-ci garantissent la conservation de l'énergie et ainsi la validité de la modélisation de systèmes physiques. Le lien avec le monde physique est maintenu par des spécifications d'unités attachées aux variables et paramètres des modèles. Il faut toutefois noter que ces spécifications d'unités sont aussi applicables à d'autres modèles de calcul que les Bond Graphs.

Le problème est que les simulateurs analogiques ou à temps continu ne considèrent que les valeurs des quantités, usuellement sous la forme de nombres réels flottants. Les spécifications d'unités ne peuvent se faire que sous forme de commentaires éventuellement pris en compte pour l'affichage de résultats de simulation. Ceci ouvre la porte à l'usage de formules ou d'équations incohérentes (p.ex. un facteur de conversion manquant ou la somme de quantités incompatibles) et à l'interconnexion incorrecte de composants multi-domaines. La solution est l'intégration des unités comme partie intégrale du type de donnée et la mise en œuvre de l'analyse dimensionnelle pour vérifier la cohérence des équations et des interconnexions.

Il est possible d'exploiter les capacités de C++ relatives aux patrons (« templates ») pour supporter l'analyse dimensionnelle avec des systèmes d'unités arbitraires au moment de la compilation (analyse statique des types de données) telles que disponibles dans la bibliothèque Boost.Units [5]. Cette bibliothèque utilise des techniques de métaprogrammation basée sur les patrons pour réduire des unités composées arbitrairement complexes en un ensemble ordonné et réduit d'unités de base du système d'unité choisi (p.ex. $[N\ m] = [kg\ m^2\ s^{-2}]$). Un nouvel objet « quantité » `quantity<U, V>` est alors défini comme ayant une unité U et un type de valeur V (qui peut être le type `double` ou un autre type comme `std::complex<T>`). Toute expression incluant des quantités est ainsi vérifiée par le compilateur pour détecter des conversions d'unités illégales. Ceci concerne aussi les constantes qui doivent être annotées avec leurs unités (p.ex. `1.5 * newton * meter`). Les compilateurs C++ modernes peuvent éliminer les informations relatives aux unités de manière à ne provoquer aucune pénalité à l'exécution, au prix de temps de compilations plus élevés. SystemC-AMS facilite l'usage de ces nouveaux types pour des modèles dirigés par les événements ou TDF car les ports et les signaux sont paramétrés sur les types.

Pour démontrer la faisabilité de l'analyse dimensionnelle en SystemC-AMS, une bibliothèque générique de primitives

(additionneur, multiplieur, sources, puits, blocs fonctionnels indépendants et dépendants du temps) pour décrire des diagramme de blocs dans MoC TDF a été réalisée. Chacun des modèles de primitives peut être défini à l’instanciation avec ses types de ports et de paramètres. Les dépendances entre ces types impliquées par les équations constitutives des primitives sont automatiquement satisfaites ou au moins vérifiées. Ces modèles peuvent supporter des types quantités, mais aussi des types simples comme le type double. Ceci assure la compatibilité et l’interopérabilité avec d’autres modèles qui ne supportent pas l’analyse dimensionnelle.

La prochaine étape est d’utiliser le mécanisme d’analyse dimensionnelle présenté ici dans le modèle de calcul Bond Graph en développement pour SystemC-AMS. Les domaines physiques seront implantés sous la forme de classes « traits » contenant les définitions `typedef` requises pour représenter des efforts, des flots, des énergies et des puissances pour les domaines électrique, mécanique, fluide, thermique, etc. Les équations dérivées des primitives Bond Graph et de leur interconnexion seront ainsi automatiquement assurées d’être cohérentes pour la simulation.

RÉFÉRENCES

- [1] *IEEE Standard 1666-2005, SystemC Language Reference Manual*, IEEE, Mar. 2006.
- [2] *Draft Standard SystemC AMS Extensions Language Reference Manual*, OSCI, Dec. 2008.
- [3] A. Vachoux, C. Grimm, and K. Einwich, “Extending SystemC to support mixed discrete-continuous system modeling and simulation,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2005*, May 23–26 2005, pp. 5166–5169.
- [4] T. Maehne, A. Vachoux, and Y. Leblebici, “Development of a bond graph based model of computation for SystemC-AMS,” in *4th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME) 2008*.
- [5] M. C. Schabel and S. Watanabe, *Boost.Units 1.0.0*, 2003-2008. [Online]. Available : <http://www.boost.org/>