Sujet de Thèse de Doctorat

« Génération automatique et certifiée de code embarqué pour les dispositifs médicaux implantables »

D. Andreu (CAMIN[1]), D. Delahaye (MaREL[2])
LIRMM – Université de Montpellier

Contexte

Dans le cadre de l'assistance, voire de la suppléance, aux déficiences fonctionnelles humaines par stimulation électrique fonctionnelle, l'équipe INRIA CAMIN (Control of Artificial Movement & Intuitive Neuroprosthesis) a conçu une nouvelle génération de neuroprothèses. Cette nouvelle génération de dispositifs médicaux implantables est basée sur la décentralisation de la stimulation neurale au plus près des électrodes. Dès lors, un implant embarque l'étage de génération du courant de stimulation ainsi que toute l'intelligence lui permettant d'en contrôler et surveiller l'exécution, en communicant à travers une architecture distribuée. Dans des considérations de conception rigoureuse et de fiabilité de fonctionnement, la méthodologie outillée HILECOP a été mise au point par CAMIN [1]. Elle consiste à concevoir l'architecture numérique complexe et critique d'un dispositif médical implantable à partir d'une approche à composants, dont le comportement et l'assemblage des composants est spécifié à l'aide des réseaux de Petri [2]. Ce formalisme permet de réaliser une analyse formelle (logique et temporelle) du comportement de l'architecture résultante. Une fois l'architecture numérique validée, le logiciel HILECOP permet la génération automatique du code embarqué dans les dispositifs implantables.

L'outil logiciel, et au delà la méthodologie qu'il supporte, est alors crucial dans le processus de création de cette nouvelle génération de neuroprothèses. Cette préoccupation est d'autant plus essentielle que lesdits travaux ont été transférés à la startup INRIA Neurinnov qui vise à produire et commercialiser ces implants innovants ; en effet, l'accès au marché impose de procéder à la certification (marquage CE) des dispositifs et donc nécessite de qualifier l'outil logiciel exploité pour leur création, à savoir le logiciel HILECOP.

Le logiciel HILECOP relève fondamentalement de l'ingénierie dirigée par les modèles : le modèle à analyser ainsi que le code à embarquer dans l'implant sont tous deux issus de transformations de modèles qu'il est nécessaire de qualifier. L'ingénierie dirigée par les modèles est une des thématiques cœur de l'équipe MaREL, qui s'intéresse au développement de méthodes et techniques pour le génie logiciel en général. Plus récemment, l'équipe a acquis des compétences en méthodes formelles et en preuves formelles, que nous souhaitons mettre à profit dans la qualification des transformations de modèles impliquées dans HILECOP.

Sujet

L'objectif de la thèse est de certifier le logiciel HILECOP en amenant une preuve formelle d'équivalence entre la modélisation à base de réseaux de Petri et le code VHDL produit par la transformation mentionnée ci-dessus.

Plusieurs preuves d'équivalence peuvent être considérées dans ce travail. A minima, nous souhaiterions une preuve de préservation structurelle, c'est-à-dire qu'un composant HILECOP est bien traduit vers un composant VHDL avec les mêmes services, et cela pourrait être la

[1] http://www.lirmm.fr/camin/
[2] https://www.lirmm.fr/recherche/equipes/marel

première étape de ce travail de thèse. Ensuite, l'idée est d'aller vers une preuve plus sémantique, afin de démontrer que le comportement du système est préservé par la transformation effectuée par HILECOP.

Cette preuve de préservation sémantique nécessite plusieurs étapes progressives. Il faut tout d'abord formaliser la sémantique des deux langages cibles, à savoir le langage de modélisation de HILECOP (à base de réseaux de Petri) et le langage VHDL. Ensuite, il faut formaliser la transformation entre les deux langages. Enfin, il faut démontrer que cette fonction de transformation préserve la sémantique des deux langages, c'est-à-dire qu'un programme HILECOP (modélisé par un réseau de Petri) s'exécute de la même manière que le programme VHDL traduit par HILECOP.

L'originalité (et aussi la difficulté) d'une telle preuve de préservation sémantique réside essentiellement dans les langages impliqués. Dans le domaine des preuves formelles, les travaux sur la sémantique des langages sont nombreux, mais concernent essentiellement les langages séquentiels (voir par exemple la certification d'un compilateur C en Coq [5]). Ici, dans notre cas, d'autres aspects apparaissent, comme la concurrence et le temps, et qui vont radicalement changer notre façon de formaliser ces sémantiques.

Une fois la preuve de préservation sémantique réalisée sur papier, nous souhaitons mécaniser cette preuve en utilisant l'outil d'aide à la preuve Coq [3]. Cette mécanisation nous permettra de vérifier que la preuve réalisée sur papier est bien correcte. Par ailleurs, en Coq, le moyen idiomatique de formaliser les sémantiques (et même de manière générale) est d'utiliser des types inductifs, et il existe des travaux [4] qui permettent d'extraire des comportements calculatoires de ces types inductifs. Ainsi, au niveau modélisation de HILECOP, il serait possible d'extraire un animateur de réseaux de Petri, qui serait certifié, c'est-à-dire conforme à la sémantique formalisée. De même, Coq possède un mécanisme d'extraction, qui nous permettrait d'extraire la fonction de transformation vers VHDL, qui serait également certifiée correcte et qui pourrait potentiellement être intégrée au développement du logiciel HILECOP.

Compétences requises
Le (la) candidat(e) doit être titulaire d'un master ou d'un diplôme d'ingénieur en Informatique. Rigueur, curiosité intellectuelle, capacité d'abstraction, esprit logique et capacité de travail en équipe sont des prérequis. De plus, le (la) candidat(e) doit maîtriser les langages de modélisation tels que UML, Ecore, etc. La connaissance d'approches et/ou d'outils de preuves formelles, tels que Coq, Atelier B, etc. serait un plus.

Contacts **:** David.Andreu@lirmm.fr, David.Delahaye@lirmm.fr

Références :
[1] H. Leroux, D. Andreu, K. Godary-Dejean. *Petri nets based digital architecture: from formalism to implementation on FPGAs*. IEEE Transactions on Industrial Informatics, Vol. 11, N. 4, 2015.
[2] G.W. Brams. *Réseaux de Petri : Théorie et Pratique.* Masson, 1983. ISBN 2-903607-12-5.
[3] The Coq Development Team. *Coq, version 8.6.* Inria, Oct. 2016. http://coq.inria.fr/.
[4] P.-N. Tollitte, D. Delahaye, and C. Dubois. *Producing Certified Functional Code from Inductive Specifications.* In Certified Programs and Proofs (CPP), volume 7679 of LNCS, pages 76-91, December 2012. Springer.
[5] X. Leroy. *A Formally Verified Compiler Back-End.* Journal of Automated Reasoning, 43(4) : 363-446, 2009.