

VERGER/Aide à la Validation tEmpoRelle des logiciels critiques GERICOS de l'observatoire de Paris dans un contexte d'architectures multicoeurs

Direction/Encadrement
Philippe Plasson, LESIA
Gabriel Brusq, CNES
Stéphane Rubini, Lab-STICC
Nam Tran Hai, Lab-STICC
Frank Singhoff, Lab-STICC

Laboratoires d'accueil : LESIA UMR CNRS 8109, Lab-STICC UMR CNRS 6285, CNES

Mots clés : Spatial, systèmes temps réel critique, analyses des contraintes temporelles, multi-coeurs, ressources matérielles partagées, Ingénierie dirigée par les modèles

1. Contexte

Depuis plusieurs années, le LESIA de l'Observatoire de Paris développe la plateforme logicielle GERICOS [1]. La plateforme GERICOS est un intergiciel dédié au développement de logiciels embarqués pour les systèmes spatiaux. Il a, par exemple, été utilisé pour la mise en œuvre de la charge utile du satellite PLATO, conjointement avec le système d'exploitation RTEMS (version 4.8 produite par Edisoft).

GERICOS propose au développeur logiciel un ensemble de composants réutilisables et organisés en couche permettant de faciliter le développement de logiciels critiques temps réel.

La plateforme est par ailleurs utilisée dans le cadre d'une approche d'ingénierie dirigée par les modèles afin d'en automatiser la production du logiciel.

Depuis 2002, le Lab-STICC développe des méthodes et des outils permettant d'assurer la vérification des contraintes temporelles de systèmes temps réel critiques [11] en phase amont de la réalisation du système, c-à-d lors de la conception de l'architecture logicielle. Il a été démontré que cette phase amont de conception d'architecture logicielle est un moment clef pour garantir la qualité du système réalisé tout en maîtrisant son coût de production. Ces méthodes et outils d'analyse amont des contraintes temporelles s'articulent autour du langage AADL et d'une démarche dirigée par les modèles. En collaboration avec l'entreprise Ellidiss Technologies, ces méthodes et outils sont intégrés dans différentes plateformes tant open-source que commerciales, telles que le produit AADL Inspector de la société Ellidiss ou la plateforme open-source TASTE de l'ESA. Dans [6], nous avons montré comment ces méthodes et outils peuvent être appliquées sur des engins spatiaux, et en particulier sur un satellite [10].

Compte tenu des applications ciblées, le LESIA est aujourd'hui confronté au besoin de produire automatiquement ces vérifications du respect des contraintes temporelles des logiciels construits avec GERICOS. Or aujourd'hui, le processus d'ingénierie de GERICOS ne comporte pas de validation automatique de ces contraintes, ce qui peut être une tâche difficile à atteindre au vu des spécificités des composants GERICOS. Cet objectif est d'autant plus difficile que les charges utiles développées par les acteurs du domaine requièrent de plus en plus fréquemment l'utilisation d'architectures multi-coeurs.

2. Problème traité et verrous étudiés

L'objectif de la thèse consiste donc à mettre en commun les compétences des deux équipes (LESIA et Lab-STICC) afin d'accroître les moyens de vérification de la plateforme GERICOS.

Automatiser la vérification des contraintes temporelles imposées aux applications GERICOS dans le cadre d'architectures multi-coeurs posent plusieurs verrous. La thèse propose d'en étudier 4, qui, en cas de succès, permettront de mettre en place dans GERICOS des premiers éléments de réponse concrets au problème traité. Les 4 verrous sont les suivants:

1. **Compatibilité de la sémantique des composants GERICOS avec les méthodes d'analyse de la littérature [12].** Il s'agira ici d'analyser la sémantique des composants GERICOS et d'identifier les spécificités qui les rendent compatibles ou non avec les méthodes d'analyse des contraintes temporelles de la littérature. A titre d'exemple, GERICOS propose le concept d'objet actif. Un objet actif comporte un thread d'exécution pouvant exécuter plusieurs fonctions déclenchées par des messages. Ce concept n'est pas directement transposable à des fins de vérification.
2. **Extension des méthodes d'analyse de la littérature afin de les rendre applicables aux architectures logicielles conformes à GERICOS.** Dans ce verrou, nous tenterons de proposer de nouvelles méthodes d'analyses dans le cadre d'un environnement multiprocesseur.
3. **Intégration dans un processus de conception des logiciels spatiaux basé sur le langage GERICOS.** Dans ce verrou, il s'agira de proposer aux utilisateurs de la

plateforme GERICOS une méthode de conception des architectures du logiciel exprimées avec le langage GERICOS en conservant la génération du code et de documentation actuellement offerte par cette plateforme, mais en offrant également des moyens des vérifications en amont des propriétés temps réel. Ce travail doit conduire également à élaborer des règles de conception de l'architecture, voire à une évolution de GERICOS, permettant de faciliter l'activité de vérification. Ces règles de conception devraient faire évoluer les pratiques de conception des architectures actuellement utilisées par les acteurs du domaine afin de rendre celles-ci plus robustes et analysables.

4. **Prise en compte des ressources matérielles partagées entre les cœurs du processeur.** Une des originalités de GERICOS est la façon dont est prise en compte la variabilité des temps d'exécution dues aux ressources matérielles partagées des architectures multi-cœurs. GERICOS utilise une méthode basée sur la mesure en cours d'exécution de ces délais. Dans ce verrou, nous évaluerons les forces et faiblesses de cette méthode au regard de la littérature et nous regarderons comment intégrer l'approche GERICOS et les méthodes de la littérature en phase amont pour l'analyse des contraintes temporelles.

Les travaux sur ces 4 verrous donneront lieu à des expérimentations permettant de vérifier la pertinence des propositions, mais aussi d'apporter au LESIA un prototype d'outil d'analyse applicable aux applications GERICOS. Une attention particulière sera portée à l'automatisation de ces vérifications. Par ailleurs, un démonstrateur à destination de la communauté scientifique (éventuellement sous la forme d'un benchmark) sera réalisé.

3. Déroulement de la thèse et résultats attendus

Le déroulement de la thèse est prévu en 5 phases (démarrage de la thèse au mois M0). Nous donnons dans le tableau ci-dessous une brève description de chaque phase et les résultats attendus.

Phases, durée	Description de la tâche	Résultats attendus
Phase 1 M0 à M0+9	<p>Analyse de la sémantique des composants GERICOS et proposition de méthodes d'analyse des contraintes temporelles conforme à ces composants.</p> <p>Dans cette phase, on suppose un environnement d'exécution multi-cœurs, type d'environnement supporté par la plateforme GERICOS actuelle.</p>	<p>Modèle de simulation temporelle</p> <p>Modèle mathématique permettant de vérifier d'ordonnancement de tâches s'exécutant avec l'intergiciel GERICOS dans un contexte AMP (asymmetric multiprocessing).</p>

<p>Phase 2 M0+9 à M0+12</p>	<p>Intégration logicielle des modèles construits lors de la phase 1 dans les outils GERICOS (LESIA) et Cheddar (Lab-STICC).</p>	<p>Extension du profil UML de GERICOS et outils du LESIA mis à jour pour la modélisation des données nécessaires à ces méthodes d'analyse.</p> <p>Nouvelle version de Cheddar qui implante les méthodes d'analyse élaborée en phase 1.</p> <p>Diffusion des résultats vers la communauté sous la forme de publications, de logiciel libre (Cheddar) et d'un benchmark illustrant les applications ciblées.</p>
<p>Phase 3, M0+12 à M0+18</p>	<p>Proposition d'une méthode de conception permettant, à partir d'un modèle exprimé avec le langage GERICOS d'analyser automatiquement les propriétés temps réel du systèmes tout en maintenant les services de génération de code et de documentation actuellement offerts par GERICOS.</p> <p>Cette phase doit également permettre d'élaborer des règles de conception des architectures garantissant l'application des méthodes de vérification élaborées lors des phases 1 et 2.</p>	<p>Spécification de la méthode de conception et des règles de modélisation de l'architecture.</p>
<p>Phase 4, M0+18 à M0+30</p>	<p>Expérimentations et validation des propositions des phases 1 à 3 avec des applications du LESIA.</p> <p>Mesures sur matériels réels du LESIA et du Lab-STICC (cartes « durcies » LEON pour une utilisation dans le domaine spatial).</p>	<p>Modèles des applications utilisées pour la validation.</p> <p>Résultats des expérimentations.</p> <p>Publications.</p>

Phase 5, M0+30 à M0+36	Rédaction du mémoire de thèse.	Mémoire de thèse.
------------------------	--------------------------------	-------------------

4. Références

[1] GERICOS: A Generic Framework for the Development of On-Board Software. Philippe Plasson, Christopher Cuomo, Gary Gabriel, Nicolas Gauthier, Loïc Gueguen, Lee-Roy Malac-Allain. In proceedings of DASIA Conference, ESA-SP Vol. 736, May 2016.

[2] TASTE: An open-source tool-chain for embedded system and software development. Maxime Perrotin, et al. Embedded Real Time Software and Systems (ERTS2012). 2012.

[3] PLATO N-DPU on board software: an ideal candidate for multicore scheduling analysis. Philippe Plasson, Gabriel Brusq, Frank Singhoff, Hai Nam Tran, Stéphane Rubini, Pierre Dissaux. In Embedded Real Time Software and Systems (ERTS 2022). 2022, Toulouse.

[4] ECTM: A Network-on-Chip Communication Model to Combine Task and Message Schedulability Analysis. Mourad Dridi, Frank Singhoff, Stéphane Rubini, Jean-Philippe Diguët. Journal of Systems Architecture, Volume 114, March 2021.

[5] Design and Multi-Abstraction Level Evaluation of a NoC Router for Mixed-Criticality Real-Time Systems. Mourad Dridi, Stéphane Rubini, Mounir Lallali, Martha Johanna Sepulveda Florez, Frank Singhoff, Jean-Philippe Diguët. ACM Journal on Emerging Technologies in Computing (JETC), Volume 15 Issue 1, February 2019. Article No. 2

[6] Timing analysis of TASTE models for reconfigurable software. Jean-Charles Roger, Pierre Dissaux, Jérôme Legrand, Mourad Dridi, Stéphane Rubini, Frank Singhoff. Model-Based Systems and Software Engineering (MBSE 2021) ESA Workshop 29 - 30 September 2021.

[7] ECTM: A New Communication Model to Network-On-Chip Schedulability Analysis. Mourad Dridi, Frank Singhoff, Stéphane Rubini and Jean-Philippe Diguët 24th International Conference on Reliable Software Technologies – Ada-Europe 2019, Warsaw, Poland.

[8] DAS: An Efficient NoC Router for Mixed-Criticality Real-Time Systems. Mourad Dridi, Stéphane Rubini, Mounir Lallali, Martha Johanna Sepulveda Florez, Frank Singhoff, Jean-Philippe Diguët. Proceedings of the 2017 IEEE International Conference on Computer Design (ICCD), Boston, MA, 2017, pp. 229-232. doi: 10.1109/ICCD.2017.42

[9] Modeling and Validation of a Mixed-Criticality NoC Router Using the IF Language. Mourad Dridi, Mounir Lallali, Stéphane Rubini, Frank Singhoff, Jean-Philippe Diguët. Proceedings of the 10th International Workshop on Network on Chip Architectures. NoCArc'17. Boston, MA, USA, October, 2017

[10] MOSAR: Modular spacecraft assembly and reconfiguration demonstrator. Letier, Pierre, et al. 15th Symposium on Advanced Space Technologies in Robotics and Automation. 2019.

[11] Cheddar: a flexible real time scheduling framework. Singhoff Frank, Legrand, J., Nana, L., & Marcé, L. Proceedings of the 2004 annual ACM SIGAda international conference on Ada. November 2004. Atlanta, USA.

[12] A survey of hard real-time scheduling for multiprocessor systems. Davis Robert I., Alan Burns. ACM computing surveys (CSUR) 43.4 (2011): 1-44.

[13] The Solar Orbiter Radio and Plasma Waves (RPW) instrument. M. Maksimovic, S. D. Bale, Thomas Chust, Y. Khotyaintsev, V. Krasnoselskikh, M. Kretzschmar, D. Plettemeier, H. O. Rucker, J. Souček, M. Steller, et al. Astronomy and Astrophysics - A&A, EDP Sciences, 2020, 642, pp. A12. September 2020