

Master 2 Projet

Quelle logique pour spécifier des interactions ? Application aux systèmes IoT

Simon Bliudze (simon.bliudze@inria.fr)

Sophie Cerf (sophie.cerf@inria.fr)

Olga Kouchnarenko (olga.kouchnarenko@univ-fcomte.fr)

Contexte

Nous travaillons sur un formalisme [1] pour modéliser et contrôler des systèmes auto-adaptatifs. Plus précisément, on considère des systèmes composés d'une multitude d'agents, de petite taille et de nature similaire, qui collaborent dans un but global commun, comme par exemple dans le contexte de l'*Internet-of-Things* (IoT). Les [robots modulaires \(https://projects.femto-st.fr/programmable-matter/\)](https://projects.femto-st.fr/programmable-matter/) constituent un exemple de tels systèmes.

Nous visons un formalisme qui soutient une modélisation hiérarchique de tels systèmes et intègre naturellement le contrôle adaptatif pour optimiser des aspects physiques de systèmes comme, par exemple, la consommation d'énergie.

En effet, la coordination et le contrôle doivent reposer sur des mesures agrégées, telles que le pourcentage d'éléments dans un état donné. Par exemple, on s'intéresse à des exigences comme la suivante : *si moins de 75 % de robots modulaires ont assez d'énergie pour émettre la lumière pendant 5 minutes, alors 20% de ces robots doivent éteindre leurs haut-parleurs* (permettant alors à ces robots de continuer d'émettre la lumière plus longtemps). Or, le calcul efficace de ces mesures dépend des interactions entre des éléments/niveaux hiérarchiques. Pouvoir spécifier des exigences (mesures agrégées) et des interactions dans un formalisme approprié devient alors un besoin d'une grande importance.

Objectifs du projet

L'objectif principal de ce projet est une étude bibliographique de différentes logiques pour la spécification de mesures agrégées et d'interactions.

Après une brève introduction du contexte applicatif IoT et robots modulaires en particulier, l'étudiant(e) devra effectuer les tâches suivantes :

1. Identifier des mesures agrégées et des exigences fonctionnelles et non-fonctionnelles (cf. [2,3] par exemple) pour un/des cas d'études IoT.
2. Etudier des méthodes pour les spécifier formellement (cf. [4,5,6] par exemple).
3. Etudier des algorithmes pour les calculer efficacement.
4. Sur cette base, il s'agira ensuite de faire une comparaison de méthodes sur un cas d'étude (de robots modulaires par exemple) pour des mesures/exigences identifiées.

En cas de réussite, ce projet sera poursuivi d'un stage recherche au sein du laboratoire FEMTO-ST, avec des interactions avec une équipe Inria/Lille.

Références

- [1] S. Bliudze, S. Cerf, and O. Kouchnarenko, "Controlling Hierarchical Motifs to Avoid Systems' Blackout" in preparation.
- [2] J. Beal, D. Pianini and M. Viroli, "Aggregate Programming for the Internet of Things," in *Computer*, vol. 48, no. 9, pp. 22-30, Sept. 2015, doi: [10.1109/MC.2015.261](https://doi.org/10.1109/MC.2015.261) (<https://doi.org/10.1109/MC.2015.261>). [PDF (<https://jakebeal.github.io/Publications/Computer-AggregateProgramming-2015.pdf>)]
- [3] R. Scattolini, "Architectures for distributed and hierarchical model predictive control --- a review", in *Journal of process control*, 19(5), 723-731, 2009. [PDF (<https://api.panist.fr/document/2971B5F39F8BF90233A7D1E179F23572A28ADE4D/fulltext/pdf?sid=clickandread>)]
- [4] O. Maler, D. Nickovic, "Monitoring temporal properties of continuous signals", in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. pp. 152–166. Springer Berlin Heidelberg (2004)[PDF (<https://www-verimag.imag.fr/~maler/Papers/monitor.pdf>)]
- [5] V. Lopez, J. Montero, and J. Tinguaro Rodríguez, "Formal Specification and implementation of computational aggregation Functions", in *Computational Intelligence: Foundations and Applications*, pp. 523-528. 2010. doi: [10.1142/9789814324700_0078](https://doi.org/10.1142/9789814324700_0078) (https://doi.org/10.1142/9789814324700_0078)
- [6] K. Sen, A. Vardhan, G. Agha, and G. Rosu. "Efficient decentralized monitoring of safety in distributed systems." In *Proceedings. 26th International Conference on Software Engineering*, pp. 418-427. IEEE, 2004. doi: [10.1109/ICSE.2004.1317464](https://doi.org/10.1109/ICSE.2004.1317464) (<https://doi.org/10.1109/ICSE.2004.1317464>) [PDF (<http://formalmethods.web.engr.illinois.edu/papers/SVAR-distmon.pdf>)]