Collaboration de méthodes formelles pour l'analyse d'architectures applicatives réparties

Frédéric Loulergue Yohan Boichut Frédéric Dabrowski

1er semestre 2021

Introduction

FRAMA-C [5] est un framework pour l'analyse de programmes C. L'une de ses forces est d'avoir un ensemble de greffons qui couvrent un large spectre de méthodes formelles : analyse statique, vérification déductive, vérification dynamique, tests. Pour l'analyse de logiciels de grande taille, ces différents greffons sont utilisés de façon combinée pour obtenir des garanties très fortes sur les parties les plus critiques des systèmes tout en permettant d'avoir des garanties plus faibles mais plus faciles à obtenir sur les autres parties.

Les architectures applicatives réparties sont au cœur de l'activité de nombreuses entreprises, les frameworks les plus déployés étant basé sur l'écosystème JAVA. Il existe des travaux académiques sur la spécification [4] et la vérification déductive de programmes JAVA [3, 6, 1], ainsi que sur l'analyse statique [8], mais des logiciels aussi complexes ne sont pas considérés. En pratique bien sûr, le test est utilisé couramment, mais les techniques les plus pointues [7] ne sont pas toujours connues ou appliquées à ce type de systèmes. La situation commence un peu à changer avec des travaux tels que ceux de [2]. Si divers outils existent, il n'y a pas de framework intégré comme FRAMA-C.

Le stage

L'objectif de ce stage est de choisir une architecture applicative répartie réaliste mais de taille raisonnable, et d'en effectuer l'analyse par une combinaison d'outils récents de vérification déductive, vérification dynamique, et analyse statique de programme JAVA. Cette étude de cas devra permettre d'établir un état de l'art dans l'application de méthodes formelles sur les programmes JAVA et de dégager des pistes de recherche pour l'amélioration et l'intégration des outils existants, ou des besoins pour des outils futurs à concevoir.

Information pratiques

- Lieu : Laboratoire d'Informatique Fondamentale d'Orléans, site d'Orléans
- Durée : 6 mois à partir du mois de Janvier
- Indemnités : montant légal (3,90€ / heure), temps plein

- Contact :
  - Frederic.Loulergue@univ-orleans.fr
  - Yohan.Boichut@univ-orleans.fr
  - Frederic.Dabrowski@univ-orleans.fr

References

[1] Wolfgang Ahrendt, Bernhard Beckert, Richard Bubel, Reiner Hähnle, Peter H. Schmitt, and Mattias Ulbrich, editors. *Deductive Software Verification - The KeY Book - From Theory to Practice*, volume 10001 of *Lecture Notes in Computer Science*. Springer, 2016. ISBN 978-3-319-49811-9. doi:10.1007/978-3-319-49812-6.

[2] Anastasios Antoniadis, Nikos Filippakis, Paddy Krishnan, Raghavendra Ramesh, Nicholas Allen, and Yannis Smaragdakis. Static analysis of java enterprise applications: frameworks and caches, the elephants in the room. In Alastair F. Donaldson and Emina Torlak, editors, *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, pages 794–807. ACM, 2020. doi:10.1145/3385412.3386026.

[3] Jean-Christophe Filliâtre and Claude Marché. The Why/Krakatoa/Caduceus Platform for Deductive Program Verification. In W. Damm and H. Hermanns, editors, *19th International Conference on Computer Aided Verification*, LNCS. Springer, 2007.

[4] Christoph Gladisch and Shmuel Tyszberowicz. Specifying linked data structures in JML for combining formal verification and testing. *Science of Computer Programming*, 107-108:19 – 40, 2015. doi:10.1016/j.scico.2015.02.005.

[5] Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. Frama-C: A software analysis perspective. *Formal Asp. Comput.*, 27(3): 573–609, 2015. doi:10.1007/s00165-014-0326-7. URL `http://frama-c.com`.

[6] Claude Marché, Christine Paulin-Mohring, and Xavier Urbain. The KRAKATOA tool for certification of Java/JavaCard programs annotated in JML. *J. Log. Algebr. Program.*, 58 (1-2):89–106, 2004. doi:10.1016/j.jlap.2003.07.006.

[7] Long H. Pham, Quang Loc Le, Quoc-Sang Phan, Jun Sun, and Shengchao Qin. Testing heap-based programs with Java StarFinder. In *Proceedings of the 40th International Conference on Software Engineering (ICSE)*, page 268–269. ACM, 2018. doi:10.1145/3183440.3194964.

[8] Xavier Rival and Yi Kwangkeun. *Introduction to Static Analysis*. MIT Press, 2020.