

Génération automatique de tests pour la couverture fonctionnelle de logiciels

Stage Master 2, 2018–2019

Thème : Logique du premier ordre.

Laboratoire : U2IS, ENSTA ParisTech

Adresse : 828 boulevard des maréchaux 91762 Palaiseau Cedex

Encadrants : Alexandre Chapoutot et Laurent Raffaelli

alexandre.chapoutot@ensta-paristech.fr

laurent.raffaelli@naval-group.com

Durée : 6 mois

Rémunération : suivant la législation en vigueur.

Contexte. Les logiciels dont les dysfonctionnements conduisent à des pertes humaines ou financières importants sont qualifiés de critiques. Ces logiciels apparaissent dans tous les secteurs d'activités actuels : dans le secteur des transports dans les systèmes d'assistance à la conduite ou dans la conduite autonome ; dans le secteur de l'énergie avec la régulation automatique des réseaux de distribution énergétiques ; ou encore dans le secteur de la défense. Dans tous ces domaines, les concepteurs de ces systèmes doivent suivre les règles de l'art dans le développement logiciel et mettre en œuvre des procédures de vérification démontrant que le logiciel fait bien ce qu'il doit faire.

L'activité de tests logiciels est une composante essentielle et indispensable dans la démonstration de sûreté des logiciels critiques. Cette activité s'appuie sur les spécifications logicielles pour montrer, grâce à des jeux de tests bien choisis, que le logiciel développé respecte, ou au moins n'enfreint pas, les spécifications. L'objectif est de montrer que le code fait ce qu'il doit faire et pas plus. La principale difficulté est de mesurer au travers des différents jeux de tests que les spécifications sont couvertes complètement. En général, des mesures sur les constructions du code source (décisions, branches, etc.), on parle alors de couverture structurelle, sont mises en œuvre pour décider si tous les cas fonctionnels sont couverts par les tests.

La difficulté de la mesure de la couverture fonctionnelle tire son origine dans la façon dont sont rédigés les spécifications, qui est en général en langue naturelle. Ce choix conduit à des spécifications qui sont ambiguës. Ces ambiguïtés se traduisent par des choix d'implantation qui ne correspondent pas aux fonctionnalités qui devraient être codées.

Travail à réaliser. Le but du stage est de définir une méthodologie de couverture de tests fonctionnels de logiciel par génération automatique de jeux de tests. Pour cela, il faudra définir une méthodologie de rédaction des spécifications à l'aide de langage naturel structuré ou de logique temporelle qui permet la mathématisation des spécifications. Une étude de l'état de l'art, par exemple, [2, 3, 1, 4], sera menée pour choisir au mieux le formalisme de description des spécifications. Ensuite, un algorithme de génération automatique de jeux de tests sera définis et la couverture fonctionnelle s'appuiera sur les métriques de couverture structurelle.

Ce travail de stage s'inscrit dans le développement de logiciel critique de commande dans le monde naval et un cas d'étude issu des systèmes de Naval Group sera utilisé pour démontrer l'intérêt de cette méthodologie.

Profil et candidature. Ce sujet s'adresse à des étudiants en informatique aimant programmer (en C++ ou Java) et ayant des connaissances en logique et tests de logiciels.

Le candidat devra soumettre par courrier électronique les documents suivants :

- une lettre de motivation ;
- un curriculum vitæ ;
- une copie des diplômes et des relevés de notes de licence et master.

1 Contacts

- Alexandre Chapoutot `alexandre.chapoutot@ensta-paristech.fr`
- Laurent Raffaelli `laurent.raffaelli@naval-group.com`

Références

- [1] E. H. Choi, H. Nishihara, T. Ando, N. V. Tang, M. Aoki, K. Yoshisaka, O. Mizuno, and H. Ohsaki. Formal specification based automatic test generation for embedded network systems. *Journal of Applied Mathematics*, 2014 :21, 2014.
- [2] G. Fraser and A. Gargantini. An evaluation of specification based test generation techniques using model checkers. In *2009 Testing : Academic and Industrial Conference - Practice and Research Techniques*, pages 72–81, Sept 2009.
- [3] A. Rajan. Automated requirements-based test case generation. *SIGSOFT Softw. Eng. Notes*, 31(6) :1–2, Nov. 2006.
- [4] E. Shamsoddin-motlagh. Article : A review of automatic test cases generation. *International Journal of Computer Applications*, 57(13) :25–29, November 2012.