Proposition de thèse de doctorat / PhD Subject

Validation à l'exécution de propriétés temps-réel pour systèmes critiques

Mots-clés : validation à l'exécution, enforcement, monitoring, test à base de modèles, temps-réel, traces temporisées, cockpit

Détails

Titre: Validation à l'exécution de propriétés temps-réel pour systèmes critiques
Durée: 36 mois
Type: thèse de doctorat

Encadrants principaux

Antoine Rollet
LaBRI - University of Bordeaux
(+33)5 40 00 35 34
email: rollet__AT__labri.fr

Ylies Falcone
LIG - Université de Grenoble
(+33)4 76 82 72 14
email: Ylies.Falcone__AT__ujf-grenoble.fr

Directeur officiel

Mohamed Mosbah
LaBRI - University of Bordeaux
(+33)5 40 00 69 17
email: mosbah__AT__labri.fr

Contexte

Dans les cockpits d'avion, l'introduction d'interfaces Homme / Machine de nouvelle génération (Tactile, PostWimp,...) augmente la complexité générale du système et peut induire de nouveaux types d'erreur, qui peuvent être dues soit à des pannes, soit à des facteurs humains mal pris en compte. Il est donc nécessaire de mettre en œuvre de nouveaux concepts d'interaction permettant la mise en œuvre de fonctions critiques sur le plan de la sûreté de fonctionnement, et notamment en mettant en place de nouvelles méthodes de validation. Ces concepts doivent notamment permettre de prendre en compte la problématique de facteurs humains.

D'autre part, les récents travaux dans le domaine des méthodes formelles proposent des solutions de surveillance (vérification à l'exécution) et éventuellement de correction (enforcement à l'exécution) qui ne nécessitent pas de connaître la spécification complète du système, mais uniquement les propriétés que l'on souhaite assurer pendant l'exécution du système. Dans le cas de la vérification à l'exécution [Hav00,HR01,HR02,LKK+99,LS08], un « moniteur » est utilisé pour vérifier si une trace vérifie une propriété P sans modifier le comportement du système, alors que dans le cas de l'enforcement à l'exécution

[HMS06,Sch00,Vis00,LBW09,LBW05,Fal09,FFM12], « l'enforceur » va agir comme une sorte de filtre (à la manière d'un firewall) et peut être amené à modifier l'exécution dans un cadre prédéfini afin de garantir la propriété en sortie. Il est pourvu d'une mémoire et est capable de stocker des événements. Si c'est possible, il relâche les actions stockées en mémoire, assurant que la propriété souhaitée est vérifiée. Dans ce cas, l'enforceur modifie l'exécution du système. Les séquences de sortie de l'enforceur doivent être "correctes" (les séquences vérifient la propriété) et "transparentes" (les entrées correctes ne sont pas modifiées, ceci afin de garantir une qualité de service). Selon le type de propriétés et les capacités de modification du mécanisme enforceur, il est envisageable de générer automatiquement cet enforceur. Plus généralement, des mécanismes permettant de contraindre un comportement ont été proposés (synthèse de contrôleur) ainsi que des approches de gestion de systèmes distribués (supervision). Ces méthodes reposent sur des fondements mathématiques prouvés.

Les propriétés temps-réel sont souvent utilisées pour spécifier le comportement de systèmes critiques. Par exemple, une requête du type « la requête req doit être satisfaite », peut devenir « la requête req doit être satisfaite avant 10 millisecondes ». L'utilisation de spécifications du comportement qui prennent en compte les contraintes de temps offre ainsi une expressivité accrue, mais implique de nouveaux challenges dans le processus de validation. Ce genre de propriétés se retrouve dans les cockpits et notamment les interfaces de nouvelles génération. Par exemple, le fait de laisser un doigt appuyé un certain temps sur une interface tactile se modélise aisément avec des propriétés temps-réel.

Les techniques de validation à l'exécution commencent à trouver un écho auprès de partenaires industriels innovants. On peut citer par exemple Microsoft et Google qui ont appliqué ces techniques pour vérifier des traces réseaux. Plus récemment, la NASA a utilisé ces techniques pour garantir la validité des séquences de commandes envoyées au robot Discovery.

Sujet

Les objectifs de cette thèse sont à la fois théoriques et pratiques.

Une première approche d'enforcement à l'exécution utilisant des automates temporisés a été proposée dans [PFJMRN12]. Il n'en existe pas d'autre à notre connaissance. De façon générale, l'objectif de cette thèse de doctorat est d'étendre la vérification et l'enforcement à l'exécution dans un contexte temps-réel, de compléter l'état des résultats théoriques, et de proposer une application concrète sur un exemple industriel, par exemple les cockpits et les Interfaces Hommes Systèmes (IHS).

Certaines questions restent ouvertes. Quelles propriétés temporisées peuvent être monitorées / enforcées? Comment l'ensemble des propriétés enforçables / monitorables évolue en fonction de l'architecture et des différentes primitives du moniteur? Comment peut-on synthétiser un moniteur d'enforcement à partir de spécifications expressives? Intuitivement, en enforcement à l'exécution, une des difficultés est que les opérations de l'enforceur influencent les contraintes de temps du comportement initial.

L'objectif de ce travail est de fournir un framework complet de monitoring / enforcement de systèmes temporisés, avec plusieurs types de propriétés (safety, co-safety, et des propriétés plus expressives) décrites en logiques temporelles temporisées et utilisant plusieurs règles de transparence. L'implémentabilité de cette approche sera aussi étudiée. Les résultats théoriques devront être mis en œuvre dans un outil afin de démontrer l'efficacité de l'approche.

Le domaine d'application de ces méthodes sera la sûreté et l'intégrité des cockpits d'avion. En effet, l'ajout récent d'interfaces Homme / Machine de nouvelle génération (Tactile, PostWimp,...) augmente la complexité générale du système et peut induire de nouveaux types d'erreur, qui peuvent être dues soit à des pannes, soit à des facteurs humains mal pris en compte. Ainsi, la mise en place d'un mécanisme formel non seulement de détection mais aussi de correction type enforceur semble être une solution encourageante, à condition de bien définir le cadre et les règles autorisées. La réalisation concrète de moniteurs d'enforcement appliqués aux cockpits sera étudiée.

De façon plus générale, il s'agit d'étudier les possibilités offertes par les méthodes formelles, et plus

spécifiquement les méthodes de validation à l'exécution pour assurer le contrôle d'intégrité et ainsi améliorer la confiance dans les systèmes de cockpit avec les IHM sécurisées de nouvelle génération. En effet, compte tenu de la complexité générale du système, les nouvelles approches formelles de validation à l'exécution semblent prometteuses étant donné qu'elles ne nécessitent pas la connaissance de la spécification globale du système. De plus, dans certaines situations, le fait de transférer une information erronée (par exemple, la saisie erronée sur un écran tactile) peut avoir de graves conséquences. Ainsi, la mise en place d'un mécanisme non seulement de détection mais aussi de correction type enforceur est une solution prometteuse.

Prérequis

- Master en informatique
- Une première expérience dans le domaine des méthodes formelles serait souhaitable.

Références

[PFJMRN12] S. Pinisetty, Y. Falcone, T. Jéron, H. Marchand, A. Rollet, and O. Nguena Timo.
"Runtime enforcement of timed properties"
In RV12: International conference on Runtime Verification (LNCS)
To appear.

[AD94] Rajeev Alur and David L. Dill.
"A Theory of Timed Automata".
Theoretical Computer Science 126(2), pages 183-235, 1994.

[Hav00] Klaus Havelund.
"Using runtime analysis to guide model checking of Java programs".
In Proceedings of the 7th International SPIN Workshop on SPIN Model Checking and Software Verification, pages 245-264, London, UK, 2000. Springer-Verlag.

[HR01] Klaus Havelund and Grigore Rosu.
"Monitoring Java programs with Java PathExplorer".
Technical report, RIACS, 2001.

[HR02] Klaus Havelund and Grigore Rosu.
"Synthesizing monitors for safety properties".
In TACAS '02 : Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pages 342-356, London, UK, 2002. Springer-Verlag.

[LKK+99] Insup Lee, Sampath Kannan, Monjoo Kim, Oleg Sokolsky, and Mahesh Viswanathan.
"Runtime assurance based on formal specifications".
In In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 1999.

[LS08] Martin Leucker and Christian Schallhart.
"A brief account of runtime verification".
Journal of Logic and Algebraic Programming, 78(5) :293-303, may/june 2008.

[HMS06] Kevin W. Hamlen, Greg Morrisett, and Fred B. Schneider.
"Computability classes for enforcement mechanisms".
ACM Trans. Program. Lang. Syst., 28(1) :175-205, 2006.

[Sch00] Fred B. Schneider.

"Enforceable security policies".
ACM Trans. Inf. Syst. Secur., 3 1) :30-50, 2000.

[Vis00] Mahesh Viswanathan.
"Foundations for the run-time analysis of software systems".
PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2000.

[LBW09] Jay Ligatti, Lujo Bauer, and David Walker.
"Run-time enforcement of nonsafety policies".
ACM Transactions on Information and System Security, 12(3) :1-41, January 2009.

[LBW05] Jay Ligatti, Lujo Bauer, and David Walker.
"Enforcing Non-safety Security Policies with Program Monitors".
In ESORICS, pages 355-373, 2005.

[Fal09] Ylies Falcone
"Etude et mise en oeuvre de techniques de validation à l'exécution"
Phd thesis, University of Grenoble 1, France

[JJ05] Claude Jard and Thierry Jeron.
"Tgv : theory, principles and algorithms".
International Journal on Software Tools for Technology Transfer (STTT), 7(4) :297-315, 2005.

[BJSK11] N. Bertrand, T. Jéron, A. Stainer and M. Krichen.
"Off-line Test Selection with Test Purposes for Non-Deterministic Timed Automata".
In Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'11), Saarbrücken, Germany, April 2011. LNCS 6605, pages 96-111. Springer.

[NR11] O. Nguena-Timo and A. Rollet
Test Selection for Data-Flow Reactive Systems based on Observations
In 7th Workshop on Advances in Model Based Testing A-MOST 2011, March 21, 2011, Berlin, Germany, 8p.

[NMR10] O. Nguena Timo and H. Marchand and A. Rollet
Automatic Test Generation for Data-Flow Reactive Systems with time constraints
In 22nd IFIP International Conference on Testing Software and Systems (ICTSS10), (ex Testcom/Fates), November 8-12, 2010, Natal, Brazil, 6p. (short paper)

[FFM12] Y. Falcone, J-C. Fernandez, L. Mounier.
What can you Verify and Enforce at Runtime?
In STTT: Software Tools for Technology Transfer - Special issue on Runtime Verification.