FNRAE 6ème appel à propositions
10 juin 2008

Méthodes de vérification des logiciels et systèmes

Ce 6ème appel à propositions de la *Fondation de Recherche pour l'Aéronautique et l'Espace* (FRAE) couvre le champ des méthodes et outils de spécification, vérification, synthèse et certification des logiciels, designs FPGA et systèmes, en particulier mais pas exclusivement pour l'aéronautique et l'espace.

Dans le contexte toujours croissant des technologies numériques, la maîtrise des logiciels et des systèmes informatiques nécessite la prise en compte de nouveaux facteurs : l'environnement dans lequel ils sont déployés qui est par nature non fiable et évolue très rapidement ; l'intégration à grande échelle de composants distribués, hétérogènes, souvent autonomes ; l'évolution des architectures matérielles avec l'apparition des processeurs multi-cœurs et l'usage des FPGAs; les objets physiques sur lesquels sont embarqués les systèmes et les applications et les exigences fortes de sûreté, de sécurité et de temps réel qu'ils imposent. Dans ce contexte sont apparus des concepts nouveaux de plus en plus adoptés par les industriels tels que les architectures logicielles orientées services, la programmation agile, et les systèmes de systèmes.

Pour prendre en compte ces évolutions, les méthodes et outils de spécification, vérification, synthèse et certification visant à produire des systèmes et logiciels sûrs et sécurisés, doivent faire l'objet de recherches et de développements visant à élargir la classe de problèmes auxquels ils s'appliquent aujourd'hui et à favoriser leur passage à l'échelle. L'objectif de cet appel est d'étendre le champ des méthodes formelles et de la vérification selon l'un des trois axes suivants.

Axe 1 : Méthodes formelles et modèles pour le co-design (hardware/software) et les systèmes de systèmes

L'appel sur cet axe porte en particulier sur les objectifs suivants:

- Prendre en compte dans les spécifications formelles les propriétés non fonctionnelles issues des contraintes matérielles du déploiement et des ressources disponibles, les contraintes de temps, les exigences de sécurité, en particulier en considérant les travaux de conception et d'utilisation des plateformes de confiance telles que le TPM (Trusted Platform Module).

- Etendre les classes de modèles et de propriétés étudiées (modèles hybrides associant grandeurs discrètes et continues, modèles stochastiques, propriétés d'autonomie…), étudier leur combinaison et leur complémentarité en intégrant différents points de vue, en particulier pour la conception de systèmes de systèmes.

Axe 2 : Les techniques d'analyse pour la sûreté et la sécurité

Dans le cadre du déploiement de grands systèmes intégrant des logiciels, les techniques d'analyse sont importantes pour répondre aux exigences de sûreté (garanties sur le comportement nominal attendu) et de sécurité (garanties de résistance dans des contextes malveillants). Parmi les voies à explorer on peut en particulier envisager :

- d’adapter aux logiciels et designs FPGA les techniques développées en sûreté de fonctionnement afin de réaliser des analyses de dépendances, de vulnérabilités, de défaillances, de dysfonctionnements

- de développer les techniques d’interprétation abstraite et d’analyse de codes pour permettre leur passage à l'échelle.

Axe 3 : Les techniques de vérification de code et l’intégration de composants

Cet axe privilégie les objectifs suivants :

- Prendre en compte les composants du commerce, les logiciels préalablement développés et les logiciels libres : il s’agit d’étudier la vérification à posteriori de codes existants, la rétro-ingénierie de codes permettant d’extraire des propriétés de correction et de démontrer l’absence de risques résiduels.

- Etendre les techniques de vérification et de certification de programmes aux codes exécutables et aux systèmes, en poursuivant deux voies complémentaires :

  - la certification des outils de production et de validation de code (compilateurs, générateurs de code, synthétiseurs de logique, analyseurs statiques, model checkers, démonstrateurs de théorèmes), la production de certificat attaché au code et vérifiable a posteriori par ses utilisateurs.

  - l’utilisation combinée de méthodes de tests et de preuves existantes (vérification de modèles, analyse statique, interprétation abstraite, raffinement, preuve interactive, génération de tests) et l’intégration de ces méthodes dans des environnements de conception de systèmes et de production de code, tant pour le logiciel que pour le matériel.