Présentation du projet ASTRÉE au séminaire RNTL Thème 1

Analyse Statique de Logiciel Temps Réel Embarqué

Patrick Cousot (coordinateur)

Radhia Cousot

Famantanantsoa Randimbivololona

École normale supérieure, École polytechnique, Airbus France

Patrick.Cousot@ens.fr, Radhia.Cousot@polytechnique.fr, famanta.randim@airbus.com

http://www.astree.ens.fr/

Présentation du projet (1) Objectifs, Méthode, Verrous à lever

- **Objectifs:**
  - Construire un **analyseur statique** capable:
    - de démontrer *l'absence d'erreurs à l'exécution* et de *respect de bonnes pratiques de programmation* pour des programmes critiques de contrôle-commande, temps-réel, synchrones, embarqués, écrits en C ;
    - de manière automatique, exhaustive, rapide et précise (pas ou peu de fausses alarmes).
- **Méthode:**
  - Application de la théorie de l'**interprétation abstraite** (approximation effective de la sémantique du programme) ;
  - Adaptation et paramétrisation de l'analyse pour le domaine d'application des **programmes synchrones**.
- **Verrous technologiques à lever:**
  - démontrer que les méthodes formelles automatiques basées sur l'interprétation abstraite passent à l'échelle industrielle.

Présentation du projet (2) Partenariat et Organisation

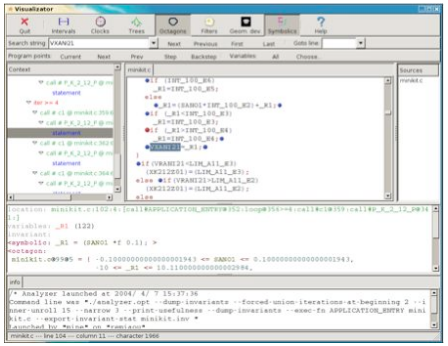
- **Partenariat:**
  - ENS (École normale supérieure, équipe de P. Cousot, Paris)
  - X (École polytechnique, équipe de R. Cousot, Palaiseau)
  - Airbus France (EYY, F. Randimbivololona & J. Souyris, Toulouse)
- **Organisation:**
  - L'analyseur est développé à l'ENS en partenariat avec l'X
  - L'analyseur est testé, mis en œuvre et évalué par Airbus France sur différents codes de commande de vol électrique (A340 et A380).

Présentation du projet (3) Coûts et délais

- **Coûts:**
  - Les coûts académiques les plus importants sont ceux des personnels : principalement supportés par le CNRS et l'ENS ;
  - L'apport du RNTL est essentiel (machines, voyages Paris/Toulouse & conférences, personnel non statutaire pour les situations de transition).
- **Délais:**
  - Le support du RNTL au projet a pris du retard au démarrage ;
  - Le projet est dans les délais, plutôt en avance, mais les applications sont sur des logiciels qui évoluent et peuvent faire surgir de nouvelles difficultés ;
  - Nouveaux verrous à lever pour la commande de vol électrique de l'A380 (500 000 lignes au final ?)

Résultats obtenus à ce jour : (1) Principaux livrables

- La livraison principale consiste en la **démonstration d'un analyseur statique** qui fonctionne, aux différents stades successifs de son développement sur 3 ans



Résultats obtenus à ce jour : (2) Résultats Scientifiques et Publications

- **Résultats scientifiques:**
  - **Précision:** en novembre 2003, l'analyseur ASTRÉE a été capable de démontrer *complètement automatiquement* l'absence de toute erreur à l'exécution dans le logiciel de commande de vol électrique primaire de l'Airbus A340 (132 000 lignes de C) ;
  - **Performances:**
    - 1h20 sur un PC 32 bits à 2.8 GHz, 300 Mo de mémoire,
    - 50 mn sur un Athlon 64 d'AMD, 580 Mo de mémoire.

  Une première mondiale!
- **Publications:**
  - Fondements, structure générale et domaines abstraits de l'analyseur statique [1,2]
  - Domaines abstraits spécialisés pour l'analyse de code réactif (filtres digitaux [3] et flottants [4])

Résultats obtenus à ce jour : (3) un exemple … difficile à analyser

```
/* filter.c */
typedef enum {FALSE = 0, TRUE = 1} BOOLEAN;
BOOLEAN INIT; float P, X;
void filter () {
  static float E[2], S[2];
  if (INIT) {
      S[0] = X;
      P = X;
      E[0] = X;
  } else {
       P = (((((0.5 * X) - (E[0] * 0.7)) + (E[1]
   * 0.4)) + (S[0] * 1.5)) - (S[1] * 0.7));  }
   E[1] = E[0];  E[0] = X;  S[1] = S[0];  S[0] =
   P;}

void main () {
X = 0.2 * X + 5;
INIT = TRUE;
while (1) {
     X = 0.9 * X + 35;
    filter ();
    INIT = FALSE;  }}
```

P 2 [-1327.05, 1327.05] (en tenant compte des erreurs d'arrondi)

Simulation des conditions d'appel de la fonction `filtre.c` trouvées par l'analyse statique

Retombées scientifiques

- **Les retombées scientifiques concernent :**
  - La preuve pratique que l'interprétation abstraite passe à l'échelle
  - La preuve de faisabilité pour des programmes industriels
  - La méthode de conception et la structure de l'analyseur est validée et peut servir de modèle général [1, 2]
  - Les domaines abstraits conçus spécialement pour cet analyseur sont d'un usage général pour l'analyse de programmes de contrôle/commande [2,3,4]

Retombées d'exploitation et perspectives

- **Conditions de réalisation de l'analyseur:**
  - La conception et la réalisation de l'analyseur demande la conjugaison de rares compétences en interprétation abstraite et en programmation
  - Le travail réalisé va bien au delà du simple prototype académique classique
  - L'analyseur a été entièrement réalisé par des chercheurs et enseignants/chercheurs (sans le soutien technique d'ingénieurs)
- **Conséquences sur l'exploitation industrielle:**
  - Impossible pour des chercheurs d'assurer le suivi technique nécessaire pour une exploitation industrielle avec des utilisateurs nombreux et non spécialisés
- **Perspectives d'exploitation:**
  - Le problème de l'*industrialisation* est maintenant très urgent.

Bilan "RNTL"

- **Processus de sélection:** essentiellement inconnu des soumissionnaires,
- **Mise en route:** 2 décembre 2002.
- **Suivi:** envoi de rapports d'activité tous les 6 mois.
- **Fin de projet:** 1er décembre 2005.
- **Ce que le RNTL a apporté au projet:** seul cadre possible pour établir des liens université-industrie, soutien matériel indispensable: machines, voyages Paris/Toulouse & conférences, personnel non statutaire pour les situations de transition
- **Problèmes rencontrés:** financement relativement faible mais fortement apprécié, instabilité des situations des jeunes chercheurs, aucun autre problème rencontré à ce jour, problème futur de l'industrialisation de l'analyseur.

Conclusion et propositions pour la suite

- **Enseignements principaux:**
  - Les projets ambitieux ont besoin d'une continuité à long terme (exploratoire –> pré-compétitif –> industrialisation)
  - Les équipes universitaires ont besoin de moyens et de stabilité pour satisfaire aux exigences de réactivité de la coopération industrielle
- **Thèmes à développer:**
  - Vérification de la qualité des logiciels critiques
- **Nouveaux verrous à lever:**
  - +1,5 an : Commande de vol de l'A380 (500 000 lignes)
  - +2,5 ans : Analyse statique de logiciels asynchrones
  - +2,5 ans : Analyse dès la spécification (SAO, Scade) pour des propriétés fonctionnelles complexes.

Annexe
(1) références des publications

[1] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux & Xavier Rival. Design and Implementation of a Special-Purpose Static Program Analyzer for Safety-Critical Real-Time Embedded Software, invited chapter. In *The Essence of Computation: Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones*, T. Mogensen and D.A. Schmidt and I.H. Sudborough (éditeurs). Lecture Notes in Computer Science 2566, pp. 85—108, © Springer.

[2] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, & Xavier Rival. A Static Analyzer for Large Safety-Critical Software. In *PLDI 2003 — ACM SIGPLAN SIGSOFT Conference on Programming Language Design and Implementation*, 2003 Federated Computing Research Conference, 7—14 juin 2003, San Diego, Californie, USA, pp. 196—207, © ACM.

[3] Jérôme Feret. Static analysis of digital filters. In *ESOP 2004 — European Symposium on Programming*, D. Schmidt (éditeur), Mar. 27 —Apr. 4, 2004, Barcelone, Espagne, Lecture Notes in Computer Science 2986, pp. 33—48, © Springer.

[4] Antoine Miné. Relational abstract domains for the detection of floating-point run-time errors. In *ESOP 2004 — European Symposium on Programming*, D. Schmidt (éditeur), Mar. 27 —Apr. 4, 2004, Barcelone, Espagne, Lecture Notes in Computer Science 2986, pp. 3—17, © Springer.

Annexe
(2) Site web du projet ASTRÉE

http://www.astree.ens.fr