*R*éseau
d'*I* ngénierie
de la *S* ûreté de fonctionnement

Atelier Thématique

Usages et perspectives pour la production de logiciels sûrs

Vendredi 19 octobre 2001 - LAAS-CNRS, Toulouse

Contexte

- La production de logiciel sûr engendre des coûts élevés
  - ✓ effort consacré au développement : 5 à 10 homme.année par KLOC
  - ✓ 75 % de cet effort consacré aux vérifications
  - ✓ maintenance très coûteuse
- Stratégie de développement
  - ✓ approche d'ingénierie structurée et contrôle de processus
  - ✓ place importante aux activités de vérification
    - revues, analyses, test
  - ✓ utilisation limitée de vérifications formelles et preuves

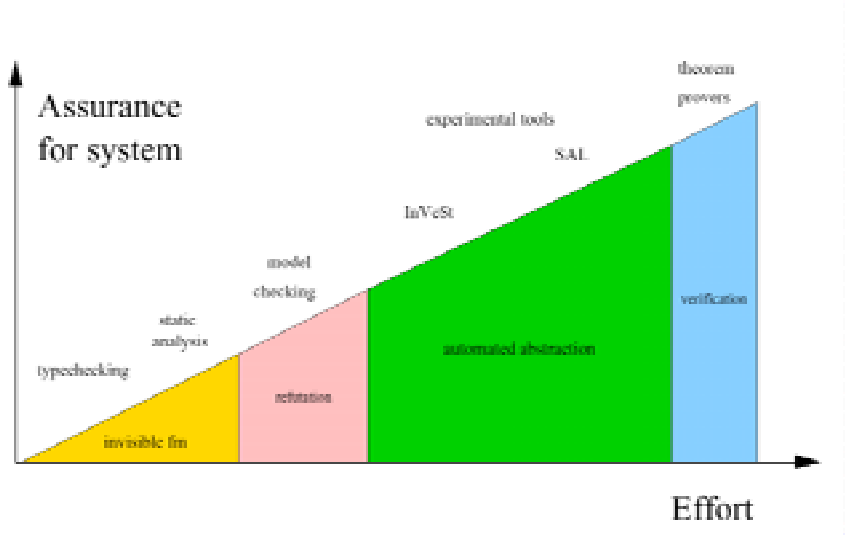
Problèmes et Défis

- Peu d'informations relatives au processus de développement sont effectivement utilisées pour faciliter/alléger la vérification
- Difficulté de maîtriser le passage des spécifications au logiciel
  - ✓ Complexité
  - ✓ Évolutions
- Comment aboutir à une meilleure coopération entre les activités de création et les activités liées à la vérification
  - ✓ Sachant que le logiciel a été développé de façon rigoureuse, comment s'y prendre pour le vérifier ?
  - ✓ Peut - on supprimer certaines activités traditionnelles sans dégrader le niveau de confiance ?

Apport des méthodes formelles

- Réduire les imprécisions, ambiguïtés, incohérences via l'utilisation de spécifications formelles
- Rechercher les fautes au plus tôt dans le cycle de développement
  - ✓ Vérification à base de modèle
  - ✓ Preuve de propriétés
- Contribuer au déroulement des tests
  - ✓ Oracle
  - ✓ Génération de séquences de tests à partir de modèles

Méthodes formelles et assurance



John Rushby, "Disappearing formal methods", http://www.csl.sri.com/~rushby

Quelques pistes

- La formalisation du développement peut contribuer à optimiser l'effort de vérification
  - ✓ utilisation de B dans le ferroviaire
  - ✓ méthode cleanroom

  ➥ dans quelle mesure une telle approche est applicable à des systèmes aussi complexes que les systèmes avioniques ?

- Intégration de formalismes, outillage et automatisation
  - ✓ SafeAir (avionique)
  - ✓ Crysis (contrôle-commande)

- Vérification modulaire et composition de propriétés
  - ✓ maîtrise de la complexité
  - ✓ réutilisation pour la sûreté de fonctionnement

Programme

9h - 9h30	*Accueil*	
9h30 - 9h50	*Introduction et présentation de l'atelier*	M. Kaâniche (LAAS-CNRS)
9h50 - 10h30	*Vérification de logiciels avioniques par analyse statique*	F. Randimbivololona (Airbus France)
10h30 - 11h10	*État des lieux et perspectives pour le développement de logiciels embarqués sur satellite*	P. Welby (Astrium)
11h10 - 11h30	*Pause*	
11h30 - 12h10	*Maîtrise du développement des logiciels de sécurité à Technicatome*	J-M. Tabart (Technicatome)
12h10 - 12h50	*B : une méthode de développement de logiciel sûr*	L. Pelhate (RATP)
12h50 - 14h	*Buffet*	
14h - 14h40	*Définition d'une stratégie de test d'un protocole à méta-objets*	J-C. Ruiz Garcia (LAAS-CNRS)
14h40 - 15h40	*Discussion générale*	
15h40 - 16h	*Pause*	
16h - 17h	*Synthèse et conclusions*	