HI-LITE

Simplifying the use of formal methods

Rapprocher les méthodes formelles, l'analyse statique et les tests

29 mai 2013

AdaCore
The GNAT Pro Company

ALTRAN

cea
list

ASTRIUM
AN EADS COMPANY

Inria
informatics mathematics

THALES

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

HI-LITE

- Projet de recherche du pôle de compétitivité System@tic
- Durée: 3 ans
- Enveloppe globale : 3,9 M€
  - Financement : 1,4 M€
  - OSEO, Conseil Général Essonne
- 6 partenaires

AdaCore The GNAT Pro Company

ALTRAN

cea list

ASTRIUM AN EADS COMPANY

Inria informatics mathematics

THALES

HI-LITE Approche hi-Lite

Des logiciels de plus en plus complexes

Phases de développement réduites

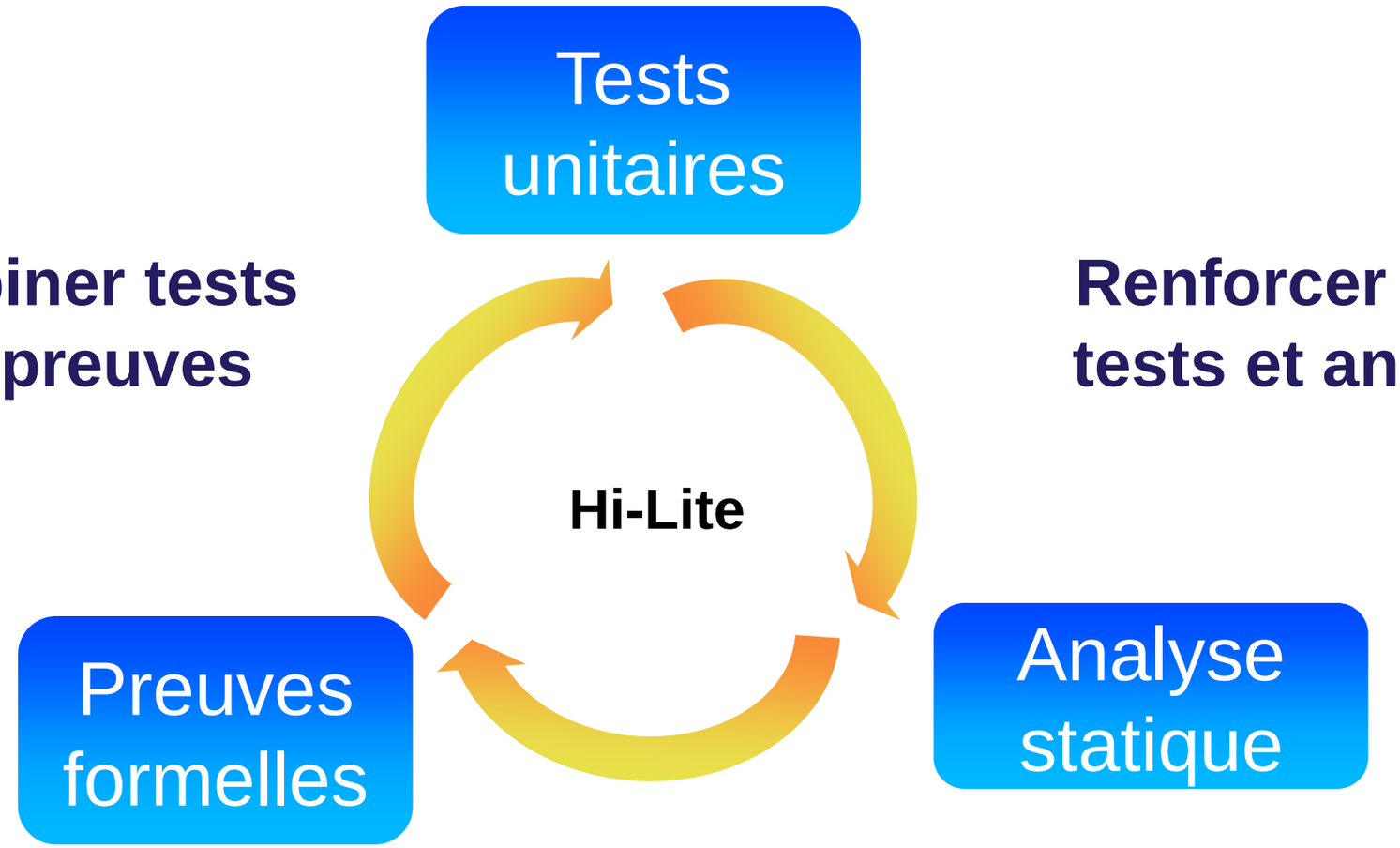
Budgets réduits

Limite d’une approche “tests” seule

Limite d’une approche “méthode formelle”

Unification des deux approches

Le projet Hi-Lite



Combiner tests et preuves

Renforcer mutuellement tests et analyse statique

Faciliter la preuve formelle grâce à l’analyse statique

HI-LITE Buts du projet

Rendre l’utilisation des méthodes formelles plus simple

- Barrière d’entrée faible, gains incrémentaux, meilleure IHM

Décloisonner les méthodes formelles

- Faciliter une utilisation mixte tests et preuves à partir d’un langage commun

Logiciel libre

Bénéfices industriels

- Réduction des coûts
- Augmentation de la qualité
- Faciliter la réutilisation de composants logiciels

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

HI-LITE Déroulement du projet

1ère année

- Compilation des exigences
- Définition des langages

2ème et 3ème années

- Création de traducteurs
- Amélioration d’outils d’analyse et de test
- Bibliothèques et interfaces utilisateurs
- Applications industrielles

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

HI-LITE Spécifications

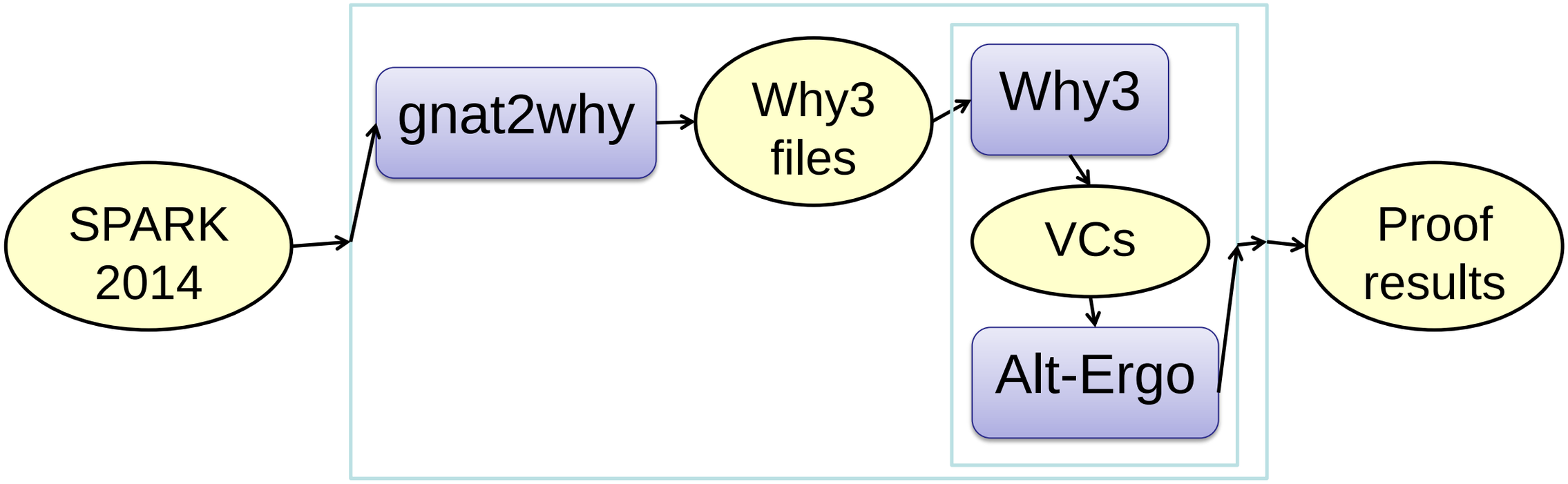
- Exigences collectées lors de la première année, affinées l’année suivante, et validées en dernière année
- Voir points principaux sur le retour d’expérience des études de cas

HI-LITE Définition des langages

- 2 nouveaux langages ont été conçus
  - ALFA, puis SPARK 2014 (basé sur Ada 2012)
  - E-ACSL (basé sur ACSL et C)
  - Manuels de référence complets disponibles
- Langage commun pour la preuve et le test

Création de traducteurs

- Outils **gnatprove** et **gnat2why** développés pour traduire le langage SPARK 2014 vers le langage intermédiaire Why et générer des résultats de preuve



- Greffon Frama-C: E-ACSL vers C
  - Fournit la sémantique exécutable du langage ACSL

HI-LITE Amélioration d'outils

- Ajout du support de SPARK 2014 dans GNAT Pro et GNATtest
  - Fournit la sémantique exécutable du langage SPARK 2014 (compilation et test)
- Amélioration de CodePeer (analyse statique de code Ada)
  - Support du langage SPARK 2014
  - Nouvelles détections d'erreurs de logique
  - Complète les preuves formelles par l'analyse statique

HI-LITE Amélioration d'outils

- Évolutions du prouveur Alt-Ergo
  - Plus efficace, permet de prouver automatiquement plus de formules, géneration de contre-exemples
- Nouvelle génération pour la plate-forme Why
  - Why3 : réecriture complète, avec plus de fonctionalités et une plus grande facilité d'extension
  - Fournit une API complète de programmation
  - Gestion des sessions de preuve
- Évolutions de la plate-forme Frama-C
  - Meilleure utilisabilité
  - Combinaisons inter-analyses

HI-LITE Bibliothèques et interfaces utilisateur

- Bibliothèque formelle de conteneurs SPARK 2014
- Intégration de gnatprove et gnattest dans GPS
- Support de SPARK 2014 dans GPS et GNATbench (plug-in Eclipse)
- Améliorations dans AltGr-Ergo

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

HI-LITE Applications Industrielles

- Logiciel de vol spatial (Astrium Space Transportation)
  - Gestion de la mission et du véhicule spatial
  - Algorithmes de contrôle/commande : déploiement & orientation de panneaux solaires
- Applications SPARK (Altran, AdaCore)
  - Exemples/didactiques
  - Tokeneer : identification biométrique sécurisée
- MyCCM (Thales)
  - Génération de code à partir de modèles architecturaux

HI-LITE Retour d’expérience des études de cas

- Preuve exhaustive d’absence d’erreurs d’exécution
  - “buffer overflow”, division par zéro, dépassement de bornes, …
- Preuve de propriétés fonctionnelles définies par l’utilisateur
- Cohérence entre unités logicielles
- Formalisation de cas de test
  - Au lieu d’une description en langage naturel, parfois ambigüe et difficile à vérifier

HI-LITE Retour d’expérience des études de cas

- Langage et sémantique communs pour les preuves et le test
  - Aide à l’écriture et mise au point des contrats
  - Incite à l’utilisation de contrats logiciels
  - Permet de ne pas dépendre seulement d’une approche
  - Facilite le passage de test à la preuve et vice-versa
- Langage de programmation étendu
  - Rend possible l’utilisation des méthodes formelles dans plus de cas
  - Aide à la réutilisation de composants logiciel existants

HI-LITE Retour d’expérience des études de cas

- Meilleur outillage/interface utilisateur
  - Aide à la compréhension des messages d’erreur
  - Développement et méthode de travail facilités
    - Utilisation simple, rapide et continue des outils
    - Utilisation modulaire
- Possibilité d’utiliser les methodes formelles à partir de modèles architecturaux

HI-LITE

Présentation du projet

Déroulement du projet

Réalisations

Démonstrations

Perspectives

Hi-Lite Dissémination des produits du projet

- Forge Hi-Lite : http://forge.open-do.org
  - Projets hi-lite, spark2014
- SPARK/Hi-Lite GPL 2013 : http://libre.adacore.com
- Frama-C: http://frama-c.com/
- Why3/Alt-Ergo: http://toccata.lri.fr
- Listes de diffusion, dépot de sources, …
- Nombreuses publications et participations à des conférences

HI-LITE Fonctionnement du projet

- Collaboration étroite entre les partenaires du projet
- Nombreux retours et améliorations
- Taille idéale pour un projet de recherche
- Permettant une structure relativement légère
- Ouverture complète du projet avec partitions externes
- Amélioration de l’état de l’art se basant sur des outils existants et quelques nouveaux outils

HI-LITE Travaux futurs

- Preuves mixtes Ada/C (gnatprove/frama-c)
- Combinaison des résultats de preuve et de test automatisée
- Amélioration nouvelles dans Why3 et Alt-Ergo (confiance dans l'outil/qualification, performance et fiabilité, support des nombres flottants)

HI-LITE Perspectives

- Nouveau produit AdaCore/Altran : SPARK 2014
  - http://spark-2014.org
    - Disponible au 1er trimestre 2014

- AdaCore: 20+ salariés en 2010, 30+ en 2013
- Création de 3 emplois et 3 autres emplois confortés

HI-LITE | Questions ?