

Modélisation: outillage et intégration



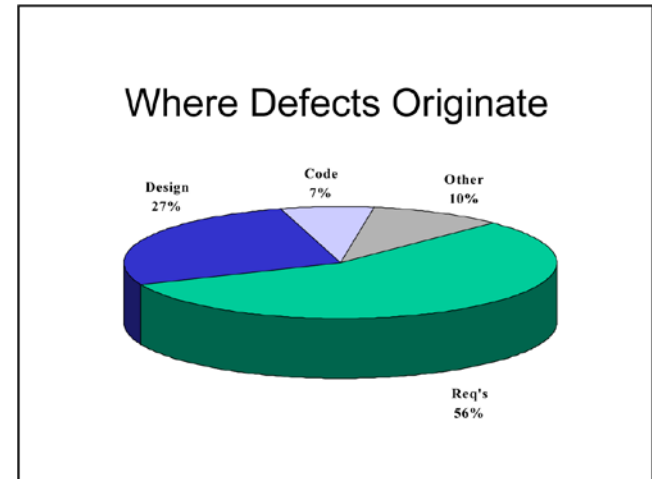
Emmanuel Gaudin
emmanuel.gaudin@pragmadev.com

Un réel besoin

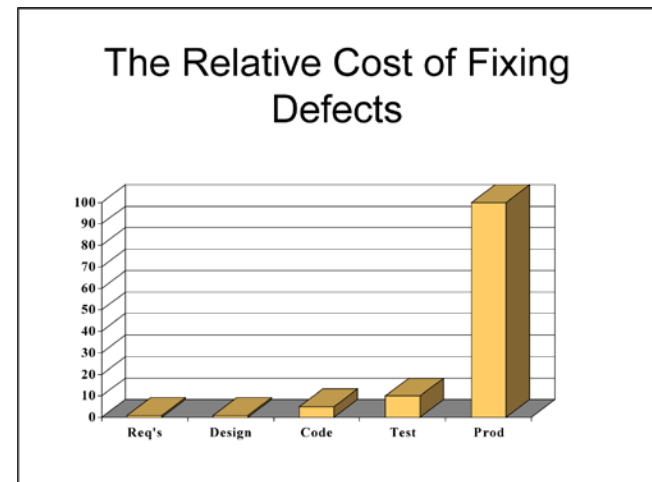
“Le logiciel **double** tous les **deux** **ans**. Le volume final rend extrêmement difficile de garantir le niveau de qualité et de test avec les process et les outils traditionnels.”

Wind River Market Survey of Device Software Testing Trends and Quality Concerns in the Embedded Industry

Juin 2010



Ne pas oublier la règle des 1:10:100



Source: James Martin study

Un réel besoin de modélisation

- Se concentrer sur le **Quoi** plutôt que sur le **Comment**,
- Modéliser c'est **communiquer** et **documenter**,
- La **lisibilité** des systèmes complexes devient critique: qui construirait une maison sans dessiner le moindre plan ?
- **Maîtrise** de la productivité,
- Amélioration de la **qualité**.



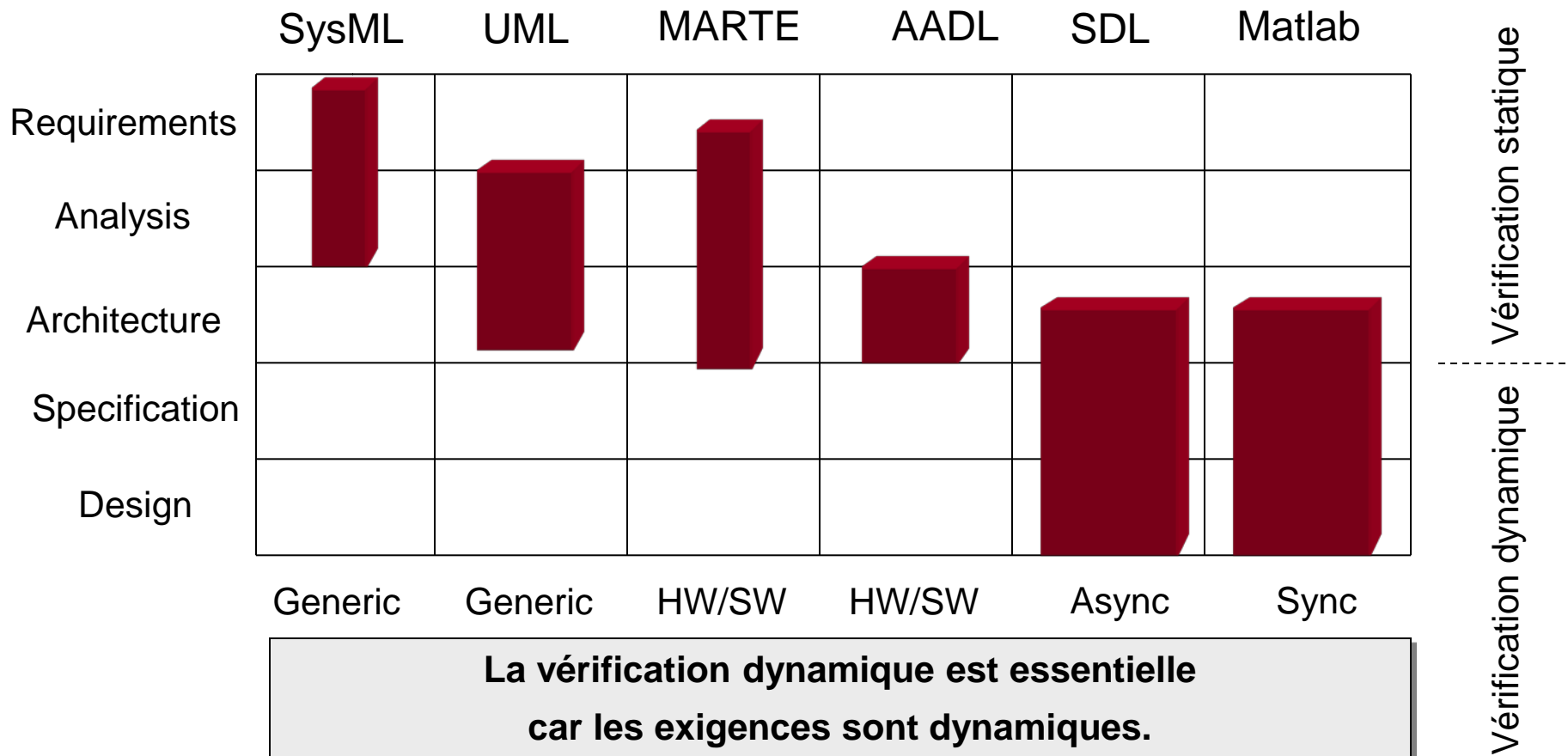
Caractéristiques d'un bon langage de modélisation

- Le modèle abstrait doit être indépendant de la plateforme d'implémentation,
- Le modèle abstrait doit pouvoir être traduit sur une plateforme d'implémentation,
- Pour cela on s'appuie généralement sur une machine virtuelle qui offre:
 - un certain nombre de **services**,
 - une **sémantique** d'action.

Les langages existants

SDL	Specification and Description Language is an ITU-T standard. <ul style="list-style-type: none"> • Event oriented, • Used by ETSI to standardize telecommunication protocols, • Formal (complete and non-ambiguous). 	✓
UML	Unified Modeling Language is an OMG standard. <ul style="list-style-type: none"> • Can be used to represent any type of systems, • Informal. 	✗
SysML	System Modelling Language	✗
AADL	Architecture Analysis Description Language	✓
MARTE profile	Modeling and Analysis of Real-Time and Embedded systems	✗
Z.109	UML profile based on SDL	✓
Lustre / Esterel	Synchronous programming languages for the development of complex reactive systems	✓
MATLAB	MATrix LABoratory	✓
Autosar	AUTomotive Open System Architecture	✗
SART	Structured Analysis for Real Time (obsolete)	✗

Positionnement des langages



Pas de spécificité temps réel UML

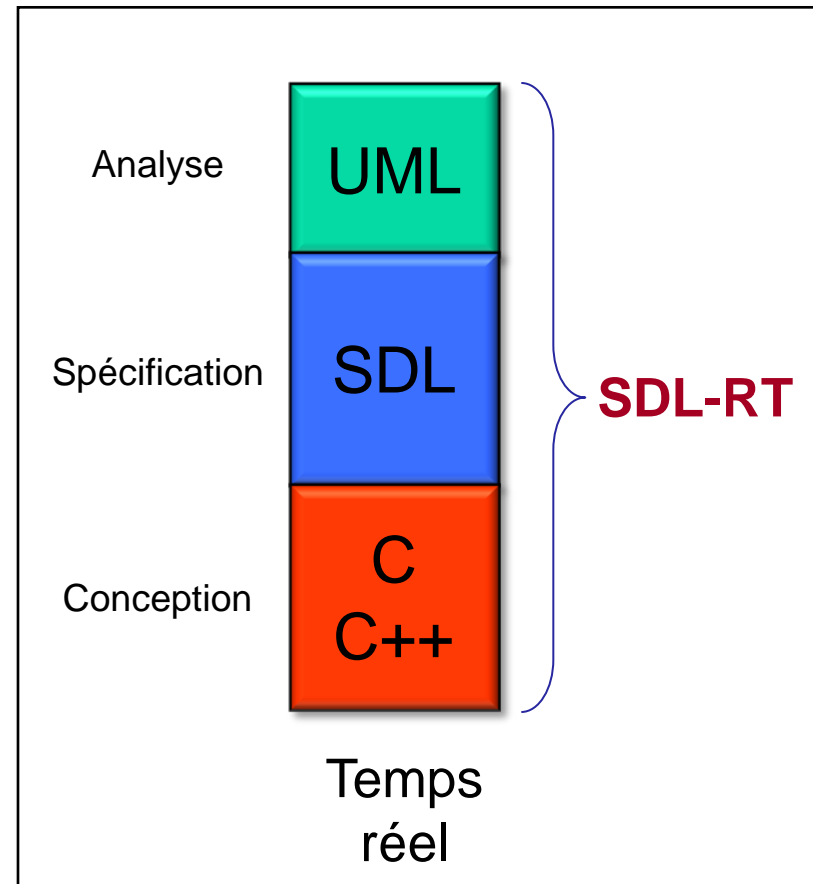
- UML1.x est trop générique pour décrire la dynamique du modèle,
- UML 2.x a introduit des profils spécifiques par domaine:
 - Les outils ont implémenté des profils propriétaires,
 - Les utilisateurs passent plus de temps à définir leur profil que leur système,
 - Il existe un profil standard basé sur le SDL: Z.109.

Intérêts du SDL

- Les niveaux d '**abstraction graphiques** du SDL (architecture, communication, comportement) apportent beaucoup aux équipes de développement pour représenter l'aspect dynamique du système.
- Le SDL étant formel il est possible de **simuler** le comportement du système sur station de développement avec des outils de débogue graphiques.
- Le SDL étant formel on peut **générer** depuis la description SDL le code pour la cible.
- Le SDL est **orienté objet** et permet donc une réutilisation aisée de composants logiciels existants (les protocoles de télécommunication sont écrits en ' objet ' par l 'ETSI).
- SDL a les caractéristiques pour décrire un bon **PIM**.

Une approche semi-formelle: SDL-RT

- Utiliser les concepts UML à très haut niveau: analyse / expression des besoins.
- Conserver les niveaux d'abstraction graphiques du SDL (architecture, communication, comportement).
- Utiliser la syntaxe et les types de données du langage C à la place de ceux du SDL.
- Supprimer les concepts SDL sans réalité technique d'implémentation.
- Compléter les concepts SDL pour coller à la réalité (interruptions, sémaphores).





specification & description language - real time

SDL-RT est:

- Permet de généraliser l'utilisation de SDL à toutes les applications temps réel s'appuyant sur un RTOS,
- Lisible et basé sur un format de stockage textuel normalisé (XML),
- Proposé à l'ITU pour intégration dans le SDL,
- Est un profil UML pour les applications temps réel.





- Specification and Description Language, standard de l'ITU-T pour décrire les spécifications détaillées des protocoles afin d'assurer l'inter opérabilité des implémentations.
- Mis à jour tous les 4 ans depuis 1976.
- Version majeures:
 - SDL 1988: Version de maturité
 - SDL 1992: Introduction de l'orientation objet
 - SDL 2000: Alignement UML
 - SDL 2010: Introduction de SDL-RT
- Conférence annuelle
 - SDL Forum (<http://www.sdl-forum.org/>)
 - SAM workshop (satellite event of Models)
- 11 outils commerciaux, 10 outils dans le domaine public.
- Technologie intégrée dans le framework TASTE de l'ESA

Test

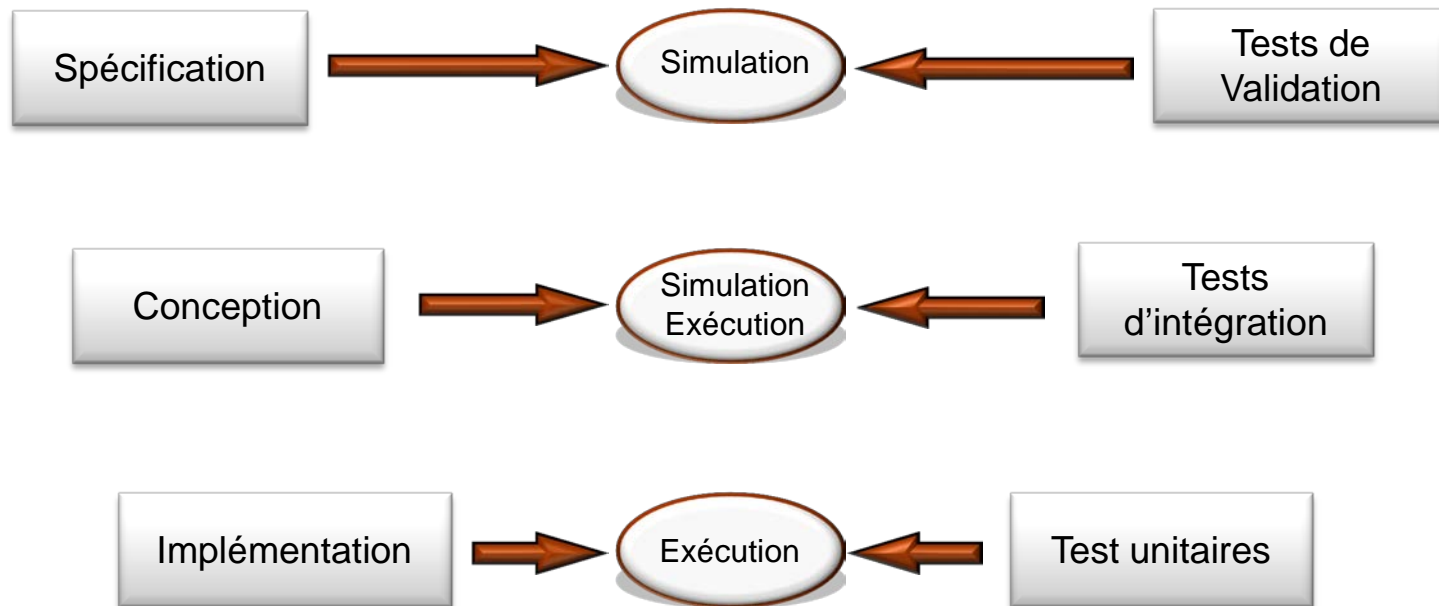
- Basé sur le standard international

TTCN-3:

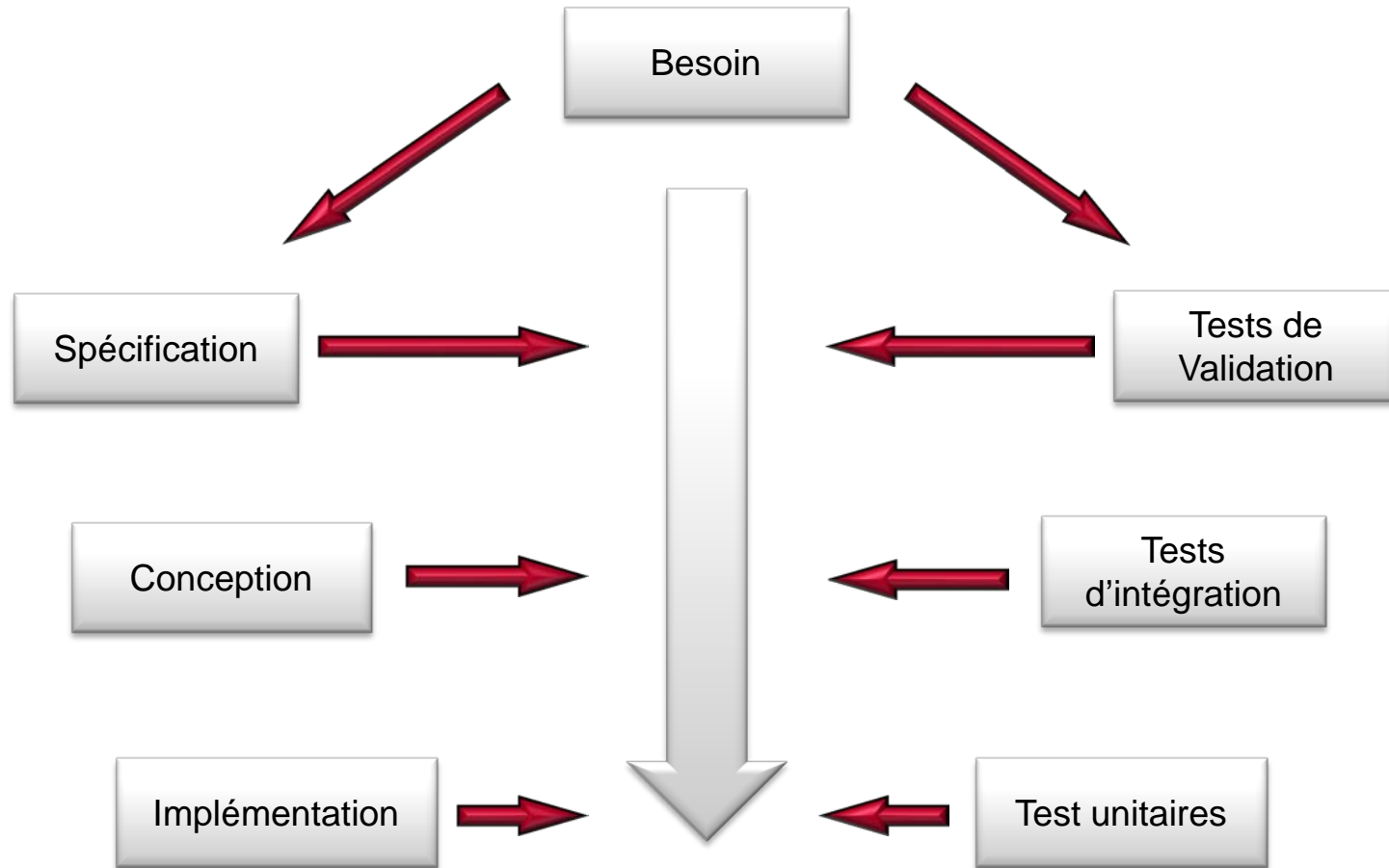
- Définition des types de données,
 - Définition de Templates,
 - Cas de Test,
 - Control d'exécution.
- Exécutable:
 - Simulation contre un modèle,
 - Génération de code contre une implémentation.



Intégration continue



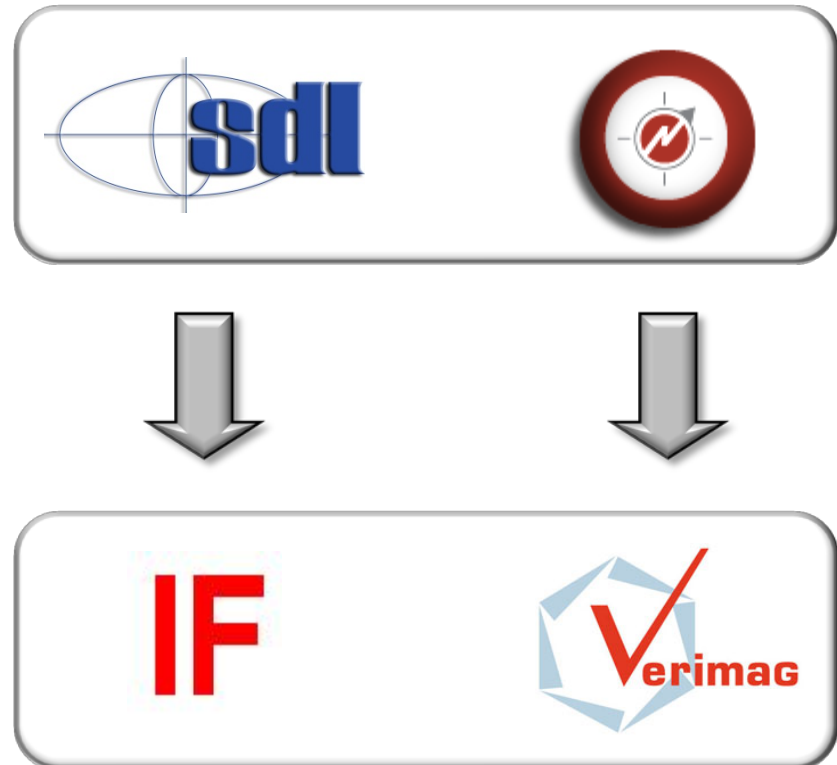
Le vrai sens du développement



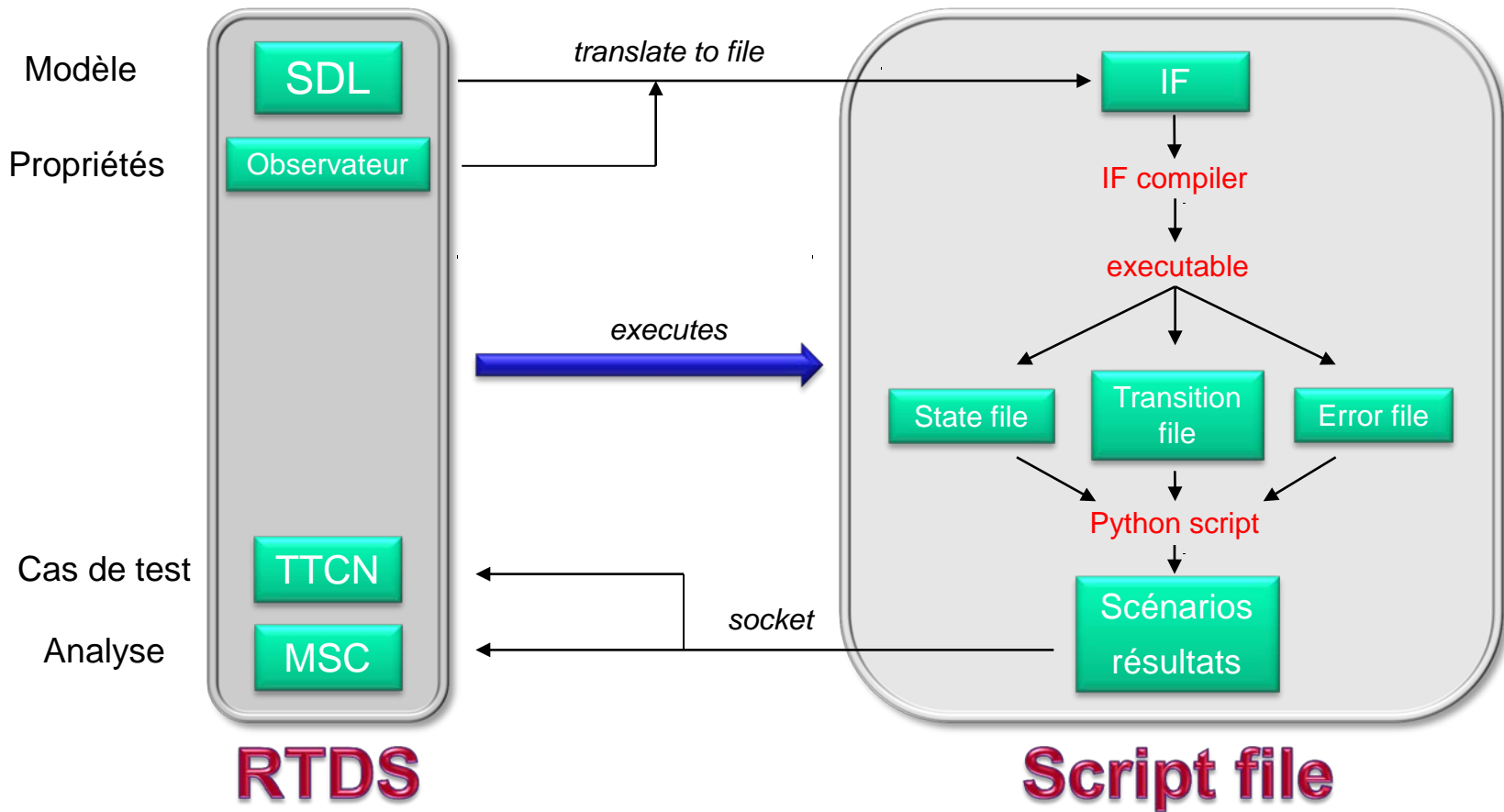
Vérification de propriétés

Exoticus

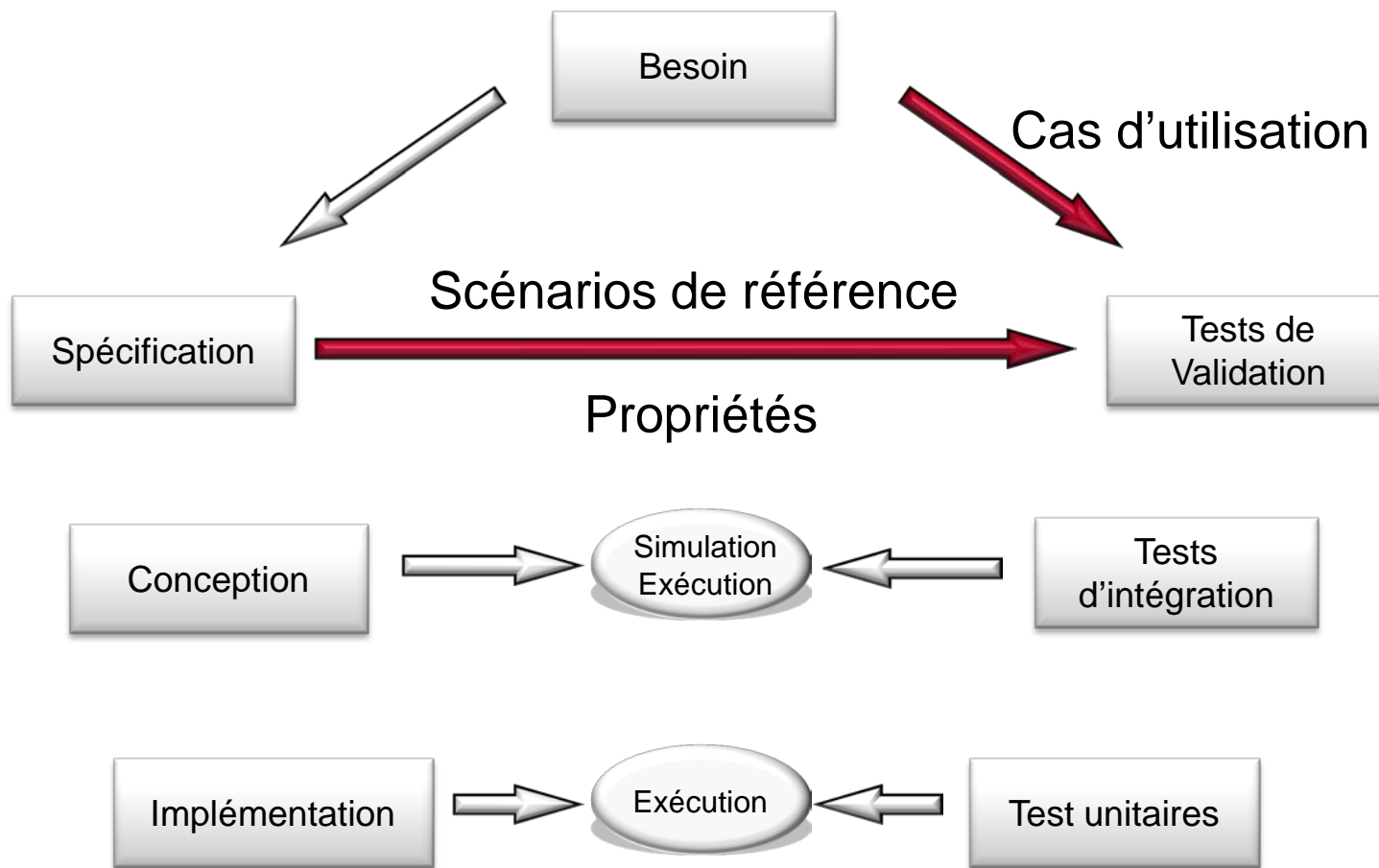
- Partenariat avec Verimag sur la technologie IF.
 - Simulation exhaustive,
 - Observateurs:
 - Vérification de propriétés,
 - Génération de tests.
- Exemple d'implémentation:
 - Export IF,
 - Exécute un script,
 - Trace MSC,
 - Case de test TTCN.



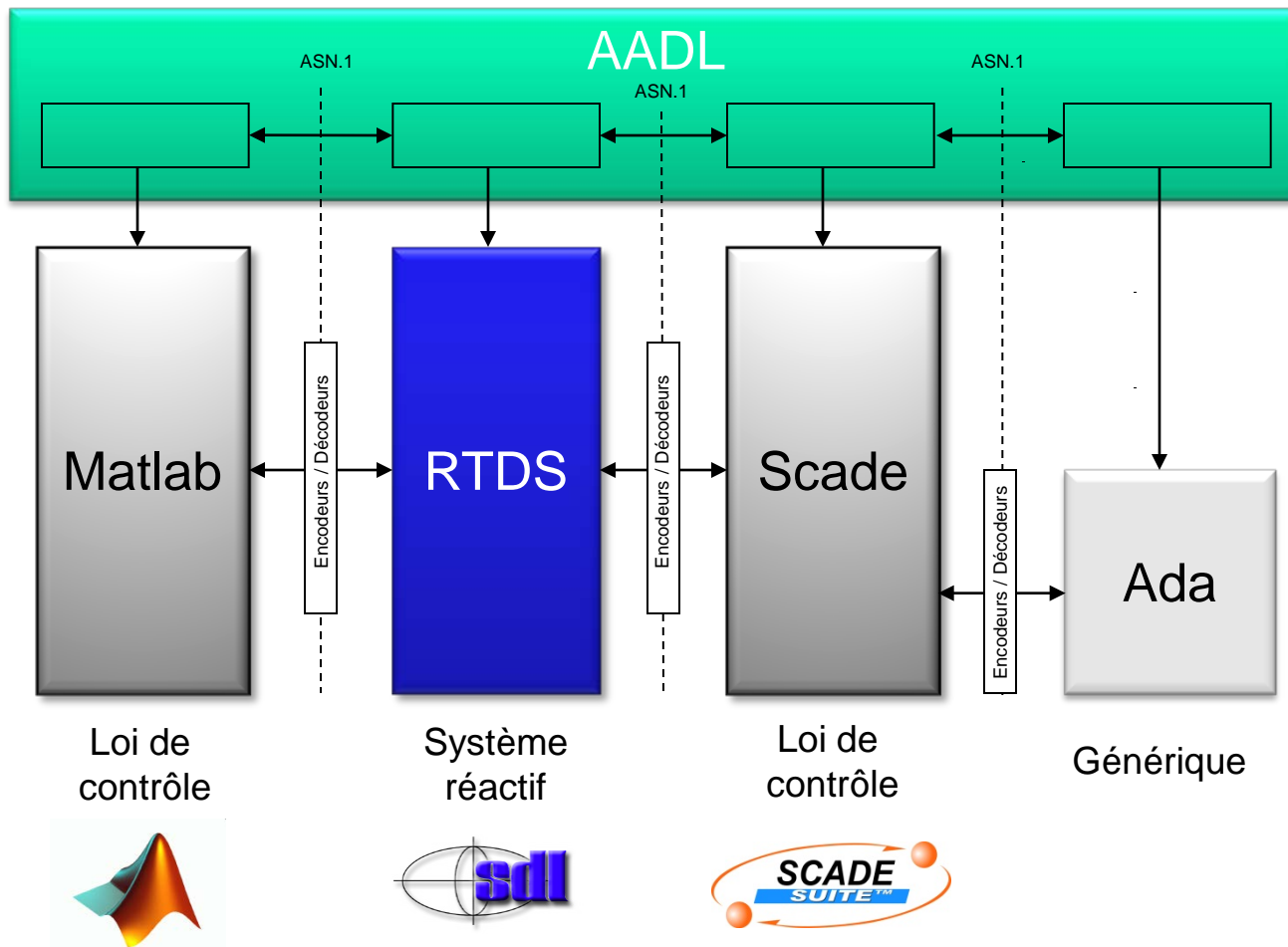
Mise en oeuvre



Des tests de référence



Intégration dans le framework de l'ESA



taste
The Assert Set of Tools for Engineering

esa