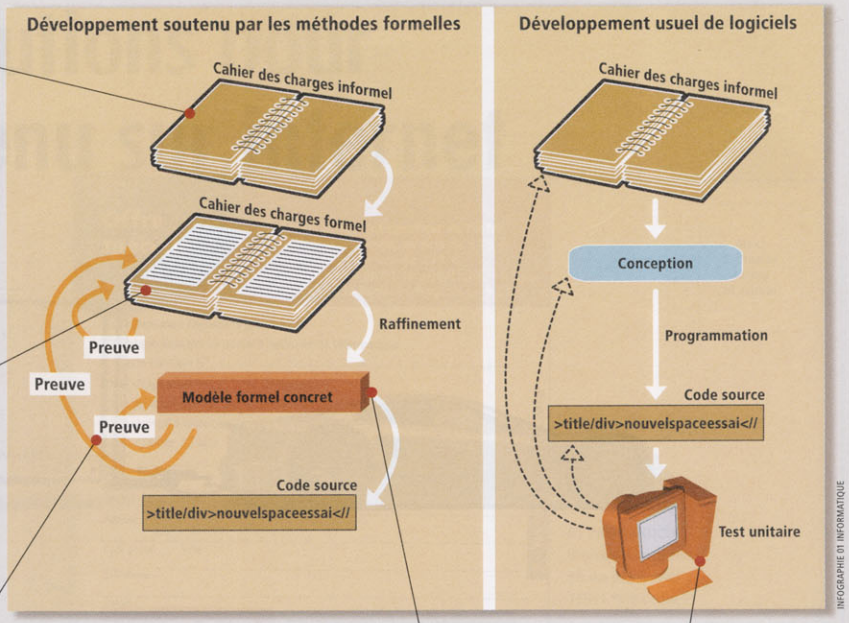
Les méthodes formelles

Une révision complète du cycle de développement logiciel traditionnel permet de se dispenser des tests unitaires.

Qui n'a jamais pesté contre une application qui « plante » ou qui fait le contraire de ce qu'on lui demande ! Les méthodes formelles apportent une réponse à ce type de problème. De fait, elles connaissent déjà un succès croissant dans le monde des logiciels embarqués qui ne tolèrent aucun dysfonctionnement. Le programme sécurité des passagers du système de pilotage automatique de la ligne 14 du métro parisien a ainsi été développé à l'aide de la méthode B. Depuis ses premiers essais, en 1997, ses systèmes de sécurité – soit quatre-vingt-six mille lignes de code, dont la conformité vis-à-vis des spécifications d'origine a été prouvée – n'ont connu aucune défaillance. La preuve intervient à chaque étape du cycle de développement logiciel et en garantit l'uniformité. A tel point que les tests unitaires deviennent inutiles. Les méthodes formelles nécessitent cependant une plus grande rigueur dans l'élaboration des spécifications. **Jean-Marie Portal**

La preuve remplace les tests unitaires



INFOGRAPHIE 01 INFORMATIQUE

Le cahier des charges informel
Rédigé en langage naturel, il définit les exigences que doit réunir le produit logiciel, conformément aux choix de conception effectués au niveau du système, du sous-système et de l'équipement. Il formalise l'ensemble des fonctionnalités attendues, ainsi que les contraintes opérationnelles et de mise en place. C'est-à-dire la façon dont le logiciel doit s'intégrer dans le système et interagir avec.

Le cahier des charges formel
C'est une traduction en langage abstrait du cahier des charges informel des besoins. Sa sémantique repose, entre autres, sur des ensembles, des propriétés et des opérateurs mathématiques. Le modèle obtenu reprend toutes les fonctionnalités du logiciel, en se souciant le moins possible de la façon dont il sera mis en application par la suite.

La preuve
C'est l'élément essentiel des méthodes formelles. Elle garantit l'uniformité interne des spécifications formelles, ainsi que celle entre les différentes phases du développement formel. L'activité de preuve, réalisée avec des outils automatiques, est considérée comme une seconde chaîne de validation, indépendante de l'activité de conception des modèles.

Le modèle formel concret
C'est l'aboutissement de l'étape de raffinement, qui consiste à transformer les spécifications formelles en un modèle concret. Celui-ci prend désormais en compte les contraintes de codage. C'est-à-dire les caractéristiques propres au développement d'un logiciel. Il est écrit en pseudo-code, qui sera traduit, ensuite, dans un langage de programmation.

Les tests unitaires
Intervenant en fin de cycle de développement, ils consistent à vérifier le bon fonctionnement de chaque processus élémentaire – fonctions, procédures, etc. – au sein du produit logiciel final. L'utilisation des méthodes formelles avec preuves, issues du raffinement par étapes, permet d'éliminer les tests unitaires.

POUR/CONTRE

- ✪ Le niveau de fiabilité logicielle résultant de l'emploi des méthodes formelles n'a jamais été atteint avec les méthodes de développement traditionnelles.
- ✪ La suppression des tests unitaires liée à l'utilisation de la preuve allège considérablement le cycle de développement.
- ⛔ Le surcoût, qui sera néanmoins compensé par la réutilisation des méthodes formelles sur d'autres projets.

POUR EN SAVOIR PLUS

www.atelierb.societe.com/LETTRE_B
La lettre B donne la parole aux utilisateurs de la méthode B.

pvs.csl.sri.com/
Une partie du site du laboratoire de sciences SRI International consacrée à PVS.

Introduction aux méthodes formelles, un livre de Jean-François Monin ; Hermes Science Publications ; 2000.

L'OFFRE

MÉTHODES, LANGAGES ET ASSISTANTS DE PREUVE

Nom	Commentaires
B	Cette méthode a été mise au point par Jean-Raymond Abrial à partir du langage Z. Elle repose sur la théorie des ensembles et permet le développement de logiciels en utilisant le mécanisme de raffinement et la preuve. Elle encadre le processus de développement complet (modèle abstrait), de la spécification à l'implémentation (modèle concret). En cela, c'est la méthode la plus complète. Elle est déjà très présente dans le monde du transport ferroviaire et automobile.
COQ	Cet assistant de preuve, assimilable à une méthode formelle, a été développé par l'Inria (Institut national de recherche en informatique et en automatique) de Rocquencourt. Il repose sur une logique permettant l'extraction de programmes et la preuve de théorèmes complexes. Il est utilisé, entre autres, pour le développement de logiciels pour cartes à puce.
HOL	Cet assistant de preuve est employé pour soutenir le raisonnement formel et pour prouver des théorèmes. Il est disponible gratuitement.
PVS	Cet assistant de preuve, développé par un institut de recherche américain (SRI International), est distribué gratuitement dans le cadre d'une exploitation non commerciale.
UML (Unified modeling language)	Ce langage de modélisation est considéré comme semi-formel, car il ne dispose d'aucune sémantique. Il ne repose d'ailleurs pas sur les mathématiques.
Z	Ce langage permet de décrire formellement le fonctionnement d'un programme. Il repose sur la théorie des ensembles, enrichie de types et de notations (appelées schémas) pour structurer les spécifications. Il est à l'origine de la méthode B.

DEMAIN

Une application en amont du développement

Circonscrire l'utilisation des méthodes formelles au seul développement logiciel serait un tort. *« L'intérêt, c'est de monter le plus haut possible dans la spécification »*, affirme d'ailleurs Thierry Servat, directeur général de Clearsy, une société de services spécialisée dans l'utilisation de la méthode B. *« Il faut considérer le système comme un tout, et ne pas essayer de séparer le logiciel du matériel. »* Ce qui reviendrait à appliquer les méthodes formelles en amont de la partie logicielle. Elle réduirait les tests d'intégration en garantissant une meilleure adéquation logiciel/matériel. D'ici là, la première étape consistera à y avoir recours lors de tout développement logiciel. Alors, à quand la fin des écrans bleus dans certains systèmes d'exploitation ?