

Avant-propos

Je faisais depuis quelque temps des expériences d'utilisation de méthodes algébriques de spécification, dans ce qui s'appelait à l'époque Bull Systèmes, lorsque j'ai découvert en 1985 ou 1986, à l'occasion d'une formation donnée par Jean-Raymond Abrial, ce qui quelques années plus tard deviendrait le langage et la méthode B. Et ce fut un véritable coup de foudre !

Tout n'était pas en place, loin s'en faut. Les concepts de base, oui : théorie des ensembles, substitutions généralisées de base, machine abstraite – ou plutôt variables d'état définies par un invariant et modifiées uniquement par des opérations, raffinement de données et algorithmique et obligations de preuve. Mais ils n'étaient pas sous leur forme actuelle. Il n'y avait pas, ou peu, de « sucre syntaxique », pas de composants bien différenciés et pas de composition de composants. Et comme outil d'aide au développement il y avait le B-Tool, un outil d'aide à la démonstration de théorèmes interactif développé par J.-R. Abrial, rustique, mais facile d'utilisation et déjà programmable avec un langage de théories, proche de celui que nous connaissons aujourd'hui.

J'ai vainement essayé, en développant des exemples et quelques outils avec le B-Tool, de convaincre les responsables, et mes propres collègues, que, de toutes les méthodes formelles que nous examinions, B, malgré sa jeunesse, était la plus adaptée au développement de programmes. Les arguments « industriels » l'ont emporté et c'est VDM qui a été retenue. C'était en 1988.

J'ai retrouvé B deux ans plus tard, tout à fait par hasard, à Alstom Transport Signalisation. Alors que j'avais été engagé pour m'occuper de techniques de codage des programmes, j'ai intégré en arrivant l'équipe de Alstom qui collaborait avec J.-R. Abrial, British Petroleum, la RATP et la SNCF à l'industrialisation de B !

Au cours de ces deux années B avait bien changé. Les composants MACHINE, REFINEMENT et IMPLEMENTATION, avec les clauses VARIABLES, INVARIANT, INITIALISATION et OPERATIONS, avaient fait leur apparition, ainsi qu'un composant appelé CONTEXT contenant uniquement des ensembles et des constantes. Les clauses de construction modulaire, INCLUDES et IMPORTS, avaient fait leur apparition aussi. Et un outil d'analyse des sources B et de génération des obligations de preuve avait été développé par J.-R. Abrial et British Petroleum en utilisant le B-Tool.

Les quatre années suivantes ont été particulièrement enrichissantes. D'une part, nous développons avec B les logiciels de sécurité de plusieurs systèmes opérationnels, avec les obligations de résultat que cela comporte, et nous découvrons en temps réel les joies et les peines imposées par la méthode, le langage et les outils. Et, d'autre part, poussés par les besoins des applications, nous étudions ou proposons des évolutions du langage. Le tout en étroite collaboration avec J.-R. Abrial.

Au cours de cette période ont été introduits ou supprimés un certain nombre de traits du langage : les machines avec paramètres, les clauses SEES et ASSERTIONS, les constantes abstraites et les variables concrètes ont été introduites ; les composants CONTEXT ont été supprimés et leurs clauses SETS, CONSTANTS et PROPERTIES ont été intégrées aux machines abstraites. D'autres évolutions du langage, je pense notamment à la clause USES, ont été demandées par d'autres utilisateurs et introduites. Le langage B tel qu'il est présenté dans le *B-Book* a été conçu en partie au cours de cette période.

C'est également au cours de cette période, en 1991 précisément, que, face aux incertitudes qui pesaient sur l'avenir du partenaire qui développait les outils et pour répondre aux besoins de ses applications, Alstom a décidé de développer ses propres outils. Un contrôleur de types, un générateur d'obligations preuve, des démonstrateurs de théorèmes automatique et interactif, des traducteurs de B en Ada et C, des outils de références croisées et de formatage des sources B, des composants réutilisables, une interface graphique ont été ainsi réalisés en l'espace de deux ans et demi.

Grâce à ces travaux, Alstom a pu livrer dès 1993 aux métros de Calcutta et du Caire, à la SNCF et à la RATP les premiers systèmes de contrôle de vitesse de trains intégrant des logiciels développés avec B.

Mais Alstom n'était pas la seule entreprise à s'intéresser à la méthode B. Matra Transport, qui avait obtenu fin 1990 le marché des automatismes d'aide à la conduite de la ligne Météor, se préparait activement à appliquer la méthode B. Son équipe B développait des études de cas, des guides méthodologiques, des outils de

preuve, des études théoriques sur la bonne définition des expressions du langage et participait aux discussions sur ses évolutions.

La RATP, toujours pour les besoins de Météor, cherchait avec le concours de J.-R. Abrial un partenaire industriel pour pérenniser la méthode et les outils. Après quelques péripéties, le choix s'est porté sur la société Digilog, aujourd'hui ClearSy après avoir été Steria, à qui Alstom a transféré en 1994 son atelier de développement et qui le développe et commercialise sous le nom d'Atelier B.

Le développement en B des logiciels de sécurité de Météor et l'impulsion et les financements de la RATP et de la SNCF ont permis à ClearSy de consolider et d'améliorer considérablement l'Atelier B. Cependant, fondamentalement, ils n'ont pas eu une influence notable ni sur l'évolution du langage et ni sur celle des outils.

Les évolutions les plus récentes du langage sont le fruit de deux nouveaux types d'applications de B, la rétro-ingénierie de logiciels et la spécification de systèmes.

La rétro-ingénierie avec B, *i. e.* la construction *a posteriori* d'un modèle B d'un logiciel existant, est née des besoins des industriels de vérifier ou de rénover des logiciels de sécurité développés classiquement et réutilisés à plusieurs reprises. Les premières expériences de rétro-ingénierie ont montré la nécessité d'introduire dans le langage les articles, « records » en anglais, et les opérations locales, à l'image de ce qui existe dans les langages de programmation conventionnels. C'est aujourd'hui chose faite, puisque la version 3.6 de l'Atelier B intègre ces deux concepts.

La spécification de systèmes avec B est née, elle, de deux constatations. La première est que c'est seulement au niveau global, au niveau système, que certaines propriétés fondamentales peuvent être formulées. C'est le cas par exemple des propriétés d'anti-collision des systèmes de protection des trains qui sont composés d'au moins trois sous-systèmes dont chacun assure une partie seulement de la sécurité. La deuxième constatation est qu'il est tout à fait possible de développer, même formellement, des logiciels corrects qui ne remplissent pas leur fonction parce que le système dont ils font partie n'a pas été rigoureusement défini.

Le langage B-événementiel proposé par J.-R. Abrial pour la spécification de systèmes constitue une petite révolution par rapport au langage B classique. Pas du point de vue des fondements mathématiques du langage qui restent inchangés, mais du point de vue de l'objet modélisé qui de passif devient réactif. Les premières études de cas menées avec le B-événementiel sont très prometteuses. Elles montrent sa capacité à modéliser abstraitement les propriétés et le comportement de systèmes relativement complexes et, plus inattendu, son utilité pour le développement d'algorithmes séquentiels.

Je conclurai par trois remarques personnelles sur le bilan et des perspectives de B. Premièrement, je crois réellement que le langage B classique, quoique perfectible, a atteint sa maturité industrielle dans la mesure où il satisfait aux critères exigés habituellement par les industriels lorsqu'ils sélectionnent une technologie : des filières de formation existent ; un outil commercial permettant la réalisation d'applications industrielles est disponible ; de donneurs d'ordre et des industriels de premier plan l'utilisent avec succès ; enfin, de plus en plus de sociétés de services, indépendantes des éditeurs d'outils, maîtrisent la technologie. Je considère même que la méthode B est actuellement la seule à offrir les outils théoriques et pratiques permettant de démontrer effectivement des programmes dans un contexte industriel.

Deuxièmement, l'introduction de B dans une entreprise, malgré sa maturité, reste une action délicate dont le succès dépend de facteurs en grande partie indépendants de la technologie. Avec le recul, je dirais que les facteurs de réussite sont, par ordre d'importance : l'engagement et le soutien sans faille des responsables, un objectif d'application opérationnelle ferme et clairement défini, une période d'apprentissage et de préparation suffisantes avant l'application opérationnelle, la maturité de l'entreprise en matière de développement logiciel, enfin le conseil d'experts internes ou externes à l'entreprise pendant la période d'apprentissage et de préparation. Ainsi, on peut être à peu près certain que l'utilisation de la technologie, sans aucune application réelle en vue, par un stagiaire débutant ne connaissant pas non plus le sujet sur lequel il doit l'appliquer et sans aucune aide d'experts – situation que l'on rencontre souvent, hélas, dans l'industrie – conduit directement à l'échec.

Troisièmement, le langage B est bien vivant, il évolue, il s'améliore. J'ai déjà parlé de B-événementiel. D'autres évolutions proposées semblent utiles et intéressantes : le raffinement des paramètres des opérations pour pouvoir abstraire davantage les spécifications ; les modules partagés pour permettre la programmation de logiciels distribués implantant des systèmes spécifiés avec B-événementiel. D'autres évolutions peuvent être imaginées, par exemple, la promotion des machines en « classes ». La validation de toutes ces évolutions nécessite le développement d'études de cas et d'études théoriques. Des équipes de recherche en France, Grande-Bretagne, Australie, Finlande travaillent activement sur tous ces sujets.

Comme on le voit, B est un monde foisonnant. Il est donc tout à fait bienvenu qu'un livre fasse le point sur la méthode B. Qu'il précise certains points de la sémantique du langage ; qu'il y dégage l'essentiel de l'accessoire ; qu'il explique et illustre les processus de construction d'applications industrielles ; qu'il présente les évolutions ses plus intéressantes.

Je me félicite que ce soient Henri Habrias, Jean-Yves Lafaye et Marie-Laure Potet qui aient eu la volonté et la ténacité nécessaires à mener ce livre à terme. Leur expérience des études théoriques et de l'enseignement de la méthode B garantit la qualité scientifique et pédagogique de l'ouvrage. Je suis certain que tous les lecteurs, comme moi-même, tireront le plus grand profit de sa lecture.

Fernando Mejia

Responsable Méthodes et Outils de Développement Logiciel
Alstom Transport Signalisation