Un environnement pour le prototypage rapide du langage B : ABTools

Jean-Louis BOULANGER
Université de Technologie de Compiègne
HEUDIASYC
Centre de Recherches de Royallieu
BP 20529
60205 Compiègne cedex

Keywords : ABTools, ANTLR, Méthode Formelle, Méthode B, B événementiel, Outils.

Résumé étendu

Another's B Tools

Le but de ce papier est de montrer comment l'environnement de développement B intitulé **Another's B Tools** (noté *ABTools*[1] voir [3]), qui est une suite d'outils pour la méthode B, peut-être utilisé pour prototyper rapidement une extension du langage B. Cette rapidité de prototypage est dû à l'utilisation de l'environnement de génération de compilateur *ANTLR*[2].

ANTLR est un environnement de génération de compilateur. Il permet de couvrir l'ensemble des phases de compilation : lexing, parsing, typing et génération de code (JAVA, C, ..) et de génération de documentation (TEX, HTML, XML, ...).

Le caractère objet de l'environnement *ANTLR* garantit une grande flexibilité. Etant donné que nous utilisons JAVA comme langage support pour cette application, cet environnement open source est accessible sur différentes plateformes (Linux, Windows NT, ... et Windows XP).

Le but de ce travail n'est pas de concurrencer les ateliers commerciaux tels que l'*Atelier B* de la société Clearsy ou le *BTool* de la société BCore, mais bien de mettre à la disposition de la communauté des outils *open source* permettant d'étudier le langage B et ses extensions.

[1] http ://www.chez.com/abtools
[2] http ://www.antlr.org

Prototypage d'extension

Plusieurs travaux présentent des extensions intéressantes de la méthode *B*. Afin de valider celles-ci, il est intéressant de pouvoir rapidement et facilement les implanter.

L'utilisation d'ANTLR permet d'atteindre facilement cet objectif car les grammaires peuvent être étendues par héritage. Les notions objets, telles que l'héritage et/ou la surcharge, permettent de mettre en place une extension de la méthode *B* tout en contrôlant la cohérence de l'ensemble. Toute nouvelle grammaire peut être considérée comme une extension (par héritage et/ou surcharge) d'une grammaire déjà validée.

B Prime Comme premier exemple d'évolutivité, nous avons implanté une extension nommée *B Prime* qui reprend certaines évolutions que nous avions proposées dans [4]. Cette version du langage *B* permet de typer les variables lors de leur déclaration. La mise en place de cette extension au sein de notre environnement, se traduit par la mise en place d'une nouvelle grammaire nommée *BPrime.g*. Cette grammaire hérite de la grammaire décrite précédemment et surcharge certaines règles.

Il faut reporter cette évolution aux seins des autres outils (décompilateur et vérificateur de type). Pour les outils manipulant les arbres syntaxiques (treewalker), il n'est pas utile de toujours séparer les grammaires, en effet ces outils supposent que les arbres abstraits sont corrects.

En fait, le langage *BPrime* étend cette notion de typage implicite aux clauses *CONSTANTS* (abstaites ou non), aux profils d'opérations et aux substitutions *ANY*, *VAR* et *WHILE*.

Le langage *BPrime* introduit deux autres évolutions :
- La prise en compte de commentaire "à la C++",
- La prise en compte du caractère "," comme séparateur d'ensemble dans la clause *SETS*.

B système Dans [2], Jean-Raymond Abrial a montré qu'il était possible d'utiliser la méthode *B* pour spécifier des systèmes distribués. Mais c'est dans [1] qu'il propose une véritable extension de la méthode *B* à la notion d'événement et de contraintes dynamiques.

Cette extension introduit de nouvelles clauses (*SYSTEM*, *EVENTS*, *DYNAMICS*, *VARIANT* et *MODALITIES*), de nouvelles substitutions et un nouveau type de prédicat.

B événementiel Une seconde extension orientée événement a été introduite dans le cadre du projet MATISSE[3]. Cette extension est formalisée sous forme d'un manuel de référence (voir [5]) et d'un manuel utilisateur (voir

[3]http ://www.xxx.xxx

[6]). Cette extension reprend plusieurs évolutions préconisées dans l'extension précédente, qui devient très certainement obsolète. Une version corrigée de la grammaire a été mise en place au sein de l'environnement *ABTools*, elle est associée à un décompilateur (ASCII, TEX). Le typeur de base a été étendu afin d'être applicable.

Petite conclusion

La réalisation de notre environnement *ABTools* a permis de montrer la puissance de l'environnement de génération de compilateur nommé *ANTLR*. Nous avons pu dans un premier temps réaliser une version de notre environnement (parser, typeur, decompilateur, gop, ...) B et dans un deuxième temps nous avons pu tester certaines extensions du langage B .

Aucune difficulté majeure n'a été rencontrée et l'utilisation de l'environnement *ANTLR* pour la construction de notre environnement B semble un succès.

Le principal intéret de cette approche, de la modélisation du langage B , réside dans la capacité d'expression du langage de règle utilisé par *ANTLR* et par ces caractéristiques objets. La définition d'une nouvelle grammaire est réalisée au travers de l'héritage d'une grammaire déjà validée, de l'extension et/ou de la surcharge.

References

[1] J-R Abrial and L Mussat. Introducig dynamic constraints in b.

[2] Jean-Raymond Abrial. Extending B without changing it (for developing distributed systems). In Henri Habrias, editor, *Proceedingsof 1st Conference on the B method*, Putting into Practice methods and tools for information system design, pages 169–191, 3 rue du Maréchal Joffre, BP 34103, 44041 Nantes Cedex 1, November 1996. IRIN Institut de recherche en informatique de Nantes.

[3] Boulanger Jean-Louis. Abtools, une suite d'outil pour la méthode b développé avec antlr. In *Journées «Outils pour et autour de la méthode B», 15 et 16 octobre 2001.*

[4] Boulanger Jean-Louis, Mariano George, and Tatibouet Bruno. Revisiting b language syntax. Technical Report 99-07, CNAM Laboratoire CEDRIC, 1999.

[5] MATISSE. Event b reference manual. Technical report, Methodologies and Technologies for Industrial Strength Systems Engineering, 2001.

[6] MATISSE. Event b to b translator user manual. Technical report, Methodologies and Technologies for Industrial Strength Systems Engineering, 2001.