

The B Method

Rob Tiemens

1 Introductie

2 Uitleg

- Abstract Machine Notation
- Generalised Substitution Language
- Refinement
- Proof obligations

3 Taal

4 Demo

5 Bronnen

The B Method - Introductie

Wat is B - Method

"a proven construction approach based on B Language, refinement and proof."

- Een formele methode
 - Specificaties gebaseerd op wiskundige formalisme
 - Eerste orde logica
 - Groepentheorie
- Met refinement
 - Structureer het bewijs zodanig dat er code uit gegenereerd kan worden.
- Bewijs
 - Door de wiskundige modellen is er bewijs dat het model consistent is.
- B language (programmeertaal)
- Tools

The B Method - Introductie

Wat is B - Method

Bedacht door

- Jean-Raymond Abrial
- Andere belangrijke mensen: G. Laffite, F. Mejia, I. McNeal

The B Method - Introductie

Wat is B - Method

Onderhouden door door

- ClearSy
- Oxford University (Programming Research Group)

The B Method - Introductie

Wat is B - Method

Voorgangers, nakomelingen

- Z-notation
- Event-B
 - Makkelijkere notatie
 - Bosch: cruise control system
 - Siemens Transportation: Trein en signalen

The B Method - Introductie

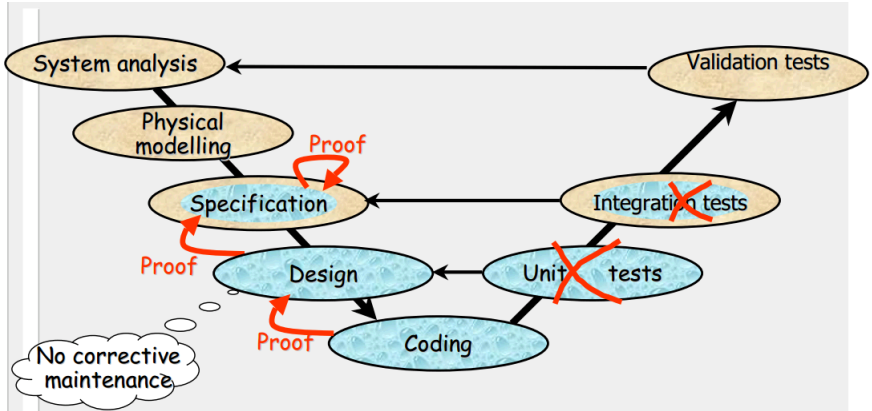
Wat is B - Method

B method is gebruikt voor

- Peugeot
 - Airbags
 - Motormanagement
- Paris Métro Line 14
- Ariane 5

The B Method - Introductie

Wat is B - Method



The B Method - Uitleg

- Abstract Machine Notation
- Generalised Substitution Language
- Refinement

The B Method - Uitleg

Abstract Machine Notation

- State based modellen
- Type checking
- Obligations
- Consistentie

The B Method - Uitleg

Abstract Machine Notation

- State variables
- Invariant
 - beperking en relatie voor state variabelen.
- Operaties op de state variables
- Kan geparametriseerd worden.

The B Method - Uitleg

Generalised Substitution Language

- Postconditie R
- Predikaat transformer S
- Weakest precondition $[S]R$

The B Method - Uitleg

Generalised Substitution Language

Vormen van substitutie

- $x := E$

Waarbij E een set theoretische expressie is

- $x_1, \dots, x_n = E_1, \dots, E_n$

Waarbij E_n een set theoretische expressie is

- $[skip]R \Leftrightarrow R$

Geen substitutie

- $[P \mid S]R \Leftrightarrow P \wedge [S]R$

Pre conditie

- $[P \Rightarrow S]R \Leftrightarrow P \Rightarrow [S]R$

Guarding $P \Rightarrow S$

The B Method - Uitleg

Generalised Substitution Language

Dus eigenlijk je programmeertaal.

BEGIN	block substitution
skip	identity substitution
$:=$	becomes equal to substitution
$:()$	becomes such that substitution
$:\in$	becomes element of substitution
IF	IF Conditional Substitution
SELECT	conditional bounded choice substitution
CASE	condition by case substitution
ANY	unbounded choice substitution
LET	local definition substitution
VAR	local variable substitution
;	sequencing substitution
WHILE	while loop substitution
$\leftarrow$	operation call substitution

The B Method - Uitleg

Refinement

- Verifieerbare transformatie van abstracte formele specificatie.
- Naar concrete (low level) code

The B Method - Uitleg

Refinement

Voorbeeld

$$x \in \{1, 2, 3\}$$

kan verfijnt worden naar

$$x \in \{1, 2\}$$

en dan naar

$$x \in \{1\}$$

dit resulteert dan in de code

```
int x = 1;
```

The B Method - Uitleg

Proof obligations

- Kloppen de instanties (type)
- Kloppen de asserties
- Van de initialisatie kan dan de invariant bereikt worden
- Behouden de operaties de invariant.

The B Method - Uitleg

Abstract Machine Notation

C Predikaat in context van Abstract Machine

K Constraint predikaat K

I Invariant

T Initialisatie

S Operatie

P Pre-conditie

The B Method - Uitleg

Abstract Machine Notation

Proof obligations

- $C \wedge K \Rightarrow \exists x. I$
- $C \wedge K \Rightarrow [T]I$
- $C \wedge K \wedge I \wedge P \Rightarrow [S]I$
 - Voor elke operatie S in Machine

$[S]I$ moet gelezen worden als "S estableishes I"

The B Method - Uitleg

Proof obligations

```
increment =  
  begin  
    val := val + 1  
  end
```

Levert Proof obligation

$$val \in \mathbb{N} \Rightarrow [val := val + 1] val \in \mathbb{N}$$

Dit wordt gesubstiueert in

$$val \in \mathbb{N} \Rightarrow val + 1 \in \mathbb{N}$$

En dit mag je nog bewijzen.

The B Method - Uitleg

Proof obligations

MACHINE

EXAM

VARIABLES

xx, yy, zz

INVARIANT

$xx < 0 \wedge$

$yy > 10 \wedge$

$zz \cdot yy \cdot xx$

ASSERTIONS

$zz < 0$;

$zz \cdot xx > 0$

END

$xx < 0 \wedge$ “Machine invariant”

$yy > 10 \wedge$

$zz \cdot yy \cdot xx$

$\Rightarrow$

$zz < 0$ “First assertion for the machine”

$xx < 0 \wedge$ “Machine invariant”

$yy > 10 \wedge$

$zz \cdot yy \cdot xx \wedge$

$zz < 0$

“First assertion for the machine”

$\Rightarrow$

$zz \cdot xx > 0$ “Second assertion for the machine”

Het concept van type is gebaseerd op sets

*Laat E een expressie zijn en S en T sets zodanig dat $S \subseteq T$
Als $E \in S$ dan $E \in T$. De grootste set die E bevat is dan
het type.*

The B Method - Taal

Types

- Type van expressie 3 is $\mathbb{Z}$
- Type van expressie $\{-5, 3, -1, 8\}$ is $\mathbb{P}(\mathbb{Z})$
- Type van expressie $\text{rec}(a : 5, b : \text{TRUE})$ is $\text{struct}(a : \mathbb{Z}, b : \text{BOOL})$

The B Method - Taal

Types

- $VarRaf1 \in INT = \mathbb{P}(\mathbb{Z})$
- $VarRaf2 \subseteq NAT = \mathbb{P}(\mathbb{Z})$

Arrays zijn te definiëren als totale functie.

- $Tab1 \in (0..4) \rightarrow INT$

Nog meer types mogelijk

The B Method - Taal

Commando's

- **ah(P)** : Add hypothesis
 - Begin $h_1, \dots, h_n \Rightarrow G$
 - Eerst bewijzen $h_1, \dots, h_n \Rightarrow P$
 - Dan mag je $h_1, \dots, h_n, P \Rightarrow G$ bewijzen
- **cg** : Current goal
 - Toont huidige doel
- **ct** : Contradiction
 - Als G goal is
 - dan $\neg G \Rightarrow \text{bfalse}$ nieuw doel

The B Method - Taal

Commando's

- **ah(P)** : Add hypothesis
 - Begin $h_1, \dots, h_n \Rightarrow G$
 - Eerst bewijzen $h_1, \dots, h_n \Rightarrow P$
 - Dan mag je $h_1, \dots, h_n, P \Rightarrow G$ bewijzen
- **cg** : Current goal
 - Toont huidige doel
- **ct** : Contradiction
 - Als G goal is
 - dan $\neg G \Rightarrow \text{bfalse}$ nieuw doel

The B Method - Taal

Commando's

- **dc(X) : Do Case**

$$X \Rightarrow P$$

$$\neg X \Rightarrow P$$

- **dd : Direct Deduction**

$$P \Rightarrow Q$$

- **eh : Use equality in hypothesis**

Herschrijf naar iets in de hypothese

- **pr: Prove**

Automatic prover (Hyp)

- **pp : Predicate prover**

Automatic prover (Resolutie)

- **sh : Search hypothesis**

Voro als je de weg kwijt bent.

- **ss: Simplify Set**

The B Method - Demo

Demo

The B Method - Bronnen

Belangrijkste Bronnen

- Demo

- [http://gergo.erd.hu/blog/2010-02-16-the_b_method_for_programmers_\(part_1\)](http://gergo.erd.hu/blog/2010-02-16-the_b_method_for_programmers_(part_1))

- J.-R. Abrial, M.K. O.Lee, D.S. Neilson P.N.Scharlach. The B-Method. BP International Ltd 1991, september 1991

- D. Cansell, D. Meery, FOUNDATIONS OF THE B METHOD, August 7, 2003

- Atelier B, Interactive Prover, Reference Manual, version 3.7, Clearsy

- Atelier B, Proof Obligations, Reference Manual, version 3.7, Clearsy

- Industrial Use of THE B METHOD, Clearsy

- www.bmethod.com

The B Method - Einde

Vragen

Vragen.