The B-Method

Serge MBITOM

Radboud University Nijmegen

June 12th 2009

Definition

Definition

B is a formal specification method which, thanks to an adequate language, allows for highly accurate expressions of the properties required by specifications.
One can then **prove in a fully automated fashion** that these properties are unambiguous, coherent and are not contradictory.
This then allows us to mathematically prove that these properties are taken into account as the design stages progress.

Definition

B-Method usually refers to the set that includes: B language, refinement, proof and related tools.

Definition

B-Method usually refers to the set that includes: B language, refinement, proof and related tools.

- The B method is based on the results of:
  - Dijkstras weakest precondition calculus
  - The set-theoretical notation of the Z specification language
  - C.A.R. Hoares and C. Jones developments on refinement
  - Z and VDM as specification and design languages

Definition

B-Method usually refers to the set that includes: B language, refinement, proof and related tools.

- The B method is based on the results of:
  - Dijkstras weakest precondition calculus
  - The set-theoretical notation of the Z specification language
  - C.A.R. Hoares and C. Jones developments on refinement
  - Z and VDM as specification and design languages
- Developed in 85 - 88 at Oxford University Programming Research Group

Definition

B-Method usually refers to the set that includes: B language, refinement, proof and related tools.

- The B method is based on the results of:
    - Dijkstras weakest precondition calculus
    - The set-theoretical notation of the Z specification language
    - C.A.R. Hoares and C. Jones developments on refinement
    - Z and VDM as specification and design languages
- Developed in 85 - 88 at Oxford University Programming Research Group
- The main investigator is Jean-Raymond Abrial, with contributions from D. Gries, J. Prinz, C.C. Morgan, P. Gardiner, I.H. Sorensen and others

Definition

B-Method usually refers to the set that includes: B language, refinement, proof and related tools.

- The B method is based on the results of:
    - Dijkstras weakest precondition calculus
    - The set-theoretical notation of the Z specification language
    - C.A.R. Hoares and C. Jones developments on refinement
    - Z and VDM as specification and design languages
- Developed in 85 - 88 at Oxford University Programming Research Group
- The main investigator is Jean-Raymond Abrial, with contributions from D. Gries, J. Prinz, C.C. Morgan, P. Gardiner, I.H. Sorensen and others
- 1988-1994: development of the B-Tool and B-Toolkit.

Main components

Main components

- First-order logic (similar to Z)

Main components

- First-order logic (similar to Z)
- Set theory (similar to Z)

Main components

- First-order logic (similar to Z)
- Set theory (similar to Z)
- Integer arithmetics (similar to Z)

Main components

- First-order logic (similar to Z)
- Set theory (similar to Z)
- Integer arithmetics (similar to Z)
- Generalized substitutions (specific of B)
    - The mean to describe state changes
    - Predicate transformers
    - The substitution [V := V+1] substitutes all occurrences of V with the expression V+1

B for systems

- Goal: help to understand, specify, design, verify a system development
  - not a method to create a system, but to check it
  - requires contact with the system creators to deeply understand the system
- A B-System model formalizes:
  - the system (hardware and software)
  - its environment (other systems, infrastructure, procedures handled by operators)
- Covers functional logical angle of the system, not digital calculus, not real-time requirements

B for softwares

- Goal: to develop a code that complies with its specification and to be sure of it (to know exactly what is proved)
- Covers a subpart of the software with functional logical procedures, only for one task or thread, not low-level Operating System features, no direct input/output

B in education

- Over 100 universities/research labs currently active
- 2000 graduates per year with some experience

B in the industry

- KVB: Alstom
    - Automatic Train Protection for the French railway company (SNCF), installed on over 6,000 trains since 1993
    - 60,000 lines of B; 10,000 proofs; 22 000 lines of Ada
- SAET METEOR: Siemens Transportation Systems
    - Automatic Train Control: new driverless metro line 14 in Paris (RATP), 1998.
    - 3 safety-critical software parts: onboard, section, line
    - 107,000 lines of B; 29,000 proofs; 87,000 lines of Ada
- Roissy VAL: ClearSy (for STS)
    - Section Automatic Pilot: light driverless shuttle for Paris-Roissy airport (ADP), 2006
    - 183,000 lines of B; 43,000 proofs; 158,000 lines of Ada

B in the industry (2)

- EADS
  - Model of tasks scheduling of the software controlling stage separation of Ariane rocket
- Peugeot Automobiles
  - Model of the functioning of subsystems (lightings, airbags, engine etc) for Peugeot aftersales service
  - Goal: Understanding precisely the functioning of cars to build tools to diagnose breakdowns

Specification

- The basic module for specification in B is the machine
- Abstract Machine Notation (AMN)
- The state is defined with variables
- The operations are defined with generalized substitutions
- Valid states need to be explicitly specified with an invariant predicate

MACHINE *Name (Parameters)*
VARIABLES *list of variables*
INVARIANT
invariant predicate
INITIALISATION
initialisation substitution
OPERATIONS
$outputs \longleftarrow name(inputs) \triangleq substitution$
END

Obs. The other clauses provided by the B notation for specification are omitted.

Refinement

- A module specification is refined: it is reexpressed with more information:
    - adding some requirements
    - refining abstract notions with more concrete notions
    - getting to implementable code level
- A refinement must be consistent with its specification (this should be proved)
- A refinement may also be refined

REFINEMENT *Name (Parameters)*
REFINES *machine_ref*
INVARIANT
invariant predicate
INITIALISATION
initialisation
OPERATIONS
operations
END

Obs. The other clauses provided by the B notation for refinement are omitted.

Implementation

- The final refinement is called the implementation
- Proof obligations are generated
- A concrete model is obtained, and can be translated into Ada, C, C++

Available tools

- Atelier B (ClearSy, www.clearsy.com)
  - Current version 4.0
  - Created to develop industrial B-Software projects
  - A set of tools integrated into a project manager tool
    - static checker
    - automatic proof obligation generator
    - automatic provers and interactive prover
    - code translators: Ada, C, C++
- B4free (www.b4free.com)
  - Free but restricted to academic users and owners of Atelier B
  - The core tools of Atelier B + xemacs interface

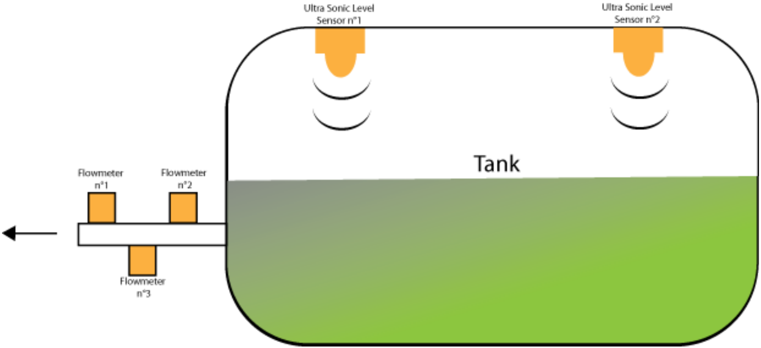
c.154

System specification

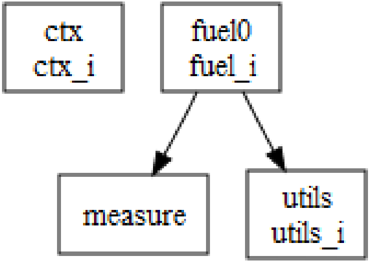
- Control system to determine the position of the switch
- Rendundancy is mandatory because of sensor failures ⇒ 3 sensors
- 3 measures: m1, m2, m3
- The possible values are:
  - normal (right position)
  - reverse (left position)
  - void (middle position while the switch is moving from a position to another)

System specification (2)

- A function to calculate an estimate of the switch position
- If at least, one normal and one reverse are measured, then the result is void
- If all measures are void, then the result is void
- In all other cases, it should return normal or reverse


Ultra Sonic Level Sensor n°1
Ultra Sonic Level Sensor n°2
Tank
Flowmeter n°1
Flowmeter n°2
Flowmeter n°3

System architecture



Conclusion

- B-Method is successfully used in the industry
- No classic programming error in the code (overflow, division by 0, out of range index, infinite loop etc)
- Unit tests are no longer needed
- Program meaning is controlled
  - by proving that the B specification is consistent
  - by proving that the B code complies with its specification

Avez-vous des questions?