Eseményalapú rendszertervezés helyességbizonyítással

Majzik István
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék
majzik@mit.bme.hu

2004

A B-módszer alapjai

- A B-módszer kidolgozása: J.-R. Abrial, 1985-1992, a Z továbbfejlesztése.
- Alapok: E.W. Dijkstra és C.A.R Hoare munkái.
- A teljes elméleti alapozás és a módszer leírása: a *B book*.
- A matematikai modellen alapú szoftverfejlesztési módszerek közé tartozik:
  - ⋆ a VDM módszerhez hasonló,
  - ⋆ könnyebben használható (?),
  - ⋆ hatékony eszköztámogatás.

Hibamentes szoftvertervezés (*zero defect design*)

A B-módszer és a tipikus szoftverfejlesztési folyamat kapcsolata:

- Követelmény-specifikáció:
  A B-módszer a követelmények formalizálásában
  és ezek konzisztenciájának megállapításában segít.

- Architektúratervezés és részletes tervezés:
  A B-módszer a modellfinomítási lépések helyességének vizsgálatával
  egészíti ki ezt a folyamatot.

  ⋆ A kezdeti modell finomítása történik,
  míg el nem jutunk a felhasznált programozási nyelvi konstrukciók
  illetve könyvtári elemek szintjére,
  amikor a közvetlen megvalósítás (kódgenerálás) lehetséges.

A B módszer

- *Jelölésrendszert* és *módszert* ad rendszerek specifikációjához és tervezéséhez.
- A tervezés a kiindulási specifikáció lépésenkénti finomításával történik.
- Az egyes finomítási lépések, illetve az eredményként előálló specifikáció ellenőrzéséhez *matematikai apparátus* áll rendelkezésre (kritériumok és bizonyítási szabályok készlete).
- A használt formalizmus az úgynevezett absztrakt állapotgép (*abstract state machine*).
  - ⋆ Ennek alapja Dijkstra feltételekkel védett parancsnyelve (*guarded command language*).
  - ⋆ Az egyes utasításokhoz elő- és utófeltételek adhatók meg.
  - ⋆ Az adatleíró nyelv pedig a halmazelmélet jól ismert fogalmaira épül.

Az absztrakt állapotgép

Az absztrakt állapotgép elemei:

- a globális kényszerek (*constraints*),
- az előre ismert konstansok (*constants*),
- a tulajdonságok (*properties*),
- az állapot megadására használt változók (*variables*),
- az ezekre vonatkozó invariánsok (*invariants*),
- az inicializálás (*initialization*),
- az események/műveletek leírása (*operations*),
  itt az előfeltételek (*preconditions*) és az eredmények.

Formalizálás

MACHINE $Machine\_name(p)$

CONSTRAINTS $Cst(p)$

CONSTANTS c

PROPERTIES $Ctx(c, p)$

VARIABLES v

INVARIANT $Inv(p, c, v)$

INITIALISATION $Init$

OPERATIONS $Operation\_name$ PRE $Pre(p, c, v)$ THEN St END

END

- Cst, Ctx, Inv és Pre elsőrendű logikai predikátumok,
- p, c, v változók listája, St pedig egy művelet.

A specifikáció konzisztenciájának bizonyítása

- A specifikált környezet (konstansok, formális paraméterek) létezik:

$$\begin{aligned} \exists p &: Cst(p) \\ Cst(p) &\Rightarrow \exists c : Ctx(c,p) \\ Cst(p) \wedge Ctx(p,c) &\Rightarrow \exists v : Inv(p,c,v) \end{aligned}$$

- Az inicializálás biztosítja az invariánsok teljesülését:

$$Cst(p) \wedge Ctx(p,c) \Rightarrow [Init]Inv(p,c,v)$$

- Minden esemény (művelet) megtartja ezeket az invariánsokat:

$$Cst(p) \wedge Ctx(p,c) \wedge Inv(p,c,v) \wedge Pre(p,c,v) \Rightarrow [St]Inv(p,c,v)$$

$[St]R$ jelentése: St végrehajtása teljesíti az R predikátumot.

Műveletek és predikátumok

Az St műveletek és az ekvivalens predikátumok:

Művelet megadása B-ben	Ekvivalens predikátum
[BEGIN S END]R	$[S]R$
[PRE P THEN S END]R	$P \wedge [S]R$
[CHOICE S OR ... OR T END]R	$[S]R \wedge ... \wedge [T]R$
[IF P THEN S ELSE T END]R	$(P \Rightarrow [S]R) \wedge (\neg P \Rightarrow [T]R)$
[IF P THEN S END]R	$(P \Rightarrow [S]R) \wedge (\neg P \Rightarrow R)$
[ANY l WHERE P THEN S END]R	$\forall l : (P \Rightarrow [S]R)$, ha l kötött R-ben
[VAR l IN S END]R	$\forall l : [S]R$, ha l kötött R-ben
$[v := e]R$	R, ahol v helyett e szerepel

- Minden St művelet mint egy predikátum-transzformátor értelmezhető, amely új predikátumot állít elő.

- Ezeket a táblázatban felírt szabályokat helyettesítési szabályoknak is felfoghatjuk, amik axiomatizálhatók.

Bizonyítandó állítások

- A pszeudo-nyelvű leírásokból
  elsőrendű logikai illetve halmazelméleti állítások képezhetők.

- Ezeket a bizonyítandó állításokat a B eszközkészlet automatikusan generálja.

A formalizmus néhány tulajdonságát külön is kiemeljük:

- Az absztrakt állapotgépek paraméterezhetők (lásd p),
  ugyanaz a specifikáció más paraméterrel újra használható.

- A CHOICE helyettesítés a nemdeterminisztikus modellezés lehetőségét adja.

- Az ANY lehetővé teszi általános feltételek használatát.

- A választás $\|$ operátora segítségével kombinálhatók műveletek,
  így összetett helyettesítések adhatók meg.

További szintaktikai elemek a strukturáláshoz

A beágyazott állapotgépek paraméterezhetősége és átnevezhetősége:

- PROMOTES kulcsszó: a beágyazott állapotgép azon műveletei, amelyek (változás nélkül) az összetett állapotgép műveletei lesznek.
- Beágyazás: INCLUDE
- Láthatóság: SEES
  (a csak látható változók invariánsokban sem használhatók fel)
- Használhatóság: USES relációk
  (invariánsokban használható, de nem módosítható a használó állapotgépben).

A B eszközkészlet

- B-Tool (1991, BP), *Atelier B Tools* (Steria Mediterranee, 1994), *B Toolkit* (B-Core Ltd.)
  - ⋆ Szintaxis- és típusellenőrzés (Analyser, TypeChecker).
  - ⋆ Automatikus állításgeneráló (Proof Obligaton Generator)
  - ⋆ Háromféle bizonyító rendszer:
    - ∗ automatikus (Autoprover),
    - ∗ interaktív (InterProver)
    - ∗ felhasználói (BToolProver)
  - ⋆ A specifikáció futtatása és tesztelése modell animátor segítségével (Animator).
  - ⋆ C programkód generálása (Translator).
    - ∗ Könyvtári elemek is részt vesznek, pl. a ki- bemeneti kódrészletek (InterfaceGenerator)
    - ∗ A kódrészleteket összekapcsolása és fordítása (Linker).
  - ⋆ LaTeX dokumentáció generálás (DocumentMarkUp).
  - ⋆ Verziókezelés és a függőségek nyilvántartása (Manager).

Felhasználási példák

- GEC Alstom, Matra Transport: vasúti és metró fejlesztések.
- TA Group Ltd.: ejtőernyő aktiválási rendszer.
- IBM: CISC/ESA rendszer, szoftver modellezési eszköz.
- Atomic Weapon Establishment: fegyverzet kezelő szoftver fejlesztése.
- SAET Meteor: biztonsági rendszer modellezése:
  - ⋆ több tízezer soros B kód,
  - ⋆ az Atelier B Tools állításgenerátora: 30.000 állítás,
  - ⋆ a bizonyítások kb. 80%-a automatikusan elvégezhető.