

XI. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2006. március 24-25.

MUNKAFOLYAMATOK SZIMULÁCIÓJA ÉS FORMÁLIS ANALÍZISE

Kovács Máté

With the use of electronically executed workflows, the efficiency of certain type of organizations may be improved. The execution of a workflow may involve several databases, so it is very important that it meets certain requirements. However the testing of workflow programs is problematic. A possible solution seems to be the formal analysis of them. In this paper a practical method is shown to check certain properties of a workflow.

Az elektronikus munkafolyamatok bevezetésével lehetőség van egy szervezet munkamenetének hatékonyabbá tételére. A munkafolyamat végrehajtás közben számos külső erőforrást elérhet, ezért nagyon fontos, hogy az bizonyos elvárásoknak megfeleljen. A munkafolyamatok tesztelése azonban nehézségekbe ütközik. Lehetséges megoldás, a munkafolyamatok egyes tulajdonságainak formális ellenőrzése, melyre egy gyakorlati módszert mutatok.

Bevezetés

Napjaink adminisztratív tevékenységet folytató intézményeiben egyre több informatikai segédeszközt használnak. Egyik közülük az elektronikus munkafolyamat végrehajtó motor, melynek segítségével a működés hatékonysága növelhető. A munkafolyamatok megfogalmazására több magas szintű nyelv áll rendelkezésre. Ilyenek például a BPEL (Business Process Execution Language) [3] és az XPD (XML Process Definition Language) [5].

Egy munkafolyamat példány végrehajtása során a folyamat több különböző szervezet számítógépén futó adatbázis-kezelővel vagy egyéb erőforrással érintkezhet közvetlen vagy közvetett módon, ezért nagyon fontos, hogy a munkafolyamat leíró program ne tartalmazzon bizonyos hibákat, melyeket a következőkben tárgyalok. A program tesztelése azonban nehézségekbe ütközik. A munkafolyamat tesztpéldánya által generált változtatásokat vissza kellene pörgetni az összes érintett adatbázisban, melyek több, egymástól független informatikai rendszerben is lehetnek. Egy másik

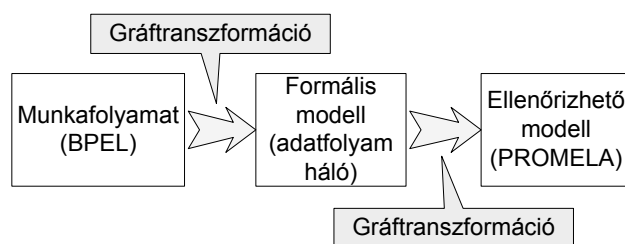
This work was partially supported by the SENSORIA European project (IST-3-016004).

lehetőség, a teljes tesztkörnyezet létrehozása az összes adatbázissal, és az általuk tárolt fiktív adatokkal.

Mindkét megoldás egyaránt drága. Ennek ellenére a munkafolyamat bevezetése előtt a biztonságos futás érdekében meg kell győződnünk annak egyes tulajdonságairól:

- A munkafolyamat nem tartalmazhat dead lock-ot. Egy dead lock esetén a végrehajtás félbeszakadna, melyet csak a folyamat újraindításával lehetne orvosolni.
- Egy változót előbb írni kell, és csak utána lehet olvasni. Egy változó idő előtti olvasása előre meg nem jósolható eredménnyel járna.
- Minden résztvevőnek elérhetőnek kell lennie. A felesleges tevékenységek eliminálásával erőforrásokat takaríthatunk meg.

Ebben a munkában egy olyan módszert ismertetek, melynek segítségével egy BPEL nyelvű munkafolyamat bizonyos tulajdonságait formálisan ellenőrizhetjük, és a fent említett tulajdonságairól meggyőződhetünk. Az első fázisban a munkafolyamatot modelltranszformáció segítségével adatfolyam hálóvá [1][6] képezem le. A második lépésben kódgenerálás segítségével automatikusan létrehozom az adatfolyam háló PROMELA (Process Meta Language) [4] nyelvű megvalósítását. A SPIN (Simple Promela Interpreter) [4] használatával ellenőrzöm a PROMELA program tulajdonságait, melyeket visszavetíték a munkafolyamatra. A transzformációs lépéseket a 1. ábra illusztrálja.



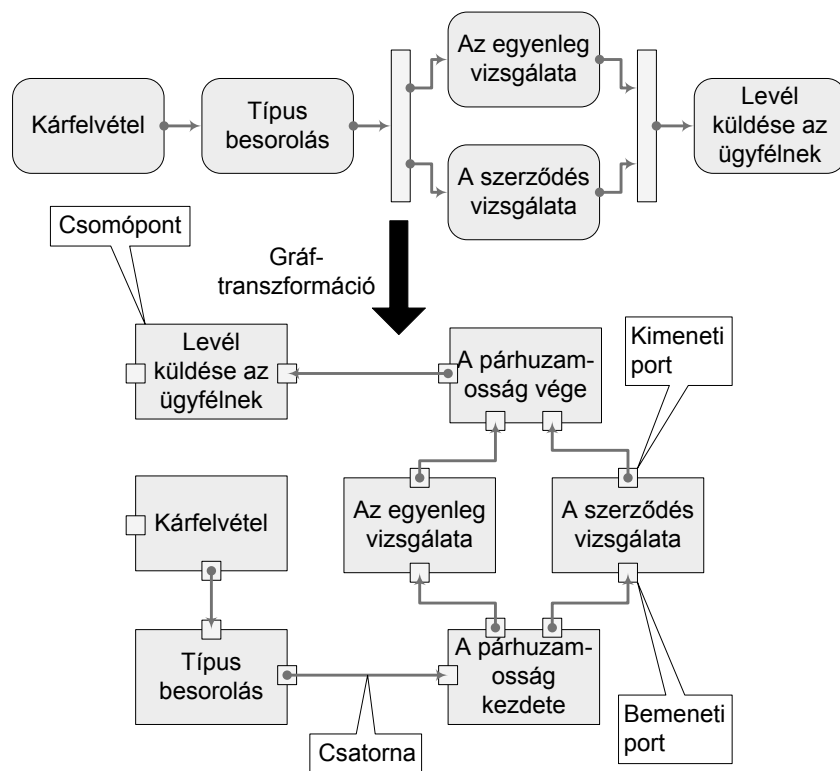
1. ábra Az ellenőrzés módszerének transzformációs lépései

A munkafolyamatok transzformációja adatfolyam hálóvá

Egy munkafolyamat az elemi tevékenységekből, és azokból a strukturális elemekből áll, melyek az elemi tevékenységek végrehajtási sorrendjét szabályozzák. A végrehajtási sorrendet a szekvencia, szelekció, iteráció és párhuzamos végrehajtás eszközeivel alakíthatjuk ki.

A munkafolyamatok precíz vizsgálatához szükség van egy egzakt szemantikával rendelkező matematikai formalizmusra, melynek transzformáció segítségével egy munkafolyamat megfeleltethető. Erre a célra az adatfolyam hálók elméletét választottam. Egy adatfolyam háló csatornákkal összekötött kommunikáló véges automatákból áll. A csatorna az elemi adategységek, a tokenek egyirányú szállítását végzi.

A transzformáció munkafolyamat mintákat adatfolyam háló mintákba képez le. A leképezést gráftranszformáció felhasználásával implementáltam, melyet a VIATRA2 [2] keretrendszer hajt végre.



2. ábra A munkafolyamat leképezése adatfolyam hálóra

A 2. ábra szemlélteti a leképezést, ami egy biztosítótársaság kisebb munkafolyamatának adatfolyam hálót felelt meg. A biztosítótársaság a „Kárfelvétel” nevű tevékenység végrehajtásával regisztrálja a káresetet, valamint a „Típus besorolás” alkalmazásával megállapítja a káreset kategóriáját. Annak vizsgálata, hogy az ügyfél biztosítása fedezi-e a kárt, valamint annak megállapítása, hogy az ügyfélnek van-e tartozása párhuzamosan is végrehajtható. A párhuzamosság leképezéséhez szükség van további két csomópont. Az első, „A párhuzamosság kezdete”, mely létrehoz egy további vezérlési szálát a megfelelő token beiktatásával. A második, „A párhuzamosság vége”, megvárja amíg a két párhuzamos vezérlési szál befejezi működését, majd továbbadja a vezérlést a szekvenciában következő tevékenységnek. Az adatfolyam háló modell PROMELA nyelvű megvalósítását automatikusan hozom létre, szintén gráftranszformáció alkalmazásával.

Formális ellenőrzés

Az előző fejezetben leírt transzformációk végrehajtásával megkapjuk a BPEL nyelven megvalósított munkafolyamat adatfolyam háló absztrakciójának PROMELA nyelvű megvalósítását. A SPIN segítségével elemezhetjük a programot. Előbb azonban meg kell fogalmaznunk a bizonyítani kívánt tulajdonságot LTL (Linear Temporal Logic) [4] formula formájában.

Ha például ellenőrizni szeretnénk, hogy a 2. ábrán látható „Típusbesorolás” nevű tevékenység bemenetétől szolgáló változót előbb minden esetben írja-e a munkafolyamat minthogy a tevékenység végrehajtna, a következő logikai állítást kell felírunk:

$G(\text{Típusbesorolás.bemenet=inicializálatlan} \rightarrow \text{nem } F(\text{Típusbesorolás végrehajtása}))$, ahol az F és G a lineáris temporális logika operátorai (F : a jövőben legalább egyszer, G : minden időpillanatban). A SPIN rendszer a PROMELA modell állapotterének teljes bejárásával bizonyítja a kifejezés igazságtartalmát.

A fent leírt transzformáció kisebb módosításával lehetőség van a munkafolyamat hibaszimulációjára. Szükség van hibainjektor tevékenységre. A változók állapotterét ki kell terjeszteni, hogy tárolni tudják azt az információt is, hogy épp hibás értéket írtak beléjük, és hogy esetlek már ki is olvasták azt. Szükség van további tokenekre, melyek a hibás vezérlést reprezentálják. A hibaszimuláció alkalmazásával lehetőség van megvizsgálni, hogy egy tevékenység hibája, a munkafolyamat mely további részére van kihatással. Megvizsgálhatjuk, hogy az esetlegesen betervezett redundancia elérte-e célját, és megvalósítja-e a szándékolt hibatűrést.

Összefoglalás

A munkafolyamatok tesztelése nehézségekbe ütközik. Ugyanakkor egy munkafolyamat akár egy teljes intézmény ügyvitelét is irányíthatja, ezért nem vehetjük használatba mielőtt meg nem győződünk annak bizonyos tulajdonságairól. Ebben a cikkben egy módszert mutattam, melynek segítségével formálisan elemezhetünk egy munkafolyamatot, így elkerülve a tesztelés költségeit. A vizsgálati módszer gráfranzformáción alapszik, melyet a VIATRA2 keretrendszer hajt végre.

Hivatkozások

- [1] Gy. Csértán, A. Pataricza, P. Harang, O. Dobán, G. Biros, A. Dancsecz, and F. Friedler. BPM based robust e-business application development. Proc. Dependable Computing - EDCC-4, 4th European Dependable Computing Conference, p. 32-43. Springer, Berlin etc., 2002.
- [2] VIATRA2 Eclipse GMT subproject, <http://www.eclipse.org/gmt/>
- [3] Specification of the Business Process Execution Language Version 1.1. 2003: <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>
- [4] Simple Promela Interpreter: <http://www.spinroot.com>
- [5] XML Process Definition Language Version 2.0.: <http://www.wfmc.org/standards/XPDL.htm>
- [6] A.J. Anderson, Data Flow Systems. In Multiple Processing: Systems Overview, ch. 10. p. 441-488, Prentice Hall, UK, 1989.

Kovács Máté

Méréstechnika és Információs Rendszerek Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem
H-1521, Budapest,
Magyar tudósok körútja 2.
tel: +36 1 463 3579 fax: +36 1 463 2667
e-mail: km432@hszk.bme.hu