ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS
SBORNÍK MENDELOVY ZEMĚDĚLSKÉ A LESNICKÉ UNIVERZITY V BRNĚ

Ročník LIII 13 Číslo 3, 2005

KONZISTENCE PODNIKOVÝCH PROCESŮ VYJÁDŘENÁ FORMÁLNÍM APARÁTEM TRANSAKCÍ

M. Mišovič, I. Rábová

Došlo: 20. prosince 2004

Abstract

MIŠOVIČ, M., RÁBOVÁ, I.: *Enterprise process consistency expressed by a formal description of transactions*. Acta univ. agric. et silvic. Mendel. Brun., 2005, LIII, No. 3, pp. 117-126

Using a progressive Information Technology for development of Software Modules for Enterprise Information Systems brings a lot of practical and theoretical problems. One of them is a verification of results achieved in Life Cycle Stages. Object Oriented Analysis has the main position in the Object Life Cycle of Information Systems. It gives fundamental diagrams that will be processed in the Design and Implementation phases. We mention diagrams for enterprise processes and their refining to enterprise transaction diagrams. The Unified Modeling Language (UML) has been very often used to enterprise processes and transactions modeling. There is one of very practical and theoretical problems concerning enterprise processes – their internal consistency that can be observed on the level of object transactions. We have in mind such problem as transaction feasibility and object cooperation feasibility in transactions that are strongly bound with states of objects. Therefore, a testing of object complex transactions before their programming appears to be very useful activity.
This article introduces a formal description of the object cooperation logic. Therefore there is defined not only an elementary transaction feasibility but also an elementary object cooperation feasibility. It enables to search the feasibility of certain strings of elementary transactions and elementary object collaborations. One string of elementary transactions is very often regarded as a path. There are found two different systems of state logical equations. The first describes path transaction feasibility and the second path object cooperation feasibility. The functional correctness of any complex transaction is founded on a functional correctness of all its paths.

object Cooperation logic, elementary transaction feasibility, state logical equations, elementary object cooperation feasibility, path, path transaction feasibility, path object cooperation feasibility, functional correctness of a path and functional correctness of a complex transaction

Vzhledem k rychle se měnícímu konkurenčnímu prostředí a k rostoucí potřebě dynamicky na tyto změny reagovat přechází organizace často na procesní způsob řízení a vyvstává potřeba vizualizovat a dokumentovat pracovní toky, procesy, vstupy a výstupy těchto procesů i jiné podnikové koncepty související s provozováním podniku.

Základním předpokladem efektivního fungování jakékoliv společnosti je jasná definice vizí, cílů a strategií a popis postupů, jak těchto cílů dosáhnout. Zmapování podnikových činností a jejich formalizace pomocí vhodné notace je důležitým aspektem pro zavedení promyšleného procesního řízení založeného na velmi dobré znalosti firemních procesů, pro kontrolu jejich dodržování, adaptaci na měnící se prostředí a doplňování v závislosti na měnící se cíle organizace a také pro implementaci softwarových produktů podporujících procesy.

Podnikové modelování jako inženýrská disciplína není příliš propracovanou oblastí. Chybí dostatek kvalitních zdrojů pro modelování a analýzu podnikových procesů, událostí, zdrojů i cílů. Tento fakt souvisí s nedoceněním významu modelování obecně a především vyplývá z nízké úrovně nebo z nedostatku možností či přístupu k metodikám a technikám pro prezentaci podnikové architektury. Jako každá inženýrská disciplína i podnikové modelování vyžaduje vedle metodologií a vzorů pro znuvupoužitelnost vhodnou a vyhovující notaci, eventuálně nástroje. Ale ukazuje se, že správná inženýrská disciplína potřebuje také dobře stanovené matematické základy, tj. konzistentní framework matematických modelů pro své modely.

V kontrastu technologií soustředěných na „Jak“, pak matematika především vyjadřuje „Co to je“ a „Jak to provést“ a na tomto základu tak přispívá komponentě „Jak“ nabídnutím vhodné notace a efektivního aparátu, často už existujícího, dokončeného a rozpracovaného bez vztahu k inženýrské doméně, o které mluvíme.

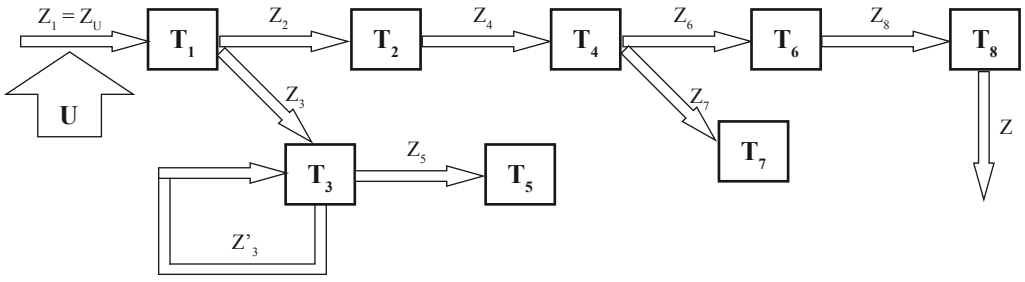
MATERIÁL A METODIKA

V příspěvku se snažíme o matematické vyjádření některých aspektů podnikového modelování týkajících se vnitřní konzistence procesů. Máme na mysli podmínky proveditelnosti procesů při jejich vyjádření pomocí diagramů objektové spolupráce, které jsou složeny z transakcí. Základem těchto diagramů je spolupráce mezi transakcemi a spolupráce objektů v jednotlivých transakcích. Zmíněná spolupráce vede k realizaci podnikových procesů a je podmíněna konzistencí cesty transakcí a stavů objektů, do kterých se cestou dostávají. Jelikož transakce (dále nedělitelné aktivity) a objekty (prvky reality s elementární aktivitou) v nich používané jsou základem podnikových procesů, musíme nejdříve vysvětlit jejich podstatu a zavést nad nimi nové pojmy jako „*stavová formule transakce*“, „*elementární transakce*“, „*cesta v transakci*“, „*proveditelnost cesty*“, „*průchod cestou*“.

1. Pojetí transakce

Celková aktivita podniku se odehrává v dílčích komplexních transakcích a proveditelnost procesů je závislá na proveditelnosti transakcí, z nichž se skládají. Každá transakce má svůj popis, v němž:

- je uvedena **událost U**, s níž je transakce svázaná a inicializační **zpráva $\mathbf{Z_U}$**, která událost reprezentuje (viz Obrázek 1),
- jsou známé **třídy $\mathbf{T_1}$**, **$\mathbf{T_2}$**, …objektů, které se účastní transakce,
- je zřejmý **tok požadavků na služby** objektů, vedený prostřednictvím zpráv $\mathbf{Z_U \equiv Z_1}$, **$\mathbf{Z_2}$**, …a jsou známé důsledky provedení transakce.



1: *Diagram formální transakce*

Ačkoliv je transakce prezentovaná nad třídami, je potřeba si uvědomit, že instance transakce již uvádí konkrétní objekty jednotlivých tříd, které se transakce účastní. Každá zpráva **Z** obsahuje operaci **ω**, vstupní **X** a výstupní **X′** parametry, tedy

$$\mathbf{Z = (\omega, X, X')}.$$

Spolupráce objektů v transakci probíhá vždy po jisté spojité **cestě**, přičemž diagramy transakcí často obsahují i několik cest. Ptáme se, co je zárukou toho, že spolupráce tříd (na nižší úrovni spolupráce konkrétních objektů) proběhne po jisté cestě a „co“ je zárukou toho, že elementární transakce této cesty jsou proveditelné? Úkolem tedy bude najít „co“ stanovuje spolupráci objektů cesty a „co“ je hlídačem proveditelnosti elementárních transakcí.

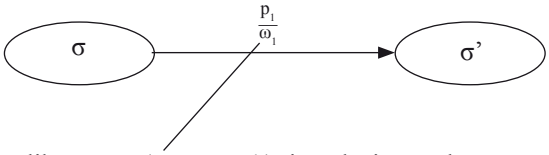
To, že spolupráce objektů proběhne po jisté cestě, je dáno vysíláním zpráv z jednotlivých objektů prostřednictvím jejich operací ω. V jádru topologie komplexní transakce je jedna třída (centrální třída), která má za úkol vyplnit služby požadované z rozhraní. Od této služby se odvíjí spolupráce objektů. Požadavky na služby vycházející z centrální třídy na jiné třídy musí přinést další údaje, které centrální třída potřebuje k úpravě svých objektů. Velmi často zase jedna terminální třída cesty vysílá výsledné zprávy do rozhraní.

Předpokládejme, že spolupráce objektů v diagramu na Obrázku 1 proběhne po cestě α_1, kde

$\alpha_1 = [(Z_1, T_1, Z_2), (Z_2, T_2, Z_4), (Z_4, T_4, Z_6), (Z_6, T_6, Z_8), (Z_8, T_8, Z)]$.

Zaveďme nyní pojem *elementární transakce* a vyšetřeme její proveditelnost. K tomu nám poslouží následující obrázek, který uvádí do vzájemné souvislosti proveditelnost elementární transakce se stavovým diagramem třídy uvedené v transakci.

Elementární transakce	Přechod v diagramu stavů dané třídy
(Z_1, T_1, Z_2)	této elementární transakci musí odpovídat elementární přechod mezi stavy ve stavovém diagramu pro třídu T_1



Jelikož $Z_1 = (\omega_1, X_1, X'_1)$, je náležitost tohoto přechodu k elementární transakci zřejmá. Asociace je dána přes ω_1, X_1, X'_1.

2: *Elementární transakce a stavový diagram třídy uvedené v transakci*

Z uvedeného plyne, že *transakce (Z_1, T_1, Z_2) je proveditelná tehdy a jen tehdy, je-li podmínka $\boldsymbol{p_1}$ pravdivá*. Korektnost provedení transakce a současný přechod objektu z předepsaného stavu výchozího do předepsaného stavu výsledného ještě závisí na provedení operace ω_1. Dvojici **(p_1, ω_1)** můžeme proto považovat za **hlídače provedení elementární transakce (Z_1, T_1, Z_2)**. Provedeme-li podobnou úvahu i pro ostatní elementární transakce, dostaneme pro cestu α_1 množinu hlídačů **$\{(p_i, \omega_i)\}_{\alpha 1}$** elementárních transakcí. Tato množina je komplexní **podmínkou pro provedení elementárních transakcí** cestou α_1. Nejsou to ale hlídače následnosti elementárních transakcí.

Poznámka

Síť spolupracujících objektů může být chápána jako orientovaný ohodnocený graf, kde hrany jsou tvořeny relací přechodu $\Longrightarrow$, uzly jsou dány třídami a ohodnocení je dáno zprávami. Toho může být využito k jinému přístupu k transakcím.

Diagramy transakcí mohou být poměrně rozsáhlé a je pravděpodobné, že analytik může udělat četné chyby. Tyto chyby mohou mít logickou povahu a často k nim patří:

1. nepříjemné smyčky v transakcích,
2. nerealizovatelné přechody mezi stavy,
3. neproveditelné transakce nebo přechody mezi transakcemi.

Zajímá nás také, jestli můžeme dát odpověď na následující dvě otázky:

1. **Jsme schopni nalézt takové stavy spolupracujících objektů vybrané cesty v dané transakci, aby cesta proběhla korektně, tj. dílčí transakce jsou proveditelné a je zabezpečen průchod vybranou cestou?**
2. **Jsme schopni nalézt takové populace tříd abychom je považovali za testovací bázi objektů pro systém vytvořený v Objektově orientované implementaci?**

Otázka první se týká nejprve **logické bezespornosti**, tedy konzistentnosti diagramů jednotlivých transakcí a existujících modelů stavů objektů zúčastněných tříd, což zabezpečuje proveditelnost cesty a potom **zabezpečení průchodu cestou**. Proveditelnost cesty v transakci je pouze nutnou podmínkou její korektnosti, ale ne podmínkou dostatečnou (**neproveditelná transakce nemůže být korektní**).

Formulace této otázky, tj. **proveditelnosti** cesty může být založena na následující myšlence:

- Vyjádřeme provedení cesty α v transakci (řetěz spolupracujících objektů, řetěz elementárních transakcí) systémem tzv. stavových formulí, v nichž jako neznámé veličiny vystupují stavy zpracovávaných objektů.
- Hledejme takové stavy objektů, po jejichž dosazení do systému stavových formulí jsou stavové formule tautologiemi[1]. Najdeme-li takové stavy, potom je cesta proveditelná.
- Jsou-li proveditelné všechny cesty v transakci, je proveditelná i komplexní transakce.

[1] Tautologie je formule predikátového kalkulu, která má jedinou výslednou hodnotu „**pravda**“.

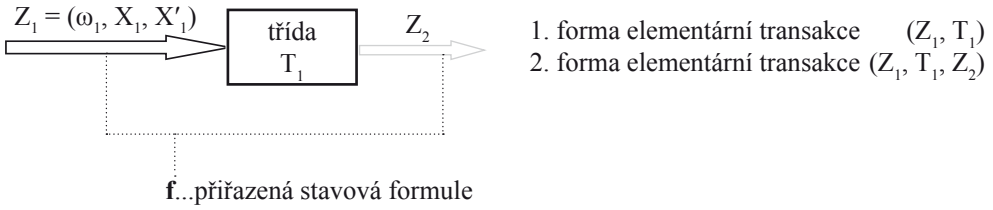
- Existuje-li v transakci smyčka, můžeme vyšetřit zda je proveditelná a najít princip definující „nezbytný“ počet průchodů (jako reprezentant případného nekonečného opakování).

Řešením druhé otázky může být nalezení algoritmu, jenž je schopen neznámé stavy zpracovávaných objektů „vypočítat“ tak, aby uvažovaná transakce byla korektní. To je ale příliš silný požadavek na hledaný algoritmus. Nároky zmírníme, jestliže tento algoritmus umožní interaktivní pomoc analytika pro prověrku aktivity transakce. Jestliže tento algoritmus aplikujeme na každou transakci, získáme tak všechny praktické životní cykly objektů jednotlivých tříd objektového modelu. Takto nalezené stavy objektů zabezpečí korektní provádění všech transakcí a takové objekty a jejich třídy mohou být **určujícími testovacími daty** pro implementační model informačního systému. Kardinální otázkou uvedené myšlenky je nalezení požadovaného algoritmu.

Abychom vyjádřili naznačené myšlenky řešení obou zmíněných otázek, potřebujeme nejdříve vybudovat formální reprezentaci stavů tříd, kterých nabývají její objekty. K tomu nám poslouží tzv. ***stavové formule transakce***.

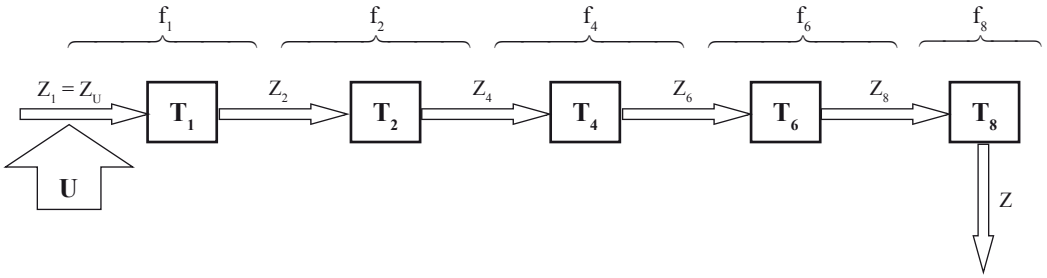
2. Stavové formule transakce

Zaveďme nyní pojem tzv. *stavové formule* pro libovolnou cestu v transakci. Jedna formule bude náležet jedné elementární spolupráci, tedy *elementární transakci*. Tyto formule budou obsahovat podmínky, které jsou-li splněny, zaručují proveditelnost spolupráce dvou objektů. Zavedení stavových formulí ilustrujme na následujících dvou transakcích, elementární a obecné.



3: *Elementární stavová transakce*

Pro cestu α_1=[(Z_1, T_1, Z_2), (Z_2, T_2, Z_4), (Z_4, T_4, Z_6), (Z_6, T_6, Z_8), (Z_8, T_8, Z)] dojde k následující asociaci elementárních transakcí a stavových formulí:



4: *Obecné stavové transakce pro cestu* α_1

Prozatímní hlídač $(\mathbf{p_1}, \boldsymbol{\omega_1})$ bude později plně zahrnut ve stavové formuli $\mathbf{f_1}$. Splnění stavové formule bude považováno za nutnou podmínku provedení spolupráce dvou objektů v elementární transakci, vysílacího a přijímacího (je-li stavová formule nepravdivá, spolupráce se nemůže provést).

Zpracování vybraného objektu v jednotlivých transakcích operacemi z jisté množiny **Ω**, často vede ke změně hodnot v n-tici $(x_1, x_2, \ldots, x_n)$ – informační struktuře objektu nebo změně jeho spojení s jinými objekty a tím k vytvoření dalšího stavu. Životní cyklus objektu je postaven na jeho stavech. Objekt je

generován – zřízen a dán do počátečního stavu $\boldsymbol{\sigma}^{\text{počá-}}_{\text{teční}}$, potom je zpracováván v transakcích prováděním operací a tak může pro každou transakci přecházet ze stavu výchozího $\boldsymbol{\sigma}^{\text{výchozí}}$ postupně až do stavu výsledného $\boldsymbol{\sigma}_{\text{výsledný}}$. Jeden z výsledných stavů může být stavem koncovým $\boldsymbol{\sigma}_{\text{koncový}}$, kterým končí životní cyklus objektu.

Pro následující stav objektu jsou určující:

- jeho aktuální stav (stav výchozí pro následující operaci),
- podmínka **p** a
- operace ω, která se má provést.

Přechod z jednoho stavu do druhého můžeme zapsat jednoduchou stavovou formulí

$$\mathbf{f}: \boxed{\begin{array}{l} \sigma^{\text{výchozí}} = \sigma \overset{p,\,\omega}{\Longrightarrow} \sigma_{\text{výsledný}} \in \{\sigma_1', \sigma_2', \ldots, \sigma_n'\} \\ \sigma^{\text{výchozí}} = \sigma \overset{p,\,\omega}{\Longrightarrow} \sigma_{\text{výsledný}} = \sigma' \end{array}} \quad (1)$$

Zápis můžeme číst takto:

„Jestliže je výchozí stav objektu roven σ a provede se operace ω za platnosti podmínky p, potom to implikuje, že výsledný stav bude σ′ nebo jeden ze stavů z množiny $\{\sigma_1', \sigma_2', \ldots, \sigma_n'\}$“.

Stavová formule (1) zahrnuje obě situace, v nichž je možný deterministický nebo nedeterministický přechod do výsledného stavu. Na druhé straně může nastat situace, kdy výběr výchozího stavu je nedeterministický. Tomuto případu odpovídá tvar stavové formule (2):

$$\mathbf{f}: \boxed{\begin{array}{l} \sigma^{\text{výchozí}} \in \{\sigma_1, \sigma_2, \ldots, \sigma_m\} \overset{p,\,\omega}{\Longrightarrow} \sigma_{\text{výsledný}} = \sigma' \\ \sigma^{\text{výchozí}} \in \{\sigma_1, \sigma_2, \ldots, \sigma_m\} \overset{p,\,\omega}{\Longrightarrow} \sigma_{\text{výsledný}} \in \{\sigma_1', \sigma_2', \ldots, \sigma_n'\} \end{array}} \quad (2)$$

Každá stavová formule tedy řeší jeden přechod v diagramu stavů. Pokaždé to ale může být přechod ve stavovém diagramu jiné třídy. Podmínka **p** je obecně predikátovou formulí, tj. formulí, jejíž individuální proměnné mají za své hodnoty stavy různých objektů. Jelikož cesta představuje v jednotlivých elementárních transakcích realitou odůvodněný sled spolupráce objektů, potom její „kvalitu“ můžeme popisovat stavovými formulemi.

Stavová formule, asociovaná s jistou elementární transakcí, říká, že přijde-li na objekt zpráva **Z** s požadavkem provést službu **ω**, potom aby to bylo možné, musí být objekt v požadovaném stavu $\sigma^{\text{výchozí}}$ a musí být splněna jistá podmínka **p**. Jelikož cesta **α** v komplexní transakci je posloupností elementárních transakcí, vzniká pro ni systém stavových formulí $\mathbf{N}_{\alpha}(\mathbf{f}_i)$. Tento systém budeme považovat za hlídače provedení elementárních transakcí cesty **α**.

Definice 1:
Cestou v libovolné transakci nazveme posloupnost

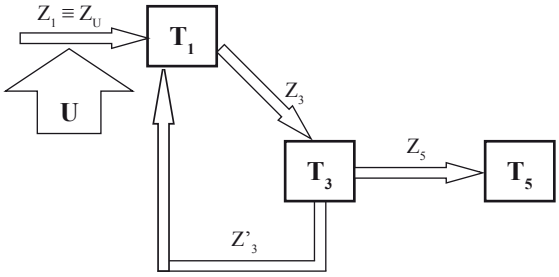
$$\alpha = [(Z_1, T_1, Z_2), (Z_2, T_2, Z_3), \ldots, (Z_k, T_k, Z_{k+1})],$$

kde Z_i jsou zprávy, T_i třídy
a pro i = 1, 2, …, k – 1 platí, že (Zi, Ti, Z_{i+1}) je elementární transakce pouze 1. formy (1) a pro i = k může být koncová transakce 1. nebo 2. formy (2).

Transakce může obsahovat jednu nebo více cest. Zprávy Z_1 a Z_{k+1} mohou být externí. Cesty, které mají obě krajní zprávy externí, se nazývají **START-STOP cesty**. Takové cesty vždy představují rozumnou komunikaci okolí se systémem přes stanovenou hranici.

Některé cesty mohou obsahovat smyčky, tj. opakuje se vícekrát spolupráce stejných tříd. Kardinální otázkou spojenou s těmito cyklickými cestami je „ukončení“ opakování smyčky. Je to pro teoretickou informatiku adekvátně nerozhodnutelný problém jako ukončení cyklu v programu. Kardinální otázce se dá „vyhovět“ tak, že najdeme kritérium pro ukončení smyčky. Kritérium by mělo zabezpečit, že dalším opakováním smyčky „nedosáhneme nic nového“.

Následující Obrázek 5 ilustruje jednu z možných cyklických cest.
$\beta_4 = [(Z_1, T_1, Z_3), \{(Z_3, T_3, Z'_3), (Z'_3, T_1, Z_3), \ldots (Z_3, T_3, Z'_3), (Z'_3, T_1, Z_3)\}^*, (Z_3, T_3, Z_5), (Z_5, T_5)]$ je cyklická cesta, jejíž smyčka je složena ze dvou elementárních transakcí (Z_1, T_1, Z_3) a (Z_3, T_3, Z'_3).



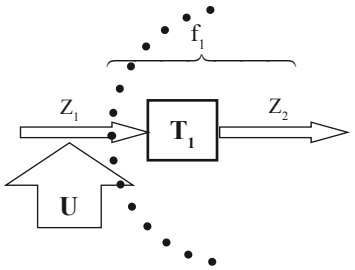
5: *Smyčka v transakci*

Každá cesta má konečný počet větví a platí to i o transakci. Má-li transakce cyklickou cestu, není obecně počet cest konečný. Hlavní význam větví spočívá v možnosti vyhnout se problému cyklických cest. Větve umožňují přenést vyšetřování proveditelnosti a korektnosti na nižší – parciální úroveň. To sice nezaručuje proveditelnost a korektnost celé cesty, ale testovací výsledky jsou přijatelné. My se větvemi dále zabývat nebudeme, šlo pouze o upozornění na existenci této možnosti.

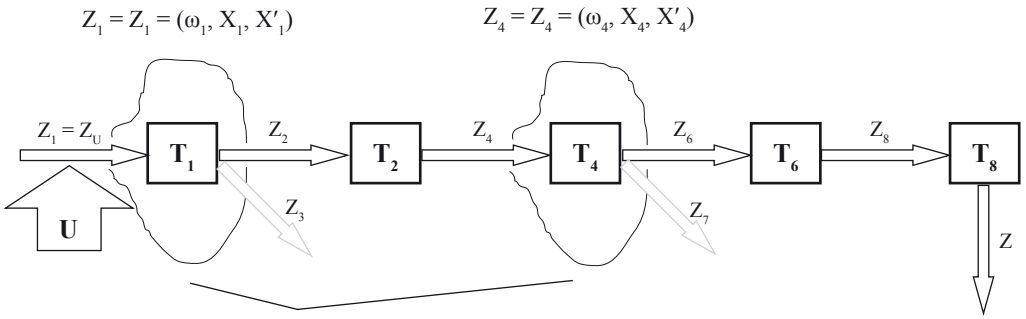
3. Průchod cestou a proveditelnost cesty

Průchod cestou v transakci je nerozlučně spjat s *proveditelností cesty*. Jak bylo uvedeno výše, jsou to právě stavové formule cesty, které, jsou-li postupně pravdivé, tak zabezpečují proveditelnost jednotlivých elementárních transakcí cesty.

Pravdivost stavové formule $\mathbf{f_1}$ k elementární transakci (Z_1, T_1, Z_2) cesty α_1 v Obrázku 6 je nutnou podmínkou provedení spolupráce externího objektu (vydávajícího zprávu Z_1) a přijímajícího objektu z třídy T_1. Pochopitelně, pravdivost systému $\mathbf{N_{\alpha 1}(f_i)}$ je nutnou podmínkou proveditelnosti cesty α_1. Jsou-li proveditelné všechny cesty v transakci, můžeme mluvit o proveditelnosti celé transakce. Je tady ještě otázka zabezpečení přechodů mezi elementárními transakcemi cesty (tedy mezi spolupracujícími objekty), tedy otázka průchodu cestou. Jestliže operace ω vydává pouze jednu zprávu pro spolupráci s dalším objektem, je přechod zcela přirozený. Jestliže operace ω může vydat více zpráv, potom jejich vydávání musí sama hlídat pomocí souhrnu podmínek. Tak je tomu i v případě cesty $\boldsymbol{\alpha_1}$ z Obrázku 1.



6: *Stavová formule cesty*



7: *Poloha hlídačů průchodu cestou* $\boldsymbol{\alpha_1}$

Jak je naznačeno na Obrázku 7, na cestě $\boldsymbol{\alpha_1}$ jsou dva hlídače průchodů cestami v operacích ω_1 a ω_4. Případů, kdy v jedné elementární transakci vydává operace ω více výstupních zpráv na jiné objekty (což je chápáno jako událost pro tyto objekty), je mnoho. Otázkou pouze je, jestli vydávání zpráv může být provedeno paralelně nebo sériově. V běžných implementacích je situace převážně řešena sériově (tím ale nechceme nikterak potlačovat roli paralelismu v objektovém modelování). V obecném případě je operace ω, která vysílá např. n zpráv na jiné objekty, vybavena podmínkami $q_1, q_2, \ldots, q_n$. Hlídač je pak ve tvaru $\{\omega, q_{i.\,i} = 1, 2, \ldots, n\}$. Z těchto podmínek je pravdivá pouze q_i, jejíž zpráva má být vydána. Poté, co se vrátí výsledný parametr vydané zprávy, stane se její podmínka nepravdivá a podmínka jiné zprávy se stane pravdivá. Podmínky $q_1, q_2, \ldots, q_n$ jsou vzájemně spjaté, což je odůvodněno tím, že jde o „dokončení" operace ω nad objektem téže třídy. Pro celou cestu α můžeme tedy sestavit systém $H_\alpha(q_i)$, kde q_i jsou vybrány z hlídačů průchodů cestami. Můžeme již definovat korektnost cesty.

Definice 2:
Buď $\alpha = [(Z_1, T_1, Z_2), (Z_2, T_2, Z_3), \ldots, (Z_k, T_k, Z_{k+1})]$ libovolná cesta komplexní transakce, $\mathbf{H_\alpha(q_i)}$ systém podmínek pro průchod cestou a $\mathbf{N_\alpha(f_i)}$ systém stavových formulí cesty.

Cesta α je korektní tehdy a jen tehdy, jestliže:

1. existují takové hodnoty atributů multi-vysílajících objektů cesty α, že $\mathbf{H_\alpha(q_i)}$ jsou pravdivé podmínky (což zaručuje průchod cestou) a
2. existují takové stavy spolupracujících objektů cesty α, že převádí formule systému $\mathbf{N_\alpha(f_i)}$ na tautologie (což zaručuje proveditelnost cesty).
3. Transakce je korektní, jsou-li korektní všechny její cesty.

Definice 2 je důsledkem dříve uvedeného objasněním koncepce a role obou systémů $\mathbf{H_\alpha(q_i)}$, $\mathbf{N_\alpha(f_i)}$.

DISKUSE A ZÁVĚR

Vyslovme některé poznámky k užitečnosti systému podmínek $\mathbf{H_\alpha(q_i)}$ z Definice 2:

1. Jelikož objekt může být žádán o provedení i stejných služeb různými objekty, mají hodnoty atributů hodnotovou historii. To se ale týká každého atributu jiným způsobem. Např. „výpočtářské" atributy mají silnou **hodnotovou historii**, definiční – primární atribut ji obvykle nemá. Jestliže **a** je atribut objektů třídy T, tak jeho hodnotovou historii můžeme zobrazit posloupností $a^0, a^1, a^2, \ldots$ Pokud o hodnotách atributů budeme uvažovat jen abstraktně, potom se tato označení, případně doplněná ještě indexem třídy, dají použít ve formulaci podmínek $\{q_i\}$, které využívají právě poslední prvky hodnotových historií atributů.
2. Systém $\mathbf{H_\alpha(q_i)}$ může být naformulován v abstraktních označeních hodnot atributů. Jestliže je považujeme za neznámé a pokusíme se je řešením systému získat, potom vypočtené hodnoty atributů zabezpečí, že multi-vysílací objekty vydávají pouze zprávy ležící na zkoumané cestě α. Tím je ale zabezpečen průchod touto cestou.

Se systémem stavových formulí cesty $\mathbf{N_\alpha(f_i)}$ můžeme manipulovat dvěma způsoby:

1. *na abstraktní úrovni* budeme predikátové proměnné, jejichž hodnotami by měly být stavy objektů, považovat za neznámé veličiny a budeme hledat jejich takové hodnoty, které převedou formule na tautologie. Podaří-li se takové hodnoty najít, je cesta proveditelná. Otázkou je, je-li to právě jedno řešení nebo jich je nekonečně mnoho. Na druhé straně, podaří-li se nám dokázat, že řešení neexistuje, je cesta neproveditelná a nemá smysl mluvit o její korektnosti. Jde tedy o obecný problém řešení systému predikátových formulí. Jednoduché systémy $\mathbf{N_\alpha(f_i)}$ dokáže vyřešit šikovný logik, složitější se musí řešit pomocí resoluční metody (problematika, která je běžnou součástí teoretické informatiky).
2. *v praktické rovině*, kdy vyzkoušíme, zda-li některé prvky relevantních stavů (relevantní stavy jsou množiny) asociovaných s danou cestou, nepřevádí formule v tautologie. Jestliže tam takové prvky nemáme, potom buď doplníme relevantní stavy o nové prvky, anebo zjistíme, že stavy, které by vyhovovaly formulím, nemohou být zařazeny do žádného relevantního stavu (souhrn relevantních stavů je neúplný a musíme ho revidovat).

Zamysleme se na závěr nad dalším aspektem spojeným s výše popsanými úvahami, a to nad možnostmi testování tříd (testovací báze tříd a jejich instancí). Zařazení testování výsledků objektově orientované analýzy ihned po jejím ukončení ještě není běžné. Testování se totiž obvykle soustřeďuje do fáze implementace, kdy se získají první programové moduly. V tom okamžiku je nutné mít prověřenou testovací bázi. Myslíme tím komplex tříd a jejich konkrétních instancí, které jsou úplně analyzované, tj. jsou pro ně vytvořené všechny potřebné diagramy:

- diagram struktur (generalizace a specializace, informační asociace, agregace),
- diagramy stavů,
- diagramy dílčích transakcí a sjednocený model transakcí.

Je poměrně nevhodné konstruovat takovou bázi dat až v době implementace. Nevýhody jsou pochopitelné. Jsme již natolik vzdáleni od analýzy, že zpětný návrat k ní může být zdrojem mnoha chyb.

Snahou teoretické informatiky je sestrojit testovací bázi dat automaticky bez pomoci analytika. Takovéto snahy ale ztroskotaly již při konstrukci úplné množiny testovacích dat k programům, dále ztroskotaly pro konstrukci testovací báze entit po provedení klasické strukturované analýzy a jistě nebudou mít dlouho úspěch pro tak složité struktury jako jsou třídy a jejich instance – objekty.

Pojetí úplné množiny testovacích tříd a jejich instancí je adekvátní pojetí úplné množiny testovacích entit a jejich instancí. Algoritmus je ale složitější:

1. Najdeme všechny cesty ve sjednoceném modelu transakcí, každou cyklickou cestu převedeme na konečný počet reprezentantů.
2. Ke každé cestě α sestrojíme systémy $H_\alpha(q_i)$ a $N_\alpha(f_i)$ a nalezneme jejich řešení R_α a R'_α
3. Vynecháme neprůchodné cesty.
4. Potom úplná testovací množina tříd a jejich instancí je dána vztahem $\bigcup_\alpha R'_\alpha$

Tento postup má jednu velkou závadu. Nejsou obecně známé metody na řešení zmíněných predikátových systémů $H_\alpha(q_i)$ a $N_\alpha(f_i)$. Interaktivní postup prováděný konstruovaným „testerem“, dávající analytikovi všechny potřebné informace, je možnou a schůdnou cestou pro konstrukci požadované testovací báze. To ale žádá poskytnout analytikovi všechny potřebné údaje, aby v interaktivním režimu měl možnost pro celý sjednocený model transakcí:

- nahradit každou cyklickou cestu konečným souhrnem reprezentantů,
- vyšetřit každou cestu na průchodnost, tj. stanovit podmínky pro průchod cestou a vyřadit neprůchodné cesty,
- vyšetřit každou průchodnou cestu a zjistit stavy kooperujících objektů,
- sepsat všechny vyhovující stavy objektů a sestrojit testovací bázi dat.

SOUHRN

V příspěvku je uvedena původní formalizace pojetí komplexní transakce (sekvence objektové spolupráce) a její cesty. Podnikové procesy jsme nahradili pomocí transakcí a převedli jsme konzistenci procesů na konzistenci transakcí. Zavedli jsme pojmy „průchod cestou“, „proveditelnost cesty“ a formulovali podmínky, na základě kterých jsou problémy „průchod“ a „proveditelnost“ rozhodnutelné. Je zavedena korektnost cesty, když je cesta průchozí a realizovatelná. Na základě korektnosti všech cest dané transakce je tato prohlášena za korektní a tedy bezesporná. Verifikace komplexní transakce je navržena v interakci s analytikem, protože řešení obou systémů $\mathbf{H_\alpha(q_i)}$ a $\mathbf{N_\alpha(f_i)}$ pro všechny cesty transakce je obecně velmi obtížné. Ukazuje se také, jak pomoci analytikům při verifikaci sjednoceného modelu transakcí a kolekce diagramů stavů tříd pomocí užitečného testeru.

Všechny uvedené aspekty podnikového modelování jsme uvažovali pouze na abstraktní úrovni. Použitelnost výsledků pro podnik je zřejmá. Aby podnik pracoval optimálně, musí pracovat jeho podnikové procesy optimálně. Aby byl podnikový proces korektní a optimální, musí být korektní všechny transakce, ze kterých se skládá, musí být průchodné a proveditelné. Ve většině případech jsou procesy automatizované, podporované informačními technologiemi a aplikačním softwarem. Pokud má být tato softwarová podpora účinná a implementovaný systém správný, je nezbytné, aby byl analyzován a navržen v kontextu výše popsaných doporučení a otestován pomocí vhodné testovací báze dat.

diagramy objektové spolupráce, logika objektové spolupráce, proveditelnost procesu, transakce, elementární transakce, cesta transakce, stavové formule transakce, stavový diagram

LITERATURA

RUMBAUGH, J. a kol.: Object - Oriented Modeling and Design. Prentice Hall, 1990.

BOOCH, G.: Object – Oriented Design with Applications. Redwood City, 1991.

MIŠOVIČ, M.: Verifikace výsledků objektově orientované analýzy. OBJEKTY'99, Praha, 1999.

MIŠOVIČ, M.: Functional Correctness of Complex Transactions. MOSIS'2000, Rožnov pod Radhoštěm, 2000.

RÁBOVÁ, I.: Význam podnikové architektury pro vývoj softwarového systému. Mezinárodní konference Firma a konkurenční prostředí, Brno, 2002, str. 112-124, ISBN 80-7302-036-X.

Adresa

Prof. RNDr. Milan Mišovič, CSc., Ing. Ivana Rábová, Ph.D., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: misovic@mendelu.cz, rabova@mendelu.cz