

OBJEKTOVÁ ALGEBRA

Zdeněk Pezlar

Ústav Informatiky, Provozně-ekonomická fakulta MZLU, Brno, ČR

Abstrakt

V objektovém modelu dat definujeme objektové schéma (třidu) jako čtveřici skládající se ze jména třídy, atributů, domén a procesů. Nad tímto schématem definujeme objekty. Strukturu systému popisujeme jak ve statickém pohledu – pomocí vztahů mezi třídami a objekty, tak v dynamickém pohledu – popisujícím chování objektů v čase pomocí pětic skládajících se z původního objektu, jeho události, jí způsobené zprávy, cílového objektu a jeho aktivity iniciované onou zprávou.

Nad tímto modelem můžeme dále nadefinovat operace objektové algebry, které nám umožní formalizovat manipulace s objekty (zejména dotazy).

1. Úvod

Softwarové inženýrství se pomalu stává součástí našeho každodenního života. Většina lidských aktivit používá software, který se tak postupně stává komplexnějším a nákladnějším. Poptávka po sofistikovaném software pak zvyšuje požadavky kladené na vývojové týmy. Z toho také vyplývá poptávka po kvalitních metodách.

Metoda definuje opakovatelný způsob získání spolehlivých výsledků. Metody obecně umožňují vystavět modely z elementů, tvořících základní způsoby popisu systémů. Metody také definují reprezentaci – často grafickou – která umožňuje jak snadnou manipulaci s modely, tak komunikaci a výměnu informací mezi různými částmi. Dobrá reprezentace usiluje o vyrovnanost mezi hutností a čitelností informací. Metody také nad modely a jejich grafickou reprezentací definují pravidla popisující výsledky různých pohledů, pořadí úkolů, nebo alokaci zodpovědnosti.

V současné době se můžeme setkat s nástupem objektově-orientovaných metod (OOM). Byl vytvořen Unified Modelling Language (UML), metamodel standardizující objektovou notaci (formálně popisující syntaxi i sémantiku notace). Tento metamodel umožňuje konstrukci modelu, standardizace procesu tvorby modelu však stále chybí. Také chybí prostředky pro formální zhodnocení kvality vytvořeného modelu (metriky).

Pokud by se podařilo objektově-orientovaný přístup formalizovat, potom bychom za pomoci této formalizace mohli jak popsat a porovnat jednotlivé procesy tvorby modelů, tak formalizovat hodnocení jejich kvalit. Uživatelé jednotlivých vytvořených modelů časem získávají určité zkušenosti, které nelze vždy jasně formulovat, či předat dále. I tyto zkušenosti by mohly být formalizovány a zapracovávány do vytvořených procesů.

Jednou z možností, jak formalizaci provést, je použití prostředků teorie množin – výsledkem bude *objektový model dat*.

Využití teorie množin se v objektovém přístupu jeví jako zvláště vhodné z několika důvodů. Jedním je blízkost konceptu množin a objektů. Instance objektů sdílí obecné charakteristiky třídy, které tvoří charakteristické vlastnosti množiny instancí. Pro tuto množinu a její charakteristické vlastnosti platí obdobné zákonitosti jako pro třídy a jejich elementy.

Koncept teorie množin byl také velmi úspěšně použit při tvorbě relačního modelu dat, pomocí kterého formalizujeme datové struktury při strukturovaném modelování. V tomto relačním modelu máme velmi silné prostředky pro popis sémantiky relačních jazyků (sloužících zejména pro manipulaci s daty) – relační algebru a relační kalkul, prostředky pro návrh vhodných relačních struktur (umožňující normalizaci), prostředky pro oddělení dat od jejich implementace.

Podobné prostředky existují i v objektovém modelu dat – popsáném v následující kapitole – nebo by z něj mohly být v budoucnu vytvořeny. Jejich využitím bychom vyřešili problémy nastíněné v úvodu článku.

Objektový model dat je také výhodný pro převod struktur a procesů z relačního modelu – vzhledem ke stejné struktuře relačního a objektového modelu dat (založených na množinové teorii) by měly být výsledky přenositelné (po úpravách například vztahujícím se k procesům).

2. Objektový model dat

Máme množiny

$$J = \{J \mid J \text{ je platné jméno}\}$$

$$D = \{D \mid D \text{ je doména}\}$$

$$Q = \{Q \mid Q \text{ je proces}\}$$

$$P = \{P \mid P \subset Q\}$$

Pak *objektovým schématem* (třídou) nazveme jakoukoliv čtveřici skládající se ze jména třídy C , jmen atributů J_1, J_2, \dots, J_n , příslušných domén D_1, D_2, \dots, D_n a z procesů P . Formálně zapsáno

$$C(J_1 : D_1, J_2 : D_2, K, J_n : D_n, P), \text{ kde}$$

$$C, J_1, J_2, K, J_n \in J$$

$$D_1, D_2, K, D_n \in D$$

$$P \in P$$

Přičemž jména J_i jsou unikátní, domény D_i se mohou opakovat. Dále si můžeme nadefinovat i množinu všech tříd (objektových schémat)

$$C = \{C \mid C(K) \text{ je objektové schéma}\}$$

Pro každé schéma $C \in C$ budeme dále uvažovat objekty, o kterých budeme hovořit jako o objektech nad objektovým schématem C , nebo jednodušeji jako o objektech ze třídy C . Tyto objekty budou dodržovat dané objektové schéma a budou odpovídat základním charakteristikám objektů – identita bude dána jménem objektu O , stav hodnotami atributů a chování množinou procesů z objektového schématu. Jelikož se hodnoty atributů objektů mohou v čase měnit, budeme uvažovat objekty v daném čase t . Formálně zapsáno

$$O_t^C(a_1, a_2, K, a_n), \text{ kde}$$

$$a_i \in D_i$$

$$C \in C$$

a množinu atributů objektů nad daným schématem (v daném čase) si označíme

$$M_t^C = \{(a_1, a_2, K, a_n) \mid O_t^C(a_1, a_2, K, a_n) \text{ je objekt v čase } t \text{ nad daným objektovým schématem } C\}$$

Někdy ovšem můžeme potřebovat pracovat i s objektem obecně (zejména při specifikacích platných po celou dobu života objektu), proto zavedeme i obecný „statický“ objekt

$$O^C = \{O_t^C \mid t \text{ je libovolný čas}\}$$

$$O = \{O^C \mid O^C \text{ je statický objekt}\}$$

$A(C_1, C_2, J_1, J_2, K_1, K_2)$, kde

$C_i \in \mathcal{C} \cup \mathcal{O}$

$A, J_i \in \mathcal{J}$

$K_i \in \mathcal{K}$, přičemž \mathcal{K} je množina kardinalit

$\mathcal{K} = \{0, 1, N, \infty\}$

Takto tedy můžeme formálně popsat objekty a třídy, neboli základní stavební elementy objektového přístupu. Pokud má být formalizace využitelná, musíme ještě nadefinovat nástroje, pomocí kterých popíšeme strukturu systému. V jazyce analýzy informačních systémů jde o logický pohled, a to jak o jeho statickou, tak o jeho dynamickou část.

Statickou část, kromě struktur tříd a objektů, tvoří i jejich vzájemné vazby. Obecnou vazbu mezi třídami si můžeme popsat pomocí jejího jména A , jmen obou tříd do vztahu vstupujících, názvu jejich rolí ve vztahu a kardinalit. Formálně

Rozeznáváme více druhů vztahů mezi třídami, ale všechny mohou být popsány pomocí této obecné šablony – například u vztahu generalizačně-specializačního budou jména rolí supertřída a subtřída, přičemž budou platit omezení pro atributy, domény a procesy ze supertříd $C_1(J_1^S : D_1^S, J_2^S : D_2^S, K, J_n^S : D_n^S, P^S)$ i subtříd $C_2(J_1^B : D_1^B, J_2^B : D_2^B, K, J_m^B : D_m^B, P^B)$ tato:

Dynamická část popisuje chování objektů v čase. Přičemž změny nastávají takto – nastane událost, která způsobí vyvolání zprávy, které může vyvolat aktivitu vedoucí ke změně stavu. Máme-li tedy

můžeme jeden vztah popsat jako uspořádanou pětici složenou z výchozího stavu jednoho objektu, událostí tímto objektem způsobené, zprávy, aktivity druhého objektu

$n \leq m$

$J_1^S, K, J_n^S = J_1^B, K, J_n^B$

$D_1^S, K, D_n^S \supseteq D_1^B, K, D_n^B$ $E = \{E | E \text{ je událost}\}$

$P^S \subseteq P^B$ (bez polymorfismu) $M = \{M | M \text{ je zpráva}\}$

(aktivity iniciované zprávou) a koncového stavu druhého objektu

Celý objektový model dat pak bude tvořen objekty, třídami, statickým i dynamickým

$D = (O1_{t_1}^{C1}, E, M, Q, O2_{t_2}^{C2})$, kde

$E \in \mathcal{E}$

$M \in \mathcal{M}$

$Q \in \mathcal{P}$, takového že

$C2(J_1 : D_1, K, J_n : D_n, P)$

(COSD), kde

$S = \{S | S \text{ je statický vztah}\}$

$D = \{D | D \text{ je dynamický vztah}\}$

pohledem, tedy

Konkrétní manipulace s objekty pak budeme moci vytvářet nad konkrétními objekty v čase, tedy nad množinami M_t^C , pro všechna dostupná C .

3. Objektová algebra

Pro manipulaci s objekty bude nezbytné popsat operace objektové algebry, která umožní formalizovat zejména dotazy a manipulace s objekty. Základní operace můžeme převzít z relační algebry a budeme je aplikovat na množiny atributů objektů nad danými schémata M_t^C .

Operace množinové sjednocení \cup , průnik \cap a rozdíl $-$ budou klasické množinové operace definované na množiny atributů daných objektů. Jedinou podmínkou bude kompatibilita operandů – tedy skutečnost, že do těchto operací mohou vstupovat pouze ty množiny objektů, jejichž schémata mají stejný počet atributů a jejichž odpovídající domény se navzájem rovnají.

Pro výslednou množinu hodnot atributů v daném čase musíme ještě vytvořit i odpovídající objektové schéma, v tomto případě lze převzít objektové schéma libovolného z objektů do operace vstupujícího.

Operace kartézský součin \times žádná omezení nemá, výsledné objektové schéma bude obsahovat atributy a domény z obou vstupních schémat, množina procesů v novém schématu bude sjednocením množin procesů v původních schématech.

Operace projekce $M_t^C[A]$, kde A je podmnožina množiny atributů třídy C , odstraní z n-tic hodnot atributů z množiny M_t^C atributy, které nejsou obsaženy v množině A . Výsledkem bude tedy množina odpovídající objektovému schématu s atributy z množiny A (a s odpovídajícími doménami) a s procesy zpracovávajícími také pouze tyto atributy (tedy z původního schématu odstraníme procesy, které zpracovávají atributy, které se nevyskytují v množině A).

Selekce $M_t^C(\phi)$ ponechá pouze ty prvky množiny M_t^C , které vyhovují podmínce ϕ (logický výraz porovnávající hodnoty atributů mezi sebou a s konstantami). Objektové schéma zůstane beze změny.

Spojení $M_t^C * M_t^D$ množin nad schémata C a D s množinami atributů A , B a s množinami procesů P a Q vytvoří nad schématem složeným ze sjednocení atributů $A \cup B$ a ze sjednocení procesů $P \cup Q$ maximální množinu takovou, že její projekce na atributy A je M_t^C a projekce na atributy B je M_t^D .

Kromě výsledků získatelných použitím těchto „tradičních“ operací existuje ovšem řadu dalších skutečností, které by nás mohly zajímat, které můžeme získat z objektového modelu, ale které těmito sedmi operacemi nezískáme. Vesměs jde o skutečnosti související se vztahy mezi objekty.

Pro jejich získání můžeme využít operaci *levý vztah*, která určí pro objekt (třidu) D množinu objektů (tříd), které jsou s ním ve vztahu a které odpovídají podmínce ψ , operace bude nadefinována takto:

$$D\{\psi\} = \{E \mid \text{existuje vztah } A(D, E, J_1, J_2, K_1, K_2) \text{ splňující podmínku } \psi\}$$

Obdobně nadefinujeme operaci *pravý vztah*, která pracuje stejně jako levý vztah, pouze pro druhý objekt (třidu) přítomný ve vztahu. Tedy

$$\{E\}D = \{D \mid \text{existuje vztah } A(D, E, J_1, J_2, K_1, K_2) \text{ splňující podmínku } \psi\}$$

Tímto způsobem můžeme zjistit například množinu všech supertříd pro danou třídu X
 $\{J_1 = \text{"supertřída"}\}X$

Operace levý a pravý vztah obsahují určitou nekompatibilitu vzhledem k „tradičním“ operacím – jejich výsledkem není množina hodnot atributů, ale množina tříd, případně objektů. Pro tyto výsledné třídy C ale máme množiny M_t^C , na které jsou už „tradiční“ operace aplikovatelné.

4. Závěr

Další rozšíření objektového modelu by mohlo být vytvořeno jako nástavba na zde publikovaném modelu a v souladu s notací popsanou v UML. Jde například o zahrnutí různých typů zpráv a jejich synchronizace, o rozlišení různých úrovní zapouzdření u tříd (public, private, protected), o omezování asociací a generalizací, o asociální kvalifikátory, nebo o zahrnutí celé koncepce polymorfismu.

Užitečné by bylo stanovení určitých omezení na procesy – omezení procesů na atributy příslušné třídy – tedy něco jako 1. objektová normální forma. Na ni by mohl navazovat popis dalších normálních forem omezujících vztahy a umožňujících vytvoření kvalitních struktur (podobně jako formy založené na funkčních závislostech u relací, ovšem rozšířeně o procesy).

U objektové algebry bude muset následovat komparace s dotazovacími jazyky, na jejímž základě teprve budeme moci kvalitu takto prezentovaného modelu vyhodnotit.

Literatura

1. Booch, G., Jacobson, I., Rumbaugh, J. The Unified Modelling Language User Guide, Addison-Wesley Pub Co, 1999
2. Eriksson, H. E., Penker, M. Instant UML, Wrox Press Ltd., Birmingham, 1997
3. Pezlar, Z., Rábová, I. Od De Marca k UML, a co dál?, výroční sborník MZLU, Brno, 1999 (v tisku)
4. Pokorný, J. Dotazovací jazyky, Science, Veletiny, 1994