

O JEDNÉ ZÁLUDNOSTI INTERAKCE «INCLUDE» V MODELU PŘÍPADŮ UŽITÍ

1. část

RNDr. Ilja Kraval, květen 2010

<http://www.objects.cz>

ÚVOD

Princip fungování interakce «include» mezi případy užití v USE CASE MODELU není vůbec složitý a je v podstatě poměrně dost lehce vysvětlitelný (viz například kniha [„Analytické modelování IS pomocí UML v praxi“](#), str. 115, bod 4.5.1, kde je také vysvětleno, kdy je nutno a kdy je pouze možno interakci «include» použít).

Pokud bychom chtěli interakci «include» vysvětlit pomocí nějaké srozumitelné programovací techniky, nejlépe by se hodilo srovnání s voláním jedné funkce druhou funkcí. Mimochodem takto k volání funkcí mezi sebou je tato interakce přirovnávána také ve specifikaci UML. V interakci «include» jeden případ užití prostě a jednoduše „zavolá“ druhý případ užití a tím jej použije.

Pokud používáte techniku psaní scénářů (což doporučuji), mělo by se také vyžadovat, aby byl ve scénáři volajícího případu užití v bodě „volání“ zaveden odkaz na daný volaný případ užití a v USE CASE DIAGRAMU by každému takovému odkazu ve slovním scénáři měla odpovídat jedna znázorněná interakce.

Příklad takového scénáře se zavoláním jiného případu užití může vypadat například takto:

...

Obsluha vybere osobu ze seznamu fyzických osob, viz UC [Výběr fyzické osoby ze seznamu fyzických osob](#)

...

V diagramu by se také měla objevit interakce «include» ve směru od tohoto případu užití s daným scénářem k druhému případu užití, který by se měl nazývat stejně, tedy v tomto případě „*Výběr fyzické osoby ze seznamu fyzických osob*“

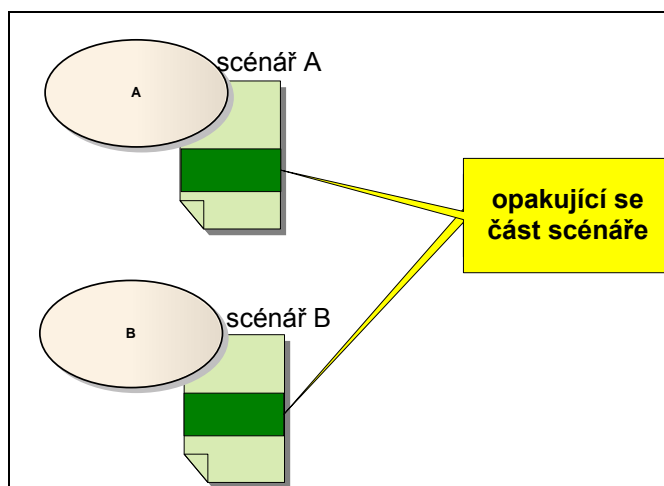
Situace je mišlím zřejmá, avšak právě v této jednoduchosti se skrývá nečekané úskalí! Přiznám se, že jsem v minulosti netušil, že právě v tak primitivní věci, jakou je interakce „volání“ mezi případy užití, se může skrývat nějaká záludnost. Ale, jak ukazuje praxe a konzultace v mnoha firmách a jak prozrazují také odpovědi účastníků na testy internetového školení „[Kurz profesního růstu analytika od základů \(distanční e-kurz\)](#)“, právě jednoduchost této interakce bývá nejčastějším zdrojem chyb při tvorbě USE CASE MODELU!

A právě o tom bude pojednávat tento článek.

KLASICKÉ SITUACE, KDE SE CHYBUJE

Paradoxně největší problémy se záludností interakce «include» se vyskytují ještě před jejím nalezením a to již v prvních krocích při samotném vyhledávání případů užití, kdy o této interakci nemáme ani potuchy. V té chvíli totiž máme před sebou „nepopsaný list papíru“ a případy užití teprve hledáme, což je opravdu velký rozdíl oproti situaci, kdy máme před sebou již hotový model.

Povšimněte si například, jak jednoduše vypadá model na následujícím obrázku, kde je vysvětlen princip interakce «include». Nejprve najdeme shodu ve scénářích dvou případů užití (viz např. obr. 4.4 zmíněné knihy [Analytické modelování](#) i s vysvětlením):



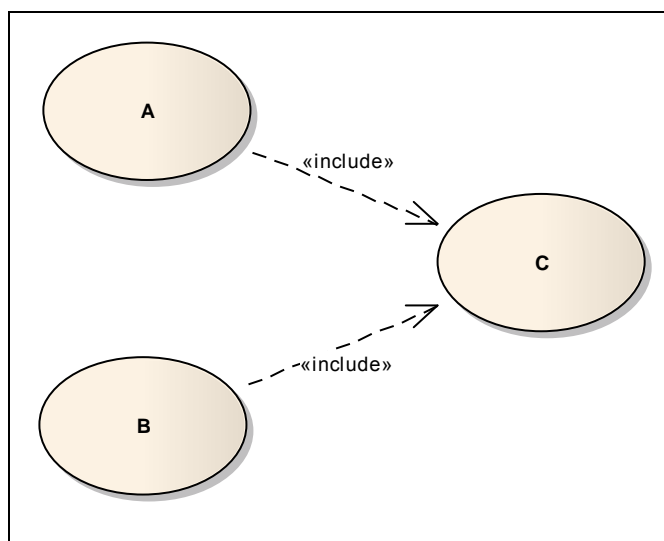
Obrázek 1 Podnět k zavedení interakce «include» při nalezení shody ve scénářích

Tuto situaci dobře známe z prvních kapitol školení: Je to situace „před vytknutím“ při aplikaci principu opětovné použitelnosti neboli re-use. Pokud tato situace nastane, jsme povinni část opakujícího se scénáře vytknout do nového případu užití pomocí interakce zvané «include».

Poznámka: Samozřejmě tuto situaci odhalíme pouze v případě, když dodržujeme zásady psaní scénářů: žádná synonyma a používat pouze ustálená slovní spojení.

Při zavedení interakce «include» postupujeme takto:

1. V diagramu zavedeme nový případ užití C a provážeme dvakrát interakcí «include» ve směru od A k C a od B k C takto:



Obrázek 2 Zavedení operace «include» v diagramu

2. Současně však provedeme i úpravy v textech scénářů: Opakující se část scénáře umístíme do případu užití C a na původních místech ve scénářích A a B použijeme větu vyjadřující „volání“ případu užití C, například „použije se případ užití C“ nebo „zavolá se případ užití C“ apod.

Tato situace je natolik jasná, že by se dalo předpokládat, že nenarazíme na problém. Avšak kámen úrazu zmíněné chyby se skrývá hned v první kapitole již zmíněné knihy [Analytické modelování v praxi](#), tedy v objektovém paradigmatu, jak si ukážeme v tomto článku.

Celý problém je v tom, že předešlý jednoduchý příklad na zavedení interakce «include» začíná dvěma již nalezenými případy užití, které jsou pro přehlednost nazvány úmyslně a názorně a zřetelně odlišně jako A a B. Jenomže v tomto příkladu jsme již jeden krok za daným úskalím a nebezpečím uvedené chyby. Jděme tedy o krok zpět a zamysleme se nad tím, jak se tyto dva příklady užití A a B našly. Tam se totiž nejčastěji chybuje, protože situace není na první pohled díky existujícímu re-use mezi případy užití až tak přehledná, jak by se nyní mohlo zdát.

Vyjdeme-li totiž ze zadání zákazníka, který popisuje celou „story“ okolo žádaného informačního systému a snaží se nám mnohdy dost neumětelsky sdělit, co vše by chtěl nechat naprogramovat, nemáme v první chvíli až tak úplně jasno, o kolika případech užití je v jeho sdělení vlastně řeč. Analytik musí znát určité postupy a mít určitou praxi v rozboru sdělení od zákazníka, aby byl schopen dobře díky těmto zadáním odhalit chování procesů podniku a také nalézt případy užití.

Tomuto problému je proto věnována náležitá pozornost i ve zmíněném [internetovém školení a testech](#), které se věnují problematice BPM a UCM. V daném školení se také někdy okolo tohoto problému rozpoutává vášnivá diskuse, právě díky tomuto úskalí.

Zkusme si zde vyjmenovat pro zajímavost několik takovýchto příkladů, kdy jsme ještě krok před nalezením případů užití. Z nich bude ihned jasné, že problém není až tak jednoduchý:

1. situace: Příjemka zboží

Budoucí uživatel systému v supermarketu sděluje svoje požadavky a od něj se dovíme následující „story“:

„K rampě přijíždí nákladní automobil se zbožím. Skladník má dvě možnosti: buď vezme do ruky zařízení typu PDA se čtečkou kódu, jde na korbu vozu a vytvoří příjemku zboží a to tak, že přijímá zboží snímáním čárového kódu, tj. založí hlavičku, a pak dokola provádí: sejme čárový kód, zadá počet balíků, a tak pokračuje, až vytvoří příjemku, anebo vezme do ruky

papír a tužku, protože nemá PDA, a vytvoří příjemku tak, že si vše zapíše na papír a pak to zadá do systému v kanceláři.“

Otázka zní: Bude tam jen jeden případ užití „Vytvoření příjemky na sklad“ anebo jich bude vícero? Dovedli byste jej (pokud je jeden) nebo je (pokud je jich vícero) pojmenovat?

2. Situace - Převodní příkaz v bance

Budoucí uživatel systému v bance sděluje svoje požadavky a od něj se dovíme následující „story“: „Chceme, abyste nám naprogramovali toto: Zákazník chce podat převodní příkaz a bude mít tyto možnosti:

- a) Klient napíše převodní příkaz na papírový formulář, jde na přepážku a předá jej obsluze. Ta jej zadá do systému.
- b) Klient přistoupí k tzv. kiosku (obdoba bankomatu), ten už je připraven jako externí zařízení od dodavatele. Přes kiosek klient zadá převodní příkaz (identifikuje se svou čipovou kartou).
- c) Klient nepůjde vůbec do banky a zadá převodní příkaz přes internet banking.
- d) Klient napíše převodní příkaz na papírový formulář, vhodí jej do boxu a obsluha v bance po otevírací době vysype box a zadá převodní příkaz do systému.

Opět následují podobné otázky: Na co byste se museli po tomto sdělení ihned zeptat, aby bylo úplně jasno? Bude tam jen jeden případ užití „Vytvoření převodního příkazu“ anebo jich bude vícero? Jaké případy užití tam budou, kolik jich bude a jak se budou jmenovat?

Pokračování příště

Poznámka: Pokračování tohoto článku plánujeme cca asi za týden, ale pokud chcete, můžete se zúčastnit diskuse nad těmito příklady již nyní. Diskusi najdete na adrese

<http://www.objects.cz/forum>

Věnujte pozornost novému termínu

Kurz profesního růstu analytika od základů (distanční e-kurz) startuje již od 1.8.2010!

- Nyní cenově výhodněji pro jednotlivce
- Výrazná sleva pro rychle přihlášené

Blíže [viz zde](#)