

Posudek oponenta diplomové práce

Michal DANIHELKA: Úložiště pro rozvolněné objekty

Cílem této práce bylo na základě článku, popisujícího koncept tzv. rozvolněných objektů navrhnout a implementovat prostředí pro práci s nimi nad databází Oracle.

Autor zvolil implementaci ve formě knihovny v jazyce Java verze 1.6 a vyšší, která zapouzdřuje vlastnosti rozvolněných objektů a poskytuje programátorovi rozhraní pro manipulaci s nimi. Dále autor implementoval zkušební aplikaci nad touto knihovnou a provedl testy, měřící výkon takto napsané aplikace.

Za kladný rys předkládané práce považuji rozšíření, dovolující simulaci polymorfismu metod, který nebyl v původním konceptu nijak popsán a byl naopak explicitně vynechán a neřešen. Vhodné je rovněž zajištění kontroly splnění vstupních podmínek (náležení do implicitní třídy) před vykonáním každé metody. Pozitivním rysem knihovna je také schopnost spolupráce s volně šiřitelnou databází PostgreSQL.

Zatímco s programovým řešením jsem spokojen, výtky mám k psanému textu. Text práce dle mého názoru poněkud nepřehledný a jeho přeuspořádáním by se zvýšila srozumitelnost i pro čtenáře, který původní článek nečetl.

Kapitola 2 „Základní koncepty rozvolněných objektů“ začíná příklady aplikací a vlastnostech výsledného řešení, aniž by byl model nejdříve popsán. Souvislosti tak čtenář získá až po přečtení další poloviny práce. Spíše bych očekával popis těchto aplikací spolu s předpokládanými problémy v klasickém OO modelu, následně popis modelu samotného a teprve poté důsledky tohoto přístupu na řešení zmíněných problémů.

Obdobně sekce 2.2 seznamuje se souvisejícími pracemi, aniž by byly tyto souvislosti zřejmé. Vhodnější by bylo související práce uvádět vždy až v době popisu konkrétního problému jako jednu z variant řešení a diskuzí výhod/nevýhod tohoto přístupu ve srovnání s ostatními. Například zmínka o *C-Store* má smysl až v okamžiku diskuze ukládání rozvolněných objektů do databáze.

Na obrázku 2.3. na straně 9 by zřejmě neměla být vazba *can execute* mezi *objekty* a *metodami*, ale naopak vazba *fulfill* mezi *objekty* a *podmínkami*. Znázorněná vazba je – předpokládám pouze odvozená. Navíc popis toho, co podmínky jejich aplikovatelnost na instance objektů znamenají, jsou až v sekci následující.

Popis emulace polymorfismu, tedy toho, jakým způsobem je k dané instanci na základě datového obsahu vybrána metoda s nespécifičtější podmínkou, by byla čtenáři rozhodně srozumitelnější, kdyby byla na straně 10 popsána formálnější formou nějakého pseudo-algoritmu, než volnými slovy přímo v textu.

Vzhledem k tomu, že byl navržen zcela nový framework, očekával bych, že v práci nebo alespoň na příloženém CD najdu podrobný popis instalace knihoven a potřebných nástrojů, nastavení cest a parametrů a práce s frameworkem. Pomohl by i podrobný popis toho, jak zopakovat příložené testy.

Přes to, že text práce vykazuje výše zmíněné nedostatky a popis řešení na textové úrovni nedosahuje kvality odevzdaného softwarového díla, domnívám se, že autor splnil podmínky kladené na diplomové práce a doporučuji proto práci k obhajobě.

V Praze dne 31. 8. 2009

RNDr. Michal Kopecký, Ph.D.
KSI MFF UK

Dotaz:

- Čím je dán nárůst doby zpracování v SRO? Není problém v nevhodném naplánování dotazu na serveru?