

Posudek oponenta diplomové práce

Jméno a příjmení autora posudku: RNDr. Jiří Helmich

Jméno a příjmení autora práce: Bc. Jan Navrátil

Název práce Integrace legacy databází do soudobých informačních systémů

Text posudku

Předkládaná diplomová práce si klade za cíl navrhnout a implementovat universálně použitelný framework, který by umožnil integraci informací z legacy databázových aplikací do soudobých informačních systémů, tedy například i jejich postupnou migraci na soudobé databázové technologie. Jedná se o aktuální a zajímavý problém vycházející z praxe. Je tak naprosto logické, že řešitel přistoupil ke zhotovení případové studie s cílem aplikovat a evaluovat navržené a implementované řešení v praxi a demonstrovat tak jeho funkčnost.

Řešitel předkládá modulární řešení s konfigurovatelným jádrem. Jádro je znovupoužitelnou knihovnou, která umožňuje dalším programátorům integrovat několik různých databázových systémů do jedné aplikace. Těmto programátorům tak stačí implementovat moduly specifické pro jejich datovou doménu a připravit konfiguraci jádra. Jádro pak skryje rozdíly mezi jednotlivými databázemi a jeho uživatel během migrace mezi různými databázemi nemusí upravovat kód svojí aplikace.

Kladně hodnotím využití existujících technologií pro objektové mapování a konfiguraci (NHibernate), neboť se v dané kategorii jedná o průmyslový a dobře známý standard, čímž se zvyšuje přidaná hodnota práce. Pozitivně hodnotím také existenci API pro programatickou konfiguraci jádra. Vítám také existenci tzv. defaultního modulu, který jádro používá pro vykonání metody v případě, že v konfiguraci pro tuto metodu nenažde explicitní mapování.

Za slabinu práce považuji především velice stručnou programátorskou dokumentaci s ne příliš detailně komentovanými příklady, u znovupoužitelného řešení bych očekával detailnější manuál. Kód obsahuje rozsahlé zakomentované fragmenty a nerespektuje žádný pevný styl zápisu (např. míchání českých a anglických fragmentů). Kód samotný se tedy nedá považovat za příliš dobrý návod pro další programátory. Moduly implementují jednotné rozhraní, které se stává z několika málo obecných metod (např. `GetByID`), uvnitř kterých se pomocí programátorského konstrukturu *switch* rozhoduje, jaký dotaz se nad související databází vykoná. Tuto část by patrně bylo možné moderními programátorskými prostředky zvládnout lépe.

Nové návratové typy (objekty) a dotazy je třeba implementovat úpravami aplikace (vytvoření nové třídy) a následně upravit konfiguraci jádra tak, aby aplikace fungovala dle očekávání. Řešení

obsahuje malou ukázkovou aplikaci, která demonstruje některé schopnosti jádra. Řešitel demonstroval funkčnost předloženého řešení.

K práci mám následující otázky:

- Jakým způsobem může jiný programátor využít existující moduly vaší aplikace?
- Existuje v aplikaci nějaký mechanismus pro přepnutí mezi více konfiguracemi jádra?

Doporučení k obhajobě

Z výše uvedených důvodů práci *doporučuji* k obhajobě.

Soutěž studentských prací

Vynikající práce vhodná soutěže studentských prací: **NE**.

V Praze dne 22. 8. 2014

Podpis: