

Posudek vedoucího diplomové práce

Jméno a příjmení autora posudku: **RNDr. Michal Kopecký**

Jméno a příjmení autora práce: **Bc. Jan Navrátil**

Název práce: **Integrace legacy databází do soudobých informačních systémů**

Vlastní text:

Cílem této práce bylo navrhnout a implementovat framework, který by umožnil integraci informací z legacy databázových aplikací jako například FoxPro a podobné, do soudobých informačních systémů a jejich postupnou migraci. Rozhraní frameworku by měla být navržena s ohledem na bezpečnou souběžnou manipulaci s daty více uživateli s obecně rozdílnými oprávněními.

Framework by měl rovněž počítat s možností havárie jakékoli komponenty, výskytem chybných (nekonzistentních) dat v databázi.

Návrh vychází z analýzy konkrétního ne zcela triviálního problému, který se vyskytl v informačním systému, který z historických důvodů používá několi aplikací, využívajících několik databází včetně částí a modulů ve FoxPro a dalších komunikujících s SQL databází Firebird.

Základním požadavkem tedy bylo navrhnout framework tak, aby dokázal na jedné straně usnadnit vytvoření moderní třívrstvé aplikace s webovým rozhraním a používajícím některou ze dnes běžných SQL databází, a zároveň dokázal komunikovat s moduly a daty uloženými v historických úložištích, kde není žádoucí provádět žádné změny datového modelu, ani žádné přímé vnější zásahy do dat.

Přenositelnost řešení nebyla požadována, a tak autor práce zvolil s ohledem na plánované pilotní nasazení za základní prostředí, pro které bude aplikace vyvinuta platformu Microsoft .Net a za hlavní jazyk pro vývoj a C#.

Místy autor věnuje prostor popisu aplikací nad samotným frameworkem, místo aby se zaměřil na detailnější popis frameworku samotného, a detailnímu popisu všech rozhraní mezi navrženými moduly a podrobnější diskusi možných alternativ k realizovanému řešení. Leckdy se tak jen obtížně zjišťují detaily implementace nebo motivy, které k nim vedly.

Za rozumné považuji rozčlenění frameworku na jádro a back-end moduly, které komunikují s danou databází. Stejně tak je vhodné data konvertovat na objekty a zpět, což odstíní řadu rozdílů v práci s databázemi. Řešení tak má potenciál umožnit postupně přehazovat funkcionalitu běžící nad jednou databází do jiné databáze.

Za problematické ale považuji naopak to, že v textu práce chybí diskuze toho, nakolik je jádro konfigurovatelné s ohledem na strukturu datového modelu, který obsluhuje. Očekával bych, že primárně bude model dat v databázi (případně včetně sémantiky) popsán nějakým konfiguračním souborem. V ideálním případě by mělo jít získat u běžné databáze konfiguraci přečtením v ní uložených metadat. Z popisu v samotném textu práce se zdá, že při změně dat, nad kterými framework pracuje, bude potřeba měnit kód frameworku. To souvisí s tím, že pro daný objekt framework nebo příslušný modul implementuje odpovídající getter a setter. Práce však nezmiňuje to, co bych očekával, tedy nějaký obecný getter pro objekt pojmenovaného typu který by na základě konfigurace vše vyřešil transparentně.

Dovedu si představit i smíšené řešení, kdy se bude k většině většina dat přistupovat jednotně na základě konfigurace a pro speciální případy bude kód pro čtení (a především pro zápis) do legacy databáze napsán přímo ve specifickém modulu. V textu jsem ale bohužel nic takového popsaného nenašel.

Framework tak ve stávající na jedné straně podobě plně řeší požadavky konkrétní situace konkrétní firmy na přístup k datům. Na druhé straně se z vlastního textu práce jen těžko orientuji v tom, jak framework detailně funguje. Především s ohledem na flexibilitně získávání stejných dat z jiné databáze nebo naopak jiných dat ze stejné (stejného typu) databáze.

Mám proto na diplomanta řadu dotazů:

- Jak je řešen převod dané funkcionality z jedné (řekněme Legacy) databáze do druhé (moderní), nakolik je to záležitost konfigurace a nakolik kódu? Pokud je nutná změna kódu, jaké to má důvody?
- Jak moc je záležitostí konfigurace překlopení celého schématu z jedné moderní databáze do jiné moderní databáze, pokud budu v databázi uvažovat jen hloupé úložiště (pouze tabulky)?
- Pokud bych uvažoval aplikaci psanou od základu v .Net nad moderní databází bez využití daného frameworku, jak moc musím aplikaci změnit, abych mohl využít přístup i k nějakým legacy tabulkám?

Doporučení k obhajobě:

I přes řadu výše uvedené výtky *doporučuji práci* k obhajobě.

| | |
|---|------------------------------|
| Vynikající práce vhodná pro soutěž studentských prací | ANO <input type="checkbox"/> |
|---|------------------------------|

Seznam soutěží studentských prací, viz <http://www.mff.cuni.cz/studium/bcmgr/prace/>

Pokud jste výše zaškrtnli ANO, zdůvodněte prosím svůj návrh, případně uveďte konkrétní soutěž, pro kterou je práce vhodná (rámeček lze nechat prázdný, pokud za dostatečné zdůvodnění považujete text posudku):

| |
|--|
| |
|--|

V Praze dne: 10. 6. 2016

Podpis: