



Dokončení projektu Migdb Bc. Martin Lukeš

2015

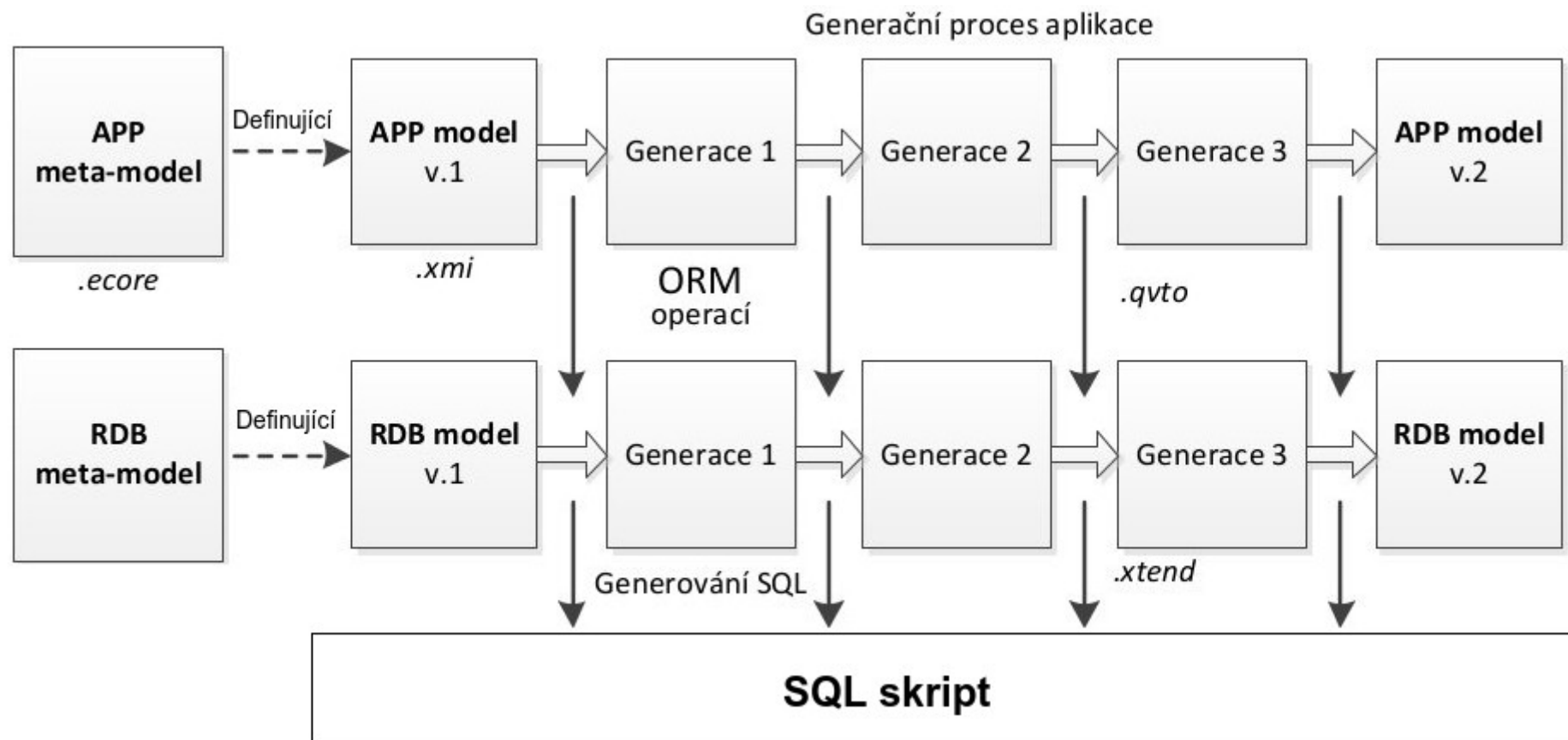
Projekt Migdb

- Zabývá se transformací změn aplikačního modelu na změny modelu databázového (SQL)
- Využito paradigma MDA
- 2012 prezentován v Cambridge na konferenci Code Generation

Cíle diplomové práce

- Dokončení framework Migdb
- Prozkoumání tématu Rozpoznávání operací

Framework Migdb - schema



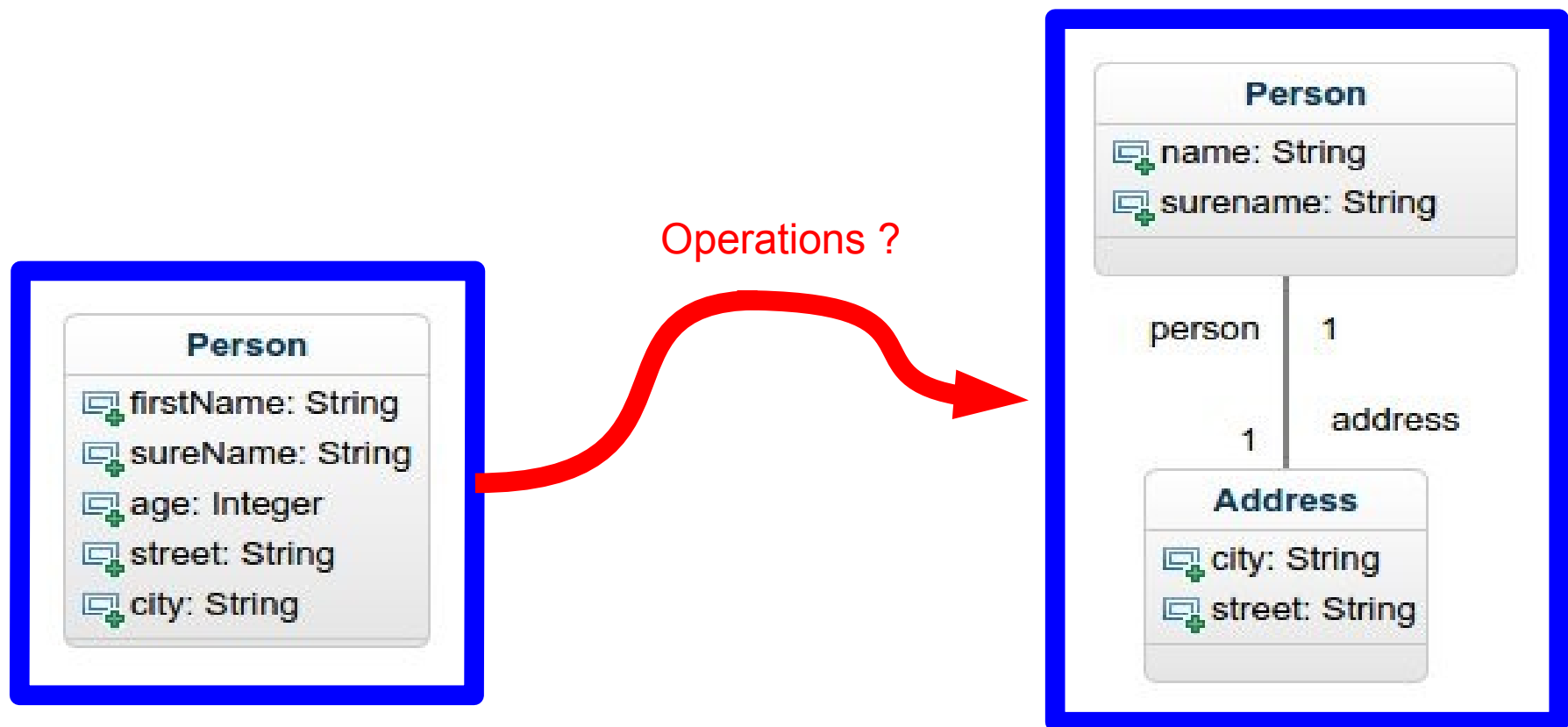
Dokončení frameworku - úkoly

- Úprava metamodelů
- Implementace ORM transformací aplikačních operací
- Testování
- Úprava generátorů kódu

Rozpoznávání operací – definice problému

- Mějme zdrojový aplikační model S a cílový aplikační model T
- Hledáme sekvenci operací Operations, jejíž aplikace transformuje model S na model T
- Automatizace
- Zachování dat

Rozpoznávání operací



Rozpoznávání - úkoly

- Rešerše
- Popis, implementace a otestování 2 různých algoritmů
- Definice energie modelu a vzdálenost dvou modelů
- Definice Diff metamodelu

Energie a vzdálenost modelů

- Vzdálenost (S, T) – celočíselná hodnota, maximální počet operací, které transformují model S na model T
- Energie $(M) = \text{Vzdálenost}(M, \text{Empty}) = \text{Vzdálenost}(\text{Empty}, M)$
- Energie (M) – celočíselná hodnota, určuje maximální počet operací potřebný k vytvoření nebo „zničení“ modelu M

Stavový algoritmus

- Algoritmus založený na minimalizaci vzdálenosti dvou modelů
- V každé iteraci najde pro každou třídu operací nejvyšší hodnotu energie, o kterou by mohla aplikace instance třídy operace přiblížit aktuální model modelu cílovému
- Nejvýhodnější operace je aplikována

Energie – základní vzorec

$$E(M_1) = \sum_{C \in M_1} E(C) = \sum_{C \in M_1} (1 + \sum_{P \in C} 1)$$

- Algoritmus dokáže rozpoznat strukturální operace

Vzdálenost – vzorec

$$\begin{aligned} distance(M_X, M_Y) &= E(\Delta(M_X, M_Y)) + E(\Delta(M_Y, M_X)) = \\ &= E(C_{XY}, P_{XY}) + E(C_{YX}, P_{YX}) + E(PRIM_{XY}) = \\ &= E(C_{XY}) + E(P_{XY}) + E(C_{YX}) + E(P_{YX}) + \\ &\quad + E(PRIM_{XY}) \end{aligned}$$

- Algoritmus dokáže rozpoznat všechny operace
- Personalizovatelnost

Párovací algoritmus

- Algoritmus založený na párování entit stejného typu
- Algoritmus inspirovaný články o Model Matchingu (Cincetty)
- Párování provedeno v 1. fázi pomocí shodnosti identifikátoru a v 2. fázi pomocí podobnosti entit

Podobnost entit

- Standardní algoritmy párují 1 entitu ze zdrojového modelu s 1 entitou z cílového modelu
- M x N párování
- Ostatní dokážou AddClass, AddProperty, RemoveClass, RemoveProperty
- My dokážem víc

Porovnání algoritmů

- Párovací algoritmus je efektivnější
- Stavový je paralizovatelnější
- Stavový algoritmus je personalizovatelný
- Stavový algoritmus dokáže řešit víc instancí úlohy, složitější na implementaci, definici energie...
- Párovací algoritmus má problém s rozpoznáním operace RenameClass, RenameProperty

Děkuji za pozornost

- Popis a experimentální implementace dvou rozpoznávacích algoritmů
- Naimplementování ORM operací
- Otestování nové i staré funkcionality
- Definice Vzdálenosti modelů a Energie modelu