Problematika transformace datových modelů do vývojového prostředí

Jan Dujka

Ústav automatizace a informatiky, VUT v Brně, 616069 Brno, Česká Republika

Abstrakt
Tento příspěvek si klade za cíl, seznámit zájemce se zkušenostmi s využitím nástroje Pdbridge v oblasti vzájemného propojení technologické úrovně aplikace vyvíjené v prostředku CASE/4/0 a implementační úrovně této aplikace, tedy schéma PROGRESS databáze.

ÚVODNÍ POZNÁMKA

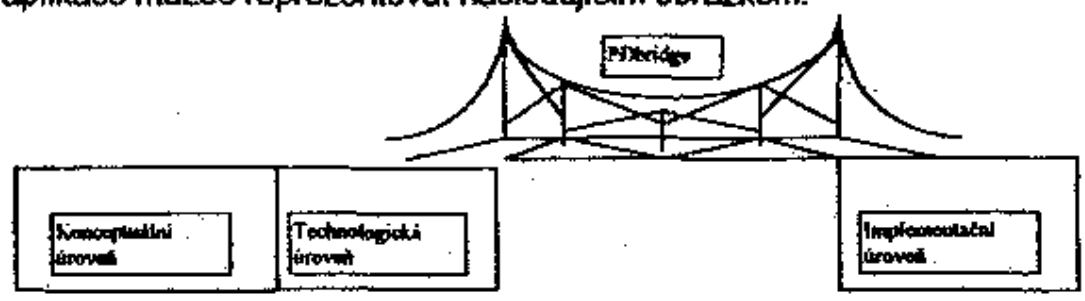
Jelikož jsou LOWER CASE, které již umí generovat kód, implementačně závislé (uvažme například závislost na hardwarových prostředcích, na operačním systému či databázovém prostředí), snižuje se značně počet zájemců o tyto case systémy. Toto se samozřejmě odráží i na trhu software, kde jsou tyto nástroje zastoupeny jen ve velmi omezeném počtu. Jsou příliš konkrétní a jejich cena je vysoká. Opticky se zdá, že jsou tyto nástroje v útlumu. Je však nutné si uvědomit, že reálnou produktivitu dělá právě to vlastní generování aplikace. Case vyšší třídy (MIDDLE CASE či dokonce UPPER CASE) se podílejí pouze na zvýšení kvality výsledného softwarového produktu. Můžeme tedy konstatovat, že cena LOWER CASE je jen zdánlivě vyšší, poněvadž tyto case nejen zajišťují zvýšení kvality výsledného produktu, ale především významně zkracují dobu realizace.

Dnes se již upustilo od samotných 4GL a vznikají nové nástroje GUI, které jsou charakteristické především možností grafického programování a kde přímo vytvoříme aplikaci (CENTURA TEAM DEVELOPE, POWER BUILDER, VISUAL BASIC). LOWER CASE pak generuje výstup do jazyka, který používá toto vývojové prostředí.

Protože jsou tyto case na trhu zastoupené jen velmi málo, málo se též o nich píše. Proto jsem si dovolil napsat následující příspěvek.

CO JE TO PDBRIDGE?

Pdbridge je softwarový produkt, dodávaný firmou ITC PragoData a.s., jehož cílem je překlenout propast mezi vývojem aplikace v case/4/0 a vývojem v PROGRESSu.
Vývoj aplikace můžee reprezentovat následujícím obrázkem:



- *Konceptuální úroveň* obsahuje řešení provblémové oblasti nezávisle na použitém technologickém prostředí. V case/4/0 užijeme pro znázornění této úrovně ER model.
- *Technologická úroveň* již zohledňuje vliv použitého technologického prostředí. V case/4/0 ji realizujeme pomoci relačního modelu dat.
- *Implementační úroveň* představuje konkrétní definici struktury dat v daném implementačním prostředí, v našem případě tedy v PROGRESSu.

Z obrázku plyne, že Pdbridge umožňuje propojení mezi technologickým modelem dat v case/4/0 a PROGRESSovskou databází při zajištění konzistence a možnosti oboustranného přechodu mezi oběma modely, tedy :
- generování schématu databáze na základě relačního modelu dat v case/4/0,
- zpětnou analýzu databáze a zdokumentování její struktury formou relačního modelu dat v case/4/0,
- iterativní použití, které kopíruje životní cyklus vývoje aplikace, kdy jsou změny na jedné straně periodicky promítány na stranu druhou a naopak .

VZÁJEMNÝ VZTAH MEZI DATOVÝM MODELEM V CASE/4/0 A V PROGRESSU

1) Relační model dat v case/4/0

Objekty reálného světa jsou v relačním modelu zobrazeny jako *relace*. Relace může mít kromě svého jména definováno i jméno technické (pro přenos do PROGRESSu).

Relace jsou charakterizovány svými *atributy*. Každý atribut má přiřazen tzv. *Data Element*, přičemž platí, že jeden Data Element může být sdílen i několika atributy. Data Element je nositelem vlastností, jako jsou např. datový typ, formát, validační podmínka, inicializační hodnota. Každá relace má definován tzv. *primární klíč*, což je atribut nebo skupina atributů, které slouží k jednoznačné identifikaci jednoho řádku relace.

Relace mohou realizovat vztahy s jinými relacemi. Relace, která je ve vztahu primární, jednoznačná , se označuje jako *master*, relace, která je ve vztahu závislá se označuje jako *detail*. Vztahy mají definovanou kardinalitu (1:1, 1:N). Vztahy jsou realizovány tak, že primární klíč relace master je obsažen v relaci detail jako *cizí klíč*. V case/4/0 je obecně možné, aby korespondující atributy vztahu měly rozdílné jméno. Obvykle mívají stejný popis, protože sdílejí společný Data Element.

Nad relacemi lze definovat přístupové indexy. Indexy mohou být tvořeny jedním nebo více atributy relace a mohou být označeny příznakem unikátnosti. V definici indexů však nelze vyjádřit pořadí polí v rámci indexu.

2) Relační model dat v PROGRESSu.

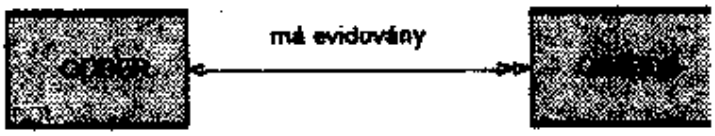
Základním stavebním prvkem relačního modelu PROGRESSu je *tabulka* (do verze 6 File, od verze 7 Table). Tabulka má definována nějaká *pole* (Fields). Pokud jsou vlastnosti tabulek a polí popsány přímo v databázovém schématu, pak jsou v aplikačních programech automaticky používány jako default.
Nad poli jednotlivých tabulek lze definovat indexy (unikátní, neunikátní). Cílem užití indexů v PROGRESSu je :

- zajistit jednoznačnost řádků tabulky,
- realizovat vztahy mezi tabulkami,
- urychlit přístup k řádkům tabulky,
- zajistit výběr a zpracování řádků tabulky v určitém pořadí.

Vztahy mezi tabulkami nejsou definovány explicitně. Vztah existuje, jsou - li jména polí unikátního indexu tabulky 1 obsažena v jiné tabulce 2. Je-li nad těmito poli v tabulce 2 také unikátní index, pak jde o vztah typu 1:1. Je-li nad výše zmíněnými poli neunikátní index nebo žádný index, pak jde o vztah typu 1:N.

Obecně lze říci, že pro realizaci vztahu je nutné, aby korespondující pole v obou tabulkách měla stejná jména. (Mohou však mít - a obvykle mívají - v obou tabulkách rozdílný popis. Při definování více unikátních indexů nad jednou tabulkou , může každý unikátní index vystupovat ve vztazích jako primární klíč.

Vzájemná korespondence jmen polí mezi dvěma tabulkami umožňuje užívat na straně PROGRESSu při vyhledávání frázi „OF" např. :



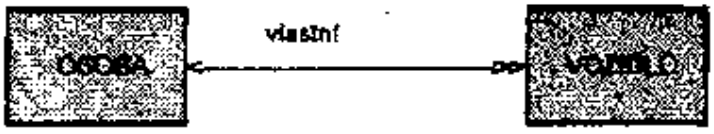
cis_odb
cis_obj

cis_obj
datplat_od

```
FOR EACH ODBER:
        DISPLAY jmeno kredit.
        FOR EACH OBJEDN OF ODBER:
        DISPLAY cis_obj datplat_od .
        END.
END.
```

Kde ODBER je tabulka odběratelů a OBJEDN je tabulka objednávek.

Vztahy při vzájemné nekorespondenci jmen polí dvou tabulek, které se realizují přímo v aplikačních procedurách pomocí klauzule „WHERE", nebudou z popisu dat zjistitelné např. :



cis_RP
jmeno

cis_SPZ
cis_mot
majitel

```
PROMPT-FOR OSOBA.jmeno.
FIND OSOBA USING jmeno NO-LOCK NO-ERROR.
        DISPLAY OSOBA.
```

```
FIND VOZIDLO WHERE VOZIDLO.majitel = OSOBA.jmeno.
     DISPLAY VOZIDLO.
```

Kde OSOBA je tabulka osob a VOZIDLO je tabulka evidovaných vozidel.

V tomto případě platí, že:
- při výpisu vztahů v PROGRESS Data Dictionary nebude tento vztah detekován,
- při zpětné analýze databáze pomocí Pdbridge nemusí být tento vztah nalezen,
- v kódu 4GL pro vyhledávání řádků související tabulky nelze použít frázi „OF".

3) Vzájemná transformace datových modelů mezi case/4/0 a PROGRESSem.

Relaci v case/4/0 odpovídá tabulka na straně PROGRESSu. Předpokládá se, že tabulka a relace mají stejná jména.
- jméno sémantické, které je zobrazeno v case/4/0,
- jméno technické, které představuje název cílové tabulky v databázi.

Atributu relace odpovídá v PROGRESSu pole tabulky. Opět předpokládáme, že mají stejná jména.

Dále platí:
- Pořadí polí v rámci tabulky (Order) odpovídá pořadí atributů v rámci relace.
- Datový typ pole (Data Type) odpovídá datovému typu Data Elementu.
- Rozsah opakovaného pole (Extent) odpovídá výrazu v datovém typu Data Elementu.
- Povinnost (Mandatory) pole odpovídá příznaku atributu Not Null.
- Formát zobrazení pole (Format) odpovídá specifikaci Picture Data Elementu.
- Počáteční hodnota pole (Initial Value) odpovídá specifikaci Default Value Data Elementu.

Ostatní vlastnost polí budou při transformaci do PROGRESSu nastaveny na hodnoty PROGRESS default a dále mohou být měněny přímo v Data Dictionary v PROGRESSu.

Chceme-li vztahy z case/4/0 realizovat v PROGRESSu a naopak, je nutné splnit tyto podmínky:
- korespondující dvojice atributů / polí (primární klíč a cizí klíč) mají stejná jména,
- nad atributy / poli primárního klíče je definován unikátní index,
- je-li nad cizím klíčem definován unikátní index, pak se jedná o vztah 1:1
- je-li nad cizím klíčem definován neunikátní nebo žádný index, pak jde o vztah 1:N.

Platí, že vztahy, u nichž nesouhlasí jméno primárních a cizích klíčů, nebudou rozpoznány. Cizí klíč sdílí defaultně v case/4/0 společný Data Element s primárním klíčem. Je možno definovat ručně každému samostatný Data Element. Ve směru z PROGRESSu jsou vytvářeny automaticky pro primární a cizí klíče samostatné Data Elementy.

Při generování schématu PROGRESSovské databáze jsou Pdbridgem automaticky vytvořeny unikátní indexy nad primárními klíči všech vztahů. U vztahů 1:1 (viz.

relační model case/4/0) se při generování PROGRESSovské databáze vytváří automaticky unikátní indexy nad cizími klíči vztahů.U vztahů typu 1:N není z relačního modelu dat jasné, zda indexy nad cizími klíči vytvořit či nikoli. Proto je tato tvorba volitelná. Nad cizími klíči lze indexy definovat trojím způsobem:

- nastavením parametru „Indexes on Foreign Keys" na hodnotu „YES",
- ručním definováním vybraných indexů v case/4/0,
- ručním definováním indexů v PROGRESSu v Data Dictionary.

Dále platí, že:

- Pdbridge vytvoří automaticky unikátní index nad celým primárním klíčem relace, která nevstupuje do žádných vztahů,
- Pdbridge vytvoří automaticky unikátní indexnad celým primárním klíčem relace, která vstupuje do vztahů pouze na cizích stranách vztahů.

Ve zpětném směru (z PROGRESSu do case/4/0) se do case/4/0 přenášejí veškeré indexy definované v PROGRESSu. Pokud je to možné, jsou jména indexů generovaných *Pdbridgem* odvozena od jmen indexů v case/4/0. Nastane-li situace, že je index vytvořen přímo Pdbridgem, pak je jeho jméno tvořeno jménem tabulky a pořadovým číslem. Pořadí polí indexu je dáno pořadím atributů v rámci relace na primární straně vztahu.

LITERATURA

[1] CASE/4/0, Příručka k české verzi, ITC PragoData, 1996
[2] Pdbridge, Příručka uživatele, ITC PragoData, 1996
[3] PROGRESS, Základní učebnice 4GL, Progress Software Corporation, 1990
[4] PROGRESS, Referenční příručka, Progress Software Corporation, 1990
[5] Tvorba software „95", sborník semináře, Tanger s.r.o., 1995
[6] Business Re-engineering with Information Technology, J.J. Donovan, Cambridge Technology Group, 1994
[7] CASE/4/0, BenutzerLexikon1, microTOOL GmbH, 1994
[8] CASE/4/0, RepositoryHandbuch, microTOOL GmbH, 1994
[9] CASE/4/0, MethodenHandbuch, microTOOL GmbH, 1994
[10] CASE/4/0, BenutzerHandbuch, microTOOL GmbH, 1994
[11] CASE/4/0, BenutzerLexikon2, microTOOL GmbH, 1994