

Progresivní technologie v malých softwarehousech

Šebesta Ivo

BM Servis s.r.o., Školní 3, 370 10 České Budějovice, Česká Republika

1. ÚVOD

Tento příspěvek se bude zabývat technologií tvorby software v malém softwarehouse. Pojmem malý softwarehouse je zde míněno, že na procesu tvorby software se podílí tým, kdy počet členů týmu nepřesáhne počet 10 a kdy tento tým má za úkol vypracovat poměrně rozsáhlý systém, podponující např. komplexní řízení středního až většího podniku včetně ekonomiky, mzdové a personální agendy a správy majetku. Cílem při tvorbě takového systému je zorganizování lidských zdrojů a technických prostředků tak, aby byly využity optimálním způsobem, aby bylo dosaženo vysoké kvality konečného produktu v co nejkratším termínu.

2. POUŽITÉ METODIKY A TECHNIKY

K plnění stanovených cílů popsanych v úvodu je přistupováno tak, aby bylo dosaženo kvality s přihlednutím k normám ISO 9000. Cílem je vytvoření rozsáhlého systému řízení podniku. Cesta k tomuto cíli je rozdělena do relativně samostatných částí - projektů a každý projekt je samostatně řízen metodou řízení projektů podle norem ISO 9000. Bylo rozhodnuto, že při řešení každého projektu bude používána osvědčená metodika LBMS hierarchické struktury rozkladu prací (WBS). Ta poskytuje doporučení pro sledování celého životního cyklu projektu. Bylo přitom bráno do úvahy, že celý úkol bude plněn přínústkovým způsobem z důvodu velkého objemu dodávaných prací. Proces tvorby je rozdělen do etap, které se rozpadají na kroky a ty dále na úkony. Pro každou etapu se definují dokumenty a produkty, které umožňují zahájit další etapu. Každá etapa je ukončena kontrolním krokem. Smyslem tohoto všeho je dodržování kvality výsledného produktu, který z projektu vznikne. V průběhu realizace se ukázalo, že ne všechny kroky metodiky LBMS budou prováděny a některé kroky budou modifikovány tak, aby vyhověly požadavkům i způsobu práce řešitelů. Metodika LBMS sama uvádí možnost přizpůsobit se v některých situacích projektu či řešitelskému týmu. Už také proto, aby nevznikaly zbytečné dokumenty a metodika nebyla pouze pro metodiku.

Pro každý projekt je určen vedoucí projektu, který má odpovědnost za správnost projektu, sestavuje plán projektu, přiděluje zdroje, podílí se také na analýze a řídí projekt. Plán a řízení projektu podle sestaveného plánu probíhá v nástroji MS Project.

Metodika LBMS popisuje několik technik, z nichž řada je podporována nástrojem CASE LBMS Systém Engineering. Pomocí tohoto nástroje jsou podporovány hlavně techniky datové, funkční a procesní.

Z datových technik je využívána nejvíce technika datové analýzy, kdy je využíváno kombinované datové modelování kombinací datového modelu a normalizovaného

relačního schématu. Z funkčních a procesních technik je využívána hodně technika tvorby hierarchické struktury dat (DFD) a technika popisu funkcí systému odvozených z DFD a seznam požadavků. Technika generování programového kódu je řešena zcela specificky a je jí věnována další kapitola.

V etapě analýzy je pomocí zmíněných technik vytvořen datový model systému, který je konzultován a musí být schválen, aby mohlo dojít k dalším etapám. V etapě analýzy dochází k následujícím krokům podle metodiky LBMS.

Provádí se Předběžné zkoumání systému, kdy je zkoumán stávající systém vytvořený již v minulosti. Je vytvořen datový model a jsou definovány jeho funkce. Zjišťují se nedostatky stávajícího systému a určují funkce, které nejsou využívány nebo jsou využívány velmi málo.

Následuje Definice požadavků na nový systém a Specifikace požadovaného systému, kdy dochází k dopřesnění požadavků na systém. Vytváří se prvotní logický datový model, definují se požadavky na utajení a přístupová práva.

Etapy logického návrhu jsou řešeny pomocí nástroje CASE LBMS System Engineering. Je vytvořen logický datový model v normalizované formě a následně dochází k převodu logického datového modelu na fyzický. Použití CASE výrazně zkvalitní proces tvorby a usnadní diskuse nad projektem. Navíc při vhodném použití generátoru sestav v různých etapách projektu lze dosáhnout vypovídací schopnosti projektu nejen pro vlastní vývojáře, ale také pro pracovníka se zaměřením na kvalitu. Tímto lze dosáhnout poměrně snadno kontrolu nad postupem prací v návaznosti na normy ISO 9000. Sladěním popisu procesní části s datovým modelem se dostávají manažerovi kvality potřebné informace, které následně využívá pro definici testů.

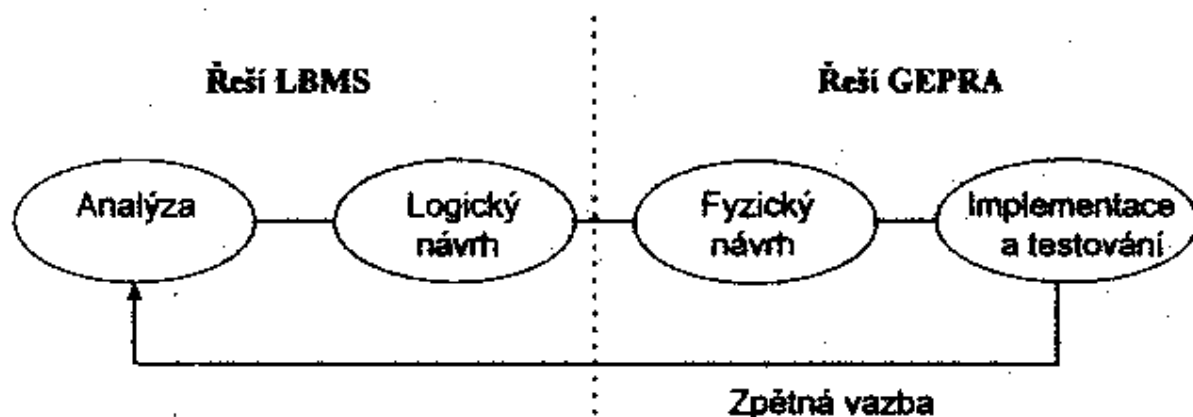
Etapa fyzického návrhu se řeší převodem logického datového modelu na fyzický a následně převodem datových struktur do systému Gepra 5.0, který je dále popsán. V souvislosti se zaváděním nových metod při tvorbě software ve firmě BM Servis bylo nutno definovat i některé vnitřní směrnice, které by spolu se standardními metodikami zajistily dobrou kvalitu výsledného produktu, umožnily jeho snadnou údržbu a to i jiným pracovníkem, než tím, který se podílel na vytvoření prvních verzí produktu. Byla vytvořena metodika definice modulů, entit a atributů. Metodika stanoví rozdělení projektu do relativně samostatných modulů. Každý modul je označen dvouznačkovou identifikací, jejíž použití je závazné ve všech dokumentech týkajících se projektu. Také byla stanovena metodika pro jména entit, kdy je důležité, aby ihned ze jména entity bylo patrné, do kterého modulu entita náleží, neboli, kde je původ vzniku entity. Pojmenování musí co nejlépe vystihovat význam entity. Dále se entitě přiřadí číslo. Taktéž je stanovena metodika pro tvorbu jmen atributů, kdy opět je důležité, aby bylo ihned ze jména atributu patrné, která entita je mateřská pro tento atribut, neboli, do které entity je atribut vstupován poprvé. Pokud je předpoklad, že se atribut použije ve více entitách, jako entita do které atribut patří se určí nejčastěji ta z entit, která je silnou entitou.

Protože CASE LBMS neumožňuje implicitní číslování relací, bylo z důvodu lepší čitelnosti diagramů stanoveno, že relace budou číslovány ve svých názvech Master-to-detail. Tato identifikace bude uvedena také u poznámky k relaci a uvedena i v diagramu datového modelu jako „volný text“. Zvolený způsob umožní lepší

orientaci v datových modelech a v datovém inventáři CASE a i při případných servisních zásazích pracovníkem služeb při implementaci konečného produktu u uživatele.

Zavedena byla také metodika identifikace programových zdrojů. Definována byla také metodika evidence množství času spotřebovaného na jednotlivé etapy tvorby projektu, aby bylo možno po ukončení projektu stanovit skutečné množství času oproti plánovanému a vyjádřit finanční hodnotu díla.

Poslední etapou je etapa implementace a testování, součástí které je také dokumentace a helpy. Pro vlastní implementaci je využíván nástroj GEPRA 5.0, který je popsán v následující kapitole. Datový model vytvořený v CASE LBMS/SE je převeden do nástroje Gepra. Nevýhodou je, že v současné době neexistuje zpětná vazba mezi LBMS/SE a Geprou, takže eventuelní změna datového modelu při implementaci musí být taktéž provedena i v LBMS/SE. Je zcela přirozené, že k nějakým drobnějším změnám dojde minimálně jednou vždy a opět musí být určena metodika, jak v takovýchto případech postupovat, aby byla zaručena kvalita výsledného produktu. Následující obrázek ukazuje jednotlivé etapy, které se v průběhu realizace projektu vyskytují.



Obr. 1

3. GEPRA 5.0 PRO WINDOWS A CLIENT/SERVER

Gepra (GEnérátor PRogramových Aplikací) pro Windows je produkt vyvinutý ve firmě BM Servis. Pracuje jako nadstavba nad databázovým prostředkem Visual FoxPro 5.0 pro Windows, ve kterém je také vyvinuta. Je používána k tvorbě programových produktů. Gepra umožňuje zvládnout programátorskému týmu práci na rozsáhlém projektu, kterého se účastní více pracovníků. Základem práce týmu v systému Gepra je rozdělení projektu na několik samostatných modulů, z nichž každý je vytvářen pouze jedním pracovníkem, čímž je stanovena odpovědnost za kvalitu té které části systému a zaručena bezpečnost ve smyslu změny kódu činnosti nepovolanou osobou. Podporuje také vytváření verzí modulů a generování koncové aplikace pro jednotlivé zákazníky podle konfigurace systému zakoupeného zákazníkem, včetně vytvoření instalačních disket. Gepra umožňuje odstranit rutinní práci programátora a eliminovat chyby, které vznikají především při programování částečně shodných částí kódu (otevírání souborů, udržování relačního prostředí, zamykání souborů při práci v síťovém prostředí, sestavování menu nebo ovládacích

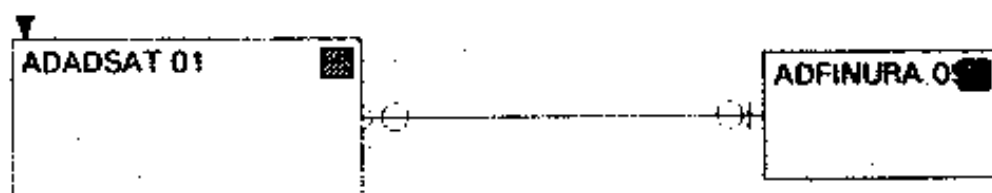
prvků jako programového kódu). Umožňuje udržet jednotný vzhled obrazovek a způsob ovládání ve všech modulech systému, i když jsou realizovány různými vývojáři. Umožňuje soustředit práci programátora na ty části programového kódu, který nelze generovat.

V prostředí Client/Server generuje klienta a zabezpečí promítnutí datového modelu včetně uložených procedur a triggerů do SQL databáze v průběhu tvorby projektu i implementace konečného produktu u uživatele. V současné době je podporována databáze Sybase SQL Anywhere 5.0.

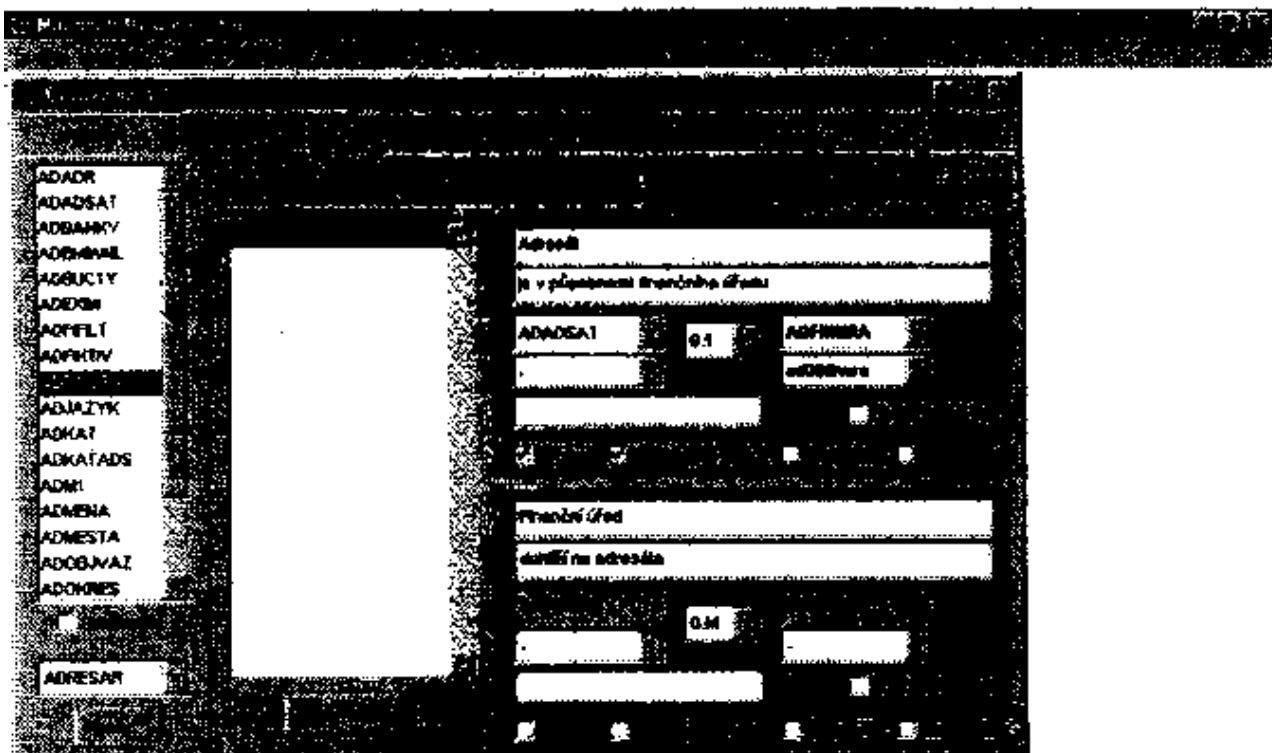
Produkt Gepra není firmou prodáván a je považován za konkurenční výhodu firmy.

3.1 Datový model

Pro každý modul se definuje jeho datový model. Jsou definovány entity, jejich atributy a relace mezi nimi. U entit se určuje zda jde o entitu běžnou (tedy běžnou databázi), pracovní entitu, která existuje pouze dočasně při běhu programu, entitu paměťových proměnných nebo entitu fiktivní, která se používá při definici činností, které nepracují s databázovými záznamy (nastavení parametrů ap.). Dále se definují parametry uplatňované při generování koncové aplikace a činností obsluhujících objekty a zda bude entita distribuována jako číselník a to buď stálý nebo modifikovatelný. U atributů se definuje popis atributu, poznámka atributu, vstupní podmínka, validační funkce, vstupní maska, inicializační kód a kód po vstupu, případně též styl zobrazení atributu. Pro speciální typy vstupu je možno uvést direktivu generování. Všechny parametry atributů je možno definovat alternativně, tak jak je to nutné pro použití v jednotlivých obrazovkách. Definuje se též příslušnost atributu do primárního či do běžných indexů. Jednotlivé indexy je možno pojmenovat pro další použití při definici třídících hledisek. U relací se uvádí master a detail entita, mocnost relace ve směru "master to detail" i ve směru "detail to master", indexový výraz, přes který je relace realizována, případně relační výraz pokud se liší od indexového. Určení mocností relací se uplatňuje při generování kódu pro definování pravidel zachování referenční integrity dat. I relace je možno pojmenovat, což zlepšuje orientaci při sestavování relačního prostředí jednotlivých činností. Následující obrázek ukazuje příklad vazby adresáta, který přísluší do působnosti některého finančního úřadu tak, jak je vytvořen v LBMS a tak, jak je použit v Gepře.



Relační vazba mezi dvěma entitami vytvořená v LBMS/SE



Interpretace této vazby v systému Gepra 5.0
Obr.2

3.2 Funkční model

Dále je definován strom činností modulu. U jednotlivých činností se definuje jejich název pro použití v menu (včetně alternativních názvů), relační datové prostředí, atributy obsažené v jednotlivých obrazovkách (pokud je činnost obsahuje), entity a atributy zahrnuté do tiskových sestav a pomocné kódy činností, jako je kód před otevřením souborů, kód po otevření souborů, kód po vstupu jedné věty, kód po vstupu poslední věty, kód před návratem a replace kód. Definují se statické a dynamické podmínky pro spuštění činnosti a její zobrazení v menu. Definují se klávesové zkratky pomocí kterých je možno činnost vyvolat. Též je možno definovat podmínky pro umožnění zařazení nového záznamu, opravy stávajících záznamů nebo zrušení záznamu. Definují se programové zdroje, které jsou v činnosti použity. Je umožněna jejich editace a kompilace. Pro editaci oken a jejich uspořádání na obrazovce je použit standardní generátor obrazovek a projektů FoxPro pro Windows. Pro tvorbu uživatelských sestav se používá generátor reportů FoxPro.

3.3. Dokumentace a helpy

Jedním z atributů činnosti funkčního modelu je také help k dané činnosti. Tento help lze průběžně aktualizovat, ale Gepra má v sobě nástroj, který umožní k vygenerovanému produktu připojit rutinu, pomocí které lze helpy poříditi při

spuštění činnosti. Současně také podstatnou část helpu dokumentátorovi předpíjí, podle dalších atributů popisované činnosti. Celkový help je potom vygenerovaný jako kontextový a slouží současně jako podklad pro dokumentaci.

3.4 Verze a spolupráce modulů

Správce Gepry definuje přístupová práva programátorů k modulům a verzím. Verze, která je vytvářena je zpravidla umístěna na lokálním disku. Spolupráce a využívání již vytvořených a odladěných programových částí se provádí pomocí verzí umístěných na síťovém disku, jejichž přesun provádí správce Gepry. To umožňuje řešit situace, kdy je nutno v určitém modulu použít části z jiných modulů, což je v programových systémech časté. Každý objekt definovaný v Gepře (entita, popis atributu, okno, činnost, programový zdroj ap.) má uvedenu identifikaci modulu, ve kterém vznikl a právo ho modifikovat má pouze ten, který má správcem určené právo k danému modulu, tedy většinou jeho autor.

4. ÚDRŽBA SYSTÉMU

Neméně důležitou součástí, jakou je tvorba nějakého systému, je také údržba tohoto systému po dobu jeho životnosti. Z tohoto důvodu je každoročně dodáván upgrade existujícího systému a je vytvořena nová verze. Každoroční tvorba nové verze je řízena jako samostatný projekt se svými etapami a kroky tak, jak bylo popsáno. Velkou důležitostí při údržbě zde má právě Gepra. V datové základně dochází ke změnám. Za tímto účelem Gepra vygeneruje jako součást systému modul, který slouží k údržbě dat. Při změnách se tento modul stává součástí dodaného systému a převádí data do nových datových struktur se zárukou správného obsahu dosud vytvořených dat.

5. ZÁVĚR

Přechod na tvorbu software podle popsaných technik je nutností. Projevilo se to ve zkvalitnění práce řešitelů, v rychlosti tvorby, v komunikaci mezi řešiteli a hlavně v komunikaci s manažerem kvality. Navíc údržba funkčního i datového modelu v systému Gepra je jednoduchou a přehlednou a umožní ve velmi krátkém čase zapracování i poměrně složitých změn. Podporuje kvalitní práci a tudíž zaručuje kvalitu výsledného produktu. Došlo k prověření tohoto způsobu tvorby software na systému, který řeší komunikační a řídicí složku v informačním systému každého podniku. Vznikl systém Harmonie, který obsahuje řádově desítky entit. Tomu úměrně odpovídá počet relačních vazeb. Tvorba takového systému trvala přibližně 4 měsíce práce jednoho programátora. V současné době je řešen ekonomický systém pro střední a větší podnik, který již obsahuje řádově stovky entit s poměrně složitými vazbami. Současně dochází k převodu systému Harmonie na platformu Client/Server.