

FORMÁLNÍ SPECIFIKACE PRO REGISTRACI VÝVOJE PODNIKOVÉHO IS

M. Mišovič

Došlo: 15. června 2006

Abstract

MIŠOVIČ, M.: *Registration of an enterprise information system development by formal specifications.* Acta univ. agric. et silvic. Mendel. Brun., 2006, LIV, No. 6, pp. 125–132

The economical view from the Enterprise process sets ERP, SCM, CRM, BI, ... to a functionality and Enterprise Information System structure by informaticians is demonstrable reality. A comprehensive Enterprise Information System software solution, that respects the mentioned economical platform by large software firms, has got required attributes of a data, process and communication integrity but there is not financially sustainable for small enterprises.

These enterprises are predominantly oriented to progressive computerization of enterprise processes and rather gradually buy application packages for individual process sets. Large and small software firms provide needed partial solutions, nevertheless small firms solutions are connected with the data, process and communication disintegration. Since the compatibility requirement is not generally accepted, finding of an EAI solution have become one of the main System Integration tasks. This article provides one specific style for a complex or partial Enterprise Information System solution. This solution is founded on formal and descriptive specifications that can sustain required data, process and communication integration among packages of applications. As a result, this style provides the new view for the effectiveness of the associated process of information modeling.

enterprise Information System, economical process sets, structural development paradigm, formal specifications, descriptive specifications, meta-base of descriptive specifications, meta-model of formal specifications, data, process and communication integrity, package of applications

Hledání koncepcí (principů) řešení EAI (Enterprise Application Integration) pro IS tvořené balíky aplikací je považováno za jeden z hlavních soudobých úkolů systémové integrace. Problematika EAI pro PIS velkých podniků je diskutována ve studii (Sodomka, Habáň; 2005) a v publikaci (Gála, Pour, Toman; 2006), kde jsou uváděny konkrétní koncepce (principy), které umožní narušení integrity odstranit. Jsou však možné i jiné přístupy, které mění styl řešení PIS od samého začátku.

Proto je hlavním cílem příspěvku zavedení specifického komplexního řešení vývoje podnikového IS, s možností využití jeho parciálních částí, na základě použití **formálních specifikací**. Použitím formálních specifikací k popisu výsledků vývoje PIS vznikají jejich instance, tj. **deskriptivní specifikace**, které je

možno uložit do relační databáze. Příspěvek nabízí hypotézu tvořenou následujícími myšlenkami:

- Databáze deskriptivních specifikací, přístupná všem vývojářům, se stane vědomostním jádrem komplexního (parciálního) řešení vývoje PIS typového podniku jisté oborové orientace.
- Každý z vývojářů může z databáze čerpat a rovněž do ní vkládat popis výsledků analýzy dalších procesních setů, které již mohou využívat dříve zavedených entit a procesů.
- Databáze zaručí zachování datové, procesní a komunikační integrity v rámci všech výsledků vývoje PIS.

Tuto hypotézu můžeme považovat za specifický styl postupu ve vývoji PIS, který umožní zachovat

nejen integritu, ale umožnit rovněž kolektivní vývoj PIS. Hlavním požadavkem naznačeného specifického stylu je povinnost nejen používat navržené formální specifikace a jejich instance – deskriptivní specifikace každým vývojářem, ale rovněž navázat na již uložené výsledky modelování předchozích procesních setů. Metabáze bude mít zabudovanou autoverifikaci, která přiměje vývojáře neporušovat datovou, procesní a komunikační integritu mezi již modelovanými a právě modelovaným procesním setem. Ačkoliv se mohou uvedené myšlenky považovat za utopii, přesto je vhodné se jimi na akademické půdě zabývat.

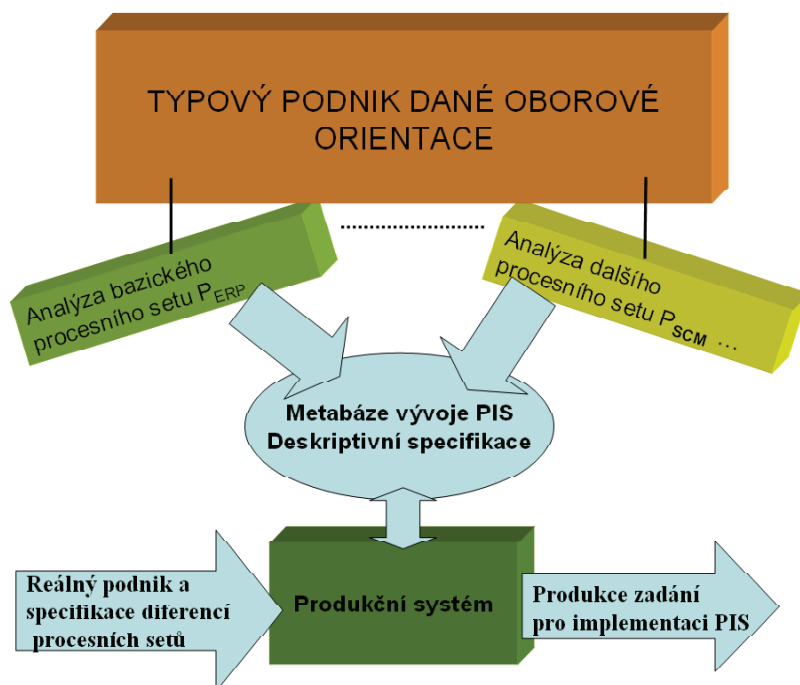
Jestliže se orientujeme na strukturované paradigma vývoje PIS, potom deskriptivní specifikace zaznamenávají realitu výsledků dílčích analýz (datové, procesní, transakční a interakční) strukturované metodiky. Deskriptivní specifikace musí poskytovat všechny informace pro typické operace, pomocí kterých se snadno získají a vizualizují finální diagramy (ERD, DFD, diagramy transakcí a ELH) dílčích analýz.

A právě metabáze musí být vědomostní bázi tohoto stylu s interpretací do strukturovaného paradigmatu, protože bude obsahovat výsledky modelování, které se dají měnit a doplňovat. Metabáze by měla být vytvořena nejen pro evoluční vývoj IS podniku, ale rovněž pro typový podnik jistého oborového zamě-

ření s možností její modifikace na „tělo“ reálného podniku. Postup dílčích analýz bude prováděn počínaje procesním setem ERP (jádro každého PIS s procesní architekturou odrážející vztahy procesních setů ERP, SCM, CRM, ...) a jeho subsety (řízení lidských zdrojů, financí, materiálu, prodeje a výroby). Později by byly dílčí analýzy uplatněny na další procesní sety a jejich subsety.

Deskriptivní specifikace, uložené v metabázi, obsahují stavební materiál, na jehož základě můžeme vystavět relevantní diagramy (ERD, DFD, diagramy transakcí a ELH). Deskriptivní specifikace transakcí obsahují schéma kódu procedur a na základě deskriptivních specifikací transakcí, procesů a procesních řetězců v daném procesním subsetu, můžeme vybudovat zadání pro implementaci jedné a dalších aplikací. Tímto postupem pro další subsety téhož setu můžeme vybudovat jeho celý balík aplikací.

Najdeme-li možnost formulace specifikací pro PIS reálného podniku, náležícího do stejného oboru jako typový, můžeme potom konstruovat produkční systém, který na základě obsahu metabáze a vstupních specifikací reálného podniku, dokáže produkovat velmi užitečné **zadání pro programátory** vedoucí k implementaci na balík nebo dokonce postupně na balíky aplikací, viz Obrázek 1.



1: Možnost produkce zadání pro implementaci integrovaných balíků aplikací

MATERIÁL A METODIKA

Základním zpracovávaným materiálem jsou formální specifikace a jejich definice, které jsou v souladu s pojmovým rámcem strukturovaného paradigmatu vývoje PIS, který formalizují. Od formálních specifikací pokročíme k jejich praktickým výskytům – deskriptivním specifikacím, které již zapisují reálné výsledky vývoje PIS. Ukážeme způsob uložení deskriptivních specifikací do metabáze a některé zajímavé operace nad metabázemi. Článek se již nezabývá detaily algebry operací nad metabázemi a problematikou produkce balíčků nebo balíků aplikací PIS. Tato problematika je v následných článcích autora.

1. Formální specifikace vývoje podnikového IS

Formální specifikace jsou obecné notace (zápisy), které používají formální prvky k zobecnění pohledu na reálné poznatky, vlastnosti produktů strukturované analýzy.

Formální specifikace mohou být sourodé a vycházet z jistého formálního jazyka. Formální specifikace se potom vytvářejí z výrazů a příkazů daného formálního jazyka. Jsou ovšem četné případy, kdy formální specifikace nemusí mít v pozadí jistý formální jazyk. Častým případem je interpretace formálních specifikací v jisté bazické množině nebo logice. V informatice se formální specifikace používají velmi často, protože dokážou charakterizovat obecné vlastnosti programů, systémů, ... Takovými jsou i formální specifikace funkčních modelů (např. sémantiky imperativních programů) vycházející z jazyků OBJ3 a LEGO, vytvářející uspořádané množiny logických rovnic jako jistých pravidel pro aktivity v programu.

Za případ sourodých formálních specifikací považujeme rovněž grafické specifikace (graphical notations) výsledků vývoje PIS v objektovém paradigmatu, které vznikají na základě jazyka UML (Unified Modeling Language). Strukturované paradigma podobný formální jazyk nemá.

Na druhé straně, pro strukturované paradigma, které jsme se rozhodli zde používat, navrhne jednoduché notace formálních specifikací – formálních předpisů, které budeme interpretovat v množinách datových entit, informačních asociací, procesů, transakcí a procesních řetězců. Tato interpretace převede formální specifikace do různých množin tzv. deskriptivních specifikací. Deskriptivní specifikace již nejsou formálními předpisy, ale zaznamenávají výsledky vývojové činnosti. Narozdíl od formálních specifikací, deskriptivní specifikace již uložíme do relační báze dat a navrhne mezi nimi existující asociace, tj. binární relace na základě jistého předpisu.

Pro strukturované paradigma potřebujeme navrhnout formální specifikace zaměřené na *entity*, *asoci-*

ace mezi entitami, *procesy*, *rozklady procesů na transakce*, *transakce*, *spojení procesů do řetězců procesů*, *asociace dané transakce a entit*. Těmito specifikacemi bude formalizována celá datová a procesní podstata vývoje PIS.

2. Formální specifikace entit a jejich informačních asociací

Entity jsou datové struktury vycházející z datové analýzy a víme, že jejich informační asociace jsou základem báze dat systému. Formalizace celého teoretického systému entit vyžaduje nejen formalizaci samotných entit, ale i jejich asociací. Tomu úspěšně slouží modernizovaná Coddova algebra a sémantická algebra závislostí atributů entity. Pro účely vývoje PIS musíme entity a jejich informační asociace popsat takovými formálními specifikacemi, které se jednoduše interpretují a tyto interpretace snadno ukládají do speciální báze dat. Narozdíl od zmíněných algeber mají tyto interpretace již metacharakter. Definice 1 a 2 zavádí potřebné pojetí formální specifikace entity a její interpretaci – deskriptivní specifikaci.

Definice 1: Formální specifikace entity

Entita e s atributy (a_1, a_2, \dots, a_n) , kde $\forall i (a_i \in A, a_i: t_i)$, je specifikována formální notací $e = [A, b, T]$, kde $A = \{a_1, a_2, \dots, a_n\}$ je uspořádaná množina atributů, $b \in A$ je definiční atribut a $T = \{t_1, t_2, \dots, t_n\}$ je uspořádaná množina datových typů atributů.

Tato formální specifikace popisuje strukturu entity z atributů a vlastností samotných atributů, což je pro informační modelování zcela dostačující. Velmi důležité je ovšem to, jak tento formální předpis interpretovat, protože reálně existujících entit je v podniku kolem několika stovek. Je pochopitelné, že teď musíme definovat pojetí interpretace formální specifikace entity, které musí vést již na deskripce reálně existujících instancí. Tyto deskripce jsme nazvali *deskriptivními specifikacemi*.

Definice 2: Deskriptivní specifikace entity

Jestliže A' je uspořádaná množina skutečných jmen atributů a'_1, a'_2, \dots, a'_n , $b' \in A'$ a T' je uspořádaná množina datových typů atributů množiny A' , potom $e' = [A', b', T']$ je *deskriptivní specifikací*, tj. interpretací formální specifikace $e = [A, b, T, D]$ do množin A' , T' a atributu b' .

Deskriptivní specifikace entity podniku je již materiálem, který je možno zpracovávat. Entita e' má již ve své deskriptivní specifikaci reálný popis atributů (definiční atribut, datové typy atributů) a je tedy možné sestavit v relační bázi dat její prázdnou populační tabulku. Je ovšem patrné, že se nevyžaduje zadání

populace instancí entity e' . K takovému požadavku dojde až v době konstrukce reálné báze dat (operace naplnění populací, realizace informačních asociací, nastavení jejich kardinalit a referenčních integrit).

Na rozdíl od formální specifikace entity bude specifikace informační asociace dvou entit složitější. Nechť je E množinou všech entit podniku. Označme informační asociaci dvou entit zápisem (e_1, e_2) . Pochopitelně, $E^2 = E \times E$ je množinou všech možných asociací mezi entitami. Typy 1:1, 1:N a M:N informačních asociací rozkládají množinu E^2 na dvě disjunktní podmnožiny F_1 a F_2 asociací, pro které platí:

$$E^2 = F_1 \cup F_2$$

$$F_1 = \{(e_1, e_2) \mid e_1, e_2 \in E \wedge (e_1, e_2) = (e_2, e_1)\} \dots\dots\dots$$

pro typ 1:1

$$F_2 = \{(e_1, e_2) \mid e_1, e_2 \in E \wedge (e_1, e_2) \neq (e_2, e_1)\} \dots\dots\dots$$

1:N, M:N

Jelikož formální specifikace musí popisovat asociaci obecně, tj. ukázat, co má obsahovat a jaké vlastnosti o ní se musí znát, potom přichází v úvahu následující veličiny:

- které entity a v jakém pořadí jsou asociovány,
- společný asociační atribut b ,
- typ t asociace,
- jméno j asociace,
- min. kardinalitu k_1 a max. kardinalitu k_2 asociace a
- referenční integrita i pro danou asociaci.

Můžeme tedy vyslovit následující definici.

Definice 3: Formální a deskriptivní specifikace informační asociace (e_1, e_2)

Buďte e_1, e_2 dvě formální entity. Formální specifikace asociace (e_1, e_2) je dána zápisem $(e_1, e_2) = [j, e_1, e_2, b, t, k_1, k_2, i]$, kde j je jméno asociace, e_1, e_2 jsou odkazová jména entit, b je společný asociační atribut, t je typ asociace, k_1, k_2 je minimální a maximální kardinalita asociace a i je typ referenční integrity asociace.

Jestliže interpretujeme všechny komponenty formální specifikace asociace, potom $(e'_1, e'_2) = [j', e'_1, e'_2, b', t', k'_1, k'_2, i']$ je deskriptivní specifikací asociace (e'_1, e'_2) reálně existujících entit e'_1, e'_2 .

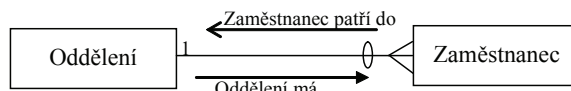
Je zřejmé, že jména entit podniku jsou výlučná, což plyne rovněž z definice množiny E . Deskriptivní specifikace asociace dvou reálných entit obsahuje vše pro vizualizaci této asociace. K realizaci asociací entit však dojde až v době konstrukce reálné relační báze dat, přesněji až poté, co byly zřízeny tabulky a naplněny populacemi ke všem entitám množiny E .

Příklad

Nechť jsou dány dvě entity:

Oddělení s atributy {jm-oddělení, č-budovy, č-poschodí} a Zaměstnanec s atributy {č-zaměstnanec, příjmení, křestní, dat-nar}.

Nechť jsou entity spojeny následující informační vazbou



Pro obě entity můžeme napsat následující deskriptivní specifikace entit a jejich asociace:

popisy entit

Oddělení = [{jm-oddělení, č-budovy, č-poschodí }, **jm-oddělení**, {text, celé číslo, datum}]

Zaměstnanec = [{č-zaměstnanec, příjmení, křestní, dat-nar}, **č-zaměstnanec**, {text, text, text, datum}]

Asociace entit

(Oddělení, Zaměstnanec) = („Oddělení má“, Oddělení, Zaměstnanec, jm-oddělení, 2, 1, n, 1)

(Zaměstnanec, Oddělení) = („Zaměstnanec patří do“, Zaměstnanec, Oddělení, jm-oddělení, 1, 1, 1, 1), kde t , typ asociace, je kódován jako 1... pro typ 1:1, 2... pro typ 1:N a 3... pro typ M:N.

Typ referenční integrity i asociace je kódován čísly 1, 2, 3 a 4.

Poznámka 1:

Z definic 1, 2 a 3 je zřejmé, že deskriptivní specifikace k dané formální specifikaci vznikají interpretací komponent formální specifikace skutečnými hodnotami, množinami, ... Na základě tohoto poznání se dále orientujeme jen na samotné formální specifikace a vznik deskriptivních specifikací považujeme za triviální.

3. Formální specifikace procesů, procesních řetězců a transakcí

Podnikové procesy jsou podle (Řepa, 2006) fenoménem aktivit každého podniku, což se také odráží v implementaci **procesně orientované architektury** založené na význačných procesních sítích ERP, SCM, CRM, BI, ... do podnikových IS.

Chceme-li popsat obecný proces p , potřebujeme znát jeho jméno, přináležení procesu k jistému procesnímu subsetu, jméno spouštěcí události, množinu rozkladových transakcí a kód zápisu tzv. procesního programu (vyvolání transakcí a pravidla potenciálních přechodů mezi transakcemi). Detailní informace o zpracovávaných datových entitách daného procesu je uvedena v deskriptivních specifikacích jeho transakcí. „Rozumné“ aktivity podniku se realizují jako posloupnost procesů, tedy přesněji jako řetězec procesů $p_1 - p_2 - p_3 - \dots - p_n$. Procesy a procesní řetězce jsou základním vědomostním vybavením managementu podniku. Start procesního řetězce je evokován

událostí (nejčastěji externí), např. příchodem nových instancí dokumentů, kterou je spuštěn čelní proces p_1 .

Management podniku a informatici sice považují procesy za stavební kameny pro Data Flow Diagrams (diagramy prezentující znalost toků dat a jejich zpracování v podnikových procesech), ale za dále nedělitelné a logicky ucelené jednotky aktivity podniku považují transakce. Procesy se proto specifikují na základě jejich rozkladů do dílčích transakcí. Transakce jsou aktivity, které se musí realizovat počítačem, čímž se vlastně vytváří aplikační software IS podniku.

Transakce patří do jedné z následujících skupin:

- 1) Dotazové transakce.
- 2) Aplikační transakce (vzniklé rozkladem procesů podniku).
- 3) Ostatní transakce (rozklady dodatečně přidaných procesů, které nejsou podnikové).

Ačkoliv je popis transakce nutno provést velmi podrobně (jméno transakce, název spouštěcí události, obsah – algoritmus – kód, podmínky měnící efekt, nižší vstupní/výstupní datové struktury, zpracovávané vstupní/výstupní entity, vizualizační okna pro zpracování entit a nižších datových struktur, pravidla podniku), uvedeme ve formální specifikaci jen jeho základní parametry.

Definice 4: Formální specifikace procesu, řetězce procesů a transakce

Bud' $Tr = \{tr_1, tr_2, \dots, tr_n\}$ množina transakcí procesu p . Formální specifikací procesu p nazýváme uspořádanou pěticí $p = (j, u, Tr, s, p_{kód})$, kde

- Tr je konečná množina transakcí, na které se proces rozkládá,
- s jednoznačné jméno procesního subsetu k němuž proces náleží,
- $p_{kód}$ kódový zápis procesního programu.

Uspořádanou posloupnost $Pr = (p_1, p_2, \dots, p_n)$ procesů p_1, p_2, \dots, p_n nazveme formální specifikací procesního řetězce Pr .

Formální specifikací transakce t nazveme uspořádanou šestici $t = (j, p, u, E_1, E_2, t_{kód})$, kde

- E_1, E_2 jsou množiny zpracovávaných vstupních a výstupních entit,
- j, p je jednoznačné jméno transakce a jméno procesu jemuž náleží,
- u jednoznačné jméno spouštěcí události,
- $t_{kód}$ kódový zápis transakce – transakční procedura.

Asociace entit a transakcí je obecně velmi pestrá, protože jedna transakce zpracovává více entit a naopak tatáž entita může být zpracovávána ve více transakcích (asociace typu M:N). Pro rozklad této asociace je možno zavést novou formální specifikaci.

Poznámka 2:

Mezi formální specifikace budeme počítat i tzv. *globální struktury PIS* $= (PS_p, \dots, PS_r)$ a $PS_i = (PSU_p^i, \dots, PSU_r^i)$, které stanovují složení podnikového IS z jednotlivých procesních setů a vztah procesních subsetů PSU k procesním setům PSU. Ve formální specifikaci procesu se potom nastavuje jeho vztah k procesnímu subsetu. Ostatní formální specifikace jsou na procesní subsety navázány přes formální specifikaci procesu. Vedle toho zřídíme formální specifikace pro zdroje dat, příjemce dat, datové toky a úložiště dat.

4. Relační metabáze deskriptivních specifikací a operace nad metabází

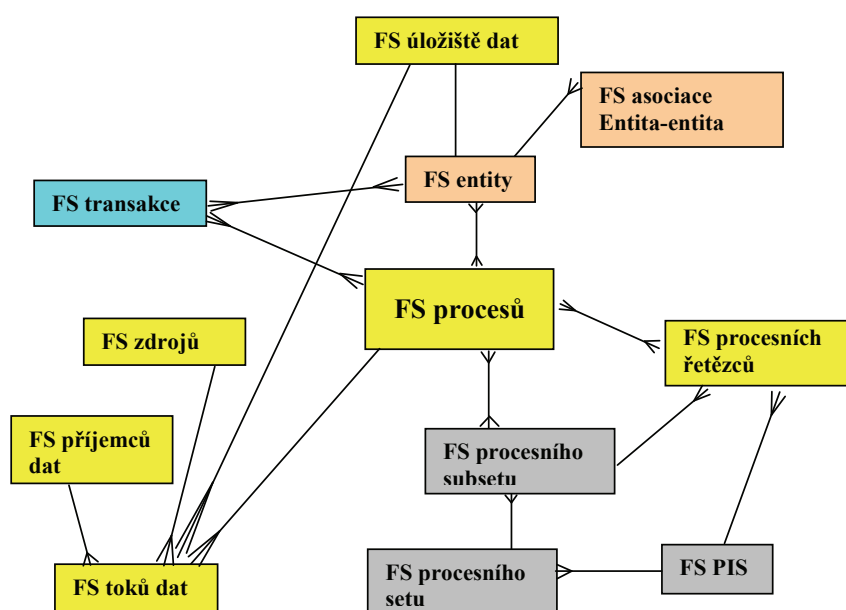
Mezi formálními specifikacemi jistě existují velmi užitečné souvislosti – vztahy, které se na fyzické úrovni přenášejí rovněž na deskriptivní specifikace. Existuje tedy metamodel, který tyto vzájemné vztahy ilustruje, viz Obrázek 2.

Tak jak již bylo naznačeno, všechny instance formálních specifikací entit, asociací mezi entitami, procesů, rozkladů procesů na transakce, transakcí, spojení procesů do řetězců procesů a samotných transakcí uložíme do metaskladu, který převedeme v souladu s metamodellem do relační metabáze.

Na základě speciálních operací, které řeší rovněž úlohu vizualizace, získáme z metabáze základní diagramy strukturovaného paradigmatu – ERD a DFD. Pochopitelně, nad relační metabází můžeme provádět řadu velmi užitečných operací. Protože výsledky vývoje PIS (např. diagramy ERD, DFD, ...) máme v metaformě, můžeme žádat jejich přirozenou podobu a dokonce ji vizualizovat. Doplnovat, modifikovat a vyměňovat obsah metabáze je zcela přirozeně základním požadavkem specifického vývoje PIS.

Jestliže si podrobně všimneme materiálu, který je uložen v metabázi, zjistíme její detailní vlastnosti a široké možnosti využití. Uvedme některá z nich:

1. Metabáze může být orientována na typový podnik jistého oborového zaměření.
2. Formální specifikace a jejich instance mohou být považovány za platformu nástroje CASE (Computer Aided Software Engineering) pro komputelizaci jednotlivých fází strukturované metodiky.
3. Metabáze je platformou pro strukturovaný vývoj



2: Metamodel vzájemných vztahů formálních specifikací

podnikového IS stylem uplatnění architektury vycházející z procesních setů (ERP, SCM, CRM, BI, ...) a jejich vazeb. Metabáze umožní kolektivní vývoj IS mnoha malými softwarovými firmami.

4. Metabáze zachovává běžné pojetí procesní, datové a komunikační integrity výsledků vývoje. Umožní přenést tuto integritu na balíky aplikací IS (každý balík aplikací IS je orientován na jeden procesní set).
5. Metabáze je zdrojem pro produkci jak komplexního řešení IS podniku, tak i jeho parciálních částí k jednotlivým procesním setům, se zachováním integrity balíků aplikací. Pro produkci jedné aplikace je ovšem potřebné vyvinout specifický produkční systém.
6. Pomocí množiny speciálních operací pro metabázi je možné vyšetřovat případy narušení konzistence výsledků jednotlivých analýz strukturovaného paradigmatu, tj. datové, procesní a transakční analýzy.

Operace nad metabází mohou mít velmi zajímavou funkcionalitu. Za velmi důležité považujeme např. následující skupiny operací: **vizualizace**, **verifikace** a **editace** výsledků vývoje IS.

Editace výsledků vývoje umožní např. nejen reengineering všech procesů podniku, ale i jejich „založení“ pro reálně existující podnik.

VÝSLEDKY

Hlavními výsledky příspěvku jsou návrhy formálních specifikací pro zápisy výsledků vývoje PIS ve strukturovaném paradigmatu a jejich instancí – deskriptivních specifikací. Vedle toho je pro další použití relevantní rovněž způsob uložení deskriptivních specifikací tak, aby se vytvořila dále použitelná relační metabáze. Relační metabáze umožní pomocí specifických operací vizualizovat výsledky vývoje PIS a produkovat podle potřeby nejen známé strukturní diagramy jako jsou ERD a DFD, ale i procesní a transakční materiál. Na druhé straně je možno pomocí jiných operací provádět verifikaci konzistence datové, procesní a transakční analýzy.

Jestliže do metabáze uloží výsledky analýzy vybraného procesního setu (po jeho subsetech) jedna malá softwarová firma, může jiná analyzovat další procesní set, přičemž má možnost navázat na předešlé výsledky a dodržet definitorické požadavky datové, procesní a komunikační integrity mezi dvěma balíky aplikací.

Je zřejmé, že postup zavedení formálních a deskriptivních specifikací a konstrukce metabáze by byl úspěšně využitelný konstruktéry prostředků typu CASE. Na základě poznatkové metabáze by se dala organizovat i produkce kódu procesních řetězců – samostatných aplikací náležících do balíčku aplikací jistého procesního setu.

SOUHRN

V příspěvku je uvedeno původní pojetí formálních specifikací a jejich instancí – deskriptivních specifikací k registraci a možné modifikaci a využití výsledků modelování PIS ve strukturovaném paradigmatu. Zavedení formálních specifikací a jejich instancí umožnilo sestavit nad nimi relační metabázi dat a ukázat možnosti jejího využití. Má-li být metabáze základem vývoje PIS, musí pomocí deskriptivních specifikací absorbovat výsledky všech fází vybrané metodiky strukturovaného paradigmatu, tedy Úvodní studie, Analýzy (datové, procesní, transakční a interakční) a Hrubého a detailního návrhu.

PIS – podnikový informační systém, ekonomické procesní sety, paradigma strukturovaného vývoje PIS, formální specifikace, deskriptivní specifikace, metabáze deskriptivních specifikací, metamodel formálních specifikací a jejich vztahů, datová, procesní a komunikační integrita, balík aplikací

LITERATURA

- BASL, J.: *Podnikové informační systémy. Podnik v informační společnosti*. Grada Publishing, Praha, 2002.
- FIALA, F., MINISTR, J.: *Průvodce analýzou a modelováním procesů*. Ostrava: VŠB-TU, 2003. 110 s. ISBN 20-248-0500-6.
- SODOMKA, P., HABÁŇ, J.: *Integrace podnikových aplikací. Analytická studie CVIS*, 2005. Uvedeno na adrese http://www.cvis.cz/index_cz.htm
- MIŠOVIČ, M., TRENC, O.: *Malé SW firmy a komplexní řešení podnikového IS*. Konference Informatika XVII, Karlov pod Pradědem, 2005. Sborník s. 126-133. ISBN 80-7302-110-2.
- VOKŮRKOVÁ, L.: *Co trápí firmy v oblasti IT?* COMPUTERWORLD č. 5, roč. XVII, IDG Czech, Praha, 2006.
- GÁLA, L., POUR, J., TOMAN, P.: *Podniková informatika*. Grada Publishing, Praha, 2006.
- ŘEPA, V.: *Podnikové procesy. Procesní řízení a modelování*. Grada Publishing, Praha, 2006.

Adresa

Prof. RNDr. Milan Mišovič, CSc., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: misovic@pef.mendelu.cz

