MULTIDIMENZIONÁLNY VÝVOJ INFORMAČNÝCH SYSTÉMOV POMOCOU CASE SYSTÉMOV

Iveta Kremeňová[a]
Milan Petrík[b]

a) Žilinská univerzita, Fakulta prevádzky a ekonomiky dopravy a spojov, Katedra spojov, Moyzesova 20, 010 26 Žilina, SR
b) Žilinská univerzita, Fakulta strojnícka, Katedra aplikovanej matematiky, Moyzesova 20, 010 26 Žilina, SR

Abstrakt

Príspevok približuje možnosti uplatnenia reengineeringových postupov zmien organizácie využitím modelovania pri návrhu novej formy organizácie ako i podporou zo strany IT. Riešenie problematík, ktoré súvisia s tak **softwarovými ako aj nesoftwarovými systémami,** ovplyvňujú najnovšie metodiky a metódy objektovej analýzy a návrhu systému. Dobrý CASE nástroj musí byť schopný vyhovieť všetkým požiadavkám užívateľa bez ohľadu na to, ako sa tieto v čase menia. Produkty metaCASE systémov sú zaujímavé práve svojou flexibilitou a to nielen z časového hľadiska (vo vzťahu k zmenám vyplývajúcich z neustáleho vývoja systému), ale aj z hľadiska odlišnosti požiadaviek rôznych užívateľov – schopnosť vyhovieť širokému spektru užívateľov je daná práve množstvom metód, možnosťou upravovať ich, ako aj vytvárať vlastné, možnosťami a vlastnosťami jednotlivých nástrojov.

1. Úvod

V oblasti systémového vývoja spoločnosti čelia užívatelia SW nepretržitej potrebe zlepšovať vývojové postupy tvorby softwarových i nesoftwarových systémov. Často sa stretávajú s dvoma výzvami: **podpora ekonomických procesov spoločnosti a schopnosť rýchlejšie budovať SW vyššej kvality**. Je jasné, že informačné systémy a obchodné (ekonomické) procesy sú veľmi úzko previazané. Čím viac SW podporuje obchodné procesy, tým je cennejší. Zmeny v ekonomickej činnosti a stratégia podniku musia byť adekvátne odzrkadlené v podpornom SW a v celom systéme. Toto sa stáva jedným z kľúčových problémov v integrovaní rozvoja infraštruktúry IT ako aj v rozvoji obchodnej činnosti. Na katedre sa v súčasnosti zaoberáme problematikou aplikovania nástrojov metaCASE systému, (konkrétne metaEDIT+) na podmienky transformujúcej sa Slovenskej pošty, š.p. pre účely analýzy a návrhov nových systémov. Nesoftwarových i softwarových, spojených napr. s novou organizačnou štruktúrou a budovaním informačných systémov.

2. Reengineeringové ostupy zmien organizácie

K novým smerom manažerskej práce v 90. rokoch patrí ***reengineering***, pod ktorým obvykle chápeme zásadné prehodnotenie a radikálna zmena podnikateľských procesov s cieľom dosiahnuť dramatické zlepšenie doterajších parametrov hospodárenia. V procese reengineeringu kľúčovú úlohu hrá schopnosť tvorivým spôsobom nachádzať a aplikovať nové možnosti, ktoré poskytujú informačné technológie. *Reengineering* môžeme chápať aj všeobecnejšie ako schopnosť začať s čistým listom papiera kvalitatívnu inováciu procesov

v súlade s potrebami a možnosťami informačnej spoločnosti. Nepreberať, ale kriticky spochybňovať konvenčné múdrosti a zásady.

Reengineering zasahuje nielen do oblasti vlastnej ***podnikateľskej činnosti***, ale i do ***procesu vlastného zavádzania IT***.

3. Druhy modelovania pri návrhu systemov

Pri návrhu systémov v rámci softwarového inžinierstva sa v priebehu vývoja metodík a metód podporovaných CASE systémami môžeme stretnúť s nasledovnými druhmi modelovania:

- **Dátové modelovanie** – systém je zložený z veľkého množstva spracovávaných údajov. Podstata je vytipovanie nosných objektov modelovaného systému (entít). Každá entita je opísaná svojimi vlastnosťami (atribútami) a vzťahmi (reláciami) k ostatným entitám dátového modelu. Najznámejšou technikou sú ERD diagramy, používajú sa pre stabilné systémy, relatívne málo sa meniace.
- **Procesné modelovanie –** dôraz sa kladie na procesy (procedúry), ktoré v modelovanom systéme prebiehajú. Vytipujú sa procesy prebiehajúce v systéme a I/O informačné toky, informácie, ktoré si tieto procesy vymieňajú. Okrem procesov a tokov sú navrhnuté rôzne zásobníky údajov (súbory, DB servery) a externé objekty. Používajú sa pre systémy, ktoré sa neustále menia.
- **Objektovo-orientované modelovanie –** základom je návrh tzv. objektov, pomocou ktorých je možné vyjadriť všetky predchádzajúce pohľady na daný systém (dátové, procesné, stavové). Objekt je daný svojimi vlastnosťami (atribútmi), a metódami (procesmi), ktoré je možné nad týmto objektom vykonať. Ďalej je definovaná skupina udalostí, na ktorú daný objekt reaguje, jeho odoziev na tieto udalosti, ako aj vzťahy medzi jednotlivými objektmi. Pomocou takto navrhnutých objektov je možné vytvárať rôzne typy diagramov opisujúcich modelovaný systém (dátové, procesné, stavové diagramy).
- **Unifikácia v modelovaní -** zjednotenie existujúcich modelovacích nástrojov. **UML – Unified Modeling Language** zavádza určitý stupeň štandardizácie v oblasti modelovania. Bol vyvinutý autormi už existujúcich modelovacích nástrojov, ktorý v sebe zlučuje výhody týchto nástrojov. UML – je modelovací jazyk slúžiaci na špecifikáciu, vizualizáciu, konštruovanie a dokumentáciu pri tvorbe tak **SW ako aj nesoftwarových systémov.**

Modelovanicia technika – je najlepší spôsob zrozumiteľného opisu vyvíjaného systému v súčasnom i budúcom stave. Nástroje, ktoré vytvárajú rôzne typy modelov sú spravidla súčasťou CASE systémov.

4. Charakteristika CASE a METACASE nástrojov

Pojem **CASE** (Computer Aided Software/Systems Engineering) znamená počítačom podporovaný vývoj IS, resp. počítačom podporované softwarové inžinierstvo. Nasadenie CASE vyžaduje reorganizáciu práce so zameraním na disciplínu, systematičnosť a formalizáciu celého procesu, čo je mnohokrát obtiažnejšie než samotné zvládnutie CASE produktu.

Tradičné CASE nástroje sú založená na **dvoj úrovňovej architektúre**: systémové návrhy sú ukladané a uchovávané v schránke (s*ystémová encyklopédia* -Repositary), v ktorej sa udržiavajú všetky informácie o projektovanom systéme, výsledky všetkých projekčných krokov. V princípe ide o databázu, ktorá sa automaticky udržiava v konzistentnom stave. Všetky pravidlá, ktoré sa dajú v rámci použitej metodiky algoritmizovať, CASE neustále aplikuje na uložené výsledky. Zároveň bráni činnostiam, ktoré by viedli k logickým alebo formálnym chybám. Od CASE systémov sa v najbližšej dobe očakáva:

- zavedenie jednotného štandardu (normalizácia) – UML modelovanie,
- väčšia integrácia nástrojov, posun k objektovej integrácii.

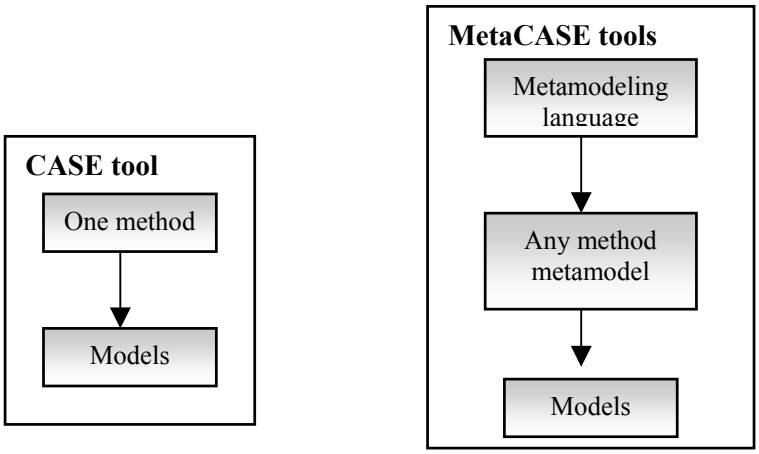
4.1 MetaCASE nástroje

MetaCASE nástroje sú založené na **trojstupňovej** architektúre. *Najnižšia* úroveň modelu zahŕňa systémové návrhy ako modely. *Prostredná* úroveň obsahuje určitý metódový model – metamodel. Metamodel zahŕňa koncepty, pravidlá a diagramové (grafické) zápisy danej metódy. Napr. metamodel metódy UML špecifikuje koncepty ako „trieda“ a „dedičstvo“, ich vzťahy, súvislosti, prepojenie a reprezentáciu.

MetaCASE nástroje umožňujú modifikovať metamodel. Teda metaCASE je založený na flexibilite metódových špecifikácií. Toto je dosiahnuté *treťou*, najvyššou úrovňou, ktorá zahŕňa metamodelovací jazyk pre špecifikáciu metód.

Všetky tri úrovne sú tesne prepojené: model ja založený na určitom metamodeli, ktorý na druhej strane je založený na metamodelovacom jazyku. Táto štruktúra závislosti je podobná tej medzi objektami, triedami metatriedami v niektorých objektovo-orientovaných jazykoch.

Obr.1 zachytáva rozdiel medzi CASE a metaCASE nástrojmi.



Obr. 1. CASE nástroj a metaCASE nástroj

4.2 Funkčnosť metaCASE nástrojov

Okrem trojstupňovej architektúry, metaCASE nástroj musí poskytovať funkcie, ktoré poskytujú súčasný vývoj metód a implementáciu CASE funkčnosti. Tieto funkcie predstavujú:

- **Definícia metamodelov**. Prvá obvyklá požiadavka je tá, či nejaký metaCASE nástroj dokáže špeicifikovať koncepty, pravidlá a symboly individuálnych modelingových techník, rovnako ako aj pravidlá ich vzájomného prepájania. Okrem toho definície metód majú byť také kompletné, ako je to len možné, a v porovnaní s programovaním nejakého CASE nástroja z náčrtu musí byť možné urobiť ich relatívne ľahko a rýchlo.
- **Vytvorenie nástrojov modelovania.** Založený na definíciách metód, metaCASE nástroj musí poskytovať všetky potrebné modelovacie nástroje na podporu projektovania, modelovania s danou metódou. Tieto zahŕňajú rozličné druhy editorov, nástrojových líšt, dialógov, online help atď. Vytvorenie týchto nástrojov má byť automatické, založené na metamodeloch.
- **Vytvorenie schránky.** Spoločne modely a metamodely musia byť uložené a uchovávané v multi – užívateľskej schránke a musia byť dostupné pre vývojových pracovníkov. Rovnako musia byť dostupné aj ostatné nástroje pre administráciu v schránke.
- **Definícia pre reportovanie modelu.** Okrem modelového editovania a uchovávania, by metaCASE nástroj mal umožňovať definovanie rôznych modelových analýz, kontrol, dokumentačných reportov modelu a kódovo generovaných reportov. Stojí za povšimnutie, že hoci dobrý metamodel zahŕňa všetky pravidlá danej metódy, ich kontrola nemôže byť pne automatická, niektoré pravidlá možno kontrolovať iba keď sú modely považované za kompletné.
- **Riadenie metamodelu.** MetaCASE nástroj má poskytovať funkčnosť pre riadenie metamodelu podobnú, aká je u kunkcií CASE nástrojov pre riadenie modelu. Toto zahŕňa prehliadač, nástroje dokumentácie, knižnice pre metamodely a nastavenie prístupových práv pre metódové špecifikácie alebi pre ich časti.
- **Riadenie aktualizácií metódy**. MetaCASE nástroj má poskytovať (zabezpečovať) transformačné pravidlá medzi verziami danej metódy (napr. UML 1.2 do UML 1.3 elebo do vašej rozšírenej UML) a tiež aktualizované návrhy, projekty (polo-) automaticky korešpondovať do novej verzie metódy. Táto funkcia obyčajne vyžaduje entegrovaný metaCASE nástroj a CASE nástroj.
- **Výmenný formát pre metódové definície (a model).** Výsledný CASE nástroj má zabezpečovať import a export aj modelov aj metamodelov. Import má byť prírastkový, predchádzajúce importované dáta od rovnakého exportéra sa majú aktualizovať automaticky radšej než tvoriť duplikáty.

5. Metodiky a metódy objektového modelovania

Objektové modelovanie – na akýkoľvek reálny systém môžeme pozerať ako na množinu objektov, medzi ktorými môžu existovať určité väzby. **Cieľom OM** je nájsť vo fáze **analýzy** patričné objekty, nejakým spôsobom ich popísať a predovšetkým začleniť ich do spoločenstva objektov ostatných, nájsť správne vzájomné väzby. Hľadáme hranice systému, vysledujeme väzby systému na okolie.

V nasledujúcich etapách **designu** a **implementácie** už iba rozpracovávame jednotlivé objekty, bližšie definujeme väzby medzi nimi a vôbec, snažíme sa vzniknuté objektové modely nejakým spôsobom konsolidovať, aby nám následne zostala len najnutnejšia množina základných objektov tvoriacich akési jadro systému. Tieto objekty tvoriace tzv. *obchodnú logiku* môžu byť nasledovne previazané napr. So SW objektmi zaisťujúcimi komunikáciu s užívateľom alebo objektmi uloženými v databáze.

Ak sa pozrieme bližšie na dnešný postup pri tvorbe SW systémov, zistíme, že práve vrstva objektov logiky často chýba. Príčina je samozrejme chýbajúca analýza problémov a problémovej oblasti ako celku.

5.1 História objektovo orientovaného modelovania - OOM

Intenzívnejšie sa o OM začalo hovoriť v druhej polovici 80. Rokov, t.j. v dobe, kedy sa začal presadzovať **objektový prístup.** Môžeme hovoriť o troch generáciách objektovo-orientovaného modelovania.

Metodika **prvej generácie objektového modelovania** spadá do obdobia konca 80. rokov a začiatkov 90. rokov. Predstaviteľmi boli Booch a Shlaer/Mellor.

Metodika **druhej** generácie je tzv. **Object Modeling Technic – OMT.** Predstaviteľmi sú Coud/Yourdon, Martin/Odell. Je známa napr. používaním Use-Case modelovania ako jadra metodiky Objectory/OOSE . V tomto období vzniká OMG – Object Management Group – združenie niekoľko stoviek firiem realizujúcich sa na trhu objektových technológií.

Tretia generácia je charakteristická zavedením **Objektovej analýza a návrhu systému** – OOAD. V tomto čase vzniká tiež **UML - Unified Modeling Language** (autori Grady Booch – metodika Booch, Ivar Jacobson- metodika Objectory/OOSE, James Raumbaugh - OMT, najnovšia verzia – OMG UML 1.3 je z leta 1999).

Oblasti použitia OOM:

- **analýza systémov,**
- **návrh a implementácia**
- **business modeling methods - BM,**
- **business processing methods -BPR**
- **dynamické a statické modelovanie**

Metodika OOM využíva množinu nástrojov, obsahuje napr. až 8 typov rôznych diagramov

V ďalšej časti príspevku si priblížime niektoré najnovšie oblasti použitia OOM.

Business Object Notation

Pomerne mladá a jednoduchá metodika uvedená prvýkrát K. Waldenem a J.M.Nersonom v roku 1994. Výrazne je ovplyvnená objektovým jazykom Eiffel, ktorý je vďaka svojej jednoduchosti, konzistentnému návrhu a silným výrazovým prostriedkom používaný predovšetkým pri tvorbe systémov, ktorých kľúčovou vlastnosťou je bezpečnosť a spoľahlivosť.

BON kvôli svojej jednoduchosti neobsahuje niektoré prostriedky, ktoré sú už požadované za takmer nutnú súčasť akékoľvek objektovej metodiky. BON obsahuje pomerne slabé nástroje pre modelovanie dynamiky systému. BON je vedľa Coad/Yourdona veľmi vhodná pre výuku objektového modelovania.

Business Object Relation Medeling

BORM – je viac orintovaná smerom k BPR, BM jako smerom k tvorbe SW.
Vychádza čiastočne z metodiky Coad/Yourdon, z ktorých prijíma vrstvový model a metody OBA (Open bussiness analyza). V nedávnej dobe boli niektoré z nástrojov tejto metodiky implementované do **metacase systému MetaEdit+.**

5.2 MetaEDIT+

Vzhľadom k tomu, že sa u nás na katedre v súčasnosti venujeme problematike implementovania metaCASE systéme na podmienky pošty, priblížim v ďalšom krátku charakteristiku produktu MetaEDIT+.

MetaEDIT+ podporuje najširšiu škálu metód na trhu. Sú štyri základné skupiny metód:

- **Business modelling** (obchodné modelovanie, modelovanie obchodných procesov), napr. Value Chains and Value Systems,
- **Planning and management of systém architectures** (plánovanie a riadenie systémových architektúr), napr. Businnes Systems Planning – BSP,
- **Structured methods** (štruktúrované metódy), Structured Analysis and Design
- **Object – oriented methods** (objektovo – orientované metódy), UML, OMT, OOD OOSD, OOSA.

MetaEDIT+ obsahuje sadu vopred definových metód, ale je možné vytvárať aj vlastné metódy, rozširovať už existujúce a používať ich namiesto vopred definovaných alebo súčasne s nimi. Podporuje prechod od štruktúrovaných k objektovo-orientovaným metódam, umožňuje prepájať a znovu používať dáta medzi rôznymi metódami, udržiavajúc tok informácií medzi nimi.

V MetaCase sa nachádzajú 4 hlavné kategórie nástrojov:

1. *Editory* – sú nástrojmi na prezeranie, editovanie a riadenie špecifických informácií o systémových vývojových projektoch. Patria sem diagramový, maticový a tabuľkový editor (diagram editor, matrix editor, table editor).
2. *Nástoje riadenia prostredia* – zahŕňajú „launcher“ (spúšťač) na riadenie nástrojov prostredia, ako aj schránkové prehliadače (repository browsers) na prezeranie informácií uložených v objektovej schránke.
3. *Nástroje na reporting a generovanie kódu* – slúžia na sprístupnenie informácie v schránke a transformáciu do rôznych reportov, kontrol alebo programového kódu. Patrí sem aj Report Browser, ktorý umpžňuje užívateľovi vytvárať vlastné reporty.
4. *Nástroje vývoja metód*, ktoré formulujú časť prostredia, umožňujúce užívateľovi modifikovať prostredie, metódy, grafické symboly, dialógy a reporty.

Záver

Z metodik a metód objektovej analýzy a návrhu sa postupne stávajú už pomerne vyzreté prostriedky tvorby IS. V posledných niekoľkých rokoch sa ale do popredia záujmu stále výraznejšie dostávajú aktivity vedúce na Business processing, ktoré s tvorbou IS bezprostredne súvisia. I v tejto oblasti majú objektové technológie čo ponúknuť. Ak hovoríme o metodikách OOAD, potom je zrejmý stále väčší príklon k zavádzaniu prostriedkov vhodných práve pre túto špecifickú oblasť. V ideálnom prípade je možné použiť metódy a nástroje jednej metodiky ako nosný prostriedok business modelovania, rovnako dobre ako efektívny prostriedok tvorby IS, ktorý nové pracovné postupy adekvátne podporí.

Objektové technológie nachádzajú stále širšie uplatnenie. Zďaleka ja už nejedná iba o záležitosti spojené výhradne s „computer science“. Hádam preto by sa mali metódy a metodiky budúcich generácií zamerať práve na tento fakt.

Načrtnutá metodika sa môže využiť nielen pri riadení projektov, ale konkrétne pri transformačných prácach, napr. pri reengineeringu na organizačnej štruktúre organizácie, pri zavádzaní nových foriem riadenia, pri odstraňovaní násobnosti entít v rámci organizačnej štruktúry organizácie a pri budovaní celoplošného informačného systému.

Bez vybudovania modernej siete ekonomiky a riadenia podporovaného IT, nám však žiadny model riadenia projektov ani dekompozícia procesov v organizácii a riadení neprinesú také výsledky, aké od neho očakávame. Len včasné a presné informácie, ktoré máme v každej situácii k dispozícii, nám umožnia vykonávať správne rozhodnutia. A tie je možné mať k dispozícii už pri riešení projektov, vo fázach analýz a návrhov.

Literatúra

1. Kremeňová, I.: Dekompozícia procesov v pošte prostriedkami IT, Habilitačná práca, EDIS ŽU, 1999
2. Majerčák, J.: Logistika v doprave, Učebný text pre externých bakalárov št. odboru železničná doprava, Žilina, marec 1999
3. Kremeňová, I.: Automatizované informačné systémy, Praktické cvičenia, ES ŽU 1998, ISDN 80-7100-540-1
4. Smil, P.: Vývoj metodik a metod objektové analýzy a desingu, Softwarové noviny, leden 2000, číslo1, ročník XI
5. Kremeňová, I.: Automatizované informačné systémy, časť I. (pošta, telekomunikácie, Poštová banka, PNS), ES VŠDS 1996, ISDN 80-7100-364-6