

# VÝVOJOVÉ TRENDY METODIK VÝVOJE INFORMAČNÍCH SYSTÉMŮ - VÝZVA BPR.

Václav Řepa

*Vysoká škola ekonomická, Katedra informačních technologií, nám.  
W.Churchilla 4, Praha*

**Klíčová slova:** vývoj informačních systémů, Business Processes Reengineering

**Abstrakt:** V průběhu devadesátých let získaly značnou popularitu programové systémy na podporu vývoje informačního systému - systémy CASE. Tento technologický trend upozornil na to podstatné, co stojí v pozadí těchto produktů – metodiky vývoje informačního systému. Postupem času přestaly být produkty CASE zajímavou a marketingově hodnotnou novinkou a v současnosti je již jejich význam dosti jiný, než na počátku naší dekády. Je to dáno jednak vývojem zmiňovaných metodik, jednak i novými souvislostmi, objevujícími se díky všeobecnému posunu myšlení v oblastech jiných, než čistě technologických.

Jedním z nejvýraznějších rysů devadesátých let našeho končícího století je reengineering podnikových procesů (Business Processes Reengineering - BPR). V něm hraje velice důležitou roli informační technologie se všemi svými souvisejícími aspekty, a to jsou zejména metodiky vývoje informačních systémů. Informační technologie působí jednak jako důvod a jednak jako prostředek dosažení žádoucího reengineeringu podnikových procesů. Na druhou stranu vývoj myšlenek v oblasti teorie reengineeringu podnikových procesů působí jako velice silná zpětná vazba pro samotné metodiky vývoje informačního systému. Staví se tu nové úkoly, se kterými se musí metodiky příslušně vyrovnat. Rovněžtak jsou v oblasti BPR objevovány nové skutečnosti a souvislosti, neočekávaně osvětlující řadu současných problémů metodik vývoje IS/IT.

Tento příspěvek usiluje o shrnutí nejpodstatnějších aspektů této výzvy, zvané BPR a o nastínění jejího vlivu na způsoby, metodiky a nástroje vývoje IS/IT.

## 1. EVOLUCE METODIK VÝVOJE INFORMAČNÍHO SYSTÉMU

V současnosti je situace v oblasti analýzy informačních systémů relativně stabilizovaná. Bitva mezi "strukturovaným" a "objektovým" paradigmatem je ukončena a tak nastává čas přechodu na vyšší kvalitativní úroveň myšlení. Negace již dnes není tím pravým přístupem. Dnešním cílem je učinit metodiku kompletní, konzistentní a připravenou přijmout významné trendy jak v technologii, tak i ve věcné oblasti - v životě samém. Po veškerém prodělaném vývoji zůstává universálním charakteristickým rysem metodik, v němž se projevují jejich veškeré základní principy, *separace různých pohledů na vyvíjený informační systém*. Podstatné a základní jsou přitom následující druhy této "separace":

- *Data versus procesy*, jakož i *objekty versus funkce*;
- *hierarchické abstrakce* - umožní oddělené zkoumání abstraktních pojmů a jejich vztahů, nezátížené jejich detailní dekompozicí, která je brána v úvahu až později - na detailnější úrovni;
- *modelování*, coby klíčový princip, umožní analytikovi abstraktní pohled na obecné charakteristiky informačního systému, nezátížený momentálním stavem věcí, použitou technologií a dalšími abstrahovanými charakteristikami;
- informační systém je budován postupně po jednotlivých *hierarchických úrovních návrhu* - od konceptuální, přes technologický, až do implementační model (tzv. "*Princip "tří architektury"*"), přičemž každá úroveň abstrahuje od specifických charakteristik ostatních úrovní.

Jak je vidět, při oddělování různých pohledů hraje vždy klíčovou úlohu abstrakce. Tento základní rys metodik proto bývá těž nazýván *principem abstrakce*. Hlavním důvodem existence principu abstrakce v metodách analýzy a návrhu IS je snaha *po rozdělení zkoumané problematiky na mentálně zvládnutelné části*.

- typickým rysem problematiky, zkoumané při návrhu informačního systému je totiž vždy její značná rozsáhlost a složitost;
- již samotná předmětná oblast zkoumání je ve všech svých detailech, kterými je třeba se zabývat, značně složitá;
- automatizace se týká vždy značného množství procedur a pracovních postupů, které mohou mít spoustu různých specifických variant ve specifických situacích a jsou ve složitých vzájemných vztazích;
- data, zpracovávaná v informačním systému na jednu stranu vyjadřují celou řadu specifických informací, což je závislé na způsobu jejich zpracování a kombinaci, na druhou stranu jsou ve vzájemných logických, na způsobu jejich zpracování nezávislých, vztazích.

*Oddělování různých pohledů* se tak jeví jako *životně nutné* k tomu, aby vůbec bylo možno tak složitou situaci *mentálně zvládnout*.

Následující text se zaměřuje na stručnou charakteristiku těchto, zde uvedených základních druhů separace, a to v kontextu vývoje metodik analýzy a návrhu IS.

## 1.1 Data versus procesy / objekty versus funkce

Ve strukturovaných metodách analýzy a návrhu informačního systému je rozdíl mezi dvěma základními komponentami IS – daty a procesy, velice významný. Význam tohoto rozdílu je zčásti dán vývojem těchto metod. Zhruba od počátku 70tých let začal být v těchto metodách používán první významný princip – *princip modelování*. Bylo formulováno první významné paradigma - *vývoj informačního systému musí být založen na modelu reálného světa*. Nicméně toto paradigma bylo formulováno pouze z hlediska datové základny systému, což posléze vedlo k nechvalně známému prohlášení Jamese Martina, že data jsou stabilnější, než funkce (které "se mění podle uživatelských požadavků", zatímco data zůstávají) a tudíž pouze datová základna je tím pravým modelem reality. Tato myšlenka započala dlouholetou diskusi o rozdílech mezi daty a funkcemi, která do značné míry pokračuje ještě i dnes. Navíc tyto rozdíly ještě zvýraznil vývoj databázových systémů, které dnes poskytují řadu nástrojů pro podporu "netvůřících" činností funkčního návrhu systému. Za nejdůležitější a jedinou tvůrčí činnost je v tomto modelu myšlení považován návrh datové základny, na vše ostatní již existují více, či méně dokonalé generátory (report writery, nástroje 4GL atd.), které jsou schopny podpořit vývoj těchto částí systému podle neustále se měnících požadavků uživatelů. Pozdější vývoj datové analýzy a databázových systémů sice začal akceptovat i procedurální dimenzi modelu reality (ve formě mechanismů integritních omezení, spouští apod.), nicméně separace již je učiněna: veškeré podstatné náležitosti reality byly předpokládány v datovém modelu.

Na druhé straně i vývoj funkčně orientovaných metod analýzy a návrhu systému přijal "separační paradigma" za své. Například Yourdon (Yourdon, E. (1989)) uvažuje tři oddělené části konceptuálního popisu systému: datový model, model chování a model řízení (poslední dva tvoří funkční model). Toto úzkostné oddělování je zde příznakem neschopnosti vyrovnat se se složitostí vazeb mezi top-down (agregační) hierarchickou strukturou funkcí a neslučitelně jinak (na bázi generalizace) pojatou hierarchickou strukturou entit. Tento rozpor mezi dvěma protichůdnými pojetími hierarchické abstrakce je základem prakticky všech nechvalně známých problémů strukturovaných metod a bude podrobněji diskutován dále.

Strukturovaný přístup k analýze a návrhu tedy pohlíží na informační systém ze dvou různých úhlů pohledu:

- z pohledu dat
- z pohledu funkcí

Každý jeden z nich má svou specifickou logiku a vyžaduje specifické nástroje (DFD versus ERD) k popisu struktury informačního systému.

Objektově orientované metody se pokoušejí tento rozpor překonat zapouzdřením obou - dat i funkcí - do jediného objektu. Jejich techniky a nástroje umožní chápat veškeré změny dat v kontextu operací a naopak. Bohužel je však objektový model schopen postihnout pouze jednu dimenzi reality. Konceptuální objektový model je tvořen sítí objektů, propojených vzájemně společnými akcemi ("spojením metod" objektů). Aby byl takový model čistým modelem reality, nemůže postihovat současně i takové operace, které modelují užití informačního systému samotného. Přitom tyto operace jsou stejně tak konceptuální, jako ty ostatní a tudíž by také měly být modelovány na konceptuální úrovni návrhu informačního systému.

## 1.2 Modelování

Princip modelování je základním principem metod konceptuálního návrhu informačního systému. Metodické postupy a vlastnosti jejich nástrojů a technik vycházejí z myšlenky, že informační systém je modelem reálného systému (reálného světa). Informace v informačním systému nevzniká, je jím pouze zprostředkována, neboť informační systém poskytuje informace o svém okolí (reálném světě), nikoliv o sobě.

Struktura a obsah jednotlivých prvků informačního systému jsou dány strukturou a obsahem jednotlivých prvků reality. Úkolem informačního systému je, na základě údajů o dění v realitě, poskytovat tytéž údaje ve vzájemných vztazích jednotlivých prvků reality. Vstupy dat do informačního systému jsou tedy informacemi o dění v realitě. Výstupy z informačního systému jsou informace o tomtéž, ale v jiné struktuře. Informace se nemění, ale její hodnota vlivem informačního systému vzrůstá (umístěním údajů do vzájemných vztahů na základě znalosti struktury reality zpřístupňuje potenciální skrytou informaci tím, že údajům dodává kontext).

Toto hledisko se začalo uplatňovat jako první v datové složce informačního systému, konkrétně v pohledu na datovou základnu systému. Metodika návrhu systému, vycházející z tohoto hlediska se nazývá datové modelování, nebo datová analýza a její základy tvoří především dílo P.Chena a J.Martina. Modelované objekty reálného světa jsou nazývány entitami a jejich model se skládá z dat o nich. Teorie datového modelování je doplněna principy a technikami, které určují jakou podobu má mít datová

základna, aby objektivně odrážela strukturu a obsah reálného světa (například technika normalizace a kanonická procedura). Základním nástrojem datového modelování je Chenův Entity Relationship Diagram (Chen P.P.S. (1976), který znázorňuje model reálného světa jako síť objektů a je s ním spojena řada pravidel a návodů jak jej tvořit, které rovněž považujeme za techniky a principy datového modelování.

V teorii datového modelování se také objevuje zcela nový pohled na postup návrhu informačního systému, který je znám pod názvem "*koncept tří architektur*". Jde o tříúrovňový pohled na datovou základnu, kde rozlišujeme jednotlivé modely z hlediska jejich obecnosti a konkrétnosti. Tento pohled, který implikuje postup návrhu datové základny od obecného ke zvláštnímu, je v obecnějším pojetí (ne toliko pro datovou základnu, ale pro celý systém) diskutován v následujícím textu.

Princip modelování však má, jak je vidět z jeho obecné formulace (viz výše), obecnější platnost - i jisté části procesů v systému je třeba považovat za model reality. Základním problémem tzv. strukturovaných přístupů k analýze a návrhu IS, které striktně oddělují datový a funkční pohled na věc, je propojení obou - datového a funkčního - modelů reality tak, aby celá zkoumaná realita (která je jediná a z podstaty nedělitelná) byla popsána jedním modelem. V rámci strukturovaného modelu myšlení je jedinou možností pouze formulace integritních pravidel, která popisují různé projevy téhož jevu z různých úhlů pohledů (viz dále). Tento problém je, bohužel, do jisté míry zděděn i do objektově orientovaných metodik analýzy a návrhu, které se stále ještě nevyrovnaly s faktem, že jednotlivým metodám objektu náleží vyšší společný význam i ve smyslu dynamickém, nikoliv pouze statickém, jímž je objekt sám.

Jak z předchozího plyne, je princip modelování, stejně jako ostatní zde uváděné principy, platný obecně a je tak nezávislý na existujícím myšlenkovém modelu (paradigmatu). Každé nové paradigma musí tento princip vstřebat, nemůže jej obejít, ani eliminovat.

### 1.3 Hierarchické abstrakce

Hierarchické abstrakce jsou prostředkem rozkladu prvků vyvíjeného systému do detailní úrovně pohledu. Pojmy vyšší úrovně sestávají z pojmů nižší úrovně. Na každé úrovni podrobnosti jsou popsány jednotlivé její prvky (pojmy) a vazby mezi nimi. Prvky vyšších (neelementárních) úrovní popisu jsou *abstraktními* prvky a mohou být popsány na nižší úrovni pomocí prvků, z nichž každý jeden může být jak abstraktním, tak konkrétním (elementárním . dále již nerozložitelným) prvkem (pojmem).

Obecně existují dva základní typy hierarchické abstrakce:

- abstrakce *část - celek (kolektivizace, agregace)*. Tato abstrakce se typicky používá ve funkčním modelu, kde se dělí systém na subsystémy, části subsystémů atd. Pro agregaci je typická principiální neomezenost dělení. Vyšší celek je zcela definován jako souhrn svých částí (nemá jiný význam).
- abstrakce *specifický podtyp - obecný typ (generalizace)*. Tato abstrakce se typicky používá v datovém modelu (a potažmo i v objektovém modelu), kde je možné uvažovat o jednotlivých specifických variantách nadřizeného pojmu (entity, objektu). Na rozdíl od agregace není nadřizený celek definován jako souhrn podřizených částí, ale jako nositel jejich společných vlastností (atributů).

Je důležité, že tyto dva základní typy abstrakce jsou vzájemně neslučitelné a tím tvoří jádro základního rozporu mezi funkčním a datových modelem (části datového a funkčního modelu jsou tak vzájemně těžko slučitelné).

## 1.4 Princip "tří architektur"

Princip tří architektur definuje způsob použití abstrakce pro vývoj informačního systému po jednotlivých vrstvách. Jednotlivé vrstvy se zaměřují na 3 hlavní aspekty vyvíjeného systému: *obsah, technologii a implementační/realizační specifika*. Tyto hlavní aspekty vyvíjeného systému tvoří přirozenou posloupnost: ze specifikace obsahu systému vyplývají možnosti technologického řešení a konkrétní použitá technologie určuje implementační možnosti.

Návrh informačního systému probíhá ve třech, po sobě následujících architekturách:

- *konceptuální* - zde je vytvořen zcela obecný, čistě obsahový model systému, nezatížený ani technologickou koncepcí řešení, ani jeho implementačními specifiky. Je zde abstrahováno od technologických a implementačních specifik řešení. Konceptuální návrh určuje *CO* je obsahem systému.
- *technologické* - zde je vytvořen model systému, zohledňující technologickou koncepci řešení, tj. ve strukturovaném pojetí koncepci organizace dat (technologie souborová, stromově, síťově, či relačně databázová atd.) a technologickou koncepci jejich zpracování (jazyk 3., či 4. generace, technologické prostředky architektury client - server atd.). Technologický model stále nesmí být zatížen implementačními specifiky řešení. Je zde tedy abstrahováno od implementačních specifik řešení, obsahové náležitosti jsou dány konceptuálním řešením a zde se neřeší.

Technologický návrh určuje *JAK* je obsah systému v dané technologii realizován.

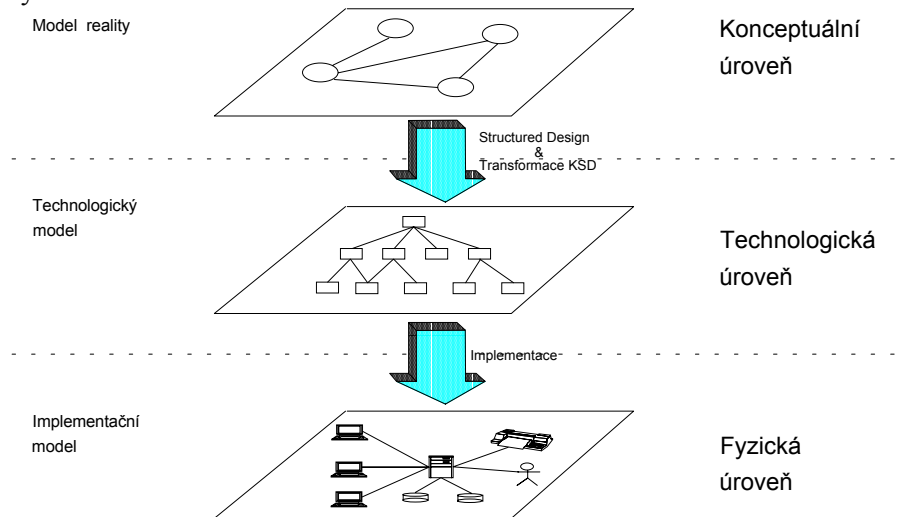
- *implementační* - zde je vytvořen model systému, zohledňující implementační specifika použitého vývojového prostředí (konkrétního databázového systému, programovacího jazyka a dalších prostředků, jako například vývojového prostředí GUI atd.). Není zde abstrahováno od žádných specifik řešení, obsahové náležitosti jsou dány konceptuálním řešením, technologie je dána technologickým řešením, implementační návrh se tedy týká pouze implementačně specifických rysů systému.

Implementační návrh určuje *ČÍM* je technologické řešení realizováno.

Tento koncept tří úrovní modelu systému je rozpracováním použití abstrakce pro odstínění nepatřičných hledisek při tvorbě systému (viz princip modelování) a současně je vidět i v obecně používaných třech základních etapách tvorby systému:

- *analýza*, čili stanovení obsahu,
- *konstrukce (design)*, neboli technologické řešení a
- *implementace*.

Následující obrázek ilustruje různé úrovně návrhu informačního systému:



Obr. 1. Princip tří architektur

Každá ze tří úrovní definuje specifickou architekturu. Každá architektura má svou specifickou logiku a specifický předmět zájmu (obsah, technologii a implementační specifika). Pro metody to znamená:

- pro každou úroveň návrhu mít specifický *jazyk a techniky návrhu*,

- pro každý přechod z jedné úrovně do následující mít specifické *techniky přechodu* z jedné úrovně do druhé.

Následující tabulky obsahují příklady nástrojů a technik různých metod pro činnosti na konceptuální a technologické úrovni:

Tab. 1. Funkční pohled

Úroveň	Činnost	Nástroje	Techniky
KONCEPTUÁLNÍ	Funkční analýza	Data Flow Diagram, Structure Diagram, State Transition Diagram	Event Partitioning
LOGICKÁ	Návrh programových modulů	Structure Chart	Modular Programming, Composite Design, Information Hiding atd.

Tab. 2. Datový pohled

Úroveň	Činnost	Nástroje	Techniky
KONCEPTUÁLNÍ	Datová analýza	Entity Relationship Diagram (Chen)	Normalizace, Integrace (např. Kanonická procedura)
LOGICKÁ	Logický návrh databáze	Entity Relationship Diagram (Martin)	Transformace dat. modelu do logických datových struktur

Tab. 3. Objektový pohled

Úroveň	Činnost	Nástroje	Techniky
KONCEPTUÁLNÍ	Objektová analýza	Objects Diagram, Objects Communication Diagram, State Transition Diagram	Normalizace, Integrace
LOGICKÁ	Objektový Design	Objects Diagram, Objects Communication Diagram	Transformace dat. modelu do logických datových struktur, Information Hiding atd.

Tabulky také ilustrují některé důležité charakteristiky objektově orientovaných přístupů k analýze a návrhu SI:

- OO přístup kombinuje nástroje a techniky jak funkčního, tak datového přístupu
- OO přístup usiluje o společný jednotný jazyk pro popis modelů jak na konceptuální, tak i technologické úrovni
- OO přístup zahrnuje veškeré aspekty datového úhlu pohledu a některé (nikoliv všechny) aspekty funkčního přístupu. Například a zejména zde není místo pro techniku analýzy událostí (Event Partitioning Approach). Tato zvláštnost, jako významný aspekt, ukazující potřebu začlenění činností analýzy a návrhu IS do širšího kontextu modelování věcné oblasti, bude diskutována v následujícím textu.



## 1.5 Integritní pravidla jako nástroj eliminace negativních důsledků oddělování

Integritní pravidla jsou pravidla, jejichž dodržení je nezbytnou podmínkou pravdivosti a bezrozpornosti celku, který je tvořen jednotlivými modely. Metody analýzy (jak strukturované, tak i objektové) používají množství nástrojů k popisu modelu systému na všech třech jeho úrovních: konceptuální, technologické i implementační. Jednotlivé tyto modely (diagramy) představují jednotlivé úhly pohledu, každý se specifickou abstrakcí - každý něco z reality zdůrazňuje, od něčeho účelově abstrahuje. Současné se jednotlivé úhly pohledu zčásti překrývají, neboť popisují jednu a tutéž realitu - jsou přirozeně částečně redundantní. Z obou těchto charakteristik je zřejmé, že k úplnosti popisu je nezbytně třeba popsat také vzájemné souvislosti mezi jednotlivými diagramy, resp. mezi jejich prvky a částmi. Je třeba explicitně stanovit v čem se modely doplňují a co mají společné - jednotlivé redundantní popisy se vzájemně liší svým významem a zde je třeba příslušný specifický význam přesně definovat.

Účelem popisu těchto vzájemných souvislostí je zajistit *integritu popisu jako celku* (vzájemnou bezrozpornost částí). Tato integrace musí jít oběma směry - jak horizontálně (integrace datového a funkčního modelu, integrace objektového modelu s modelem jednání apod.), tak vertikálně (integrace obsahu systému - konceptuálního modelu s jeho technologickou realizací, integrace technologické stavby systému s jeho implementační podobou).

Jak již bylo konstatováno - existenci integritních pravidel je třeba též považovat za obecně (na paradigmatu nezávisle) platnou. Oddělování jednotlivých úhlů pohledu je obecným nástrojem všech metod, a tak i důležitost integritních pravidel zůstává (v objektových přístupech například viz. Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991).

## 2. KLÍČOVÉ PROBLÉMY

### 2.1 Problém konceptuálního modelu

Základním obecným principem analýzy a návrhu IS je *princip modelování*. Podle tohoto principu má být model informačního systému postaven na modelu tzv. *reálného světa*. Přičemž *reálným světem* se rozumí *objektivní podstata činností, které mají být informačním systémem podporovány a skutečností, o nichž mají být v informačním systému uchovávány informace*. S čistým svědomím můžeme v metodách analýzy a návrhu IS takto hovořit pouze o "statických" popisech reality (datový a

objektový konceptuální model). V modelech chování (funkční konceptuální model, nebo model jednání v OO metodikách) jde spíše o modelování chování samotného informačního systému, nežli o čistý model dynamiky reality. Jsou zde totiž modelovány nejenom samotné objekty, ale i (a to zejména) uživatelé informačního systému, nikoliv pouze zdroje informací o realitě, ale i uživatelé těchto informací.

Na druhou stranu je ovšem třeba přiznat, že i způsob, kterým se informační systém má chovat a být používán, má co činit s modelem reality - neboť vyplývá z pravidel jejího chování. Pravidla chování reality, obvykle označovaná jako *business rules*, jsou dána těmi *podstatnými reálnými činnostmi*, které definují *smysl informačního systému* ve formě objektivních věcných požadavků na informace (*business information needs*). Klíčovou se tak stává otázka:  *které z přehršle rozličných reálných činností a procesů jsou ty podstatné a měly by tudíž být informačním systémem modelovány?*

Jisté řešení nabízejí objektově orientované metody analýzy. Model reality ve formě systému objektů, zapouzdřujících data s jim příslušejícími akcemi, je popisem nejen dat, jež mají být v informačním systému uložena, ale i akcí s těmito daty a jejich následností (procesů). Systém konceptuálních objektů tak popisuje tu část dynamiky reálného světa, která vyplývá z podstaty objektů (konkrétně jejich životních cyklů) a jejich vztahů. Avšak již nemodeluje tu část dynamiky reálného světa, která vyplývá z věcné podstaty potřeby informací - z podstaty věcných (tzv. "podnikových") procesů.

Tak tu máme přinejmenším dva druhy "dynamiky" v realitě, která musí být v procesu vývoje informačního systému analyzována:

- dynamika objektů reality a jejich vztahů, daná jejich konceptuální podstatou (podmínky a omezení reálného světa)
- dynamika věcných ("podnikových") činností, daná konceptuální podstatou reálných ("podnikových") procesů (věcná podstata - business nature).

## 2.2 Problém rozporu technik

Modelování dynamiky reálných objektů a jejich vztahů je předmětem zájmu objektově orientovaných metod analýzy a návrhu IS (např. (Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991), nebo Coad P., Yourdon E. (1990)). Objektový model je "statickým popisem" reality v tom smyslu, že popisuje, z jakých objektů a jejich vztahů se realita skládá. Takže například vhodnou technikou k základní identifikaci objektů může být "*technika normalizace*", známá z oblasti datového modelování, která identifikuje objekty pomocí shlukování atributů popisu reality, za účelem jejich *jednoznačného přiřazení entitám (objektům)*. Jedním z nejsilnějších rysů objektově orientovaných metodik je jejich důraz na

konzistenci různých pohledů na data a operace. Akce reálného světa tak musí být jednoznačně identifikovány jako metody konkrétních objektů v konceptuálním modelu - *akce jsou řazeny k objektům*.

Ve funkčním přístupu k modelování reality je, oproti tomu, základní technikou návrhu funkčního modelu "*technika analýzy událostí*", popisovaná Yourdonem (Yourdon (1989)). Tato technika je založena na shlukování akcí reality podle jejich časových (a potažmo věcných) závislostí. Funkční struktura je tak *podřízena událostem*, které jsou vzájemně časově nezávislé - vyplývá z esenciální reálné potřeby kombinace událostí. Při kombinování časově nezávislých událostí se neobejdeme bez ukládání dat o těchto událostech. Avšak uložená data jsou také záležitostí datového (objektového) modelu. Tím, že uložená data (datastory) mají ve funkčním modelu roli jakýchsi "křížovatek" pro kombinování různých událostí (formou komunikace funkcí), je tento pohled, oproti "normalizačnímu" pohledu objektového modelu, přímo protichůdným (ortogonálním) pohledem na uložená data. Přitom ale musí být časové závislosti jednotlivých akcí zjevně a zcela legitimně součástí konceptuálního modelu systému.

Zdá se, že zde máme co činit s rozporem mezi dvěma existujícími přístupy, přičemž každý z nich je zjevně správný. Taková situace dost dobře nemůže znamenat nic jiného, než že zde chybí třetí, oba rozporné pohledy shrnující, pohled na věc. Takový generalizovaný pohled by měl přinést vysvětlení tohoto rozporu mezi pohledy a definovat tak jeho obecný význam.

Ohledně teoretického rozboru možností překonání tohoto rozporu pohledů viz (Jackson, M.A. (1982), Řepa V. (1995), Řepa V. (1996)).

### 3. ANALÝZA PODNIKOVÝCH PROCESŮ V KONTEXTU VÝVOJE INFORMAČNÍHO SYSTÉMU

Shrňme si nejdůležitější závěry z předchozího textu. Máme zde dva protichůdné pohledy na tzv. "reálný svět":

- objektový pohled, zdůrazňující *strukturu* reality
- procesní pohled, zdůrazňující *chování* reality

První pohled představuje *objekty a vazby mezi nimi*, zatímco ten druhý představuje reálné (*věcné, podnikové, business*) *procesy*.

Je ovšem pravdou, že objektový model také popisuje chování - ve formě "životních cyklů objektů", vyjadřujících vzájemné řazení všech metod objektu (tento přístup sice ještě není v současných objektových metodách analýzy obvyklý, nelze však pochybovat o jeho přijetí v, snad blízké, budoucnosti). Jedná se zde však o chování jednotlivých objektů, viděné z pohledu těchto objektů, které neříká nic o nadřazených důvodech k

takovému chování. Tak je třeba chování objektů považovat za strukturální aspekt reality.

Významným aspektem procesního pohledu na chování reality, který nenajdeme v pohledu objektovém, je nutnost najít pro toto chování *nadřazený důvod, nezávislý na obecných pravidlech životů jednotlivých objektů*. Prakticky to znamená, že pro každý věcný (business, podnikový) proces musí existovat nějaký důvod ve formě účelu, cíle a případně i vnějšího podnětu (např. uživatelského požadavku). Business proces, jako shluk časově uspořádaných akcí, které ovlivňují vnitřní stavy objektů a jejich vazeb, je tak něčím víc, než pouze náhodnou hromadou akcí.

Na základě předchozích předpokladů lze považovat *techniku analýzy událostí* (Yourdon, E. (1989)) za techniku, vhodnou právě ke konceptuálnímu modelování business procesů.

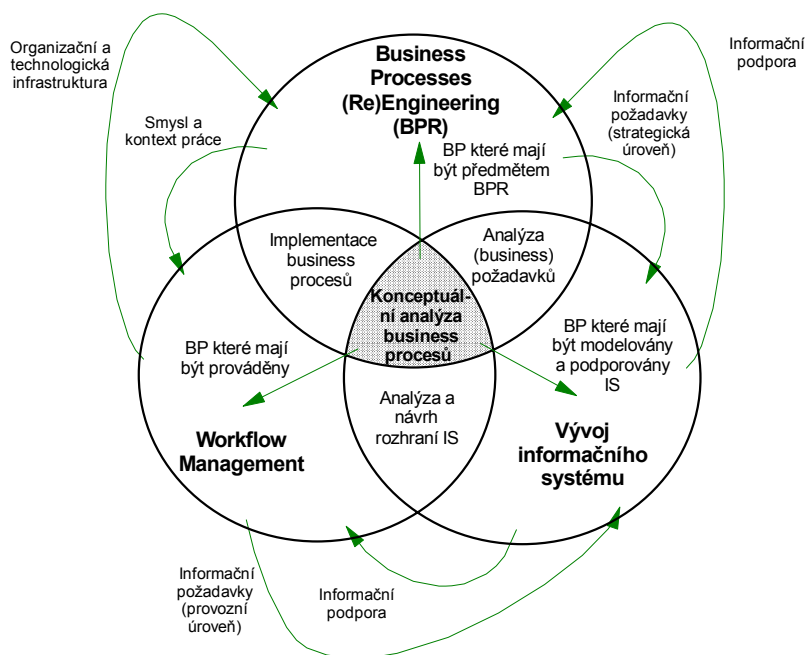
Je zřejmé, že výše popisované dva základní přístupy, jsou různé pohledy na tutéž skutečnost. Jak již bylo zmíněno v první kapitole, taková situace vždy vyvolává potřebu konzistenčních pravidel. Následující tabulka hrubě nastiňuje základní společná fakta, která by měla být předmětem zájmu těchto konzistenčních pravidel:

Tab. 4. Přehled požadavků na konzistenční pravidla (různé významy týchž faktů)

Fakt	Objektový model	Model business procesů
Událost	Podnět k: změně vnitřního stavu objektu možné komunikaci s jinými objekty (zaslání zprávy) jde-li o tzv. "společnou akci"	Podnět k: provedení operace změně stavu procesu produkci produktu možné komunikaci s jinými procesy (koordinace procesů)
Změna dat	Důsledek změny vnitřního stavu objektu	Důsledek: provedení operace (produktu) změny stavu procesu
Výjimka	Výjimečný stav objektu	Abnormální ukončení procesu

Zatímco v teorii řízení (dnes menidžmentu) je orientace na business procesy relativně novým (až módním) jevem, v metodikách analýzy a návrhu informačních systémů nejsou činnosti analýzy business procesů až tak nové. V těchto metodikách lze nalézt řadu různých přístupů k modelování dynamiky reality. Některé jsou zaměřeny přímo na business procesy (Lundeberg M., Goldkuhl G., Nilsson A. (1981), BSP (1984), Turner, W.S., Langerhorst, R.P., Hice G.F., Eilers, H.B., Uijttenbroek, A.A. (1987)), nebo alespoň na procesní - dynamické modelování (Yourdon, E. (1989)). Obvykle jsou však v těchto metodikách činnosti modelování business procesů rozesety mezi ostatní činnosti budování informačního systému ve formě analýzy současného stavu, analýzy informačních potřeb,

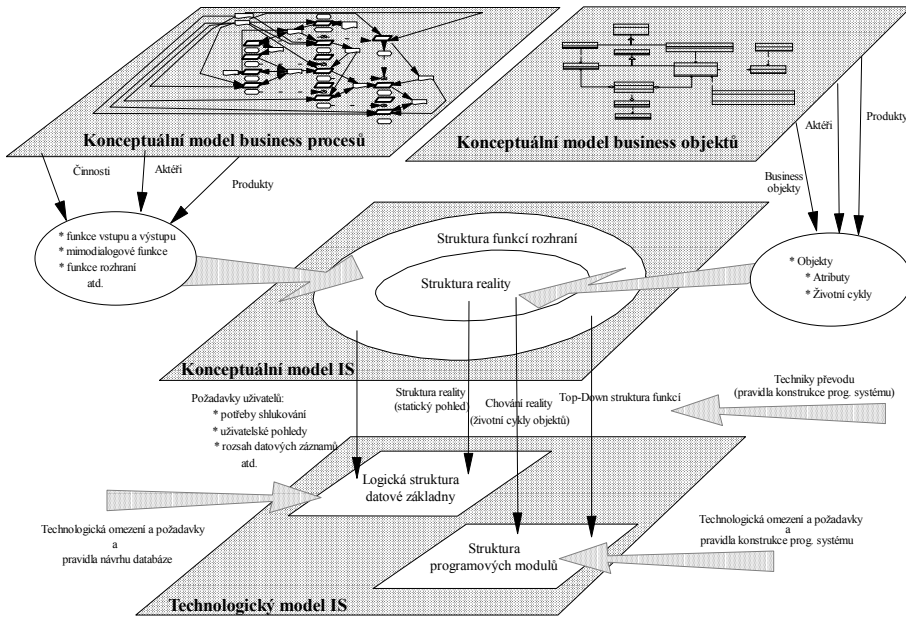
analýzy časových závislostí apod. Jako příklad metodiky asi nevíce orientované na business procesy viz ISAC (Lundeberg M., Goldkuhl G., Nilsson A. (1981)). Skutečně novým pohledem na věc je zde potřeba v metodikách analýzy a návrhu informačního systému *oddělit činnosti modelování business procesů od ostatních modelovacích činností* (tedy modelování statické objektové struktury, jakož i modelování vnitřní dynamiky objektů). Analyzování a modelování business procesů by tak mělo být samostatnou a nezávislou činností, předcházející ostatní činnosti budování IS. Hlavním důvodem tohoto požadavku je skutečnost, že konceptuální model business procesů je zcela universální. Je základem nejenom vývoje informačního systému, ale též implementace workflow, jakož i činností BPR (business process reengineering) - viz Obr. 2.



Obr. 2. BPR vs. vývoj IS vs. Workflow Management

Obr. 3. ukazuje konceptuální model business procesů (BPM) a konceptuální objektový business model (BOM) jako základnu konceptuálního modelu informačního systému. BPM poskytuje informaci o potřebné struktuře funkcí rozhraní ve formě produktů, aktérů a zjištěných potřebných činností. BOM poskytuje informaci o struktuře reality ve formě zjištěných objektů, produktů a aktérů. Ve skutečnosti by měla být většina činností, tradičně považovaných za činnosti vývoje informačního systému, prováděna formou (a za účelem) analýzy business procesů. Konceptuální

model informačního systému má dvě hlavní části. Jádro modelu tvoří objektový model informačního systému, který představuje čistý model reality z hlediska systému. Popisuje ty objekty a vazby reality, které má informační systém podporovat (a tudíž modelovat). Dynamika reality, obsažená v tomto modelu, je zde viděna z pohledu objektů samých - jako jejich životní cykly, definující obecně platná pravidla jejich chování. Okrajová - vnější část konceptuálního model informačního systému představuje model chování reality z hlediska systému. Popisuje ty reálné procesy a jejich vazby, které mají být informačním systémem podporovány (a tudíž jím jsou modelovány). Řečeno jazykem informačního systému zde jde o strukturu funkcí rozhraní vstupů/výstupů.



Obr. 3. Analýza podnikových procesů jako východisko vývoje IS

Závěrem se podíváme na dva základní druhy dynamiky, popisované konceptuálním modelem IS. Dynamika, popisovaná objektovým modelem, vyplývá z podstaty objektů (jejich obecných životních cyklů, které musí být informačním systémem respektovány). Na druhé straně dynamika, popisovaná konceptuálním modelem, vyplývá z business procesů, které mají být informačním systémem podporovány. Základní a podstatnou rolí informačního systému tak je podporovat konzistentní propojení těchto dvou pohledů. Informační systém musí podporovat věčné procesy, přičemž musí

respektovat neporušitelnou podstatu objektů, jichž se procesy týkají, a to ve smyslu konzistenčních pravidel, hrubě nastíněných v *Tab. 4*.

Role procesu vývoje informačního systému, oproti tradičnímu chápání, je tu tak redukována "jen" na modelování a konstrukci vlastního informačního systému, odhlížejíce tak od ostatních aspektů, které přímo nesouvisejí s informačním systémem. Většina analytického úsilí se v této koncepci přesunuje do mnohem obecnější a "na IS/IT nezávislé" oblasti modelování věcných procesů a objektů. Tyto činnosti předcházejí činnostem vývoje IS. K tomu je třeba, aby vznikla nezávislá metodika modelování věcných (business) procesů, založená na technice analýzy událostí. Taková metodika by měla vyhovět potřebám jak obecně manažerským (reengineering procesů, implementace workflow atd.), tak i potřebám vývoje informačního systému, a to ve smyslu souvislostí jednotlivých oblastí, nastíněných na *Obr. 2*.

Pro první nástin takové metodiky viz Řepa V., Bergner M., Chlapek D. (1997), Řepa V. (1998) a Řepa V. (1999).

## LITERATURA

- BSP (1984) "Business System Planning: Information Systems Planning Guide", IBM, GE20-0527-4.
- Coad P., Yourdon E. (1990) „Object-Oriented Analysis“, Prentice-Hall Inc., NJ.
- Chen P.P.S. (1976) "The Entity Relationship Model - Towards a Unified View of Data", ACM TODS, Vol. 1 No.1.
- Donovan J.J. (1994) „Business Re-engineering with Information Technology“, Prentice-Hall Inc., Englewood Cliffs, NJ.
- Goodland M., Mc. Lean J. (1995) „From BPR Vision to IS Nightmare in Business“, in Proceedings of 5th. Conference on Business Information Technology BIT '95, Department of Business Information Technology, Manchester Metropolitan University.
- Greenwood R.M., Robertson I., Snowdon R.A., Warboys B.C. (1995) „Active Models in Business“, in Proceedings of 5th. Conference on Business Information Technology BIT '95, Department of Business Information Technology, Manchester Metropolitan University.
- Hammer M., Champy J. (1994) „Reengineering the Corporation: A Manifesto for Business Evolution“, Harper Business, New York.
- Jackson, M.A. (1982) „System Development“, Prentice-Hall Inc., Englewood Cliffs, NJ.
- Lundeberg M., Goldkuhl G., Nilsson A. (1981) "Information Systems Development - A Systematic Approach", Prentice-Hall Inc., Englewood Cliffs, NJ.
- Řepa V. a kolektiv (1999): *Metody a techniky vývoje informačního systému*, Ekopress, Praha.
- Řepa V. (1999) „Business Processes Based Information Systems Development“, Proceedings of the BIS 99 International Conference, Springer Verlag, London.
- Řepa V. (1998) „Methodology for Business Processes Analysis“, Proceedings of the ISD 98 International Conference, Bled.
- Řepa V., Bergner M., Chlapek D. (1997) „Analýza podnikových činností“, závěrečná zpráva z projektu, Pražská energetika, a.s., Praha.

- Řepa V. (1996) „Object Life Cycle Modelling in the Client-Server Applications Development Using Structured Methodology“, Proceedings of the ISD 96 International Conference, Sopot.
- Řepa V. (1995) „Hybrid development methodology“, in Proceedings of 5th. Conference on Business Information Technology BIT '95, Department of Business Information Technology, Manchester Metropolitan University.
- Řepa V. (1994) „Seeking the Actual Reasons for the „New Paradigm“ in the Area of IS Analysis“, Proceedings of the ISD 94 International Conference, Bled.
- Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991) „Object-Oriented Modeling and Design“, Prentice-Hall Inc., Englewood Cliffs, NJ.
- Scheer, A. -W. (1992) „Architecture of Integrated Information Systems -Foundations of Enterprise-Modelling“, Berlin.
- Scheer, A. -W. (1994) „Business Process Engineering - Reference Models for Industrial Enterprises“, Berlin.
- Turner, W.S., Langerhorst, R.P., Hice G.F., Eilers, H.B., Uijttenbroek, A.A. (1987) „SDM, system development methodology, North-Holland.
- Yourdon, E. (1989) „Modern Structured Analysis“, Prentice-Hall Inc., Englewood Cliffs, NJ.