

## **BORM – Business Object Relation Modeling**

### **Popis metody se zaměřením na úvodní fáze analýzy I.S.**

**Vojtěch Merunka<sup>1)</sup>, Jiří Polák<sup>2)</sup>**

<sup>1)</sup> Katedra informačního inženýrství PEF ČZU v Praze, merunka@pef.czu.cz

<sup>2)</sup> Deloitte@Touche Czech Republic, Management and Consulting Group, jiri.polak@deloitte.cz

#### **Abstrakt**

Tento článek přináší popis metody BORM, která byla vyvíjena jako součást grantového projektu VAPPIENS Britské rady (British Council). Pozornost je zaměřena na úvodní fáze analýzy informačních systémů. Je popisována metoda OBA (Object Behavioral Analysis) a diagram ORD (Object Relation Diagram). Součástí textu je také ukázka analýzy na konkrétním příkladě modelu business procesu ve státní správě. Text navazuje na tematiku objektové analýzy a návrhu publikovanou autory na konferencích Tvorba softwaru v letech 1993 až 1998.

#### **Co je metoda BORM**

Metoda BORM (Business Object Relation Modeling) byla vyvíjena postupně od roku 1993. Od počátku byla orientována na podporu tvorby objektově orientovaných softwarových systémů založených na čistých objektově orientovaných programovacích jazycích a vývojových prostředích, jakými jsou například prostředí Smalltalku (VisualWorks, VisualWave, VisualAge, ...) a objektové databáze (Gemstone, Artbase, ...). Během práce na této metodě bylo zjištěno, že některé její techniky a nástroje je možné využít k reprezentaci znalostí a modelování business procesů.

Práce na BORMu je od svého počátku součástí grantu VAPPIENS (*research project Various Programming Paradigms in Integrated Environments*), který je součástí programu "Know How Fund of Czech Academic Link Programme" Britské rady (British Council). Od roku 1996 je další vývoj také podporován firmou Deloitte&Touche Czech Republic and Central Europe, kde je tato metoda také používána.

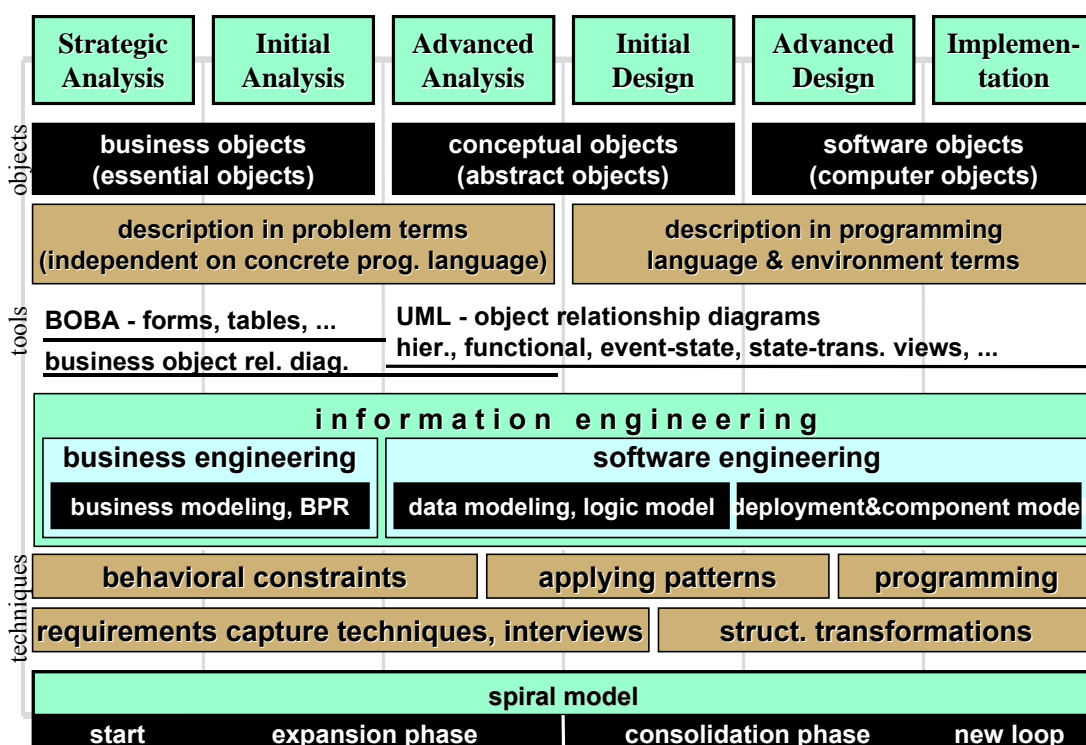
Metoda BORM a především její možnosti analýzy v počátečních fázích vývoje projektu byla prakticky použita například v projektech pro pražské zdravotnictví,

Ústav pro státní informační systém ČR, elektroenergetiku, zemědělství, telekomunikace a plynárenství. Ve všech těchto projektech se ukázalo, že BORM lze dobře využívat jako nástroj pro provádění business process reengineeringu. Výsledky takové analýzy také velmi dobře slouží pro podrobnou a úplnou specifikaci zadání softwarového projektu.

### Jak se BORM liší od jiných metod

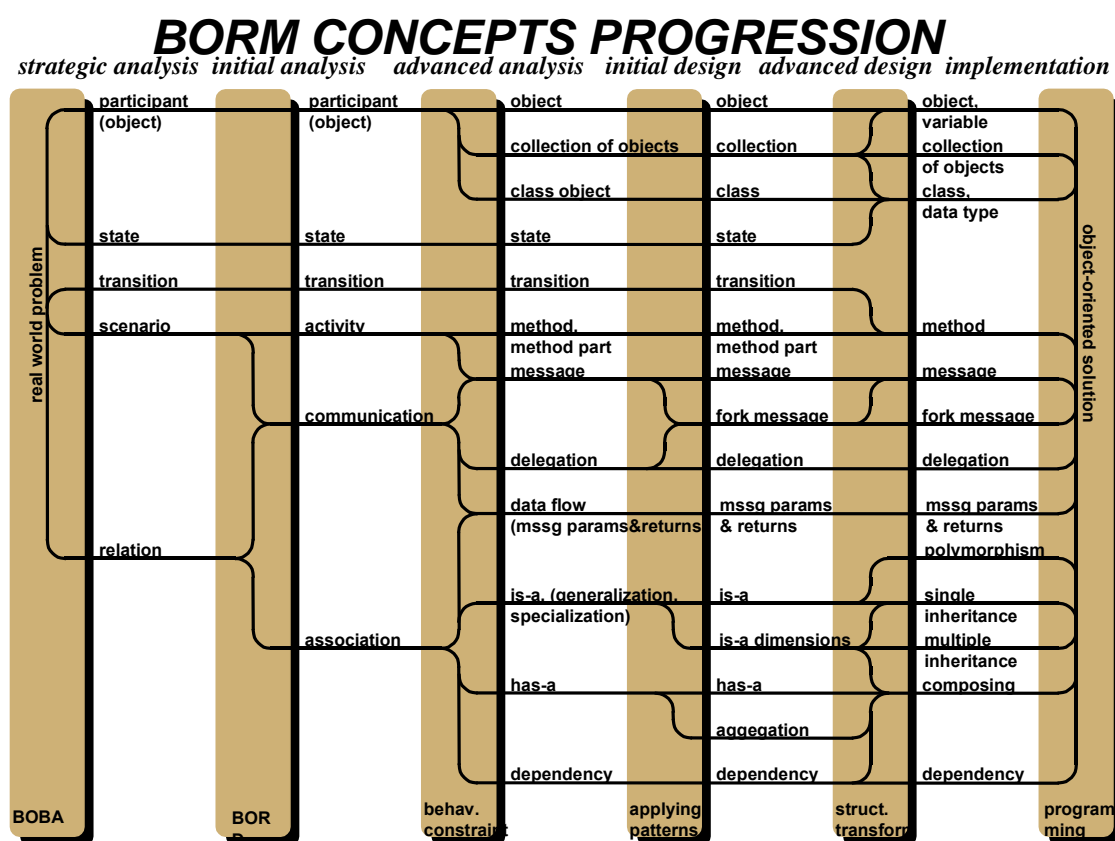
BORM byl navrhován jako metoda, která pokrývá všechny fáze vývoje softwaru (obr.1.). Velká pozornost je na rozdíl od jiných metod věnována úvodním fázím projektu. Nejpoužívanější metody se spoléhají jen na analýzu textového popisu zadání a odvozování objektů a jejich operací z podstatných jmen a sloves v jednotlivých větách. Technika "use-case", která je součástí dnes nejpoužívanějších metod OMT a UML, sice pomáhá při definování interakcí a členění systému, ale poskytuje jen malou podporu pro identifikaci objektů ze zadání. Z "use-case" diagramů se však přímo vytvářejí diagramy sekvencí a komunikací mezi objekty a třídní diagramy. U všech těchto následných diagramů se však již předpokládá, že objekty a třídy jsou již rozpoznány. Stručně řečeno, většina dnes používaných metodologií začíná nad množinou objektů, pro které se například vytvářejí nejruznější softwarově orientované diagramy, ale sofistikovanější postupy, jak tyto objekty v zadaném problému najít a zkontrolovat jejich správnost, v nich chybějí.

## BORM PROJECT LIFE CYCLE



Obr. 1.

Druhou odlišností od jiných metod je skutečnost, že BORM pro každou jednotlivou fázi životního cyklu využívá v diagramech jen omezenou sadu pojmů. Předpokládá se totiž, že během projektování dochází k postupným přeměnám pojmů na jiné (obr. 2.). Například ve fázi analýzy se nepoužívají pojmy jako agregace, jednoduchá či vícenásobná dědičnost, protože tyto pojmy jsou relevantní až pro implementaci. Naopak pojmy jako stav, přechod či asociace jsou používány během analýzy, ale ve fázi implementace, kdy se snažíme model přizpůsobit cílovému implementačnímu prostředí, se s nimi již nepracuje. Nejde jen o postupné zvyšování úrovně detailu ve vytvářeném modelu, ale skutečně o řadu transformací modelu v průběhu životního cyklu. Na rozdíl od přístupu BORMu u metody OMT (a také UML) lze například do object-class diagramu libovolně vkládat kterýkoliv z pojmů bez ohledu na to, jestli se daný diagram týká fáze analýzy nebo návrhu.



Obr. 2

Třetí a poslední specifickou vlastností BORMu je skutečnost, že metoda nevyžaduje oddělování od sebe statických a dynamické pohledů na systém do různých typů diagramů s rozdílnou notací a dovoluje je dokonce v jednotlivých diagramech kombinovat. V BORMu je každý pojem reprezentován shodnými symboly bez ohledu na to, jestli se jedná o diagramy datové struktury nebo komunikací mezi objekty. BORM používá pro znázorňování konceptuálních a softwarových pojmů většinu symbolů shodně s jazykem UML, ale dovoluje v jednom diagramu znázornit například posílání zpráv mezi metodami různých objektů v různých stavech. Tento přístup dovoluje vyjádřit konzistentním způsobem některé žádoucí detaily softwarové konstrukce, které lze výhodně aplikovat především při návrhu pro čistě objektově orientované programovací jazyky. Tento způsob je mnohem jednodušší než např.

tvorba více od sebe oddělených třídnic, stavových a kolaboračních diagramů a také dovoluje zobrazit větší množství informací. Na druhou stranu však samozřejmě platí, že samostatné stavové či interační diagramy jsou v BORMu také používány.

## "Business" analýza I.S.

Pojem „business“ analýza informačních systémů, jak jsou úvodní fáze analýzy I.S. často označovány, je poměrně novou a v současné době velmi rozvíjenou aplikací OOP v praxi. [1] Tato oblast modelování informačních systémů má dvojí na sobě poměrně nezávislý původ:

1. Prvním je potřeba formálního podchycení prvních fází vývoje informačního systému nástroji používajícími jiný pojmový aparát, než nástroje klasického softwarového inženýrství právě z výše popsaných důvodů. Tato potřeba vznikla na základě zkušenosti, že zahájení tvorby IS nějakou „klasickou“ objektovou metodou zbavuje většinu zainteresovaných osob ze strany zadavatele schopnosti sledovat projekt již od samého počátku.
2. Jako druhá příčina rozvoje se považuje rozvoj tzv. „business process“ reengineeringu (BPR) [2] právě v souvislosti se zjištěním, že myšlenky OOP jsou velmi přínosné pro konstrukci nástrojů a technik pro potřeby BPR.

Využití myšlenek OOP a znalostí a postupů známých z „klasických“ metod tvorby softwaru [7,10] a aplikovaných ve fázi business analýzy činí z této fáze důležitou a novou poměrně ucelenou oblast nástrojů a technik, která je označována jako „business“ modelování (*jiní autoři používají též označení „esenciální“*) v návrhu IS. Dochází zde k velmi zajímavému prolínání "business" inženýrství a softwarového inženýrství. Tento proces je někdy označován jako konvergenční přístup v informačním inženýrství. [1,6]

Je velmi zajímavé, že fáze „business“ modelování je v procesu návrhu IS použitelná hned třím možným způsobem:

1. Jako fáze, která předchází „klasické“ konceptuální modelování. V tomto případě je využita jako **předstupeň pro tvorbu konceptuálních modelů** (*kteřé potom lze navrhovat podle OMT či UML*). Zde „business“ modelování plní úlohu spojovacího a komunikačního mezičlánku mezi zadavateli a tvůrci softwaru v úvodních fázích vývoje.
2. Jako fáze, která může být provedena jiným subjektem, než následná „klasická“ analýza a návrh. V tomto případě je použita jako ucelená metoda pro tvorbu **strukturovaných podkladů a dokumentace**, která dále slouží jako výchozí bod tvorby IS prováděné jiným subjektem např. na základě následně provedeného výběrového řízení. [4]
3. Jako samostatná metoda, která slouží pro tvorbu informačního modelu **pro potřeby BPR** a to jak zmapování stávajícího stavu, tak i pro popis návrhu budoucího stavu jednotlivých modelovaných procesů a jejich potřebné změny [2]. Zde se objektové modelování dokonce jako metoda úplně osamostatňuje a vlastně nepřímo dokazuje větší uplatnitelnost myšlenek OOP než pouze v procesu tvorby programů, jak bylo naznačeno v úvodu článku.

Z našich zkušeností vyplývá, že varianta 2) a především varianta 3) je dnes v praxi nejvíce vyžadována, je součástí postupů např. velkých konzultačních firem a lze v blízké budoucnosti očekávat její další rozvoj.

## Úvodní fáze analýzy v UML

Notace diagramu „business“ modelu podle návrhu UML [8] ukazuje, že se jedná o poměrně jednoduchý nástroj, který dokáže popsat systém pouze na velmi vysoké úrovni zjednodušení. Kromě toho, že tento nástroj nedovoluje uspokojivě zachytit mnohé pro zadání důležité informace (*např. pravidla, návaznosti, vazby mezi entitami, ...*), tak je jeho další nevýhodou, která dále snižuje jeho použitelnost pro náležitý popis „business“ procesů, jeho samotná koncepce a pojmový aparát, protože dle názoru autorů má přímou vazbu na programovou strukturu navrhovaného systému, tedy např. na implementaci jednotlivých modulů apod. V případě potřeby zachycení většího detailu o modelovaném systému je doporučováno použití následných nástrojů UML, jako např. object-interaction diagram, object-class diagram atd.

Bohužel ani jeden z vyjmenovaných nástrojů nedosahuje takové míry srozumitelnosti, jednoduchosti a zároveň výstižnosti, jaké ve své době dosahoval dodnes známý a používaný ER diagram nebo DF diagram. Například notace UML sice pokrývá většinu situací, se kterými se při projektování může setkat tvůrce softwaru (*i když dle našeho názoru i zde jsou ještě rezervy především směrem k návrhu objektových databází*), ale je pro neprogramátora velmi komplikovaná a obtížně čitelná. Na rozdíl od dříve používaného ER či DF diagramu, který byl laikovi čitelný po čtvrt hodinovém výkladu (*zde je myšleno pouze postačující pasívní pochopení - ne schopnost takový model vytvářet*), nelze prostředky UML nasadit například při jednáních se zadavateli a s pomocí nich ověřovat a korigovat správnost zadání, což by jistě bylo žádoucí, jenomže tyto prostředky jsou orientovány více na podporu softwarové implementace než na možnost podrobného modelování z pohledu vlastního zadání.

## Úvodní fáze analýzy – business inženýrství – v BORMu

Úvodní fáze analýzy jsou v BORMu podporovány technikou analýzy objektů podle chování (OBA – Object Behavioral Analysis) a nástrojem ORD (Object Relation Diagram).

### Metoda OBA

Metoda OBA je první z šesti technik zahrnutých v BORMu, jak ukazuje obrázek č.2. Metoda vnikla počátkem 90. let na základě zkušeností s aplikacemi technik JAD (*Joint Application Design*) a CRC (*Class-Responsibility-Collaborator*) [9] pro potřeby objektové analýzy a návrhu a implementace v objektově orientovaných programovacích jazycích.

Jedná se o iterativní techniku začínající řízeným interview se zadavateli a pracující s různými typy formulářů, tabulek a modelových karet, ke kterým přísluší sada postupů a pravidel. Podrobnější popis OBA analýzy lze například nalézt v [5]. Jednotlivé kroky OBA analýzy jsou následující:

1. krok - **rozpoznání procesů** (plánování scénářů). V tomto kroku se na základě provedeného interview sestaví **seznam požadovaných funkcí systému a seznam scénářů systému**. Jedná se vesměs o textové popisy, přičemž v nejjednodušší variantě se u každého scénáře rozlišuje původ procesu, vlastní popis procesu, participující objekty a popis výsledku procesu.
2. krok - **definování objektů pomocí modelových karet**. V tomto kroku se pro každý rozpoznáný objekt z předchozího kroku vytvoří jeho modelová karta, která obsahuje jméno objektu, seznam aktivit objektu a s ním související seznam s modelovaným objektem spolupracujících objektů. Předpokládá se, že pro každý rozpoznáný spolupracující objekt je také vytvářena jeho modelová karta.
3. krok - **klasifikace objektů**. V tomto kroku dochází k přidání další informace k modelovým kartám jednotlivých objektů. Modelové karty jsou tříděny podle různých kritérií a podle určitých pravidel dochází ke vzniku nových modelových karet s novými objekty.
4. krok - **sestavení tabulky vztahů mezi objekty**. Tabulka vztahů v nejjednodušší podobě vyjadřuje jaký objekt má vztah s jiným objektem.
5. krok - **modelování životních cyklů objektů**. V tomto kroku se pro každý rozpoznáný objekt s pomocí informací v tabulce scénářů, modelových kartách a tabulkách vztahů sestaví životní cyklus objektu jako sled jeho stavů a přechodů mezi těmito stavy v podobě procesního diagramu. Tento poslední krok lze v případě první iterace provést ihned po kroku 1. a 2. a teprve poté provést kroky 3. a 4.

Metoda OBA je přímo založena na předpokladu iterativního přístupu k analýze. Například jednotlivé scénáře z 1. kroku jsou v 5. kroku příslušným předepsaným způsobem konfrontovány s životními cykly jednotlivých objektů a kontroluje se jejich vzájemná úplnost a souvislost. Následné kroky OBA tedy mohou posloužit i jako podklady pro dodatečné upřesňování informace v krocích předchozích. *(Pro varianty známých řešení se doporučuje provést 2 až 3 opakování všech kroků, ale pouze jeden průběh nestačí).*

OBA pomáhá získávat strukturovaným způsobem potřebné podklady k sestavení prvotních objektových diagramů. Má však i další zajímavé přínosy do procesu tvorby I.S.:

1. poskytuje prostředky pro dokumentování projektu od samého počátku,
2. modelové karty a další výstupy OBA jsou znovupoužitelné v dalších podobných projektech [3] (například jako návrhové vzory) a
3. úsilí vynaložené při sestavování scénářů a životních cyklů objektů lze využít při návrhu optimální funkčnosti uživatelského rozhraní.

Metodu OBA lze provádět dokonce jen s tužkou v ruce a příslušnými předtištěnými formuláři a tabulkami na papíře. Samozřejmě lepším způsobem je použití CASE nástroje, který dokáže většinu rutinních operací (*například různé vzájemné kontroly, udržování projektových dat v konzistentním tvaru a možnost tisku tabulek a formulářů*) provádět automaticky.

## CASE Nástroj pro BORM

Součástí vývoje metodologie BORM byl i výběr vhodného CASE nástroje. V současné době je metoda BORM již implementovaná v nástroji Metaedit Plus<sup>1</sup>. V tomto nástroji jsou pořízeny i obrázky z přílohy k tomuto článku.

CASE Metaedit Plus je dle údajů firmy Metacase software kategorie CAME – Computer Aided Method Engineering. Znamená to, že obsahuje prostředky pro implementaci nových metod. Nová metoda se v Metaeditu Plus poměrně snadno implementuje pomocí definování jejího metamodelu, [11] k němuž se prostředky vizuálního programování přidají příslušné dialogy a vzhled a chování symbolů. Metaedit Plus je naprogramován nad objektově orientovanou databází<sup>2</sup>. Informace uložená v databázi je prezentovatelná a zpracovatelná nejen v podobě grafů, ale i jako tabulky a matice a také přímo uvnitř databáze pomocí různých browserů. Součástí Metaeditu Plus je i skriptový jazyk, ve kterém jsou předdefinovány různé výstupy ve formátu TXT, RTF, HTML, GIF a PCT. Protože systém je otevřený, je možné programovat další výstupní sestavy pro potřeby dokumentace.

## Diagram ORD

Diagram ORD slouží k vizuální reprezentaci informace získané metodou OBA. Jedná se o velmi jednoduchý diagram, který obsahuje jen malý počet pojmů a symbolů, které jsou plně postačující pro prvotní popis modelovaných procesů. (*V dalších fázích projektování je ORD transformován do podoby podobné diagramům UML výše naznačeným způsobem*) Pojmy ORD jsou následující:

---

<sup>1</sup> Metaedit dovoluje nejen projektovat podle nějaké metody (v základní sadě nabízí dokonce 14 metod včetně OMT a UML), ale také naprogramovat metodu vlastní. Podrobnosti o vlastnostech tohoto nástroje lze nalézt na adrese <http://www.metacase.com>.

<sup>2</sup> Jedná se o nerelační objektovou databázi Artbase firmy ArtInApples Ltd. (<http://www.artbase.sk>).

<b>pojem</b>	<b>symbol</b>	<b>popis</b>
Objekt = Participant	Obdélník se jménem zobrazeným uvnitř v levém horním rohu.	Objekt (v úvodních fázích BORMu je označován jako "Participant") představuje účastníka modelovaného procesu, jak byl rozpoznán s pomocí modelových karet.
Stav	Menší obdélník odlišený barvou a typem písma pro pojmenování stavu kreslený dovnitř symbolu pro objekt. (Pro počáteční a koncový stav se používají symboly shodné s UML)	Stavy vyjadřují postupné změny participantů v čase.
Asociace	Silná černá šipka s plným zakončením mezi participanty a nebo stavy. U šipky se píše popis, který blíže specifikuje charakter vazby. Pojmy přirozeného jazyka zde mají přednost před programátorskými označeními typu "dědí", "skládá", ...	Asociace vyjadřují datově orientované vztahy mezi participanty a nebo jejich stavy, protože se mohou v čase měnit. Asociace vyjadřují jednotným způsobem vztahy, které mohou být později upřesněny jako skládání, dědění nebo závislost objektů.
Aktivita	Ovál propojený čarou s participantem nebo jeho stavem. Ovály mohou být kresleny také dovnitř k nim příslušných objektů.	Aktivita reprezentují jednotlivé aspekty chování objektů tak, jak byly rozpoznány pomocí scénářů v modelových kartách.
Komunikace	Šipka, která propojuje aktivity mezi sebou. Malé pojmenované šipky kreslené rovnoběžně k hlavní šipce komunikace vyjadřují datové toky.	Komunikace vyjadřují sled provádění a vzájemnou závislost aktivit různých objektů mezi sebou. datové toky mohou být vedeny oběma směry.
Přechod	Šipka s nevyplněným trojúhelníkovým zakončením, která propojuje aktivity a stavy jednoho objektu.	Součástí přechodu je také aktivita, ze které přechod vychází. Přechod tedy představuje činnost, kterou je třeba vykonat, aby objekt změnil svůj stav.
Podmínka	Přeškrtnutí s textovým popisem u komunikace nebo u propojení aktivity a objektu.	Podmínkou se vyjadřuje omezená platnost komunikace nebo aktivity.

ORD dovoluje modelovat jednotlivé procesy současně dvojitým způsobem:

1. Sekvence stavů a přechodů každého objektu, na které lze nazírat jako na dílčí stavový diagram, vyjadřují roli daného objektu v modelovaném procesu. Tento pohled slouží ke kontrole celkového modelovaného procesu například při interview.
2. Sled komunikací mezi aktivitami různých objektů v různých stavech vyjadřuje průběh vlastního procesu. Celkový proces je tedy znázorněn jako propojení rolí



objektů, které se tohoto procesu účastní. Nazíráme-li na participující objekty se svými stavy a přechody jako na automaty, jedná se vlastně o skládání automatů. [12]

Vzhledem k tomu, že modelovaný proces je konstruován jako propojení rolí (stavů a přechodů) účastnících se objektů, tak ORD dovoluje jednoduchým a nenásilným způsobem zachytit přesný průběh modelovaného procesu a poskytuje tím i prostředky pro ověřování jeho správnosti. (*Do ORD totiž není například možné přidat aktivitu, která by nenavazovala na nějaký již přítomný stav nebo nebyla vázána nějakou komunikací s jinou aktivitou.*) Tyto vlastnosti, které přímo vyplývají z použité teorie, jsou velmi dobře využitelné v interview, ve kterých se diagram sestavuje nebo kontroluje.

Podle našich zkušeností je takto navržený diagram vhodný pro podrobný popis business procesů a práci s nimi. Jedním z jeho největších přínosů je jeho velká srozumitelnost pro experty z problémové oblasti, kteří zpravidla nejsou vyškoleni v technikách softwarového inženýrství.

## Literatura

- [1] Taylor, David A. **Business Engineering with Object Technology**, John Wiley 1995 ISBN: 0471045217
- [2] Darnton, Geoffrey, Darnton, Moksha: **Business Process Analysis** International Thomson Publishing 1997 ISBN: 1861520395
- [3] Partridge C.: **Business Objects - Reengineering for Reuse**, Butterworth-Heinemann 1996, ISBN 07506-2082X
- [4] Yourdon E.: **Mainstream Objects - An Analysis and Design Approach for Business**, Prentice Hall 1995 ISBN 0-13-209156-9
- [5] Goldberg Adele, Kenneth Rubin S.: **Succeeding with Objects - Decision Frameworks for Project Management**, Addison Wesley 1995, ISBN 0-201-62878-3
- [6] Finkelstein Clive.: **An Introduction to Information Engineering - From Strategic Planning to Information Systems**, Addison Wesley 1989, ISBN 0-201-41654-9
- [7] Satzinger John W, Orvik Tore U.: **The Object-Oriented Approach - Concepts, Modelling and System Development**, Boyd&Fraser 1996, ISBN 0-7895-0110-4
- [8] **The Unified Modeling Language Documents**, Rational Software Corporation, <http://www.rational.com/ot/uml.html>
- [9] Bellin, David; Simone, Susan Suchman: **The CRC Card Book** Addison-Wesley 1997 ISBN: 0201895358
- [10] Norman R.J.: **Object-Oriented Systems Analysis and Design**, Prentice Hall 1996, ISBN 0-13-242819-9
- [11] B. Henderson-Sellers, A.P. Bulthuis, Rijswijk, **Object-Oriented Metamethods**, Springer Verlag 1997, ISBN 0-387-98257-4
- [12] Mellor Stephen, Shlaer Sally: **Object Lifecycles: Modeling the World in States**, ISBN 0136299407

## Příloha – příklad business analýzy

Na tomto místě přinášíme příklad analýzy konkrétního zadání podle metody BORM. jedná se o jeden "business" proces, který popisuje průběh řízení občana a orgánů státní správy pro vyřizování stavebního povolení. V procesu jsou zohledněny tři varianty staveb (běžná, jednoduchá a drobná) tak, jak je vymezuje stavební zákon č.50/1976 Sb. ve znění zákona č.103/1990 Sb. včetně souvisejících prováděcích vyhlášek.

Příklady jsou vybrány z konkrétního projektu pro Ústav pro státní informační systém ČR z roku 1998.

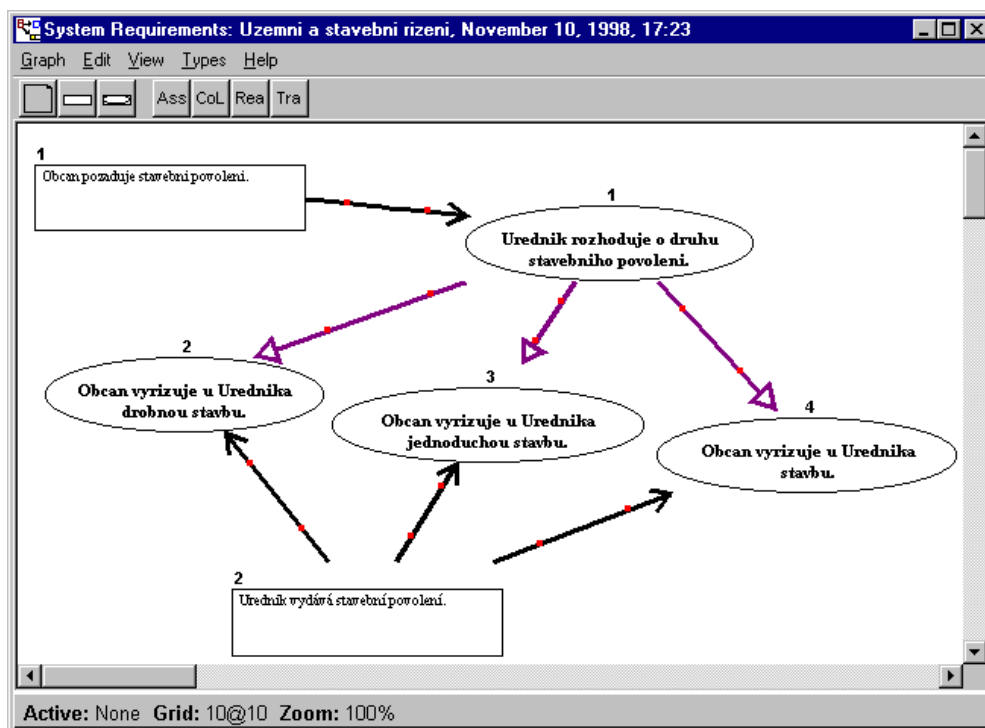
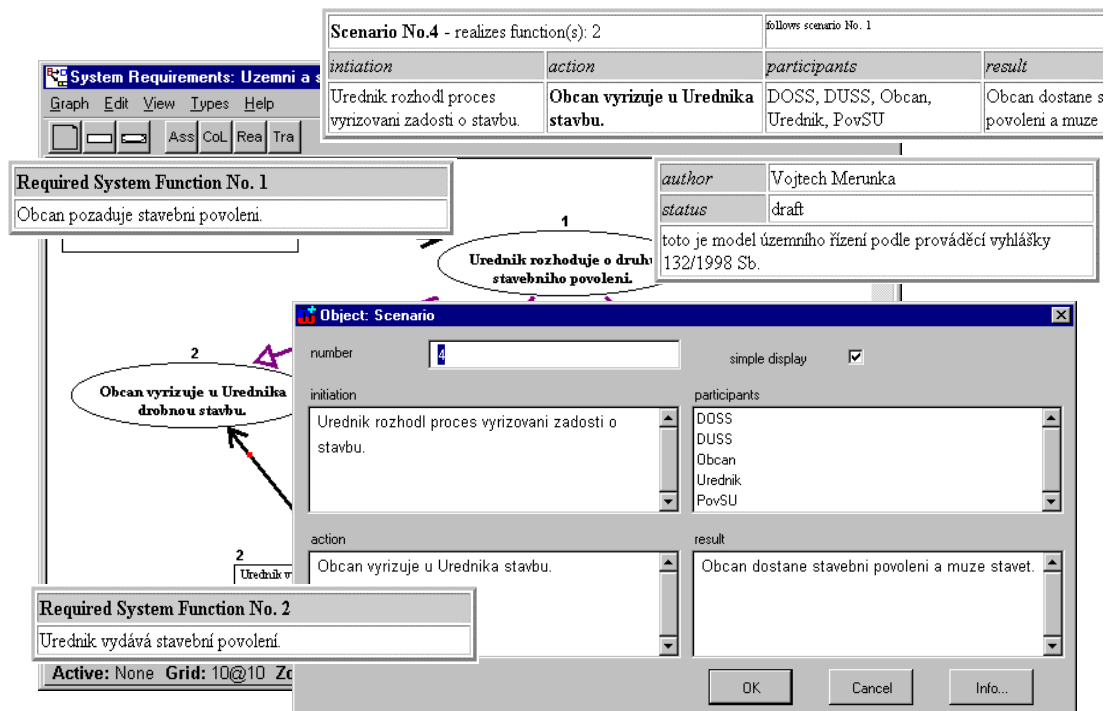
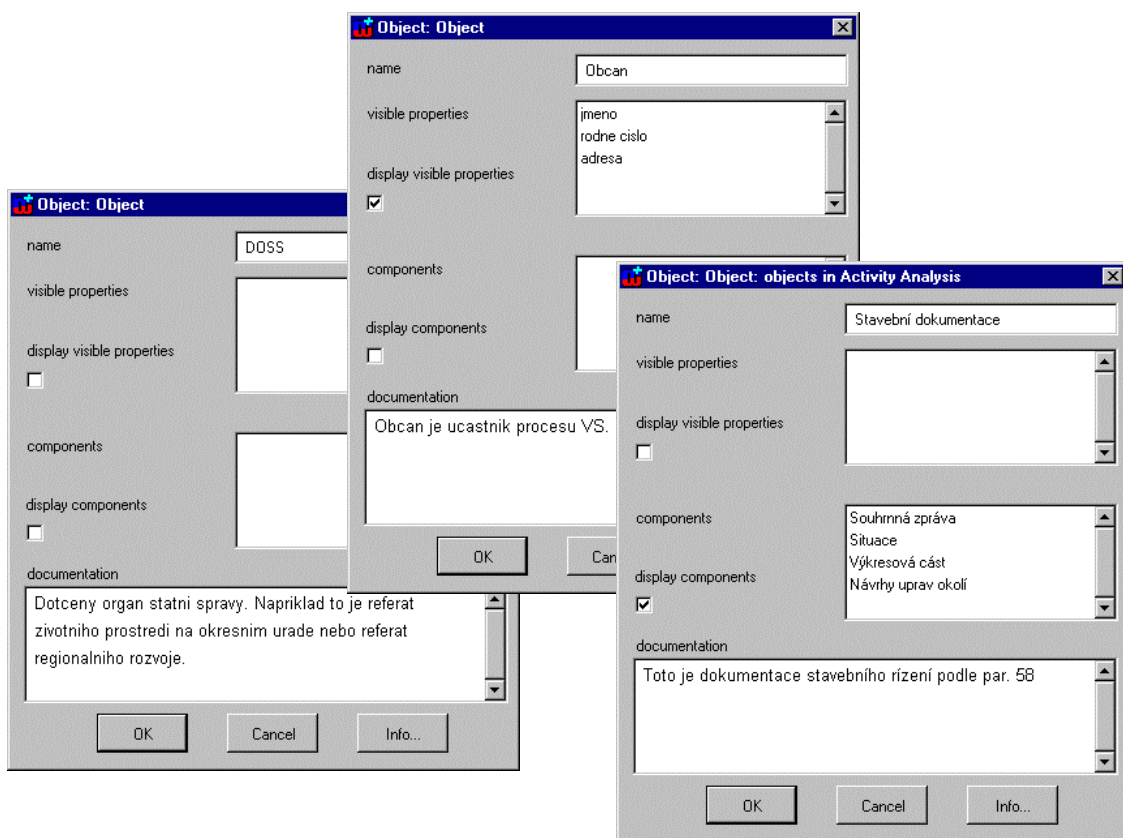


Schéma vyjadřující vzájemné souvislosti mezi požadovanými funkcemi procesu (obdélníky) a scénáři procesu (ovály).



Ukázka jednoho scénáře v detailu a příklad dokumentace generované ve formátu HTML.



Ukázky modelových karet v podobě dialogů v CASE nástroji Metaedit.

Občan		components:	
<i>activities</i>		<i>visible properties</i>	
dostava povoleni prijima sdeleni sestavuje dokumentaci pro stavebni pov. sestavuje dokumentaci pro stavebni pov. sestavuje dokumentaci pro uzemni rozho		jmeno rodne cislo adresa	
Občan je ucastnik procesu VS.			

Urednik	DOSS	DUSS	Obcan	PovSU	Urednik
potvrzuje zadost					
rozhoduje se				x	
stavebni rizeni	x	x	x		
stavebni rizeni bez uz. rozhodnuti	x	x	x		
uzemni rizeni	x	x	x	x	
vydava stavebni povoleni a eviduje ho			x		

Urednik	par 139b	par 32	par 34	par 37	par 55	par 57	par 58	par 59	par 60	par 66
<i>Rozhodnuti o typu stavebniho povoleni</i>										
prijima zadost	x									x
rozhoduje se	x									
vydava rozhodnuti	x									
<i>Vyrizovani drobne stavby.</i>										
eviduje dokumentaci drobnych staveb						x				
potvrzuje drobnou stavbu	x									
<i>Vyrizovani jednoduche stavby.</i>										
potvrzuje jednoduchou stavbu	x									

Stavebni dokumentace	GIS	Registr obyvatelstva	Registr nemovitosti
GIS			
Registr obyvatelstva	x		x
Registr nemovitosti	x		x

Registr nemovitosti inputs	from object (activity)	parameters	return
Obcan (ohlasuje stavbu)		Stavebni dokumentace	
Obcan (predava dokumentaci)		Stavebni dokumentace	

Vyrizovani stavby	GIS	Registr obyvatelstva	Registr nemovitosti
DOSS			
vyjadruje se k uzemnimu rizeni	x	x	x
vyjadruje se ke stav. rizeni	x	x	x

Ukázka tabulek různých vztahů a vlastností objektů. (např. dotčené paragrafy stavebního zákona jsou pojednány jako atributy aktivit)

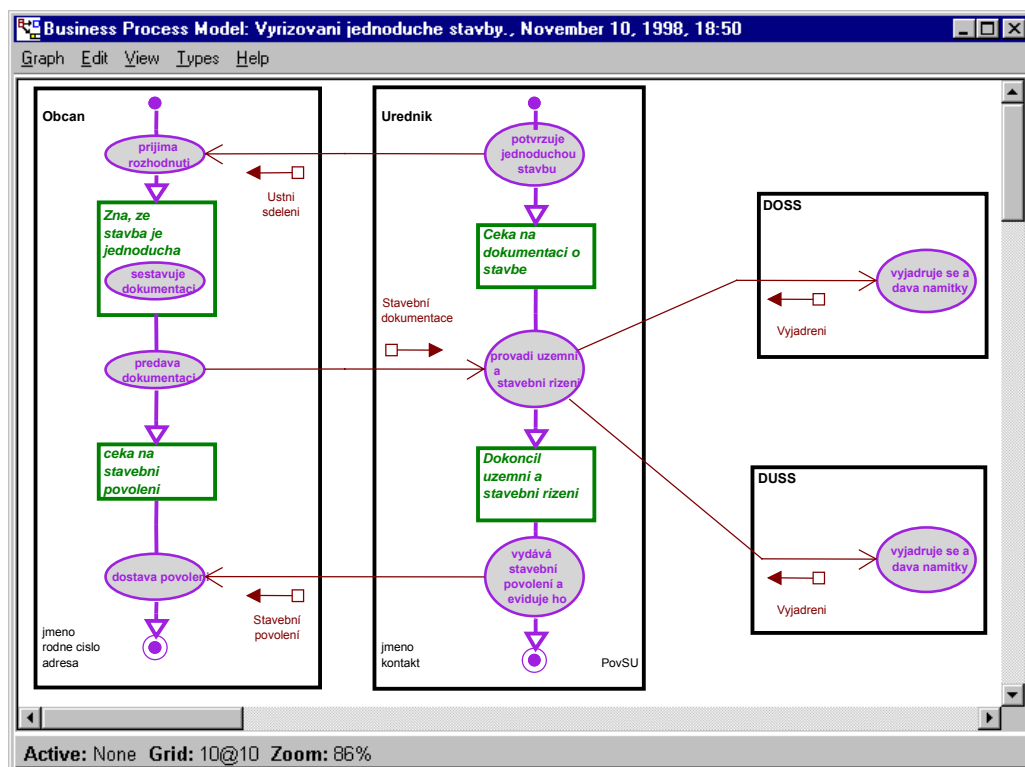
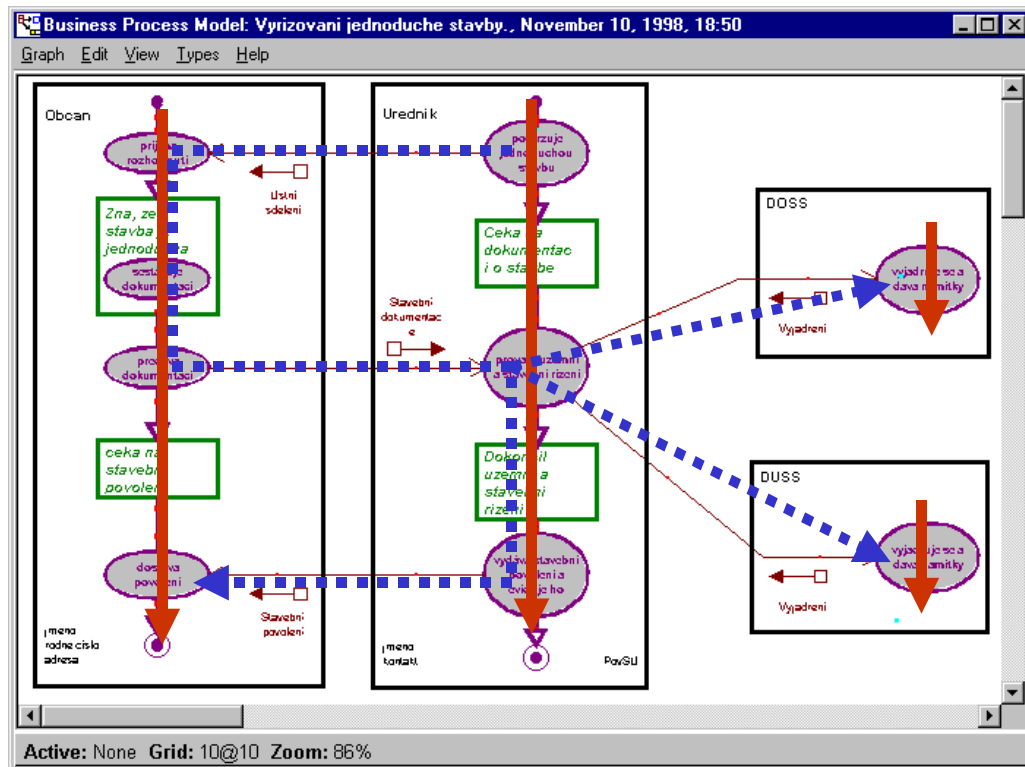


Diagram procesu vyřizování jednoduché stavby. Účastníky procesu jsou objekty Občan, Úředník, Dotčený Orgán Státní Správy a Další Účastník Stavebního Řízení.

# Role tvoří proces.



Vysvětlení vztahu mezi procesem a rolemi na procesu se účastnících se objektů.