

# **NORMY A STANDARDS MODELOVÁNÍ PODNIKOVÝCH PROCESŮ**

Autor: **Zdeněk Chadima**  
Konsultant: **Václav Řepa**  
Recenzent: **Martin Matula**

Září 2006

## **Abstrakt**

Model podnikových procesů představuje základní vyjadřovací prostředek, který umožňuje smysluplnou simplifikaci komplexních jevů, které jsou každodenní součástí chování každého podniku. Aby bylo možné procesní modely vytvářet a používat, je třeba dodržovat určité zásady a principy a užívat vhodného modelovacího jazyka. Hledáním a specifikací těchto zásad a tvorbou modelovacích jazyků se zabývá řada organizací a iniciativ.

Obsahem této práce je rozbor základních standardizačních prací v oblasti modelování podnikových procesů. Text je rozdělen do dvou základních oddílů. První oddíl je věnován „de iure“ standardům, které obsahují základní principy podnikového modelování. Druhá část práce pak obsahuje popis standardů modelovacích jazyků a tvorbu obecného metamodelu těchto jazyků. Text se věnuje jazykům IDEF 3, BPML, BPMN a UML. Zvláštní pozornost je věnována grafickým notacím jednotlivých jazyků. Na základě rozboru vlastností elementů těchto notací je pro každý jazyk vytvořen jeho metamodel.

Závěrečná kapitola je věnována tvorbě a popisu generického metamodelu, na kterém je demonstrováno uplatnění obecných modelovacích principů a práce se základními elementy procesních modelů v jednotlivých notacích.

## **Abstract**

Business process model represents basic tool that allows reasonable simplification of complex phenomenons that take part in everyday behavior of a common company. Creation and usage of process models is conditioned by observance of certain basic principles and application of a suitable modelling language. Constituting and specification of these principles and creation of modelling languages is an object of interest for many organizations and initiatives.

The goal of this text is description of fundamental standards in business process modelling domain. Text is divided into two basic parts. The first part concerns with “de iure” international standards which include basic principles of enterprise modelling. Section B of the thesis is focused on standards of modelling languages and creation of generic metamodel of these languages. Text contains description of IDEF 3, BPML, BPMN and UML. Special attention is devoted to graphical notations of particular languages. Based on notation’s elements features is then set up metamodel of the language.

Final chapter is devoted to creation and description of generic metamodel. Metamodel serves for demonstration of basic modelling principles and application of basic elements of process models in particular languages.

# Obsah

<b>1</b>	<b>ÚVOD.....</b>	<b>2</b>
1.1	CÍLE A ÚČEL TOHOTO TEXTU .....	2
1.2	STRUČNÝ PŘEHLED OBSAHU PRÁCE.....	3
1.3	VÝZKUMNÉ METODY.....	3
<b>A.</b>	<b>DE IURE STANDARDY PRO MODELOVÁNÍ .....</b>	<b>5</b>
<b>2</b>	<b>STANDARD ISO 14258.....</b>	<b>5</b>
2.1	ZÁKLADNÍ ASPEKTY MODELŮ .....	6
2.1.1	<i>Hierarchie.....</i>	<i>6</i>
2.1.2	<i>Struktura.....</i>	<i>6</i>
2.1.3	<i>Chování.....</i>	<i>7</i>
2.2	ZÁKLADNÍ POHLEDY NA SYSTÉM PODNIKU.....	7
2.3	SHRNUTÍ.....	8
<b>3</b>	<b>STANDARDY CEN ENV 40003:1990 A CEN ENV 12204:1995.....</b>	<b>9</b>
3.1	RÁMEC PRO PODNIKOVÉ MODELOVÁNÍ ENV 40003.....	9
3.1.1	<i>Pohledy na strukturu podniku.....</i>	<i>10</i>
3.2	CEN ENV 12204 .....	11
3.2.1	<i>Konstrukty.....</i>	<i>11</i>
3.3	SHRNUTÍ.....	17
<b>B.</b>	<b>DE FACTO STANDARDY MODELOVACÍCH JAZYKŮ.....</b>	<b>19</b>
<b>4</b>	<b>METODOLOGIE IDEF .....</b>	<b>19</b>
4.1	IDEF3 – METODA POPISU PROCESŮ .....	22
4.1.1	<i>Strategie zaměřená na procesy.....</i>	<i>22</i>
4.1.2	<i>Strategie zaměřená na objekty.....</i>	<i>27</i>
4.2	METAMODEL IDEF 3 .....	32
<b>5</b>	<b>JAZYKY BPML A BPMN.....</b>	<b>34</b>
5.1	BUSINESS PROCESS MODELING LANGUAGE.....	34
5.1.1	<i>Základní prvky jazyka.....</i>	<i>34</i>
5.2	BUSINESS PROCESS MODELING NOTATION .....	40
5.3	METAMODEL BPMN.....	47
5.4	SHRNUTÍ.....	48
<b>6</b>	<b>JAZYK UML.....</b>	<b>50</b>
6.1	ARCHITEKTURA UML.....	50
6.2	ZÁKLADNÍ DIAGRAMY UML .....	52
6.3	ACTIVITY DIAGRAM .....	52
6.4	METAMODEL ELEMENTŮ ACTIVITY DIAGRAMU UML .....	58
6.5	SHRNUTÍ.....	60
<b>7</b>	<b>METAMODEL STANDARDŮ MODELOVACÍCH JAZYKŮ .....</b>	<b>61</b>
7.1	VYJÁDRĚNÍ ZÁKLADNÍCH PRINCIPŮ MODELOVÁNÍ .....	61
7.2	ELEMENTY METAMODELU.....	64
7.2.1	<i>Elementy zaměřené na popis chování.....</i>	<i>64</i>
7.2.2	<i>Elementy zaměřené na popis systémové informace .....</i>	<i>67</i>
<b>8</b>	<b>ZÁVĚR .....</b>	<b>68</b>
<b>9</b>	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>70</b>
<b>10</b>	<b>TERMINOLOGICKÝ SLOVNÍK .....</b>	<b>72</b>

# 1 Úvod

Analýza podnikových procesů se postupně stává nedílnou součástí života moderního podniku. Je prostředkem, který firmě umožňuje popsat a porozumět svému vnitřnímu uspořádání a chování. Prostředí permanentního konkurenčního tlaku vyžaduje od ekonomických subjektů, aby byly schopny přizpůsobovat se měnícím se podmínkám na trhu či dokonce měnit svou organizační strukturu a produkční procesy v důsledku nezbytnosti uspokojovat potřeby zákazníků co nejlépe. Ať už se jedná o potřebu zvýšit kvalitu nebo snížit náklady či dodací lhůty, musí podnik velice dobře rozumět sám sobě, aby byl schopen nalézt prostředky, kterými lze náskok oproti soupeřícím firmám získat. Takovým prostředkem je jistě i zlepšování podnikových procesů.

**Business Process Improvement (BPI)** a **Business Process Reengineering (BPR)** jsou termíny používanými pro činnosti spojené s optimalizací a zlepšováním podnikových procesů. Aby byly osoby, zabývající se oblastí podnikových procesů, schopny procesům porozumět a hledat způsoby jejich zlepšení, je třeba využívat takové způsoby popisu, které jsou pro pochopení dostatečně přehledné a jednoduché. V dnešní době se také klade velký důraz na dokonalou spolupráci obchodních partnerů. Mezi partnery je nutné vědět, kdy, kde a jaké činnosti je třeba při spolupráci vykonávat. Je proto třeba najít nástroj umožňující partnerům jejich procesy sdílet a komunikovat. Prostředkem, který je základním východiskem pro metody a techniky BPI a BPR a který umožňuje komplexní problematiku podnikových procesů postihnout srozumitelným a komunikovatelným způsobem je model podnikových procesů.

## 1.1 Cíle a účel tohoto textu

Přístupů k modelování podnikových procesů existuje celá řada a jsou celosvětově předmětem výzkumné činnosti různých standardizačních organizací, konsorcií a iniciativ. Cíle této práce jsou následující:

- popsat základní standardizační práce mezinárodních standardizačních organizací v oblasti modelování procesů
- popsat standardy modelovacích jazyků IDEF, BPEL, BPMN a UML v 2.0, jejich notace a přístupy k modelování a umožnit tak jejich srovnání

- ke každému standardu modelovacího jazyka vytvořit metamodel
- z dílčích metamodelů vytvořit metamodel obecný

Základní motivací a smyslem práce je dát čtenáři možnost poznat popisované standardy, metody a nástroje a proniknout tak do filozofie přístupu, který daná organizace při modelování procesů uplatňuje, a umožnit tak porovnání používaných notací, pojmů a vazeb metod na mezinárodní standardy. Dílčí metamodely a finální obecný metamodel, by pak měly zprostředkovat hlubší proniknutí do zákonitostí notací pro modelování podnikových procesů a měly by sloužit pro další využití v rámci projektu OpenSoul. Tento projekt je rozvíjen na KIT VŠE a zabývá se vývojem nástroje pro metamodelování, který by podporoval tvorbu metamodelů v jazyce MOF a jejich výměnu mezi různými návrháři.

## 1.2 Stručný přehled obsahu práce

Práce je členěna do dvou základních oddílů. První oddíl se zabývá „de iure“ mezinárodními standardy pro modelování podnikových procesů, objasňuje jejich základní principy a požadavky na podnikové modely. Druhý oddíl je pak věnován standardům modelovacích jazyků a jejich notacím. Rozebírány jsou jazyky IDEF 3, BPEL, BPMN a UML. Každému jazyku je věnována samostatná kapitola, ve které jsou představeny jeho základní vlastnosti a pozadí jeho vzniku, následovány rozborem elementů notace a metamodelem jazyka. Závěrečná kapitola je pak věnována obecnému metamodelu, jeho rozboru a porovnání elementů jednotlivých jazyků.

## 1.3 Výzkumné metody

Tato práce je postavena především na studiu plných znění specifikací popisovaných standardů. Vzhledem k charakteru těchto dokumentů spočívala tedy metoda sběru podkladů a informací především ve vyhledávání a studiu relevantních dokumentů, které jsou dostupné v síti Internet. Tyto dokumenty lze rozdělit do dvou základních skupin:

- dokumenty, které jsou vydány organizací, která příslušný standard či metodu vyvíjí, a které jsou zpravidla základními dokumenty, standard či metodu definujícími

- ostatní dokumenty, které se k dané problematice vztahují (příspěvky z konferencí, články, studentské vědecké práce, webové stránky, atd.)

Metamodely byly sestaveny na základě vlastností elementů, které jsou součástí jejich popisu ve standardizačních dokumentech jednotlivých jazyků. Práce je částečně založena na bakalářské práci [CHAD01] a využívá jejích přínosů.

## **A. De iure standardy pro modelování**

V této části práce představím fundamentální práce mezinárodních standardizačních organizací v oblasti podnikového modelování. Smyslem těchto prací je vytvořit obecné principy a podmínky tvorby podnikových modelů (např. sjednocováním názvosloví, definováním základních pohledů na podnik, nezbytných vlastností modelů, základních cílů modelování apod.). Obecné poznatky ustanovené v těchto pracích pak mohou sloužit jako východiska pro tvorbu a rozvoj metodologií, metod a nástrojů modelování. Obecné standardy takto vytvářejí základní rámec, který analytikům umožňuje identifikovat, dokumentovat a sdílet klíčové pojmy.

Rozhodl jsem se věnovat pozornost dvěma mezinárodním standardům: ISO 14258, CEN ENV 40003:1990 a CEN ENV 12204:1995. Tyto standardy jsou zajímavé zejména proto, že se vztahují k takovým metodologiím modelování podnikové architektury jako jsou například GERAM, CIMOSA, GRAI/GIM, ARIS. Předkládají obecné principy, kterými je třeba se při modelování zabývat, jejichž užití je možné vysledovat i ve standardech a metodách popisovaných v oddíle B této práce.

## **2 Standard ISO 14258**

Standard ISO 14258 – Systém průmyslové automatizace – Koncepty a pravidla pro podnikové modelování (ISO 14258 – Industrial automation system – Concepts and rules for enterprise modelling) je iniciativou mezinárodní standardizační organizace ISO (International organization for standardization). Základním smyslem tohoto standardu je „definovat pojmy a pravidla pro podnikové modely“ [ISO01], čehož se standard snaží dosáhnout „definováním elementů, které se mají použít při tvorbě modelu, konceptů pro životní fáze systémů, a způsobů jak modely popisují hierarchii, strukturu a chování (viz. dále).“ [ISO01] Standard nepopisuje žádné konkrétní modelovací metody či nástroje, ale poskytuje obecný rámec toho, jakým způsobem by se mělo k modelování podniku přistupovat, a co by měly metodologie či standardy modelování postihovat.

## 2.1 Základní aspekty modelů

Modely považuje standard za nástroje, které slouží k popisu podniku, a které mají být vždy přizpůsobeny důvodům, jež vedou k jejich vzniku. Podle ISO 14258 by veškeré modely měly respektovat tři elementy systémové teorie:

- aspekt hierarchie
- aspekt struktury
- aspekt chování

### 2.1.1 Hierarchie

„Aspekt hierarchie je založen na principu, že element systému může být sám považován za systém, který je pak nazýván subsystémem. Obdobně může být zkoumaný systém považován za element jiného systému, který je pak nazýván supersystémem.“[ISO01] Tento princip vede při modelování k používání různých úrovní abstrakce, která napomáhá lepšímu pochopení modelovaného systému. Standard vymezuje dva druhy hierarchií:

- **hierarchie částí** – „...reprezentuje kompozici elementů a dekompozici systémů“[ISO01], tedy vztah nadřízených a podřízených systémů. Její užití umožňuje lepší zacílení modelů na zkoumané problémy, a možnost tvorby modelů na různé úrovni dekompozice systému
- **hierarchie druhů** – „...popisuje úrovně abstrakce, které jsou propojeny generalizací a specializací.“[ISO01], tedy vztah nadřízených a podřízených pojmů či tříd. Tato hierarchie umožňuje precizněji definovat entity systému.

### 2.1.2 Struktura

„Aspekt struktury je založen na principu, že elementy nejsou izolované, ale mají několikanásobné vzájemné vazby s ostatními elementy systému.“ Tento aspekt představuje pohled na podnik jako na souhrn elementů či objektů a jejich vazeb, pro jejichž reprezentaci je vhodné používat grafických prostředků.

Standard uvádí dva možné přístupy k chápání struktury podniku. První přístupem je takový, který chápe činnosti jako elementy a objekty (např. materiál, dokumenty, atd.) jako vazby. Příkladem může být proces, v němž výstup jedné činnosti představuje vazbu s druhou činností, jíž je vstupem. Druhý přístup pak považuje objekty za elementy systému a činnosti za vazby mezi nimi.

### 2.1.3 Chování

„Aspekt chování je založen na identifikaci proměnných a jejich funkčních nebo jiných vztahů.“ [ISO01] Tento aspekt vyjadřuje fakt, že každý podnikový systém se neustále mění. Změny jsou zpravidla způsobovány činností lidí a neživých elementů systému. Tuto změnu je třeba zachytit do vhodných modelů, které jsou schopné „...popsat chování, to znamená vyjádřit následnost (čas), události, činnosti, okolnosti, stavy, změny stavů, počáteční stavy, konečné stavy, posloupnost akcí a popis transformačních funkcí.“ [ISO01]

## 2.2 Základní pohledy na systém podniku

Požadavkem ISO 14258 vzhledem k modelům je, aby reprezentovaly ty aspekty podnikového systému, které jsou nutné :

- „k pochopení, návrhu, pořízení a vybudování podniku skládajícího se z jakékoliv množiny zvolených procesů
- k řízení a provozování podniku tak, aby mohl dosahovat svých cílů
- pro podporu schopnosti podniku modely modifikovat, znovu navrhovat, nebo odbourávat“ [ISO01]

Za zásadní pro tvorbu modelů považuje standard dva z výše uvedených aspektů systémové teorie: aspekt struktury a aspekt chování. Chtějí-li modely postihnout ty to dva aspekty, měly by ke zkoumání předmětu svého zájmu využívat dva pohledy:

- **Informační pohled:** „...je strukturovanou kompilací, popisem a reprezentací systémové informace. Tato informace popisuje systémové elementy, jejich strukturu, vazby, informace podporující spouštění funkcí a informace popisující výsledky funkcí“ [ISO01]

- **Funkční pohled:** „... je popisem a reprezentací aktivit a procesů v podniku. Funkční pohled popisuje zpracování elementů a organizování jednotlivých procesních kroků do struktur komplexních procesů, s přihlédnutím k jejich logickým spojením a závislostem. Zaměřuje se na popis chování systému, vzájemných závislostí a vlivů mezi elementy během vykonávání funkcí v podniku.“[ISO01]

## 2.3 Shrnutí

Standard neuvádí jakým konkrétním způsobem mají být pohledy transformovány do modelů nebo jaké konkrétní informace mají tyto pohledy obsahovat. Informační a funkční pohled jsou standardem vyžadovány jako minimální množství pohledů, které se mají při modelování použít. To jakým způsobem mají být konstruovány konkrétní modely, jak tyto modely mají vypadat, jaké pohledy se mají při dekompozici systému používat, a na jaké části systémů se mají pohledy soustředit, pak záleží na účelu modelování, na podmínkách v modelovaném podniku a na používané metodologii modelování. K modelování podnikových procesů by pak měl sloužit především pohled funkční. Tímto způsobem ISO 14258 vytváří sadu principů, ze kterých by se mělo vycházet při modelování a při vytváření metodologií a standardů pro modelování.

### **3 Standardy CEN ENV 40003:1990 a CEN ENV 12204:1995**

V roce 1990 organizace CEN/CENELEC WG-ARC, která je součástí Evropské standardizační komise CEN (European committee for Standardization), vytvořila evropský standard ENV 40003 - Architektonický rámec pro modelování CIM systémů (ENV 40003 – CIM systems architecture framework for modeling). V tomto standardu jsou definovány nezbytné požadavky a pojmy pro modelování podnikových procesů. Základní snahou ENV 40003 je vytvořit podmínky pro jednotné identifikování a definování podnikových procesů, pojmů a koncepcí a jejich sdílení s podnikovými partnery. Disciplína procesního modelování se jako všechny informační disciplíny velice rychle rozvíjí, proto se postupem času ukázalo, že pojmy, metody a koncepce definované ENV 40003 nejsou schopny postihnout celou problematiku podnikových procesů a proto bylo standard nutné rozšířit. Rozšířením ENV 40003 je standard CEN ENV 12204 – Konstrukty architektury systémů pokročilé výrobní technologie pro podnikové modelování z roku 1995 (CEN ENV 12204 Advanced Manufacturing Technology Systems Architecture Constructs for Enterprise Modelling), který je produktem skupiny CEN TC310 WG.

#### **3.1 Rámec pro podnikové modelování ENV 40003**

Standard jako takový je vytvořen jako univerzální koncepce pro modelování různých aspektů činnosti podniků. Slovo univerzální v předchozí větě znamená, že tato koncepce by měla postihovat výrobní podniky bez ohledu na jejich odlišnost v potřebách a vizích, bez ohledu na rozdílnost produkčních oblastí, jejich specifických dodavatelů, používaných podnikových systémů či úrovní poskytovaných služeb, přesto by však měla poskytovat dostatečné nástroje pro vyjádření podnikových specifik. Proto byly pojmy konstruovány s takovou obecností, aby tuto univerzálnost mohly zajistit. Modelem se zde pak rozumí, skupina různých modelů, které modelují důležité aspekty podniku, a které jsou vzájemně provázány.

„Ze všech možných dimenzí modelování si rámec zvolil tři pro jejich schopnost reprezentovat potřebné pojmy:

- první dimenze je zaměřena na vývoj a rozvoj podnikového modelu samotného, od počáteční specifikace požadavků až po zpracovatelný model; tato dimenze je nazývána dimenzí modelu podniku
- druhá dimenze se zabývá strukturou a chováním modelu za účelem vyjádření klíčových aspektů podniku; toto je dimenze pohledu
- třetí dimenze se zaměřuje na specializaci modelů a modelových komponent z obecného na konkrétní; nazývána je dimenzí obecnosti“ [ENV01]

### 3.1.1 Pohledy na strukturu podniku

Samotným jádrem popisu podniku v ENV 40003 je konstituce čtyř základních pohledů, kterým odpovídají základní modely:

- Funkce (Function View): „Funkční pohled zahrnuje chování podniku (procesy a události), podnikovou funkcionalitu (např. funkce nebo podnikové aktivity) a čas (dynamické aspekty). Také popisuje, jak jsou funkce a procesy propojeny, a jaké jsou jejich požadavky na informace, zdroje a organizační entity, které za ně odpovídají.“ [ENV01]
- Informace (Information View): „Podnikové objekty, jejich vztahy a stavy jsou definovány v informačním pohledu.“ [ENV01]
- Zdroje (Resource View): „Pohled, který znázorňuje funkční entity jako aktivní zdroje, jejich schopnosti a role.“ [ENV01] – reprezentace lidských, materiálních a informačních zdrojů
- Organizace (Organization View): „Organizační pohled (Organization View), organizuje funkční entity, podnikové objekty a aktivity od organizačních jednotek, které mají odpovědnost a autoritu“ [ENV01]

Standard ENV 40003 se ukázal nedostatečným ve chvíli, kdy se projevilo, že jeho obecnost brání použití pro některé nestandardní aspekty činností podniků, a především počítačové podpoře modelování, neboť „...požadované specifické vlastnosti a chování je třeba výlučně vyjadřovat v takové formě, která je způsobilá k počítačovému zpracování, a v takovém grafickém podání, které napomáhá porozumění člověkem.“ [ENV01] Proto

bylo nutné standard o prostředky, které výše uvedené problémy pomáhají odstranit, rozšířit.

## 3.2 CEN ENV 12204

Standard CEN ENV 12204 vychází ze základů ustanovených v ENV 40003, zejména z jeho čtyřech základních pohledů (funkční, informační, zdrojů, organizační). Na podnik však pohlíží především jako na:

- „otevřenou skupinu společně působících business procesů, které směřují k naplnění podnikových úkolů a cílů
- společenství funkčních entit (strojů, lidí, aplikací), které v průběhu business procesu zpracovávají podnikové objekty“ [ENV01] (zdroje, data, informace)

Při modelování business procesů je třeba přihlížet k tomu, že tyto procesy jsou sice napříč obory podobné, ale každý podnik má své specifické vlastnosti a problémy. Je proto nutné při vytváření modelů zajistit dostatečnou abstrakci, která by však měla být transformovatelná do podoby, jenž umožňuje podmínky jednotlivých podniků vyjádřit. Takto vytvořené modely by pak měly být snadno použitelné a především počítačově zpracovatelné (tedy standardizované). Tyto problémy ENV 12204 řeší vytvářením konstruktů.

### 3.2.1 Konstrukty

CEN ENV 12204 používá konstrukty jako základní prostředek standardizace. Konstrukt lze chápat jako „textový či grafický artefakt vytvořený za účelem jednotné reprezentace různorodých informací o vlastnostech a elementech určitého souboru jevů.“ [ENV01]. Jedná se tedy o konstrukci, která vystihuje základní vlastnosti určité skupiny obdobných jevů, přičemž abstrahuje od jejich konkrétních výskytů. Konstrukty představují jakési šablony jevů se, kterými se lze při vytváření modelů setkat, přičemž to, jaký fenomén bude do této šablony dosazen, záleží až na konkrétní aplikační oblasti a situaci, kterou je třeba modelovat. Jednotlivé realizace konstruktů, které reprezentují určitou třídu jevů reálného světa pak ENV 12204 nazývá **stavebními bloky**. Z těchto bloků lze pak vytvořit komplexní procesní model.

Základními konstrukty, které ENV 12204 vymezuje jsou:

- Podnikový objekt (Enterprise object)
- Objektové View (Object View)
- Stav objektu (Object State)
- Produkt
- Instrukce (Order)
- Organizační jednotka
- Zdroj
- Činnost (Enterprise activity)
- Business proces
- Vztah (Relation)
- Množina schopností (Capability Set)
- Událost

### **Podnikový objekt**

„Modelovací entita, která popisuje zobecněný, reálný nebo abstraktní, unikátně se vyskytující jev reálného prostředí výrobního podniku (tj. věc, která může být popsána nebo o ní může být myšleno jako o celku). Podnikové objekty jsou charakterizovány svou unikátní identitou a popisnými vlastnostmi.“[ENV01]

Můžeme rozeznávat různé druhy Podnikových objektů, např. podle toho jakými procesy jsou determinovány, a mohou hrát různé role, mohou být např. produkty, požadavky, surovinami nebo organizačními jednotkami. Jsou používány pro popis strukturovaných množin podnikových entit (např. produkty, formuláře, výkazy, atd.) nebo nestrukturované věci (např. neformátované dokumenty, signály, verbální příkazy, atd.) Při reálném průběhu procesu jsou pak reálné podnikové entity reprezentovány výskytem stavebního bloku Podnikový objekt.

## **Objektové view**

„Průmět jednoho nebo více Podnikových objektů skrze definovanou podmnožinu jejich atributů, kde podmnožina je definována výběrem atributů nebo jejich intervaly.“ [ENV01]

Objektové view filtruje specifikované atributy Podnikového objektu a ostatní nebere v potaz. Atributy ve view mohou být také kombinacemi atributů Podnikových objektů, nad kterými je view definováno. Tento konstrukt poskytuje možnost různým uživatelům nahlížet na Podnikové objekty z různých úhlů, což umožňuje zjednodušit modelování.

## **Stav objektu**

„Stav objektu odkazuje na současný stav Podnikového objektu v daném okamžiku jeho životního cyklu.“ [ENV01]

Reprezentuje pohled na Podnikový objekt v určitém okamžiku a je definován hodnotami atributů objektu v daném okamžiku. Umožňuje sledovat různé stavy objektu při průchodu procesem.

## **Produkt**

„Konstrukt reprezentující požadovaný výstup nebo vedlejší produkt výrobního procesu podniku, poskytující všechny potřebné informace pro výrobu nebo charakteristiku tohoto výstupu.“ [ENV01]

Tento konstrukt popisuje produkty jejichž výroba a prodej je cílem podniku. Specializace tohoto konstruktů pak představují jednotlivá stadia vzniku produktu.

## **Instrukce**

„Tento konstrukt představuje nezbytný vstup business procesu, který koordinuje a řídí jiné business procesy nebo Podnikové aktivity.“ [ENV01]

Instrukce znázorňují informace o plánování a řízení business procesů v podniku: co musí být vykonáno, jaké objekty se mají produkovat, jaké se mají využívat zdroje a kdo je odpovědný za spuštění procesu. Jsou příkazem k vykonání činnosti jedné autority v podniku jiné autoritě.

## **Organizační jednotka**

„Konstrukt, který reprezentuje část organizační struktury podniku, a který je popsán atributy, které mohou být vlastnostmi organizace nebo odkazy na nižší úroveň Organizačních jednotek.“ [ENV01]

Tento konstrukt je používán pro popsání podniku a jeho organizačních částí. Jeho hlavním smyslem je umožnit popsat způsob, jakým jsou v podniku spravovány lidské zdroje a jejich schopnosti. Modely organizační struktury by měly popsat odpovědnosti jednotlivců, týmy a buňky v organizaci a jejich vztahy, komunikační kanály, atd.

## **Zdroj**

„Zdroj je konstruktem, který představuje některé nebo všechny kapacity potřebné pro činnost nebo business proces.“ [ENV01]

Konstrukty Zdrojů popisují takové jevy v podniku, které

- mají schopnost uskutečňovat určité funkce nebo
- podporují uskutečňování nějakých funkcí nebo
- jsou nezbytné pro provádění nějakých funkcí

Konstrukt a jeho specializace popisují veškerou materiálovou a informační podporu v podniku, která je potřebná k provádění podnikových činností (např. strojové vybavení, nástroje, lidé a skupiny lidí, dokumenty, vybavení pro zpracování dat, atd.)

## **Činnost**

„Činnost je konstrukt představující určitou část podnikové funkcionality. Popisuje (žádoucí) přeměnu věcí v podniku, součástí popisu jsou: cíl Činnosti, informace nezbytné k řízení přeměny a funkční kapacity (množiny schopností), které jsou zdrojem těchto změn buď požadovány nebo poskytovány.“ [ENV01]

Činnost představuje akci, která řeší určitou úlohu v podniku. Tato akce pro své provedení vyžaduje určitý čas a zdroje (vstupy), které jsou transformovány do podoby určitého výstupu.

## **Business proces**

„Částečně uspořádaná množina Činností, která může být spuštěna za účelem dosažení určitého cíle podniku nebo části podniku.“ [ENV01]

Business proces představuje určitou část chování podniku, skládající se z různých funkcí a jejich vazeb., tzn. jde o strukturu popisující množinu Činností, jejich logickou posloupnost a vzájemnou závislost. Výsledkem Business procesu je vždy pozorovatelný nebo kvantifikovatelný materiální nebo informační objekt. Smyslem Business procesu je vytvoření požadovaného výstupu – přeměnu počátečních stavů v požadované konečné stavy.

ENV 12204 rozeznává tři základní druhy Business procesů:

- dobře strukturované procesy – je znám požadovaný konečný stav a posloupnost Činností
- středně strukturované procesy – je znám požadovaný konečný stav, ale posloupnost Činností bude známa až při průběhu procesu
- nestrukturované procesy – procesy u nichž není znám ani požadovaný výsledek ani posloupnost Činností

## **Vztah**

„Reprezentace uživatelem definovaného vztahu mezi dvěma stavebními bloky (viz výše).“ [ENV01]

Konstrukt Vztah propojuje dva nebo více konstruktů, jejichž výskyty se během průběhu Business procesu mohou dostat do vztahu. Jeho smyslem je modelovat dynamické a jiné vazby mezi modelovanými entitami.

## **Množina schopností**

„Množina charakteristik konstruktů Zdroj nebo Činnost, která popisuje schopnost určitého fenoménu být poskytován zdrojem nebo vyžadován Činností.“ [ENV01]

Množina schopností Podnikového objektu je sadou atributů, které jsou relevantní pro rozhodnutí, zda je objekt vhodný pro určitou činnost. Množinou schopností mohou být např. technické parametry výrobního zařízení.

## Událost

„Faktická změna v podniku, která indikuje, že se stalo něco, co má pro podnik význam.“ [ENV01]

Událost je zpravidla nositelem informace a je používána ke spouštění jednoho nebo více procesů. Událost může být jednorázová (např. pokles zásoby) nebo soustavná (např. chlazení).

Standard u každého konstruktů kromě definice a popisu uvádí prostředek formalizovaného zápisu konstruktů. Jedná se o šablonu díky níž lze jednotně, avšak s dostatečnou detailností, definovat daný konstrukt. Konstrukt, jehož šablona je naplněna patřičnými údaji, se pak stává oním stavebním blokem, který již stačí zasadit do konstrukce modelu. Pro názornost zde uvedu šablonu pro definici výskytu konstruktů Vztah, tak jak jej uvádí ENV 12204:

### Header Part

Identifier:	Name and/or number < <b>m</b> >
Class Name:	Relation < <b>m</b> >
Sub-class:	NIL < <b>m</b> >
Functional description:	Clear text < <b>o</b> >
Design Authority:	Organisational Unit Identifier < <b>o</b> >

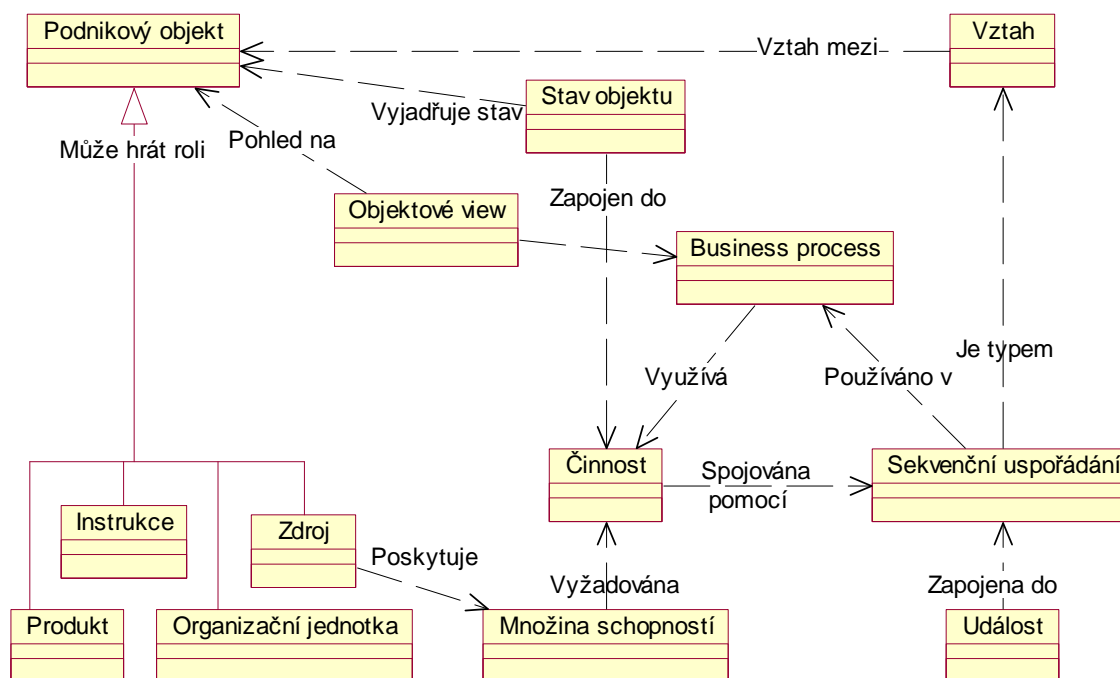
### Body Part

Originator:	Enterprise Object Identifier or Enterprise Activity-identifier or Business Process Identifier < <b>m</b> >
Related to:	Enterprise Object Identifier or Enterprise Activity-identifier or Business Process Identifier < <b>m</b> >
Cardinality:	[1:1 or 1:n or n:1 or m:n] < <b>m</b> >
Pre-defined or User-defined attributes:	List of {Enterprise Object attribute(s) and/or Enterprise Object Identifier(s)} < <b>o</b> >

Šablona konstruktů se skládá ze dvou základních částí, z hlavičky a těla. Hlavička je stejná pro všechny typy konstruktů, zatímco tělo šablony je pro každý typ specifické. V hlavičce jsou uvedeny základní identifikační atributy konstruktů, jako je jméno či název třídy. V těle jsou pak uvedeny atributy, které definují samotný obsah konstruktů.

Poznámkou <m> jsou označeny povinné atributy, poznámka <o> pak označuje atributy volitelné.

Specifikace CEN ENV 12204 nepodává pouze výčet základních elementů, které by se při modelování měly používat, ale lze z ní vyvodit také jejich základní logické vztahy. Tyto vztahy jsou zachyceny na následujícím schématu (Obr. 3.1).



Obr. 3.1 – Konstrukty dle CEN ENV 12204 [CHEN01]

### 3.3 Shrnutí

Standardy CEN ENV 40003:1990 a CEN ENV 12204:1995 jsou základními standardizačními iniciativami. První standard tvoří širší rámec pro podnikové modelování s definicí základních úhlů pohledu. Druhý standard je pak konkretizací, která má za cíl určit, jak mají být pohledy transformovány do modelů, a z čeho se tyto modely mají skládat. Zejména výše popisované konstrukty představují velmi dobrý nástroj pro reprezentaci a formalizování modelů business procesů. Konstrukty jsou obecnými šablonami, které je nutné pro dané případy specializovat a organizovat do struktur dle

míry abstrakce. Jsou dostatečně jednoduché a srozumitelné, aby mohly být snadno sdíleny mezi uživateli a tvůrci modelů. Formalismus s nímž jsou konstrukty definovány z nich pak činí elementy modelovacího jazyka, který umožňuje počítačové zpracování, bez kterého se, při složitosti v současné době řešených úloh, není možné obejít.

## **B. De facto standardy modelovacích jazyků**

V tomto oddíle představím některé standardy specializovaných nástrojů či metod pro modelování podnikových procesů. Tyto standardy jsou velice podrobnou specifikací toho, jak tvořit modely podnikových procesů. Nezabývají se již obecnými principy modelování, ale převádějí tyto principy do podoby, která je vhodná pro nasazení v konkrétních situacích. IDEF, BPML a BPMN jsem zvolil zejména pro jejich komplexnost a perspektivu dalšího budoucího vývoje. UML pak především pro jeho značnou rozšířenost a modularitu.

### **4 Metodologie IDEF**

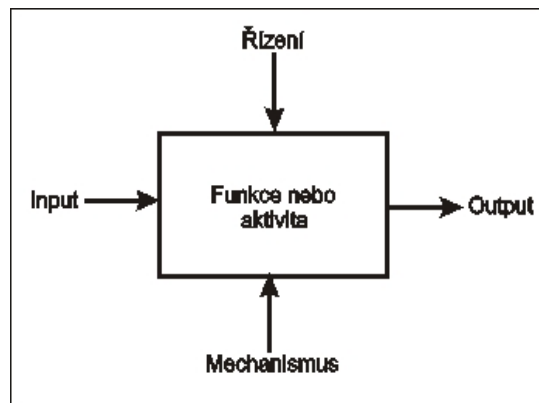
IDEF – The Integrated DEFinition metodologie je souborem či rodinou metod, jejíž účelem je komplexně podporovat potřeby podniků v oblasti modelování podnikové architektury. Metodika je produktem výzkumného programu ICAM (Integrated Computer-Aided Manufacturing program), který byl vytvořen letectvem USA za účelem zlepšení koordinace výrobních operací. V současné době je v rámci IDEF rozvinuto šest metod (IDEF0, IDEF1, IDEF1x, IDEF3, IDEF4, IDEF5) na takové úrovni, aby mohly být použity v praxi. Další osm dalších metod (IDEF6 až IDEF14) je pak ve vývoji. Přehled všech součástí metodiky je uveden v Tab. 4.1. Každá z metod je uceleným a rozsáhlým souborem nástrojů, jejichž popis přesahuje rámec této práce. Proto u metod, jejichž hlavním účelem není popis business procesů uvedu pouze jejich stručnou charakteristiku, a hlavní pozornost budu věnovat metodě IDEF3, v níž je obsaženo těžiště popisu procesů metodologie IDEF.

Přehled součástí metodologie IDEF	
IDEF0	Function Modeling
IDEF1	Information Modeling
IDEF1X	Data Modeling
IDEF2	Simulation Model Design
IDEF3	Process Description Capture
IDEF4	Object-Oriented Design
IDEF5	Ontology Description Capture
IDEF6	Design Rationale Capture
IDEF8	User Interface Modeling
IDEF9	Scenario-Driven IS Design
IDEF10	Implementation Architecture Modeling
IDEF11	Information Artifact Modeling
IDEF12	Organization Modeling
IDEF13	Three Schema Mapping Design
IDEF14	Network Design

Tab. 4.1 – přehled součástí metodologie IDEF

### **IDEF0 – Metoda pro modelování funkcí**

„IDEF0 je metoda navržená pro modelování rozhodování, akcí a aktivit v organizaci nebo systému.“ [IDEF02] Tvorba modelů vychází z grafického jazyka SADT (Structured Analysis and Design Technique). Metoda IDEF0 se používá ke specifikaci funkčních modelů podniku, které slouží k určení základních činností podniku. V modelech se znázorňují hlavní aktivity a jejich vstupy, výstupy, řídicí vstupy a mechanismus spojený s každou hlavní aktivitou. Jednotka složená z těchto komponent je pak základním stavebním kamenem funkčních modelů a je nazývána ICOM (Input, Control, Output, Mechanism)(viz. Obr. 4.1). Na této úrovni modelování ICOM představuje podnikový proces, který je v nižších úrovních modelů specifikován pomocí jazyka IDEF3.



Obr. 4.1- element ICOM [IDEF02]

### **IDEF1- Metoda pro modelování informací**

IDEF1 slouží pro modelování informací v podniku. Jedná se o metodu, která popisuje, komunikuje a analyzuje podnikové informační potřeby. Modely identifikují pojmy, které jsou v podniku používány a vztahy mezi nimi (např. pojem materiál, pojem sklad materiálu a vztah materiál se nachází ve skladu materiálu). IDEF1 není metodou pro navrhování databází, ale umožňuje organizaci porozumět s jakými informacemi pracuje. Organizace pak může lépe identifikovat a spravovat informační zdroje.

### **IDEF1x – Metoda modelování dat**

IDEF1X je metodou navrhování relačních databází. Jejím smyslem je vytvoření logického obrazu podnikových dat. Jak sami autoři uvádějí, její použití není příliš vhodné u systémů, které nejsou relačně založené, např. u objektově orientovaných systémů.

### **IDEF4 – Objektově orientovaná metoda návrhu**

IDEF4 byla vytvořena za účelem podpory objektově orientovaného navrhování aplikací.

### **IDEF5 – Metoda tvorby ontologií**

„Ontologií je doménový slovník spolu se sadou precizních definic nebo axiomů, které vymezují význam termínů natolik, aby to umožňovalo jednotnou interpretaci dat.“[IDEF03] Metoda IDEF5 je pak propracovanou technikou, která tvorbu ontologií umožňuje.

## 4.1 IDEF3 – Metoda popisu procesů

IDEF3 byl vytvořen proto, aby bylo možné v rámci IDEF metodologie popisovat chování systému. Jeho základním cílem je „poskytnout strukturovanou metodu díky níž by expert mohl vyjádřit znalosti o činnosti určitého systému nebo organizace“ [IDEF01] Metoda poskytuje jak způsoby sběru informací o procesech systému (podniku), tak i způsoby, jak získané znalosti vhodně reprezentovat a komunikovat. Pro reprezentaci znalostí využívá metoda vlastní grafický modelovací jazyk.

Základním elementem, ze kterého se při tvorbě modelů vychází je tzv. scénář. Scénář je termín, který metoda používá pro „...základní organizační strukturu pro model procesů. Scénářem je opakující se situace nebo množina situací, která popisuje typickou třídu problémů, nebo představuje uspořádání z něhož vyvstává proces.“ [IDEF01] . Jedná se tedy o takové projevy činnosti podniku z nichž se modelují procesy. Zdrojem scénářů jsou interview nebo pozorování. Ze scénářů se následně vytvářejí modely procesů.

Metoda využívá k popisu procesů dvě základní strategie:

- strategie zaměřená na procesy
- strategie zaměřena na objekty

Tyto strategie představují dva základní pohledy na systém. První pohled, zaměřený na procesy, se soustředí na zkoumání chování sledovaného systému. Druhý pohled, který sleduje objekty, pak zkoumá obsah systému a jeho podstatu. Strategie působí při vytváření modelů procesů tak, aby jejich spojením vznikl ucelený obraz. Pracují paralelně a využívají odlišné prostředky modelovacího jazyka, ačkoliv se někdy tyto prostředky mohou překrývat. Hlavní výrazové prostředky modelovacího jazyka IDEF3 představím v následujícím textu.

### 4.1.1 Strategie zaměřená na procesy

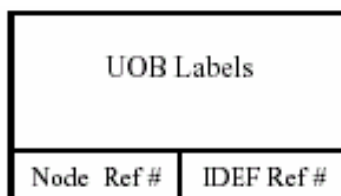
Strategie zaměřená na procesy organizuje znalosti s důrazem na procesy a jejich časové, kausální a logické vztahy uvnitř scénáře. Znalost je reprezentována s použitím grafického modelovacího jazyka v procesně orientovaných modelech procesů.

Základními stavebními kameny jazyka pro vytváření procesně orientovaných modelů jsou:

- UOB – Unit Of Behavior
- Vazby
- Uzly

### Unit of Behavior

UOB představuje obecný typ situace, který se v systému vyskytuje. Takovou situací může být například „výdej materiálu“. UOB však nepředstavuje konkrétní výskyt situace „výdej materiálu“, ale je její abstrakcí, tedy představuje třídu situací „výdej materiálu“. Při průběhu procesu jsou pak vytvářeny konkrétní instance (výskyty) UOB. UOB je v modelech reprezentována symbolem, který se nazývá UOB box (viz. Obr. 4.2), a je definována svým názvem (UOB Labels) a číslem (*Node ref#*), které umožňuje propojovat modely s různou úrovní abstrakce. Příklad takového propojení je vidět na ukázkovém procesu v příloze č. 1.



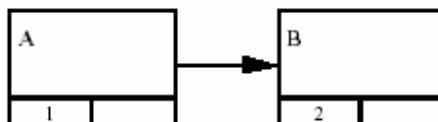
Obr. 4.2 – UOB box

### Vazby

Vazby představují pojivo mezi UOB boxy, které má znázornit dynamickou podstatu procesu a upozornit na vzájemné vztahy jednotlivých UOB. Notace modelovacího jazyka IDEF3 používá sedm symbolů pro vyjádření různých druhů vazeb.

**Jednoduché vazby** „vyjadřují vztah časové následnosti mezi instancí jedné UOB a jiné UOB.“ [IDEF01] Jedná se o nejčastěji používanou vazbu, která je symbolizována jednoduchou šipkou (viz. Obr. 4.3) a znamená, že legitimními aktivacemi schématu je množina všech kombinací výskytů A a B, kromě situace, kdy jsou aktivovány obě UOB a

B je aktivováno dříve než A. Tzn. množina situací: A samostatně, A následováno B, B samostatně. UOB od níž je šipka vedena se nazývá výchozí UOB, UOB k níž šipka směřuje se nazývá cílová UOB.



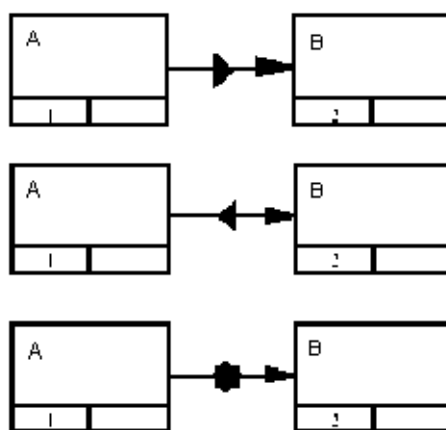
Obr. 4.3 – Jednoduchá vazba

Logika jednoduchých vazeb je taková, že neříká nic o tom, zda v případě, že nastane situace A, musí nastat také situace B. Vyjadřuje pouze, že B nesmí nastat před A. Pro vyjádření povinností obsažených ve vztazích se používají tzv. **omezující vazby**, které představují omezení vazeb jednoduchých.

První z omezujících vazeb vyjadřuje povinnost, aby výskyt výchozí UOB (A) byl následován výskytem cílové UOB (B). Výskyt B však není výskytem A podmíněn. To znamená, že jestliže nastane A pak sice musí nastat i B, ale B může nastat i samostatně (což již samozřejmě neplatí pro A). Tato vazba tedy působí donucení zleva doprava. Symbolický zápis takového vztahu je uveden na Obr. 4.4 nahoře.

Další situací, kterou může být nutné vyjádřit je nezbytnost, aby výskyt B následoval pouze po výskytu A. Jedná se tedy o omezení jednoduché vazby tak, aby se B nemohlo vyskytovat samostatně, přičemž A se samostatně vyskytnout může. Takové chování se znázorňuje symbolem uvedeným na Obr. 4.4 uprostřed.

Symbol zobrazený na Obr. 4.4 dole je kombinací předchozích dvou vazeb, a používá se v situacích, kdy A musí být následováno B, a zároveň B se nesmí vyskytnout bez předcházejícího uskutečnění A.



Obr. 4.4 – Omezující vazby

Výše uvedené možnosti chování a jim odpovídající symboly však nemusí vystihovat některé typy chování, se kterými je možné se setkat. Takovou situací může být například požadavek vyjádřit povinnost, aby se cílová UOB uskutečnila v určitém časovém intervalu po výskytu výchozí UOB. Takových typů požadavků může být velké množství, proto jazyk zavádí obecnou omezující vazbu, jejíž vlastnosti si uživatel definuje sám. Dalším symbolem, který umožňuje uživateli definovat význam vztahu je přerušovaná čára spojující dvě UOB. Symboly pro obecné vazby jsou uvedeny na Obr. 4.5.



Obr. 4.5 – Obecné vazby

## Uzly

„Uzly v IDEF3 poskytují mechanismus pro specifikování logiky větvení procesů. Kromě toho uzly zjednodušují znázornění větvení vztahů mezi možné průběhy procesu.“ [IDEF01] Uzel tedy představuje takové místo, kde se průběh procesu větví do více podprocesů, nebo naopak místo, v němž se podprocesy slučují. IDEF3 rozeznává čtyři základní typy uzlů:

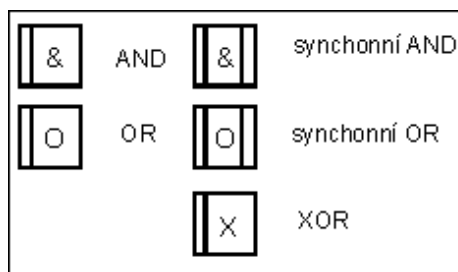
- „uzly v nichž se proces větví do více paralelních podprocesů
- uzly v nichž se více paralelních podprocesů spojuje do jednoho procesu
- uzly v nichž se proces větví do více alternativních podprocesů
- uzly v nichž se více alternativních podprocesů spojuje do jednoho procesu“ [IDEF01]

První dva typy uzlů představují konjunkční uzly či AND uzly, které se značí symbolem „&“. Druhé dva typy uzlů jsou uzly disjunkčními, označovanými písmenem „O“. Lze rozeznávat dva druhy disjunkčních uzlů:

- OR uzlem je takový uzel, do něhož vstupují či z něj vystupují takové subprocesy, které se navzájem nevylučují a mohou působit společně. Označuje se písmenem „O“
- XOR uzlem je pak takový uzel, do kterého vstupují či z něj vystupují vzájemně se vylučující subprocesy. Označuje se písmenem „X“.

Metodika dále rozlišuje zda je uzel synchronní či asynchronní:

- Asynchronní AND uzel vyjadřuje nutnost vykonání všech cílových UOB, které za ním následují, bez ohledu na pořadí či čas jejich spuštění
- Synchronní AND uzel také vyjadřuje nutnost vykonání všech cílových UOB, ale všechny musí být spuštěny najednou (pokud se jedná o uzel, do kterého paralelní UOB vstupují, pak musí všechny najednou skončit)
- Po asynchronním OR se aktivuje jeden nebo více UOB bez ohledu na pořadí či čas
- Po synchronním OR se UOB, které se mají aktivovat, spustí všechny najednou (nebo musí skončit najednou, pokud do uzlu vstupují)
- U XOR uzlu je synchronnost či asynchronnost irelevantní



Obr. 4.6 – Symboly pro uzly

Díky tomu, že metoda nepoužívá pouze jednoduché (asynchronní) AND a OR, poskytuje širokou základnu pro precizní modely business procesů. Způsoby použití jednotlivých symbolů a jejich kombinací přesahuje rámec této práce, proto bych zájemce o bližší informace odkázal na dokument [IDEF01].

#### 4.1.2 Strategie zaměřená na objekty

Strategie zaměřená na objekty používá specifickou notaci pro vytváření diagramů, které poskytují uživateli informaci o objektech účastnících se business procesů. Tato informace představuje údaje o tom, „jak jsou různé druhy objektů procesem transformovány na jiné druhy objektů, nebo jak je objektům určitého druhu procesem změněn stav.“ [IDEF01] Pohled zaměřený na objekty se tedy snaží popsat druhy věcí, které vstupují do procesů, a jejich stavy. Svou podstatou odpovídá filozofii informačního pohledu, tak jak jej koncipuje standard ENV 40003. Pro objektově orientovaný popis procesů vytváří IDEF3 vlastní symboliku, jejíž popis následuje. (příklad objektově orientovaného modelu je uveden v příloze č. 4)

#### Objekty a stavy objektů

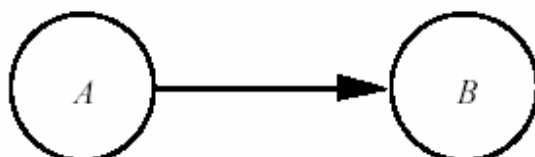
Určitý druh objektu, kterým může být například součástka výrobku, je reprezentován jednoduchým kruhem, který obsahuje název objektu. Tento symbol se nazývá „symbol druhu“. Rozšířením tohoto symbolu je „symbol stavu objektu“, ve kterém je uveden i momentální stav objektu, a který se používá, pokud objekt při zpracování procesy nabývá různých stavů. Příklady těchto symbolů jsou uvedeny na Obr. 4.7, přičemž první symbol zleva je symbolem druhu, ostatní jsou symboly stavu objektu. Tyto symboly jsou základními stavebními kameny diagramů objektů.



Obr. 4.7 – Symboly druhu a stavu objektu

## Vyjádření změn stavů objektů

Smyslem modelů orientovaných na objekty je především vyjádřit, jakým způsobem se objekty při běhu procesů přeměňují. Přejít objektu z určitého stavu do jiného stavu je v diagramech vyjadřován šipkou vedoucí od jednoho symbolu stavu objektu k jinému symbolu stavu objektu (Obr. 4.8).

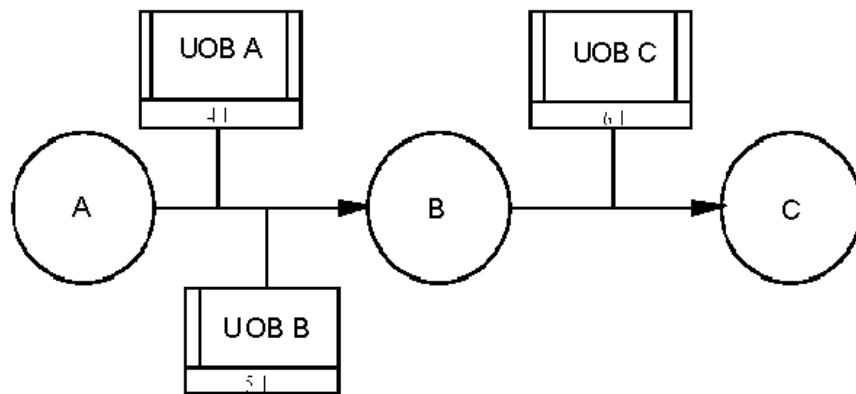


Obr 4.8. – Změna stavu objektu

Výše uvedeny symbol představuje událost nebo množinu událostí, které způsobí přeměnu objektu A v objekt B. V průběhu těchto událostí je objekt A modifikován, transformován nebo spotřebován takovým způsobem, aby byl vytvořen objekt B, přičemž objekt B může být pouze jiným stavem objektu A (např. voda ve formě ledu je přeměněna na vodu v kapalném stavu).

## Reference

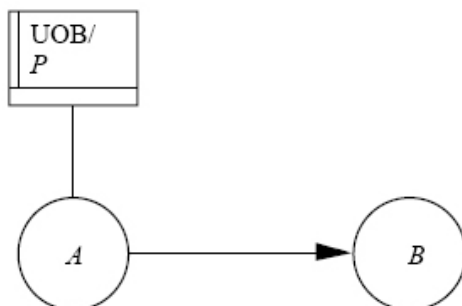
Změna stavu objektu je zpravidla uskutečňována nějakým procesem. Proto je někdy užitečné odkázat na tento proces. Odkazy se provádějí pomocí referenčních symbolů (viz. Obr. 4.9), které umožňují provázat objektově zaměřené diagramy mezi sebou nebo, což je zvláště užitečné, s procesně zaměřenými diagramy. Příklad na Obr. 4.9 znázorňuje situaci, kdy změna objektu A v B je zprostředkována činnostmi UOB A a UOB B, a změna objektu B je zprostředkována činností UOB C. Jednotlivé UOB představují činnosti nebo procesy specifikované v procesních diagramech. Činnosti jsou spouštěny v tom pořadí v jakém jsou zakresleny ve směru od výchozího k cílovému objektu.



Obr. 4.9 – Referenční symboly

Referenční symboly také umožňují specifikovat způsob vykonávání jednotlivých činností při transformaci objektů. Na Obr. 4.9 je vidět, že se symboly referencí na UOB A a C liší od reference na UOB B. Reference na UOB A a C jsou totiž tzv. „call-and-wait“ referencemi. To znamená, že aby mohla být spuštěna UOB B, musí být nejprve dokončena UOB A. Protože však je reference na UOB B obyčejná, může být transformace objektu A v B dokončena dříve než je skončena činnost UOB B. Taktéž platí, že transformace B v C nemůže být skončena dříve, než je dokončena činnost UOB C.

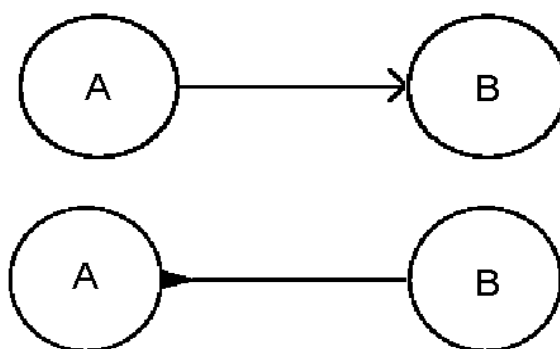
Není neobvyklé, že specifický proces či aktivita využívá objekt v určitém stavu a v tomto stavu je objekt po celou dobu běhu procesu udržován. Tuto skutečnost lze opět vyjádřit užitím reference, tentokrát přímým napojením na symbol druhu objektu. (Obr. 4.10).



Obr. 10 – Referenční symboly

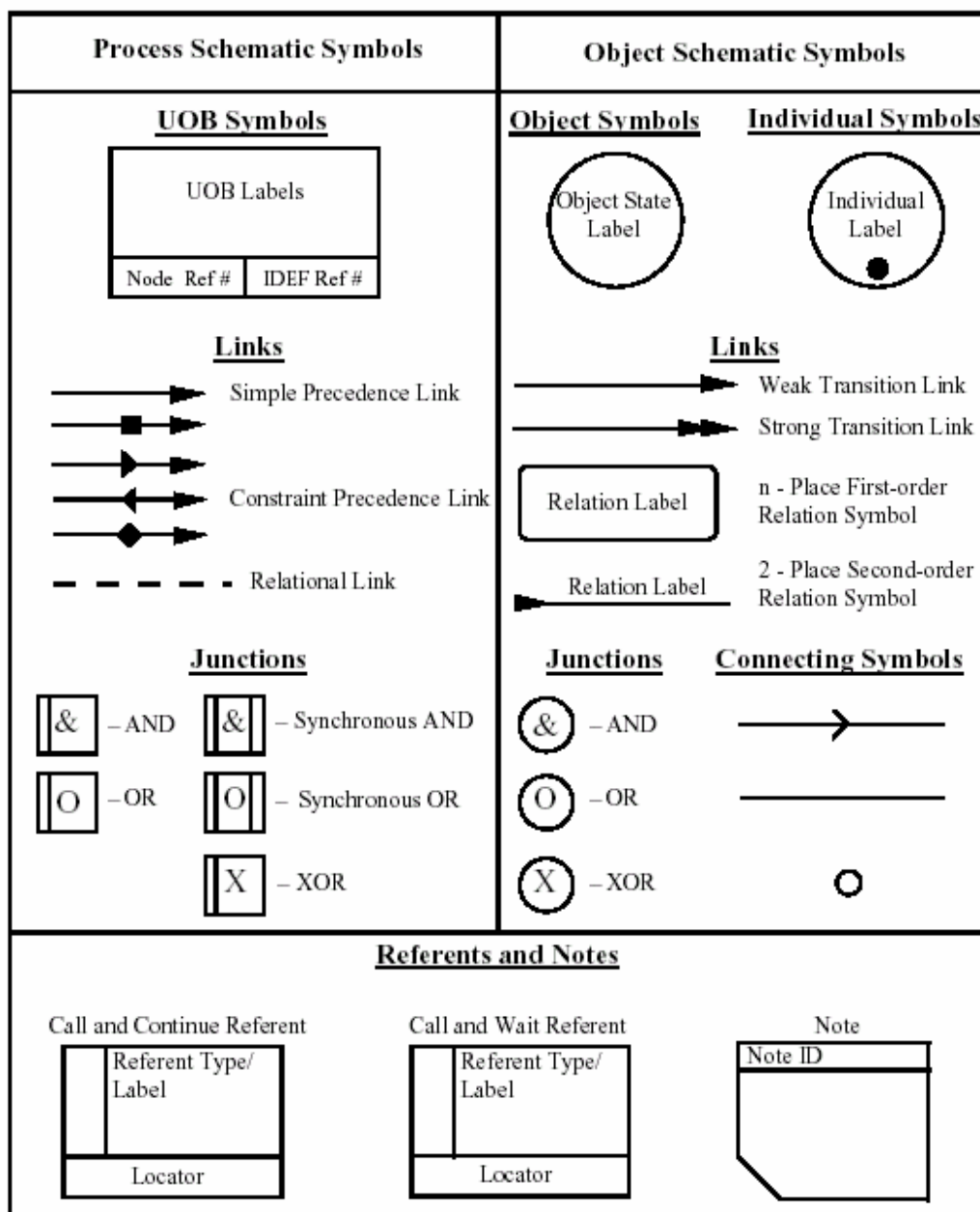
## Klasifikace objektů

Při modelování je někdy nutné vyjádřit to, že se objekt skládá z více částí, nebo že objekt představuje třídu objektů, do které spadají různé objekty. Obrázek č. 4.11 představuje symboly, které se pro tyto vztahy používají. Nahoře je znázorněn symbol který vyjadřuje vztah „A je částí B“ – kompoziční vztah. Dole je pak symbol označující vztah „ A je podtřídou třídy objektů B“ – klasifikační vztah. Ukázky kompozičního a klasifikačního diagramu jsou uvedeny v přílohách 3 a 4.



Obr. 4.11 – Kompoziční a klasifikační vztah

IDEF3 je díky své notaci poměrně flexibilní metodou, která dokáže věrně popsat modelované situace (celkový přehled symbolů zachycuje obrázek č. 4.12). Avšak porovnáme-li IDEF3 například s notací BPMN, je jasné, že, vzhledem k výrazové komplexnosti BPMN, je vhodná k modelování spíše na vyšší úrovni abstrakce, a její hlavní výhodu lze spatřovat především ve vazbě na ostatní metody IDEF a jejich společném uplatnění. Součástí specifikace IDEF3 je také popis metody pro řízení projektu analýzy a deskripce podnikových procesů. Poskytuje návrh základních řídicích dokumentů projektu a návod, jak tvořit procesní i objektová schémata, jakým způsobem dosáhnout jejich potřebné kvality, a jak zajistit jejich provázanost. V této práci nelze postihnout všechny aspekty modelování, kterými se IDEF3 zabývá, detailní informace lze nalézt v dokumentu [IDEF01], ze kterého jsem při popisu metody IDEF3 čerpal. Výše uvedené obrázky jsou rovněž převzaty z dokumentu [IDEF01].



Obr. 4.12. – Přehled symbolů notace IDEF [IDEF01]

## 4.2 Metamodel IDEF 3

Již bylo řečeno, že IDEF 3 uplatňuje při popisu procesů dvě základní strategie. Strategii soustředící se na procesy a aspekty chování podniku a strategii, která je zaměřena na strukturální aspekt reality a která je primárně zaměřena na popis podnikových objektů, jejich transformací a stavů. Takovéto pojetí popisu je velice výhodné neboť umožňuje postihnout jak strukturu tak chování reality.

Rozdělení jazyka do dvou základních sad symbolů (neuvažujeme-li symboly společné) je na první pohled viditelné i na obrázku č. 4.13. Při propojování obou pohledů dohromady hraje významnou úlohu symbol Reference.

Reference použitá v procesním diagramu může odkazovat buď na jednotlivý aktivní element (UOB nebo uzel), komplexní procesní schéma nebo objektové schéma. Pokud odkazuje na aktivní element nebo procesní schéma, vyjadřuje předání řízení těmto elementům. K propojení obou pohledů dochází v situaci, kdy je jejím obsahem objektové schéma. Pak smí být připojena pouze k určité UOB a vyjadřuje, že vykonání této UOB je podmíněno přejitím určitého objektu do určitého stavu (vykonání objektového schématu).

K propojení objektových a procesních schémat dochází také v situaci, kdy je Reference použita v objektovém schématu. V toto případě může obsahovat odkaz buď na UOB nebo na procesní schéma a může být připojena jak k Objektu, tak k transformační vazbě. Pro bližší popis použití referencí v objektových schématech viz. kap. 4.1.2.



## 5 Jazyky BPML a BPMN

Konsorcium Business Process Management Initiative (BPMI) je sdružením významných firem z oblasti vývoje informačních systémů. Vyvíjené standardy tak odrážejí požadavky a zkušenosti předních osobností oblasti modelování firemních procesů. V této kapitole bych chtěl představit dva základní standardy, jejichž vývoji se BPMI věnuje. Business Process Modeling Notation (BPMN) je standardem pro grafickou reprezentaci firemních procesů v diagramech. Business Process Modeling Language (BPML) pak představuje textový jazyk pro modelování a popis procesů, jehož syntaxe vychází z Extensible Markup Language (XML).

### 5.1 Business Process Modeling Language

Modelovací jazyk BPML je standardem exekutivního modelovacího jazyka. Jeho první verze byla po několikaletém vývoji a veřejném připomínkování uvolněna v listopadu roku 2002. Koncepce BPML je zaměřena především na spolupráci a koordinaci firemních procesů mezi obchodními partnery. K tomu účelu je definován abstraktní standard business-to-business protokolů. Jeho koncepce zaručuje platformní nezávislost jazyka, uplatnitelnost u již existujících systémů s různými typy implementačních protokolů, možnost integrace existujících aplikací do systému, možnost sdílení procesní repository, atd. BPML patří mezi tzv. exekutivní jazyky, což znamená, že jeho modely jsou spustitelné v definovaném prostředí. Tato přednost je umožněna textovou podobou jazyka a modelů, která vychází z technologie Extensible Markup Language (XML). Pro každý element jazyka je tak definována speciální XML značka s parametry. Takovýto kód je pak relativně snadno analyzovatelný pomocí speciálních softwarových nástrojů, které jej dokáží interpretovat.

#### 5.1.1 Základní prvky jazyka

Dokument [BPML01] definuje devět základních elementů, ze kterých se skládá jazyk BPML:

- Aktivita

- Kontexty
- Procesy
- Vlastnosti
- Signály
- Plány
- Výjimky
- Transakce
- Funkce

### **Aktivita**

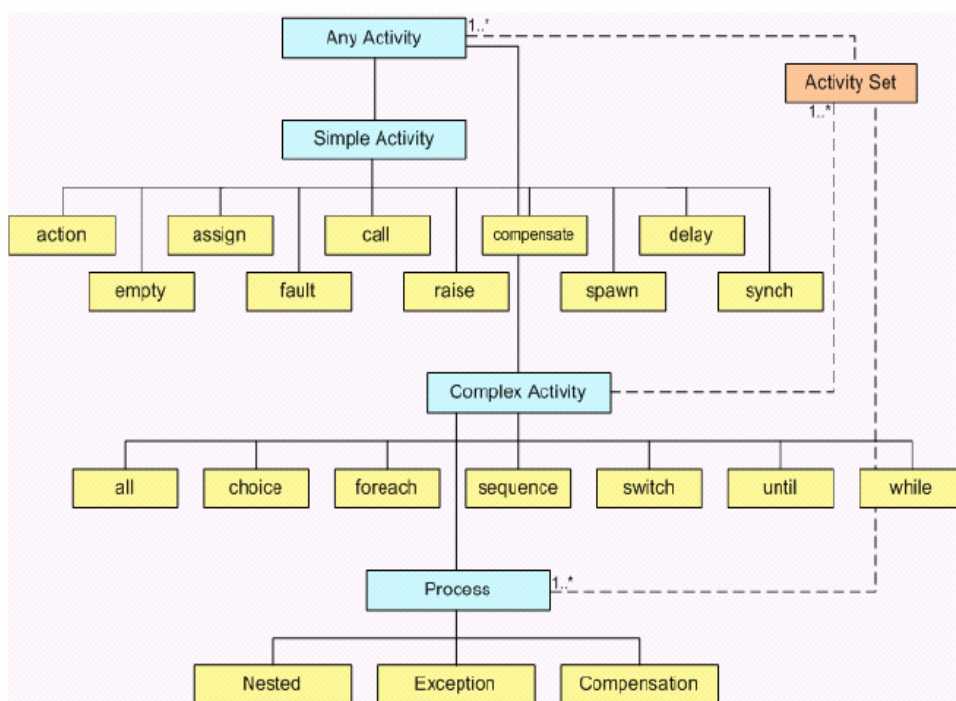
Základním prvkem jazyka je aktivita. Aktivita představuje komponentu, zajišťující určitou funkcionalitu. Komplexní aktivity jsou složeny z atomických aktivit, jejichž provádění řídí. Firemní proces je pak komplexní aktivitou, která může být sama součástí většího procesu. Definice aktivity specifikuje způsob, kterým je aktivita spouštěna a prováděna. Chování je definováno pomocí hodnot atributů aktivity. BPML popisuje 17 typů aktivit, rozdělených do dvou skupin:

- **Jednoduché aktivity** – jsou to aktivity, které nemohou být dekomponovány na jiné aktivity. Zpravidla představují určitou elementární operaci, která je součástí komplexní aktivity. Příkladem takových aktivit jsou: *call* – aktivita iniciuje proces a čeká na jeho dokončení, *assign* – aktivita přiřazuje hodnoty atributům, *raise* – vyše signál.
- **Komplexní aktivity** – jsou složeny z více aktivit. Mohou se skládat jak z jednoduchých, tak z komplexních aktivit. Komplexní aktivity umožňují řídit průběh vykonávání aktivit, které jsou do nich vnořeny. Umožňují specifikovat:
  - jaké aktivity spouštět - Komplexní aktivity se zpravidla skládají z jedné sady aktivit, která je po inicializaci aktivity provedena. Výjimku tvoří aktivity *choice* a *switch*, které se

mohou skládat z více sad aktivit, a které na základě vstupních podmínek rozhodují, která sada bude provedena.

- kolikrát se má daná sada aktivit provádět – aktivita *while* spouští operace opakovaně dokud je splněna specifikovaná podmínka, aktivita *until* spouští operace dokud specifikovaná podmínka splněna není a aktivita *foreach* provede sadu aktivit pro každý objekt seznamu
- v jakém pořadí aktivity provádět – aktivita *sequence* spouští aktivity sekvenčně, aktivita *all* spouští aktivity paralelně všechny najednou

Výše uvedené tři druhy komplexních aktivit tedy obsahují prováděné činnosti a zároveň umožňují specifikovat jejich strukturu, ve smyslu provádění selekce (*choice*, *swich*), iterace (*while*, *until*, *foreach*), sekvence/paralelizmu (*sequence*, *all*). Názorný popis vztahů mezi aktivitami, komplexními aktivitami a procesy je znázorněn na obrázku č. 5.1.



Obr. 5.1 – Klasifikace činností BPML [BPML01]

## Kontexty

Pro aktivitu lze definovat jejich kontext. „**Kontext aktivity** definuje obecné chování všech aktivit, které jsou v tomto kontextu spouštěny. Těmito vlastnostmi jsou především obsluha výjimek a chyb, základní sémantika, definice časových omezení, apod.“[BPML01] Kontext je tedy definicí prostředí, ve kterém jsou prováděny spřízněné aktivity. Aktivity, které jsou spouštěny v daném kontextu je využívají k výměně informací pomocí vlastností v tomto kontextu definovaném (např. jedna aktivita nastaví během svého průběhu hodnotu vlastnosti kontextu a jiná aktivita tuto hodnotu využije při svém spuštění). BPML umožňuje vytvářet hierarchickou strukturu kontextů. To znamená, že kontext může být součástí jiného kontextu

## Procesy

„Proces v BPML je typ komplexní aktivity, která definuje vlastní kontext pro spouštění aktivit, v procesu obsažených.“[BPML01] Z vlastností procesu jako komplexní aktivity tedy vyplývá, že se skládá z více aktivit, jejichž spouštění řídí. Proces sám pak může být součástí jiného procesu. Proces bývá navrhován jako znovupoužitelná, relativně uzavřená a komplexní jednotka práce. BPML rozeznává tři základní druhy procesů:

- top-level procesy – procesy, které jsou definovány nezávisle na jiných procesech
- vnořené procesy – procesy, které jsou definovány tak, aby se spouštěly v určitém kontextu (jejich definice je součástí definice kontextu)
- výjimekové procesy – procesy pro obsluhu výjimek a chyb
- kompenzační procesy – opravné procesy

Nová instance procesu může být vytvořena jinou aktivitou, zasláním zprávy, nebo vyvoláním signálu.

- Aktivitou mohou být spouštěny pouze ty procesy, které se nacházejí ve stejném kontextu jako daná aktivita. Tento způsob se zpravidla používá u vnořených procesů, nebo procesů, které jsou součástí větší kompozice.
- Proces, který je spouštěn pomocí zprávy může být samostatnou službou, kterou je možno spouštět pomocí procesů, které probíhají ve zcela

odlišných systémech. To umožňuje vytvářet relativně nezávislé služby, které jsou umístěny a spouštěny v odlišných prostředích.

- Pomocí signálu mohou být spouštěny pouze procesy, které jsou součástí stejného kontextu jako signál. Signály se používají pro koordinaci spouštěných procesů v rámci daného kontextu. Proces také musí být součástí nějakého rozsáhlejšího procesu, jehož ostatní aktivity signály vysílají.

### **Vnořené procesy**

Vnořený proces je proces, jenž je umístěn v daném kontextu a je používán jako aktivita v rámci většího procesu. Speciálním případem vnořených procesů jsou **výjimkové procesy** a **kompenzační procesy**.

Výjimkové procesy umožňují rodičovskému procesu reagovat na výjimečné stavy, které mohou nastat při komunikaci s ostatními procesy, nebo při chybách vnořených aktivit. Výjimkový proces může být spuštěn pomocí zprávy, nebo signálem v rámci kontextu. Jakmile je iniciován, všechny ostatní aktivity kontextu jsou přerušeny a rodičovský proces se dostává do stavu *aborted*. BPML definuje také tzv. *fault handlers*, které umožní komplexním procesům zotavit se z chyby tím, že ukončí aktivitu, aniž by se chyba přenesla na rodičovský proces. Obsluha chyb je spouštěna až po výjimkových procesech a tím jsou proti chybám ošetřeny i tyto procesy.

Druhým speciálním typem vnořených procesů jsou kompenzační procesy, které umožňují vrátit systém do stavu, ve kterém se nacházel před spuštěním rodičovského procesu. Tyto procesy mohou být spouštěny pouze jednou, po dokončení rodičovského procesu a to i v případě, že skončil ve stavu *aborted*, tedy když skončil chybou. Tyto procesy jsou prostředkem udržení konzistence systému.

### **Plány**

„Plán (ang. schedule) představuje sérii specifických časových událostí, ve kterých je spuštěn proces.“ [BPML01] Plán představuje časový rozpis, podle něhož mohou být spouštěny procesy. Čas může být určen relativně – např. proces A se má spustit po x minutách po spuštění procesu B, nebo absolutně – např. proces se má spouštět v tolik a

tolik hodin každý den. Plánem může být také časové omezení délky trvání procesu, nebo nejdelší možná doba od určité události, po kterou se proces může spustit.

### Transakce

„Transakční protokoly umožňují dvěma procesům, které spolu komunikují pomocí zpráv, koordinovat dokončení jejich aktivit. Transakční protokoly se používají, když je třeba zajistit, aby se aktivity dvou procesů dokončily nebo selhaly společně.“ [BPML01] Transakce se používají pro zachování konzistence systému nebo pro komunikaci mezi dvěma systémy.

Stejně jako ve standardu CEN ENV 12204, který pracuje s obecnými konstrukty základních jevů v podniku, je každý element jazyka BPML popsán svou definicí, která se skládá z jeho základních atributů. Tato definice je následně implementována výrazovými prostředky jazyka XML. Pro názornost příklad definice elementu Proces, převzatý z [BPML01].

Atribut	Popis
<i>name</i>	jméno procesu
<i>documentation</i>	dokumentace
<i>identity</i>	jedno nebo více identifikačních jmen
<i>persistent</i>	atribut persistence
<i>event</i>	spouštěcí událost
<i>parameters</i>	vstupní parametry
<i>activity set</i>	sada aktivit
<i>compensation</i>	kompensační proces

Syntaxe pro implementaci definice je pak následující:

```
<process
  name = NCName
  identity = list of QName
  persistent = boolean : false>
  dále obsahuje oddíly: (documentation, (event | parameters),
  context, compensation)
</process>
<event
  activity = list of NCName
  exclusive = boolean : false>
```

## 5.2 Business Process Modeling Notation

BPML je textovým modelovacím jazykem, který vytváří formální mechanismus pro specifikaci modelů v aplikacích. Právě jeho textová podoba může bránit uživatelům business modelů orientovat se v modelovaném systému. Grafická podoba informace o firemních procesech je pro člověka srozumitelnější, proto se BPMN snaží vytvořit komplexní grafickou notaci pro modelování firemních procesů. Procesy jsou modelovány v Business Process Diagramu (BPD) s různou úrovní abstrakce tak, aby byly využitelné pro širokou škálu osob, které firemní procesy modelují nebo řídí. Cílem BPMN je vytvořit takovou notaci, která by firmě umožnila rozumět jejím procesům, a aby byla schopna tyto procesy ve standardizované podobě komunikovat s jinými firmami. To by mělo umožnit, aby firmy byly schopny navzájem porozumět svým procesům, což je nezbytnou podmínkou toho, aby mohly spolupracovat na vývoji společné B2B architektury. Sama BPMI definuje cíl své snahy takto: „BPMI využívá zkušenosti business proces notací, které předcházely BPMN, aby vytvořila notaci nové generace, která kombinuje čitelnost, flexibilitu a šířitelnost.“ [BPMN01]

BPMN slouží pro potřeby různých osob a subjektů, které jsou v různém vztahu k podniku. Tyto osoby pochopitelně vyžadují různé druhy informací o podnikových procesech. Aby uživatelé mohli jasně rozlišovat, které informace jsou pro ně vhodné, a které ne, lze v BPD rozlišit tři základní oblasti či sub-modely:

- Privátní (vnitřní) procesy – jsou ty procesy, které jsou vnitřními procesy určité organizace
- Abstraktní (veřejné) procesy – představují interakce mezi privátním procesem a jiným procesem nebo účastníkem (organizací, institucí, apod.) Znázorňují jen ty informace, které stojí mimo privátní procesy. Jedná se o specifikaci toho, jakým způsobem má okolní svět komunikovat a spolupracovat s privátními procesy
- Procesy spolupráce – popisují interakce mezi dvěma nebo více business entitami (např. podniky).

Jedním z problémů, kterým se BPMN musela vypořádat byl, jakým způsobem zajistit, aby diagramy byly dostatečně jednoduché, aby jim analytici a uživatelé rozuměli, a zároveň aby byly natolik detailní, aby umožňovali snadný přechod do exekutivních modelovacích jazyků (např. BPML, BPEL4WS). Toto schizma je vyřešeno rozdělením grafických symbolů do dvou sad. První sada obsahuje nezbytné základní symboly, pomocí nichž lze modelovat většinu procesů, a která vyhovuje požadavku jednoduché a přehledné notace. Druhou skupinou je pak široká škála symbolů, která je rozšířením základní sady, a která umožňuje modelovat procesy na velice detailní úrovni. Symboly této rozšířené sady jsou doprovázeny negrafickými atributy, které umožňují převedení grafické podoby procesu do exekutivního jazyka, což z BPMN činí velice silný nástroj pro modelování. V následujícím textu představím nejdůležitější a nejzajímavější symboly z první i druhé skupiny. Přehled všech užívaných symbolů je uveden v příloze č.5.

Základními elementy, které se v BPD vyskytují jsou:

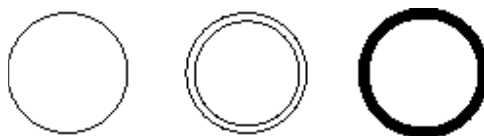
- Událost
- Činnost
- Brána
- Sekvenční tok
- Tok zpráv
- Asociace
- Bazén (Pool)
- Dráha (Lane)
- Datový objekt
- Skupina
- Poznámka

### **Událost**

Obecně je událostí „něco“, co se stane za běhu business procesu. Termín událost pokrývá takové věci jakou jsou začátek činnosti, konec činnosti, změna stavu objektu,

přijetí zprávy, atd. BPMN však do svých diagramů zahrnuje pouze takové události, které ovlivňují pořadí nebo načasování aktivit procesu, a rozděluje je do tří základních typů:

- událost typu Start – jedná se o událost, ve které určitý proces začíná. Jelikož se jedná o počáteční krok procesu, bývá tato událost zpravidla něčím spuštěna. Takovým podnětem může být například zpráva, pravidlo nebo čas. Událost Start je značena tenkou kružnicí (obr. 5.2 vlevo), v níž může být symbolem specifikována příčina spuštění (příloha č. 6).
- událost typu Konec – jedná se o událost, ve které určitý proces končí. Proces jako posloupnost aktivit má zpravidla na svém konci nějaký výsledek. Tím může být např. zpráva nebo chyba. Obdobně jako událost Start je událost Konec značena kružnicí, tentokrát však tučnou čarou (obr. 5.2 vpravo), a obdobně se uvnitř může symbolem vyjádřit druh výsledku procesu (příloha č. 7).
- událost typu mezikrok (ang. intermediate) – mezikroky jsou událostmi, které se uskutečňují v průběhu procesu a ovlivňují jeho průběh. Mohou znázorňovat např. časové lhůty nebo očekávané zprávy v rámci procesu. Značí se dvojitou kružnicí (obr. 5.2), v níž může být opět specifikován druh události (příloha č. 8).



Obr. 5.2 – Události typu Start, Mezikrok, Konec

## Činnost

„Činností je práce, která je vykonávána v rámci business procesu.“ [BPMN01]  
Jedná se tedy o základní jednotky chování systému. Činnosti mohou být atomické nebo složené. BPMN rozeznává tři druhy aktivit, které lze najít v procesním modelu:

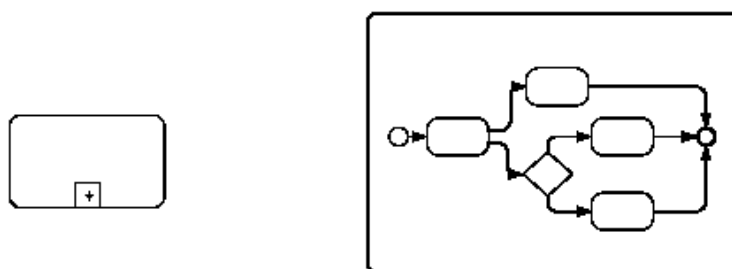
- Procesy
- Podprocesy
- Úlohy (ang. Tasks)

## Proces

Podle BPMN představuje proces složenou aktivitu vykonávající určitou práci v podniku. V BPD je proces znázorněn jako skupina činností a kontrolních prvků, které určují pořadí vykonávání činností. Procesy jsou v modelech hierarchicky strukturovány a definují se na různých úrovních této hierarchie. To znamená, že proces se zpravidla skládá z podprocesů, a ty se mohou dále dělit na další podprocesy. Každý jednotlivý proces je pak zobrazován v samostatném Bazénu (viz. dále).

## Podproces

Podproces je složenou aktivitou, která je součástí jiného procesu. V rámci diagramu je pak podproces grafickým symbolem, který odkazuje na jiný proces. V diagramech lze využít dvě základní zobrazení podprocesu. Prvním je uzavřené zobrazení, které skrývá detaily průběhu podprocesu. Druhým pak je zobrazení rozšířené, z něhož je průběh podprocesu čitelný. Příklady základních typů zobrazení podprocesu jsou uvedeny na Obr. 5.3.



Obr. 5.3- Uzavřené a rozšířené zobrazení podprocesu

## Úloha

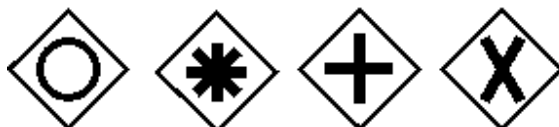
Úlohy jsou základními činnostmi, ze kterých se skládají procesy. Jedná se o atomickou činnost, kterou tedy nelze dále dělit na jiné činnosti a představuje nějaký reálný úkon nebo aktivitu. V diagramech je Úloha znázornována zaobleným obdélníkem. Symbolem uvnitř obdélníku lze také specifikovat, zda se jedná o opakující se, násobnou či kompenzační činnost (Obr. 5.4).



Obr. 5.4 – Obecná, opakující se, násobná a kompenzační úloha

### Brána

Brána je grafickým prvkem, který má obdobnou funkci jako mají Uzly v metodice IDEF. IDEF používá pro tento typ větvení výrazu Junctions, zatímco BPML používá termín Gateways. Jedná se o kontrolní body, které obstarávají větvení procesu. V těchto bodech se proces podle určitých podmínek rozděluje do více větví nebo se v něm větve naopak spojují. Brána zároveň umožňuje specifikovat jakým způsobem se mají procesy větvit nebo spojovat. Brány umožňují modelovat všechny typy logických větvení OR, XOR a AND. Dále nabízí možnost definovat vlastní podmínky větvení tzv. **komplexní bránou**. Komplexní brána slouží pro použití v situacích, které nemohou být ošetřeny pomocí uzlů OR, XOR nebo AND. Rozhodovací proces komplexní brány je třeba přesně definovat. Může například rozhodovat na základě vstupních podmínek mezi různými skupinami činností.

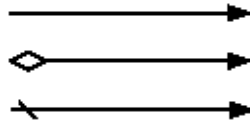


Obr. 5.5 – OR, Komplexní, AND a XOR Uzel

### Sekvenční tok

Sekvenční tok se používá pro vyjádření pořadí, ve kterém budou aktivity v rámci procesu prováděny. Sekvenční tok je symbolizován šipkou, která směřuje od zdrojového objektu k cílovému objektu. Těmito objekty mohou být Události, Činnosti nebo Brány. BPMN používá tři druhy sekvenčních toků. Základní typ sekvenčního toku je znázorňován obyčejnou šipkou (obr. 5.6 nahoře) a vyjadřuje obyčejný vztah následnosti zdrojového a cílového objektu. Druhým typem je **podmínkový sekvenční tok**, který vyjadřuje nutnost splnění určité podmínky před tím, než bude proces pokračovat (obr. 5.6

uprostřed). Třetím typem je **defaultní sekvenční tok**, který se používá v situaci, kdy je zdrojovým objektem XOR uzel. Jestliže při průběhu procesu nebyly splněny podmínky ani pro jeden objekt vycházející z daného XOR uzlu, pak bude proces pokračovat objektem, který je cílovým objektem defaultního sekvenčního toku.



Obr. 5.6 – Sekvenční toky

### Tok zpráv

Slouží pro znázornění přenosu zprávy od jedné entity procesu k jiné entitě procesu. V BPMN je pohled entity na proces znázorněn v samostatném bazénu (viz. dále), proto se symbol pro tok zpráv – přerušovaná šipka – vztahuje ke dvěma bazénům. Jeho použití je vidět na příkladu v příloze č. 10.



Obr. 5.7 – Tok zpráv

### Bazén a Dráha

Pojmy Bazén a Dráha vycházejí z podoby grafických symbolů, které pomáhají organizovat jednotlivá schémata procesů v rámci BPD, a které umožňují procesy specifikovat s ohledem na různé úhly pohledu jejich účastníků a nutnost koordinace aktivit při B2B spolupráci. Bazén (Obr.5.8) představuje kontejner v němž je umístěn model procesu z pohledu jednoho účastníka(např. podniku). Pokud položíme dva Bazény, tedy pohledy dvou podniků na jeden proces, vedle sebe je možné z nich vyčíst nutné koordinační aktivity při B2B spolupráci.



Obr. 5.8 – Bazén

Dráha (Obr. 5.9) je pak oddíl bazénu, kterým se dají vyjádřit různé pohledy na proces či odlišit vykonavatele a aktéry jednotlivých činností v rámci daného procesu.



Obr. 5.9 – Dráhy

Bazény a dráhy jsou velice dobrým nástrojem pro organizaci schémat, zvláště pokud je třeba modelovat spolupráci B2B, která je koordinována pomocí zasílání zpráv. Příklady použití jsou uvedeny v příloze č. 9.

### **Datový objekt , Skupina, Poznámka**

Tyto tři symboly (Obr. 5.10) slouží především ke zvýšení srozumitelnosti diagramů a zdůraznění kontextu.

- Datový objekt upozorňuje na data, která jsou nějakou aktivitou vyžadována nebo produkována
- Symbol skupiny se používá z analytických a dokumentačních důvodů v situacích, kdy je vhodné v diagramu zdůraznit bližší souvislost skupiny elementů. Skupina se značí čerchovaným rámečkem, který nemá žádný vliv na funkci zahrnutých prvků.
- Poznámka je klasickým nástrojem pro zdůraznění podstatných skutečností vážících se k nějakému prvku modelu.



Obr. 5.10 – Datový objekt, Skupina, Poznámka

Základní grafické symboly lze dle jejich vlastností rozdělit do čtyřech skupin: *tokové objekty*, *spojovací objekty*, *dráhy*, *artefakty* [WHI02]. Přehled této klasifikace poskytuje tabulka č. 5.1.

Tokové objekty	Spojovací objekty	Dráhy	Artefakty
Událost	Sekvenční tok	Bazén	Datový objekt
Činnost	Tok zpráv	Dráha	Skupina
Brána	Asociace		Poznámka

Tab. 5.1 – Klasifikace elementů BPMN

### 5.3 Metamodel BPMN

Vzhledem k tomu, že tato práce je zaměřena především na modelovací standardy, které se zabývají popisem základní logiky průběhu procesů v rámci podniku, bude i tento metamodel zaměřen na základní sadu symbolů BPMN a nebude přihlížet k některým specifikům, které přináší použití sady rozšířené, která je primárně určena pro přípravu modelu na převedení do programového kódu. Diagram na obrázku 5.11 zobrazuje vztahy elementů BPMN s následujícími omezeními:

- Událost typu start může být pouze zdrojovým objektem pro sekvenční tok, sekvenční tok nesmí vstupovat do start události; ze start události musí vycházet jeden nebo více sekvenčních toků
- Událost typu start nesmí být producentem zpráv, ale může být příjemce libovolného počtu zpráv (např. určitá zpráva je podnětem k zahájení procesu)
- Událost typu konec může být pouze cílový objektem pro sekvenční tok, může do ní vstupovat více sekvenčních toků, minimálně však jeden (musí být spojena se vstupním sekvenčním tokem)
- Událost typu konec může být pouze zdrojovým objektem pro tok zpráv (např. výsledek ukončeného procesu je předán ke zpracování jiným procesem), nesmí být tedy příjemcem zprávy

- Událost typu mezikrok může být pouze cílový objektem pro tok zpráv
- Tok zpráv slouží k znázornění pohybu zprávy mezi dvěma entitami, které jsou zprávou připraveny vyslat resp. přijmout. Tyto entity jsou v BPMN reprezentovány pomocí bazénů, tok zpráv může tedy spojovat pouze objekty, které jsou součástí různých bazénů, resp. nesmí spojovat objekty v jednom bazénu.
- Sekvenční tok slouží k vyjádření pořadí aktivit vykonávaných v rámci procesu. Každý tok má pouze jeden zdroj a jeden cíl. Zdrojem či cílem mohou být pouze události (start, mezikrok, konec), činnosti (úloha, podproces) nebo brány.

## 5.4 Shrnutí

Jazyk BPML je velice propracovaným nástrojem pro modelování. Ve spojení s BPMN pak vytvářejí komplexní a dobře využitelný přístup k podnikovým procesům. Vývoj notace BPMN sice není dosud ukončen, zejména modelování abstraktních procesů a procesů spolupráce není plně definováno, avšak v oblasti interních podnikových procesů již teď poskytuje širokou škálu možností. Přímá vazba na jazyky BPEL4WS a BPML pak skýtá velice dobrou možnost využití v softwarových modelovacích nástrojích. S rozvojem elektronické spolupráce podniků a virtuálních podniků je také pro využití standardů BPMI velice výhodná jejich orientace směrem k podpoře B2B spolupráce.



## 6 Jazyk UML

Modelovací jazyk UML vznikl v průběhu 90. let jako syntéza tří přístupů k modelování. Představiteli těchto přístupů byli Grady Booch, Jim Rumbaugh a Ivar Jacobson. Spojením a standardizací jejich metodologií vznikl The Unified Modeling Language (UML). Zpočátku byl vyvíjen společností Rational. Později byla kontrola nad UML převzata konsorciem Object Management Group (OMG), které v roce 1997 představilo první verzi standardu – UML 1.0. Tato specifikace byla později modifikována verzemi 1.1 až 1.5.

Vývoj v informačních technologiích a nárocích na modelování si však vyžádal rozsáhlejší zásah do tohoto jazyka. Odezvou na tuto potřebu je UML 2.0, jehož specifikace se skládá ze čtyřech základních dokumentů: UML 2.0 Object Constraint Language, UML 2.0 Diagram Interchange, UML 2.0 Infrastructure, UML 2.0 Superstructure. Druhá verze UML představuje značný pokrok v možnostech tohoto jazyka. Nová rozšíření by měla umožňovat popis elementů vyskytujících se v současných softwarových technologiích a snaží se poskytnout podporu pro principy Model Driven Architecture (MDA) a Service-Oriented Architecture (SOA).

„The Unified Modeling Language je grafickým jazykem pro vizualizaci, specifikaci, konstrukci a dokumentaci artefaktů softwarových systémů.“ [UML01] Takto definuje svůj standard, který je představován rozsáhlou skupinou nástrojů, samo konsorcium OMG. Přestože se UML používá především k objektově orientovanému modelování softwarových systémů, je koncipován jako univerzální modelovací nástroj. Lze jej tedy využít i pro modelování podnikových procesů.

### 6.1 Architektura UML

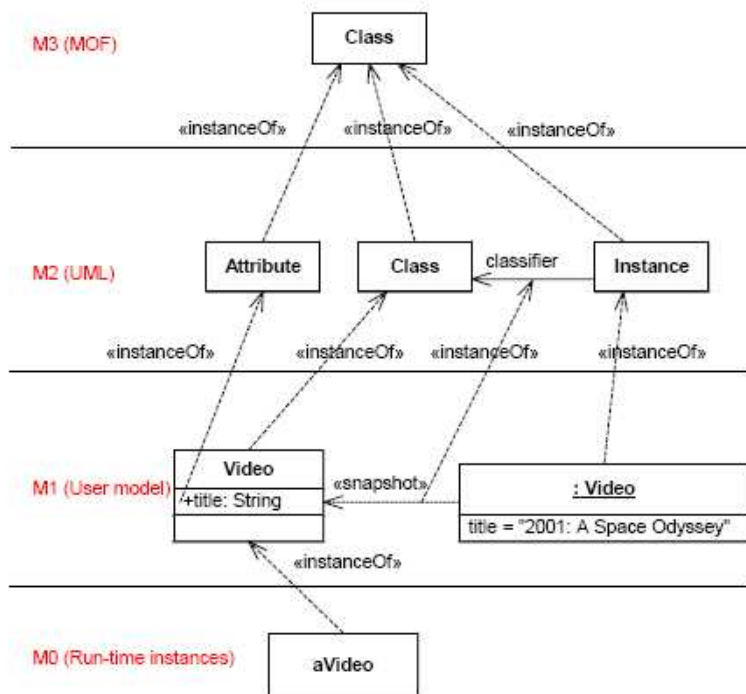
Architektura UML je založena na čtyřvrstevém modelu.

- Nejvyšší vrstvou je meta-metamodel který vymezuje základní výrazové prostředky pro metamodely. Objekty, které mohou být na této úrovni definovány jsou například MetaAtribut, MetaTřída, MetaOperace. Vrstva meta-metamodelu je v koncepci OMG reprezentována standardem Meta

Object Facility (MOF). MOF je abstraktní jazyk, který definuje klíčové elementy, syntax a strukturu metamodelů.

- Druhou vrstvu představuje metamodel, který definuje základní elementy a principy jednotlivých modelů UML. Třída metamodelu je instancí meta-třídy z meta-metamodelu. Hlavním cílem metamodelu je vytvořit jazyk pro specifikaci modelů. Jeho součástí jsou všechny modely, rozdělené do balíčků, se vzájemnými vztahy.
- Třetí vrstvu tvoří user model, který je instancí metamodelu a zároveň abstrakcí uživatelských objektů (run-time instances). Model je instancí metamodelu, která je určená pro řešení určitého typu problému. Konkrétní sada modelů tvoří návrh systému.
- Nejnižší vrstva se nazývá run-time instances. Představuje definované uživatelské objekty, které se vyskytují v modelovaném prostředí. Každý uživatelský objekt vzniká na základě specifikací uvedených v modelu.

Tuto čtyřúrovňovou architekturu vystihuje následující obrázek:

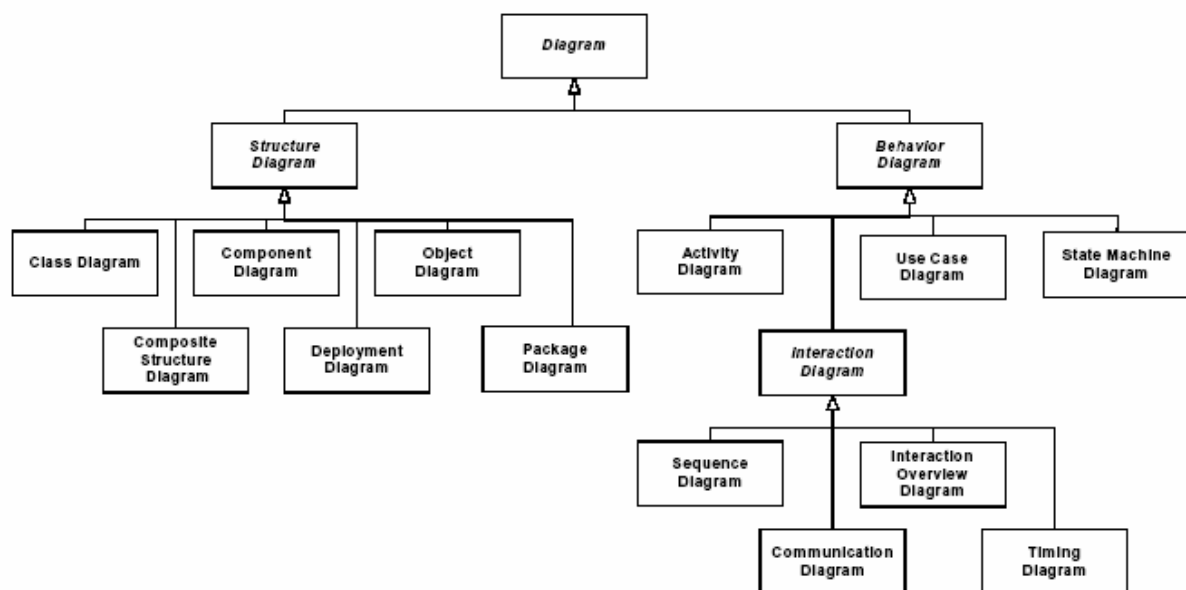


Obr. 6.1 - Čtyřúrovňová architektura UML [UML01]

## 6.2 Základní diagramy UML

UML používá pro modelování třináct základních diagramů, které jsou rozděleny do tří skupin:

- Diagramy chování: diagramy zachycující aspekty chování systému nebo business procesu. Jsou jimi activity diagram, state machine diagram a use case diagram. Zahrnují také všechny diagramy interakce
- Diagramy interakce: jsou podskupinou diagramů chování, která se zaměřuje na popis interakcí objektů
- Diagramy struktury: typ diagramů, který popisuje elementy systému, které jsou stabilní vůči času. Tato skupina zahrnuje class diagram, composite structure diagram, component diagram, deployment diagram, object diagram a package diagram



Obr. 6.2 – Přehled diagramů UML 2.0 [UML02]

## 6.3 Activity diagram

Zřejmě nejvhodnějším prostředkem pro modelování business procesů v UML je Activity diagram. Tento diagram zachycuje činnosti a události vyskytující se v systému. Jelikož business proces obecně představuje určitou sekvenci činností, je možné pro jeho

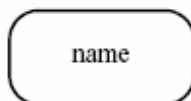
modelování Activity diagram použít. Následující text by měl čtenáři poskytnout přehled o základních elementech, které jsou zpravidla obsaženy v balíčcích Fundamental Activities, Basic activities a Intermediate activities z balíčků aktivit UML. Jedná se o vybrané elementy, kterým je přiřazena notace a které slouží k modelování procesů na takové úrovni abstrakce, která vyhovuje potřebám této práce. Obrázky v této subkapitole jsou převzaty z [UML02].

Základními symboly používanými v Activity diagramu jsou:

- Akce
- Aktivita
- AcceptEventAction
- Start (InitialNode)
- Konec (ActivityFinalNode)
- Ukončení toku (FlowFinalNode)
- Přechod
- Větvení(Fork)/Spojení(Join)
- Rozhodnutí/Sloučení (Decision/Merge)
- Dráhy (Partitions)

### **Akce**

Akce představuje atomickou činnost, již předchází jiná činnost nebo událost. Akce je spuštěna v tom okamžiku, kdy je akce, která jí předchází dokončena (může jich být i více). Akce je používána pro vyjádření dílčího kroku procesu. Jelikož se jedná o atomickou činnost, nemůže reprezentovat další dílčí činnosti nebo proces. Symbol Akce je znázorněn na obrázku č. 6.3.

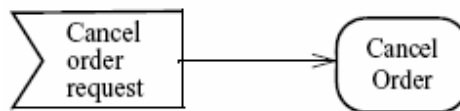


Obr. 6.3 – Akce

### **AcceptEventAction**

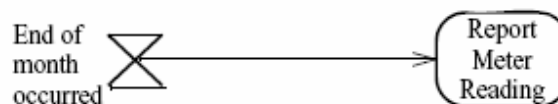
„AcceptEventAction je akcí, která čeká na výskyt události, splňující specifikované podmínky.“[UML02] Jak z definice vyplývá, je AcceptEventAction zvláštním druhem

akce. Jedná se o prvek, kterým se UML vyrovnává s existencí událostí. Notace rozlišuje dva typy AcceptEventAction. První typ (Obr. 6.4) slouží pro vyjádření specifické události, která je impulsem pro spuštění činnosti nebo procesu, nebo stavu, který je podmínkou pokračování procesu.



Obr. 6.4 – AcceptEventAction

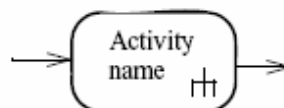
Druhý typ (Obr. 6.5) pak vyjadřuje výskyt časové události a používá se pro něj symbol přesýpacích hodin.



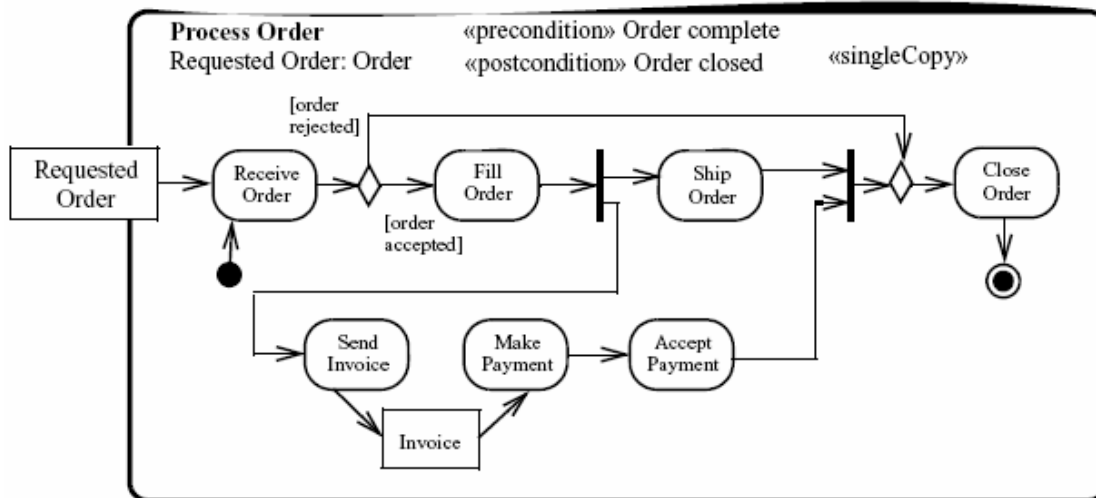
Obr. 6.5 – Časová událost

### Aktivita

Aktivita specifikuje koordinaci podřízených chování a modeluje tok vykonávání Akcí v ní obsažených. Ve své podstatě UML rozumí pojmem Aktivita specifikovaný proces, jehož základními stavebními prvky jsou Akce, řídicí toky mezi nimi a další elementy notace. Součástí Aktivit může být tvorba hierarchie Aktivit, kdy Aktivita vyvolává další Aktivitu. Notace pro Aktivitu je dvojitá. Prvním typem je zjednodušené vyjádření pomocí zaobleného čtyřúhelníku se symbolem vidličky (Obr. 6.6), druhý je pak představován vyznačením hranice kolem prvků, které jsou součástí Aktivity, a jménem Aktivity, případně podmínek vykonávání (Obr. 6.7).



Obr. 6.6 – Aktivita



Obr. 6.7 – Aktivita – rozšířené zobrazení

### Start

Start je počátečním bodem, ve kterém začíná sekvence činností, je-li inicializována Aktivita. Jeho použití není vyžadováno, ale pomáhá zvýšení srozumitelnosti diagramu. Symbolem, který je používán pro Start, je plné kolečko (Obr. 6.8).



Obr. 6.8 – Start

### Konec

Konec je naopak koncovým bodem, v němž činnosti Aktivit končí. Aktivita může mít samozřejmě více koncových bodů a ten, kterého je dosaženo nejdříve, ukončí veškeré činnosti v Aktivitě.

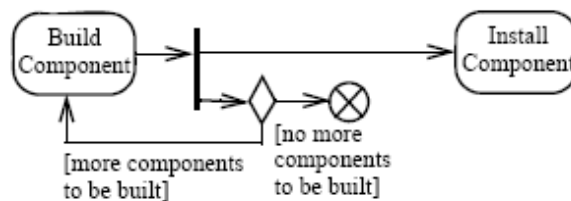


Obr. 6.9 – Konec

### Ukončení toku

Ukončení toku se používá v situaci, kdy je třeba uzavřít nějaký tok tak, aby to nemělo vliv na ostatní toky v rámci Aktivit. Příklad užití je uveden v obrázku č. 6.10. Jedná se o situaci, kdy je předpokládána iterativní výroba a instalace více komponent.

V okamžiku, kdy je vyrobena poslední komponenta, je výroba zastavena vstupem do Ukončení toku, avšak další aktivity (v tomto případě instalace komponenty) mohou pokračovat ve vykonávání.



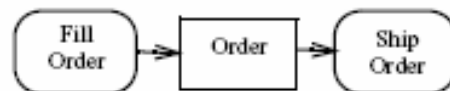
Obr. 6.10 – Ukončení toku

### Řídící tok

Řídící tok představuje vztah následnosti dvou akcí či činností. Jednoduchý přechod se značí šipkou od zdrojové entity k cílové entitě. Názorný příklad lze nalézt na obrázku č. 6.7.

### Objektový tok

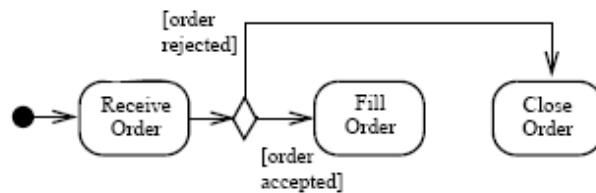
V případě, že součástí předání řízení další činnosti je také nějaký objekt, je vhodné použít v diagramu symbol Objektového toku (Viz. obr. č. 6.11).



Obr. 6.11 – Objektový tok

### Rozhodnutí/Sloučení

Symbol Rozhodnutí představuje kontrolní bod procesu, který umožňuje jeho větvení. V tomto místě diagramu je na základě vstupních podmínek rozhodováno, kterou cestou se bude proces ubírat. Tento symbol, dle toho, jak je nadefinován, může představovat jak exkluzivní tak non-exkluzivní OR rozhodnutí, neboť proces může dále pokračovat pouze jednou i více větvemi. Příklad použití je na obrázku č. 6.12.



Obr. 6.12 - Rozhodnutí

Sloučení, pro které je vyhrazen stejný symbol diamantu jako pro Rozhodnutí, se naopak používá v situacích, kdy je třeba vyznačit spojení několika toků procesu do jednoho proudu. Představuje tak tedy jednoduchý OR uzel, který propouští všechny toky bez vlivu na rozhodování či synchronizaci. Použití je opět možné vyčíst z obrázku č. 6.7 - vpravo.

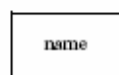
### Větvení/Spojení

Tyto dva elementy jsou označovány stejným symbolem – silnou vertikální čarou. Význam symbolu pak záleží na jeho postavení v diagramu. Větvení označuje začátek vykonávání paralelních aktivit (do symbolu vstupuje jeden tok, vystupuje jich více). Spojení pak označuje ukončení paralelního vykonávání (do symbolu vstupuje více toků, pokračuje pouze jeden). Příklady obou elementů lze nalézt na obrázku č. 6.7.

Použití symbolu Větvení znamená (na rozdíl od Rozhodnutí), že po jeho dosažení je předáno řízení všem tokům, které z něho vycházejí. Obdobně Spojení nedovolí další vykonávání procesu, dokud nejsou vykonány všechny činnosti, které do Spojení vstupují (na rozdíl od Sloučení). Jestliže uzly Rozhodnutí/Sloučení představují OR uzly, pak můžeme Větvení/Spojení považovat za AND uzly.

### Objekt

Symbol Objekt ukazuje na existenci datové či jiné entity v rámci vykonávání procesu. Jsou používány především pro vyjádření skutečnosti, že určitá činnost pracuje s určitým objektem. Pro Objekt se používá symbol obdélníku (Obr. 6.13).



Obr. 6.13 - Objekt

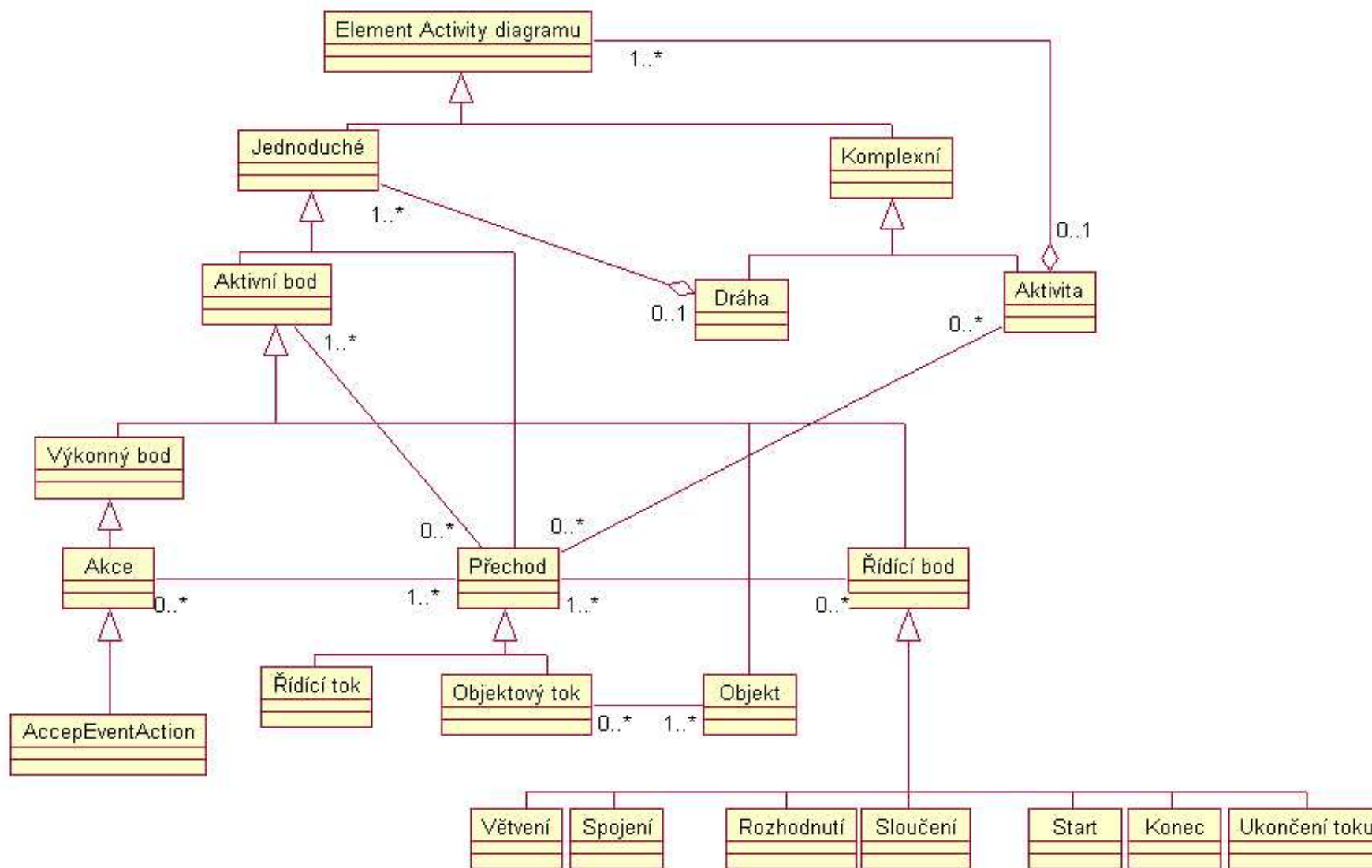
## **Dráhy**

Zatímco v BPMN je hlavním smyslem použití Drah vyjádření odpovědnosti za prováděné činnosti, UML umožňuje využití Drah také k členění akcí a aktivit dle uživatelem definovaných atributů. Příklad použití Drah je uveden v příloze č. 11.

## **6.4 Metamodel elementů activity diagramu UML**

Předchozí subkapitola obsahovala popis elementů activity diagramu a jejich vlastností, metamodel na obrázku č. 6.14 a následující poznámky pak vyjadřují jejich vztahy.

- Jednoduché elementy lze rozdělit na Aktivní body a Přejchody. Aktivní body odpovídají prvkům chování a struktury daného procesu. Přejchody pak vyjadřují předávání řízení mezi Aktivními body.
- Z metamodelu UML je převzato dělení Aktivních bodů na Výkonné, Řídící a Objekty. Výkonné body jsou reprezentovány akcemi vykonávanými v rámci procesu, Řídící body pak aktivně působí na řízení procesu.
- Asociace Aktivního Bodu a Přejchodu vyjadřuje, že přechod se musí pojit alespoň s jedním aktivním bodem
- Objekt se může pojit pouze s Objektovým tokem
- Přejchody vstupující a vystupující ze symbolů Rozhodnutí, Sloučení, Větvení a Spojení musí být stejného typu, tedy buď Řídící toky nebo Objektové toky
- Start a Konec se mohou pojit pouze s Řídícími toky



Obr. 6.14 – Metamodel elementů activity diagramu UML

## 6.5 Shrnutí

UML poskytuje širokou základnu pro různé typy modelů. Kombinace stávajících UML diagramů se snaží postihnout všechny stránky reality (struktura, chování, stavy), které jsou potřebné pro komplexní podnikový model. Activity diagram je jen jeden z nich, ale již na něm je vidět, že zavedené notace a konvence neposkytují dostatečnou variabilitu pro postihnutí všech aspektů modelování procesů. UML je především jazykem pro tvorbu softwarových systémů a poskytuje tak obrovskou škálu elementů modelu, které mají přímou vazbu k tvorbě programového kódu. Méně však umožňuje modelování procesů na vyšší úrovni obecnosti. Lze však využít jeho rozšířitelnosti a upravit jej tak, aby byl schopen požadované aspekty business procesů vyjádřit. Příkladem takové snahy může být např. Eriksson-Penker rozšíření, jehož popis lze najít např. v [REP01] nebo [CHAD01].

## 7 Metamodel standardů modelovacích jazyků

V předcházejících kapitolách byly jednotlivě představeny tři notace pro modelování podnikových procesů. Tyto notace jsou produktem různých skupin lidí s různými přístupy k modelování. I jejich primární účel může být jiný. UML bude pravděpodobně výhodnější použít při vývoji programového systému, než pro modelování procesů za účelem jejich porozumění a reengineeringu, kde bude jistě vhodnější IDEF3. Ačkoliv notace používají odlišnou symboliku a terminologii, snaží se popsat stejné aspekty reality a principy jejich chování, což determinuje použití podobných či stejných elementů v rámci notací. Cílem této kapitoly je na základě těchto společných vlastností vytvořit obecný metamodel, který by vystihoval základní principy modelování podnikových procesů a zároveň znázornil jejich uplatnění v daných notacích.

### 7.1 Vyjádření základních principů modelování

V diagramu na obrázku č. 7.1 jsou hlavní modelovací elementy rozděleny do dvou skupin, které odpovídají požadavkům definovaným mezinárodními standardy. ISO 14258 a ENV 4003 vyjadřují nutnost zohlednit dva základní pohledy na systém podniku.

- Prvním pohledem je **Informační pohled**, který by měl modelovat informaci obsaženou v systému. Tuto informaci představují objekty tvořící systém a jejich struktura. Elementy využívané notacemi pro tento druh pohledu na podnik jsou v metamodelu vyčleněny do skupiny *Elementů zaměřených na modelování systémové informace*.
- Druhým pohledem je pak **Funkční pohled**, jehož smyslem je zachytit chování systému. Popisuje tak funkce, činnosti a procesy, které v podniku probíhají a které zpracovávají, přetvářejí a produkují systémové objekty. Elementy tohoto pohledu na podnik jsou zahrnuty do skupiny *Elementů zaměřených na chování*.

Dalším požadavkem ISO 14258 je, aby modely věnovaly pozornost a byly schopny vyjádřit tři základní aspekty.

- **Aspekt hierarchie** – tento aspekt je přítomen v obou základních pohledech a vyjadřuje myšlenku, že prvek, či systém může být částí jiného

systemu. Obdobně pak systém může být dekomponován na prvky či subsystemy. Jde o uplatnění principu abstrakce. ISO 14258 rozlišuje dva druhy hierarchie, jež lze také nalézt v metamodelu:

- **Hierarchie částí** – v metamodelu je přítomna v obou základních pohledech a je uplatněna dvěma způsoby. Prvním způsobem je existence *komplexních elementů*, které mohou být součástí modelů a které mohou obsahovat další elementy. Druhou je pak existence *kompoziční vazby* v pohledu zaměřeném na objekty.
- **Hierarchie druhů** - představuje vztah generalizace - specializace a je reprezentována *klasifikační vazbou*, jež umožňuje klasifikovat objekty.
- **Aspekt struktury** – standard požaduje, aby byl modelovací jazyk schopen popsat strukturu systému, a to jak strukturu chování, tak strukturu systémových objektů. Oba dva přístupy jsou ze své podstaty vyjádřitelné pomocí elementů specifikovaných v navrhovaném metamodelu.
- **Aspekt chování** – dynamiku systému umožňují vyjádřit *Elementy zaměřené na popis chování*.



## 7.2 Elementy metamodelu

Metamodel je sestaven z konstruktů standardu CEN ENV 12204 a elementů notací IDEF3, BPMN a UML. Zvláštní postavení v metamodelu mají konstrukty definované standardem CEN ENV 12204. Tyto konstrukty jsou základními stavebními kameny modelů podnikových procesů a jsou obsaženy ve všech popisovaných notacích. Standardy jazyků pro ně však používají různé názvy a mohou jim přisuzovat specifický smysl, určený kontextem a potřebami notace. Tato subkapitola se zaměří na popis elementů metamodelu a jejich uplatnění v jednotlivých notacích.

Na metamodelu je vidět rozdělení elementů do dvou základních skupin na *Elementy zaměřené na popis chování* a na *Elementy zaměřené na popis systémové informace*. Smysl a důvody pro toto rozčlenění byly popsány výše.

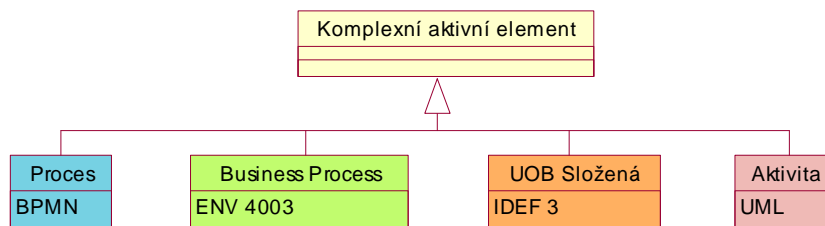
### 7.2.1 Elementy zaměřené na popis chování

Tato třída elementů je rozdělena na dvě základní skupiny. Jsou to *Hlavní elementy*, které slouží k samotnému modelování reality, a *Popisné elementy*, které se používají k vyjádření doplňující či zpřehledňující informace. Popisné a hlavní elementy jsou mezi sebou vázány jednoduchou *Asociací*.

V rámci hlavních elementů lze vysledovat tři skupiny: *Aktivní elementy*, *Vztahy* a *Bazény a dráhy*. Zásadní úlohu zde mají především Aktivní elementy a Vztahy, které slouží k modelování základní logiky procesů, zatímco Bazény a dráhy slouží především ke zlepšení orientace v modelech a k znázornění aktérů, kteří se procesů účastní.

#### **Aktivní elementy**

Pro uplatnění hierarchické abstrakce je nutné je rozdělit na *komplexní* a *elementární*. Komplexní aktivní element je zpravidla prezentován jako *Business process*, názvosloví používané standardy je patrné z obrázku č. 7.2.

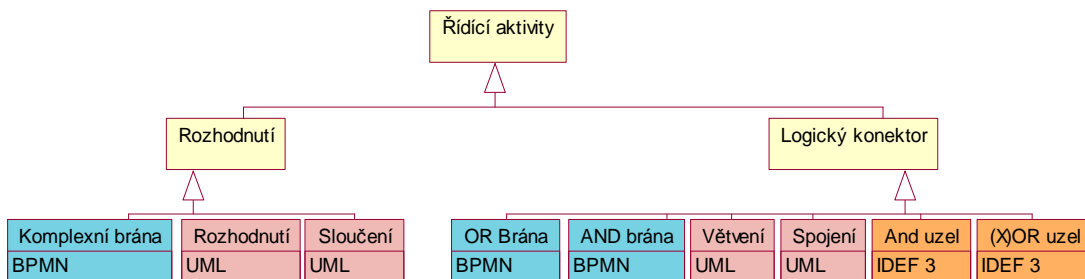


Obr. 7.2 - Komplexní aktivní element

*Elementární aktivní elementy* jsou atomické, dále nedělitelné, a je vhodné je rozlišit na *Události*, *Řídící činnosti* a *Výkonné aktivity*.

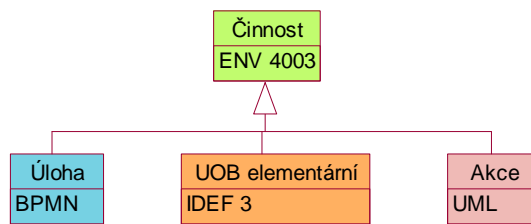
Událostí se rozumí informace o skutečnosti vzniklé mimo proces či podnět k činnosti. Způsob uplatnění událostí v jednotlivých notacích se liší, přesto je možné je rozčlenit podle klasifikace používané v BPMN na události typu *Start*, *Mezikrok*, *Konec*.

Řídícími aktivitami jsou body, ve kterých dochází k větvení procesu nebo slučování větví procesu do jednoho toku. Řídící aktivity typu *Rozhodnutí* zajišťují větvení a slučování na základě specifikovaných podmínek. *Logické konektory* pak provádějí větvení a synchronizaci paralelních procesů na základě AND a (X)OR logických operací. Členění řídicích aktivit je uvedeno na obrázku č. 7.3.



Obr. 7.3 – Řídící aktivity

*Činnost* představuje základní prvek procesu. Jedná se o atomickou aktivitu, která zpracovává vstupy na výstupy. Sémantika tohoto elementu je u všech notací stejná, liší se však v terminologii.



Obr. 7.4 - Činnost

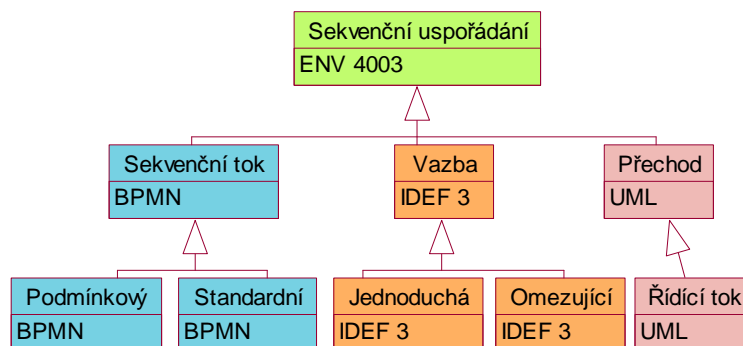
## Vztahy

Druhou základní skupinou objektů, které umožňují spojování aktivních prvků do logických posloupností (procesů) jsou *Vztahy*.

Nejjednodušším elementem pro vyjádření vztahu je *Asociace*, která slouží k připojení popisných elementů k hlavním. Používá se také k připojení *Datového objektu* a znázornění, že je tento objekt vyžadován, používán, produkován nebo má spojitost s některým z hlavních elementů zaměřených na popis chování.

*Objektový tok* je prvkem, kterým dochází k propojení obou částí metamodelu. Znázorňuje předání řízení mezi aktivními elementy, jehož součástí je nějaký objekt.

*Sekvenční uspořádání* je abstrakcí převzatou z ENV 4003 a použitou pro pojmenování typu vztahu, který spočívá v předání řízení jedním aktivním elementem elementu následujícímu v posloupnosti prvků v rámci procesu. Jedná se o vztah, který z principu musí implementovat všechny jazyky pro procesní modelování. Jakým způsobem jej implementují standardy popisované v této práci je znázorněno na obrázku 7.5.



Obr. 7.5 – Sekvenční uspořádání

### 7.2.2 Elementy zaměřené na popis systémové informace

Tato třída elementů je v podstatě převzata z metamodelu notace IDEF 3, která jediná ze tří popisovaných má v rámci přístupu k modelování procesů zpracování koncepti pro popis systémových objektů. Základními styčnými body s přístupem zaměřeným na chování systému jsou *Podnikový objekt* a *Datový objekt*. Podnikový objekt může být součástí Objektového toku, zatímco Datový objekt může být navíc pomocí Asociace připojen k některému z Hlavních elementů zaměřených na popis chování a vyjádřit tak, že je tímto elementem vyžadován, zpracováván, produkován či je s ním v jiném vztahu.

## 8 Závěr

Problematika analýzy a návrhu business procesů je rozsáhlou skupinou činností, která by nutně měla předcházet tvorbu informačního systému podniku. Přináší mnoho nových poznatků a požadavků, na základě kterých nové informační systémy vznikají nebo jsou stávající systémy upravovány. Takovéto informační systémy pak mají mnohem větší naději, že jejich koncepce a chování budou odrážet reálně fungující procesy, které ve firmě probíhají.

Smyslem tohoto textu bylo prozkoumat oblast modelování podnikových procesů ze dvou, ač to nemusí být na první pohled patrné, úzce propojených hledisek. Prvním je pohled mezinárodních standardizačních organizací a jejich tezí, které uplatňují vzhledem k podnikovému modelování. Druhé hledisko je představováno modelovacími jazyky, které představují převedení obecných modelovacích principů do podoby, která je použitelná v praxi.

Mezinárodní standardy modelování byly zkoumány v oddíle A této práce a čtenář mohl vidět, že jejich snahou je především vyzdvihnout aspekty, kterými se mají tvůrci modelovacích metodik zabývat. Definují např. úhly pohledu, ze kterých má být k podniku přistupováno a nesoustředí pouze na oblast Business Process Modellingu, ale věnují pozornost i ostatním aspektům podnikového modelování, jako je například modelování organizační struktury. Standardy tak vytvářejí sadu základních principů, kterými je třeba se při vytváření modelů řídit.

V oddíle B pak byly představeny standardy modelovacích jazyků a uskutečněn podrobný rozbor jejich elementů. Na základě tohoto rozboru byl pak vytvořen metamodel jazyka. Tento způsob popisu by měl čtenáři umožnit proniknout nejen do obecných vlastností, ale, zejména díky použití metamodelů, i do hlubších zákonitostí a vztahů mezi elementy jazyka.

K propojení obou základních hledisek dochází v sedmé kapitole, kde je pomocí souhrnného metamodelu demonstrováno jednak uplatnění obecných principů, popisovaných v mezinárodních standardech, jednak dochází ke sloučení dílčích

metamodelů na základě společných principů a vlastností elementů, obsažených v jednotlivých notacích. Použité abstraktní třídy pak umožňují porovnání elementů a terminologie používaných v jednotlivých notacích. Zejména tato kapitola by pak měla sloužit jako základ pro další práci a uplatnění v projektu OpenSoul.

## 9 Seznam použité literatury

- [BPML01] Business Process Modeling Language, BPMI, 2002  
<http://www.bpmi.org>
- [BPMN01] Business Process Modeling Notation Working Draft 1.0, BPMI, 2002  
<http://www.bpmi.org>
- [CHAD01] Zdeněk Chadima, Zmapování existujících standardů a metodik v oblasti modelování business procesů, bakalářská práce, VŠE Praha, 2004
- [CHEN01] David Chen, Francois Vernadat, Standardisation on Enterprise Integration & Engineering /Achievements, On-going Works and Future Perspectives, konference INCOM'01/GRP GT 5, 2002  
[http://www.univ-valenciennes.fr/GDR-MACS/local/Grenoble/actes-journees/ME/david\\_chen.pdf](http://www.univ-valenciennes.fr/GDR-MACS/local/Grenoble/actes-journees/ME/david_chen.pdf)
- [ENV01] ENV 12204 Advanced Manufacturing Technology Systems Architecture Constructs for Enterprise Modelling, CEN TC310 WG1, 1995
- [HRE01] Petr Hřebejk, Václav Řepa: Meta-modelování. Příspěvek na konferenci DataSem 99.  
<http://nb.vse.cz/~repa/veda/RaD.htm>
- [IDEF01] IDEF3 Process Description Capture Method Report, ICE, 1995 -  
[http://www.idef.com/Downloads/pdf/Idef3\\_fn.pdf](http://www.idef.com/Downloads/pdf/Idef3_fn.pdf)
- [IDEF02] <http://www.idef.com/idef0.html>
- [IDEF03] <http://www.idef.com/idef5.html>
- [ISO01] Industrial automation systems--Concepts and rules for enterprise models, ISO, 1997 - <http://www.mel.nist.gov/sc5wg1/std-dft.htm>
- [NORAN01] Ovidiu S. Noran, Business modeling: UML vs. IDEF, Griffith University  
<http://www.cit.gu.edu.au/~noran/Docs/UMLvsIDEF.pdf>
- [REP01] Václav Řepa, Analýza a návrh informačních systémů, Ekopress, 1999, ISBN 80-86119-13-0
- [REP02] Václav Řepa, Podnikové procesy; Procesní řízení a modelování, Grada Publishing, 2006, ISBN 80-247-1281-4
- [UML01] Unified Modeling Language: Infrastructure, version 2.0, OMG, 2005  
<http://www.omg.org/docs/formal/05-07-05.pdf>

- [UML02] Unified Modeling Language: Superstructure, version 2.0, OMG, 2004  
<http://www.omg.org/docs/formal/05-07-04.pdf>
- [UML03] OMG Unified Modeling Language Specification, OMG, 2003  
<http://www.omg.org/docs/formal/03-03-01.pdf>
- [WHI01] Stephen White, Process modelling notations and workflow patterns, IBM corp.  
<http://www.bpmn.org/Documents/Notations%20and%20Workflow%20Patterns.pdf>
- [WHI02] Introduction to BPMN, Stephen White, IBM corporation  
<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>

## 10 Terminologický slovník

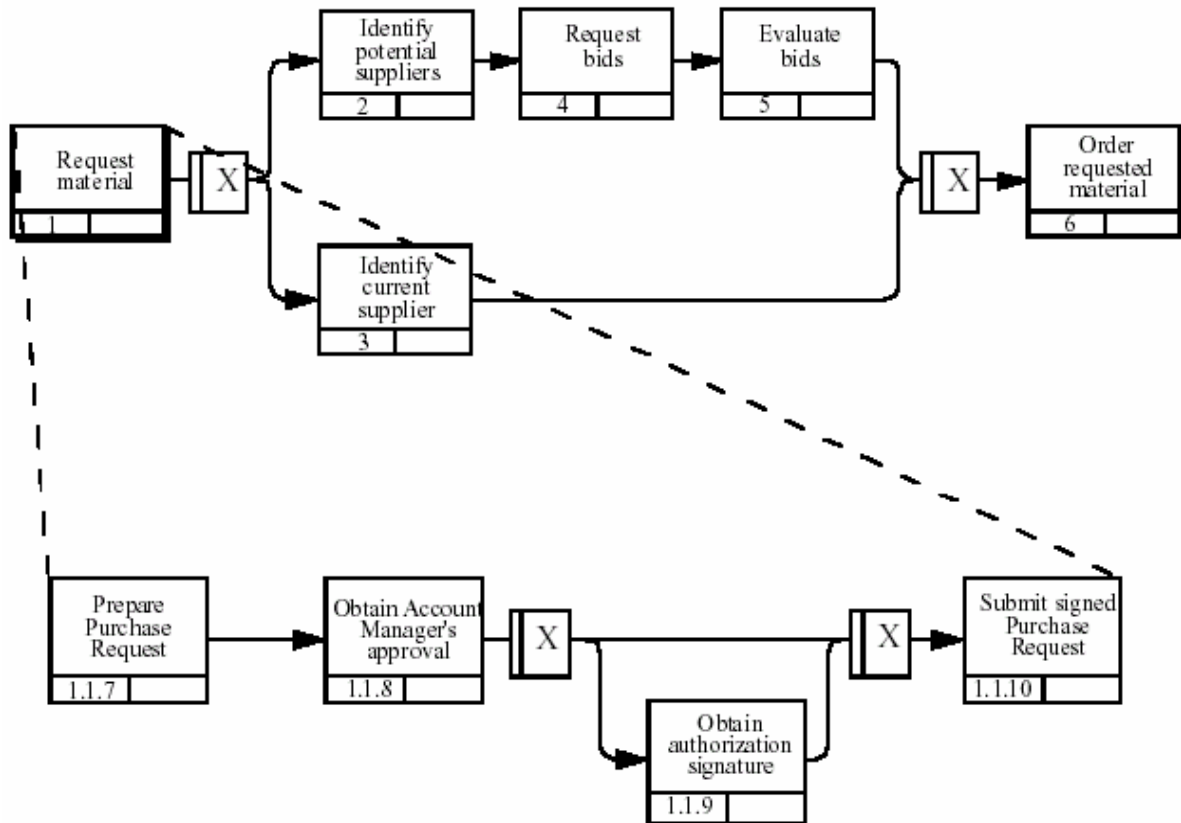
Termín	Význam
<b>Termíny přejaté z terminologického slovníku katedry informačních technologií</b>	
B2B	Business-to-business – spolupráce podniků založená na elektronické výměně dat
Extensible Markup Language	Značkovací jazyk obsahující příkazy definující syntax (strukturu) dokumentu, definovaný doporučením W3C (WWW Consortium). Navazuje na něj řada dalších standardů/technologií např. DTD (Document Type Definition), XSTL (XSL Transformations), SOAP (Simple Object Access Protocol).
Model	Dílčí pohled na vytvářený systém, souhrn všech pohledů na celý systém
UML	→ Unified Modelling Language
Unified Modelling Language	Grafická notace pro zápis modelů informačních systémů, shrnuje a standardizuje dosavadní notace a je východiskem pro vznik nových. Je určen pro zápis jednotlivých pohledů na systém v objektově orientovaných metodikách, používá se například v CASE určených pro podporu tvorby informačních systémů, jeho použití však může být širší (reorganizace podniku).
XML	→ Extensible Markup Language
<b>Další termíny</b>	
BPML	→ Business Process Modeling Language
BPEL4WS	→ Business Process Execution Language for Web Services
BPD	→ Business Process Diagram
BPI	→ Business Process Improvement
BPMN	→ Business Process Modeling Notation
BPR	→ Business Process Reengineering
Business process Diagram	Základní diagram notace → BPMN
Business Process Execution Language for Web Services	Exekutivní jazyk pro modelování podnikových procesů, které jsou uskutečňovány pomocí webových služeb.
Business Process Improvement	Způsob zlepšování podnikových procesů, který je založen na jejich postupné změně.
Business Process Modeling Language	Exekutivní jazyk pro modelování podnikových procesů založený na → XML
Business Process Modeling Notation	Grafická notace konsorcia BPMI pro modelování podnikových procesů.
Business Process Reengineering	Způsob zlepšování podnikových procesů jejich radikální změnou či výstavbou „na zelené louce“.

Termín	Význam
CEN	European Comitee for Standardisation
CIMOSA	→ Computer Integrated Manufacturing Open System Architecture
Computer Integrated Manufacturing Open System Architecture	Metodologie tvorby podnikové architektury
ENV 12204	Advanced Manufacturing Technology Systems Architecture Constructs for Enterprise Modelling – mezinárodní standard pro podnikové modelování
ENV 40003	CIM systems architecture framework for modeling – mezinárodní standard pro podnikové modelování
Generalised Enterprise Reference Architecture and Methodology	Metodologie pro tvorbu podnikové architektury.
GERAM	→Generalised Enterprise Reference Architecture and Methodology
GRAI/GIM	Metodologie tvorby podnikové architektury
ICAM	→Integrated Computer-Aided Manufacturing program
ICOM	Základní stavební kámen funkčních modelů podle metody IDEF0, představující aktivitu. ICOM je zkratka počátečních písmen slov Input, Control, Output, Mechanism
IDEF	Integrated DEFinition metodologie metodologie je souborem či rodinou metod, jejichž účelem je komplexně podporovat potřeby podniků v oblasti modelování podnikové architektury. Metodika je produktem výzkumného programu → ICAM.
IDEF 3	Metoda z rodin metod → IDEF, která se zaměřuje na modelování podnikových procesů.
Integrated Computer-Aided Manufacturing program	Výzkumný projekt letectva USA, vytvořený za účelem zlepšení koordinace výrobních operací
ISO 14258	Industrial automation system – Concepts and rules for enterprise modelling – mezinárodní standard pro modelování podnikových procesů
konstrukt	Textový či grafický artefakt vytvořený za účelem jednotné reprezentace různorodých informací o vlastnostech a elementech určitého souboru jevů.
MDA	→ Model Driven Architecture
Meta-metamodel	Meta-metamodel je popis jasně specifikovaných pravidel, které vymezují formální definice pro tvorbu metamodelů. Vztah mezi meta-metamodelem a metamodelem je analogický vztahu mezi metamodelem a modelem, jedná se pouze o vyšší stupeň abstrakce.
metamodel	Metamodel je formální definice určitého modelovacího jazyka.
Meta Object Facility	Specifikace meta-metamodelu konsorcia OMG, které jej popisuje jako „rozšiřitelný modelem řízený integrační rámec pro definování, manipulování a integrování metadat a dat na platformně nezávislém základu.“ (www.omg.com)
Model Driven Architecture	Model Driven Architecture je přístup k vývoji softwarových systémů

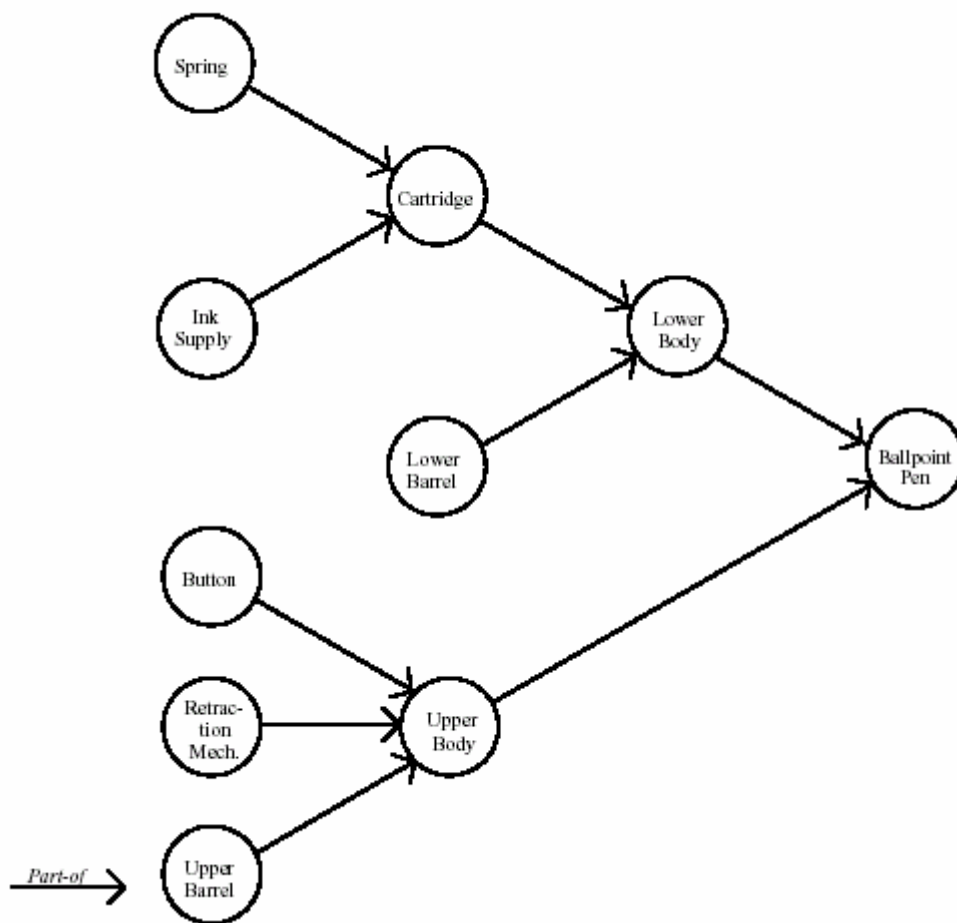
Termín	Význam
	podporovaný konsorciem OMG. Funkcionalita systému je nejprve specifikována pomocí platformně nezávislého modelu a pro konkrétní implementaci jsou nezávislé modely přeloženy do platformně specifických modelů. Na základě specifických modelů je pak tvořen programový kód.
MOF	→ Meta Object Facility
Ontologie	Ontologií je doménový slovník spolu se sadou precizních definicí nebo axiomů, které vymezují význam termínů natolik, aby to umožňovalo jednotnou interpretaci dat
podniková architektura	podniková architektura je modelem, který popisuje podnik v určitém stádiu jeho životního cyklu. Bývá představována soustavou modelů, které umožňují plánování a analýzu vývoje podniku
SADT grafický modelovací jazyk	→ Structured analysis and design technique
Service Oriented Architecture	SOA je abstraktní koncept, který určuje jak propojit a integrovat aplikace dohromady a který staví na technologiích implementovaných v → XML a webových službách.
Structured analysis and design technique	Grafický modelovací jazyk
scénář	Scénář je termín, který metoda → IDEF 3 používá pro základní organizační strukturu modelu procesů. Scénářem je opakující se situace nebo množina situací, která popisuje typickou třídu problémů, nebo představuje uspořádání z něhož vyvstává proces.
SOA	→ Service Oriented Architecture
transakce	Prvek jazyka BPML, který umožňuje dvěma procesům, které spolu komunikují pomocí zpráv, koordinovat dokončení jejich aktivit.
UOB	→ Unit Of Behavior
Unit Of Behavior	Je elementem jazyka IDEF 3 a představuje obecný typ situace, který se v systému vyskytuje. Takovou situací může být například „výdej materiálu“. UOB však nepředstavuje konkrétní výskyt situace, ale je její abstrakcí, tedy představuje třídu situací „výdej materiálu“.
UOB box	Symbol notace IDEF3 pro → UOB
user object	Element jazyka UML, který představuje definované uživatelské objekty, které se vyskytují v modelovaném prostředí.

# **Přílohy**

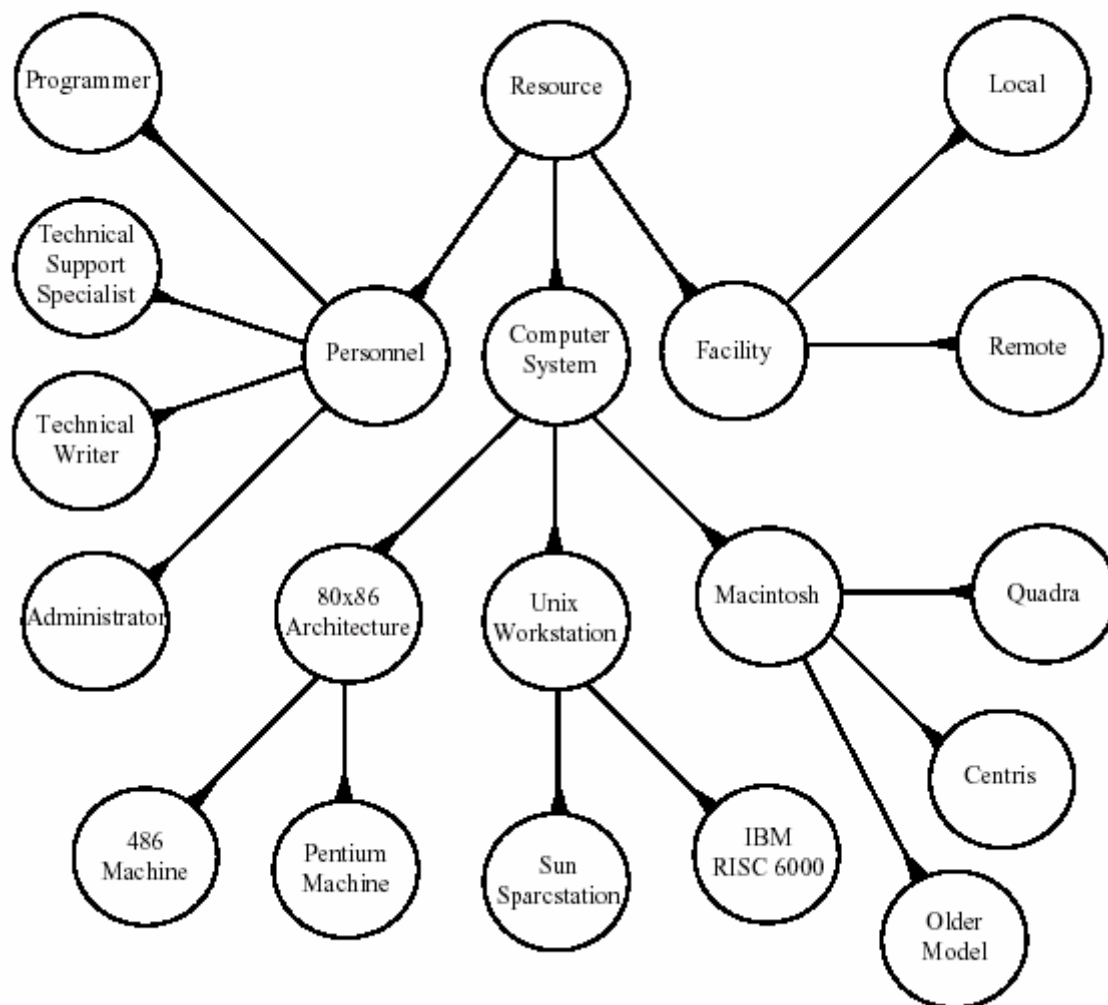
**Příloha č. 1 - příklad procesně orientovaného modelu dle IDEF3 [IDEF01]**



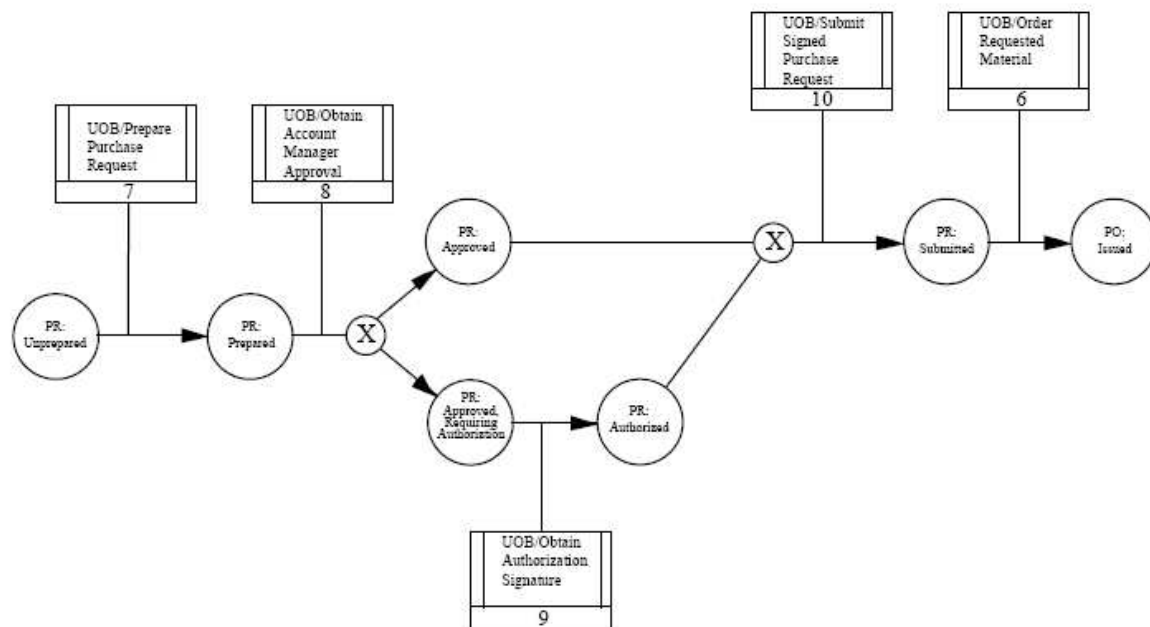
**Příloha č. 2 - kompoziční diagram dle IDEF3 [IDEF01]**
































**Příloha č. 3 - klasifikační diagram dle IDEF3 [IDEF01]**




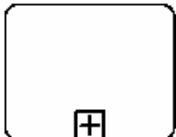
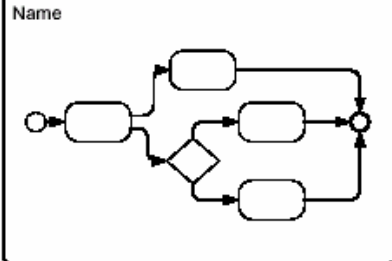







**Příloha č. 4 - příklad modelu zaměřeného na objekty dle IDEF3 [IDEF01]**



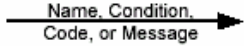
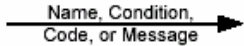
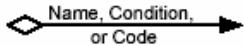
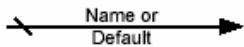
**Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]**

Element	Description	Notation
Event	<p>An event is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.</p>	
<p>Flow Dimension (e.g., Start, Intermediate, End)</p> <p>Start (None, Message, Timer, Rule, Link, Multiple)</p> <p>Intermediate (None, Message, Timer, Exception, Cancel, Compensation, Rule, Link, Multiple, Branching)</p> <p>End (None, Message, Exception, Cancel, Compensation, Link, Terminate, Multiple)</p>	<p>As the name implies, the Start Event indicates where a particular process will start.</p> <p>Intermediate Events occur between a Start Event and an End Event. It will affect the flow of the process, but will not start or (directly) terminate the process.</p> <p>As the name implies, the End Event indicates where a process will end.</p>	<p>Start </p> <p>Intermediate </p> <p>End </p>
<p>Type Dimension (e.g., Message, Timer, Exception, Cancel, Compensation, Rule, Link, Multiple, Terminate)</p>	<p>Start and Intermediate Events have “Triggers” that define the cause for the event. There are multiple ways that these events can be triggered. End Events may define a “Result” that is a consequence of a Sequence Flow ending.</p>	<p><b>Message</b>   </p> <p><b>Timer</b>   </p> <p><b>Exception</b>   </p> <p><b>Cancel</b>   </p> <p><b>Compensation</b>   </p> <p><b>Rule</b>   </p> <p><b>Link</b>   </p> <p><b>Multiple</b>   </p> <p><b>Terminate</b> </p>

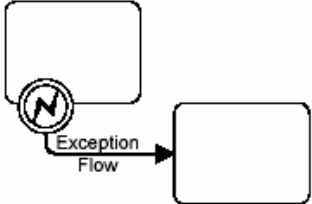
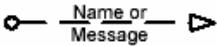
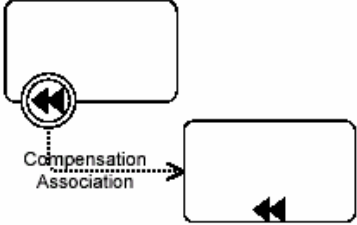

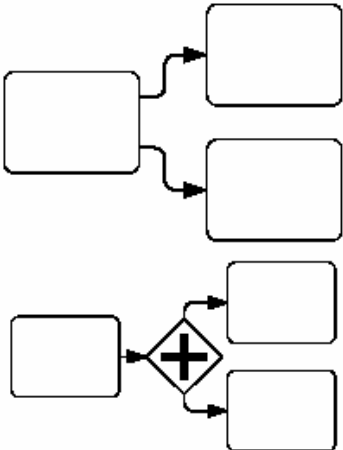
**Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]**

<p>Task (Atomic)</p>	<p>A Task is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail.</p>	
<p>Process/Sub-Process (non-atomic)</p>	<p>A Sub-Process is a compound activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail (a Process) through a set of sub-activities.</p>	<p>See Next Two Figures</p>
<p>Collapsed Sub-Process</p>	<p>The details of the Sub-Process are not visible in the Diagram. A "plus" sign in the lower-center of the shape indicates that the activity is a Sub-Process and has a lower-level of detail.</p>	
<p>Expanded Sub-Process</p>	<p>The boundary of the Sub-Process is expanded and the details (a Process) are visible within its boundary. Note that Sequence Flow cannot cross the boundary of a Sub-Process.</p>	<p>Name</p> 
<p>Gateway</p>	<p>A Gateway is used to control the divergence and convergence of multiple Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths.</p>	
<p>Gateway Control Types</p>	<p>Icons within the diamond shape will indicate the type of flow control behavior. The types of control include:</p> <ul style="list-style-type: none"> <li>• XOR -- exclusive decision and merging. Both Data-Based and Event-Based. Data-Based can be shown with or without the "X" marker.</li> <li>• OR -- inclusive decision</li> <li>• Complex -- complex conditions and situations (e.g., 3 out of 5)</li> <li>• AND -- forking and joining</li> </ul> <p>Each type of control affects both the incoming and outgoing Flow.</p>	<p><b>Exclusive (XOR)</b>  <b>Data-Based</b>  or   <b>Event-Based</b>   <b>Inclusive (OR)</b>   <b>Complex</b>   <b>Parallel (AND)</b> </p>

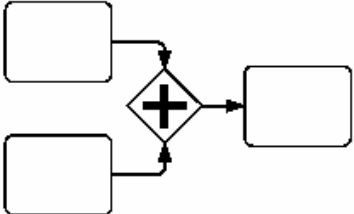
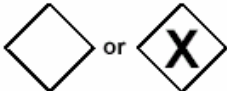

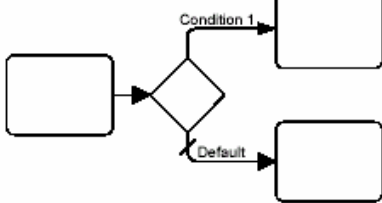
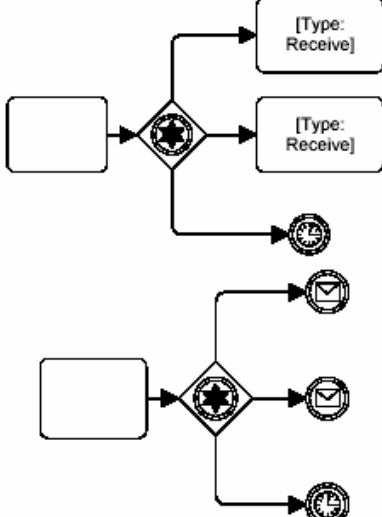
## Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]

Sequence Flow	A Sequence Flow is used to show the order that activities will be performed in a Process.	See next seven figures
Normal flow	Normal Sequence Flow refers to the flow that originates from a Start Event and continues through activities via alternative and parallel paths until it ends at an End Event.	
Uncontrolled flow	Uncontrolled flow refers to flow that is not affected by any conditions or does not pass through a Gateway. The simplest example of this is a single Sequence Flow connecting two activities. This can also apply to multiple Sequence Flow that converge on or diverge from an activity. For each uncontrolled Sequence Flow a "Token" will flow from the source object to the target object.	
Conditional flow	Sequence Flow can have condition expressions that are evaluated at runtime to determine whether or not the flow will be used. If the conditional flow is outgoing from an activity, then the Sequence Flow will have a mini-diamond at the beginning of the line (see figure to the right). If the conditional flow is outgoing from a Gateway, then the line will not have a mini-diamond (see figure in the row above).	
Default flow	For Data-Based Exclusive Decisions, one type of flow is the Default condition flow. This flow will be used only if all the other outgoing conditional flow is not true at runtime. These Sequence Flow will have a diagonal slash will be added to the beginning of the line (see the figure to the right). Note that it is an Open Issue whether Default Conditions will be used for Inclusive Decision situations.	

## Příloha č. 6 – přehled symbolů užívaných BPMN [BPMN01]

<p>Exception flow</p>	<p>Exception flow occurs outside the normal flow of the Process and is based upon an Intermediate Event that occurs during the performance of the Process.</p>	
<p>Message Flow</p>	<p>A Message Flow is used to show the flow of messages between two entities that are prepared to send and receive them. In BPMN, two separate Pools in the Diagram will represent the two entities.</p>	
<p>Compensation Association</p>	<p>Compensation Association occurs outside the normal flow of the Process and is based upon an event (a Cancel Intermediate Event) that is triggered through the failure of a Transaction or a Compensate Event. The target of the Association must be marked as a Compensation Activity.</p>	
<p>Data Object</p>	<p>Data Objects are considered artifacts because they do not have any direct affect on the Sequence Flow or Message Flow of the Process, but they do provide information about what the Process does.</p>	
<p>Fork (AND-Split)</p>	<p>BPMN uses the term "fork" to refer to the dividing of a path into two or more parallel paths (also known as an AND-Split). It is a place in the Process where activities can be performed concurrently, rather than serially.</p> <p>There are two options:</p> <p>Multiple Outgoing Sequence Flow can be used (see figure top-right). This represents "uncontrolled" flow is the preferred method for most situations.</p> <p>A Parallel (AND) Gateway can be used (see figure bottom-right). This will be used rarely, usually in combination with other Gateways.</p>	

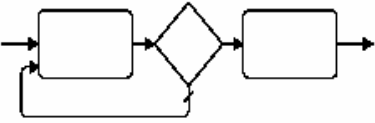


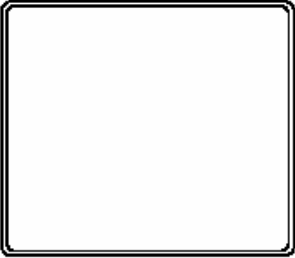
**Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]**

<p>Join (AND-Join)</p>	<p>BPMN uses the term "join" to refer to the combining of two or more parallel paths into one path (also known as an AND-Join or synchronization). A Parallel (AND) Gateway is used to show the joining of multiple flows.</p>	
<p>Decision, Branching Point; (OR-Split)</p>	<p>Decisions are Gateways within a business process where the flow of control can take one or more alternative paths.</p>	<p>See next five rows.</p>
<p>Exclusive</p>	<p>An Exclusive Gateway (XOR) restricts the flow such that only one of a set of alternatives may be chosen during runtime. There are two types of Exclusive Gateways: Data-based and Event-based.</p>	<p>Data-Based  or </p> <p>Event-Based</p>
<p>Data-Based</p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow. Only one of the Alternatives will be chosen.</p>	
<p>Event-Based</p>	<p>This Decision represents a branching point where Alternatives are based on an Event that occurs at that point in the Process. The specific Event, usually the receipt of a Message, determines which of the paths will be taken. Other types of Events can be used, such as Timer. Only one of the Alternatives will be chosen.</p> <p>There are two options for receiving Messages: Tasks of Type Receive can be used (see figure top-right). Intermediate Events of Type Message can be used (see figure bottom-right).</p>	



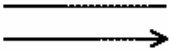


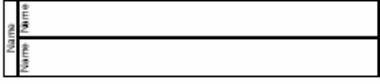
**Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]**

<p>Inclusive</p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow.</p> <p>In some sense it is a grouping of related independent Binary (Yes/No) Decisions. Since each path is independent, all combinations of the paths may be taken, from zero to all. However, it should be designed so that at least one path is taken.</p> <p>There are two versions of this type of Decision:</p> <p>The first uses a collection of conditional Sequence Flow, marked with mini-diamonds (see top-right figure).</p> <p>The second uses an OR Gateway, usually in combination with other Gateways (see bottom-right picture).</p>	
<p>Merging (OR-Join)</p>	<p>BPMN uses the term "merge" to refer to the exclusive combining of two or more paths into one path (also known as an a OR-Join).</p> <p>A Merging (XOR) Gateway is used to show the merging of multiple flows.</p> <p>If all the incoming flow is alternative, then a Gateway is not needed. That is, uncontrolled flow provides the same behavior.</p>	
<p>Looping</p>	<p>BPMN provides 2 (two) mechanisms for looping within a Process.</p>	<p>See Next Two Figures</p>
<p>Activity Looping</p>	<p>The properties of Tasks and Sub-Processes will determine if they are repeated or performed once. There are two types of loops: Standard and Multi-Instance. A small looping indicator will be displayed at the bottom-center of the activity.</p>	







## Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]

Sequence Flow Looping	Loops can be created by connecting a Sequence Flow to an “upstream” object. An object is considered to be upstream if that object has an outgoing Sequence Flow that leads to a series of other Sequence Flows, the last of which is an incoming Sequence Flow for the original object.	
Multiple Instances	The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. A small parallel indicator will be displayed at the bottom-center of the activity.	
Process Break (something out of the control of the process makes the process pause)	A Process Break is a location in the Process that shows where an expected delay will occur within a Process. An Intermediate Event is used to show the actual behavior (see top-right figure). In addition, a Process Break artifact, as designed by a modeler or modeling tool, can be associated with the Event to highlight the location of the delay within the flow.	
Transaction	A transaction is an activity, either a Task or a Sub-Process, that is supported by special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled. The attributes of the activity will determine if the activity is a transaction. A double-lined boundary indicates that the activity is a Transaction.	
Nested Sub-Process (Inline Block)	A nested Sub-Process is an activity that shares the same set of data as its parent process. This is opposed to a Sub-Process that is independent, re-usable, and referenced from the parent process. Data needs to be passed to the referenced Sub-Process, but not to the nested Sub-Process.	There is no special indicator for nested Sub-Processes









## Příloha č. 5 – přehled symbolů užívaných BPMN [BPMN01]

<p>Group (a box around a group of objects for documentation purposes)</p>	<p>A grouping of activities that does not affect the Sequence Flow. The grouping can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools.</p>	
<p>Off-Page Connector</p>	<p>Generally used for printing, this object will show where the Sequence Flow leaves one page and then restarts on the next page. A Link Intermediate Event can be used as an Off-Page Connector.</p>	
<p>Association</p>	<p>An Association is used to associate information with flow objects. Text and graphical non-flow objects can be associated with the flow objects.</p>	
<p>Text Annotation (attached with an Association)</p>	<p>Text Annotations are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.</p>	
<p>Pool</p>	<p>A Pool is a "swimlane" and a graphical container for partitioning a set of activities from other Pools, usually in the context of B2B situations.</p>	
<p>Lanes</p>	<p>A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities within a Pool.</p>	










**Příloha č. 6 - Přehled symbolů užívaných pro Start události [BPMN01]**

Trigger	Description	Marker
None	The modeler does not display the type of Event. It is also used for a Sub-Process that starts when the flow is triggered by its Parent Process.	
Message	A message arrives from a participant and triggers the start of the Process.	
Timer	A specific time-date or a specific cycle (e.g., every Monday at 9am) can be set that will trigger the start of the Process.	
Rule	This type of event is triggered when the conditions for a rule such as "S&P 500 changes by more than 10% since opening," or "Temperature above 300C" become true.	
Link	A Link is a mechanism for connecting the end (Result) of one Process to the start (Trigger) of another. Typically, these are two Sub-Processes within the same parent Process.	
Multiple	This means that there are multiple ways of triggering the Process. Only one of them will be required to start the Process. The attributes of the Start Event will define which of the other types of Triggers apply.	

**Příloha č. 7 - Přehled symbolů užívaných pro Konec události [BPMN01]**

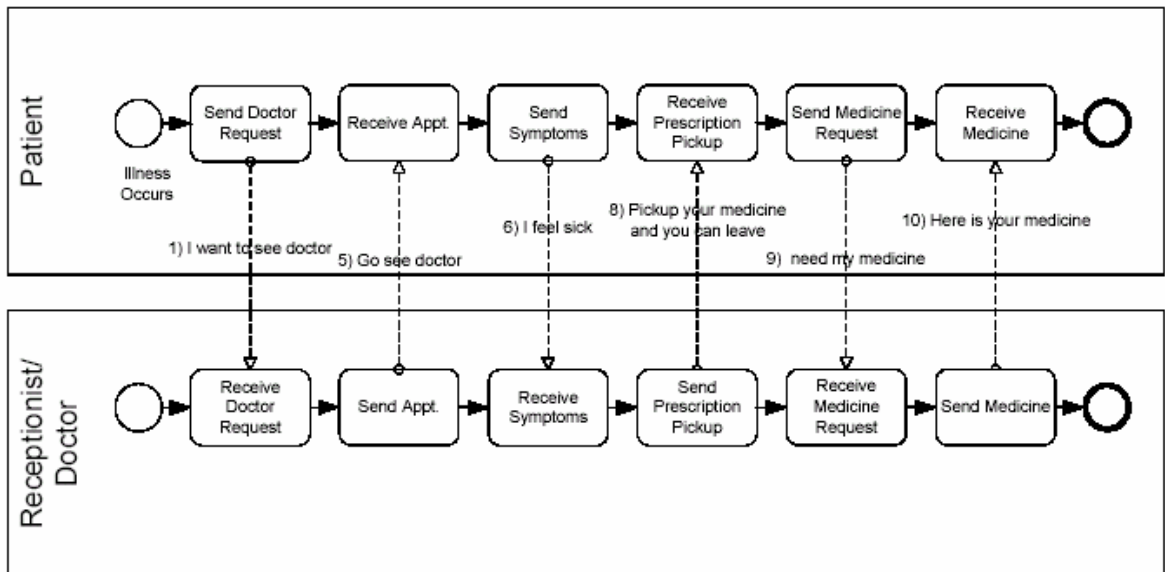
Result	Description	Marker
None	The modeler does not display the type of Event. It is also used for a Sub-Process that end and the flow goes back to its Parent Process.	
Message	This type of End indicates that a message is sent to a participant at the conclusion of the Process.	
Exception	This type of End indicates that a named Error should be generated. This Error will be caught by an Intermediate Event within the Event Context.	
Cancel	This type of End is used within a Transaction Sub-Process. It will indicate that the Transaction should be cancelled and will trigger a Cancel Intermediate Event attached to the Sub-Process boundary. In addition, it will indicate that a Transaction Protocol Cancel message should be sent to any Entities involved in the Transaction.	
Compensation	This type of End will indicate that a Compensation is necessary. This Compensation identifier will be used by an Intermediate Event when the Process is rolling back.	
Link	A Link is a mechanism for connecting the end (Result) of one Process to the start (Trigger) of another. Typically, these are two Sub-Processes within the same parent Process.	
Terminate	This type of End indicates that there is a fatal error and that all activities in the Process should be immediately ended. The Process is ended without compensation or event handling. Note that the marker for this Event is an Open Issue.	
Multiple	This means that there are multiple consequences of ending the Process. All of them will occur (e.g., there might be multiple messages sent). The attributes of the End Event will define which of the other types of Results apply.	

**Příloha č. 8 - Přehled symbolů užívaných pro události Mezikrok [BPMN01]**

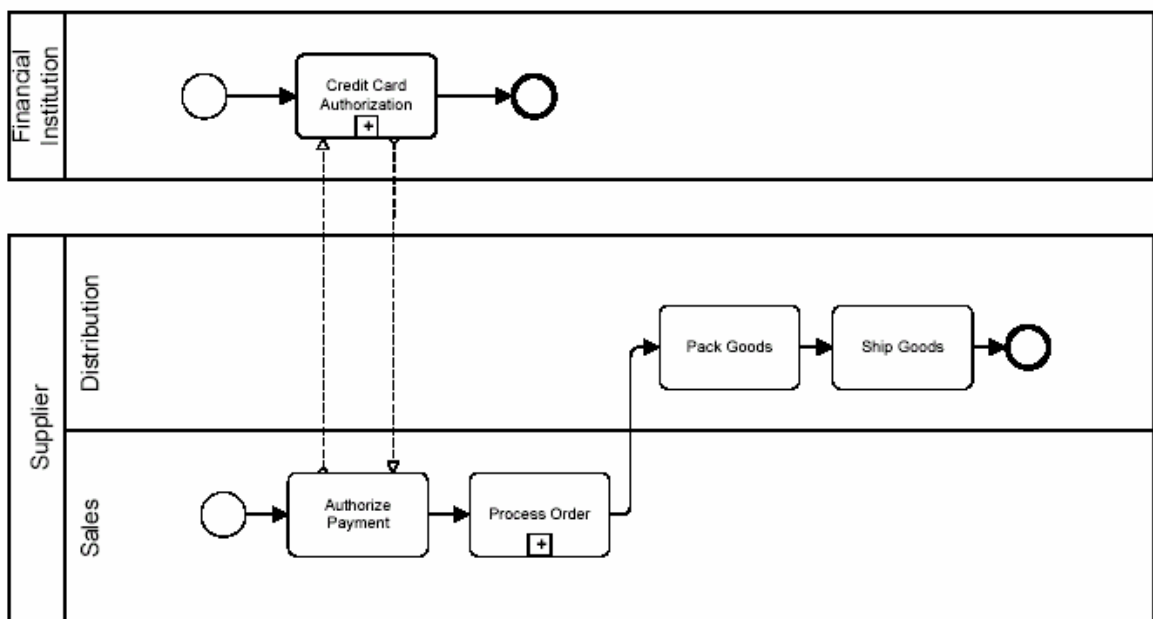
Trigger	Description	Marker
None	This is valid for only Intermediate Events that are in the main flow of the Process. The modeler does not display the type of Event. It is used for modeling methodologies that use Events to indicate some change of state in the Process.	
Message	A message arrives from a participant and triggers the Event. This causes the Process to continue if it was waiting for the message, or changes the flow for exception handling.	
Timer	A specific time-date or a specific cycle (e.g., every Monday at 9am) can be set that will trigger the Event. If used within the main flow it acts as a delay mechanism. If used for exception handling it will change the normal flow into an exception flow.	
Exception	This is used for exception handling--both to set (throw) and to react to (catch) exceptions. It sets an exception if the Event is part of a normal flow. It reacts to a named exception, or to any exception if a name is not specified, when attached to the boundary of an activity.	
Cancel	This type of Intermediate Event is used within a Transaction Sub-Process. This type of Event MUST be attached to the boundary of a Sub-Process. It SHALL be triggered if a Cancel End Event is reached within the Transaction Sub-Process. It also SHALL be triggered if a Transaction Protocol "Cancel" message has been received while the Transaction is being performed.	
Compensation	This is used for compensation handling--both setting and performing compensation. It call for compensation if the Event is part of a normal flow. It reacts to a named compensation call when attached to the boundary of an activity.	
Rule	This is only used for exception handling. This type of event is triggered when a named Rule becomes true. A Rule is an expression that evaluates some Process data.	
Link	A Link is a mechanism for connecting the end (Result) of one Process to the start (Trigger) of Event-Based Exclusive Decision.	
Multiple	This means that there are multiple ways of triggering the Event. Only one of them will be required. The attributes of the Intermediate Event will define which of the other types of Triggers apply.	

## Příloha č. 9 - Příklady použití bazénů a drah v BPMN [BPMN01]

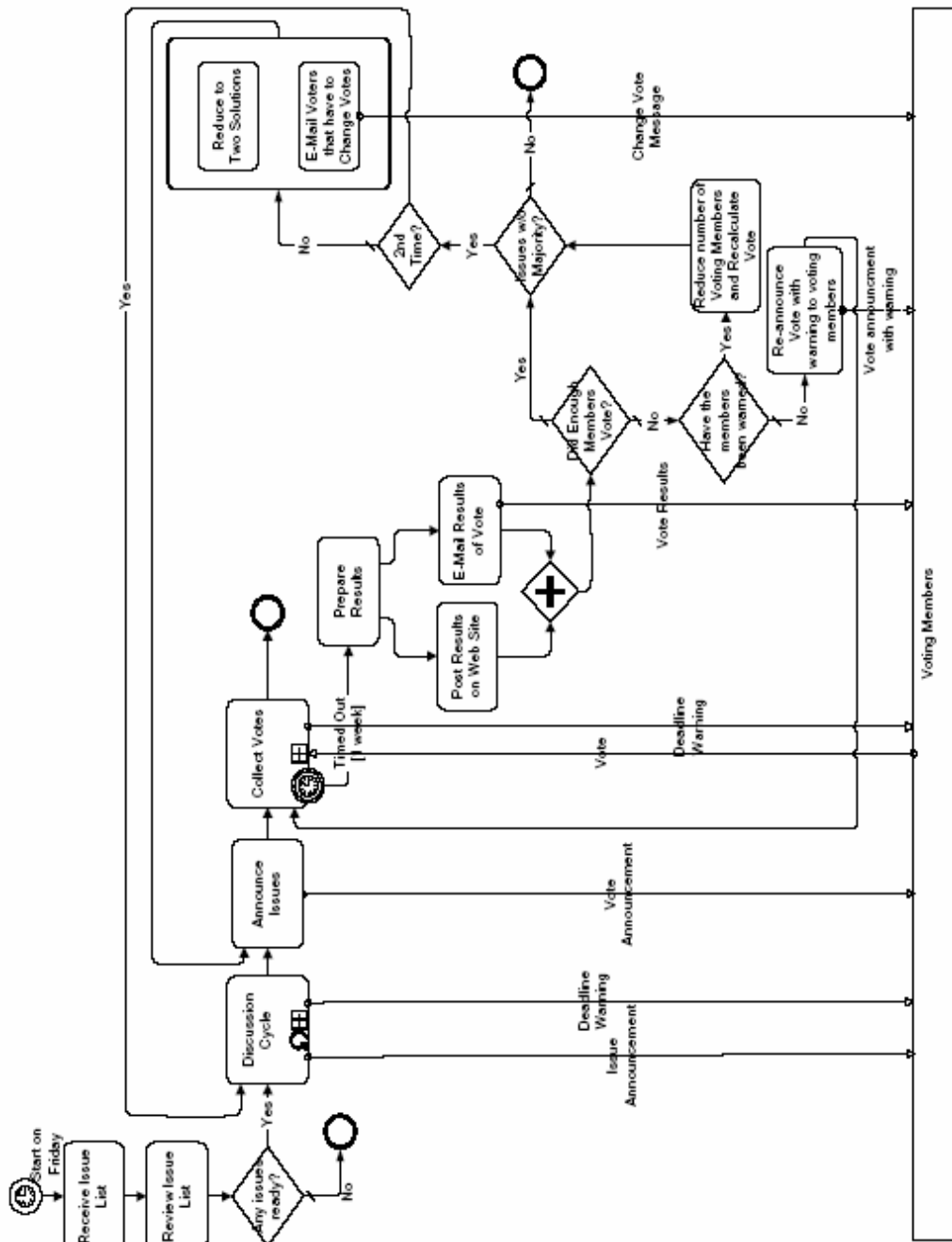
### Příklad 1 - Bazény



### Příklad 2 – Bazény a dráhy



**Příloha č. 10 Příklad modelu procesu hlasování pomocí e-mailu v notaci BPMN [BPMN01]**



Příloha č. 11 Příklad použití Drah v UML [UML02]

