

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Diplomová práce**

**Analýza a návrh informačního systému**

**Miloš Rajdl**

© 2012 ČZU v Praze

### Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Analýza a návrh informačního systému" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 5.4.2012

---

### Poděkování

Rád bych touto cestou poděkoval vedoucímu mé diplomové práce Ing. Miloši Ulmanovi, Ph.D. za jeho odborné vedení a cenné rady, které mi při tvorbě této práce velice pomohly.

# Analýza a návrh informačního systému

---

## Analysis and design of the information system

### Souhrn

Tato diplomová práce se zabývá analýzou zvoleného podnikového procesu a návrhem informačního systému pro zefektivnění uvedeného procesu. V teoretické části práce jsou nejprve představeny základní pojmy z oblasti podnikových procesů a jejich zlepšování. Následuje charakteristika a porovnání nejpoužívanějších grafických notací pro formální zápis procesů. Další část přináší bližší pohled na vývoj přístupů k analýze a návrhu informačních systémů od vzniku klasických strukturovaných technik, přes přechod k objektově orientovanému přístupu, až po vznik standardizovaného jazyka UML, kterému je věnována samostatná kapitola.

Úvod praktické části je vyčleněn pro představení zvoleného procesu v kontextu procesů daného pracoviště, aby mohl být dále podrobněji analyzován a identifikovány jeho problematické vlastnosti. Na základě těchto vlastností je následně navržena nová podoba procesu a formulovány požadavky na informační systém, který by průběh procesu s navrženými změnami realizoval. Požadavky na systém jsou vyjádřeny prostřednictvím případů užití, pro jejichž bližší specifikaci byly využity příslušné diagramy jazyka UML. Závěr praktické části obsahuje návrh statické struktury systému v podobě UML diagramu tříd.

### Summary

This thesis deals with an analysis of selected business process and with a design of an information system for make the process more efficient. At first there are basic concepts of business processes and their improvement introduced in a theoretical part of the thesis. Then characterization and comparison of the most widely used graphic notation for formal description of processes follows. Next section provides a closer view of the evolution of approaches for analysis and design of information systems from the traditional structured techniques, through the transition to object-oriented approach to the inception of a standardized UML, which is devoted a separate chapter.

Introduction of the practical part of this thesis is devoted to show the selected process in the context of other processes that apply in a given department in order to be analyzed in details and for its problematic attributes to be identified. Based on these attributes a new form of the process is designed and requirements for the information system that would implement the new process are formulated. System requirements are expressed through the use cases for which detailed specification were used appropriate UML diagrams. Conclusion of the practical part contains a design of static structure of the system in the form of UML class diagram.

**Klíčová slova:** UML, analýza, návrh, podnikový proces, informační systém, BPMN, případ užití

**Keywords:** UML, analysis, design, business process, information system, BPMN, use case

# Obsah

1 – Úvod .....	8
2 – Cíl práce a metodika .....	9
3 – Podnikové procesy .....	10
3.1 – Definice základních pojmů .....	10
3.1.1 – Podnikový proces .....	11
3.1.2 – Workflow .....	12
3.1.3 – Aktivita .....	13
3.1.4 – Instance .....	13
3.2 – Funkční vs. procesní řízení .....	13
3.3 – Klasifikace procesů .....	14
3.4 – Zlepšování procesů .....	16
3.4.1 – Průběžné zlepšování procesů .....	16
3.4.2 – Business process reengineering .....	17
3.4.3 – Business process management .....	17
3.4.4 – Shrnutí .....	19
3.5 – Analýza a modelování podnikových procesů .....	19
3.5.1 – Analýza procesů .....	20
3.5.2 – Analýza procesů dle metodiky MMABP .....	21
3.5.3 – Další metodiky a techniky analýzy podnikových procesů .....	22
3.5.4 – Notace pro modelování procesů .....	24
4 – Metody a techniky analýzy a návrhu IS .....	32
4.1 – Vývoj metod analýzy a návrhu IS .....	32
4.1.1 – Strukturované techniky .....	32
4.1.2 – Objektově orientované techniky .....	36
4.2 – UML (Unified Modelling Language) .....	41
4.2.1 – Mechanismy UML .....	41
4.2.2 – UML a architektura modelovaného systému .....	43
4.2.3 – Diagramy UML .....	44
5 – Analýza a návrh informačního systému .....	48
5.1 – Agenda Úseku správy příspěvků .....	48
5.1.1 – Procesy v Úseku správy příspěvků .....	50
5.1.2 – Typy plateb .....	51
5.1.3 – Typy příspěvků .....	52
5.1.4 – Rozpisy hromadných plateb .....	52
5.1.5 – Příklad rozpisu .....	52
5.2 – Specifikace problému .....	53
5.3 – Proces zpracování rozpisů .....	54
5.3.1 – Slovní charakteristika .....	54
5.3.2 – Model procesu v notaci BPMN .....	56
5.3.3 – Use-Case model procesu .....	59
5.4 – Návrh nového procesu .....	60
5.5 – Datový slovník .....	63
5.6 – Požadavky na systém .....	64
5.7 – Specifikace aktérů a případů užití .....	65
5.7.1 – Aktéři .....	65
5.7.2 – Případy užití .....	66
5.8 – Statický pohled na model systému .....	68
5.8.1 – Charakteristika tříd .....	69
6 – Výsledky a diskuze .....	72
7 – Závěr .....	74
8 – Seznam použitých zdrojů .....	75
9 – Přílohy .....	77

# 1 – Úvod

Efektivní fungování subjektů podnikové sféry si lze v současné době jen obtížně představit bez využití moderních informačních technologií, zejména informačních systémů. Tyto systémy mohou nalézt svá využití téměř při všech činnostech podniku. Může se jednat o jednoduché evidenční systémy, distribuční a prezentační nástroje, ale také o rozsáhlé podnikové systémy, které automatizují řadu procesů v organizaci jako celku.

Pro úspěch v konkurenčním boji na trhu statků a služeb se ekonomické subjekty neustále snaží přizpůsobovat nabídku svých produktů tak, aby byla pro zákazníka co nejpřitažlivější a snadno dostupná. K tomuto účelu slouží zejména informační systémy využívající webové technologie. Příkladem takových systémů mohou být různé interaktivní prezentace, rezervační systémy nebo internetové obchody. Dalším důležitým předpokladem úspěchu podniku je jeho efektivní činnost bez zbytečných nákladů. Také v této oblasti jsou možnosti informačních technologií nezastupitelné. Informační systémy dokáží výrazně zefektivnit zavedené podnikové procesy, v rámci kterých usnadňují a také urychlují sběr, vyhodnocování a prezentaci dat. S tímto přínosem také úzce souvisí vyšší spolehlivost a přesnost poskytnutých výstupů. Řídícím pracovníkům pak získané informace mohou usnadnit proces rozhodování a snižují pravděpodobnost omylu. Před nasazením informačního systému do podnikového prostředí je však často třeba zavedené procesy optimalizovat, aby se případná neefektivita nepromítla také do jejich automatizovaného provádění. Této problematice se věnuje procesní řízení, v jehož zájmu je právě analýza, návrh a zlepšování podnikových procesů. V souvislosti s pronikáním stále dokonalejších informačních systémů do různých oblastí lidské činnosti, dochází také k vývoji technik systémové analýzy a návrhu. S příchodem jazyka UML jsou původní strukturované metody postupně nahrazovány objektově orientovaným přístupem, který lépe vyhovuje současným trendům ve vývoji softwarových produktů.

Instituce, které chtějí získat na trhu konkurenční výhodu, jsou v současné době nuceny zavádět informační systémy do podnikového prostředí jak pro komfortnější služby svým zákazníkům, tak pro svůj efektivnější provoz. Při uvážení rostoucí nabídky možností, které informační technologie nabízejí, lze předpokládat, že tento trend bude pokračovat i nadále.

## 2 – Cíl práce a metodika

Hlavním cílem této diplomové práce je provést analýzu vybraného zavedeného podnikového procesu a za použití jazyka UML navrhnout konceptuální model informačního systému, který by tento proces zefektivnil. Dílčí cíl práce, z něhož analýza a návrh informačního systému vychází, je vytvoření uceleného přehledu řešené problematiky.

Teoretická východiska jsou v této práci rozdělena do 3. a 4. kapitoly. Ve 3. kapitole je věnován prostor zejména základním pojmům týkajících se podnikových procesů definovaných v použitých informačních zdrojích, funkčnímu a procesnímu řízení podniku a používaným metodám zlepšování procesů. Dále jsou v této kapitole uvedeny možné reprezentace podnikových procesů se zaměřením na nejčastěji používané grafické notace.

4. kapitola se zaměřuje na vývoj přístupů k analýze a návrhu informačních systémů. V úvodu jsou nejprve charakterizovány klasické strukturované metody včetně nejznámější Yourdonovy moderní strukturované analýzy. Následuje přehled objektově orientovaných metod a technik, které na strukturovaný přístup navázaly. Kapitola je uzavřena představením principů a nástrojů objektově orientovaného modelovacího jazyka UML s uvedením přehledu diagramů, které nabízí jeho poslední verze 2.4.

Praktická část práce je obsažena v 5. kapitole. V jejím úvodu autor představuje činnost Úseku správy příspěvků ve zvoleném penzijním fondu. Činnost úseku je nejprve charakterizována jako celek, z něhož je poté na základě specifikace problému identifikován proces, který má být zefektivněn zavedením informačního systému. Tímto procesem je souhrn činností vykonávaných při zpracování rozpisů hromadných plateb zaměstnavatelů. Dle získaných charakteristik autor poté vytvořil popis a model zkoumaného procesu. Na základě poznatků z předchozí fáze je navržena nová podoba procesu, jehož automatizace informačním systémem povede k zefektivnění postupu při zpracování rozpisů hromadných plateb. Následuje formulování požadavků na systém jako výchozí bod k objektově orientované systémové analýze a návrhu, kterým je věnován zbytek kapitoly.

V závěru jsou shrnuty poznatky získané při analýze a návrhu daného informačního systému a zhodnoceny výsledky praktické části práce. Součástí jsou rovněž doporučení formulovaná na základě problémů, které se v průběhu řešení objevily.

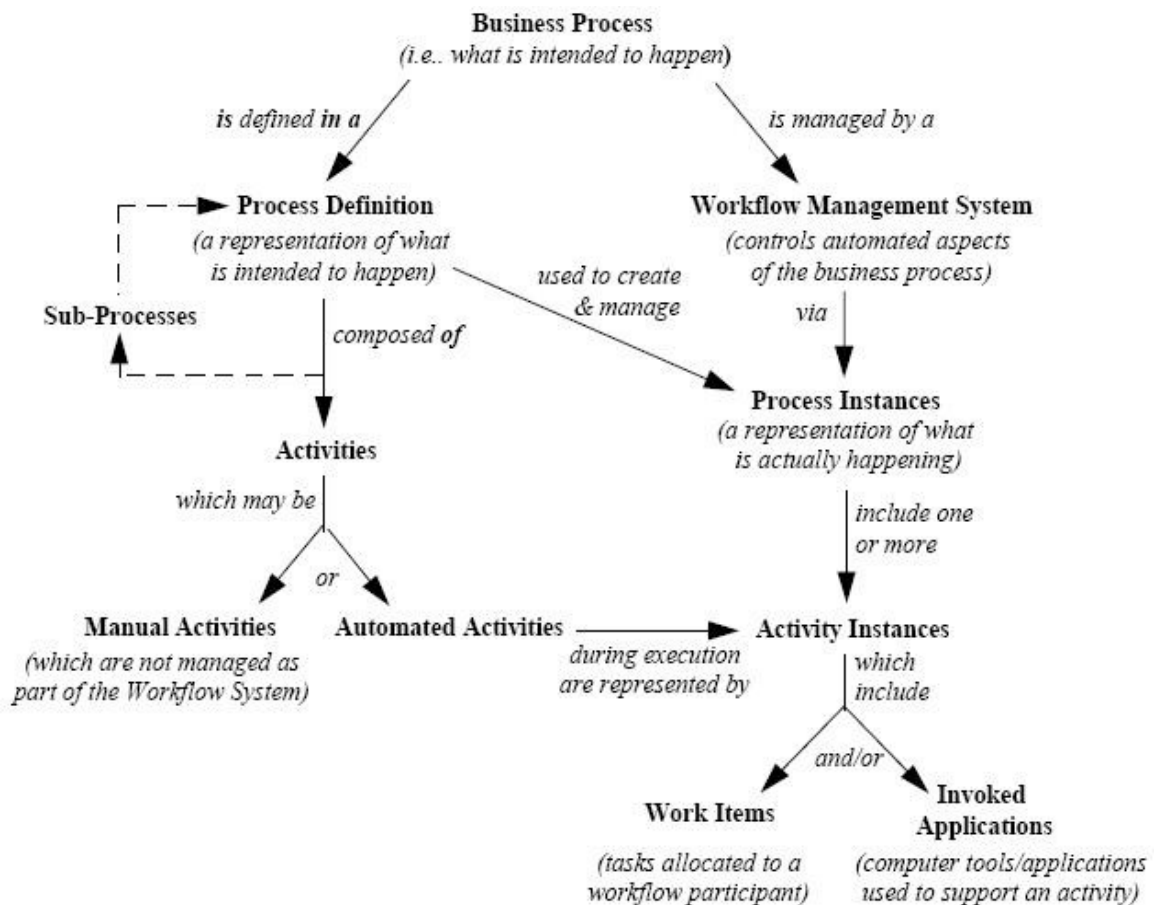
### 3 – Podnikové procesy

V první části této kapitoly jsou představeny základní pojmy z oblasti procesního řízení následované stručnou charakteristikou funkčního a procesního přístupu k řízení podniku. Dále jsou zde uvedeny jednotlivé druhy procesů a zhodnoceny přístupy k jejich zlepšování. Závěr kapitoly je vymezen pro problematiku modelování procesů a používaným grafickým notacím.

#### 3.1 – Definice základních pojmů

V této části jsou na základě definic z několika odborných zdrojů charakterizovány základní pojmy z oblasti procesního řízení včetně jejich vzájemných vztahů. Pro snadnější orientaci v dále uvedených termínech a jejich souvislostech lze využít následujícího schématu.

Obrázek 1: Vztahy mezi základními pojmy procesního řízení [25]





### 3.1.1 – Podnikový proces

V odborných informačních zdrojích definice pojmů proces a podnikový proces často splývají nebo jsou velmi podobné. Vymezení podnikového procesu zpravidla odpovídá některé z obecných definic procesu aplikované do podnikového prostředí. Obecné definice mohou vypadat následovně:

**Proces** je po částech uspořádaná množina činností, jež na základě jednoho nebo více vstupů tvoří opakovatelným způsobem požadovaný výstup. [18]

**Proces** je jakákoliv sekvence předem definovaných činností, vykonávaných za účelem dosažení předem specifikovaného typu nebo rozsahu výsledků. [6]

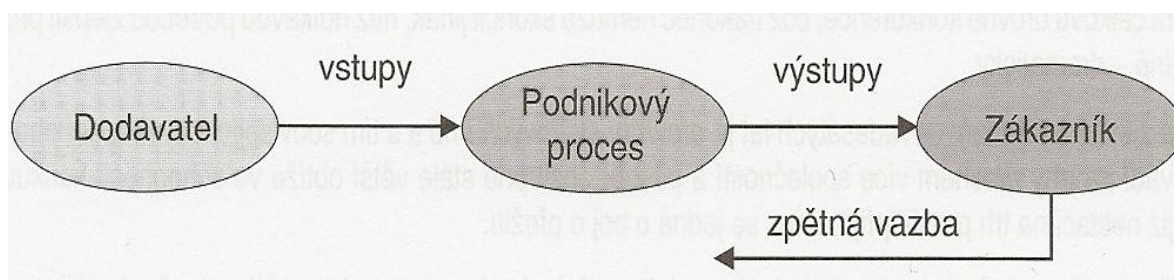
Na tomto místě je vhodné upozornit, že proces se nemusí skládat z činností následujících v řadě po sobě, ale může se jednat o souhrn činností probíhajících také paralelně. Tuto skutečnost vyjadřuje první z uvedených definic termínem „po částech uspořádaná množina“. Vymezení podnikového procesu lze získat úpravou a doplněním výše uvedených obecných definic:

**Podnikový proces** je souhrnem činností, transformujících souhrn vstupů do souhrnu výstupů (zboží nebo služeb) pro jiné lidi nebo procesy, používající k tomu lidi a nástroje. [5]

**Podnikový proces** je tok práce postupující od jednoho člověka k druhému a v případě větších procesů i z jednoho oddělení do druhého, přičemž procesy lze definovat na celé řadě úrovní. Vždy však mají jasně vymezený začátek, určitý počet kroků uprostřed a jasně vymezený konec.[18]

První z uvedených definic podnikového procesu lze pro lepší názornost znázornit také graficky:

Obrázek 2: Základní schéma podnikového procesu [5]



Šmída uvádí ještě několik dalších vymezení pojmu proces resp. podnikový proces, ale zároveň zdůrazňuje, že ani jedno z nich není úplné. Zdůvodňuje to tím, že žádná z definic neuvádí souběžně možnost dělení procesu na subprocessy, konkrétní vstupy procesu, existenci externích a interních zákazníků ani průchod procesu několika odděleními nebo dokonce i několika podniky. Nabízí proto definici vlastní, jejímž cílem je co nejpřesnější vymezení pojmu proces:

**Proces** je organizovaná skupina vzájemně souvisejících činností a/nebo subprocessů které procházejí jedním nebo více organizačními útvary či jednou (podnikový proces) nebo více spolupracujícími organizacemi (mezipodnikový proces), které spotřebovávají materiální, lidské, finanční a informační vstupy a jejichž výstupem je produkt, který má hodnotu pro externího nebo interního zákazníka.[6]

Vzhledem k tomu, že všechny definice pojmů proces a podnikový proces vycházejí ze stejného nebo velmi podobného základu, není dále v této práci mezi zmíněnými pojmy rozlišováno.

### 3.1.2 – Workflow

V oblasti podnikových procesů je termín workflow chápán v poněkud užším významu než jako obecný tok práce nebo schéma provádění nějaké činnosti. Přesnější vymezení v procesním řízení spočívá v podmínce automatizace tohoto toku. Workflow Management Coalition<sup>1</sup> uvedený termín definuje takto:

**Workflow** označuje automatizaci celého nebo části podnikového procesu, během kterého jsou dokumenty, informace nebo úkoly předávány od jednoho účastníka procesu k druhému na základě množiny procedurálních pravidel.[25]

Automatizace workflow je zabezpečována workflow management systémy (systémy řízení workflow), které lze charakterizovat jako:

Systémy, které definují, vytvářejí a řídí provedení workflow prostřednictvím speciálních softwarových aplikací (workflow engines), jež jsou schopné interpretovat definici procesu a komunikovat s jednotlivými účastníky za případného využití dalších IT nástrojů a aplikací [25].

---

<sup>1</sup>Organizace zabývající se zejména standardizací a zvyšováním interoperability workflow systémů (www.wfmc.org)

### 3.1.3 – Aktivita

Aktivita (někdy také označována jako činnost) je definována jako jeden logický krok uvnitř procesu.[25] Při modelování procesů je aktivita obvykle vnímána jako nejmenší jednotka, kterou v rámci procesu uvažujeme. Tento pohled je však relativní. Podle Řepy může být obecně každá aktivita (činnost) samostatně popsána jako proces. Úroveň abstrakce vždy závisí na potřebě srozumitelnosti modelu, použitém nástroji, omezení možné velikosti modelu, stylu autora apod.[5]

Aktivita dále dělíme na manuální, jejichž provádění není podporováno informačními technologiemi, a automatizované (workflow aktivity). Manuální aktivity nelze automatizovat a nejsou tedy součástí workflow. Mohou se však vyskytovat v definici procesu např. při jeho modelování.[25]

### 3.1.4 – Instance

Instancí rozumíme jeden konkrétní výskyt procesu (instance procesu) nebo činnosti v rámci procesu (instance aktivity). Instance může být řízena nezávisle, má svůj vlastní vnitřní stav a zvenčí viditelnou identitu.[25]

Příkladem instance procesu, kterým se zabývá tato práce, může být zpracování jedné konkrétní hromadné platby od jejího přijetí na účet penzijního fondu až po dokončení rozúčtování odpovídajících příspěvků jednotlivým klientům. Jako příklad instance aktivity lze uvést (dle zvoleného stupně abstrakce) např. spárování konkrétní hromadné platby s příslušným rozpisem (více v kapitole 5)

## 3.2 – Funkční vs. procesní řízení

Do současné doby byl funkční styl řízení hojně využívaným manažerským přístupem k vedení podniku. Tento styl vychází z myšlenek skotského ekonoma a filozofa Adama Smith<sup>2</sup>, podle nějž mají být výrobní procesy rozloženy na nejjednodušší úkony. Funkční jednotky tedy slouží k rozdělení složitějších činností a k jejich dekompozici na jednoduché kroky, které může zvládnout i nekvalifikovaný pracovník.[10] Jde o zcela logický přístup, který je však v současné době pro řadu organizací v mnoha ohledech nevýhodný. Jedním ze zásadních problémů je rozdělení práce mezi několik různých pracovních týmů, z nichž každý má na starosti pouze jednu konkrétní činnost. Tímto způsobem je možné zdokonalovat jednotlivé kroky vedoucí k výslednému produktu, ale již není možné

---

<sup>2</sup> Tyto myšlenky Smith uvedl ve svém nejznámějším díle *Pojednání o podstatě a původu bohatství národů* (1776), které je také známé pod zkráceným názvem *Bohatství národů*.

optimalizovat systém jako celek. Vylepšením jednoho samostatného článku může celý systém na efektivitě dokonce ztratit. Dalšími nevýhodami funkčního řízení jsou možné komunikační bariéry mezi jednotlivými týmy, nedefinovaná nebo nejasná odpovědnost za některé části procesu a v neposlední řadě také obtížné nahrazování klíčových pracovníků, kteří si odchodem z organizace s sebou odnášejí také část podnikových znalostí (know-how).[10]

Uvedené nevýhody funkčního řízení se snaží eliminovat procesní přístup, který umožňuje vysokou míru optimalizace. Je to dáno množstvím informací obsažených v popisech procesů a jejich modelech. Informace tak nejsou udržovány pouze v hlavách zaměstnanců, ale v modelech a popisech procesů. Je tedy velice usnadněno sdílení těchto informací a jejich změna, což je podpořeno možností zápisu procesů unifikovaným a snadno interpretovatelným způsobem. Oproti funkčnímu řízení procesní řízení dále také zcela určuje zodpovědnost za proces. (...) Jelikož proces definuje aktivity, které nejsou předávány dále pryč z procesního týmu, je zodpovědnost striktně dodržována a zpětně vysledovatelná. [10]

V ekonomicky nestabilní době posledních let je pro většinu organizací nezbytné dokázat pružně reagovat na změny v tržním prostředí. Z výše uvedeného vyplývá, že společnosti využívající procesní způsob řízení mají značnou konkurenční výhodu před společnostmi orientovanými funkčně.

### **3.3 – Klasifikace procesů**

Odborné literatura nabízí řadu hledisek, podle nichž jsou procesy klasifikovány a rozdělovány. Smyslem této části je představení nejčastěji rozlišovaných typů procesů, se kterými se lze v praxi setkat.

Pro svoji přehlednost a jednoduchost se často používá rozlišování procesů na hlavní, řídicí a podpůrné. Základní charakteristika těchto typů procesů je uvedena v následující tabulce.

Tabulka 1: Typy, způsob řízení a všeobecná charakteristika podnikových procesů [6]

Typ procesu	Způsob, jakým má být řízen	Charakteristika procesu			
		Přidává hodnotu?	Probíhá napříč organizací?	Má externí zákazníky?	Generuje tržby (zisk)?
hlavní	výkonově	ANO	ANO	ANO	ANO
Řídící	nákladově	NE	ANO	NE	NE
podpůrný	výkonově, možnost outsourcingu	ANO	NE	NE	NE

Další pohled nabízí norma ISO 9001:2000, podle níž existují 4 typy procesů:

- procesy řídicí
- procesy přípravy zdrojů
- procesy realizace produktu
- procesy dalšího rozvoje (měření, analyzování, zlepšování).

Tohoto dělení se musí držet podniky pokud chtějí být certifikovány podle ISO. [6]

Jiný přístup ke klasifikaci procesů vychází z globálního procesního modelu, který zobrazuje statickou strukturu procesů se vzájemnými vazbami. Na základě uvedeného modelu rozlišujeme:

- klíčové procesy – procesy poskytující základní produkt, přinášejí organizaci hodnotu
- podpůrné procesy – ostatní procesy, které poskytují služby jiným procesům (klíčovým i podpůrným). Tyto procesy jsou dále rozdělovány na servisní a průřezové.
  - servisní – procesy specializované na jasnou službu/produkt, který dodá svým průběhem od začátku do konce. Příkladem může být proces Přijímací řízení na VŠ jako podproces procesu Vzdělávání
  - průřezové – mají relativně samostatnou logiku průběhu. Ostatním procesům poskytují svoje částečné výstupy podle potřeby (např. proces Správa studijních programů a akreditací, nebo proces Provozování ICT). [5] Již z podstaty uvedených příkladů je zřejmé, že tyto procesy nejsou zaměřeny na jedinou službu/produkt, ale mají průřezový charakter.

Šmída dále uvádí několik přístupů, podle nichž jsou procesy rozdělovány podle svých vlastností pouze do dvou skupin. Mezi nejčastější tzv. bipolární dělení patří zejména:

- rozlišování na vnitropodnikové procesy a procesy jdoucí za hranici firmy
- dělení procesů na procesy zaměřené na externího zákazníka (prodej produktu a jeho úspěch na trhu) a procesy zaměřené na interního zákazníka (realizace produktu)
- dělení procesů na procesy zajišťující krátkodobou prosperitu (výroba, prodej produktu) a procesy zajišťující dlouhodobou prosperitu (výzkum a vývoj, tvorba strategie)[6]

Z výše uvedeného přehledu je patrné, že existuje celá řada možností jak podnikové procesy kategorizovat. Nelze však jednoznačně určit který přístup je nejlepší nebo nepraktičtější. Vždy záleží na konkrétních potřebách a situaci. Jako nejuniverzálnější doporučuje Šmída používat dělení procesů na hlavní, řídicí a podpůrné.

### **3.4 – Zlepšování procesů**

Vlivem razantních změn, kterými v posledních desetiletích prochází podnikatelské prostředí, jsou organizace neustále nuceny zdokonalovat své produkty, aby byly schopné čelit konkurenci a udržet se na trhu. Zlepšování podnikových procesů je pro tento účel zcela nezbytné. Tématem této kapitoly jsou přístupy k procesnímu zlepšování, jehož vývoj se přirozeně odvíjí od změn v podnikatelském prostředí.

#### **3.4.1 – Průběžné zlepšování procesů**

Tento přístup je založen na přírůstkovém zdokonalování již zavedeného procesu. Základem je popis současného stavu a určení ukazatelů, na jejichž základě lze proces hodnotit. Zmíněné ukazatele mají většinou vazbu na zákazníky daného procesu. Následuje fáze sledování běhu procesu a identifikování jeho případných úprav. Po implementaci navržených změn je možné celý postup (teoreticky neustále) opakovat. Z tohoto důvodu se tento přístup také někdy označuje jako soustavné zlepšování podnikových procesů.[5]

Obrázek 3: *Průběžné zlepšování procesů* [5]



Postupem času však stávající způsob zlepšování procesů přestával podnikům pro úspěch na trhu stačit a nedovoloval využít potenciál nabízený stále dokonalejšími technologiemi. Právě technologický pokrok je dle Řepy hlavním důvodem přechodu z přírůstkového zlepšování procesů na jiný důslednější způsob označovaný jako business process reengineering (dále jen BPR) – viz následující kapitola.

### 3.4.2 – Business process reengineering

Odlišnost BPR od Průběžného zlepšování procesů je patrná již z jeho základní definice:

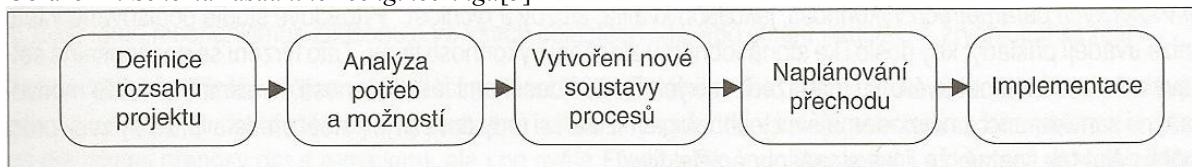
„BPR znamená zásadní přehodnocení a radikální rekonstrukci (redesign) podnikových procesů tak, aby mohlo být dosaženo dramatického zdokonalení z hlediska kritických měřítek výkonnosti, jako jsou náklady, kvalita, služby a rychlost.“ [4]

Řepa charakterizuje BPR ještě radikálněji:

„BPR je kulturně zcela jiným přístupem, než průběžné zlepšování procesů. Ve své extrémní podobě BPR předpokládá, že stávající podnikový proces je zcela nevyhovující – nefunguje a je třeba jej z podstaty změnit, od počátku.“[5]

Z uvedených definic je patrné, že při BPR není aktuální stav procesu pro jeho novou podobu nijak podstatný. Výchozím bodem zásadního BPR je definice rozsahu a hlavních cílů chystaného projektu. Následuje nejdůležitější část, kterou je důkladná analýza prostředí, ve kterém bude nový proces implementován (potřeby zákazníků procesu, stav konkurence, technologické možnosti apod.). Po analýze je již možné přistoupit k návrhu nového procesu případně soustavy procesů a jejich souvislostí. Před samotnou implementací je ještě třeba naplánovat změny nezbytné pro zavedení nového procesu (organizační a technologické).[5] Názorně je schéma BPR uvedeno na obrázku 4.

Obrázek 4: Schéma zásadního reengineeringu[5]

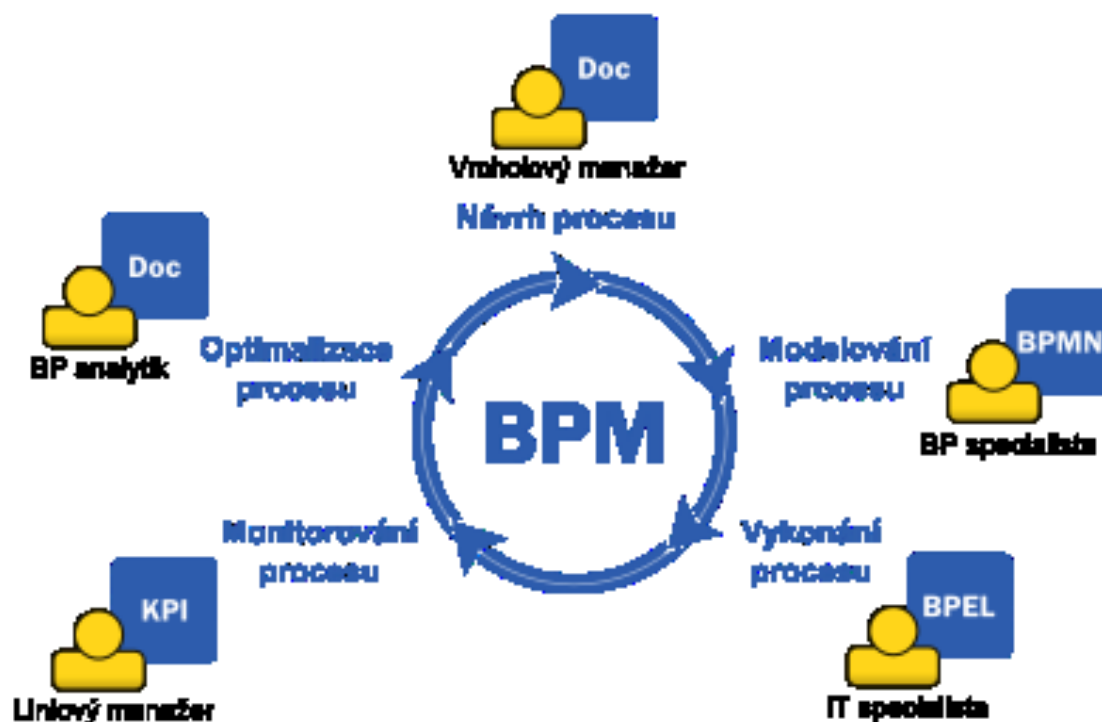


### 3.4.3 – Business process management

Od předchozích přístupů ke zlepšování procesů se Business process management (dále jen BPM) liší zejména svým komplexním pojetím. Cílem BPM není pouze jednorázová radikální změna procesu jako v případě BPR, ani není v jeho zájmu vylepšování již zavedených procesů postupným způsobem. Záměrem BPM je kontinuální zlepšování

cestou řízení procesů v průběhu jejich celého životního cyklu – viz obr. 5. Prakticky to znamená, že se v rámci BPM mohou prolínat oba dříve popsané přístupy s dalšími podnikovými aplikacemi pro zlepšování procesů (podpora workflow a IT, měření klíčových ukazatelů výkonnosti – KPI, začlenění kulturních aspektů – řízení změny apod.).[14]

Obrázek 5: Životní cyklus BPM [12]



„Význam BPM spočívá ve schopnosti vytvořit jedinou definici procesu, v níž mohou být poskytnuty různé úhly pohledu na ten samý proces (...) To znamená, že různí lidé s různými odbornostmi mohou vidět stejný proces různě a nakládat s ním tak, jak jim to vyhovuje. Všichni přitom pracují s jediným zdrojem (...) Například manažer může pracovat s výkonností procesu a porovnávat ji s KPI. Analytik si může zobrazit podrobnou procesní mapu, pro výkonné pracovníky je k dispozici procesní portál, výstupem pro programátora může být jazyk procesu kompatibilní s programovacím jazykem atd.“ [6]



### 3.4.4 – Shrnutí

V této části jsou přehledným způsobem ve formě tabulky představeny základní rozdíly v uvedených přístupech ke zlepšování procesů.

Tabulka 2: Porovnání průběžného zlepšování procesů, BPR a BPM [upraveno dle 6]

Faktor	Zlepšování procesů	BPR	BPM
Úroveň změny	inkrementální	radikální	týká se celého životního cyklu
Interpretace „as is“ „to be“	současný proces nová vylepšená verze	starý proces zcela nový proces (diskontinuita)	žádná způsobilost BPM způsobilost BPM
Výchozí bod	existující procesy	čistý list papíru	nové nebo existující procesy
Frekvence změn	jednorázové nebo kontinuální změny	periodicky prováděné jednorázové změny	jednorázové, pravidelné, pokračující i evoluční
Potřebný čas	krátkodobý horizont	dlouhodobý horizont	v reálném čase
Participace	zdola nahoru	shora dolů	zdola nahoru i shora dolů
Počet dotčených procesů	simultánní realizace, napříč několika procesy	každý proces samostatně	simultánní realizace, napříč mnoha procesy
Typický rozsah působnosti	úzký, uvnitř funkcí	široký, mezifunkční	všechny procesy v rámci hodnotového řetězce
Horizont	minulost a současnost	budoucnost	minulost, současnost i budoucnost
Riziko	mírné	vysoké	nízké
Primární umožňující nástroj	statistická regulace	informační technologie	procesní technologie
Nástroje	off-line	žádné	on-line
Zapojení odborníci	odvětvoví specialisté	všestranní pracovníci v oblasti businessu	procesní inženýři a všichni zaměstnanci
Práce	praxe, zkušenost	procesní	procesní praxe, zkušenost
Cesta k realizaci	kulturní změna	kulturní i strukturální změna	matematický základ, procesní technolog. standardy

### 3.5 – Analýza a modelování podnikových procesů

Předmětem této kapitoly je analýza podnikových procesů a techniky procesního modelování. V úvodu jsou přestaveny obecné principy procesní analýzy se zaměřením na její jednotlivé fáze a očekávané výstupy. Dále je kapitola věnována nejznámějším metodám a standardům modelování procesů. Na závěr jsou uvedeny typicky používané notace pro grafické vyjádření procesních modelů.

### 3.5.1 – Analýza procesů

Analýza procesů je poměrně široký pojem zahrnující řadu úkonů a postupů směřujících většinou ke stejnému cíli, který však může dále sloužit různým účelům. Z tohoto důvodu ani odborná literatura nenabízí jedinou obecně uznávanou definici procesní analýzy, ale většinou ji charakterizuje dle použité metody nebo techniky.

Bez ohledu na zvolený přístup k analýze je jejím požadovaným výstupem konceptuální procesní model organizace - tedy základní model popisující všechny hlavní procesy, které zabezpečují činnost podniku, včetně jejich struktury a vzájemných vazeb. Potřeba analyzovat procesy vychází z aktuálních podnikových záměrů a plánů. Výstupy analýzy procesů mohou spolu s konceptuálním objektovým modelem sloužit jako východisko při vývoji IS podniku. V některých případech chce společnost své procesy analyzovat a vytvořit jejich model za účelem dokumentace pro budoucí využití.[5] Dále mohou výstupy z procesní analýzy nalézt svá uplatnění při odhalení slabých míst v činnosti podniku a jejich následném řešení. Na základě poznatků získaných při tvorbě konceptuálního procesního modelu (jako předstupně implementačního modelu<sup>3</sup>) a následného návrhu zlepšení procesů lze tato slabá místa odstranit – princip představuje Šmída následovně:

„Ve většině klasických podniků (funkčně specializovaných, bez zaměření na procesy) existují problémy, často skryté. K jejich odhalení je potřebné pochopit, co všechno je zapojené do procesu. Namísto toho, abychom se při analýze ztratili v jednotlivých činnostech, je lepší vytvořit si hrubou představu procesu od začátku do konce a po vytvoření hrubého náčrtu procesu přejít k jeho konkrétnějšímu popisu.“ Cílem zjistit:

- poslaní procesu, jeho produkty a komu jsou určeny,
- kde a čím proces začíná a končí,
- jaké procesy na sebe navazují a jak jsou vzájemně propojené,
- průběh základních podprocesů a jejich činností,
- oddělení, kterými proces probíhá,
- vstupy, které proces spotřebovává (včetně IT),
- vstupy a výstupy každé činnosti,
- zodpovědnosti za činnosti, podprocesy a procesy. [6]

---

<sup>3</sup> Implementační model procesů je poslední úrovní modelu procesů a je podkladem k dalším navazujícím činnostem zavedení systému procesů (tj. vytvoření příslušných organizačních a technických podmínek pro běh procesů, naplánování a následné provedení projektu zavedení procesů).[5]

Podrobněji představuje analýzu procesů Řepa v rámci charakteristiky metodiky modelování a analýzy podnikových procesů MMABP (Methodology for Modeling and Analysis of Business Process)

### **3.5.2 – Analýza procesů dle metodiky MMABP**

Výchozím bodem analýzy dle MMABP je výsledek analýzy událostí a vnějších reakcí (tzv. 0. krok). Samotná analýza se dále skládá ze 3 paralelně probíhajících fází (1 – Analýza elementárních procesů, 2 – Specifikace klíčových procesů, 3 – Specifikace podpůrných procesů).

#### **0. krok – Analýza událostí a vnějších reakcí**

Cílem tohoto kroku je zjistit veškeré relevantní reálné události (vznikající mimo organizaci), které vedou, nebo jsou podstatné pro dosažení cíle, vznik produktů a provádění činnosti podnikových procesů, a tyto události přiřadit vnějším reakcím (směřujícím mimo organizaci). Události lze rozdělit do dvou základních typů:

- události věcné – odrážejí nějakou akci objektu z okolí podniku (zákazníka, konkurence, legislativního objektu, apod.)
- události časové – jsou dané časem, kdy je od procesu něco požadováno (konec měsíce, účetního období, apod.)

Jedna událost se typicky vyskytuje jako příčina různých reakcí. Události uspořádané k jedné reakci mají vždy určité pořadí. Každé takové uspořádání potom představuje jeden elementární přirozený proces v organizaci [5].

#### **Fáze 1 – Analýza elementárních procesů**

V průběhu první fáze dochází k identifikaci elementárních procesů organizace včetně jejich základní vnitřní struktury a vzájemných souvislostí uvažovaných v kontextu hlavního smyslu a účelu činnosti podniku. Výsledkem této fáze je systém elementárních procesů sloužící jako podklad ke specifikaci klíčových procesů.[5]

#### **Fáze 2 – Specifikace klíčových procesů**

Druhá fáze si klade za cíl rozpoznat klíčové procesy v organizaci prostřednictvím objektové analýzy produktů organizace. Na základě výsledků předchozí etapy je dále jejím cílem zjistit jejich vnitřní strukturu a vzájemné vazby. Výstupem uvedeného postupu je

system konceptuálních klíčových procesů, jenž představuje základní podklad (po doplnění podpůrných procesů) ke konstrukci procesního modelu organizace.[5]

### **Fáze3 – Specifikace podpůrných procesů**

Závěrečná fáze analýzy procesů slouží k identifikaci podpůrných procesů prostřednictvím objektové business analýzy organizace, jejímž výsledkem je objektový model organizace<sup>4</sup>. Výsledkem specifikace podpůrných procesů je kompletní systém konceptuálních procesů v organizaci, který je základním podkladem ke konstrukci procesního modelu organizace a k následné implementaci procesů.[5]

Po dokončení analýzy se předpokládá fáze implementace procesů, kde se jednotlivé procesy transformují do konkrétní podoby dle organizační a technologické podoby organizace. Tomuto kroku může ještě předcházet případný reengineering procesů. Na základě vzniklého implementačního modelu jsou vytvořeny příslušné organizační a technické podmínky pro běh procesů a zrealizován projekt zavedení systému procesů do organizace.[5]

Autor zvolil podrobnější charakteristiku analýzy procesů dle metodiky MMBAP, protože se jedná o obecnou metodiku zkoumání procesů, která se nespécializuje na žádný konkrétní aspekt podnikových procesů. Z toho důvodu je MMBAP aplikovatelná do různých prostředí při zachování své jednoduchosti a zároveň plné funkčnosti. Další významné metodiky a techniky analýzy podnikových procesů jsou předmětem následující kapitoly.

### **3.5.3 – Další metodiky a techniky analýzy podnikových procesů**

#### **Metodika ARIS prof. Scheera**

Tato metodika nedefinuje žádný přesný postup analýzy procesů, spíše poskytuje řadu pohledů a nástrojů (především softwarových) k modelování jednotlivých aspektů fungování podniku. Princip metodiky je postaven na pěti základních pohledech na podnik, které jsou vzájemně obsahově propojeny (organizační, datový, funkční, procesní a výkonový). [5]

---

<sup>4</sup> Objektový model produktů z fáze 2 doplněný o ostatní objekty, jejichž role v procesu jsou jiné (např. aktéři, organizační jednotky, vstupy a výstupy procesu).

### **Bussines system planning (BSP)**

Metoda firmy IBM určená primárně k analýze a návrhu informační architektury podniku v rámci realizace jejího informačního systému. Při mapování všech podstatných faktorů působících na informační potřebu organizace dle BSP je nutné se zaměřovat na základy fungování organizace jako celku se všemi souvisejícími činnostmi (procesy). Z toho důvodu je BSP také metodou analýzy podnikových procesů a jejich vzájemných vazeb.[5]

### **ISAC (Information System Work and Analysis of Change)**

ISAC je metoda zaměřená zejména na počáteční fáze vývoje informačního systému. Hlavním předmětem zájmu této metody je především reálný systém (business system), podle kterého je informační systém vyvíjen. Zaměřuje se na řešení problémů ještě na úrovni reálného systému, aby nedošlo k jejich přenosu do systému informačního. Z uvedeného důvodu se ISAC řadí mezi tzv. problémově orientované metody. [5]

### **Select Perspective**

Metodika spojená s modelovacím nástrojem Select enterprise určeným pro modelování podnikových procesů. Select Perspective slouží k vytvoření procesního modelu organizace, který je sestavován v rámci tzv. analýzy podnikově-procesních požadavků na systém a je dle této metodiky základním východiskem analytické specifikace informačního systému. [5]

### **First Step**

Stejně jako Select Perspective vychází také metodika First Step z nástroje určeného pro modelování podnikových procesů (v tomto případě First Step Designer) Na rozdíl od předchozí metodiky však není zaměřena primárně na informační systémy, ale jejím předmětem zájmu je využití technologie v procesech obecně. Při tvorbě procesního modelu dle First Step je prováděna simulace namodelovaného procesu a zkoumání jeho technických aspektů jako např. doba trvání, náklady na proces, zaměstnanost zdroje apod. Na základě výsledků simulace mohou být v modelu provedeny případné změny za účelem zlepšení požadovaných parametrů. Celý tento optimalizační postup se zpravidla několikrát opakuje. [5]

Pro praktické využití kterékoliv z metodik analýzy podnikových procesů je nezbytné disponovat příslušnými nástroji resp. technikami umožňujícími modelování podnikových procesů v souladu s principy dané metodiky. Pro vizualizaci jednotlivých prvků procesních

modelů vytvářených dle různých metodik jsou používány různé grafické notace (sady grafických symbolů).

### 3.5.4 – Notace pro modelování procesů

Pro zobrazení procesů ve formální grafické podobě, která je čitelná pro všechny participanty životního cyklu BPM (též se používá označení „business uživatelé“), dnes existuje několik navzájem více či méně odlišných notací. Standardem v této oblasti má ambice být notace Business Process Modeling Notation (dále jen BPMN). Ostatní významné notace se však pro modelování procesů mohou zcela jistě v mnoha případech také uplatnit. Přehled v současné době nejpoužívanějších notací je v této kapitole uveden následně.

#### **BPMN**

První specifikace této notace označené jako BPMN v. 1.0 byla vydána v květnu roku 2004 organizovaným sdružením BPMI<sup>5</sup>, jehož členy byly významné společnosti na poli modelování podnikových procesů. V roce 2005 BPMI vstoupila do konsorcia OMG<sup>6</sup>, které se zabývá standardy pro objektově orientovanou specifikaci informačních systémů. O rok později byla notace BPMN přijata jako standard OMG pro modelování podnikových procesů[24]. Od ledna 2011 je BPMN ve verzi 2.0 (od této verze ve významu Business Process Model and Notation), která přináší oproti předchozím verzím několik vylepšení. Změny se týkají především přenositelnosti mezi různými nástroji podporujícími BPMN a také nových prvků vytvářených diagramů.

BPMN definuje Business Process Diagram (BPD) a od verze 2.0 ještě 2 podpůrné diagramy – diagram konverzace a diagram choreografie. BPD vychází z vývojových diagramů a je upraven pro vytváření vizuálních modelů operací business procesů. Model business procesů je potom sítí grafických objektů a kontrolních toků, které definují pořadí vykonání aktivit.[10]

Notace BPMN se snaží poskytnout jednoduchý nástroj pro modelování procesů, současně však umožňuje zachytit veškeré složitosti procesů. Uvedená vlastnost vychází z existence čtyř kategorií základních popisných elementů:

---

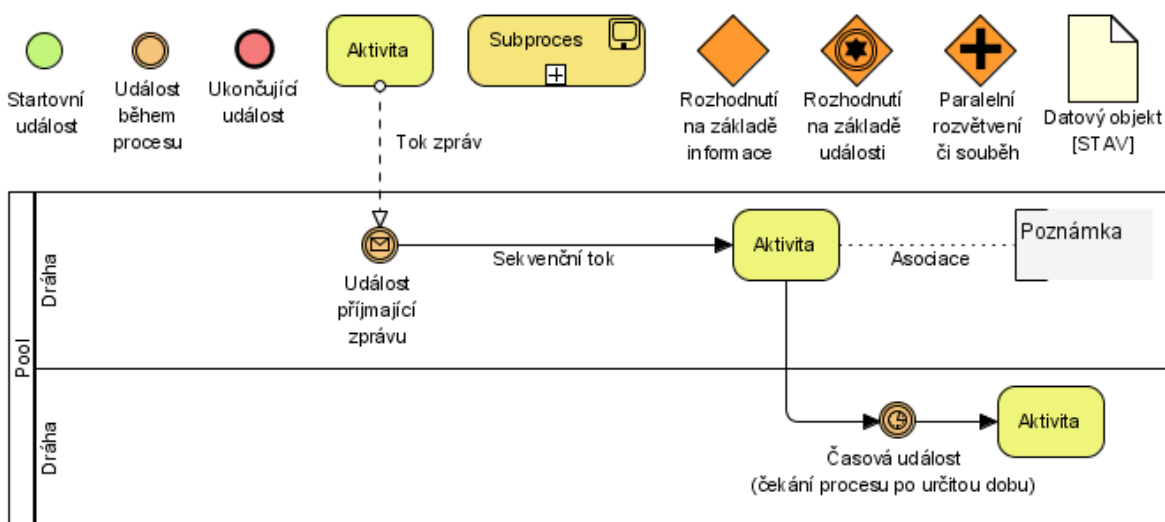
<sup>5</sup> Business Process Modeling Initiative

<sup>6</sup> Object Management Group

- **Plovoucí objekty (Flow objects)** – jde o objekty spojené s tokem informací při běhu procesu. Plovoucí objekty dále dělíme do tří kategorií:
  - i) události – něco co se děje v průběhu procesu. Události ovlivňují tok procesu a obvykle mají příčinu a/nebo důsledek. Termín událost může označovat např. zahájení činnosti, konec činnosti, přijetí zprávy, změna stavu zprávy, apod.
  - ii) aktivity – charakteristika aktivity odpovídá její obecné definici z kapitoly 3.1.3
  - iii) brány – představují elementy reprezentující místo v procesu, kde se tok procesu větví, na základě určité podmínky, nebo naopak slučuje.[20]
- **Propojovací objekty (Connecting objects)** – slouží k propojování tokových objektů v procesním diagramu. Podle jejich vyjadřovací schopnosti je dělíme do následujících tří kategorií:
  - i) sekvenční tok – určuje pořadí provádění jednotlivých aktivit v rámci procesu. Každý sekvenční tok má právě jeden zdroj a právě jeden cíl, kterými mohou být pouze události, aktivity nebo brány.
  - ii) tok zpráv – představuje komunikaci předáváním zpráv mezi dvěma entitami, které jsou reprezentovány bazény. Komunikace může být zobrazena jako předávání zpráv mezi dvěma bazény nebo přímo mezi plovoucími objekty v rámci bazénů. Toky zpráv však nemohou spojovat dva objekty v rámci jednoho bazénu.
  - iii) asociace – je používána ke spojení dodatečné informace (např. poznámka uživatele) nebo jiného artefaktu (viz níže) s plovoucím objektem.[20]
- **Dráhy (Swimlanes)** – BPMN používá koncept plaveckých drah pro vizuální oddělení aktivit v rámci jednoho procesu nebo více spolupracujících procesů. Aktivity jsou zobrazovány odděleně, aby mohla být rozlišena odpovědnost za ně. Rozlišujeme dva typy plaveckých drah:
  - i) bazény (pools) – reprezentuje účastníka procesu. Účastníkem může být specifická entita (např. firma, společnost, apod.) nebo obecněji určitá role (např. výrobce, prodejce, zákazník, apod.). Každý bazén může obsahovat vždy pouze jeden proces. Komunikace mezi procesy je poté zobrazena jako předávání zpráv mezi bazény

- ii) dráhy (lanes) – rozdělují bazén na více částí po celé jeho délce. Dráhy se používají pro přehlednější organizaci a kategorizaci aktivit v rámci jednoho procesu resp. bazénu (např. dle organizačních oddělení dané společnosti).[20]
- **Artefakty (Artifacts)** – umožňují uživateli zobrazovat dodatečné informace o procesu, které však přímo neovlivňují jeho běh. Artefaktem může být:
  - i) datový objekt – reprezentuje vstupní nebo výstupní data související s daným plovoucím objektem (nejčastěji s aktivitou). Datové objekty mohou být také zobrazovány jako součást sekvenčního toku nebo toku zpráv při komunikaci dvou objektů.
  - ii) skupina – umožňuje vizuálně zahrnout zvolené elementy diagramu do skupin. Seskupení neovlivňuje průběh procesu – slouží pouze jako dodatečná informace, která může zvýšit informační hodnotu diagramu.
  - iii) anotace – text s dodatečnými informacemi spojený s objektem, ke kterému se váže.[20]

Obrázek 6 :Přehled notace BPMN [12]



### EPC – Event-driven Process Chain

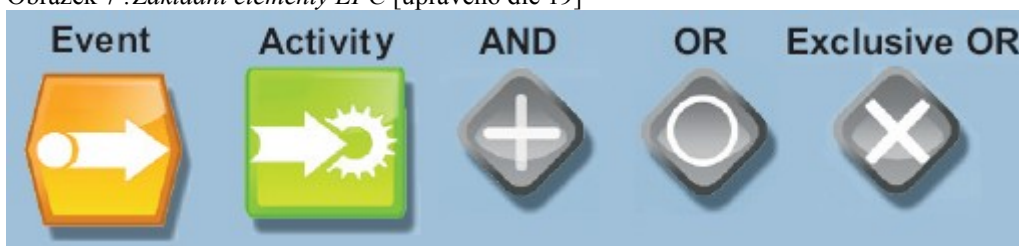
Notace EPC byla vyvinuta v roce 1992 na univerzitě Saarland v Německu ve spolupráci se společností SAP, která je v současné době předním dodavatelem podnikových aplikací. Notaci EPC spol. SAP později integrovala např. do nástroje NetWeaver určeného pro efektivní zlepšování podnikových procesů. [2][21]. EPC se také uplatňuje při modelování procesního pohledu na podnik dle metodiky ARIS. Podstata modelování za použití této



notace spočívá v řetězení událostí a aktivit do posloupnosti realizující požadovaný cíl. Základními elementy výsledného diagramu jsou:

- **Aktivity (*Activities*)** - představují základní stavební bloky procesního diagramu a určují, co má být v rámci procesu vykonáno.
- **Události (*Events*)** - popisují situace před a/nebo po vykonání aktivity. Aktivity jsou vzájemně propojeny pomocí událostí. Tzn. nějaká událost může vyjadřovat výstupní podmínku jedné aktivity a současně vstupní podmínku jiné aktivity.
- **Logické spojky (*Connectors*)** – používají se ke spojování aktivit a událostí. Tímto způsobem je popsán řídicí tok procesu. EPC používá tři typy spojek: (*AND* – a současně), (*OR* – nebo) a *XOR* (*exclusive OR* – vzájemně se vylučující nebo). [11]

Obrázek 7 :Základní elementy EPC [upraveno dle 19]



Postupně byla pro zvýšení popisné schopnosti notace EPC obohacena o další přídavné elementy, mezi něž zejména patří:

- **Role** – určuje, kdo provádí aktivitu
- **IT systémy** – symbol IT systému je spojován s aktivitou, kterou lze provádět automaticky
- **Rizika** – označují aktivity, které mohou mít na korektní běh procesu zásadní vliv.
- **Vstupní a výstupní data** – představují data, která jsou generována v průběhu procesu, nebo data nezbytná pro pokračování procesu.

Obrázek 8 :Rozšiřující elementy EPC [upraveno dle 19]

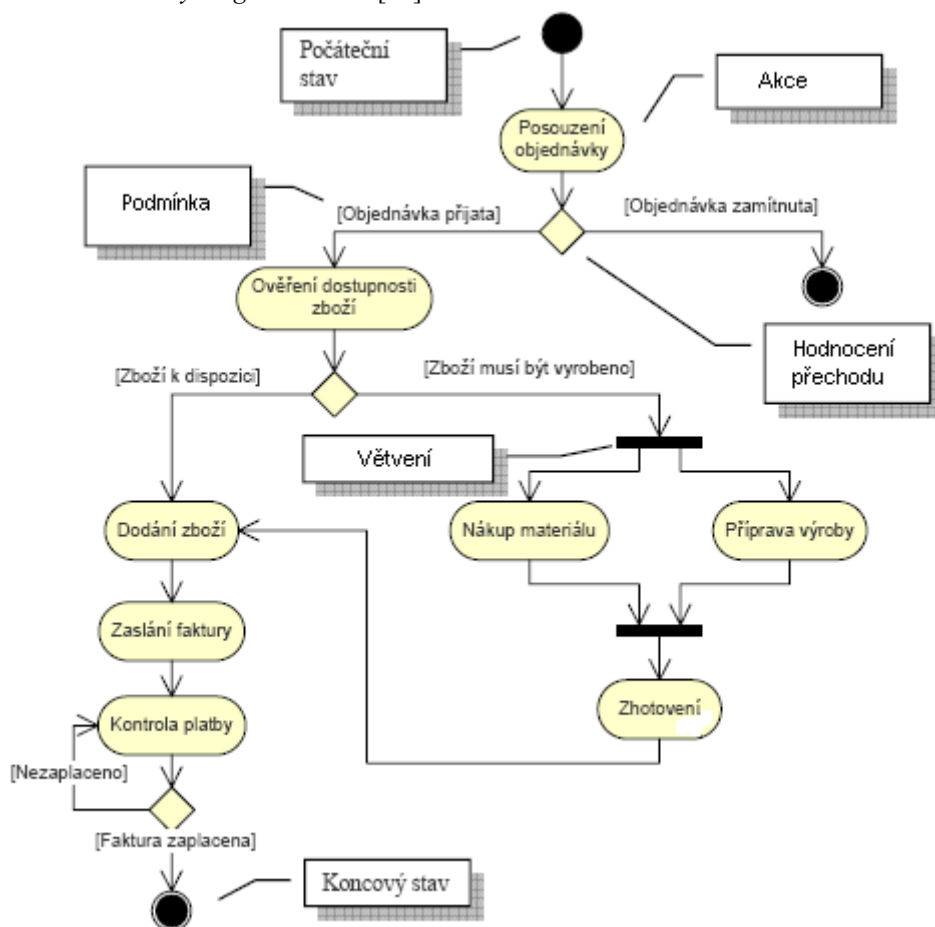


## UML – diagram aktivit

Jazyk UML byl primárně navržen za účelem sjednocení různých přístupů při vytváření specifikací požadovaných v rámci procesu tvorby softwarového produktu. V dnešní době se však stal univerzálním standardizovaným nástrojem pro vytváření modelů libovolného systému. V oblasti podnikových procesů se z jazyka UML uplatňuje především diagram aktivit, který modeluje procesy jako kolekce aktivit a přechodů mezi nimi. Pomocí tohoto diagramu lze zobrazit jak sekvenční, tak paralelní chování procesu. Diagram aktivit se skládá z následujících prvků:

- **Akce** – základní jednotka diagramu, vyjadřuje provádění nějaké činnosti, po jejímž dokončení je vyvolán přechod k další akci (akce je také někdy přímo nazývána aktivitou).
- **Počáteční a koncový symbol** - explicitně určují počáteční a koncový stav procesu.
- **Přechod** – vyjadřuje propojení mezi jednotlivými elementy diagramu
- **Hodnocení přechodu** – je znázorněno symbolem kosočtverce a slouží pro vyjádření logické podmínky, která určuje konkrétní přechod k další akci. Stejný symbol umožňuje ukončit podmíněné chování procesu sloučením větví oddělených v hodnocení.
- **Větvení a spojení** – umožňuje modelovat paralelní průběh procesu. Přitom dochází k rozvětvení přechodu na několik paralelně běžících vláken, která se následně opět spojí.
- **Plavecké dráhy** – podobně jako dráhy v BPMN slouží pro určení zodpovědnosti za danou aktivitu v rámci procesu[3].

Obrázek 9: Prvky diagramu aktivit[11]



### Porovnání BPMN, EPC a diagramu aktivit

Všechny 3 představené notace pro modelování procesů dokáží více či méně formálně zachytit téměř jakýkoliv podnikový proces. Každá z notací se však lépe uplatní pro jiný typ procesu resp. pro jiný typ podnikového prostředí.

Výhodou UML a zároveň tedy i diagramu aktivit je jeho standardizovaná notace, která je součástí řady modelovacích nástrojů, z nichž některé jsou i volně dostupné [11]. Jak již bylo uvedeno výše, UML je primárně technicky orientovaný jazyk pro modelování IS, a proto má také diagram aktivit největší vyjadřovací sílu pro procesy z technicky zaměřeného prostředí.

Notace EPC je vhodná pro tvorbu komplexních modelů složitých procesů. V každém případě klade důraz na jednoduchost a přehlednost výsledných diagramů, což však může mít v některých případech vliv na snížení úrovně jejich formalizace. Další nevýhodou EPC je její horší dostupnost vzhledem k tomu, že je součástí pouze nástrojů spol. ARIS a SAP[11].

BPMN nemá na rozdíl od EPC schopnost modelovat složité procesy přehledným způsobem. Je to dáno jejím původem ve vývojových diagramech obecných algoritmů, jenž se s tímto problémem potýkají také[8]. Při dostatečně jednoduché logice procesu je však BPMN naprosto vyhovující. Mezi výhody BPMN patří také snaha uskupení OMG o jeho standardizaci, což napomáhá jeho rozšíření do většího množství modelovacích nástrojů, než je tomu v případě EPC.

Vzhledem k výše uvedenému použil autor pro modelování průběhu procesu, jenž je předmětem praktické části této práce, notaci BPMN. Uvedený proces pochází z obecného podnikového prostředí, pro které se lépe hodí BPMN než diagram aktivit, a zároveň má dostatečně jednoduchou logiku, aby byl jeho diagram přehledný. Pro vytvoření diagramu by bylo možné použít samozřejmě i notaci EPC, ale dle subjektivního názoru autora je notace BPMN lépe čitelná a má lepší vyjadřovací schopnosti.

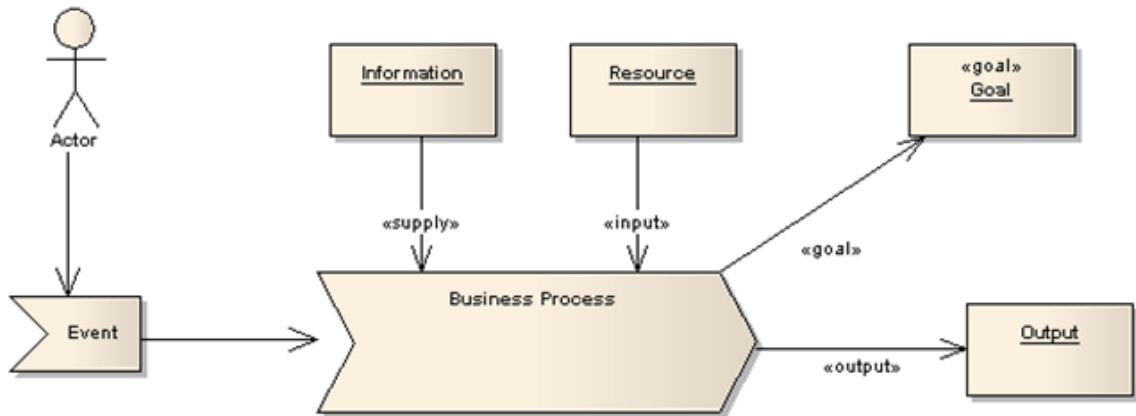
### **Notace Eriksson-Penker**

Všechny tři výše uvedené notace popisují dynamickou stránku procesu – tedy logiku postupu jednotlivých činností uvnitř procesu. Notace Ericsson – Penker, kterou lze označit za rozšíření jazyka UML, reprezentuje zcela odlišný přístup k modelování procesů. Jejím prostřednictvím je možné modelovat statickou strukturu procesů. Notace se zaměřuje na samotnou existenci procesů, jejich vzájemné vztahy a atributy (vstupy, výstupy, cíl, apod.). Jejím prostřednictvím je možné modelovat statický pohled na jednotlivé procesy nebo vytvořit globální procesní model celé organizace (tzv. procesní mapu). Součástí notace jsou následující elementy:

- **Podnikový proces** (Business Process) – samotný proces zobrazený bez vnitřní logiky jako „černá skříňka“
- **Cíl** (Goal) – představuje žádoucí výsledek po dokončení běhu procesu. Cíl je důvodem proč je konkrétní proces v podniku prováděn (např. spokojenost zákazníka).
- **Informace** (Information) – řídí běh procesu. Informace jsou nezbytné ke korektnímu provedení a dokončení jednotlivých aktivit.
- **Zdroj** (Resource) – je využíván procesem jako vstup, který je na rozdíl od informace v průběhu zpracování spotřebován (např. všechny druhy surovin, lidská práce, apod.).

- **Výstup** (Output) – procesy obvykle produkují jeden nebo více výstupů s hodnotou pro zákazníka procesu. Výstupy jednoho procesu mohou sloužit jako vstupy jiného procesu v podobě informace nebo zdroje.
- **Událost** (Event) – označují předání nějakého objektu konkrétnímu procesu ke zpracování. Proces může být událostí spuštěn (vstupní událost), nebo může událost vyvolat a spustit tak jiný proces (výstupní událost). Událost může spouštět také externí aktér (Actor - člověk nebo systém)[23].

Obrázek 10: Prvky notace Eriksson – Penker [23]



## 4 – Metody a techniky analýzy a návrhu IS

První část této kapitoly je věnována vývoji metod analýzy a návrhu informačních systémů. Nejprve jsou charakterizovány klasické strukturované metody v čele s Yourdonovou moderní strukturovanou analýzou. Následuje představení vývoje objektově orientovaných metod, které vyvrcholilo vznikem jazyka UML. Principy a vlastnostmi uvedeného jazyka se zabývá druhá část této kapitoly.

### 4.1 – Vývoj metod analýzy a návrhu IS

Vývoj metod analýzy a návrhu softwarových produktů resp. informačních systémů úzce souvisí s vývojem programovacích technik. Nejstarší strukturované technologie programování se do podvědomí dostala v průběhu 60. let 20. století a masově se rozšířila v průběhu 80. let s rozvojem osobních počítačů. Uvedený vývoj s sebou také přinesl obrovský nárůst produkce softwarových aplikací, což postupně vyvolalo snahu o jejich rychlejší a především efektivnější tvorbu[4]. Za tímto účelem se od počátku 70. let začaly přirozeně formovat první strukturované metody analýzy a návrhu software s jejichž modifikacemi se lze setkat i v současné době. Od 90. let však začíná stoupat obliba objektově orientovaného přístupu k programování a s ním také přichází rozvoj objektově orientovaných metod analýzy a návrhu, které klasický strukturovaný přístup začaly postupně nahrazovat.

#### 4.1.1 – Strukturované techniky

Za čtyři desetiletí vývoje vznikla celá řada více či méně úspěšných metod strukturované analýzy a návrhu. Mezi nejvýznamnější zástupce tohoto tzv. klasického přístupu patří zejména následující metody:

##### **Strukturovaná analýza a specifikace systému (SASS)**

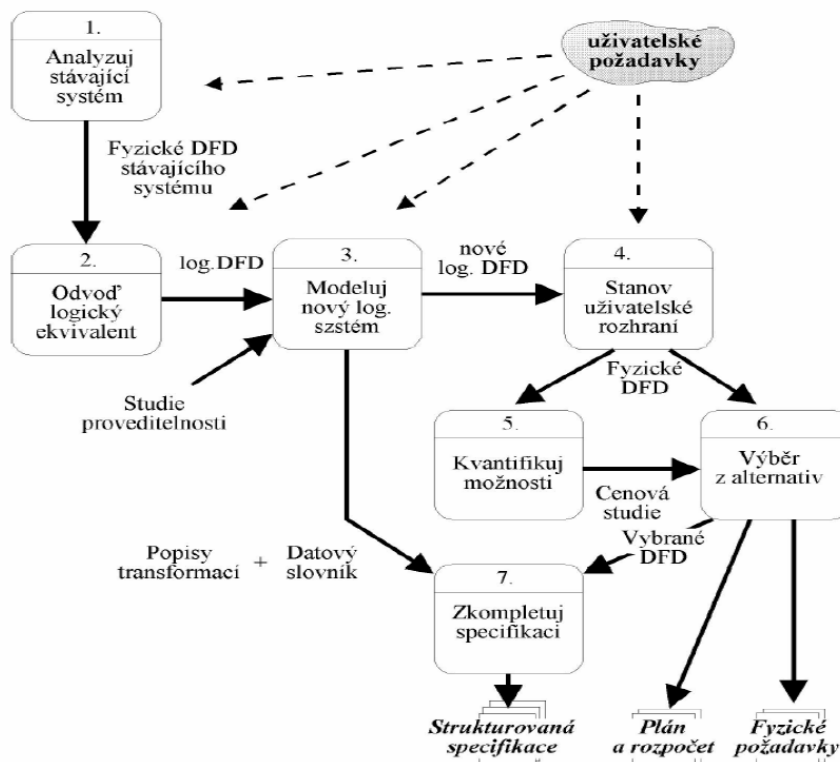
Metodu SASS zveřejnil v roce 1979 Tom DeMarco v publikaci s názvem *Structured analysis and system specification*. Základem jeho přístupu je diagram datových toků (data flow diagram - DFD) umožňující zobrazit systém jako síť procesů, které plní určené funkce a předávají si mezi sebou data – DFD tedy poskytuje funkčně orientovaný pohled na systém. SASS využívá 4 typy DFD diagramů:

- fyzický DFD stávajícího systému – nabízí pohled na stávající podobu systému
- logický DFD stávajícího systému – znázorňuje logickou strukturu stávajícího systému

- logický DFD nového systému – zachycuje navržené změny v logické struktuře
- fyzický DFD nového systému – znázorňuje jak bude nový systém implementován

V DeMarcově metodě je dále využit datový slovník, strukturovaná angličtina, rozhodovací tabulky a stromy. Uvedené nástroje spolu s DFD vytváří tzv. Strukturovanou specifikaci systému[18]. Princip její tvorby v rámci SASS je uveden na obrázku 11.

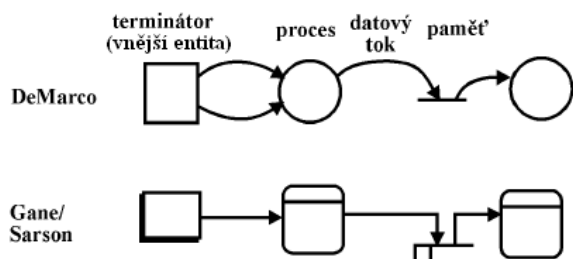
Obrázek 11: *Princip SASS* [18]



### Logické modelování

Tato metoda byla zveřejněna stejně jako SASS v roce 1979. Jejími autory jsou Chris Gane a Trish Sarson. Logické modelování rovněž využívá pro znázornění funkčně orientovaného pohledu na systém diagram DFD. Autoři pro jeho jednotlivé prvky pouze zvolili jinou notaci než DeMarco ve svém přístupu – viz obr. 12. Narozdíl od SASS využívá logické modelování navíc entitně-relační diagram (entity relationship diagram - ERD), který slouží pro zachycení datové struktury systému. ERD tedy modeluje neměnné atributy systému a slouží jako stabilní základ procesního modelu. Tento typ diagramu se stal později součástí řady dalších metodik.

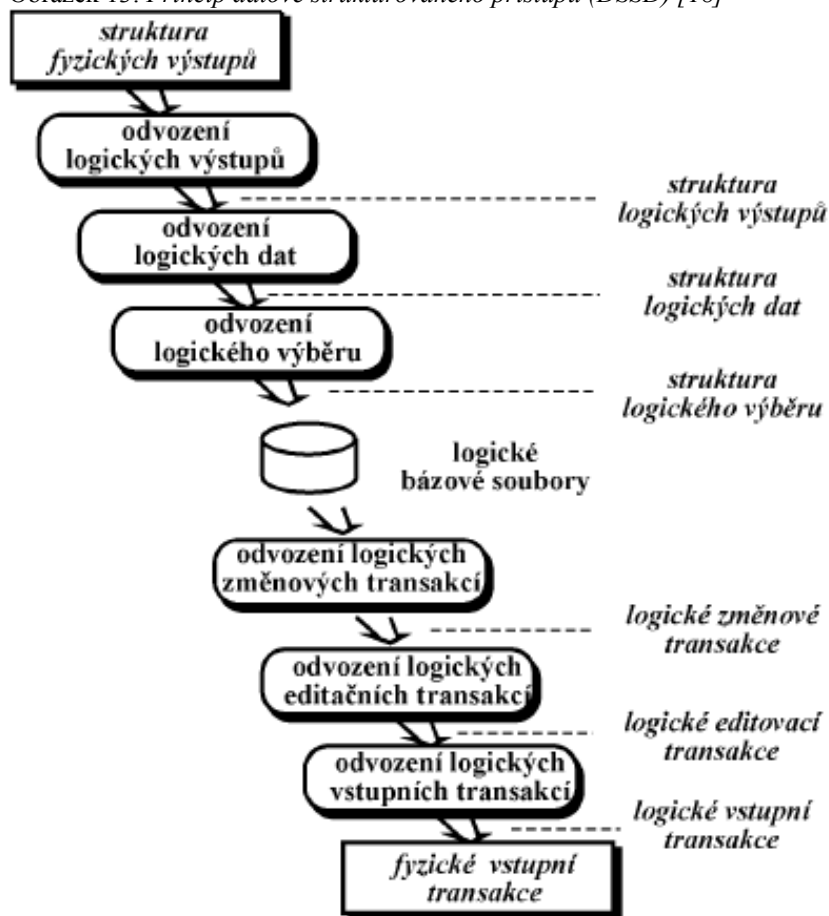
Obrázek 12: *Notace pro DFD [upraveno dle 18]*



### Datově strukturovaný přístup (Data Structure Systems Development - DSSD)

Autoři Jean-Dominique Warnier a Kenneth Orr začali tento přístup vyvíjet již v roce 1972. Základní myšlenkou je tzv. návrh shora dolů. Při řešení problému se vychází z hierarchického členění na méně složité části propojené co nejjednoduššími vazbami, které jsou znázorňovány diagramy entit. Funkční stránku systému definují assembly-line diagramy[4]. Empirickým poznatkem autorů metody je skutečnost, že nejlepších výsledků je dosaženo v případech, kdy struktura programu odpovídá hierarchické struktuře datového modelu[18]. Princip metody je znázorněn na obrázku 13.

Obrázek 13: *Princip datově strukturovaného přístupu (DSSD) [18]*





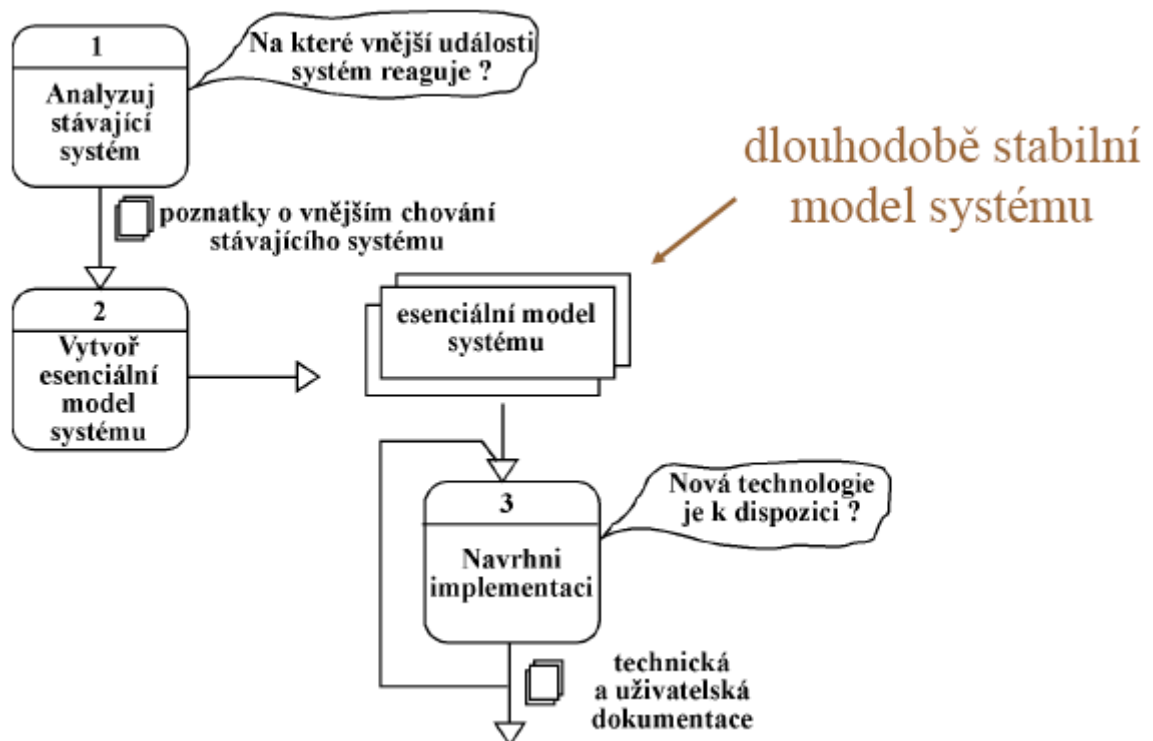
## Yourdonova moderní strukturovaná analýza (Yourdon Modern Structure Analysis - YMSA)

Strukturovaná analýza autora Edwarda Yourdona je pravděpodobně nejznámější strukturovanou metodikou analýzy a návrhu informačních systémů. YMSA je částečně založena na kritice DeMarcovy analýzy pomocí 4 modelů a zároveň je souhrnem myšlenek softwarového inženýrství od doby jeho vzniku[18][4]. Základním modelem YMSA je esenciální model, který vystihuje podstatu systému bez ohledu na implementační omezení, což je výhodné zejména pro komunikaci se zákazníkem. Tento model se dále dělí na dvě části:

- model okolí – popisuje rozhraní mezi systémem a okolním světem
- model chování – popisuje vnitřní část systému

Význam esenciálního modelu a rozdíl oproti DeMarcovu přístupu je znázorněn na obrázku 14. V jednotlivých fázích YMSA jsou mimo jiné použity diagramy datových toků (DFD), entitně-relační diaframy (ERD) a nově také stavové diagramy (state transition diagram - STD) vyjadřující závislost systému na čase[18].

Obrázek 14: *Princip YMSA* [18]



### 4.1.2 – Objektově orientované techniky

Na přelomu 80. a 90. let 20. století pokračuje trend zkracování vývojových, realizačních i produkčních cyklů softwarových aplikací, které je zapříčiněno jejich neustálým pronikáním téměř do všech složek lidských aktivit. S tímto vývojem jsou úzce spjaté snahy po dokonalejším, spolehlivějším a efektivněji vytvořeném programovém vybavení, čímž došlo k postupnému přechodu od strukturovaného k objektově orientovanému přístupu (dále jen OOP) k tvorbě softwarových produktů[9]. Zejména se to týká rozsáhlejších aplikací typu IS, u nichž je kladen zvýšený důraz na udržení nákladů na jejich tvorbu v přiměřené výši. Z počátku nacházely OOP uplatnění pouze v implementační fázi a pro fázi analýzy a návrhu byly využívány klasické strukturované metodiky. Během poměrně krátké doby se však začaly vyvíjet také metodiky pro objektově orientovanou analýzu a návrh IS[9]. Mezi nejnámější přístupy, které ovlivnily budoucí vývoj OOP k analýze a návrhu IS, patří:

#### OOD – Object-Oriented Design

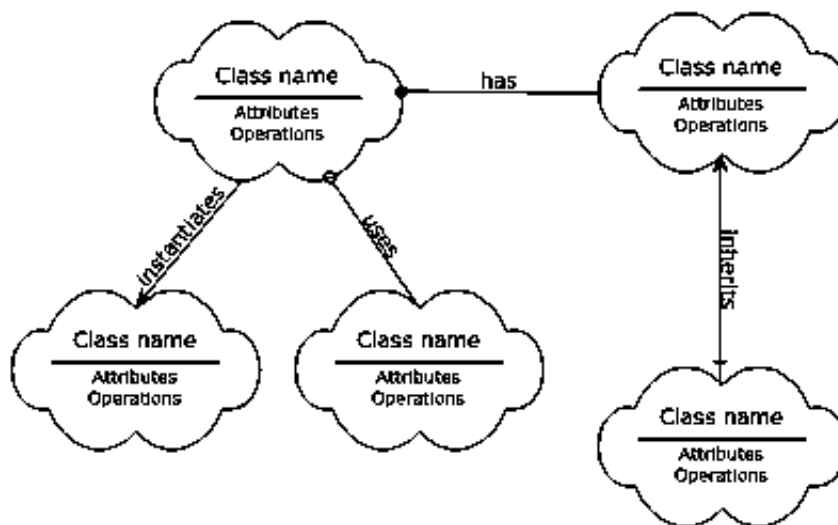
Metoda OOD7, jejímž autorem je Grady Booch, byla poprvé publikována v roce 1991 v knize „Object-Oriented Design with Applications“. Jde o poměrně komplexní metodiku, která byla určena pro týmový vývoj rozsáhlých aplikací v jazyce C++ a Smalltalk. [9]. Pro popis modelované reality je možné využít 6 různých typů diagramů: diagram tříd, diagram objektů, stavový diagram, diagram modulů, diagram procesorů<sup>8</sup> a diagram interakcí.[13]. Notace používaná v OOD se od ostatních přístupů liší zejména u diagramů pro popis statické struktury systému, tedy u diagramu tříd a diagramu objektů – viz obr. 15

---

<sup>7</sup> Metoda je také někdy označována zkratkou OOSD – Object-Oriented Software Development

<sup>8</sup> Z anglického Process diagram je překládáno jako diagram procesorů kvůli možné záměně s procesním diagramem business procesů. V rámci OOD je terminologie následující: proces = program, procesor = část hardware, která provádí program.

Obrázek 15: Diagram tříd v notaci OOD (u diagramu objektů je použito silnějšího ohraničení symbolu oblaku)[22]



### OMT – Object Modelling Technique

Metoda byla zveřejněna v roce 1991 v publikaci s názvem „Object-oriented Modelling and Design“. Za vznikem OMT stojí kolektiv autorů, z nichž nejznámější je James Rumbaugh. Dle této metody se v rámci analýzy vytváří 3 základní modely informačního systému:

- objektový model (tvořen diagramem tříd, resp. diagramem objektů)
- dynamický model (tvořen diagramem událostí a stavovým diagramem)
- funkční model (tvořen diagramem datových toků)

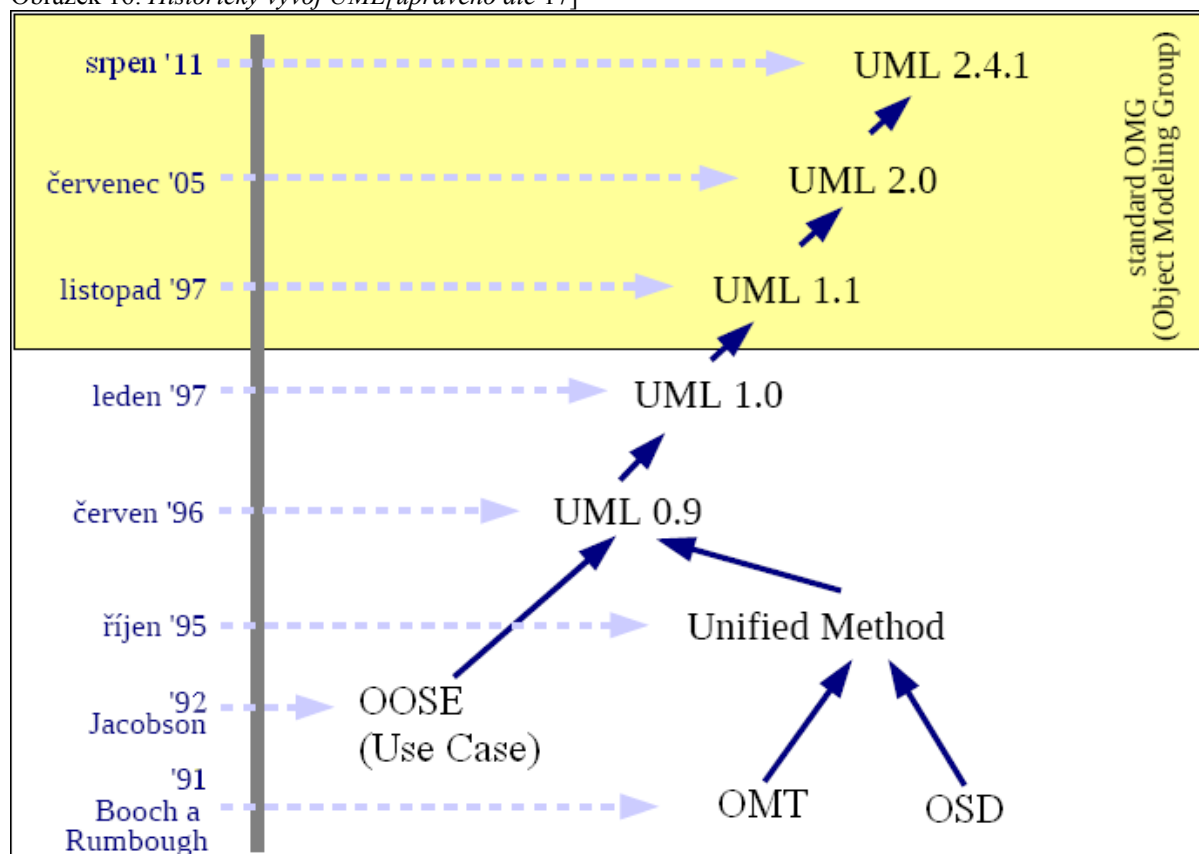
Ve fázi návrhu jsou uvedené diagramy dále doplňovány a optimalizovány z hlediska následné implementace.[13] Technika je zajímavá tím, že zahrnuje jak vybrané objektové, tak i klasické nástroje (např. DFD pro funkční model). Z tohoto důvodu ji nelze označit za ryze objektově orientovaný přístup, ale částečně přístup hybridní.[9]

### OOSE – Object-Oriented Software engineering

Metoda Object-Oriented Software engineering byla poprvé publikována ve stejnojmenné knize v roce 1992. Autorem metody, která je rovněž známá pod názvem „Objectory“, je Švéd Ivar Jacobson. OOSE je prvním přístupem, který se zabývá problematikou získávání a modelování informací v úvodních fázích životního cyklu softwarového produktu ještě před tvorbou konceptuálního diagramu.[9] V těchto úvodních fázích je vytvářen tzv. use case diagram zachycující interakci mezi aktéry (osoby nebo jiné objekty) a systémem. Technika use case byla později adoptována i do ostatních objektových metodik a je dodnes používána.[9]

Výše uvedené metodiky obsahují velké množství společných rysů, ale zároveň má každá z nich i svá specifika a také navzájem různé grafické notace. Z těchto důvodů se postupem času začaly projevovat snahy o určité sjednocení OOP k analýze a návrhu informačních systémů. Nejprve došlo v roce 1995 ke spojení myšlenek Boocha, Rumbaugh a několika dalších autorů na platformě OMT[9], což dalo vzniknout metodě nazvané jednoduše Unified Method (UM). O rok později byly k UM navíc připojeny některé nástroje používané v OOSE Ivary Jacobsna a vznikl tak první návrh Unifikovaného modelovacího jazyka UML, který je v současnosti doporučovaným průmyslovým standardem pro notaci diagramů pro OOP[9]. Nutno však poznamenat, že UML je pouze grafický jazyk, k němuž neexistuje odpovídající standardizovaná metodika a tak se za metodiku mnohdy považuje samotná znalost UML[9]. Vývoj UML je znázorněn na obrázku 16.

Obrázek 16: Historický vývoj UML [upraveno dle 17]



## Unified Process (UP)

Při absenci standardu metodiky objektově orientovaného přístupu k tvorbě softwarových produktů je UP v současnosti metodikou nejpoužívanější. Za jejím vznikem a rozšířením stojí již uvedení autoři jazyka UML Booch, Rumbaugh a Jacobson. Kořeny metodiky však sahají až do roku 1967, kdy vznikl tzv. Ericssonův model, jehož pojetí vycházelo z faktu, že se složité systémy mají modelovat jako množiny vzájemně propojených bloků.[17]

Základní myšlenkou UP je rozdělení komplexního problému na řadu jednodušších činností. Každá tato činnost je považována za iteraci, v rámci které existuje 5 základních pracovních postupů:

- **Požadavky** – zachycují to, co by měl systém dělat
- **Analýza** – vybroušení požadavků a jejich strukturování
- **Návrh** – realizace požadavků v architektuře systému
- **Implementace** – tvorba softwaru
- **Testování** – ověření, zda implementace funguje dle očekávání [17]

Přestože může každá iterace obsahovat všech pět základních pracovních postupů, závisí důraz kladený na určitý pracovní postup na jeho umístění v životním cyklu daného projektu[17]

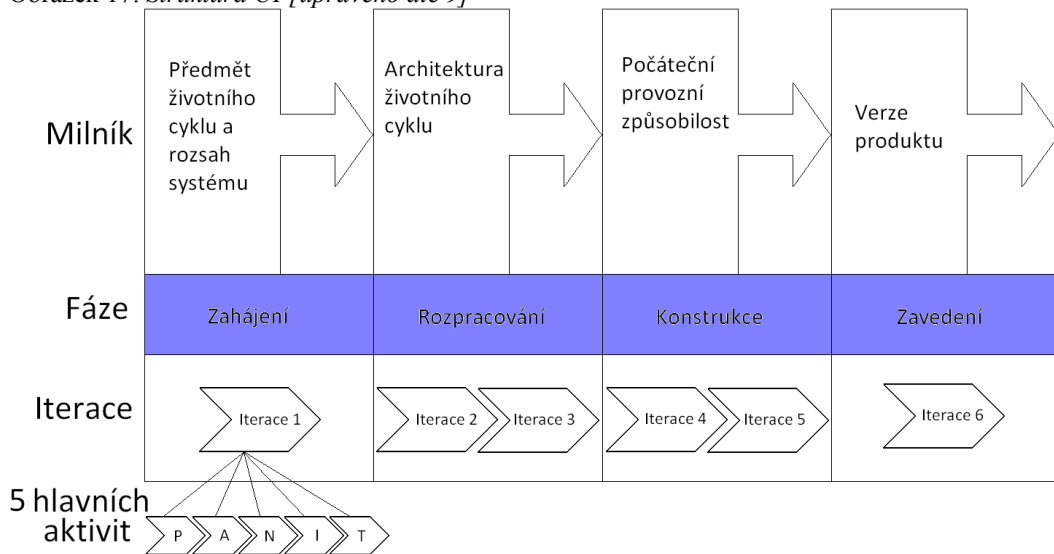
Dle UP je životní cyklus projektu rozčleněn na čtyři po sobě následující fáze:

- **Zahájení** – hlavní důraz je v této fázi kladen na pracovní postupy zabývající se specifikací požadavků a jejich analýzou. Ve fázi zahájení obvykle nedochází k testování
- **Rozpracování** – v této fázi nabývají na důležitosti pracovní postupy zabývající se požadavky, analýzou a návrhem. Implementace se uplatňuje s blížícím se koncem fáze rozpracování při tvorbě spustitelného architektonického základu.
- **Konstrukce** – cílem konstrukční fáze je splnit všechny požadavky analýzy a návrhu a vyvinout za základu získaného z předchozí etapy konečnou verzi systému. Klíčovým zadáním konstrukční fáze je zachování integrity architektury vytvářeného systému.
- **Zavedení** – v této fázi je důraz kladen zejména na implementaci a testování. Již by neměly vznikat nové požadavky a neměla by se provádět analýza.[9]

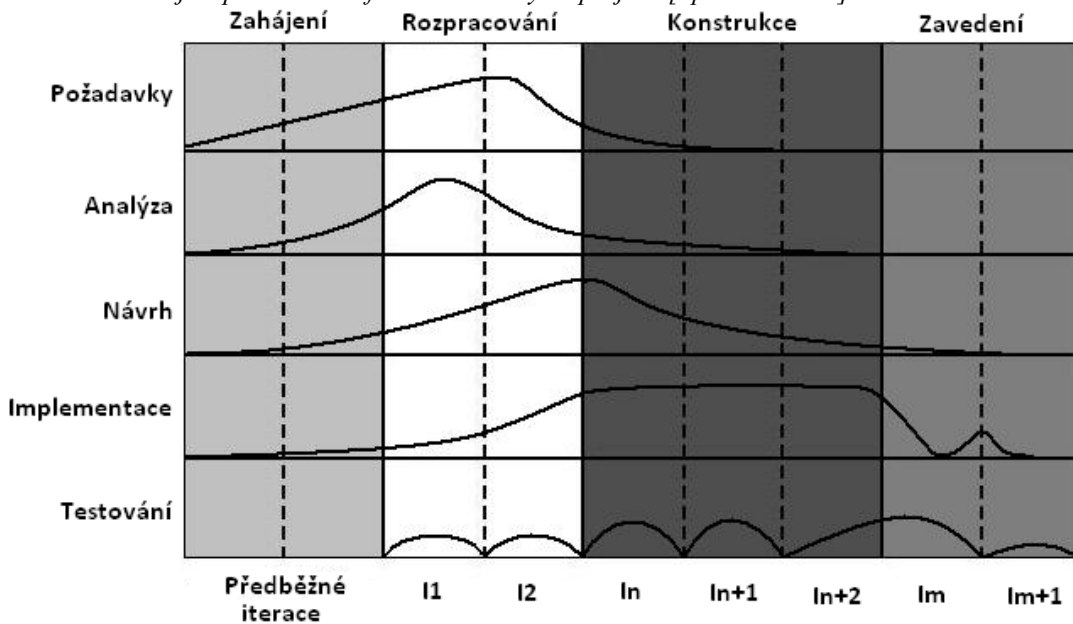
Vztah mezi iteracemi základních pracovních postupů a fázemi životního cyklu je uveden na obrázku 17 představujícím strukturu metodiky UP. Objem práce vynaložené při základních pracovních postupech během jednotlivých fází životního cyklu projektu charakterizuje obrázek 18.

Závěrem je ještě vhodné uvést, že UP je obecnou metodikou tvorby softwaru. Znamená to tedy, že pro každou organizaci a pro každý projekt je třeba vytvořit její novou instanci, protože každý softwarový projekt se od ostatních liší.[17]

Obrázek 17: *Struktura UP[upraveno dle 9]*



Obrázek 18: *Objem práce během fází životního cyklu projektu[upraveno dle 1]*



## **4.2 – UML (Unified Modelling Language)**

Jazyk UML byl vytvořen jako univerzální standard pro vizuální modelování systémů. Přestože je nejčastěji spojován s modelováním objektově orientovaných softwarových systémů, tak má mnohem širší využití, což je možno pomocí zabudovaných rozšiřovacích systémů.[9]

Z předchozí kapitoly vyplývá, že jazyk UML byl navržen, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství. Jak již bylo uvedeno výše jazyk UML poskytuje pouze vizuální syntaxi, ale jeho definice neobsahuje žádný druh metodiky. Dle [9] UML navíc velmi špatně podporuje první fáze analýzy informačních systémů, kdy je třeba modelovat business kontext budované aplikace.

### **4.2.1 – Mechanismy UML**

Jazyk UML obsahuje čtyři společné mechanismy používané v celém jazyku konzistentně. Popisují čtyři strategie cesty k modelování objektů, jež jsou opakovatelně používány v různých kontextech v celém jazyce UML.[17] Těmito mechanismy jsou specifikace, ornamenty, podskupiny a mechanismy rozšiřitelnosti. Jejich charakteristika je uvedena následně:

#### **Specifikace**

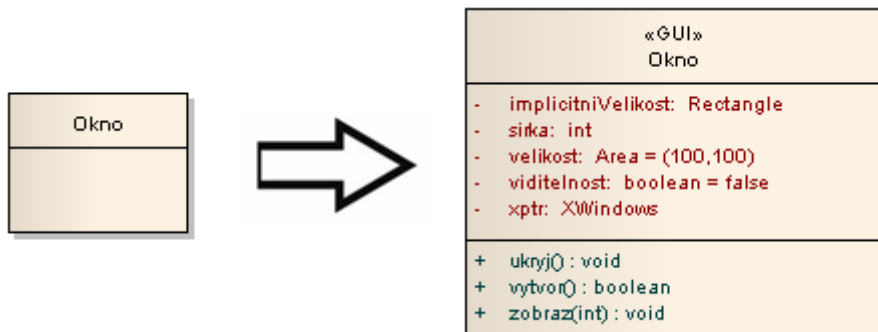
Modely UML mají alespoň dva rozměry – grafický, který umožňuje vizualizovat model prostřednictvím diagramů a symbolů, a textový jenž se skládá ze specifikací různých prvků modelu. Specifikace jsou textovým popisem sémantiky jednotlivých prvků.[17] Specifikace tedy dává jednotlivým diagramům a ostatním prvkům modelu smysl. Model, který je tvořen pouze diagramy bez příslušné specifikace nelze považovat za kompletní.

#### **Ornamenty**

Jednou z vlastností jazyka UML je, že elementy modelu jsou vyjádřeny jednoduchým symbolem, ke kterému lze přidávat mnoho „ozdob“ (ornamentů) a tím ho obohacovat o doplňující informace.[9] Na obrázku 19 je uveden příklad pro třídu s názvem Okno nejprve bez ornamentů a poté s ornamenty. Obvyklým postupem modelování v UML je, že jsou ornamenty přidávány k jednotlivým prvkům modelu až v průběhu jeho upřesňování. Ozdoby coby upřesňující informace má smysl do modelu přidávat pouze v případě, že

zvyšují čitelnost nebo srozumitelnost diagramů nebo zdůrazňují nějakou důležitou funkci.[9] Neplatí tedy, že čím více ornamentů model obsahuje, tím lépe.

Obrázek 19:Ornamenty [upraveno dle 9]



### Podskupiny

Podskupiny popisují různé způsoby pohledu na prvky systému. V UML rozlišujeme dvě takové podskupiny – skupinu klasifikátorů a instancí a skupinu rozhraní a implementací.

- **Klasifikátory a instance** - předpokladem jazyka UML je, že můžeme mít abstraktní představu o typu předmětu, ale i představu o konkrétní instanci této abstrakce. Abstraktní představa o předmětu je klasifikátor a konkrétní představy jsou jeho instance. Ilustrací tohoto vztahu je třída objektů jako klasifikátor a objekt jako instance.[9]
- **Rozhraní a implementace** - jazyk UML vychází ze zásady oddělení toho, co předmět vykonává (jeho rozhraní) od toho jak to vykonává (jeho implementace). Rozhraní definuje dohodu, která zaručuje, čím se budou jednotlivé implementace řídit.[9]

### Mechanismy rozšiřitelnosti

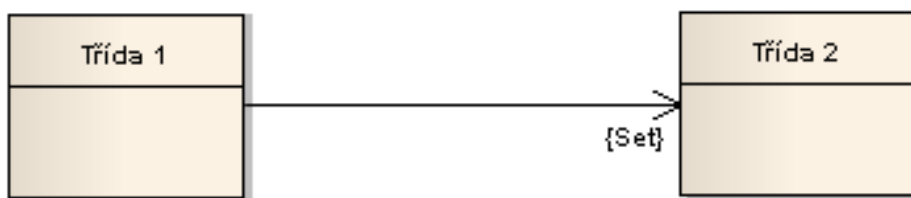
Tyto mechanismy slouží jazyku UML pro zachování jeho univerzálnosti. Bez možnosti rozšiřitelnosti by UML mohl jen stěží uspokojit potřeby všech uživatelů. Z tohoto důvodu byly do jazyka zahrnuty mechanismy omezení, stereotypy a označené hodnoty, umožňující jeho rozšiřitelnost.

- **Omezení (constraints)** - představují omezující podmínky, které rozšiřují sémantiku daného elementu tím, že k němu umožňují přidávat nová pravidla. Omezující podmínka je textový řetězec uzavřený do složených závorek {}. Tento text specifikuje podmínku nebo pravidlo, které musí být pravdivě vyhodnoceno. Tímto způsobem lze omezit určité chování daného elementu.[9]



- **Stereotypy** - umožňují vytvářet nové elementy modelu založené na stávajících elementech. Stereotypem jsou do UML zaváděny nové elementy, ke kterým je rovněž třeba definovat jejich sémantiku. Ke každému stereotypu je také možné přidružit samostatný symbol, barvu nebo texturu. Tímto způsobem lze tedy rozšířit grafickou notaci UML.[9][17] Uvedený postup se používá často např. v diagramu nasazení při vizualizaci jednotlivých komponent (osobní počítače, tiskárny, servery apod.).
- **Označené hodnoty – používají se pro přidání nové vlastnosti elementu, která není předdefinovaná standardem UML.** Příklad označené hodnoty je uveden na obrázku 20, kde je zobrazen vztah mezi dvěma třídami. Označená hodnota {set} u jedné z tříd určuje, že instance této třídy jsou uloženy v kolekci typu množina, což eliminuje možnost uložení duplicitních prvků.

Obrázek 20: Příklad označené hodnoty[upraveno dle 9]

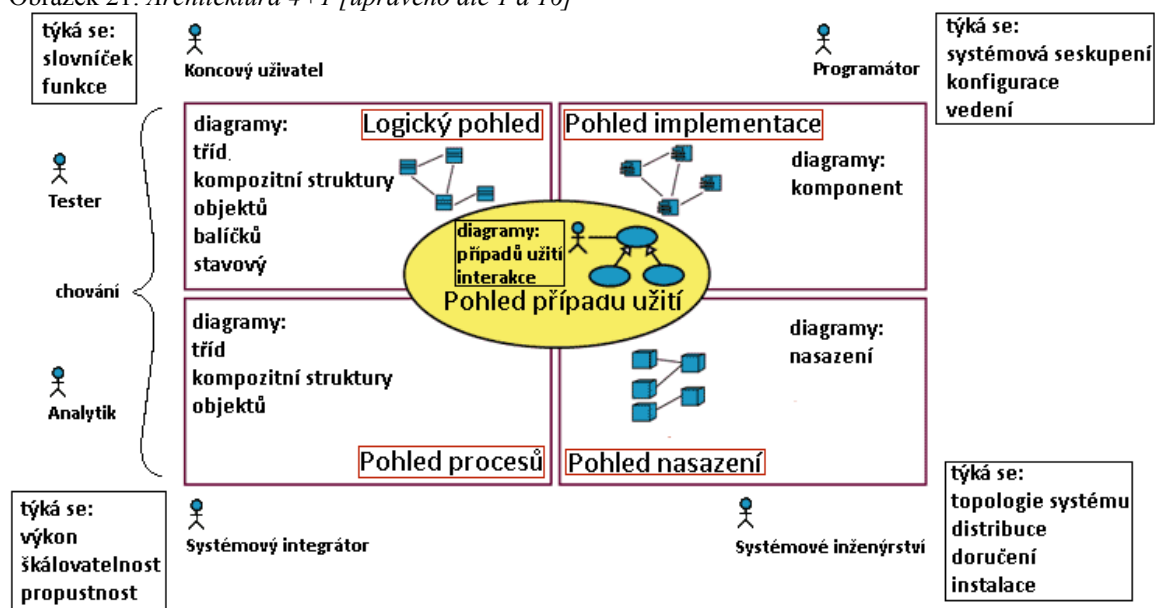


#### 4.2.2 – UML a architektura modelovaného systému

Architektura je organizační struktura systému, včetně jeho rozkladů na součásti, jeho propojitelnosti, interakce, mechanismů a směrných zásad, která proniká do návrhu systému [Rumbaugh at all 1998 dle 9]

Pro zachycení všech podstatných aspektů architektury modelovaného systému jazyk UML definuje čtyři různé pohledy na systém: logický pohled, pohled procesů, pohled implementace a pohled nasazení. Všechny tyto pohledy jsou dále integrovány do pátého pohledu, jíž je pohled případů užití[17] (tzv. architektura 4+1 – viz obr. 21)

Obrázek 21: *Architektura 4+1 [upraveno dle 1 a 16]*



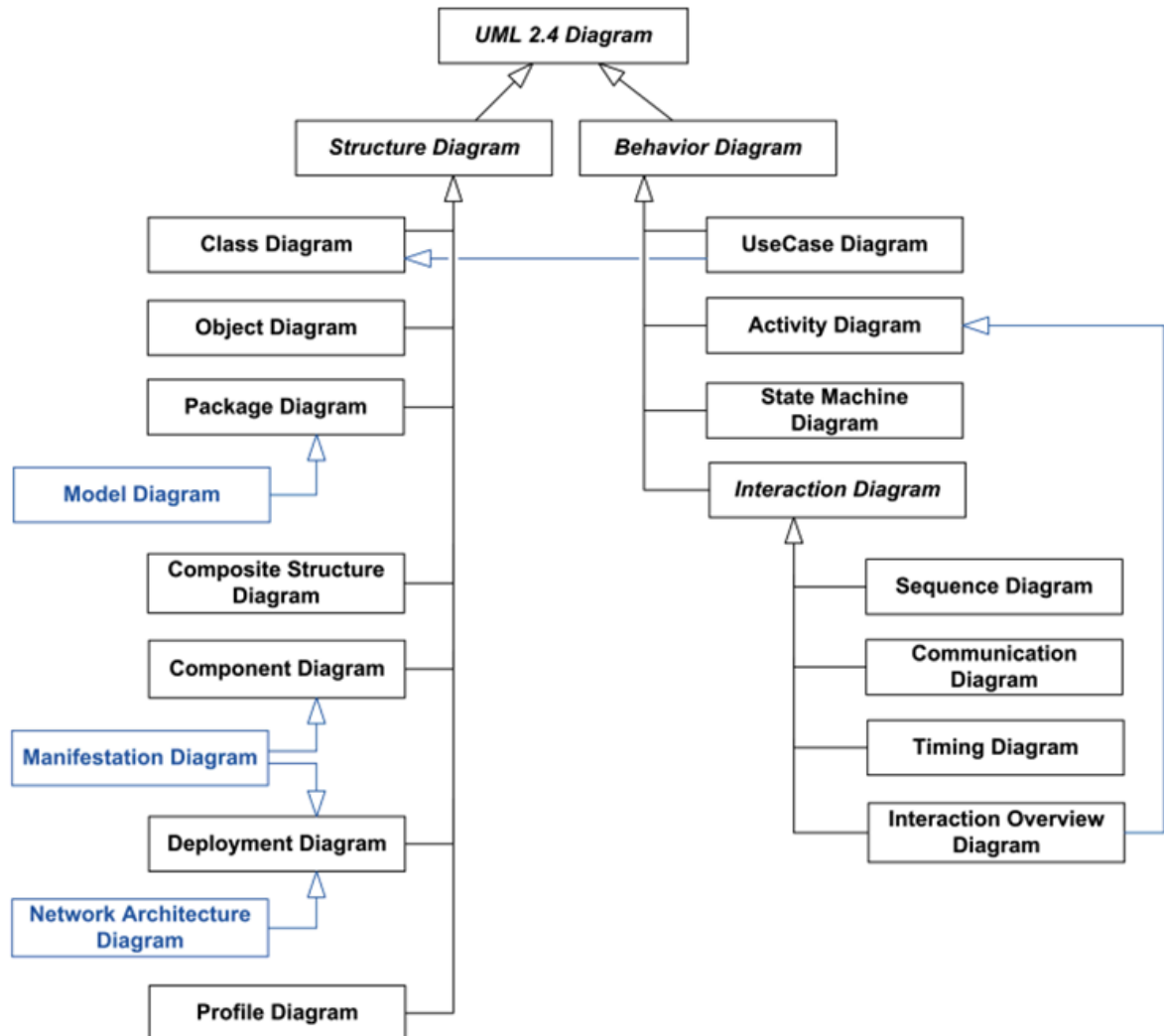
- **Logický pohled** – zachycuje slovník oblasti problému jako množinu tříd a objektů. Důraz je přitom kladen především na zobrazení způsobu, jakým objekty a třídy tvořící základ systému implementují jeho chování
- **Pohled procesů** – modeluje spustitelná vlákna a procesy jako aktivní třídy. Je to procesně orientovaná varianta logického pohledu, která obsahuje stejné artefakty.
- **Pohled implementace** – modeluje soubory a komponenty, které utvářejí hotový kód systému. Slouží jednak ke znázornění závislosti mezi komponentami, a také toho, jak spravovat konfiguraci množin vytvořených z těchto komponent. Umožňuje definici verze systému.
- **Pohled nasazení** – modeluje fyzické nasazení komponent na fyzické uzly systému (počítače a periferní zařízení). Umožňuje modelování distribuce komponent na příslušné uzly distribuovaného systému.
- **Pohled případů užítí** – Všechny výše uvedené pohledy jsou odvozeny z pohledu případu užítí. Tento pohled zachycuje základní požadavky kladené na příslušný systém.

#### 4.2.3 – Diagramy UML

Smyslem této kapitoly je poskytnout přehled a stručnou charakteristiku diagramů pro popis modelované reality, které jazyk UML nabízí. Vzhledem k obsáhlosti problematiky zde není možné jednotlivé diagramy charakterizovat vyčerpávajícím způsobem.

Diagramy UML lze rozdělit podle pohledu na model, který poskytují, na diagramy struktury (statický pohled) a diagramy chování (dynamický pohled). Základní rozdělení diagramů UML 2.4. je uveden na následujícím obrázku.

Obrázek 22: Rozdělení diagramů UML 2.4 [15]



**Diagramy struktury** – poskytují statický pohled na model systému

- **diagram tříd a objektů** (class diagram, object diagram) popisují statickou strukturu systému, znázorňují datové struktury a operace objektů a souvislosti mezi objekty. Znázorňují datový model systému od konceptuální úrovně až po implementaci.[7]
- **diagram komponent** (component diagram) popisuje rozdělení systému na funkční celky (komponenty) a definuje náplň jednotlivých komponent a jejich vztahy.[7] Rovněž může (stejně jako diagram nasazení) zobrazovat vnitřní implementaci komponent.[15] (na obrázku 22 zobrazen jako Manifestation diagram).

- **diagram nasazení** (deployment diagram) – popisuje umístění funkčních celků (komponent) na výpočetních uzlech informačního systému.[7] Tento typ diagramu může být použit rovněž pro zobrazení logické nebo fyzické síťové architektury systému [15](na obrázku 22 zobrazen jako Network Architecture Diagram).
- **diagram kompozitní struktury** (composite structure diagram) – zobrazuje spolupráci prvků systému na uskutečnění komplexního cíle. Tento typ diagramu může být použit pro zobrazení interní struktury klasifikátoru, interakce klasifikátoru s okolím nebo chování spolupráce uvnitř klasifikátoru s vlastností chování.[15]
- **diagram balíčků** (package diagram) – popisuje vztahy mezi balíčky. Balíček představuje abstrakci sdružování – slouží pro logické seskupování elementů modelu do sémanticky konzistentních skupin.[1] Speciálním diagramem balíčků je tzv. diagram modelu (model diagram), který dovoluje zobrazit celý systém z jednoho specifického pohledu. Používá se např. pro popis systémů s vícevrstevnou architekturou.[15]
- **diagram profilů** (profile diagram) – pomocný diagram umožňující definovat vlastní mechanismy rozšiřitelnosti[15] - viz. kapitola 4.2.1

#### **Diagramy chování** - poskytují pohled na dynamickou stránku modelu

- **diagram případů užití** (use case diagram) – dokumentuje možné případy použití systému, vyvolané událostmi, na které musí systém reagovat.[7]
- **diagram aktivit** (activity diagram) – popisuje průběh procesu v systému nebo činnosti.[7] Lze jej využít i pro modelování business procesů – viz kapitola 3.5.4
- **stavový diagram** (state machine diagram) – popisuje dynamické chování objektu nebo systému, možné stavy a přechody mezi nimi [7]
- **sekvenční diagram** (sequence diagram) – spolu s diagramy přehledu interakcí, spolupráce a časování patří mezi diagramy interakce. Sekvenční diagram slouží k zachycení průběhu případu užití a je zaměřen zejména na znázornění časových závislostí.[9]
- **diagram spolupráce** (communication diagram) – stejně jako sekvenční diagram slouží diagram spolupráce k zachycení průběhu případu užití (tyto 2 diagramy lze

navzájem transformovat). Tento diagram se však více zaměřuje na vyjádření vzájemných vztahů mezi objekty, než na časovou posloupnost.[9]

- **diagram časování** (timing diagram) – slouží ke zdůraznění aspektů souvisejících s načasováním interakcí. Jejich hlavní úlohou je znázornění významu toku času a časových omezení. Tento typ diagramu se uplatňuje zejména u modelování systémů pracujících v reálném čase.[1]
- **diagram přehledu interakcí**<sup>9</sup> (interaction overview diagram) – představuje zvláštní formu diagramu aktivit pro znázornění interakcí a jejich výskytů. Používá se k modelování obecných přechodů mezi interakcemi. Tento typ diagramu lze využít např. pro znázornění cest mezi případy užití [1]

---

<sup>9</sup> někdy také označovaný jako diagram zjednodušené interakce

## 5 – Analýza a návrh informačního systému

V praktické části této práce se autor zabývá analýzou zvoleného podnikového procesu, pro jehož zefektivnění je následně navržen informační systém. Zvolen byl proces zpracování rozpisů hromadných plateb v Úseku správy příspěvků penzijního fondu (dále jen „ÚSP“). V úvodu kapitoly je nejprve činnost ÚSP charakterizována jako celek, v němž je poté uvedený proces identifikován. Následuje podrobnější analýza průběhu zpracování rozpisů, na základě které jsou zjištěny problematické vlastnosti procesu. Další část kapitoly obsahuje formulaci a následnou specifikaci požadavků na systém, které vyplývají především ze zjištěných vlastností stávajícího procesu. Závěr kapitoly je poté vymezen pro návrh statické struktury systému v podobě UML diagramu tříd.

### 5.1 – Agenda Úseku správy příspěvků

ÚSP zabezpečuje kompletní agendu spojenou se zpracováním příchozích plateb na účet penzijního fondu. Téměř veškeré operace spojené s činností ÚSP se provádějí v informačním systému (dále jen „ISPF“), který byl vytvořen speciálně pro potřeby penzijního fondu. Zpracováním platby je v této souvislosti nazýván proces, jehož průběh začíná přijetím platby (načtení informací z bankovního výpisu o přijaté platbě do databáze ISPF) a končí připsáním příspěvku resp. příspěvků ve správné výši správnému klientovi resp. klientům (uložení nezbytných informací o identifikaci příspěvků odpovídajících přijaté platbě do databáze ISPF). V rámci své působnosti ÚSP také zabezpečuje vrácení plateb, které nemohly být z jakéhokoliv důvodu připsány klientům.

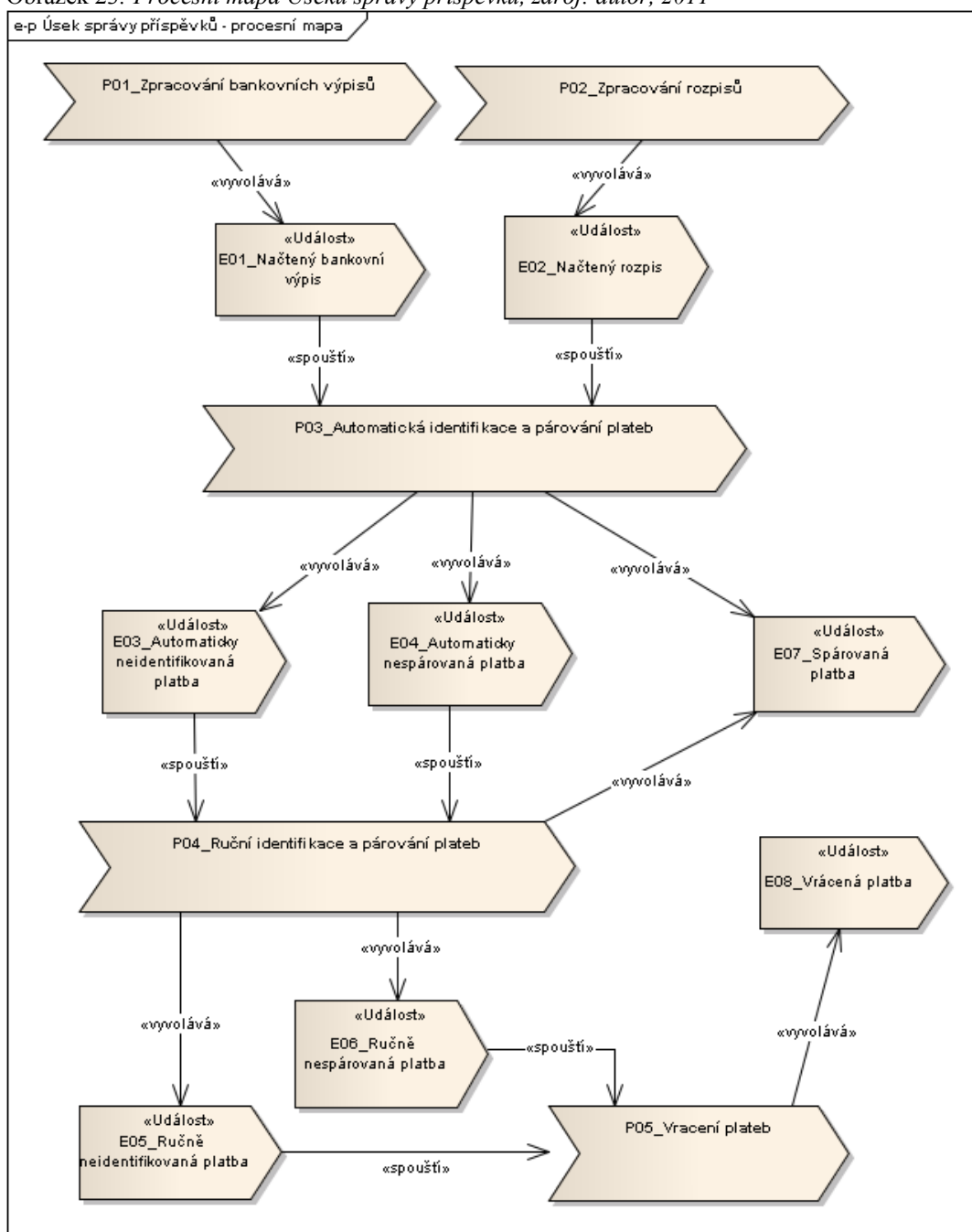
V souladu s metodikou modelování a analýzy podnikových procesů MMABP byla v ÚSP identifikována následující uspořádání vnějších událostí a odpovídajících reakcí:

Tabulka 3: *Vnější události a reakce v ÚSP, zdroj: autor, 2011*

<b>Událost</b>	<b>Reakce</b>
doručen bankovní výpis	načtení výpisu do ISPF
zjištěna příchozí individuální platba, zjištěna příchozí hromadná platba a je k dispozici její rozpis	zpracování platby
doručen rozpis hromadné platby	zpracování rozpisu
nastal termín vrácení příspěvků	odeslání nezpracovatelných příspěvků

Z uspořádání ve výše uvedené tabulce jsou dále odvozeny elementární procesy probíhající v ÚSP, jejichž vzájemné vztahy vycházejí z dalšího kroku metodiky MMABP, kterým je specifikace elementárních procesů (viz kapitola 5.1.1). Výsledkem dosavadního postupu je celkový pohled na činnost ÚSP zachycený prostřednictvím procesní mapy v notaci Ericsson – Penker na obrázku 23. Přesto, že metodika MMBAP je určena především k analýze procesů organizace jako celku, lze zejména její úvodní fázi využít i pro analýzu procesů menšího funkčního útvaru.

Obrázek 23: *Procesní mapa Úseku správy příspěvků, zdroj: autor, 2011*



### 5.1.1 – Procesy v Úseku správy příspěvků

- **P01\_Zpracování bankovních výpisů** – tento proces zabezpečuje načtení informací o veškerých přijatých platbách na účet penzijního fondu. Tyto informace jsou do systému zadávány jednou denně vždy za předchozí pracovní den. Pro potřeby této práce budou uvažovány pouze individuální a hromadné platby, kterými jsou hrazeny příspěvky účastníků – více v kapitole 5.1.2.
- **P02\_Zpracování rozpisů** – tento proces zabezpečuje přijetí, zpracování a nahrání rozpisů příspěvků, které byly zaslány ve formě hromadné platby, do IS. Procesy P01 a P02 jsou navzájem nezávislé. Tzn. rozpis může být do systému nahrán bez ohledu na to, jestli již byla přijata příslušná platba. Rozpisy jsou charakterizovány v kapitole 5.1.3.
- **P03\_Automatická identifikace a párování plateb** – v IS probíhá tento proces v předem naplánovaných nepravidelných intervalech (cca 2 x týdně). Jeho cílem je připsání příspěvků ve správné výši správným klientům. Identifikace znamená určení klienta, kterému příspěvek náleží. Spárováním se nazývá skutečné připsání příspěvku klientovi zjištěnému při identifikaci.
- **P04\_Ruční identifikace a párování plateb** – pokud nemohl být příspěvek z nějakého důvodu identifikován (např. při chybném variabilním symbolu platby) nebo spárován (např. kvůli určitému nedostatku na smlouvě klienta) může se o identifikaci resp. spárování pokusit pracovník ÚSP s využitím příslušných funkcí ISPF.
- **P05\_Vrácení plateb** – v případě, že ani při ruční identifikaci a párování plateb není možné příspěvek správnému klientovi připsat dojde k jeho vrácení dle interních směrnic Penzijního fondu.



## 5.1.2 – Typy plateb

Na účet penzijního fondu je možné zasílat individuální nebo hromadné platby. Tyto varianty se od sebe liší svoji identifikací.

### Individuální platba

Jde o platbu, jejíž výše přesně odpovídá výši příspěvku, který je určen jedinému konkrétnímu účastníkovi<sup>10</sup>. Individuální platbou mohou hradit příspěvky účastníci, zaměstnavatelé i třetí osoby. Penzijní fond používá pro rozlišení druhu individuální platby následující identifikaci:

- **Variabilní symbol** – desetimístné číslo smlouvy účastníka
- **Specifický symbol** – v případě příspěvku účastníka se neuvádí  
0000666600 označuje příspěvek zaměstnavatele  
0000333300 označuje příspěvek třetí osoby

### Hromadná platba

Platba zaslaná v jedné částce (výjimečně i více) obsahující příspěvky určené zpravidla více než jednomu účastníkovi. K platbě tohoto typu je nezbytné doručit penzijnímu fondu její rozpis, podle něhož je možné platbu rozdělit na jednotlivé příspěvky a ty poté připsat ve správné výši příslušným účastníkům. Tento typ plateb je určen především pro úhradu příspěvků zasílaných zaměstnavateli účastníků, ale je možné ho použít také pro úhradu příspěvků třetí osoby – jednotlivé typy příspěvku jsou blíže specifikovány v kapitole 5.1.3. Identifikace hromadných plateb používaná v penzijním fondu je následující:

- **Variabilní symbol** – identifikační číslo zaměstnavatele (resp. třetí osoby)
- **Specifický symbol** – v případě příspěvku účastníka se neuvádí (platí také pro případ různých typů příspěvku v jedné hromadné platbě)  
0000666600 označuje příspěvek zaměstnavatele  
0000333300 označuje příspěvek třetí osoby

---

<sup>10</sup> Účastníkem je označován klient penzijního fondu, který má uzavřenou smlouvu o penzijním připojištění. Toto označení vychází ze Zákona č.42/1994 Sb. o penzijním připojištění.

### **5.1.3 – Typy příspěvků**

Rozlišení příspěvků na 3 různé typy vychází ze Zákona č. 42/1994 Sb. o penzijním připojištění. Uvedený legislativní předpis definuje příspěvky účastníka, příspěvky zaměstnavatele a příspěvky třetí osoby. Od tohoto rozdělení se dále odvíjí fiskální efekt pro příjemce i zasílatele daných příspěvků

#### **Příspěvky účastníka**

Příspěvky, které si účastník (klient) hradí sám z vlastních prostředků. Výše příspěvku za jeden kalendářní měsíc je dána smlouvou o penzijním připojištění uzavřenou mezi účastníkem a penzijním fondem.

#### **Příspěvek zaměstnavatele**

Příspěvky zasílané jednotlivým klientům jejich zaměstnavatelem, který také určuje jejich výši. Zaměstnavatelé jsou k poskytování příspěvku svým zaměstnancům motivováni daňovými úlevami definovanými příslušnou legislativou.<sup>11</sup>

#### **Příspěvek třetí osoby**

Příspěvek, který klientovi hradí jiná osoba (fyzická nebo právnická). Příspěvek se svým charakterem v podstatě neliší od příspěvku účastníka.

### **5.1.4 – Rozpisy hromadných plateb**

V ISPF může být hromadná platba zpracována pouze na základě rozpisu příslušného dané platbě. ISPF přijímá rozpisy prostřednictvím importního adresáře na souborovém serveru v podobě souborů obsahujících prostý text se zakončením jednotlivých řádků sekvencí CR-LF (tedy soubory vytvořené v operačních systémech Windows). Tyto soubory dále musí mít svůj obsah strukturovaný dle formátu APF<sup>12</sup>. Struktura rozpisů dle formátu APF je uvedena v příloze. Další možností jak rozpis do systému zadat je jeho ruční natypování dle (většinou papírové) předlohy nebo prostřednictvím uživatelského rozhraní ISPF provedení kopie rozpisu, který v systému již existuje.

### **5.1.5 – Příklad rozpisu**

Pro ilustraci je níže uveden rozpis představující situaci, kdy zaměstnavatel s názvem Firma s.r.o., IČ 12345678 zaslal v průběhu měsíce listopadu 2011 svým dvěma zaměstnancům

---

<sup>11</sup> Zákon č. 586/1992 Sb. o daních z příjmů a Zákon č. 42/1994 Sb. o penzijním připojištění

<sup>12</sup> APF je formát definovaný Asociací penzijních fondů, podle jejíž zkratky je označován.

příspěvky hromadnou platbou v celkové výši 4000Kč. Jde o příspěvky účastníka sražené ze mzdy spolu s příspěvky zaměstnavatele.

```
S;12345678;12345678;Firma,s.r.o.;87654321;PFCR;3558;;12345678;4000;201111;1;4
U;1122334455;7512121111;Novák;Jan;1500;;
Z;1122334455;7512121111;Novák;Jan;500;;
U;9988775566;7211122222;Dvořáková;Petra;1000;;
Z;9988775566;7211122222;Dvořáková;Petra;1000;;
```

Z uvedeného příkladu vyplývá, že příspěvky v rámci jednoho rozpisu a tedy i jedné hromadné platby mohou být různých typů. K jejich rozlišení dochází na základě uvozujícího znaku jednotlivých řádků. Také je možné si všimnout, že v sumární větě není uvedena položka 9 – specifický symbol. Je to z toho důvodu, že penzijní fond pro identifikaci hromadné platby složené z různých typů příspěvků specifický symbol nepoužívá (resp. používá nevyplněný).

## 5.2 – Specifikace problému

ÚSP zpracovává měsíčně cca 3000 rozpisů, z nichž převážná většina (přibližně 95%) je přijímána e-mailem jako příloha v různých formátech. Zbývajících 5% rozpisů přichází do penzijního fondu v papírové podobě prostřednictvím České pošty. Pracovníci ÚSP jsou však motivováni ke snižování počtu rozpisů přijímaných v listinné podobě a lze předpokládat, že zastoupení těchto rozpisů bude mít i nadále klesající tendenci. Níže uvedená interní statistika společnosti udává podíl zastoupení jednotlivých formátů rozpisů přijímaných e-mailovou cestou.

**APF (cca 45%)** – výstup ze mzdových systémů zaměstnavatelů. Tyto rozpisy mohou být v řadě případů přímo přemístěny do importního adresáře ISPF. Často však jejich struktura zcela neodpovídá požadavkům ISPF, a proto je tyto rozpisy třeba upravit. Z tohoto důvodu jsou veškeré přijaté rozpisy APF kontrolovány pomocí SW vytvořeném IT oddělením penzijního fondu speciálně pro tento účel (dále jen SWPF).

**XLS (cca 40%)** – tabulka vytvořená v MS excel. V SWPF lze provést konverzi z xls do APF. Tato konverze však vyžaduje nezanedbatelné množství „manuálních“ úkonů, protože penzijní fond netrvá na jednotné podobě rozpisů ve formátu xls. Akceptována je jakákoliv tabulka, ze které lze jednoznačně určit rozdělení hromadné platby na samostatné příspěvky.

**DOC (cca 10%)** – rozpis v dokumentu MS Word v nejrůznějších podobách.. Často je lze také převést v SWPF do APF, ale jde o obtížnější převod než u XLS a zároveň je zde zvýšené riziko vzniku chyby.

**Ostatní formáty (cca 5%)** – mezi tyto formáty patří např. html, jpg a pdf. Obecně platí, že pokud lze z rozpisu v jakémkoliv formátu získat text, je možné ho s menšími či většími obtížemi převést v SWPF do APF, v opačném případě je postup téměř shodný jako při zpracování rozpisů v listinné podobě (viz následující kapitola).

Za základní problém v činnosti ÚSP byla při současném stavu identifikována oblast zpracování rozpisů hromadných plateb a zejména jejich úprava do podoby přijímané ISPF. Jedná se tedy o proces *P02\_Zpracování Rozpisů* z procesní mapy na obr. 23. Tento proces představuje pro pracovníky ÚSP vysokou administrativní zátěž. V dalších kapitolách je provedena bližší analýza uvedeného procesu jako výchozí bod k návrhu informačního systému pro zefektivnění zpracování rozpisů.

### **5.3 – Proces zpracování rozpisů**

Tato kapitola obsahuje podrobnou charakteristiku výše uváděného procesu a přesné vymezení jeho hranic. Východiskem je slovní formulace procesu, která je následně převedena do formálního zápisu v notaci BPMN.

#### **5.3.1 – Slovní charakteristika**

Vzhledem k tomu, že proces *P02\_Zpracování Rozpisů* uvedený na procesní mapě (viz obr. 23) lze rozdělit na 2 samostatné podprocesy s výrazně odlišným průběhem, je i následující text této skutečnosti přizpůsoben. Nejprve je charakterizován postup při zpracování výhradně elektronicky doručených rozpisů a poté je v závěru kapitoly vymezen prostor pro představení práce s rozpisy v listinné formě.

#### **Zpracování rozpisů doručených e-mailem**

Proces začíná volbou libovolného e-mailu s příloženým rozpisem, který má být zpracován. Z celkového počtu rozpisů doručených do ÚSP elektronickou cestou je cca 5% šifrováno veřejným PGP klíčem penzijního fondu. Obecně veškerá elektronická pošta šifrovaná prostřednictvím PGP je v penzijním fondu předávána e-mailem určenému zaměstnanci IT oddělení, který provede dešifrování privátním PGP klíčem a takto zpracovanou zprávu vrátí zpět pracovníkovi, který o dešifrování požádal. Pokud původní odesílatel použil pro šifrování zprávy nesprávný PGP klíč, není dešifrování možné a zaměstnanec IT oddělení

tuto skutečnost oznámí žádajícímu pracovníkovi, který vzniklou situaci dále řeší s odesílatelem.

Pokud se rozpis nachází ve standardním čitelném formátu (není šifrován nebo již proběhlo dešifrování), je uložen pracovníkem ÚSP do určeného adresáře na souborovém serveru penzijního fondu, odkud probíhá třízení prostřednictvím SWPF. Tímto nástrojem lze rozpisy hromadnou akcí roztrždit dle jejich formátu na *APF* a *ostatní* do dvou různých podadresářů. U většiny rozpisů zařazených mezi *ostatní* je poté provedena jejich konverze na formát APF. Pro tuto činnost nabízí SWPF grafické uživatelské rozhraní zpřístupňující určité pomocné funkce (dále jen „GUI SWPF“), přesto je však nutné každý soubor otevřít v odpovídající SW aplikaci (xls v MS Excel, doc v MS Word, apod.) a potřebná data vytěžit jejich prostým přenesením (kopírováním a vložením) po sloupcích do příslušného formuláře v GUI SWPF. Vzhledem ke značnému množství takto zpracovávaných rozpisů jde o časově náročnou činnost, při níž navíc vzniká nezanedbatelné riziko vzniku chyb. Nejčastější takovou chybou je záměna příspěvku zaměstnavatele za příspěvek účastníka a naopak. Po konverzi je každý takto vytvořený rozpis automaticky přesunut do adresáře, odkud probíhá import do ISPF. Zbývající rozpisy, které nebylo vzhledem k jejich formátu možné vytěžit jsou vytištěny a zpracovány stejně jako rozpisy doručené Českou poštou (viz dále).

Následující činností v procesu je kontrola struktury rozpisů nacházejících se v podadresáři pro formát APF, kam byly umístěny při třízení. Součástí zmíněné kontroly je otestování správného počtu oddělovačů jednotlivých údajů na každém řádku rozpisu a také ověření, že údaje v hlavičce rozpisu skutečně odpovídají jeho obsahu. Tuto automatizovanou kontrolu spouští pracovník ÚSP prostřednictvím SWPF jedním pokynem nad všemi soubory, které se v danou chvíli nacházejí v příslušném podadresáři. Výstupem celé operace je seznam rozpisů s nesprávnou strukturou včetně určení chybných řádků a seznam rozpisů s neodpovídajícími sumárními údaji. Pracovníkovi ÚSP je dále prostřednictvím GUI SWPF umožněno tyto nedostatky opravit. Pokud oprava není možná (především v případě, kdy nesouhlasí sumární údaje se skutečností), je o odstranění nesrovnalostí požádán zaměstnavatel, který rozpis zaslal. Soubory, jež byly při kontrole vyhodnoceny jako bezchybné jsou stejně jako konvertované rozpisy automaticky přesunuty do adresáře pro import do ISPF. Chybné rozpisy své úložiště nemění dokud nejsou opraveny a automaticky přesunuty, nebo odstraněny obsluhou bez uskutečnění opravy. Proces je

ukončen spuštěním importu rozpisů do ISPF, který může být proveden třemi různými způsoby:

- import všech rozpisů spuštěný automaticky každou hodinu
- import všech rozpisů spuštěný na pokyn obsluhy
- import vybraných rozpisů spuštěný na pokyn obsluhy

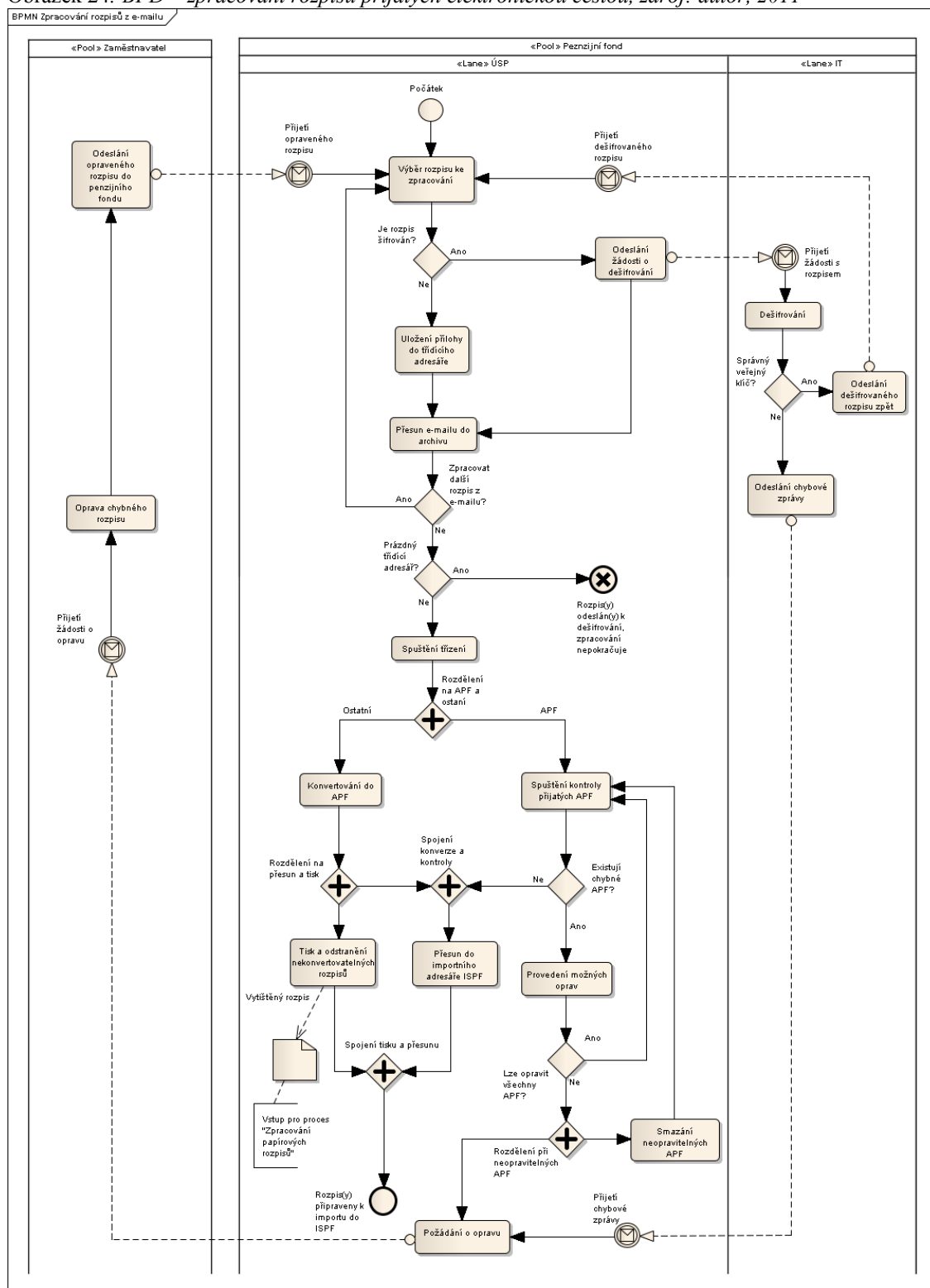
### **Zpracování papírových rozpisů**

Zpracování rozpisů v listinné podobě má od výše uvedeného postupu zcela odlišný charakter a probíhá přímo v ISPF. První ze dvou možností je prostřednictvím GUI ISPF veškeré nezbytné údaje z papírového rozpisu do systému přepsat, což může být zejména v případě velkého množství údajů značně pracné a náchylné ke vzniku chyb. Druhou variantou je využití funkce ISPF umožňující provedení kopie rozpisu, který v ISPF již existuje. Do této kopie jsou dále provedeny pouze potřebné změny bez nutnosti vypisování všech údajů. Postup při zpracování rozpisů v listinné podobě záleží pouze na volbě konkrétního pracovníka ÚSP.

### **5.3.2 – Model procesu v notaci BPMN**

Model procesu *P02\_Zpracování Rozpisů* obsahuje dva samostatné diagramy v notaci BPMN, které zachycují podprocesy představené v předchozí kapitole. Tyto diagramy (viz obr. 24 a 25) nabízí komplexní pohled na posloupnost činností vykonávaných při zpracování rozpisů přijatých v ÚSP v elektronické resp. listinné podobě. Míra podrobnosti diagramů odpovídá potřebě identifikovat problematická místa procesu, aby bylo možné snáze předejít případným komplikacím při následném návrhu informačního systému. Rozdělení procesu *P02\_Zpracování Rozpisů* do dvou samostatných částí autor zvolil zejména pro lepší čitelnost diagramů a pro snazší pochopení dané problematiky.

Obrázek 24: BPD – zpracování rozpisů přijatých elektronickou cestou, zdroj: autor, 2011



Průběh procesu zachycený na výše uvedeném diagramu odpovídá standardnímu postupu při zpracování rozpisů zaslaných elektronickou poštou. Jako problémový lze v procesu označit způsob přesouvání rozpisů do třídicího adresáře, které probíhá pro každou přílohu

zvlášť a je časově poměrně náročné. Tento postup vyplývá především z nutnosti přečtení každého mailu s rozpisem kvůli možnému připojení žádosti o potvrzení přijetí zprávy nebo jiného sdělení. Rovněž je třeba vytřídit šifrované zprávy a odeslat je určenému pracovníkovi IT. Dále má poměrně komplikovaný průběh konverze ostatních formátů rozpisů na APF, což je zapříčiněno tím, že ÚSP nemá definovaný žádný standard pro podobu rozpisů v jiném formátu než APF. Posledním identifikovaným problémem je relativně obtížná oprava přijatých APF v SWPF. Zde je důvodem nejednotné nastavení výstupů mzdových programů zaměstnavatelů.

Při bližším pohledu na diagram je dále možné si všimnout, že proces může skončit dvěma různými způsoby. Při prvním z nich dojde k umístění zpracovaných rozpisů do importního adresáře ISPF, což v reálném provozu nastává ve většině případů. Druhý cílový bod označený „Rozpis(y) odeslán(y) k dešifrování“ slouží pouze pro situaci, kdy dojde k odeslání jednoho nebo více rozpisů k dešifrování a zároveň nejsou ve stejném průběhu procesem zpracovány jiné rozpisy. Znamená to, že třídící adresář je prázdný a zpracování rozpisů nepokračuje.

Obrázek 25: BPD – zpracování papírových rozpisů, zdroj autor, 2011

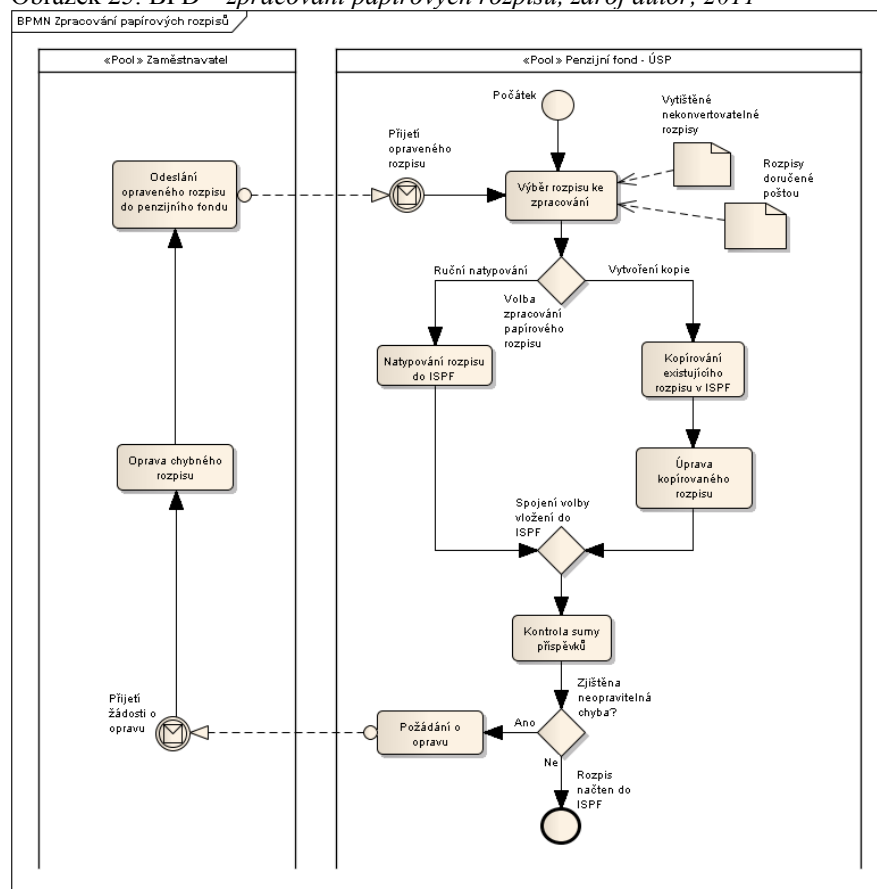


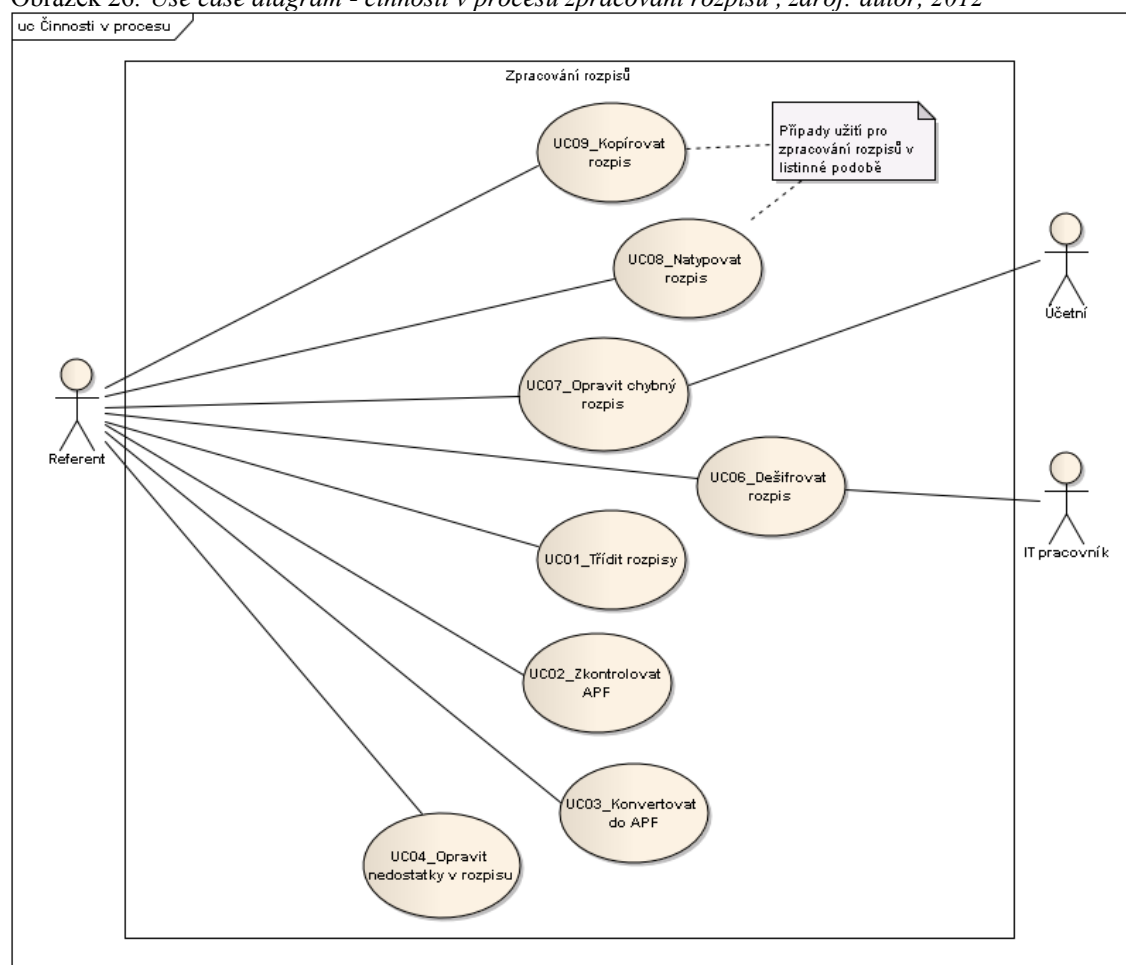


Diagram na obrázku výše představuje proces zpracování papírových rozpisů. Jeho vstupy tvoří rozpisy vytištěné v průběhu procesu zpracování souborů z e-mailových příloh nebo rozpisy doručené do ÚSP v listinné podobě Českou poštou. Na rozdíl od předchozího procesu zde neprobíhá žádný import do ISPF, ale údaje jsou do tohoto systému zadávány přímo jednou z dostupných možností. Jak již bylo uvedeno, listinné rozpisy tvoří pouze malou část celkového počtu rozpisů a jejich zastoupení má klesající tendenci.

### 5.3.3 – Use-Case model procesu

Proces zpracování rozpisů, jehož dynamické chování je zachyceno na výše uvedených diagramech lze zobrazit také statickým způsobem prostřednictvím UML diagramu případů užití. Tento diagram (viz obr. 26) zobrazuje jednotlivé činnosti (resp. skupiny činností) v procesu jako případy užití spolu s aktéry, kteří se daných činností určitým způsobem účastní.

Obrázek 26: Use case diagram - činnosti v procesu zpracování rozpisů , zdroj: autor, 2012



## **Aktéři**

*Referent* – představuje roli hlavního účastníka procesu, který zabezpečuje správné zpracování rozpisů a v případě potřeby komunikuje s ostatními dvěma aktéry.

*Účetní* – aktér v této roli zasílá za konkrétního zaměstnavatele rozpisy do penzijního fondu a řeší s referenty případné chyby v rozpisech

*IT pracovník* – aktér, který od referenta přijímá šifrované rozpisy a dešifrované je vrací zpět

## **Případy užití**

*UC01\_Třídít rozpisy* – roztřídění přijatých rozpisů na APF a ostatní formáty

*UC02\_Zkontrolovat APF* – kontrola struktury rozpisů přijatých přímo ve formátu APF

*UC03\_Konvertovat do APF* – převedení rozpisů v ostatních formátech na formát APF včetně vytištění rozpisů, které nelze konvertovat

*UC04\_Opravit nedostatky v rozpisu* – provedení oprav nedostatků zjištěných při kontrole APF bez součinnosti odesílatele rozpisu

*UC05\_Dešifrovat rozpis* – dešifrování rozpisu zašifrovaného prostřednictvím PGP

*UC06\_Opravit chybný rozpis* – zajištění opravy chybných rozpisů v součinnosti s jejich odesílatelem

*UC07\_Natypovat rozpis* – přepsání údajů z papírového rozpisu přímo do formuláře přístupného z GUI ISPF

*UC08\_Kopírovat rozpis* – provedení kopie již existujícího rozpisu prostřednictvím GUI ISPF

## **5.4 – Návrh nového procesu**

Pro návrh nové podoby procesu zpracování rozpisů vycházel autor především z identifikovaných problémových vlastností stávajícího procesu, kterými jsou tyto:

- časově a administrativně náročná úvodní fáze zpracování, která probíhá pro každý rozpis zvlášť
- nutnost relativně komplikované konverze rozpisů v různých formátech do APF
- nezbytná oprava některých rozpisů přijatých ve formátu APF
- používaný způsob dešifrování přijatých zpráv šifrovaných PGP

Základním předpokladem pro zefektivnění zpracování rozpisů je tyto problémové vlastnosti eliminovat a zároveň zajistit požadovanou kvalitu výstupů procesu.

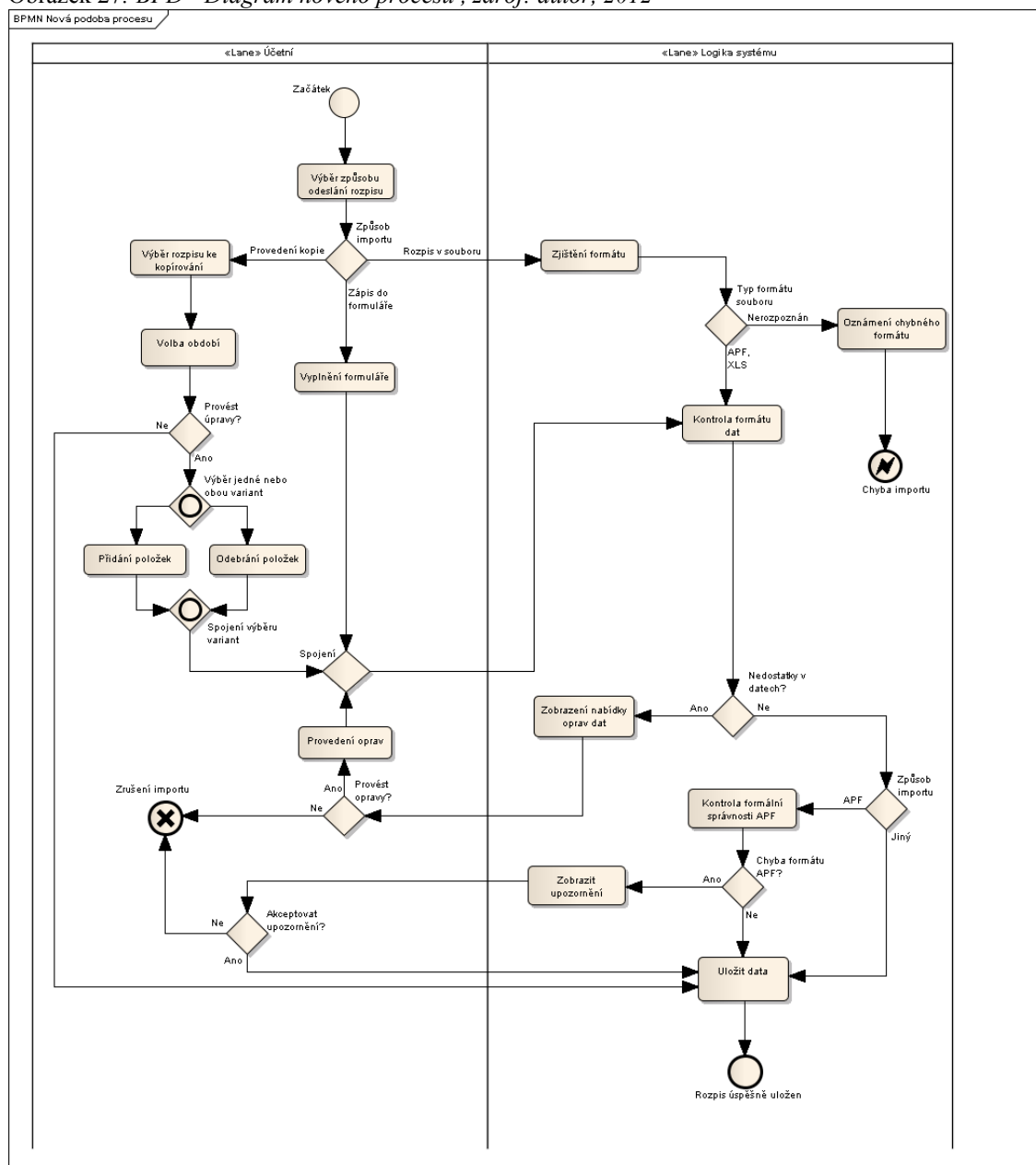
Dle výše uvedeného autor jako řešení navrhuje zavedení webového informačního systému, který umožní zástupcům zaměstnavatelů zasílat rozpisy v předem definovaných formátech a na straně penzijního fondu bude tyto rozpisy generovat přímo ve formátu APF akceptovaném ISPF. Veškerá komunikace mezi uživatelem a systémem by měla být odpovídajícím způsobem zabezpečena dostupnými technologiemi, což vyřeší potřebu některých zaměstnavatelů své rozpisy šifrovat.

Při odesílání rozpisů je rovněž třeba zajistit, aby systém kladl uživatelům minimální překážky a současně dokázal rozpoznat nedostatky v rozpisech. Z tohoto důvodu je v návrhu nového procesu uvažována možnost výběru ze čtyř různých variant odeslání rozpisu:

- odeslání rozpisu v souboru ve formátu APF
- odeslání rozpisu v souboru ve formátu xls
- provedení kopie dříve vloženého rozpisu s provedením případných úprav
- zadání rozpisu prostřednictvím formuláře

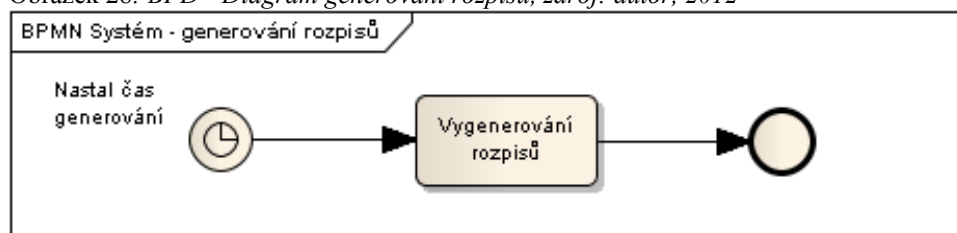
Případné nedostatky v rozpisech, resp. odesílaných datech, bude systém kontrolovat dle způsobu odeslání každého rozpisu. Blíže jsou jednotlivé varianty charakterizovány v kapitole 5.6 ve specifikaci příslušných případů užití. Kompletní průběh nového procesu je uveden na obrázku 27.

Obrázek 27: BPD - Diagram nového procesu , zdroj: autor, 2012



Bezproblémový průběh nového procesu je ukončen uložením dat z rozpisu systémem do databáze. Vygenerování rozpisů ve formátu APF do importního adresáře ISPF zabezpečuje zcela nezávislý proces spouštěný časovou událostí (viz obr. 28).

Obrázek 28: BPD - Diagram generování rozpisů, zdroj: autor, 2012



## 5.5 – Datový slovník

V datovém slovníku jsou zachyceny klíčové pojmy z problémové oblasti včetně jejich definic a případných synonym. Přesné vymezení těchto termínů slouží pro eliminaci možných nejasností, které vnáší do procesu analýzy a návrhu informačního systému riziko chybného postupu.

**hromadná platba** – platba zaslaná zaměstnavatelem do penzijního fondu v jedné částce (výjimečně i více) za účelem poskytnutí příspěvku svým zaměstnancům

*Synonyma:* žádná

**rozpis** – datový soubor nebo listina, podle které je možné hromadnou platbu rozdělit správným způsobem

*Synonyma:* soubor, soubor s rozpisem

**zpracování rozpisů** – převedení rozpisů do formátu APF akceptovaného ISPF a jejich umístění do importního adresáře.

*Synonyma:* žádná

**příspěvek** – část hromadné platby, která náleží právě jednomu zaměstnanci. V souvislosti s rozpisem označuje řádek rozpisu.

*Synonyma:* řádek rozpisu

**zaměstnavatel** – společnost, která některému z klientů penzijního fondu zasílá příspěvky hromadnou platbou

*Synonyma:* žádná

**zaměstnanec** – klient penzijního fondu, kterému jeho zaměstnavatel zasílá příspěvek

*Synonyma:* klient (ve vztahu s penzijním fondem)

**SWPF** – softwarový nástroj pro podporu zpracování rozpisů hromadných plateb

*Synonyma:* žádná

**ISPF** – podnikový informační systém penzijního fondu

*Synonyma:* žádná

**účetní** – osoba, která má na starosti zasílání rozpisů hromadných plateb do penzijního fondu za určeného zaměstnavatele

*Synonyma:* zástupce zaměstnavatele

**hlavička rozpisu** – údaje na prvním řádku rozpisu APF jednoznačně identifikující hromadnou platbu a příslušného zaměstnavatele.

*Synonyma:* žádná

## **5.6 – Požadavky na systém**

Obecným požadavkem na navrhovaný informační systém je zefektivnění činnosti ÚSP v oblasti zpracování rozpisů hromadných plateb. Funkce systému vyplývající z níže uvedených konkrétních požadavků lze charakterizovat následovně: Systém zástupcům zaměstnavatelům umožní zasílat rozpisy hromadných plateb, tak aby byly v penzijním fondu přijímány přímo ve formátu APF odpovídajícímu nastavení ISPF.

### **Funkční požadavky**

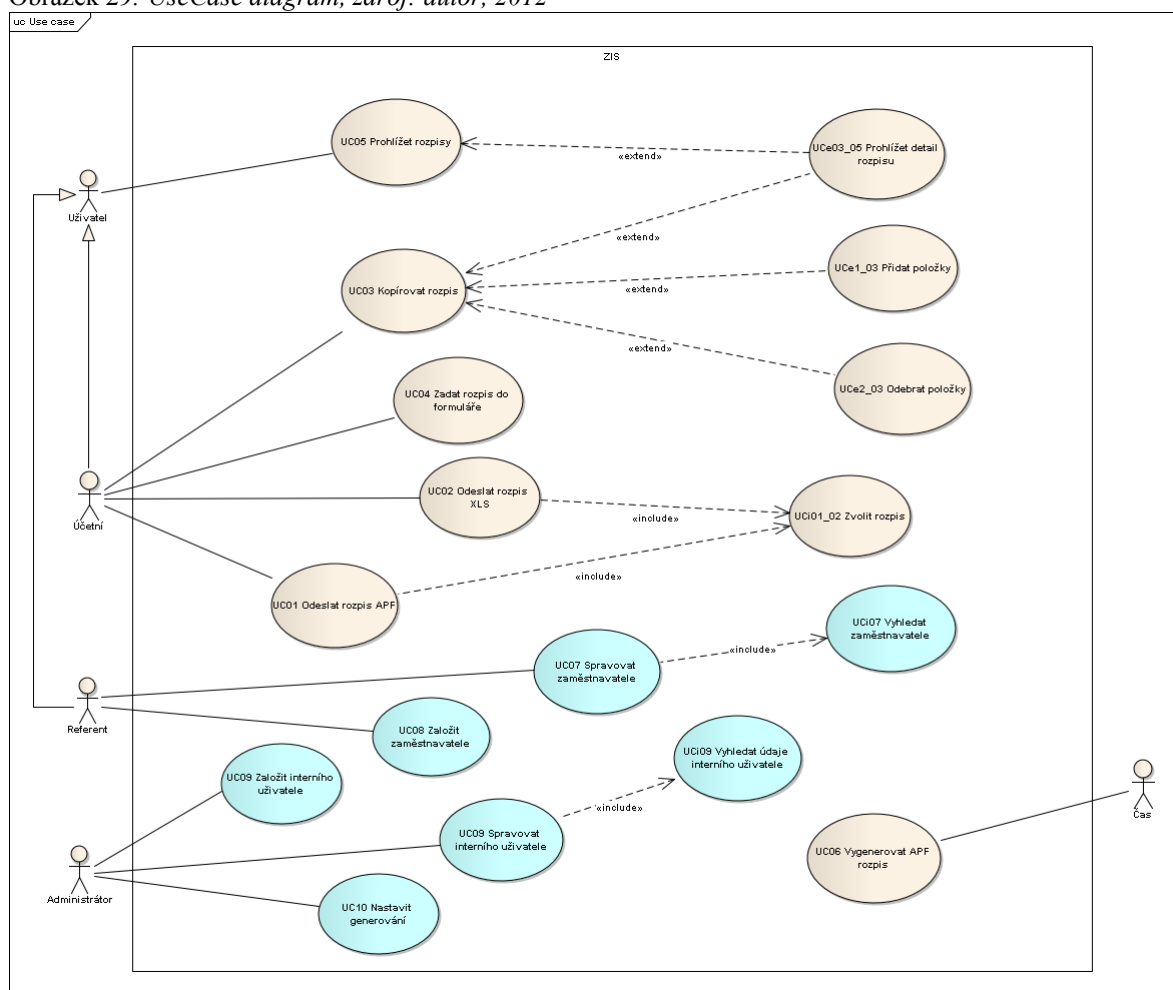
- systém zaměstnavatelům umožní zasílat rozpisy do ÚSP
- po každého zaměstnavatele systém povede evidenci odeslaných rozpisů
- systém bude při zasílání rozpisů provádět kontrolu správnosti formátu rozpisů a správnosti formátu odesílaných dat
- systém bude na straně ÚSP generovat rozpisy ve formátu APF a předávat je přímo ISPF

### **Nefunkční požadavky**

- systém bude rozlišovat tři typy uživatelů: zaměstnavatel, referent, administrátor
- uživatelé budou k systému přistupovat prostřednictvím internetu
- systém bude využívat zabezpečený způsob přenosu dat

Pro upřesnění funkčních požadavků je na obrázku 29 uveden diagram případů užití vytvořený v souladu s metodikou UP, který zobrazuje hranice systému, role uživatelů systému (aktéry) a jejich vztahy s jednotlivými případy užití. Modrou barvou jsou odlišeny případy užití, které souvisí se správou uživatelů systému.

Obrázek 29: UseCase diagram, zdroj: autor, 2012



## 5.7 – Specifikace aktérů a případů užití

V této kapitole je nejprve uvedena stručná charakteristika jednotlivých aktérů případů užití z obrázku 29 a následně blíže představeny samotné případy užití, které přímo souvisí s procesem zpracování rozpisů hromadných plateb (UC01 – UC06). Formální zápis těchto případů užití v podobě scénářů a příslušných UML diagramů je (spolu s charakteristikou případů užití UC07 – UC10 týkajících se administrace uživatelů systému) součástí přílohy.

### 5.7.1 – Aktéři

#### Účetní

Tento aktér představuje roli uživatele, který za určitého zaměstnavatele vkládá do systému rozpis hromadných plateb jedním ze čtyř dostupných způsobů. Ve skutečnosti nemusí být tímto uživatelem přímo účetní, ale jakýkoliv zástupce daného zaměstnavatele.

### **Referent**

Referent je role pro zaměstnance ÚSP, kteří budou mít na starosti správu uživatelů v roli účetních.

### **Uživatel**

Uživatel je obecnou rolí vyčleňující možnost prohlížet odeslané rozpisy jak pro účetní, tak pro referenty, aby bylo možné snáze řešit případné reklamace týkající se odeslaných rozpisů.

### **Administrátor**

Aktér administrátor představuje roli pro uživatele zabezpečující správu uživatelů v roli referenta a administrátora. Současně může uživatel v této roli nastavovat parametry generování rozpisů na výstupu systému.

### **Čas**

Aktér spouštějící generování rozpisů v definovaných časových intervalech

## **5.7.2 – Případy užití**

### **UC01 Odeslat rozpis APF**

Případ užití *Odeslat rozpis APF* reprezentuje využití systému pro vložení zaměstnavatelského rozpisu přímo ve formátu APF do systému. Počáteční průběh tohoto UC zahrnuje volbu rozpisu a je shodný s počátečním průběhem *UC02 Odeslat xls rozpis* (viz níže). Z tohoto důvodu je pro volbu rozpisu vazbou <<include>> vyčleněn dodavatelský *UCi01\_02 Zvolit rozpis*, který je na začátek klientských *UC01* a *UC02* zahrnut. Hlavním aktérem je v tomto případě zaměstnavatel, který musí být před odesláním rozpisu do systému přihlášen. Na pokyn uživatele je systémem umožněn výběr souboru s rozpisem ze zvoleného úložiště. Po potvrzení výběru rozpisu systém provede rozpoznání jeho typu (zjištění, že se jedná o rozpis APF). Následuje kontrola formátu obsažených dat, která zahrnuje ověření, zda jsou údaje na řádcích s příspěvkem zadány ve správném formátu. V případě zjištěných nedostatků je uživateli zobrazena nabídka s možností opravy nesprávně zadaných údajů. Dále je u APF rozpisu provedena kontrola, zda údaje na řádcích s příspěvkem odpovídají údajům na prvním (sumárním) řádku rozpisu. Systém dle údajů o jednotlivých příspěvcích a o příslušném zaměstnavateli může jednoznačně určit, jak má sumární řádek vypadat. Z tohoto důvodu, v případě nalezení nesrovnalostí, systém pouze zobrazí možnost přijmout nebo odmítnout automatickou opravu nesprávně zadaných položek. Po uložení dat z rozpisu je uživatel informován o úspěšně provedené akci.



V alternativních scénářích uvedeného případu užití jsou dále řešeny možné chybové události při odesílání APF rozpisu.

### UC02 Odeslat rozpis xls

Vzhledem k zahrnutí *UCi01\_02*, má tento případ užití shodný počáteční průběh s *UC01* a odlišuje se až od bodu, kdy systém rozpoznáním typu rozpisu zjistí, že se jedná o rozpis ve formátu xls. Dále následuje kontrola formátu zadaných dat, jejíž průběh je stejný jako v případě rozpisu ve formátu APF včetně řešení chybových situací. V případě, že nejsou zjištěny žádné nedostatky následuje uložení dat a potvrzení zaměstnavateli, že akce byla úspěšně dokončena. Také u tohoto případu užití jsou v rámci alternativních scénářů řešeny případné chybové události, které mohou při odesílání xls rozpisu nastat.

Aby mohlo být zasílání rozpisů ve formátu xls realizováno, je třeba stanovit jejich (nejlépe jednotnou) strukturu, kterou bude systém akceptovat a z níž bude možné na straně penzijního fondu vytvořit rozpis ve formátu APF. Z poznatků získaných z průběhu analýzy procesu zpracování rozpisů vyplývá, že systém dokáže vygenerovat rozpis ve formátu APF, pokud bude rozpis xls obsahovat minimálně tyto údaje: období, rodná čísla klientů, čísla smluv klientů, výše a typ jednotlivých příspěvků. Na obrázku 30 je uveden návrh rozpisu ve formátu xls se všemi nezbytnými údaji.

Obrázek 30: *Návrh rozpisu ve formátu xls, zdroj: autor, 2012*

	A	B	C	D	E	F	G
1	Jméno	Příjmení	Rodné číslo	Číslo smlouvy	Příspěvek účastníka	Příspěvek zaměstnavatele	Příspěvek 3. osoby
2	Jan	Novák	7512121111	1122334455	1500	500	0
3	Petra	Dvořáková	7211122222	9988775566	1000	1000	0
4	Období	VI.12					

### UC03 Kopírovat rozpis

Kopírování rozpisů systém umožní pro vytvoření a odeslání rozpisu, jehož obsah má být stejný nebo podobný jako obsah rozpisu již zasláného v minulosti. Tvorba kopie se uskuteční na úrovni systému a uživatel tak nemusí mít k dispozici rozpis v souboru. Hlavním aktérem je opět zaměstnavatel, kterému systém po příslušné volbě nabídne ke kopírování dostupné rozpisy z předcházejících období včetně možnosti zobrazení jejich detailu, reprezentované rozšiřujícím případem užití *UCe03\_05 Prohlížet detail rozpisu*. Uživatel může zvolený rozpis odeslat nebo ještě před odesláním provést potřebné úpravy (odebrání nebo přidání příspěvku), což je také realizováno rozšiřujícími případy užití *UCe1\_03 Přidat položky* a *UCe2\_03 Odebrat položky*. Potvrzením těchto úprav zaměstnavatelem, systém provede kontrolu správnosti formátu dat, jež se uplatňuje rovněž

při odesílání APF a xls rozpisů. Pokud k žádným změnám v rozpisu nedošlo, kontrola spuštěna není. Po zvolení období uživatelem systém uloží odeslaná data, aby mohla být při následující časové události vygenerována do formátu APF.

#### **UC04 Zadat rozpis do formuláře**

Podobně jako kopírování rozpisů umožní tento případ užití zadávání rozpisů přímo v systému bez nutnosti odesílání údajů v souboru. Zvolením příslušné volby nabídne systém uživateli formulář pro zadání všech potřebných údajů. Po jeho potvrzení uživatelem systém provede kontrolu formátu zadaných dat, která poté uloží do databáze. Pokud je zjištěn nějaký nedostatek, umožní systém provedení opravy stejně jako v předchozích případech. Je zřejmé, že tuto možnost odesílání rozpisů využijí zejména zaměstnavatelé s malým počtem zaměstnanců, kterým poskytují příspěvek na penzijní připojištění.

#### **UC08 Prohlížet rozpisy**

Tento případ užití představuje možnost zobrazení jednotlivých rozpisů, které zaměstnavatel do systému v minulosti zadal. Rozšiřujícím případem užití je *UCe03\_05 Prohlížet detail rozpisu* reprezentující možnost zobrazení informací o příspěvcích zahrnutých v daném rozpisu.

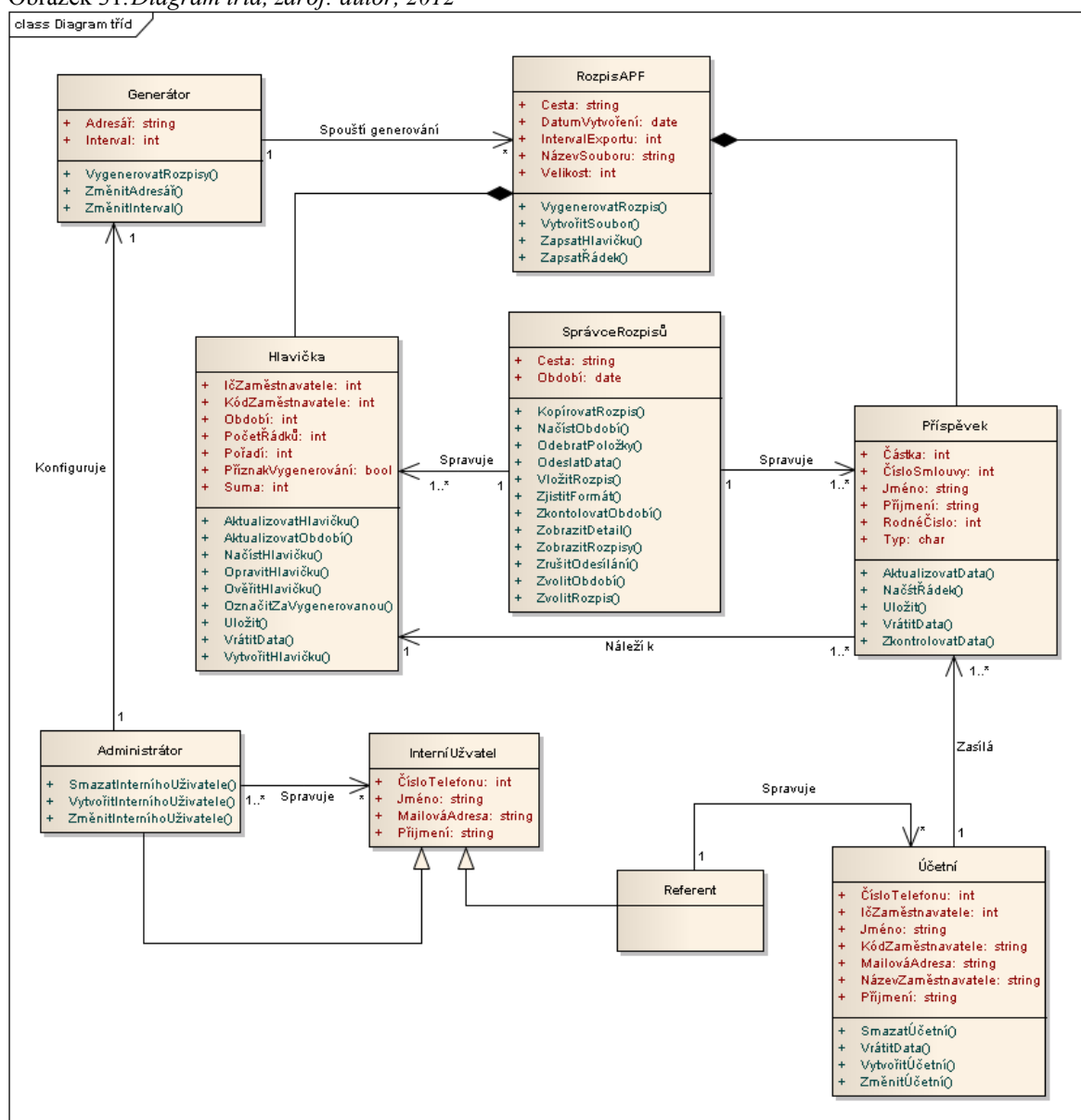
#### **UC10 Vygenerovat APF rozpis**

Časem spuštěný případ užití *Vygenerovat APF rozpis* představuje akci, při níž jsou uložené a dosud nevygenerované rozpisy systémem vygenerovány do importního adresáře ISPF. V návrhu je uvažován nastavitelný interval generování rozpisů

### **5.8 – Statický pohled na model systému**

V této části je představena statická struktura navrhovaného systému prostřednictvím UML diagramu tříd. Tento diagram znázorňuje vztahy mezi objekty při realizaci chování systému vyjádřeného v případech užití. Pro nalezení jednotlivých tříd a jejich vlastností autor vycházel z analýzy podstatných jmen a sloves obsažených v požadavcích na systém, specifikaci případů užití a dalších dostupných zdrojích. Získané výstupy byly dále upřesňovány během specifikace případů užití a především při tvorbě jednotlivých sekvenčních diagramů. Výsledkem je diagram tříd systému uvedený na obrázku 31.

Obrázek 31: Diagram tříd, zdroj: autor, 2012



### 5.8.1 – Charakteristika tříd

#### SprávceRozpisů

Instance této třídy v systému zabezpečují zpracování vstupních dat pro všechny čtyři možnosti odeslání rozpisů. Práce se vstupními daty objektů třídy SprávceRozpisů zahrnuje získání uvedených dat, zabezpečení jejich kontroly s případnou opravou a následné uložení do databáze.

### **Hlavička**

Třída Hlavička obsahuje objekty s vlastnostmi odpovídajícími definici hlavičky dle formátu APF. Metody těchto objektů slouží pro vytvoření, resp. ověření hlavičky na základě vlastností řádků příslušných dané hlavičce, údajů o období a hodnotě atributů IčZaměstnavatele, KódZaměstnavatele a NázevZaměstnavatele objektu z třídy Účetní. Tento objekt reprezentuje uživatele, který hlavičku vytvořil vložením rozpisu do systému.

### **Příspěvek**

Objekty třídy Příspěvek reprezentují řádky rozpisů s údaji o příspěvcích dle formátu APF. Podobně jako v předchozím případě slouží metody těchto objektů k načtení údajů z rozpisu, jejich kontrole a následnému uložení do databáze.

### **Účetní**

Tato třída obsahuje instance, které reprezentují uživatele zasílající rozpisy za určitého zaměstnavatele. Kromě atributů s osobními a kontaktními údaji mají objekty této třídy také atributy IčZaměstnavatele, KódZaměstnavatele a NázevZaměstnavatele, které jsou nezbytné pro vytvoření resp. ověření hlavičky dle formátu APF.

### **RozpisAPF**

Třída RozpisAPF svými metodami zabezpečuje výstupní operaci v podobě vygenerování hlaviček a příslušných řádků do textových souborů ve formátu APF, čímž vznikne rozpis akceptovatelný ISPF. S touto třídou jsou ve vztahu agregace třídy Hlavička a Příspěvek, protože uložené hlavičky a příspěvky (řádky) jsou součástí vygenerovaných rozpisů APF.

### **Generátor**

Instance třídy Generátor dává v určených časových intervalech pokyn k vygenerování rozpisů objektům třídy RozpisAPF. Uvedený časový interval je určen hodnotou atributu Interval. Druhý atribut této třídy s názvem Adresář definuje cestu k importnímu adresáři ISPF. Oba uvedené atributy je umožněno modifikovat administrátorovi systému.

### **Interní uživatel**

Interní uživatel je třídou objektů reprezentujících uživatele systému zodpovědné za nastavení systému a administraci všech jeho uživatelů. Interní uživatel je nadtřídou pro třídy Referent a Administrátor.

### **Referent**

Tato třída obsahuje objekty představující interní uživatele, kteří zabezpečují správu uživatelů v roli účetní. Správa Účetních zahrnuje jejich vytváření, modifikaci a rušení.

### **Administrátor**

Instance této třídy, stejně jako objekty třídy Referent dědí veškeré atributy od třídy Interní uživatel. Jejich chování je navíc rozšířeno o metody pro správu Interních uživatelů (vytváření, modifikaci a rušení). Ze vztahu s třídou Generátor dále pro instance této třídy vyplývá možnost měnit interval generování rozpisů a také cestu k cílovému adresáři.

Navržené třídy, jejich vztahy a vlastností tvoří spolu s diagramy uvedenými v příloze konceptuální model systému nezohledňující žádné konkrétní implementační prostředí. V další fázi návrhu, která již není předmětem této práce, by bylo možné model dále přizpůsobit zvolenému programovacímu jazyku a databázi.

## 6 – Výsledky a diskuze

Možností jak přistupovat k analýze podnikových procesů nabízí odborná literatura hned několik. V zásadě se jednotlivé publikované metody liší podle plánovaného využití jejich výstupů. Všem uznávaným přístupům k analýze procesů představeným v kapitole 3.6 je však společné zaměření na identifikování procesů včetně jejich vzájemných souvislostí v podniku jako celku. Hlavním cílem práce bylo provedení analýzy a následné zefektivnění konkrétního zvoleného procesu, což zcela neodpovídá principu žádné z uvedených metod. Pro splnění cíle práce se však ukázalo velice důležité zvolený podnikový proces zasadit do kontextu všech elementárních procesů ÚSP (Úsek správy příspěvků penzijního fondu), aby mohla být znalost jejich návaznosti využita při dalších krocích analýzy. Za tímto účelem zvolil autor úvodní fáze metodiky MMABP, které byly aplikovány do prostředí ÚSP. Výsledkem je ucelený pohled na systém elementárních procesů ÚSP vyjádřený procesní mapou úseku v notaci Ericsson-Penker. Vzhledem k tomu, že tato notace umožňuje vytvořit diagram, který reprezentuje statickou strukturu procesů danou jejich existencí, vzájemnými vztahy a atributy, je dle názoru autora takový pohled na procesy velmi vhodným východiskem k dalšímu zkoumání kteréhokoliv z identifikovaných procesů a následnému návrhu informačního systému.

Rovněž variant pro formální zachycení dynamického chování modelovaných procesů existuje v podobě různých grafických notací více. Zcela přirozeně se od sebe jednotlivé notace liší jak používanými grafickými symboly, tak i důrazem kladeným na určité vlastnosti procesů. Odborná literatura nejčastěji srovnává notace BPMN, EPC a diagramy aktivit UML. Dle [5] je slabou stránkou notace BPMN skutečnost, že klade značný důraz na technologicky významné aspekty procesů, jako jsou jejich detaily, proveditelnost, jednoznačná specifikovatelnost a determinovatelnost, čímž je potlačována pozornost netechnickým a organizačním a „lidským“ aspektům procesu. Uvedená slabina je v tomto případě vztahována k možnosti následného procesního reengineeringu, nebo k ambici notace BPMN stát se obecně uznávaným standardem pro modelování procesů. Dle poznatků autora získaných v průběhu analýzy zvoleného procesu a následného návrhu informačního systému, je však diskutovaná vlastnost notace BPMN pro typ problému řešeného v této práci naopak výhodou. Důvod je ten, že jednoznačně specifikovaný proces lze snáze automatizovat informačním systémem. Na základě uvedených skutečností byla v této práci pro vyjádření daného procesu použita právě notace BPMN. Alternativně by

bylo možné vytvořit jednoznačně specifikovaný model procesu i prostřednictvím diagramu aktivit UML, ale tato notace obsahuje pouze omezené množství elementů, což oslabuje jeho vyjadřovací schopnost. UML diagram aktivit je použit v pozdější fázi analýzy informačního systému pro specifikaci vybraných případů užití.

V oblasti objektově orientované analýzy a návrhu informačních systémů je situace ve srovnání s analýzou podnikových procesů podstatně jednodušší. Díky existenci standardu pro vizuální modelování systémů v podobě jazyka UML odborné zdroje příliš jiných alternativ nenabízí. Jazyk UML je také mimo jiné charakterizován jako komunikační prostředek mezi vývojáři a zadavateli systému ve fázi analýzy. Toto využití UML vychází z tvrzení, že jeho vybrané modely jsou pochopitelné i pro zadavatele aplikace a umožní kvalitní vyjasnění požadavků uživatelů na vytvářený systém.[25] Ze zkušeností autora získaných při tvorbě praktické části této práce lze konstatovat, že UML při vyjasnění požadavků na systém pomoci může, ale v řadě případů pouze částečně. Míra použitelnosti UML pro tento účel závisí na typu a srozumitelnosti vytvořeného diagramu a také na technickém cítění uživatele, s nímž jsou požadavky diskutovány.

Odborná literatura uvádí také slabé stránky jazyka UML. Dle [9] UML velmi špatně podporuje první fáze analýzy informačních systémů, kdy je třeba modelovat business kontext budované aplikace. Tento poznatek do značné míry odpovídá zjištěním, ke kterým autor dospěl při analyzování vybraného podnikového procesu a návrhu jeho změny. Od fáze specifikace požadavků převedených do jednotlivých případů užití se však UML uplatňuje velice dobře. Jako alternativa se zde nabízí využití metody analýzy a návrhu systémů BORM, která úvodní fáze analýzy při tvorbě business modelu organizace podporuje lépe. Dle názoru autora je při použití jazyka UML pro analýzu a návrh informačního systému určeného do podnikového prostředí nezbytné nejprve podrobně analyzovat stávající procesy, které bude informační systém ovlivňovat. Pro analýzu těchto procesů je možné využít např. obecnou metodiku MMABP nebo jiný existující přístup v závislosti na požadovaných výstupech.

## 7 – Závěr

Cílem této práce bylo analyzovat zavedený proces v podniku a navrhnout informační systém, který by tento proces zefektivnil. Autor pro tento účel zvolil proces zpracování rozpisů hromadných plateb v penzijním fondu. Jako východisko k podrobné analýze uvedené činnosti byla v notaci Erikson – Penker vytvořena procesní mapa pracoviště, kde se daný proces uplatňuje, aby bylo možné identifikovat jeho vztahy s ostatními souvisejícími procesy. Pro formální zápis průběhu zpracování rozpisů hromadných plateb v podobě procesního diagramu autor zvolil notaci BPMN, protože svými vlastnostmi nejlépe vyhovovala požadavkům na jednoznačnou formulaci zvoleného procesu s ohledem na následný návrh jeho nové podoby. Požadavky na systém vycházejí ze zjištěných problematických vlastností původní podoby procesu a byly rovněž přizpůsobeny požadované realizovatelnosti navržených změn. Pro statický pohled na strukturu systému byl vytvořen diagram tříd znázorňující vztahy mezi objekty realizující chování systému. Do své konečné podoby se tento diagram dostával postupně v celém průběhu specifikace požadavků. Autor dospěl k závěru, že při využití jazyka UML pro analýzu a návrh informačního systému určeného do podnikového prostředí je třeba analyzovat stávající procesy s využitím dostupných nástrojů procesní analýzy. Toto zjištění potvrzuje názor, podle nějž UML špatně podporuje úvodní fázi analýzy informačního systému při tvorbě jeho business modelu.



## 8 – Seznam použitých zdrojů

1. ARLOW, Jim; NEUSTADT, Ila. *UML2 a unifikovaný proces vývoje aplikací*. Brno : Computer Press, 2007. 567 s.
2. DUMAS, Marlon; VAN DER AALST, Wil; TER HOFSTEDÉ, Arthur. *Process-Aware Information Systems*. New Jersey : John Wiley & sons, Inc., 2005. 409 s.
3. KANISOVÁ, Hana; MÜLLER, Miroslav. *UML srozumitelně*. Brno : Computer Press, 2004. 147 s.
4. POLÁK, Jiří; MERUNKA, Vojtěch; CARDA, Antonín. *Umění systémového návrhu*. Praha : Grada Publishing a.s., 2003. 196 s.
5. ŘEPA, Václav. *Podnikové procesy : Procesní řízení a modelování*. Praha : Grada, 2006. 268 s.
6. ŠMÍDA, Filip. *Zavádění a rozvoj procesního řízení ve firmě*. Praha: Grada, 2007. 293 s.
7. VRANA, Ivan; RICHTA, Karel. *Zásady a postupy zavádění podnikových informačních systémů*. Praha : Grada Publishing a.s., 2005. 188 s.
8. MÁČEL, Lukáš. *MODELOVÁNÍ procesů v jazyce Python : Vybrané problémy informačních systémů*. Brno, 2010. 22 s. Seminární práce. VUT Brno.
9. MERUNKA, Vojtěch; PERGL, Robert; PÍČKA, Marek. *Objektově orientovaný přístup v projektování informačních systémů*. Praha: Česká zemědělská univerzita, 2005. 238 s.
10. PROCHÁZKA, Jaroslav. *Procesní řízení realizace projektů*. první. Ostrava : Ostravská univerzita v Ostravě, 2006. 68 s
11. VONDRÁK, Ivo. *METODY BYZNYS MODELOVÁNÍ*. Ostrava : Technická univerzita Ostrava, 2004. 92 s.
12. BASL, Josef, et al. *BPM prakticky* [online]. 2008 [cit. 2011-12-04]. Dostupné z WWW: <<http://bpm-sme.blogspot.com/>>.
13. BENEŠ, Michal, et al. *Přehled OO notací a metodik* [online]. 2005 [cit. 2011-12-04]. Dostupné z WWW: <<http://objekty.vse.cz/Objekty/MethodikyANotace>>.
14. DRBOHLAV, Štěpán, et al. *Nástroje modelování a řízení podnikových procesů*. Praha : Vysoká škola ekonomická, 2010. 62 s. Dostupné z WWW: <[http://kogninfo.vse.cz/doc/Nastroje\\_modelovani\\_a\\_rizeni\\_podnikovych\\_procesu.pdf](http://kogninfo.vse.cz/doc/Nastroje_modelovani_a_rizeni_podnikovych_procesu.pdf)>.
15. FAKHROUTDINOV , Kirill. *UML - Graphical Notation Overview and Reference* [online]. c2011 [cit. 2011-12-10]. Dostupné z WWW: <<http://www.uml-diagrams.org/>>.
16. FEGGINS, Reedy. *IBM* [online]. 2003 [cit. 2011-12-10]. Designing component-based architectures with Rational Rose RealTime. Dostupné z WWW: <<http://www.ibm.com/developerworks/rational/library/797.html>>.
17. OŠLEJŠEK, Radek. *Objektově orientované metody návrhu IS* [online]. 2009 [cit. 2011-12-05]. Dostupné z WWW: <<http://is.muni.cz/el/1433/jaro2009/PA103/>>.
18. RÁČEK, Jaroslav. *Procesní řízení* [online]. Brno : MU, 2011 [cit. 2011-09-11]. Dostupné z WWW: <<http://is.muni.cz/el/1433/jaro2011/PV165/>>.
19. *ARIS BPM Community* [online]. c2011 [cit. 2011-12-04]. ARIS Express quick reference. Dostupné z WWW: <<http://www.ariscommunity.com/aris-express/poster>>.

20. OMG. *BPMN 2.0* [online]. 2011 [cit. 2011-12-04]. Dostupné z WWW: <<http://www.omg.org/spec/BPMN/2.0>>.
21. SAP Česká republika [online]. c2011 [cit. 2011-12-04]. SAP NETWEAVER. Dostupné z WWW: <<http://www.sap.com/cz/platform/netweaver/index.epx>>.
22. *Software Design Tutorials* [online]. c2011 [cit. 2011-12-04]. Introduction to OOD. Dostupné z WWW: <<http://www.smartdraw.com/resources/tutorials/>>.
23. *UML tools for software development and modelling* [online]. c2011 [cit. 2011-12-04]. The Business Process Model. Dostupné z WWW: <[http://www.sparxsystems.com/business\\_process\\_model.html](http://www.sparxsystems.com/business_process_model.html)>.
24. WHITE, Stephen. *Introduction to BPMN* [online]. 2006 [cit. 2011-12-04]. Dostupné z WWW: <[http://www.bpmn.org/Documents/OMG\\_BPMN\\_Tutorial.pdf](http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf)>.
25. Workflow Management Coalition. *Terminology & Glossary* [online]. Hampshire : Workflow Management Coalition, 1999 [cit. 2011-09-11]. Dostupné z WWW: <<http://www.wfmc.org/Download-document/WFMC-TC-1011-Ver-3-Terminology-and-Glossary-English.html>>.

## 9 – Přílohy

**Příloha 1** – Seznam obrázků

**Příloha 2** – Seznam tabulek

**Příloha 3** – Struktura rozpisů dle formátu APF

**Příloha 4** – Charakteristika případů užití pro administraci uživatelů a systému

**Příloha 5** – UML diagramy základních, alternativních a rozšiřujících scénářů případů užití

**Příloha 6** – Scénáře případů užití

### **Příloha 1 – Seznam obrázků:**

Obrázek 1: <i>Vztahy mezi základními pojmy procesního řízení</i> .....	10
Obrázek 2: <i>Základní schéma podnikového procesu</i> .....	11
Obrázek 3: <i>Průběžné zlepšování procesů</i> .....	16
Obrázek 4: <i>Schéma zásadního reengineeringu</i> .....	17
Obrázek 5: <i>Životní cyklus BPM</i> .....	18
Obrázek 6: <i>Přehled notace BPMN</i> .....	26
Obrázek 7: <i>Základní elementy EPC</i> .....	27
Obrázek 8: <i>Rozšiřující elementy EPC</i> .....	27
Obrázek 9: <i>Prvky diagramu aktivit</i> .....	29
Obrázek 10: <i>Prvky notace Eriksson – Penker</i> .....	31
Obrázek 11: <i>Princip SASS</i> .....	33
Obrázek 12: <i>Notace pro DFD</i> .....	34
Obrázek 13: <i>Princip datově strukturovaného přístupu (DSSD)</i> .....	34
Obrázek 14: <i>Princip YMSA</i> .....	35
Obrázek 15: <i>Diagram tříd v notaci OOD</i> .....	37
Obrázek 16: <i>Historický vývoj UML</i> .....	38
Obrázek 17: <i>Struktura UP</i> .....	40
Obrázek 18: <i>Objem práce během fází životního cyklu projektu</i> .....	40
Obrázek 19: <i>Ornamenty</i> .....	42
Obrázek 20: <i>Příklad označené hodnoty</i> .....	43
Obrázek 21: <i>Architektura 4+1</i> .....	44
Obrázek 22: <i>Rozdělení diagramů UML 2.4</i> .....	45
Obrázek 23: <i>Procesní mapa Úseku správy příspěvků</i> .....	49
Obrázek 24: <i>BPD – zpracování rozpisů přijatých elektronickou cestou</i> .....	57
Obrázek 25: <i>BPD – zpracování papírových rozpisů</i> .....	58
Obrázek 26: <i>Use case diagram - činnosti v procesu zpracování rozpisů</i> .....	59
Obrázek 27: <i>BPD - Diagram nového procesu</i> .....	62
Obrázek 28: <i>BPD - Diagram generování rozpisů</i> .....	62
Obrázek 29: <i>UseCase diagram</i> .....	65
Obrázek 30: <i>Návrh rozpisu ve formátu xls</i> .....	67
Obrázek 31: <i>Diagram tříd</i> .....	69

### **Příloha 2 – Seznam tabulek**

Tabulka 1: <i>Typy, způsob řízení a všeobecná charakteristika podnikových procesů</i> .....	15
Tabulka 2: <i>Porovnání průběžného zlepšování procesů, BPR a BPM</i> .....	19
Tabulka 3: <i>Vnější události a reakce v ÚSP</i> .....	48

### Příloha 3 – Struktura rozpisů dle formátu APF

První řádek rozpisů ve formátu APF představuje sumární větu, která obsahuje souhrnné informace o platbě. Ostatní řádky jednoznačně identifikují jednotlivé příspěvky, z nichž se příslušná hromadná platba skládá. Sumární věta obsahuje 13 položek, jejichž význam je uveden v tabulce 5.1. Ostatních řádky se skládají z 8 položek uvedených v tabulce 5.2. Formát APF jednoznačně stanovuje pouze počet položek řádků rozpisu, jejich pozici a význam. Počet znaků jednotlivých položek a povinnost jejich vyplnění je dána jejich podstatou a vlastnostmi ISPF.

*Položky sumární věty APF dle nastavení IS PF, zdroj: autor, 2001*

Pozice	Označení	Počet znaků	Význam
1	uvozující znak	1	znak S (vychází z definice formátu APF)
2	IČ	max. 10	IČ zaměstnavatele doplněné levostrannými nulami na 10 míst
3	ID	max. 10	ID zaměstnavatele dle číselníku penzijního fondu doplněné levostrannými nulami na 10 míst. ID slouží pro rozlišení různých divizí jednoho zaměstnavatele.
4	název	max. 50	název zaměstnavatele
5	IČ PF	8	IČ penzijního fondu
6	název PF	max. 30	název penzijního fondu
10	celková suma	max. 12	celková suma příspěvků obsažených v hromadné platbě
7	konstantní symbol	4	konstantní symbol poukázané hromadné platby
8	variabilní symbol	max. 10	variabilní symbol poukázané hromadné platby
9	specifický symbol	max. 10	specifický symbol poukázané hromadné platby
11	období	6	kalendářní měsíc, na který mají být jednotlivé příspěvky připsány účastníkům.
12	pořadí	max. 2	pořadí rozpisu v daném kalendářním měsíci pro případ zaslání více rozpisů k jedné nebo více hromadným platbám za jeden kalendářní měsíc.
13	počet příspěvků	max. 6	počet jednotlivých příspěvků, na které má být odpovídající hromadná platba rozdělena (počet příspěvkových řádků rozpisu)

*Položky jednotlivých řádků rozpisu ve formátu APF dle nastavení IS PF, zdroj: autor, 2011*

<b>Pozice</b>	<b>Označení</b>	<b>Počet znaků</b>	<b>Význam</b>
1	uvozující znak	1	znak U, Z nebo T rozlišující typ příspěvku (U – příspěvek účastníka, Z – příspěvek zaměstnavatele, T – příspěvek třetí osoby)
2	číslo smlouvy	10	číslo smlouvy o penzijním připojištění daného účastníka
3	rodné číslo	9 nebo 10	rodné číslo účastníka
4	příjmení	max. 30	příjmení účastníka
5	jméno	max. 30	jméno účastníka
6	částka	max.10	částka příspěvku
7*	začátek období	6	první kalendářní měsíc na který je příspěvek určen
8*	konec období	6	poslední kalendářní měsíc na který je příspěvek určen

*\* nepovinné údaje – ve zvoleném penzijním fondu nejsou využívány*

## **Příloha 4 – Charakteristika případů užití pro administraci uživatelů a systému**

### **UC07 Spravovat zaměstnavatele a UC08 Založit zaměstnavatele**

Případy užití *UC07 Spravovat zaměstnavatele* a *UC08 Založit zaměstnavatele* reprezentují využití systému pro zakládání nových uživatelů, úpravu uživatelských dat a rušení uživatelů v roli zaměstnavatelů. Aktérem tohoto UC je referent označující roli určenou pro pracovníky ÚSP. *UC07 Spravovat zaměstnavatele* také zahrnuje případ užití *UCi07 Vyhledat zaměstnavatele*, který představuje vyhledání uživatele před jeho samotnou úpravou nebo zrušením.

### **UC09 Spravovat interního uživatele a UC10 Založit interního uživatele**

Tyto dva případy užití slouží pro zakládání, úpravu uživatelských dat a rušení interních uživatelů systému. Interní uživatel vystupuje v roli referenta nebo administrátora systému, jenž je rovněž aktérem uvedených *UC09* a *UC10*. Podobně jako v předchozím případě *UC09 Spravovat interního uživatele* zahrnuje případ užití *UCi09 Vyhledat interního uživatele*, který představuje vyhledání uživatele před jeho samotnou úpravou nebo zrušením.

## UC11 Nastavit generování

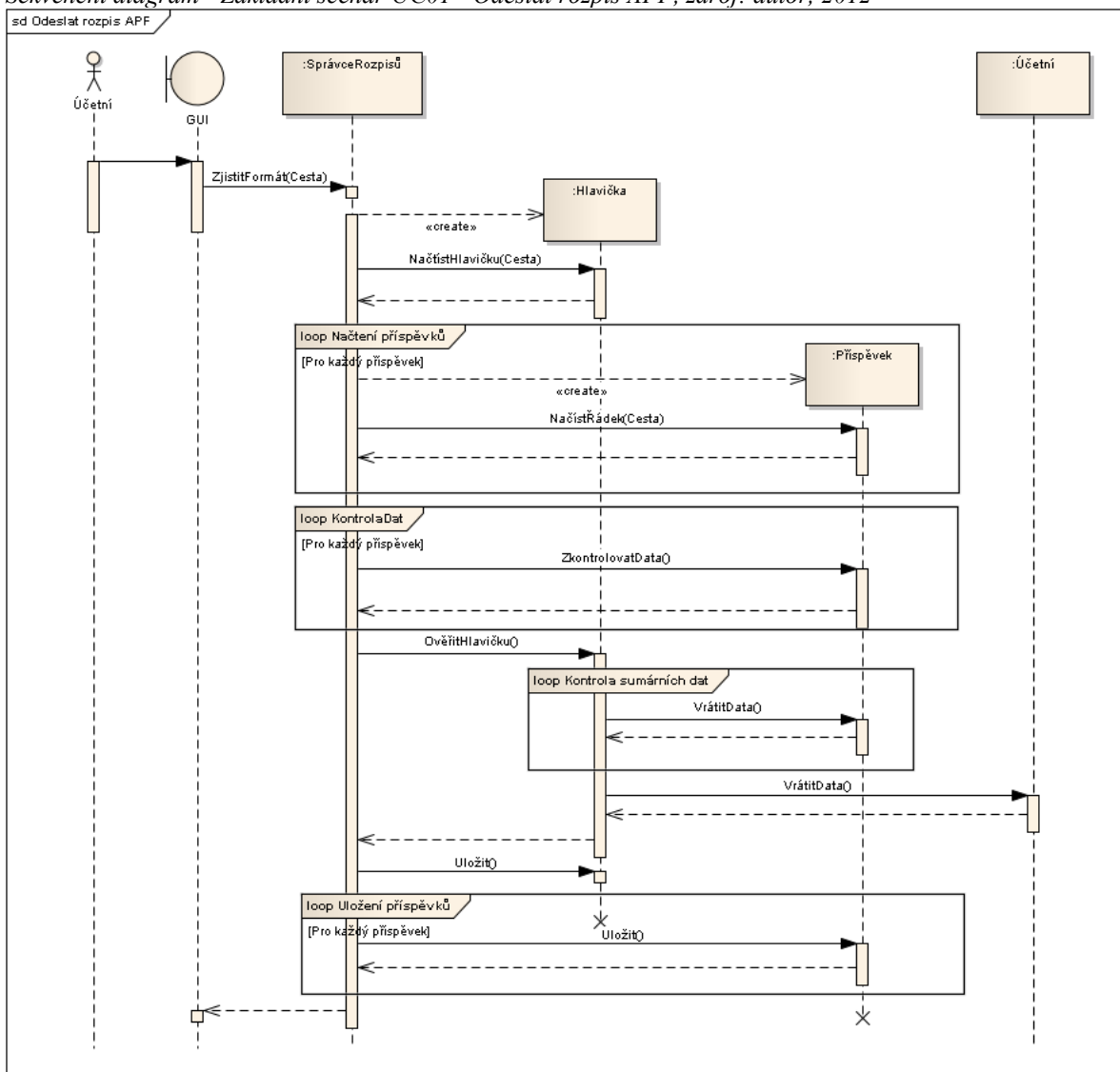
Tento případ užití představuje možnost nastavení intervalu generování APF rozpisů z uložených dat a výstupního adresáře. Aktérem tohoto případu užití je Administrátor.

## Příloha 5 – UML diagramy základních, alternativních a rozšiřujících scénářů případů užití UC01 – UC06

### Sekvenční diagram: Odeslat rozpis APF

Na níže uvedeném sekvenčním diagramu je znázorněno předávání zpráv mezi objekty systému při realizaci základního scénáře případu užití UC01 – Odeslat rozpis APF.

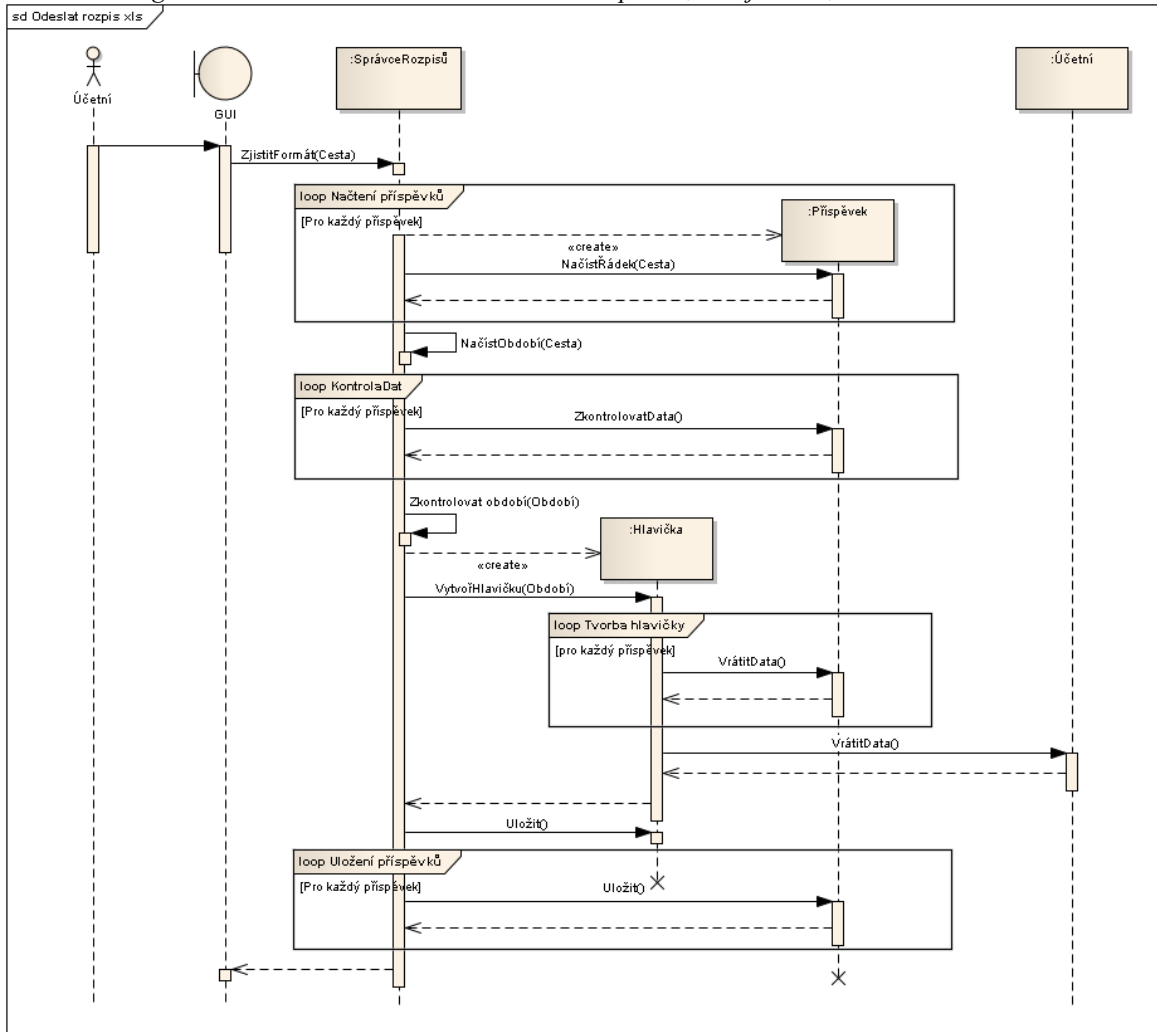
Sekvenční diagram - Základní scénář UC01 - Odeslat rozpis APF, zdroj: autor, 2012



## Sekvenční diagram: Odeslat rozpis xls

Sekvenční diagram na následujícím obrázku ilustruje základní scénář případu užití UC02 – odeslat rozpis xls.

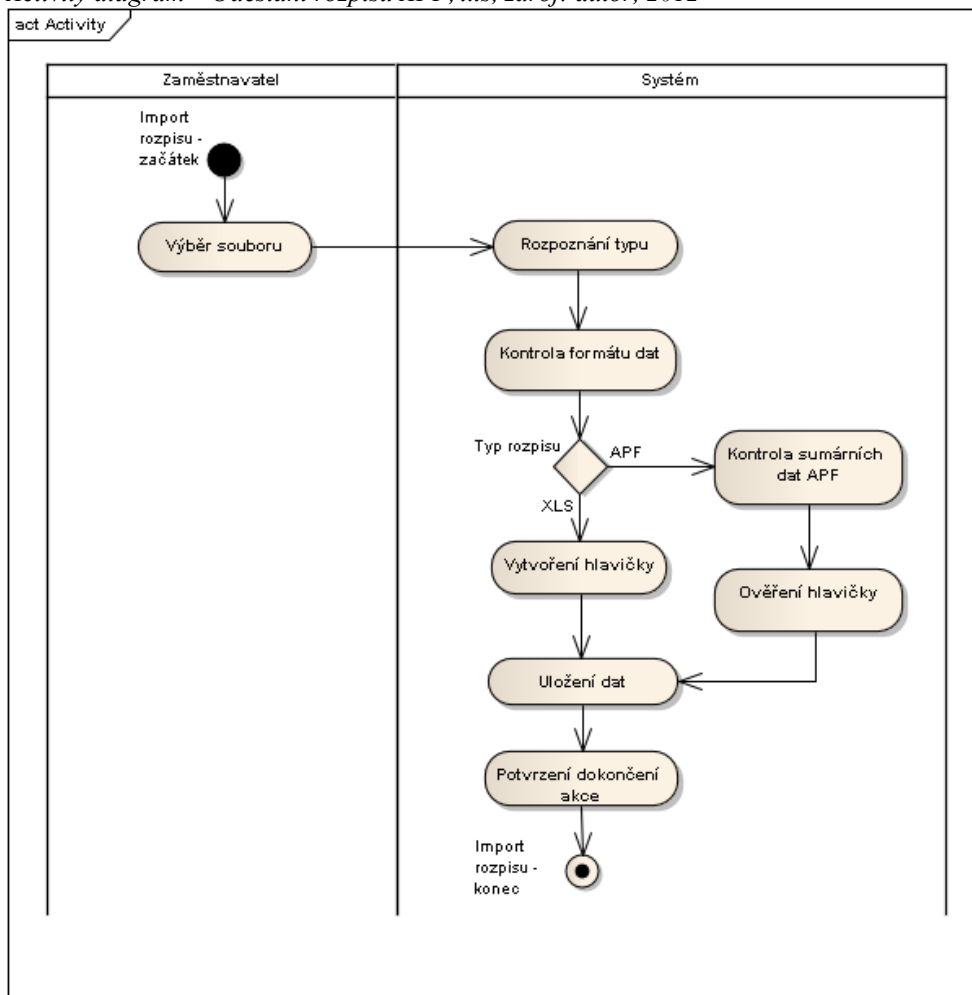
Sekvenční diagram - Základní scénář UC02 - Odeslat rozpis xls, zdroj: autor, 2012



## Activity diagram: Odeslat rozpis APF, xls

Dále je uveden UML activity diagram představující posloupnost činností na straně systému a uživatele při odeslání rozpisu ve formátu APF nebo xls.

*Activity diagram – Odeslání rozpisu APF, xls, zdroj: autor, 2012*

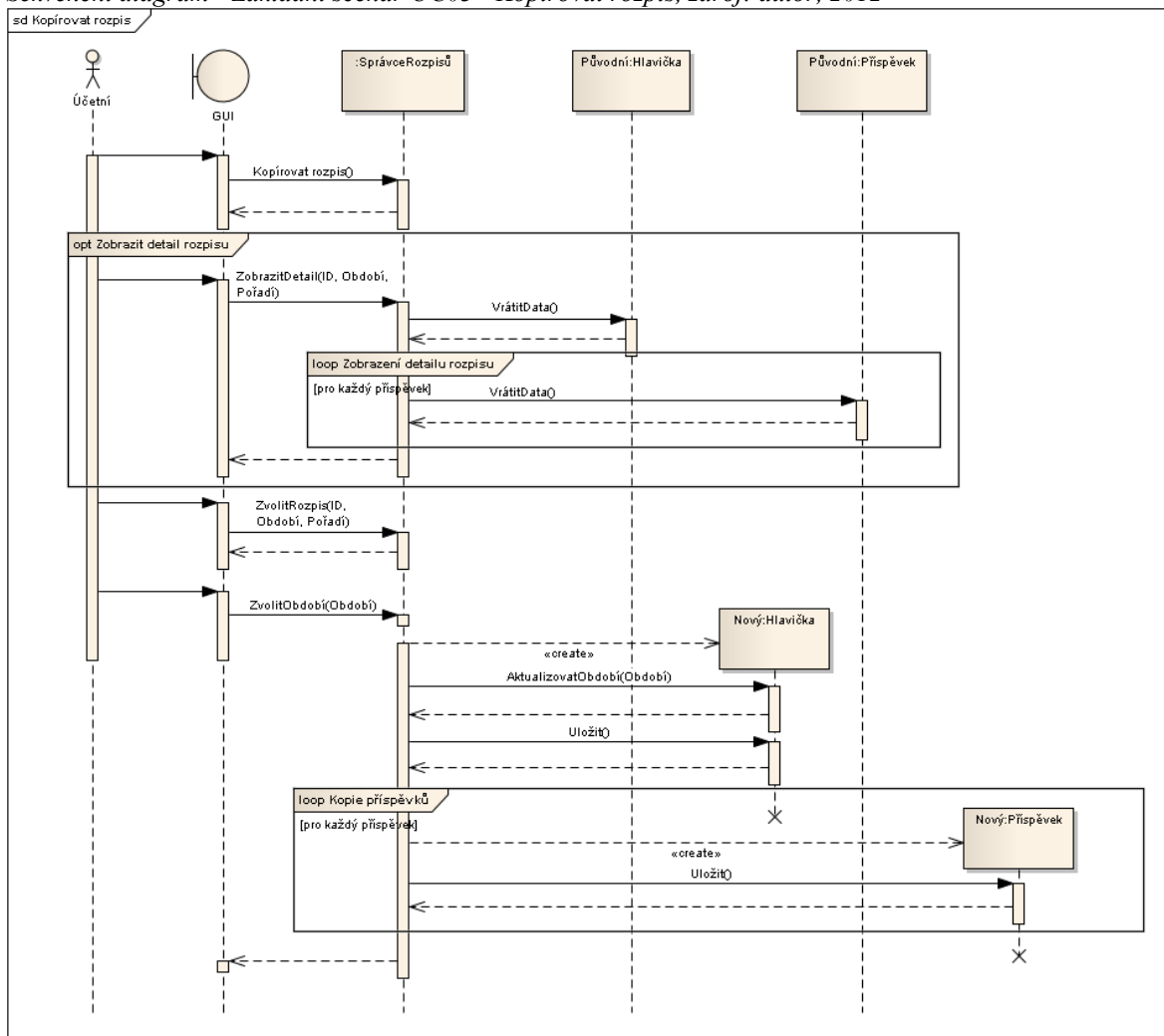




## Sekvenční diagram: Kopírovat rozpis

Na sekvenčním diagramu níže je zobrazena interakce při kopírování rozpisu bez provedení změn s možností prohlédnutí jeho detailu.

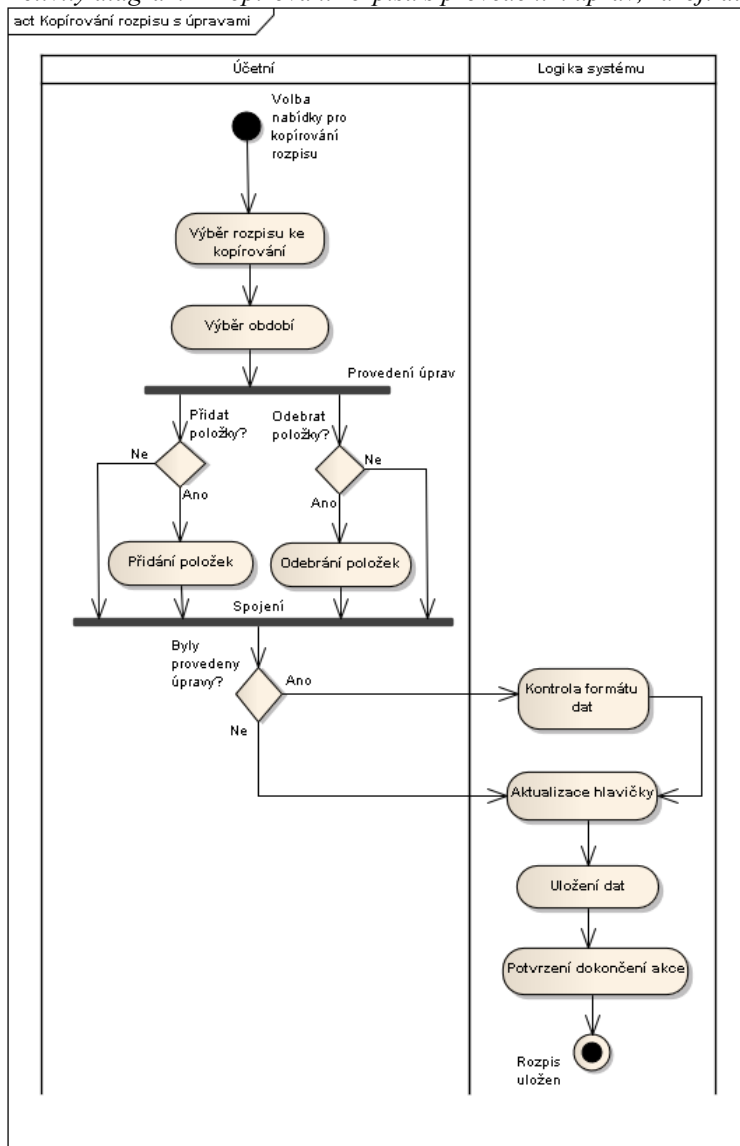
Sekvenční diagram - Základní scénář UC03 - Kopírovat rozpis, zdroj: autor, 2012



## Activity diagram: Kopírování rozpisu s provedením úprav

Na následujícím obrázku je uveden UML activity diagram představující činnosti při kopírování rozpisu s možným provedením úprav.

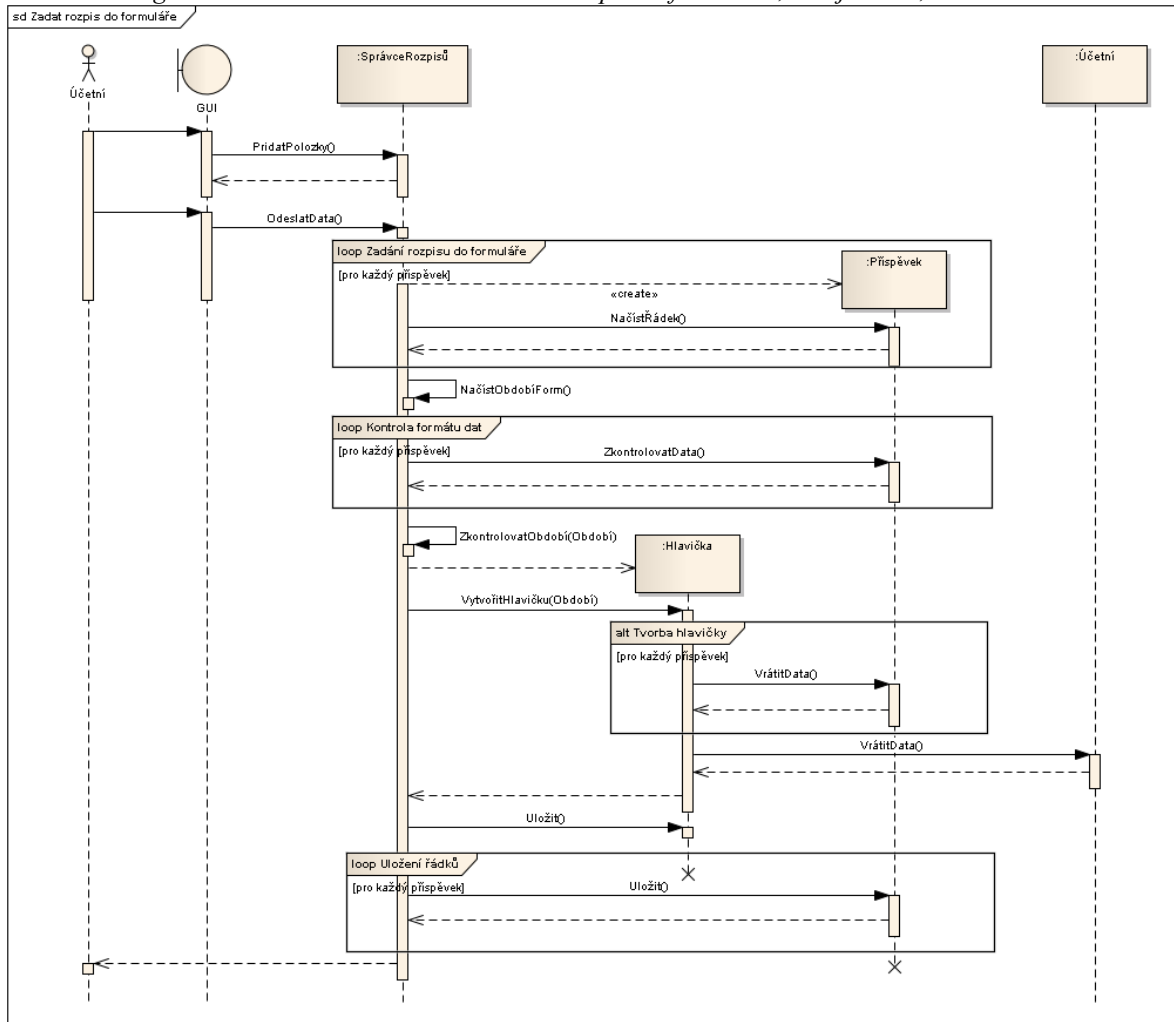
*Activity diagram – Kopírování rozpisu s provedením úprav, zdroj: autor, 2012*



## Sekvenční diagram: Zadat rozpis do formuláře

Sekvenční diagram uvedený na obrázku níže zobrazuje základní scénář případu užití UC04 - Zadat rozpis do formuláře.

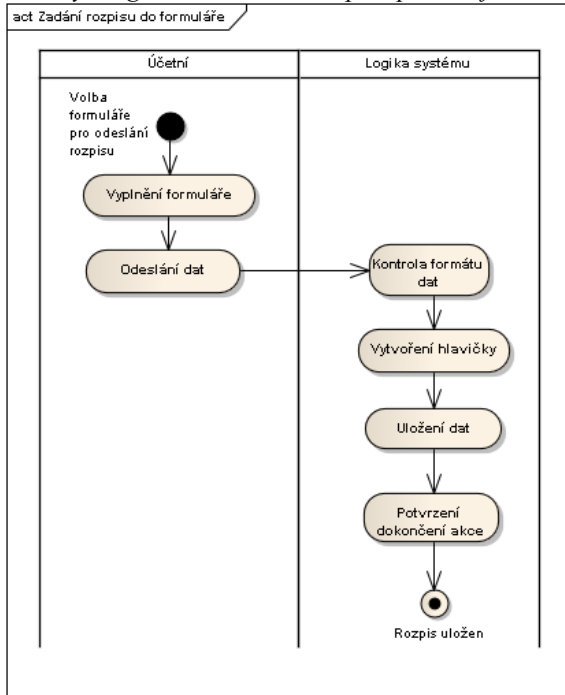
Sekvenční diagram - Základní scénář UC04 - Zadat rozpis do formuláře, zdroj: autor, 2012



## Activity diagram: Odeslání rozpisu pomocí formuláře

Následující obrázek ilustruje základní scénář posloupnosti aktivit při odeslání rozpisu prostřednictvím formuláře.

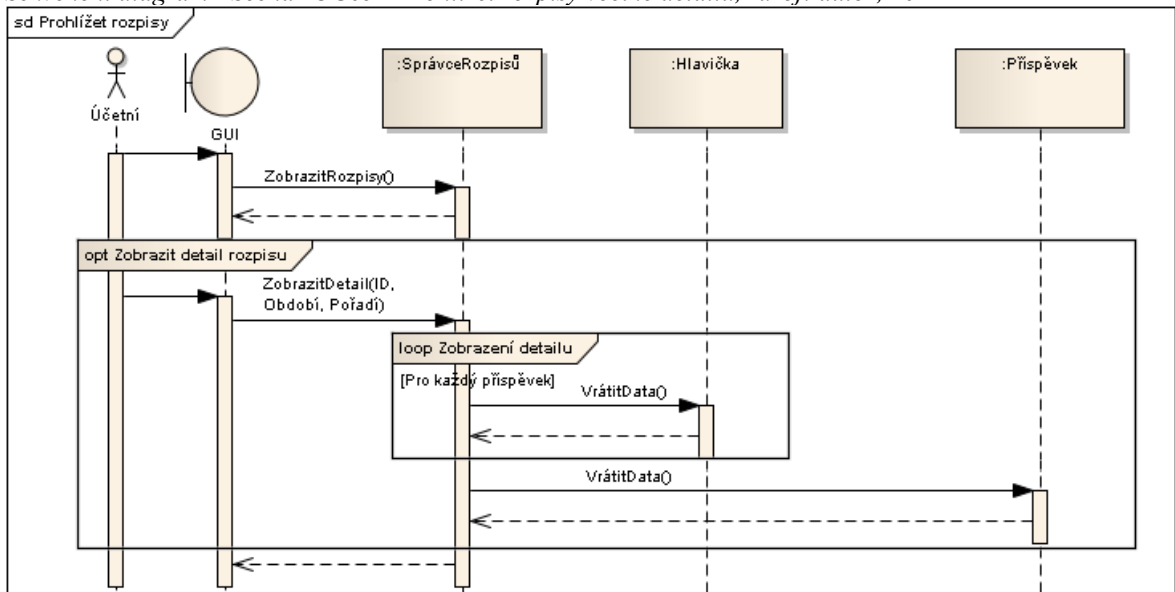
Activity diagram – Odeslání rozpisu pomocí formuláře, zdroj: autor, 2012



## Sekvenční diagram: Prohlížet rozpis včetně detailu

Na níže uvedeném sekvenčním diagramu je zobrazeno předávání zpráv mezi objekty při zobrazování vybraného rozpisu včetně jeho detailu.

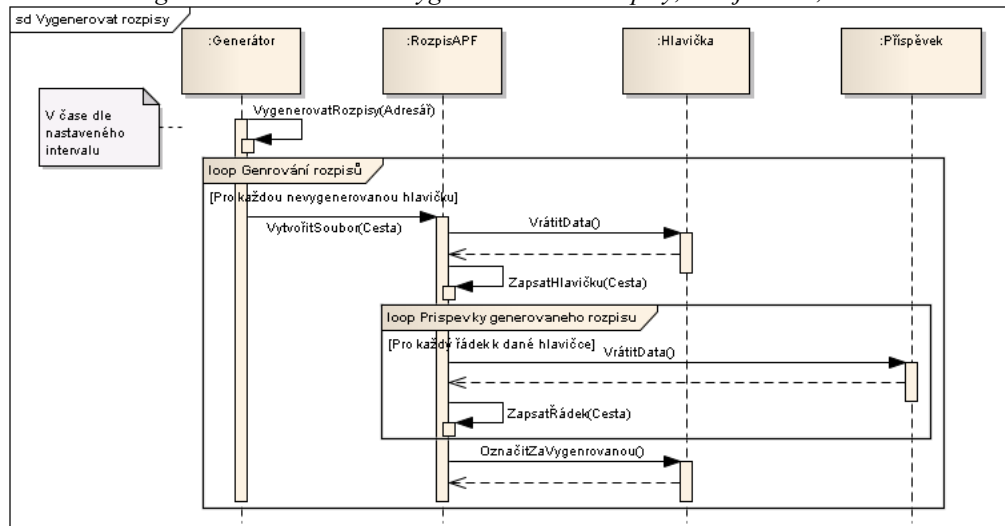
Sekvenční diagram - Scénář UC05 - Prohlížet rozpis včetně detailu, zdroj: autor, 2012



## Sekvenční diagram: Vygenerovat APF rozpisy

Tento sekvenční diagram představuje posloupnost předávání zpráv mezi objekty při pravidelně prováděném generování uložených rozpisů.

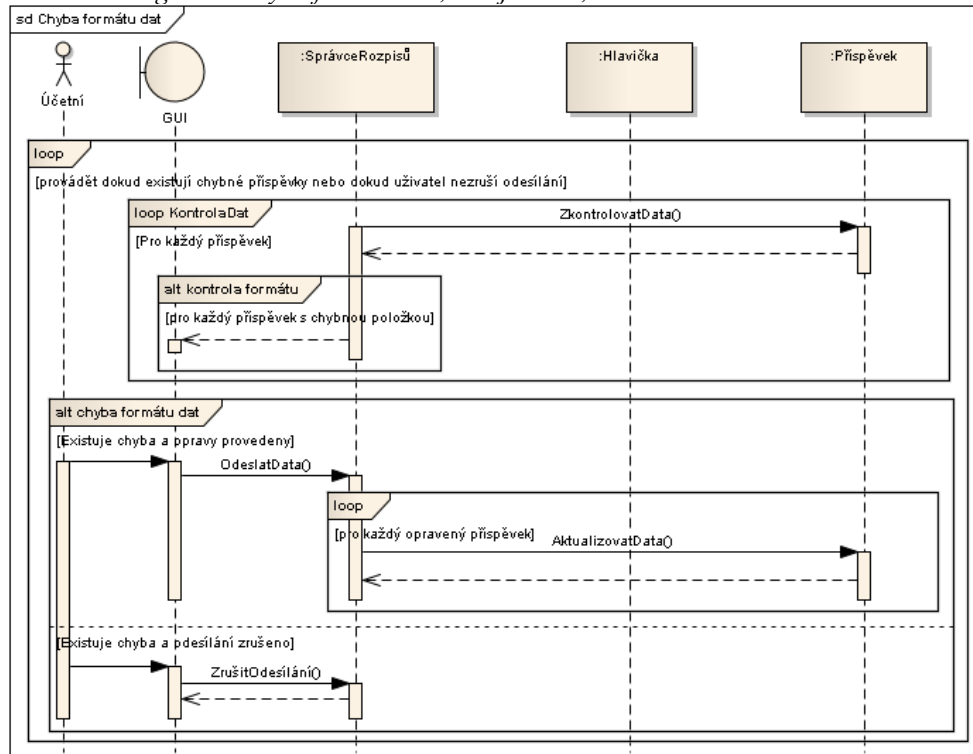
Sekvenční diagram - Scénář UC06 - Vygenerovat APF rozpisy, zdroj: autor, 2012



## Sekvenční diagram: Chyba formátu dat

Níže zobrazená interakce se uplatňuje v alternativních scénářích případů užití UC01 – Odeslat rozpis APF, UC02 – Odeslat rozpis xls, UCe1\_03 Přidat položky a UC04 Zadat rozpis do formuláře.

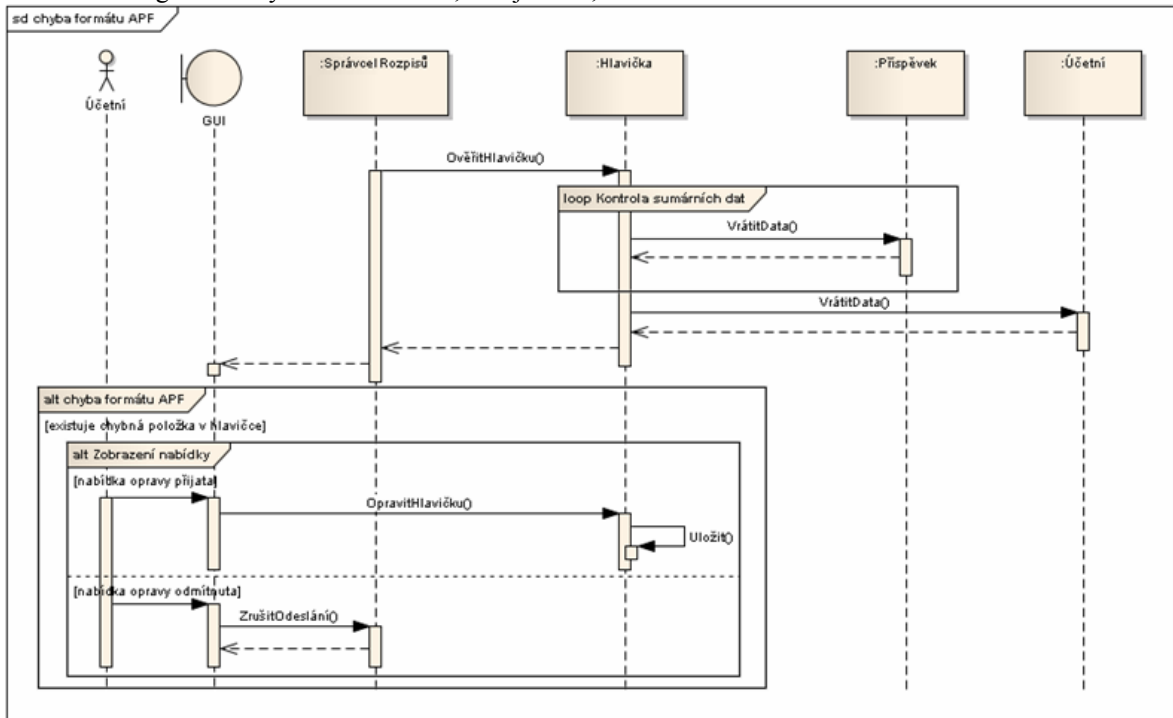
Sekvenční diagram – Chyba formátu dat, zdroj: autor, 2012



## Sekvenční diagram: Chyba formátu APF

Tento diagram zobrazuje interakci uplatňující se v alternativním scénáři *UC01 – Odeslat rozpis APF*.

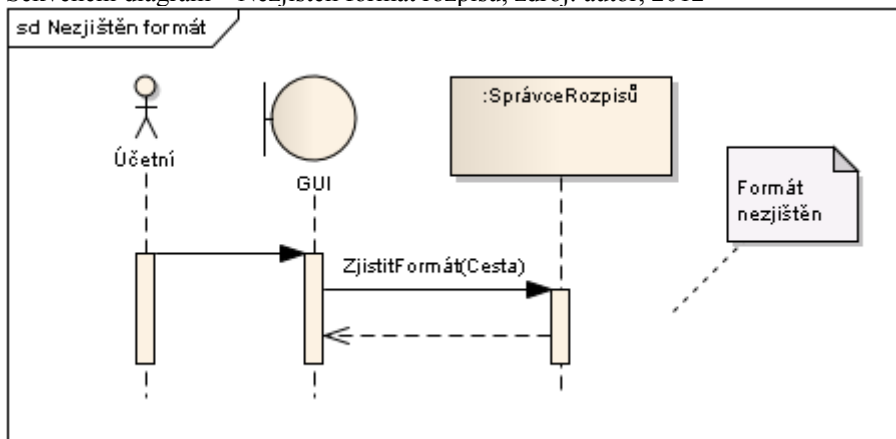
Sekvenční diagram – Chyba formátu APF, zdroj: autor, 2012



## Sekvenční diagram: Nejistěn formát rozpisu

Níže zobrazená interakce se uplatňuje v alternativním scénáři vloženého případu užití *UCi01\_02 Zvolit rozpis*.

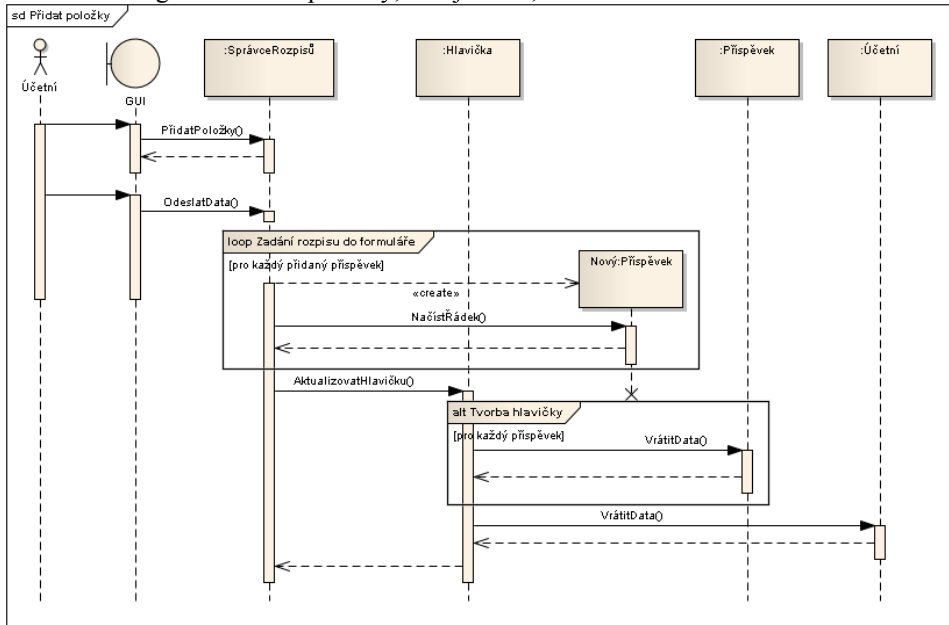
Sekvenční diagram – Nejistěn formát rozpisu, zdroj: autor, 2012



## Sekvenční diagram: Přidat položky

Tato interakce zobrazuje scénář rozšiřujícího případu užití *UCe1\_03 – Přidat položky* při tvorbě kopie rozpisu.

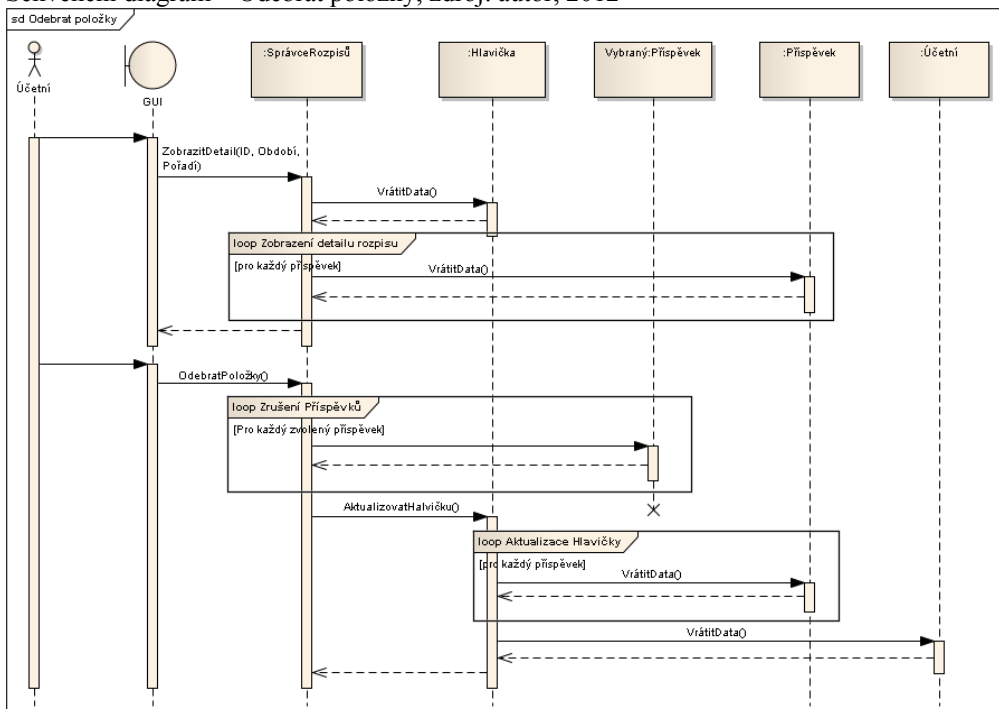
Sekvenční diagram – Přidat položky, zdroj: autor, 2012



## Sekvenční diagram: Odebrat položky

Níže zobrazené předávání zpráv reprezentuje rozšiřující případ užití *UCe2\_03 – Odebrat položky* při tvorbě kopie rozpisu.

Sekvenční diagram – Odebrat položky, zdroj: autor, 2012



## Příloha 6 – Scénáře případů užití

Případ užití: OdeslatRozpisAPF
ID: UC01
Stručný popis: Zaměstnavatel odešle APF rozpis a systém uloží jeho obsah do databáze
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: 1. Zahrnout (ZvolitRozpis) 2. Systém ověří správnost formátu dat na řádcích s příspěvky 3. Systém ověří zda údaje na jednotlivých řádcích odpovídají sumárnímu řádku dle definice APF 4. Systém uloží informace ze souboru do databáze 5. Systém zobrazí potvrzení o úspěšném odeslání
Výstupní podmínky: 1. Systém načte data z rozpisu do databáze 2. Systém zobrazil potvrzení úspěšného odeslání
Alternativní scénáře: 1. Chyba formátu dat 2. Chyba formátu APF

Případ užití: ZvolitRozpis
ID: UCi01_02
Stručný popis: Zaměstnavatel zvolí rozpis k odeslání ze svého počítače
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: 1. Zaměstnavatel vybere volbu „odeslat rozpis ze souboru“ 2. Systém nabídne Zaměstnavateli možnost výběru souboru (tlačítko „Procházet“) 3. Zaměstnavatel vybere soubor s rozpisem a výběr potvrdí 4. Zaměstnavatel odešle rozpis 5. Systém ověří strukturu rozpisu
Výstupní podmínky: 1. Systém zjistil formát souboru s rozpisem (APF nebo xls)
Alternativní scénáře: 1. Systém nezjistil formát souboru



Alternativní scénář: Zvolit rozpis: Systém nezjistil formát rozpisu
ID: UCi01_02.1
Stručný popis: Systém informuje Zaměstnavatele, že zvolil soubor, který nemá strukturu akceptovatelného rozpisu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Zaměstnavatel zvolil soubor s chybnou strukturou
Alternativní scénář: 1. Alternativní scénář začíná krokem 3 hlavního scénáře 2. Systém informuje Zaměstnavatele, že zvolil soubor, který nemá strukturu akceptovatelného rozpisu
Výstupní podmínky: 1. Odeslání rozpisu nebylo provedeno

Alternativní scénář: OdeslatRozpisAPF: Chyba formátu dat
ID: UC01.1
Stručný popis: Rozpis obsahuje na řádcích s příspěvky údaje v chybném formátu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Odesílaný rozpis je ve formátu APF 2. Odesílaný rozpis obsahuje řádky s příspěvky obsahující údaje v chybném formátu
Alternativní scénář: 1. Alternativní scénář začíná krokem 2 hlavního scénáře 2. Systém zobrazí seznam řádků s editovatelnými chybnými položkami 3. KDYŽ zaměstnavatel potvrdí provedené opravy 3.1 Systém pokračuje bodem 2 hlavního scénáře. 4. KDYŽ zaměstnavatel zruší provádění opravy 4.1 Odesílání je ukončeno
Výstupní podmínky: Nejsou

Alternativní scénář: OdeslatRozpisAPF: Chyba formátu APF
ID: UC01.2
Stručný popis: Hlavička rozpisu obsahuje některé údaje, které dle formátu APF neodpovídají údajům na jednotlivých řádcích s příspěvkem
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Odesílaný rozpis je ve formátu APF 2. Odesílaný rozpis obsahuje alespoň jednu chybnou položku v hlavičce
Alternativní scénář: 1. Alternativní scénář začíná krokem 3 hlavního scénáře 2. Systém zobrazí položky hlavičky, které neodpovídají formátu APF a zároveň zobrazí hodnoty, kterými budou tyto položky nahrazeny. 3. KDYŽ zaměstnavatel potvrdí nabízené úpravy 3.1 Systém pokračuje bodem 4 hlavního scénáře. 4. KDYŽ zaměstnavatel zruší provádění opravy
Výstupní podmínky: Nejsou

Případ užití: OdeslatRozpisXLS
ID: UC02
Stručný popis: Zaměstnavatel odešle xls rozpis do ÚSP a systém uloží jeho obsah do databáze
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: 1. Zahrnout (ZvolitRozpis) 2. Systém ověří správnost formátu dat na řádcích s příspěvkem 3. Systém určí hodnoty v hlavičce 4. Systém uloží hlavičku a informace ze souboru do databáze 5. Systém zobrazí potvrzení o úspěšném odeslání
Výstupní podmínky: 1. Systém načte data z rozpisu do databáze 2. Systém zobrazil potvrzení úspěšného odeslání
Alternativní scénáře: 3. Chyba formátu dat

Alternativní scénář: OdeslatRozpisXLS: Chyba formátu dat
ID: UC02.1
Stručný popis: Rozpis obsahuje na řádcích s příspěvky údaje v chybném formátu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Odesílaný rozpis je ve formátu xls 2. Odesílaný rozpis obsahuje údaje o příspěvcích nebo období v chybném formátu
Alternativní scénář: 1. Alternativní scénář začíná krokem 2 hlavního scénáře 2. Systém zobrazí seznam řádků s editovatelnými chybnými položkami 3. KDYŽ zaměstnavatel potvrdí provedené opravy 3.1 Systém pokračuje bodem 2 hlavního scénáře. 4. KDYŽ zaměstnavatel zruší provádění opravy 4.1 Odesílání je ukončeno
Výstupní podmínky: Nejsou

Případ užití: Kopírovat rozpis
ID: 03
Stručný popis: Zaměstnavatel odešle kopii již dříve zasláního rozpisu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: 1. Zaměstnavatel zvolí kopírování rozpisu 2. Systém zobrazí přehled rozpisů ke kopírování místo rozšíření: Prohlížet detail rozpisu 3. Zaměstnavatel zvolí rozpis ke kopírování 4. Zaměstnavatel zvolí období místo rozšíření: Přidat položky místo rozšíření: Odebrat položky 5. Systém uloží odeslané údaje do databáze. 6. Systém zobrazí potvrzení o úspěšném odeslání
Výstupní podmínky: 1. Údaje z rozpisu byly uloženy
Alternativní scénáře: Nejsou

Rozšiřující případ užití: Prohlížet detail rozpisu
ID: UCe03_08
Stručný popis: Zaměstnavatel zobrazuje detail vybraného rozpisu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Zaměstnavatel je přihlášen do systému 2. Je zobrazen přehled rozpisů
Hlavní scénář: 1. Zaměstnavatel zvolí detail vybraného rozpisu 2. Systém zobrazí jednotlivé příspěvky
Výstupní podmínky: 1. Systém zobrazil jednotlivé příspěvky
Alternativní scénáře: Nejsou

Rozšiřující případ užití: Přidat položky
ID: UCe1_03
Stručný popis: Zaměstnavatel přidává položky do vybraného rozpisu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Zaměstnavatel je přihlášen do systému 2. Je zvolen rozpis ke kopírování
Hlavní scénář: 1. Zaměstnavatel zvolí přidání příspěvků do zvoleného rozpisu 2. Systém zobrazí formulář pro zadávání údajů 3. DOKUD není vkládání nových příspěvků potvrzeno 3.1 Systém umožňuje vkládat nové příspěvky 4. Systém ověří správnost formátu dat v rozpisu 5. Systém aktualizuje hlavičku
Výstupní podmínky: 1. Do rozpisu byly přidány nové položky 2. Hlavička byla aktualizována
Alternativní scénáře: 1. Chyba formátu dat

Alternativní scénář: Rozšiřující případ užití: Přidat položky: Chyba formátu dat
ID: UCe1_03
Stručný popis: Rozšířený rozpis obsahuje údaje o příspěvcích v chybném formátu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Zaměstnavatel přidal do rozpisu údaje o příspěvcích v chybném formátu
Alternativní scénář: 1. Alternativní scénář začíná krokem 4 hlavního scénáře 2. Systém zobrazí seznam příspěvků s editovatelnými položkami v chybném formátu 3. KDYŽ zaměstnavatel potvrdí provedené opravy 3.1 Systém pokračuje bodem 4 hlavního scénáře. 4. KDYŽ zaměstnavatel zruší provádění opravy 4.1 Odesílání je ukončeno
Výstupní podmínky: Nejsou

Rozšiřující případ užití: Odebrat položky
ID: UCe2_03
Stručný popis: Zaměstnavatel odebírá položky z vybraného rozpisu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Zaměstnavatel je přihlášen do systému 2. Je zvolen rozpis ke kopírování
Hlavní scénář: 1. Zaměstnavatel zvolí odebrání příspěvků ze zvoleného rozpisu 2. Systém zobrazí příspěvky 3. Zaměstnavatel vybere příspěvky, které mají být odstraněny a změny potvrdí 4. Systém odstraní vybrané příspěvky 5. Systém aktualizuje hlavičku
Výstupní podmínky: 1. Vybrané příspěvky byly odstraněny 2. Hlavička byla aktualizována
Alternativní scénáře: Nejsou

Případ užití: Zadat rozpis do formuláře
ID: UC04
Stručný popis: Zaměstnavatel
Hlavní aktéři: Zaměstnavatel zadává údaje o jednotlivých příspěvcích do systému prostřednictvím formuláře
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: <ol style="list-style-type: none"> <li>1. Zaměstnavatel vybere volbu pro zobrazení formuláře pro vložení údajů o rozpisu</li> <li>2. Systém zobrazí formulář</li> <li>3. DOKUD není vkládání údajů potvrzeno <ol style="list-style-type: none"> <li>3.1 Systém umožňuje vkládat nové příspěvky</li> </ol> </li> <li>4. Systém ověří správnost formátu zadaných údajů</li> <li>5. Systém uloží zadaná data do databáze</li> <li>6. Systém zobrazí potvrzení o úspěšném odeslání rozpisu</li> </ol>
Výstupní podmínky: <ol style="list-style-type: none"> <li>1. Systém uložil zadaná data</li> <li>2. Systém zobrazil potvrzení úspěšného odeslání</li> </ol>
Alternativní scénáře: Chyba formátu dat

Alternativní scénář: Zadat rozpis do formuláře: Chyba formátu dat
ID: UC04.1
Stručný popis: Rozpis obsahuje údaje o příspěvcích v chybném formátu
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Odesílaný rozpis obsahuje položky r.č. nebo č. smlouvy, které jsou uvedeny nesprávně
Alternativní scénář: <ol style="list-style-type: none"> <li>1. Alternativní scénář začíná krokem 4 hlavního scénáře</li> <li>2. Systém zobrazí seznam příspěvků s editovatelnými položkami v chybném formátu</li> <li>3. KDYŽ zaměstnavatel potvrdí provedené opravy <ol style="list-style-type: none"> <li>3.1 Systém pokračuje bodem 4 hlavního scénáře.</li> </ol> </li> <li>4. KDYŽ zaměstnavatel zruší provádění opravy <ol style="list-style-type: none"> <li>4.1 Odesílání je ukončeno</li> </ol> </li> </ol>
Výstupní podmínky: Nejsou

Případ užití: Prohlížet rozpisy
ID: UC08
Stručný popis: Zaměstnavatel zobrazuje přehled odeslaných rozpisů
Hlavní aktéři: Zaměstnavatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Zaměstnavatel je přihlášen do systému
Hlavní scénář: 1. Zaměstnavatel zvolí přehled rozpisů 2. Systém zobrazí přehled rozpisů místo rozšíření: Zobrazit detail rozpisu
Výstupní podmínky: 2. Systém zobrazil přehled rozpisů
Alternativní scénáře: Nejsou

Případ užití: Vygenerovat rozpisy APF
ID: UC10
Stručný popis: Systém vygeneruje rozpisy, které byly zadány zaměstnavateli do systému, do importního adresáře ISPF ve formátu APF
Hlavní aktéři: Čas
Vedlejší aktéři: Žádní
Vstupní podmínky: 1. Nastal čas naplánované akce pro vygenerování rozpisů vložených do systému
Hlavní scénář: 1. Systém zjistí nevygenerované rozpisy 2. Systém vygeneruje rozpisy do importního adresáře ISPF
Výstupní podmínky: 1. Systém vygeneroval dosud nevygenerované rozpisy do importního adresáře ISPF
Alternativní scénáře: Nejsou