

# **Různé náhledy na procesy popsané metodou BPM**

## **Various Views on the BPM Method Processes**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2011

.....

## **Abstrakt**

Cílem práce bylo navrhnout zjednodušený model náhledu na byznys proces zachycený koordinačním modelem metody BPM a také navrhnout simulační náhled pro koordinační model. Práce provází od neformálních modelovacích metod byznys procesů přes semi-formální k formálním metodám. U každé metody ukazuje, jak zachytit byznys proces modely a co znamenají jednotlivé části modelu. Mezi hlavními formálními metodami je uvedena metoda BPM. Práce dále provází přes simulace byznys procesů pomocí modelů petriho sítě. Ukazuje, jak rozšířit koordinační model metody BPM pro potřeby simulací a simulačních náhledů. Vzhledem ke složitosti koordinačního modelu je také uvedeno, jakým způsobem náhled na byznys proces zjednodušit aktivitním diagramem.

**Klíčová slova:** byznys proces, BPM, aktivitní diagram, simulace, petriho síť, simulační náhled, BP studio

## **Abstract**

The aim of this thesis was design of simplified model at business processes captured by model of coordination of BPM method and design of simulation view for model of coordination. This thesis shows modelling methods for business process modelling by unformal, semiformal and formal methods. This thesis shows how to capture business processes by models of method and what is every part of model describing. Among main formal methods there is BPM method. This thesis shows simulation of business process by using Petri nets. It shows how to extend model of coordination of BPM method for needs simulation and simulation views. In view of complexity of the model of coordination is also shown how the view of business process simplification by activity diagrams.

**Keywords:** business process, BPM, activity diagram, simulation, Petri net, simulation view, BP studio

## Seznam použitých zkratk a symbolů

BP	– Business process
BPM	– Business process modeling
BPMN	– Business process model notation
CNC	– Computer numeric control
CPN	– Colored petri net
EPC	– Event-driven process chain
IDEF	– Integration definition
OMG	– Object management group
UML	– Unified modeling language

## Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Metody pro byznys modelování</b>	<b>4</b>
2.1	Neformální metody . . . . .	4
2.2	Semiformální metody . . . . .	4
2.3	Formální metody . . . . .	9
<b>3</b>	<b>Business process modeling</b>	<b>13</b>
3.1	Funkční model . . . . .	13
3.2	Objektový model . . . . .	14
3.3	Koordinační model . . . . .	14
3.4	BP studio . . . . .	15
<b>4</b>	<b>Simulační náhled</b>	<b>16</b>
4.1	Simulace modelu zachyceného petriho sítí . . . . .	16
4.2	Trvání aktivity . . . . .	18
4.3	Čekání na aktivitu . . . . .	18
4.4	Alternativní scénáře . . . . .	19
4.5	Zdroje v simulacích . . . . .	21
<b>5</b>	<b>Zjednodušený náhled na byznys proces</b>	<b>24</b>
5.1	Převod na petriho síť . . . . .	24
5.2	Převod na aktivitní diagram . . . . .	25
<b>6</b>	<b>Byznys proces výroby</b>	<b>34</b>
<b>7</b>	<b>Závěr</b>	<b>37</b>
<b>8</b>	<b>Reference</b>	<b>38</b>
	<b>Přílohy</b>	<b>38</b>
<b>A</b>	<b>Byznys proces výroby</b>	<b>39</b>

## Seznam obrázků

1	Byznys proces zachycený tabulkou . . . . .	4
2	IDEF0 - Diagram realizace zakázky . . . . .	6
3	Byznys proces zachycený modelem EPC . . . . .	7
4	Byznys proces zachycený aktivitním diagramem . . . . .	8
5	Byznys proces zachycený modelem BPMN . . . . .	10
6	Byznys proces zachycený konečným automatem . . . . .	11
7	Petriho síť . . . . .	12
8	Funkční model BPM . . . . .	13
9	Objektový model BPM . . . . .	14
10	Koordinační model BPM . . . . .	15
11	Simulace modelu zachycená petriho sítí . . . . .	17
12	Koordinační model rozšířený o čekání a trvání aktivity . . . . .	19
13	Aktivita koordinačního modelu s informacemi pro simulační náhled . . . . .	20
14	Přidání poolu sdílených zdrojů . . . . .	22
15	Zachycení poolu v petriho síti . . . . .	23
16	Řídící instance . . . . .	23
17	Zjednodušení koordinačního modelu petriho sítí . . . . .	24
18	Alternativní scénáře v petriho síti . . . . .	25
19	Základní konstrukce petriho sítí . . . . .	26
20	Základní blok . . . . .	27
21	Sekvence . . . . .	28
22	Alternativy . . . . .	28
23	Paralelismus . . . . .	29
24	Ukončení jedné větve paralelismu bez ukončení ostatních . . . . .	30
25	Cyklus . . . . .	31
26	Rozšířený aktivitní diagram . . . . .	32
27	Tok byznys procesu výroby . . . . .	35
28	Koordinační model BPM zachycující proces výroby . . . . .	40
29	Koordinační model BPM zachycující proces výroby s informacemi pro simulace . . . . .	41
30	Model byznys procesu výroby zachycený petriho sítí podle koordinačního modelu . . . . .	42
31	Zjednodušený náhled na byznys proces . . . . .	43
32	Zjednodušený náhled na byznys proces s přidanými objekty . . . . .	44

## 1 Úvod

Jako byznys proces můžeme označit množinu kroků, které za sebou nebo vedle sebe probíhají sekvenčně, a které realizují nějaký cíl. Tento může být v oblasti podnikání, kde dále může popisovat vztahy a funkce jednotlivých rolí. Byznys proces ale nemusí popisovat pouze podnikové procesy. Mohou se skrz něj také popisovat různé protokoly, od komunikačních až třeba po protokol návštěvy vysoce představeného politického představitele jiné země.

V každém podniku často existuje nějaký proces. Příkladem může být výroba automobilu, která se skládá ze spousty dílů a zdrojů, které do procesu vstupují. Pro zefektivnění, zkvalitnění a zlevnění výroby je vhodné procesy zachytit do byznys modelů. Ty zachycují abstraktní reprezentaci procesu a umožňují jeho další zpracování, například simulace. Zachycení byznys procesů nám umožňuje disciplína byznys modelování, která se zabývá různými modely, kterými lze proces zachytit.

Práce začíná popisem neformálních modelů, které umožňují zachycení procesů pomocí tabulky, nebo slovním popisem. Postupně provází přes semiformální modely jako je IDEF0, EPC, BPMN, nebo aktivitní diagramy. Na závěr druhé kapitoly jsou uvedeny formální modely, které umožňují zachytit a popsat proces formalismy tak, aby byl takový model použitelný i pro další účely jako je třeba simulace.

Třetí kapitola se věnuje popisu modelování metodou BPM, která patří mezi formální. Metoda BPM umožňuje v procesu zachytit více informací a je používána dále v textu práce.

Čtvrtá kapitola popisuje problematiku simulací procesů, jejich validaci, verifikaci a zabývá se také dalšími vlastnostmi, které je nutné přidat do modelů, aby právě v simulacích odrážely, pokud možno, co nejvíce reálné chování procesu. Protože v modelu BPM nejsou tyto informace zobrazeny, řeší se v této kapitole i způsoby jejich zobrazování v simulačním náhledu.

Protože koordinační model metody BPM je postaven na principech petriho sítí, je v páté kapitole popsán převod koordinačního modelu na petriho síť. Nevýhodou takových modelů je jejich složitost, a proto se tato kapitola dále zabývá převodem na jednodušší model - aktivitní diagram.

V poslední kapitole je popsán jeden z podnikových procesů pomocí koordinačního modelu BPM. Je ukázáno, jak může být tento model v praxi složitý a proto je zachycen jeho zjednodušený náhled a je také zobrazen náhled s informacemi pro simulace.

## 2 Metody pro byznys modelování

Každý byznys proces realizuje svým prováděním nějaký cíl[1]. Například cíl procesu montáže automobilu je vyrobený automobil. Výroba automobilu se ale skládá z mnoha kroků a aby mohl být vyroben tentýž automobil znova, je nutné jednotlivé pořadí kroků dodržet. To může být ale problém, protože na výrobě se podílí spousta lidí a zdrojů, které je nutné organizovat. U velkých procesů tak můžeme snadno ztratit přehled o jejich průběhu, nebo může být problém do procesu zasáhnout a zefektivnit jej. Proto se byznys procesy zachycují do modelů, které jsou popsány metodami. Modelů pro zachycení procesů existuje celá řada. Metody se rozdělují se na neformální metody, které nejsou ničím vázané, semiformální metody, ve kterých mají symboly definovaný význam a formální metody, které krom syntaxe mají definovanou i sémantiku.

### 2.1 Neformální metody

Neformální metody se často používají pro popis byznys procesu lidem, kteří nerozumí semiformálním a formálním metodám. Tyto metody tak slouží k jednoduchému zobrazení byznys procesu často formou tabulek, nebo textových popisů a nedrží se žádných pravidel, ať už syntaktických, nebo sémantických.

Kdo	Co	Prostředky	Výsledek
Karel	Ohýbá plechy	Nastříhané plechy podle délky	Ohnuté plechy
Pepa	Svařuje plechy	Ohnuté plechy z předchozí operace	Svařené plechy
Lakovna	Lakuje polotovary	Svařené polotovary z předchozí operace	Hotové polotovary
Montáž	Montuje výbavu	Hotové polotovary a požadovaná výbava	Hotový výrobek
Obchodní oddělení	Prodává výrobek	Hotový výrobek z předchozí operace a zákazník	Prodaný výrobek
Zákazník	Zaplatí za výrobek		

Obrázek 1: Byznys proces zachycený tabulkou

Na obrázku 1 je zachycený byznys proces pomocí tabulky. V tabulce je zachyceno, kdo je za danou aktivitu zodpovědný, jaké jsou požadované vstupy a výstupy aktivity. Celý proces je ukončen po zaplacení zákazníkem.

Jak je vidět, modelování procesu takovým způsobem není vhodné pro další použití. Na druhou stranu je to velmi efektivní metoda, jak zjistit jaké aktivity a zodpovědnosti za ně v procesu vystupují.

### 2.2 Semiformální metody

Semiformální metody mají definovanou syntaxi, ale nemají definovanou sémantiku. Z toho vyplývá, že diagramy těchto modelů se sestávají z definovaných obrazců. Protože nemají definovanou sémantiku, je možné těmito metodami stejný proces zachytit různými diagramy. Tyto metody jsou většinou spravovány různými organizacemi, které se věnují správě jejich syntaxe, rozšiřování pro různé použití a podobně. Často se pomocí těchto



metod zachycuje tok procesů a podprocesů, případně je možné do diagramu zachytit i zodpovědnost za části, nebo celý proces.

Mezi nejčastější modelovací metody patří následující.

### 2.2.1 IDEF0

Je grafická modelovací metoda, která umožňuje hierarchicky zaznamenávat funkčnost procesu. Protože v podniku může být a často je i více procesů, tato metoda se zabývá otázkou, jaké funkce podniku budeme popisovat[1]. Využívá následující prvky:

- Funkce - jednotlivé aktivity, ze kterých se proces skládá
- Vstupy - jsou data, která jsou pomocí funkcí transformována na výstup
- Výstupy - jsou data, která jsou produkována ze vstupních dat
- Řízení - pravidla nutné pro transformaci vstupních dat na výstupní
- Mechanismy - prostředky pro průběh funkce

Na obrázku 2 je znázorněna realizace nějaké zakázky, která se skládá ze čtyř funkcí. Každá funkce má nějaký vstup a výstup. Například ve funkci Objednání zakázky vstupuje zákazník se svými požadavky a výstupem funkce je nějaký zakázkový list - smlouva o zakázce. Funkce by také měla mít pravidla pro transformaci vstupu na výstup a prostředky pro realizaci přeměny vstupních dat na výstupní. Ve funkci Realizace zakázky tvoří pravidla seznam úkonů, které se musí provést pro zdárnou realizaci zakázky. Prostředky pro realizaci tvoří konstrukční tým. Funkce mohou také zahrnovat další podprocesy a takto zachycené diagramy pak tvoří hierarchie.

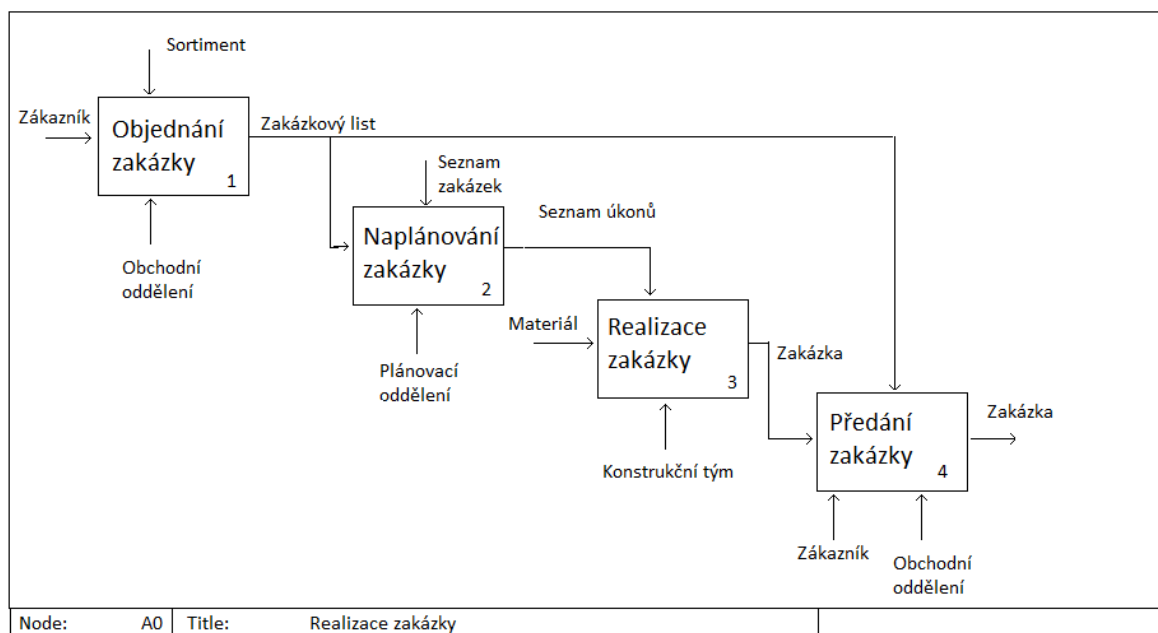
Problémem IDEF0 je, že nezachycuje proces tak, jak běží v čase a zachycuje pouze funkce procesu. Pro komplexní popis procesu je nutná znalost dalších úrovní - IDEF1, IDEF2, IDEF3[1]. Také je občas složité rozhodnout, zda by mechanismus nemohl být řízení a naopak.

### 2.2.2 EPC

Tato metoda je velmi rozšířená, protože je používána jako součást systému SAP a ARIS. Metoda sleduje aktivity procesu z časového pohledu. Každá aktivita je mezi dvěma událostmi, které definují její počátek a konec.[1] Každý proces modelovaný EPC se skládá z:

- Aktivita - je základním prvkem každého byznys procesu a je dílčí částí plnění byznys cíl procesu
- Událost - je vytvářena aktivitou a vytváří vstupní podmínku další aktivity
- Konektory - spojují aktivity a události a vytváří tak tok procesu

1. AND - slouží pro zachycení paralelních aktivit



Obrázek 2: IDEF0 - Diagram realizace zakázky

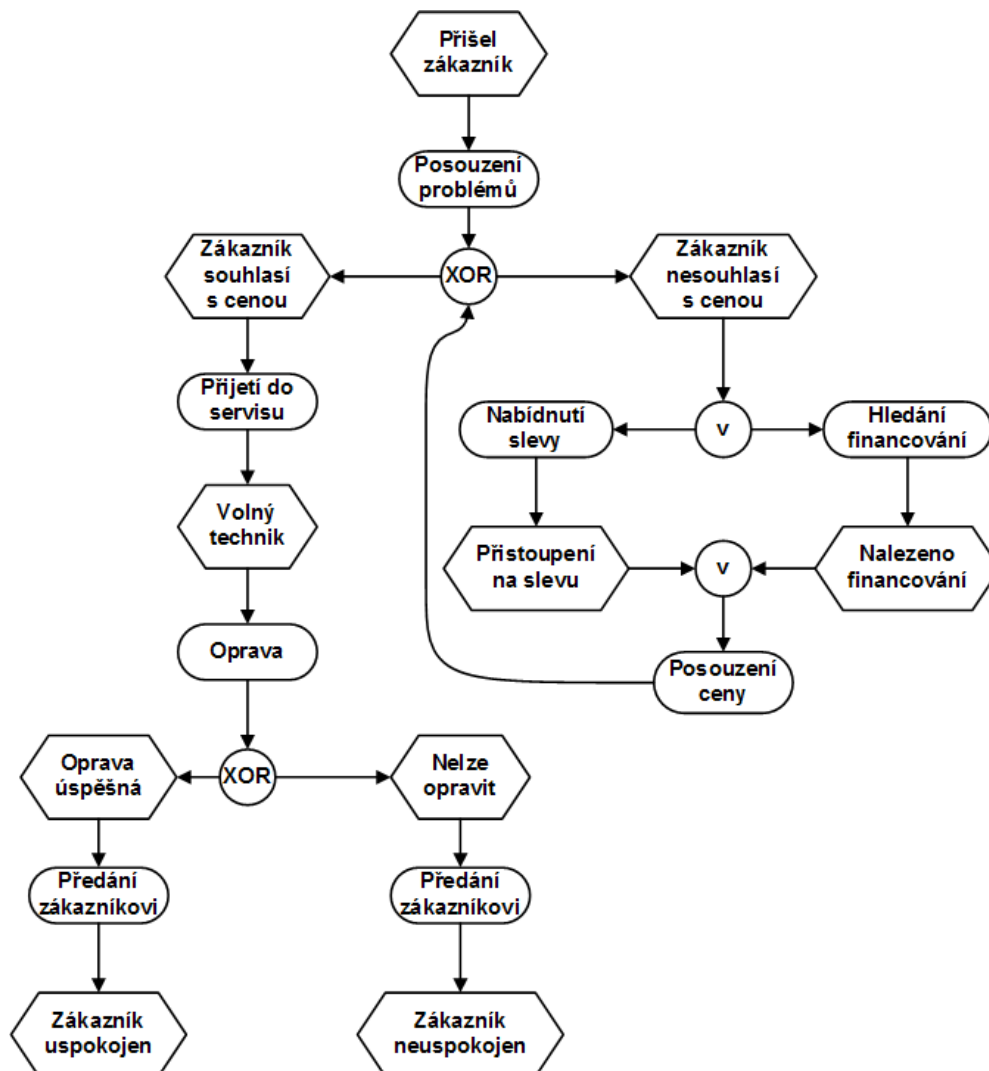
## 2. XOR - umožňují zachytit alternativní scénáře.

Konektory je možné rozdělit na konektory typu split a join. Konektor typu split slouží k rozvětvení procesu. AND split zachycuje počátek paralelního běhu aktivity. AND join naopak zachycuje ukončení paralelního běhu. Konektorem XOR split začíná rozvětvení procesu na alternativní scénáře. Tento typ konektoru je výlučný, to znamená, že není možné provést více alternativ současně. Zobecněním tohoto konektoru je konektor OR, který umožní i více alternativních toků.[1]

Obrázek 3 zachycuje byznys proces opravy automobilu modelovaný metodou EPC, kdy proces začíná příchodem zákazníka s požadavkem na opravu automobilu. Příjemčí technik provede posouzení problému a stanoví předběžnou cenu opravy. S tou buď zákazník nesouhlasí, pak odchází a nebo souhlasí. V tom případě se automobil přijme do servisu a pokud je volný technik, začne se provádět oprava. Oprava je buď úspěšná, pak zákazník odchází uspokojen, nebo je neúspěšná, potom odchází nespokojen.

### 2.2.3 Aktivitní diagramy

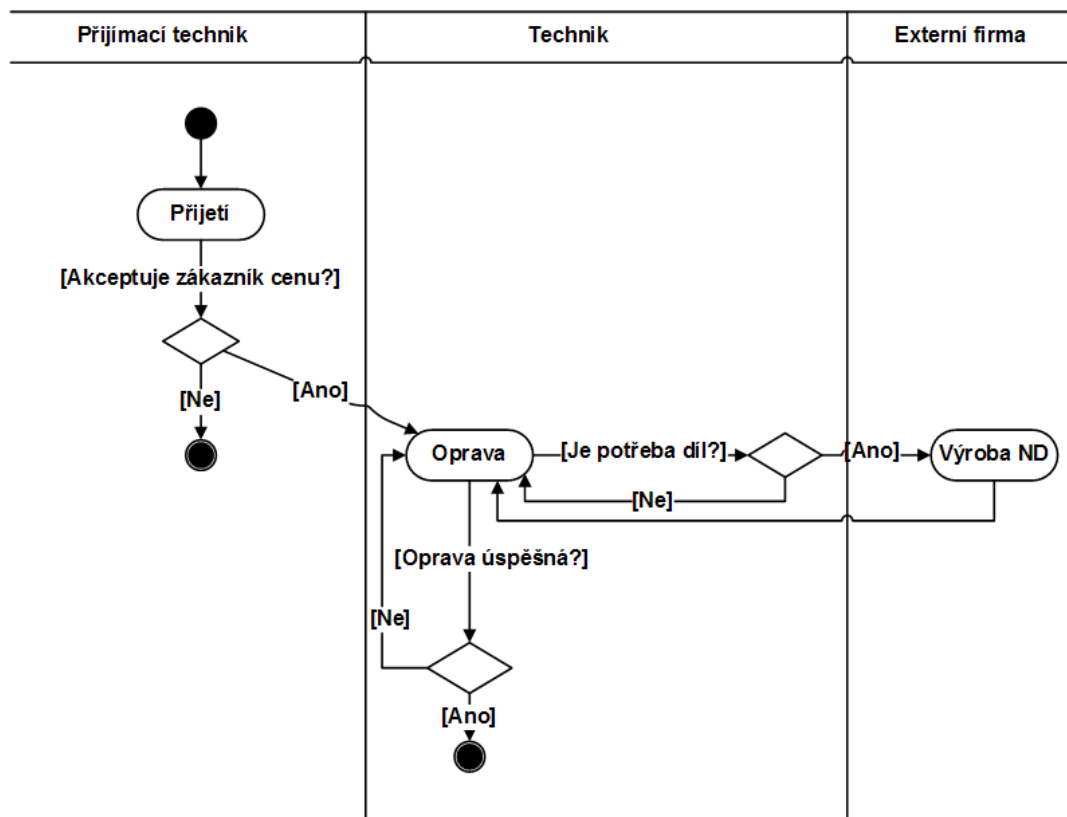
Pro modelování byznys procesů je také možné využít diagramy z balíku UML - objektový, use case, nebo aktivitní diagram. Aktivitní diagram popisuje proces v toku jeho činností, úkolů a podprocesů od startovacího symbolu po ukončovací symbol. Proces může obsahovat více startovacích i ukončovacích symbolů. Může tedy začínat i končit libovolnou aktivitou. Dále v tomto modelu existují rozhodovací bloky, které umožní zachytit alternativní scénáře, případně cykly. Paralelní provádění aktivit lze zachytit syn-



Obrázek 3: Byznys proces zachycený modelem EPC

chronizačním blokem, ve kterém leží paralelně zpracovávané aktivity. Celý model je možné obohatit o plavecké dráhy, ve kterých leží aktivity. Každá dráha má ve svém záhlaví uvedeno, kdo je za aktivity v ní ležící zodpovědný.[1]

Na obrázku 4 je zachycený proces opravy automobilu, který začíná přijetím automobilu. V této aktivitě je zákazníkovi navrhována cena opravy a pokud ji zákazník neakceptuje, proces se ukončí. V případě, že ji akceptuje, je spuštěna aktivita oprava. Je-li během opravy potřeba náhradního dílu, spustí se aktivita výroba ND a po té se opět spustí oprava. Pokud není oprava úspěšná, je znova spuštěna aktivita oprava. V případě úspěšné opravy proces končí.



Obrázek 4: Byznys proces zachycený aktivním diagramem

Výhodou takového modelu je přehlednost a jednoduchá čtivost, se kterou nemají problém ani lidé, kteří se s modelem setkali poprvé. Nevýhodou je, že nelze zachytit všechny náležitosti procesu jako jsou různé spouštěče a podobně.

## 2.2.4 BPMN

Tato metoda je velmi podobná aktivním diagramům. Proces se popisuje stejnými symboly jako v aktivních diagramech, ale používá i další symboly, které model rozšiřují. Stejně jako v aktivních diagramech, i v tomto modelu začíná proces startovacím symbolem, který se nazývá start event, koncový symbol end event. Startovacích, nebo ukončovacích symbolů může být opět i více. Mimo startovacího symbolu se využívá i tzv. intermediate event. Ten umožňuje spustit aktivitu na základě vnější události jako jsou například čas, vnější zpráva, chyba v procesu a spousta dalších. Proces také může začít na základě vnější události, případně může na svém konci, nebo při použití intermediate eventu sám vyvolat událost, na kterou mohou reagovat další aktivity, nebo procesy.[7]

U modelu BPMN je možné zachytit v rozhodovacím bloku (tzv. gateway) alternativní scénáře. Rozhodovacích bloků je několik druhů. U rozhodovacího bloku exclusive může být vybrán právě jeden jediný scénář. Výběr se může provést buď na základě splnění nějaké podmínky, nebo na základě nějaké události - přijetí intermediate eventu. Rozhodovací blok typu inclusive umožňuje vybrat více scénářů, které splňují podmínku. Paralelní zpracování aktivit se u BPMN zachycuje také typem rozhodovacího bloku.[7]

Stejně jako u aktivních diagramů, i v modelu BPMN je možné využívat plavecké dráhy. BPMN definuje pool, který plavecké dráhy obsahuje. Pool přitom nemusí obsahovat aktivity a navenek se může chovat jako black box. BPMN nespécifikuje použití plaveckých drah, ale často je stejné jako u aktivních diagramů - zachycení zodpovědnosti za aktivity. Jednotlivé plavecké dráhy se do sebe mohou zanořovat a obsahovat další dráhy.[7]

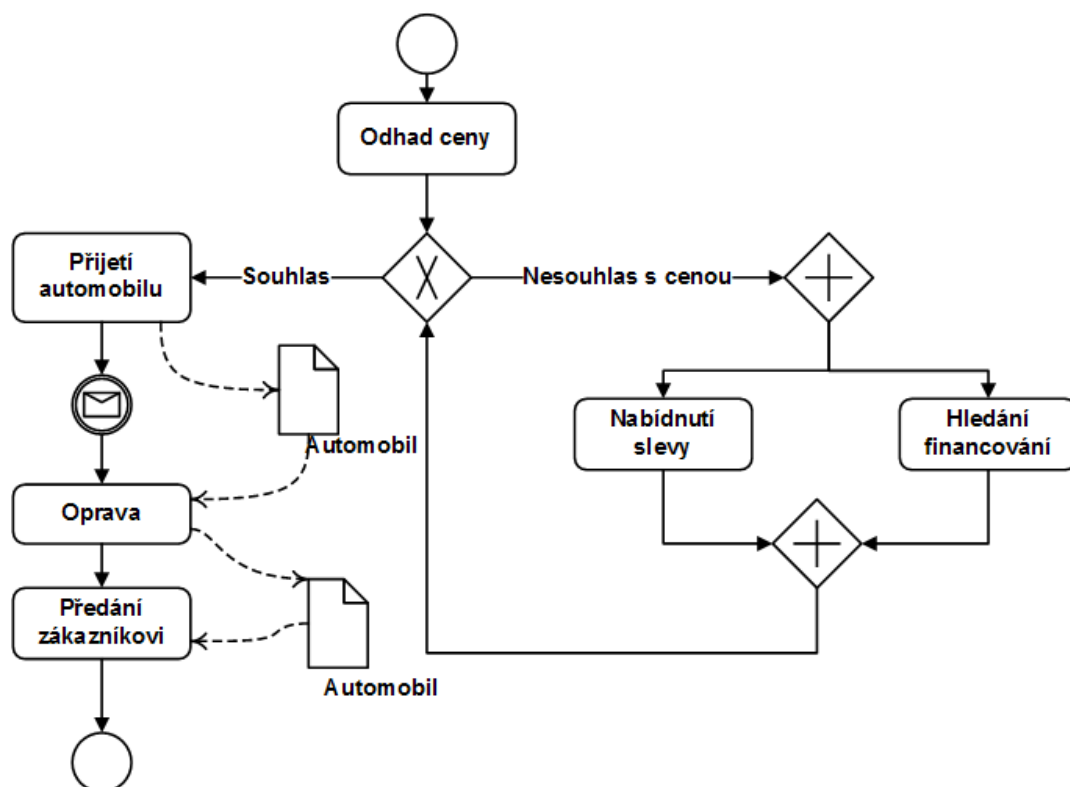
BPMN obsahuje ještě pomocné objekty, tzv. artefakty. Jsou to datové objekty, které jsou asociovány se spojením mezi aktivitami a popisují jak se dané objekty používají a mění v průběhu procesu. Krom toho model obsahuje textové anotace, které zlepšují jeho čtení.[7]

BPMN umožňuje zachytit proces tak, jak je ho možné doopravdy vidět, to znamená, zachytit jej s alternativními scénáři, spouštěči a objekty, které se v procesu pohybují. Tento pohled se může zdát jako výhodný, ale zároveň v sobě skrývá i nevýhody složitějšího čtení a ne všem jsou symboly použité v tomto modelu na první pohled srozumitelné.

Obrázek 5 znázorňuje stejný proces jako model na obrázku 3. U tohoto modelu je vidět, že aktivita přijetí generuje zprávu, na kterou reaguje aktivita oprava. Taktéž je vidět artefakt automobil, který je předáván z aktivity přijetí do aktivity oprava a z aktivity oprava do aktivity předání zákazníkovi.

## 2.3 Formální metody

Tyto metody jsou matematicky popsány a mají přesně danou syntaxi a sémantiku. Často jsou definovány matematickými modely. Tak je možné procesy modelovat striktně. Tím se eliminuje možnost jeden proces vyjádřit více diagramy. Tato výhoda je ale vykoupená vyšší složitostí diagramů a proto může být jejich pochopení například byznys manažery komplikované. Na druhou stranu nám ale tyto metody přinášejí možnost procesy otestovat a ověřit na nepříznivé stavy a situace a případně mohou pomoci proces zefektivnit.



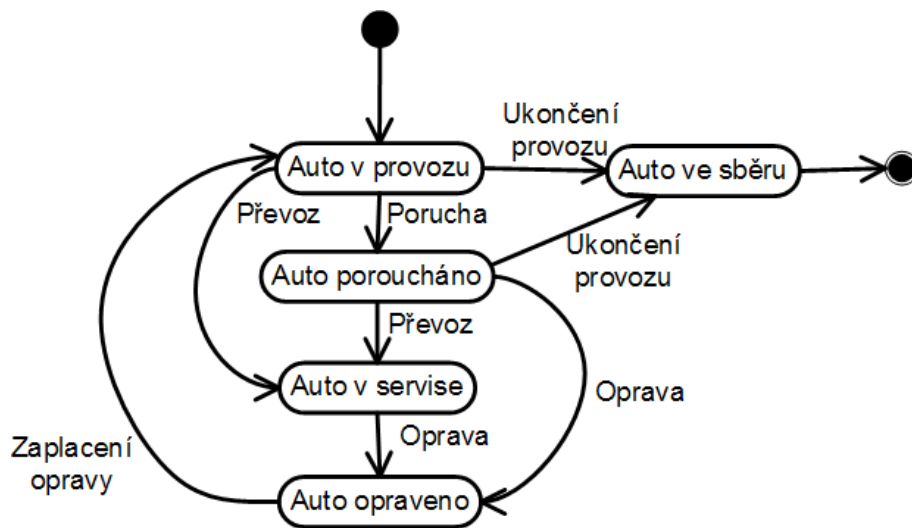
Obrázek 5: Byznys proces zachycený modelem BPMN

### 2.3.1 Konečné automaty

Byznys proces je možné zachytit i pomocí konečných automatů. Konečný automat je uspořádaná pětice  $KA = (Q, I, \delta, q_0, F)$ , kde

- $Q$  je konečná množina stavů, které reprezentují aktivity procesu
- $I$  je konečná neprázdná množina vstupů,
- $\delta : Q \times I \rightarrow Q$  je přechodová funkce. Ta znázorňuje možnosti, do kterých se může proces dostat z jedné aktivity do druhé.
- $q_0 \in Q$  je počáteční stav, který označuje začátek procesu
- a  $F \subseteq Q$  je množina koncových (přijímajících) stavů, ve kterých proces končí

Konečný automat je možné znázornit pomocí stavového diagramu jak je vidět na obrázku 6, ve kterém jednotlivé stavy reprezentují stavy automatu a hrany jeho přechody.[1] Pomocí přechodových funkcí je definováno do jakého stavu se může proces dostat. Mezi nevýhody stavových automatů můžeme zahrnout obtížné modelování paralelních procesů, protože se množina stavů může rozšířit až na počet  $2^n$  stavů. Tím vzniká



Obrázek 6: Byznys proces zachycený konečným automatem

problém, jak proces vůbec zakreslit. Tento problém odstraňují petriho sítě, které z konečných automatů vychází. Podle obrázku 6 by popis konečného automatu vypadal následovně:

$Q = \{\text{Auto v provozu, Auto poroucháno, Auto v servise, Auto opraveno, Auto ve sběru}\}$

$I = \{\text{Porucha, Převoz, Oprava, Zaplacení opravy, Ukončení provozu}\}$

$\delta(\text{Auto v provozu, Porucha}) = \text{Auto poroucháno}$

$\delta(\text{Auto poroucháno, Převoz}) = \text{Auto v servise}$

$\delta(\text{Auto v servise, Oprava}) = \text{Auto opraveno}$

$\delta(\text{Auto opraveno, Zaplacení opravy}) = \text{Auto v provozu}$

$\delta(\text{Auto v provozu, Převoz}) = \text{Auto v servise}$

$\delta(\text{Auto poroucháno, Oprava}) = \text{Auto opraveno}$

$\delta(\text{Auto poroucháno, Ukončení provozu}) = \text{Auto ve sběru}$

$\delta(\text{Auto v provozu, Ukončení provozu}) = \text{Auto ve sběru}$

$q_0 = \text{Auto v provozu}$

$F = \text{Auto ve sběru}$

Tento model je velice problematický na čtení, nebo zakreslení, pokud se jedná o paralelizmus. Ovšem pro potřeby simulace může být efektivní, neboť jsou propracované metody pro analýzu konečných automatů.

### 2.3.2 Petriho síť

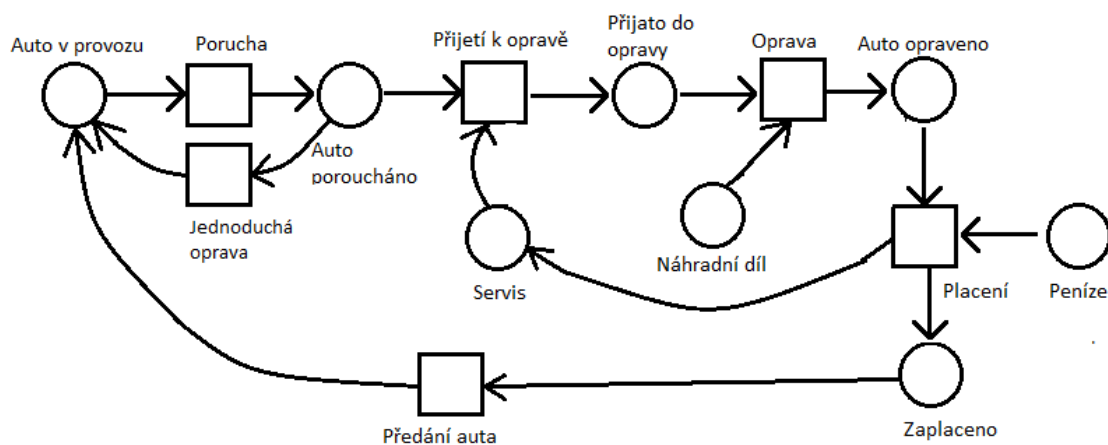
Petriho síť je orientovaný bipartitní graf, který obsahuje místa a přechody. Místa jsou vždy navázány na přechody a tvoří tak vstupní a výstupní místa.[8] Toho se dá využít pro zachycení byznys procesu. Přechody je možné znázornit jako aktivitu procesu a místa jako vstupní, nebo výstupní zdroje aktivity.

Formálně je struktura petriho sítě pětice  $\langle P, T, I, O, H \rangle$ , kde

- P - je množina míst
- T - je množina přechodů
- I - je vstupní funkce
- O - je výstupní funkce
- H - je vstupní inhibiční funkce

Na obrázku 7 je zachycený model procesu provozu automobilu. Během provozu automobilu může dojít k poruše a automobil musí být opraven. Automobil buď musí na opravu do servisu, nebo je možné provést opravu mimo servis, například výměnu žárovky. V servise se automobil opraví za pomoci náhradního dílu a následně se oprava zaplatí pomocí peněz. Ve chvíli, kdy je automobil zaplacen, dojde k předání zákazníkovi a auto je opět v provozu.

Petriho sítě jsou poměrně jednoduché na čtení i zachycování procesů. Avšak v procesech, které využívají mnoho zdrojů se tento model může stát snadno nepřehledný. Pro simulace je opět velice výhodný, protože v něm snadno lze zachytit paralelní zpracování aktivit a jsou rovněž jako u konečných automatů propracovány metody pro analýzu.



Obrázek 7: Petriho síť



### 3 Business process modeling

BPM je metoda pro modelování byznys procesů, která se skládá ze tří nezávislých pohledů.[2]

Funkční model

Objektový model

Koordinační model

V reálných byznys procesech se vyskytují zdroje, které mají, často i musí mít, požadované vlastnosti. Zachytit takové požadavky pomocí jiných diagramů byznys modelování může být proto problematické. BPM umožňuje zachytit procesy a jejich zdroje jako objekty. Je tak jednodušší popsat reálný proces se všemi náležitostmi. Navíc je možné definovat procesům dobu trvání a náklady. Tím lze co nejlépe přiblížit simulační model reálné situaci.

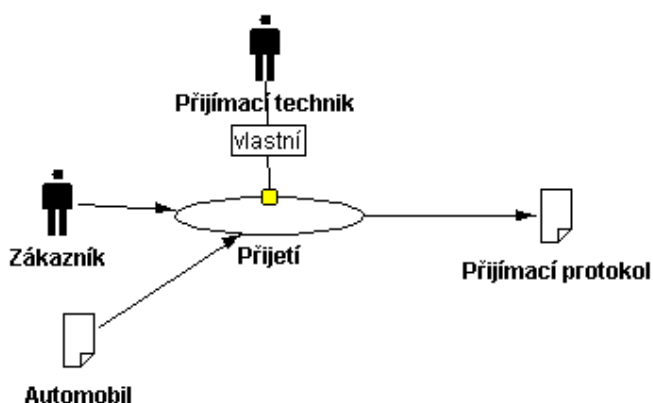
BPM rozlišuje ve všech modelech dva druhy objektů - aktivní a pasivní. Aktivní objekt zpracovává často pasivní objekt a realizuje průběh procesu. Zatímco pasivní objekty jsou v průběhu procesu přeměňovány, vytvářeny, spotřebovávány, nebo jinak manipulovány aktivním objektem.[1]

#### 3.1 Funkční model

Tento model zachycuje všechny možné procesy, jejich uživatele, vstupní a výstupní objekty a jejich vzájemné vazby. To je důležité pro identifikaci podnikových procesů a jejich architektury v organizaci. Mezi procesy mohou existovat dvě vazby (relace)[2]

Obsažení (include) - tato relace značí existenci podprocesu. V průběhu procesu je těmto podprocesům předáno řízení a po jejich vykonání se opět vrací nadprocesu.

Spolupráce (collaboration) - tato relace umožňuje realizovat paralelní procesy

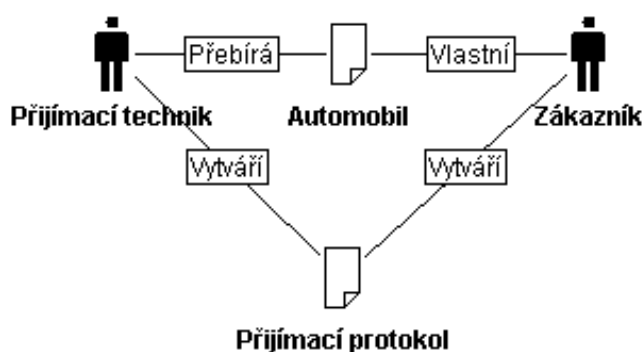


Obrázek 8: Funkční model BPM

Na obrázku 8 je znázorněný proces přijetí automobilu do opravy. Z obrázku je patrné, že do procesu, který vlastní aktivní objekt přijímací technik, vstupuje aktivní objekt zákazník, který požaduje opravu auta. Do procesu taktéž vstupuje pasivní objekt automobil, který je nutný k vytvoření přijímacího protokolu, který je výsledkem procesu.

### 3.2 Objektový model

Tento model zachycuje vazby mezi všemi objekty vyskytujícími se v procesu. Je tak velmi podobný diagramu tříd z jazyka UML. Pro vytvoření tohoto modelu je nutné znát všechny objekty, které se v procesu vyskytují. Z toho vyplývá, že tímto modelem je popsáno jakými objekty je proces realizován. V tomto modelu, stejně jako v diagramu tříd jazyka UML, je možné definovat násobnost objektů. Objektový model by měl být zvláště pro každý proces ve funkčním modelu. [2]



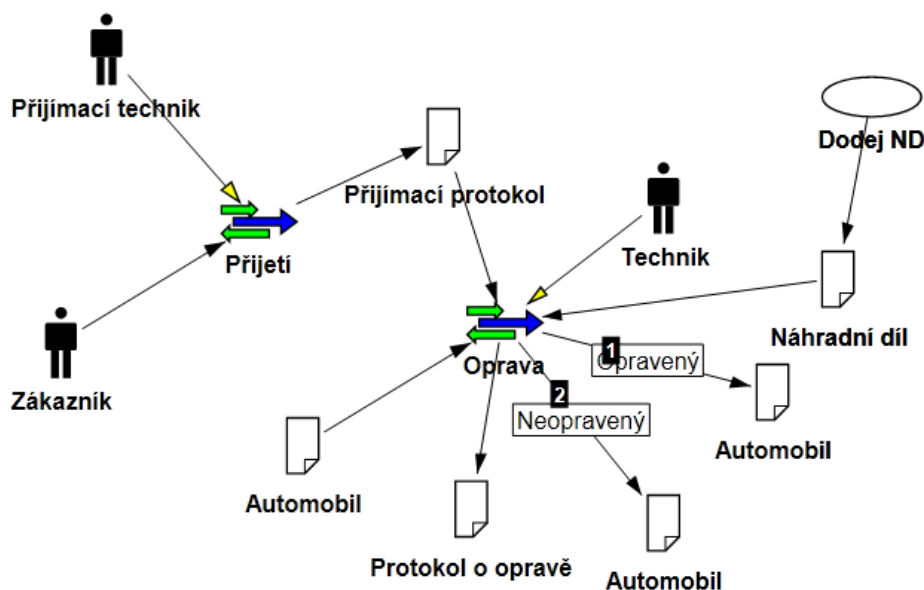
Obrázek 9: Objektový model BPM

Na obrázku 9 jsou zobrazeny objekty, které se podílejí na procesu přijetí z funkčního modelu zachyném na obrázku 8. Na modelu jsou zachyceny dva aktivní objekty, které společně vytvářejí pasivní objekt.

### 3.3 Koordinační model

Koordinační model zachytává tok byznys procesu. Aktivitu je možné provést pouze pokud jsou dostupné všechny její vstupní zdroje.[2] Tato vlastnost je i v modelu petriho sítí, kdy je přechod možné provést pouze pokud jsou ve vstupních místech tokeny. Tento model tak zachycuje předávání objektů mezi aktivy, kdy cílem je provedení byznys procesu. Právě tento model nejvíce využívá objektových vlastností přidanych do BPM a umožňuje tak vytvářet simulace podobné reálným byznys procesům.

Na obrázku 10 je zobrazen koordinační model BPM, který znázorňuje tok byznys procesu opravy auta. Tok začíná, pokud jsou dostupné objekty zákazník a automobil. Poté aktivita přijetí vytvoří pasivní objekt přijímací protokol, který vstupuje společně s objektem automobil do procesu oprava. V aktivitě oprava se mohou podle obrázku odehrát



Obrázek 10: Koordinační model BPM

dva scénáře. Buď se automobil podaří opravit, pak proběhne scénář jedna, aktivita vygeneruje objekt automobil, který je opravený, nebo se nepodaří opravu provést a aktivita vygeneruje jako objekt neopravený automobil. V modelu je také zachycen podproces dodej náhradních dílů, který pro potřebu opravy vytváří pasivní objekt Náhradní díl.

### 3.4 BP studio

Aby bylo možné koordinační model vyhodnocovat na své vlastnosti, určitě není vhodné zachytit model na papír, ale je lepší zachytit jej v nějakém programu. Speciálně pro BPM byl vytvořen program BP studio, který umí zachytit všechny tři modely BPM. Navíc umožňuje i simulace procesu. Protože se v byznys procesech pracuje s objekty a BPM umí rozlišit mezi pasivními a aktivními objekty, je možné v BP studiu zachytit ke každému objektu jeho vlastnosti. Dokonce i aktivity se chovají jako objekty a mají své vlastnosti jako je například interval trvání nebo náklady na provedení aktivity. V simulačním náhledu je možné provádět krokování modelu, sledovat pohyb tokenů skrz model a pozorovat chování procesu.

## 4 Simulační náhled

Simulace byznys procesů spočívá ve vytvoření simulačního modelu, který reflektuje chování procesu včetně zdrojů a umožňuje tak lépe pochopit běh procesu. Simulací procesu můžeme sledovat různé atributy procesu, jako minimální, průměrné a maximální časy trvání procesu, spotřebované zdroje (80% zaměstnanců je vytíženo).

Z pohledu simulací je snadnější model modelovat metodou založenou na matematických přístupech jako jsou petriho sítě, pi-kalkul, nebo konečné automaty.[3] Takové modely lze pak snadno převést do podoby počítačového programu.

Pro potřeby simulací byznys procesů je nutné vnést do nich reálné požadavky, zejména čas pro zpracování aktivity. Ten ale může být ovlivněn jak vnějšími, tak vnitřními faktory. Některé aktivity mohou čekat na uvolnění potřebných zdrojů. V procesu také může existovat spousta alternativních scénářů, které jsou v realitě jistě vykonány a je potřeba tuto skutečnost do simulace zanést, například využitím pravděpodobnosti.[3] Protože je koordinační model založený na petriho sítích - je defakto jejich rozšířením - jeví se jako vhodná možnost převést tento model na petriho síť.

### Validace a verifikace

Při návrhu modelu složitějšího byznys procesu je snadné vnést do modelu chybu, díky které nemusí model reflektovat skutečnost a je vhodné model na tuto skutečnost otestovat. Takové testování se nazývá validace. Ověření modelu proti skutečnosti je lepší nechat na člověku, který zná doménu modelovaného procesu. Pro validaci je také dobré použít simulaci procesu, kde je možné sledovat spotřebovávání a vytváření zdrojů a vykonávání aktivit.

Ověření, že model neobsahuje nevhodné konstrukce, které by mohly způsobit například uzamčení procesu, nebo neočekávané hromadění zdrojů, se nazývá verifikace. Vzhledem k matematickým popisům je možné verifikaci modelu provést automatizovaně pomocí výpočetní techniky.

V reálných procesech se setkáváme s omezeními, které je taktéž nutné v simulacích zohlednit. Na příkladu montáže automobilů je zřejmé, že pracovníci pracující na první aktivitě procesu stále znova začínají tentýž proces. Tím vytvářejí novou instanci procesu a v konečném důsledku vedle sebe existuje několik instancí téhož procesu. V každém reálném procesu existují pouze omezené zdroje - prostředky pro provádění procesu. Všechna taková omezení je nutné v simulačním náhledu zohlednit.

### 4.1 Simulace modelu zachyceného petriho sítí

Pětici  $\langle P, T, I, O, H \rangle$  můžeme rozšířit o počáteční značení  $M_0$ . Toto značení je zobrazení z množiny míst na množinu přirozených čísel ( $P \rightarrow N$ ), které přiděluje každému místu přirozené číslo, které znamená počet tokenů v místě. Rozšířením o počáteční značení dostaneme systém petriho sítě.[8] Aby bylo možné provést simulaci procesu, je nutné definovat dynamiku petriho sítě, která se skládá z:

- Pravidla pro proveditelnost přechodu
- Pravidla pro změnu stavu po provedení přechodu

Pro další popis dynamiky petriho sítě je nutné zdefinovat následující pojmy[8]

$\bullet t = \{p \in P : I(t, p) > 0\}$  - množina vstupních míst přechodu  $t$

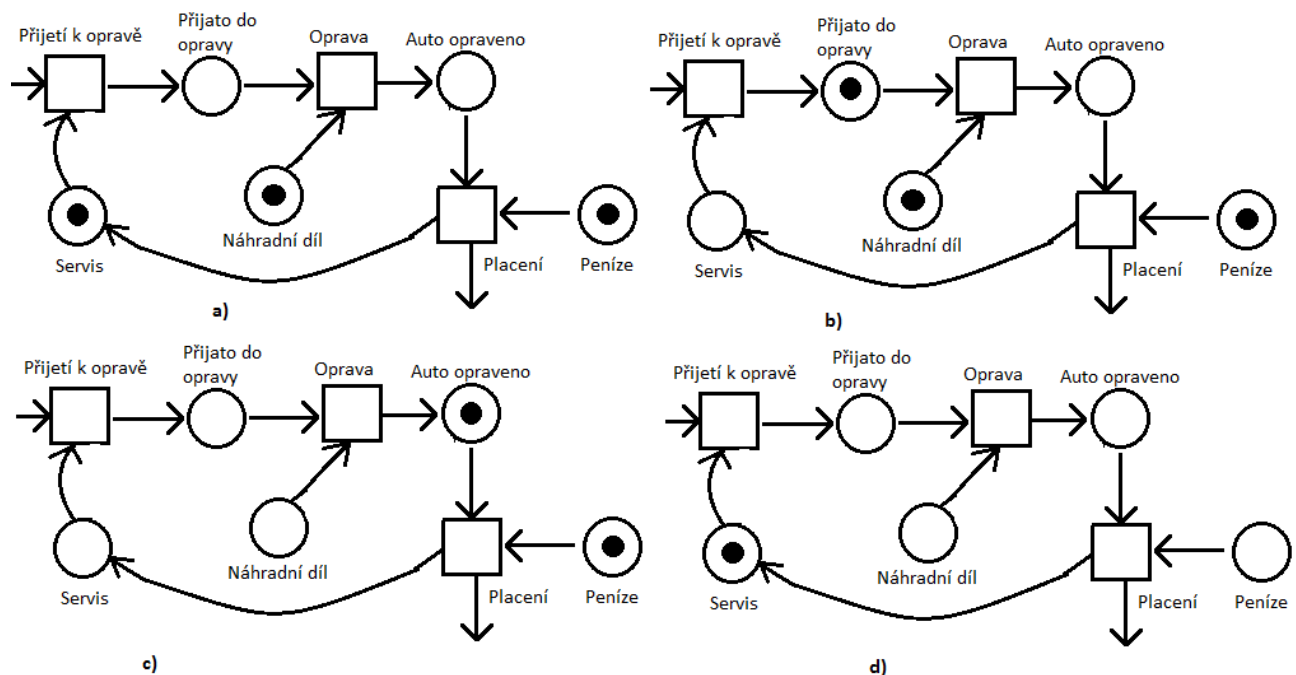
$t^\bullet = \{p \in P : O(t, p) > 0\}$  - množina výstupních míst přechodu  $t$

Jak bylo zmíněno v kapitole o petriho sítích, každý přechod znamená aktivitu a místo nějaký zdroj. Aby bylo možné aktivitu provést, je potřeba mít všechny zdroje, které aktivita využívá. Zda je zdroj dostupný je zachyceno právě značením sítě - v místě, které představuje zdroj aktivity je alespoň takový počet tokenů, kolik je násobnost hrany vedoucí z tohoto místa do přechodu. Takto lze jednoduše říct, kdy je přechod proveditelný, respektive je proveditelná aktivita. Formálně je to možné zapsat takto:

$$(\forall p \in \bullet t)[M(p) \geq I(t, p)] [8]$$

Po provedení přechodu, respektive aktivity dojde k vyprodukování nových zdrojů, které jsou vstupní pro další aktivitu a síť se ocitne v jiném stavu. Například před provedením aktivity oprava byla síť ve stavu před opravou a po provedení této aktivity je ve stavu opraveno. Provedením přechodu se tedy spotřebovaly tokeny na vstupních místech a vyprodukovaly se na výstupních, jinak řečeno, změnilo se značení sítě.

Tedy provedením přechodu, který je proveditelný při značení  $M$ , dostaneme značení  $M'$ , takové, že  $M' = M + O(t) - I(t)$ . [8] Obrázek 11 zobrazuje ve čtyřech krocích simu-



Obrázek 11: Simulace modelu zachycená petriho sítí

laci části modelu byznys procesu provozu automobilu. Na obrázku a) je automobil před přijetím do opravy. Provedením přechodu Přijetí k opravě se spotřebojuje token v místě

Servis a vygeneruje se token v místě Přijato do opravy, jak zachycuje obrázek b). To, že není token v místě Servis znamená, že v této chvíli není možné přijmout další automobil do opravy. Provedením přechodu Oprava se opět změní stav procesu, ve kterém již je automobil opraven a přitom se spotřeboval token v místě Náhradní díl a Přijato do opravy. Na poslední části obrázku je zachycen stav procesu po zaplacení. Přitom dojde k vygenerování tokenu do místa Servis.

Pokud by po provedení těchto čtyř kroků přijel další automobil, bude přijat do opravy, ale dál se síť nedostane, protože v místě Náhradní díl není token. Takovému stavu síť se říká uzamčení.

#### Instance procesu

Zachytíme-li byznys proces strukturou petriho sítě, neobsahuje takový model žádné tokeny. Je tak možné si takový model přestavit jako předpis pro instanci. Doplněním značení, často počátečního  $M_0$ , vytvoříme konkrétní instanci procesu a lze nad ní spustit simulaci. Instance tedy je konkrétní systém petriho sítě s daným značením  $M$ .

## 4.2 Trvání aktivity

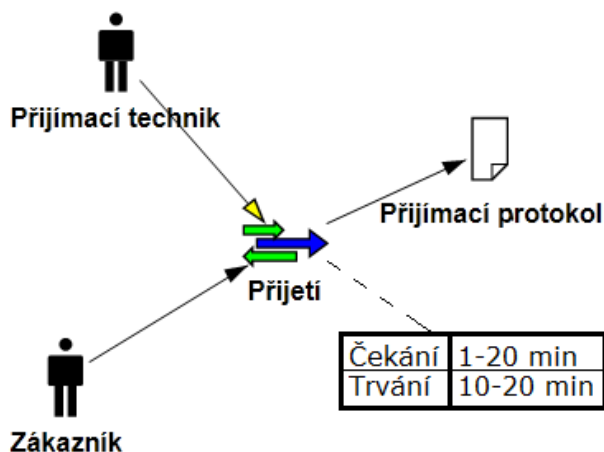
V reálném procesu trvá každá aktivita nějaký čas. Ten je ale velmi složité definovat, protože aktivita nemusí pokaždé trvat stejnou dobu a často může záviset na spoustě dalších faktorů. Například v aktivitě oprava z obrázku 10 může být za opravu považována výměna žárovky, která proběhne za pět minut, ale také se může být oprava složitější - například oprava motoru, která pak trvá i dny. Pokud by aktivita trvala pokaždé stejně, je možné čas trvání aktivity označit konstantou. V opačném případě je doba trvání jako náhodná veličina s daným pravděpodobnostním rozdělením. Protože je časový interval zadán jako minimální a maximální hodnota, jeví se jako vhodné využít normální rozdělení pravděpodobnosti, kde je nejvíce hodnot umístěno kolem střední hodnoty.[3]

BP studio umožňuje u každé aktivity zapsat interval jejího trvání. Avšak tento interval se je vidět pouze ve vlastnostech aktivity a není zobrazen během provádění simulace. Je tak vhodné simulční model rozšířit o modelovací prvek, který umožní tento interval zobrazit. Jak tedy zachytit vlastnosti aktivity jednoduchým a přehledným způsobem? Ideální pro takové zobrazení je tabulka. V tomto případě tabulka o dvou sloupcích. Levý sloupec obsahuje název vlastnosti a pravý hodnotu. Mezi tabulkou a aktivitou je potřeba vytvořit asociaci. Ta je v modelu značena přerušovanou čarou. Tabulka informací je zachycena na obrázku 12. V tabulce jsou informace o trvání a čekání na aktivitu.

## 4.3 Čekání na aktivitu

Čekání na aktivitu je velmi podobné jako trvání aktivity. Rozdíl je ale ten, že čekání na aktivitu nemá dopad na zdroje, které aktivita používá.[3] Vztaženo k časovému trvání, aktivita na začátku vezme zdroje a po definovaném časovém okamžiku vydá výstup. U čekání nejsou zdroje aktivitou spotřebovány. Příkladem by mohlo být provedení aktivity oprava, kdy se zákazník v pondělí dohodne, že se oprava začne ve středu. Auto bude dva dny stát na parkovišti servisu. U čekání na aktivitu se tedy stejně jako u trvání použije časová konstanta pokud čekání trvá vždy stejnou dobu, nebo náhodná veličina s daným

pravděpodobnostním rozdělením. Časový interval čekání se zapíše, stejně jako časový interval trvání, do tabulky.

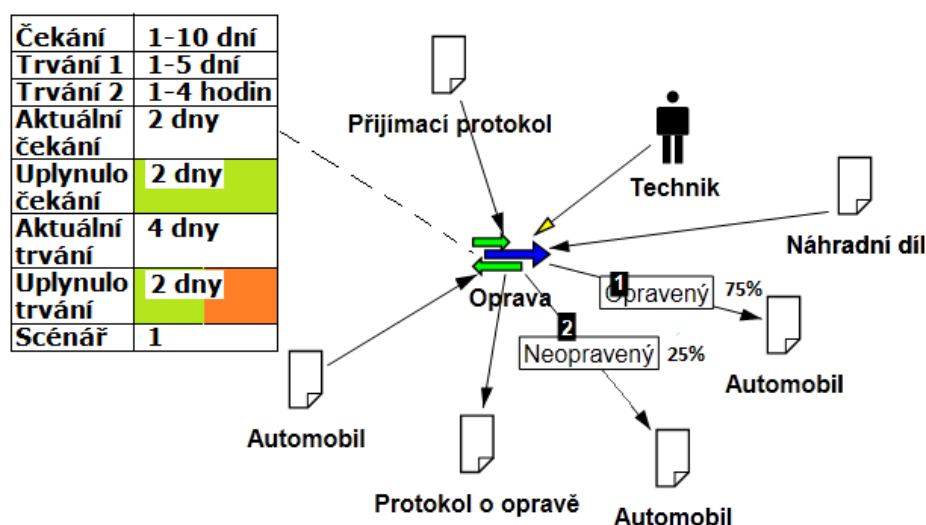


Obrázek 12: Koordinační model rozšířený o čekání a trvání aktivity

#### 4.4 Alternativní scénáře

Asi každá aktivita v reálných byznys procesech má alternativní scénář a nemusí mít pouze jeden, ale často jich má několik a většinou se nevykonávají se stejnou pravděpodobností. Každá aktivita má nějaký základní scénář, který plní část cíle byznys procesu.[3] Například v aktivitě oprava automobilu je to opravení automobilu a vystavení protokolu o opravě. V koordinačním modelu je možné zachytit alternativní scénáře, stačí přidat procentuální vyjádření pravděpodobnosti s jakou scénář nastane. To ovšem není všechno. Alternativní scénář zcela jistě bude mít jinou dobu trvání, než scénář základní a proto je nutné přidat i tuto informaci jak je zobrazeno na obrázku 13. Ten zachycuje model, ve kterém se v 75% případů vykoná základní scénář a ve 25% případů scénář alternativní. Taktéž zachycuje, že pokud nelze automobil opravit bude scénář probíhat jednu až čtyři hodiny, kdežto v případě základního scénáře bude trvání od jednoho do pěti dnů. I v případě alternativních scénářů zapíšeme informace do tabulky. Protože každý scénář může trvat různý časový okamžik, je potřeba do tabulky zapsat informaci o trvání každého scénáře tak, jak je to zobrazeno na obrázku 13. Do tabulky je možné zapsat i informace o pravděpodobnostním určení každého scénáře. Pro jednodušší procesy ale může být tato informace zaznamenána přímo u každého výstupního objektu dané alternativy.

Protože jsou intervaly trvání aktivity, nebo jednotlivých scénářů a čekání na aktivity určeny normálním rozdělením, jsou při spuštění aktivity tyto intervaly přesně určeny. K atributům, které se potom u aktivity zobrazí tak můžeme přidat konkrétní čas, který bude aktivita, nebo scénář trvat a čas, po který již aktivita trvá, případně čas, po který se bude na aktivitu čekat a kolik času už z čekání uplynulo. Další, jistě užitečnou informací může být vybraný konkrétní prováděný scénář.



Obrázek 13: Aktivita koordinačního modelu s informacemi pro simulační náhled

Pro větší přehlednost můžou být atributy v tabulce uspořádány podle klíče, například podle typu informace - statická, nebo dynamická. Statická informace je takový typ informace, který se během simulace procesu nemění. Je to tedy například informace o intervalu čekání, případně o trvání aktivity, nebo jejího alternativního scénáře. Dynamická informace je taková, která se v čase mění. Může to být například vybraný alternativní scénář, uplynulý čas trvání, konkrétně určený čas trvání a spousta dalších informací, které se během simulací generují.

U dynamických informací jako je čekání a trvání lze znázornit aktuální stav simulace pomocí jednoduchého ukazatele přímo v tabulce. Ukazatel bude mít jednoduchou funkčnost. Například u informace o aktuálním intervalu trvání bude mít po spuštění aktivity celá buňka červené pozadí. To se během provádění aktivity postupně změní na zelenou barvu. Jakmile je buňka vyplněna celá zeleně, znamená to, že vybraný časový okamžik již uplynul a provádění aktivity bylo ukončeno. Jestliže aktivita ještě nebyla spuštěna, zůstane pozadí buňky tabulky bílé.

Protože koordinační model a tím i simulační model může být velmi komplikovaný, další přidání informace způsobí, že se model stane ještě víc nečitelnějším. Je proto nutné si rozvážit, jak se budou prvky s informacemi zobrazovat. Protože se simulace budou provádět v programu BP studio, nabízí se možnost zobrazovat modelovací prvek textové notace nějak interaktivně. Pokud není simulace prováděna, nebudou u aktivit zobrazeny žádné informace. Ty bude možné zobrazit například pozastavením kurzoru nad aktivitou. Po krátkém časovém intervalu se informace zobrazí. V případě, že bude spuštěna simulace, ale aktivity nebudou v simulaci aktivní - nejsou zrovna prováděny, taktéž se u nich nezobrazí žádné informace a bude je možné zobrazit opět pozastavením kurzoru nad aktivitou. Mohlo by se zdát lepší vyžádání informací dvojklikem levého tlačítka myši, ale na takovou akci se v BP studiu zobrazí všechny vlastnosti aktivity. U aktivních



aktivit - aktuálně prováděných, se automaticky informace zobrazí ve chvíli, kdy bude mít aktivita dostupné všechny své zdroje.

## 4.5 Zdroje v simulacích

Jak bylo uvedeno v kapitole o BPM, existují zdroje aktivní a pasivní. V každém reálném byznys procesu jsou tyto zdroje využívány pro dosažení cíle procesu. Z reálného pohledu je možné zdroje rozdělit.

Vstupní zdroje - zdroje, které do procesu vstupují z vnějšího prostředí, například zákazníci, nebo materiál

Interní zdroje - jsou takové zdroje, které do procesu nevstupují, ale jsou v něm obsaženy. Takové zdroje často představují zaměstnanci, výrobní stroje, přístroje, nebo výpočetní technika.

Z tohoto rozdělení zdrojů můžeme říct, že vstupní zdroje do procesu vstupují stále znova a vytvářejí nové instance procesu. Protože vstupní zdroje vstupují do vstupních míst procesu, mluvíme o vstupu nových zakázek do procesu. Interní zdroje se naopak pohybují uvnitř procesu, často vstupují do aktivit opakovaně a často také v aktivitě slouží k přeměně vstupního zdroje aktivity na výstupní. Přestože vstupní zdroje vytváří nové instance procesu, jsou interní zdroje sdíleny přes všechny instance. Proto mluvíme o sdílených zdrojích.

### 4.5.1 Vstup nových zakázek do procesu

Je zřejmé, že v reálných podmínkách se procesy spouštějí stále znova s tím, jak přicházejí například zákazníci do obchodu, zákazníci do autoservisu s požadavkem na opravu, nebo dokud leží na skladě polotovary, které je nutné zpracovat. Ne vždy ale přijde zákazník v době, kdy je autoservis, nebo obchod prázdný. Pracovníci na výrobní lince automobilky nečekají až bude celý automobil hotový, ale po provedení své aktivity ji začínají provádět znova. V simulacích je tedy potřeba spustit simulaci několika stejných procesů. Mluvíme tedy o instanci procesu. S příchodem nového vstupního zdroje se spustí nová instance procesu, která běží paralelně vedle jiné instance, spuštěné před touto novou instancí. Jednotlivé instance jsou tak vůči sobě posunuté o časový okamžik, který je určený exponenciálním rozdělením.

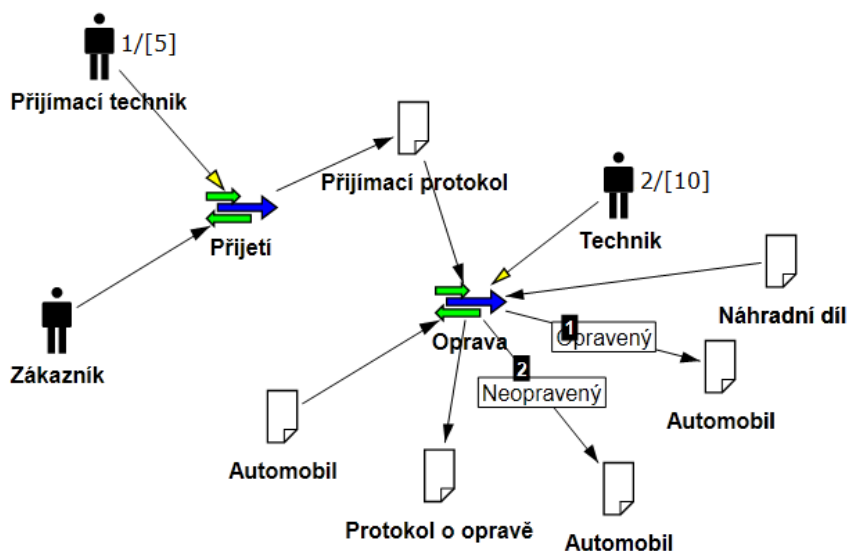
### 4.5.2 Sdílené zdroje

S problémem paralelně provádějících se instancí byznys procesu je potřeba zavést počáteční stav procesu. Ten je daný vstupním místem procesu, tedy místem, do kterého vstupují nové zakázky. A dále je určený vstupními objekty aktivit. Lze si ale představit situaci, že autoservis má určitý počet mechaniků. Jak přichází noví zákazníci s požadavkem na opravu, spouští se nové instance a pro každou instanci je potřeba pro provedení opravy dostupnost mechanika. Je tak snadné si představit, že mechanici tvoří pool zdrojů. Ten

slouží jako úschovna zdrojů, ze kterého si každá instance vezme potřebný počet zdrojů. Pool je tedy sdílený přes všechny instance procesu. Jak již bylo zmíněno, v reálném procesu není možné mít neomezené zdroje. Proto se pool sdílených zdrojů na začátku simulace nastaví na definovanou hodnotu, například počet mechaniků, programátorů, a podobně. Během simulace se pak může stát, že při velkém počtu instancí se vyčerpou všechny zdroje z poolu a proces musí čekat na jejich uvolnění.

Koordináční model BPM zachycuje tok byznys procesu. Nezachycuje však situaci, že proces má omezené zdroje. Vzhledem k tomu, že je vhodné zachytit reálný pohled na proces, je důležité takovou informaci do koordináčního modelu zavést. Během doby, kdy zachycujeme byznys proces, musíme zjistit, které objekty jsou v procesu nějak sdíleny. Jak bylo zmíněno výše, často se jedná o vnitřní zdroje, jako jsou například programátoři, nebo technici. Z těchto zdrojů vytvoříme pool zdrojů. Jakákoliv instance si během svého provádění může vzít z poolu zdrojů kolik potřebuje.

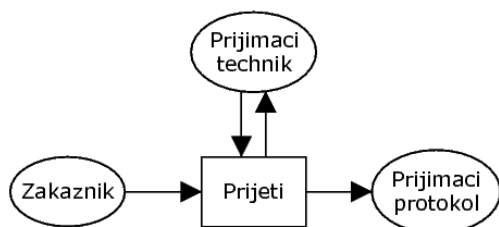
Na obrázku 14 je zachycena možnost přidání informace o sdílených zdrojích. Ke každému zdroji, který má povahu sdíleného zdroje je přidána informace o tom, kolik zdrojů z poolu je v aktivitě použito a informace o velikosti poolu. Konkrétně na obrázku je u aktivity přijetí sdílený zdroj přijímací technik. K přijetí je potřeba pouze jeden technik, ale velikost poolu je pět. Proto tuto informaci zapíšeme jako 1/[5].



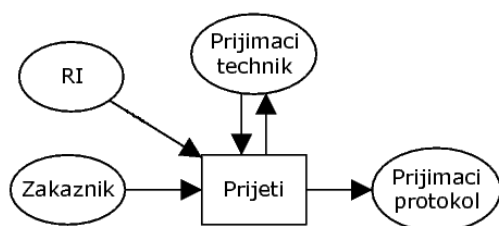
Obrázek 14: Přidání poolu sdílených zdrojů

Jak do petriho sítě zahrnout informaci o sdílených zdrojích? V petriho síti můžeme využít místo, které obsahuje stanovený počet tokenů - velikost poolu. A z tohoto místa potom budou aktivity brát zdroje. Musíme ale zabezpečit, že aktivita zdroj opět vrátí, jinak by mohlo dojít k zablokování. Například aktivitu přijetí z obrázku 14 můžeme zachytit modelem petriho sítě tak, že zdroj přijímací technik bude zahrnut do poolu. Aktivita přijetí je zobrazena jako přechod, který čerpá zdroje z poolu přijímací technik. Po

dokončení aktivity tyto zdroje opět vrátí do poolu. Přesně tuto situaci zachycuje obrázek 15.



Obrázek 15: Zachyceni poolu v petriho síti



Obrázek 16: Řídící instance

Můžeme si teď ale představit takovou situaci, že máme aktivitu, která nemá žádné vstupní zdroje, využívá pouze zdroje z poolu a jen někde v posloupnosti aktivit čeká až budou dokončeny aktivity před ní. Jestli ale aktivita využívá pouze sdílený zdroj, mohla by si ho okamžitě vzít z poolu a nečekat na dokončení aktivit před ní. Vystává tím problém, jak s pomocí sdílených zdrojů zachovat tok procesu. Proto je nutné do modelu zavést další informaci, která umožní zachování toku procesu - řídicí instanci. Aktivita se pak provede pouze tehdy, je-li na jejím vstupu právě řídicí instance. Sdílenými zdroji jsme totiž v petriho síti nahradili vstupní místo, které reprezentovalo jeden zdroj ze sdílených zdrojů, poelem zdrojů. Proto je nutné přidat místo, které zachová vstupní místo s vlastností jednoho zdroje. A tím je právě řídicí instance. V případě, že aktivita využívá více sdílených zdrojů, stačí řídicí instance pouze jedna.

Obrázek 16 zachycuje řídicí instanci u aktivity přijetí. Aktivita se vykoná tehdy, budou-li dostupné všechny zdroje a bude dostupná i řídicí instance.

Jak je vidět, simulační náhled není jen o tom, jak se bude proces simulovat, ale také o tom, jaké informace jsou pro simulační model potřebné. Důležitým tématem je, jak tyto informace zobrazit, aby byly přehledné a na pohled jasně srozumitelné. Proto ještě u tabulky informací můžeme rozlišit barevné pozadí buňky u průběhu čekání na aktivitu a pozadí průběhu trvání aktivity. Také by bylo možné prvek tabulky rozšířit o další informace jako je identifikace aktivity, neboť stejná aktivita, v případě dostupnosti zdrojů, se může provádět v rámci jedné instance paralelně sama se sebou. V takovém případě by bylo vhodné ještě tabulku rozšířit o možnost procházení přes všechny takto paralelně prováděné aktivity.

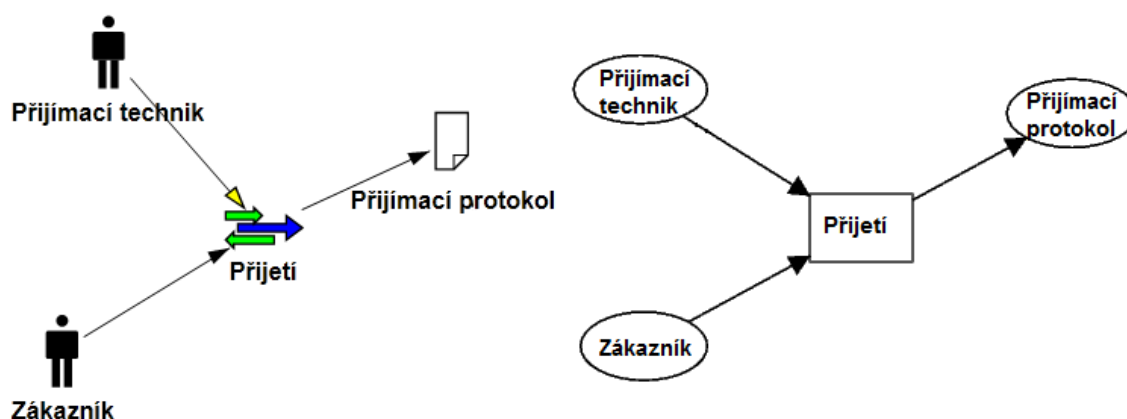
## 5 Zjednodušený náhled na byznys proces

Pro byznys manažery může být přečtení procesu zachyceného modelem petriho sítí, nebo koordinačním modelem metody BPM složité. V koordinačním modelu se vyskytuje spousta objektů a vazeb mezi nimi. Při validaci procesu je ale vhodné, aby model validoval někdo, kdo zná doménu modelovaného procesu - často byznys manažer. Vzhledem ke složitosti koordinačního modelu je potřeba model zjednodušit, nebo převést na jiný model.

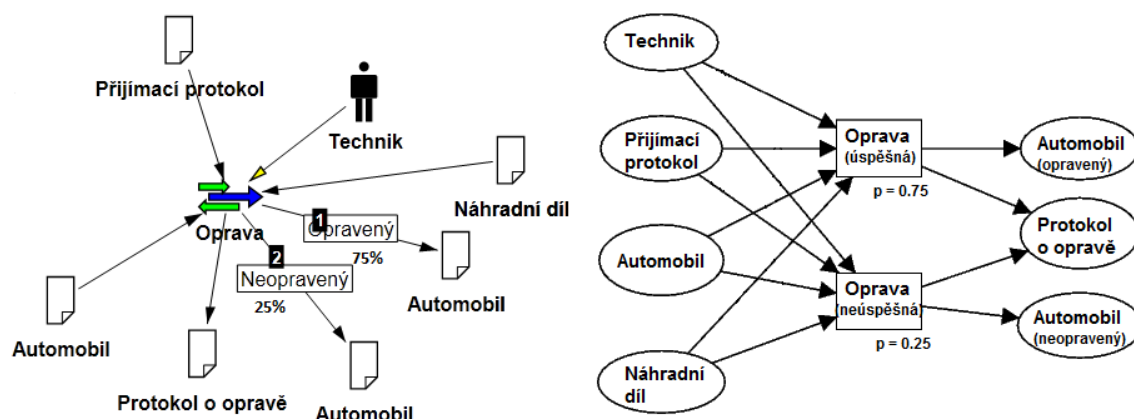
### 5.1 Převod na petriho síť

Protože koordinační model vychází z petriho sítí, nabízí se možnost tento model na model petriho síť převést. Objekty koordinačního modelu jsou v petriho síti zobrazeny místy a aktivity přechody, na které jsou tyto místa navázány. Zůstává tak zachováno, že aktivita lze provést pouze tehdy, jsou-li dostupné všechny její vstupní zdroje. Takto lze zachytit sekvence aktivit, které nemají alternativní scénáře. Obrázek 17 zobrazuje aktivitu koordinačního modelu s objekty jako petriho síť tvořící jeden přechod a počet míst odpovídá počtu objektů.

U alternativních scénářů se každý scénář může provádět s určitou pravděpodobností a tu je je potřeba zavést i do modelu petriho sítě. Proto se pro každou větev alternativy vytvoří samostatný přechod, který navíc obsahuje informaci o pravděpodobnosti provedení. Jde tedy o vytvoření rozhodnutí o tom, která větev alternativy se provede. Pokud by byl vytvořen pouze jeden přechod jako pro celou aktivitu, nebylo by možné zachytit, které objekty jsou výstupní pro kterou alternativu. Pro všechny alternativy jsou jako vstupní objekty všechny vstupní objekty aktivity - všechny vstupní místa jsou navázány na všechny přechody tvořící alternativy. Podle definice lze takovou část petriho sítě označit za volnou síť. Na obrázku 18 je zobrazena aktivita s alternativními scénáři a její převod na model petriho sítě.



Obrázek 17: Zjednodušení koordinačního modelu petriho sítí



Obrázek 18: Alternativní scénáře v petriho síti

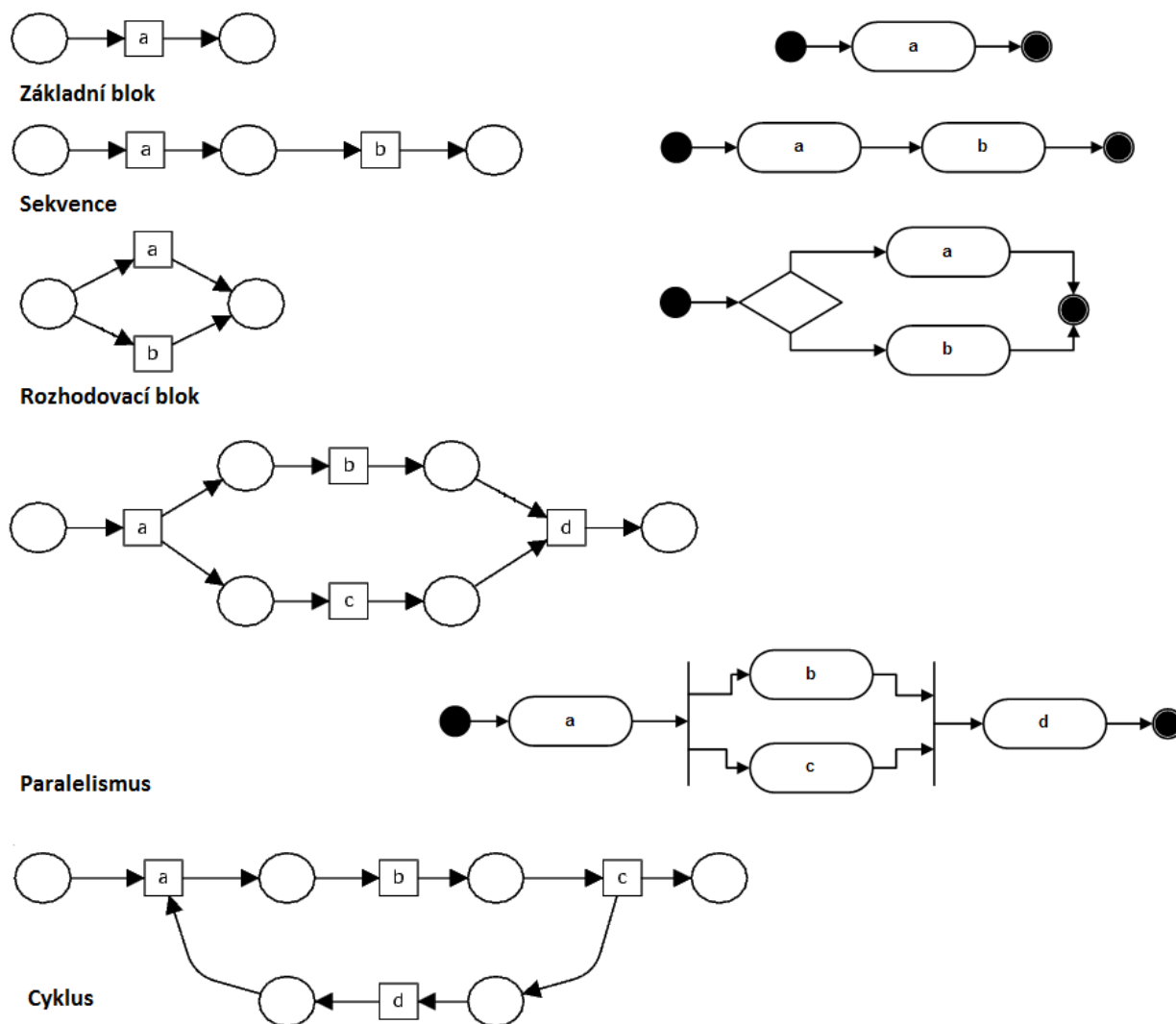
Koordináční model umí zachytit i zodpovědnost za aktivitu. Taková informace v modelu petriho síti zachytit nelze, protože pomocí míst je možné zachytit pouze objekty. Za aktivitu je zodpovědný právě objekt a zodpovědnost je vlastnost asociace mezi objektem a aktivitou. Petriho síť je graf a tedy nelze takovou asociaci zachytit. Každému místu ale můžeme přiřadit textový popis. Pro zachycení zodpovědnosti za aktivitu, tak můžeme zvolit zvýraznění popisu místa například tučným, nebo podtrženým písmem, nebo před název objektu tvořícího popis místa můžeme zařadit nějaký speciální symbol, například symbol hash (#).

## 5.2 Převod na aktivitní diagram

Během modelování procesu pomocí petriho síti je dobré držet se konstrukcí, pomocí kterých je možné zachytit proces bez nechtěných vlastností jako je třeba uzamčení, nebo nadměrné hromadění tokenů (zdrojů). Tyto konstrukce jsou odvozeny od elementárních konstrukcí programovacích jazyků:

- Příkaz - základní blok
- Sekvence příkazů
- Podmínka - rozhodovací blok
- Cyklus
- Paralelismus

Použití takových konstrukcí pro modelování byznys procesu umožňuje převést model petriho síti i na jiný model, například na aktivitní diagram. Ten je vhodný v případě, že by byl koordináční model, nebo model petriho síti příliš složitý, nebo pokud je potřeba mít jednoduchý náhled na byznys proces. Jak převést konstrukce petriho síti na aktivitní diagram je zobrazeno na obrázku 19.



Obrázek 19: Základní konstrukce petriho sítě

Stejně tak by bylo možné převést prvky petriho sítě na prvky modelu BPMN. Tento model by byl daleko lepší, co se týče náhledu na proces, neboť dokáže zobrazit i reakce na zprávy, čas a také umožňuje zachytit i zpracovávání objektů. Z pohledu vedoucího podniku je ale potřeba mít náhled na proces co nejjednodušší, protože takový člověk určitě modely zachycující byznys proces nezná a také jistě nemá čas učit se a řešit složitější modely. Zvolil jsem proto pro zjednodušený náhled aktivitní diagram. Ten umožní zachytit hlavně tok procesu a dalšími úpravami lze snadno takový model obohatit o další prvky, například o vstupní a výstupní objekty.

Podle základních konstrukcí by se intuitivně dalo definovat zobrazení prvků petriho sítě na prvky aktivitního diagramu.

Přechod  $\rightarrow$  Aktivita

Jedno místo je napojeno na dva přechody  $\rightarrow$  Rozhodovací blok typu split

Dva přechody napojené na jedno místo  $\rightarrow$  Rozhodovací blok typu join

Jeden přechod je napojen na dvě místa  $\rightarrow$  Parní blok typu split

Dvě místa jsou navázána na jeden přechod  $\rightarrow$  Paralelní blok typu join

Vstupní / výstupní objekty  $\rightarrow$  vstupní / výstupní objekty

### 5.2.1 Popis převodu

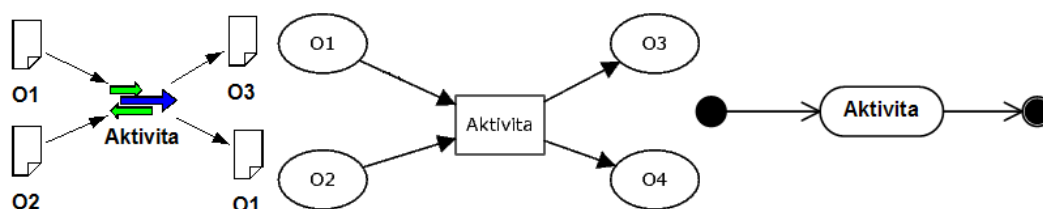
Jak bylo zmíněno v kapitole o petriho sítích, petriho síť je bipartitní orientovaný graf, který obsahuje množinu míst  $P$  a množinu přechodů  $T$ . Přechod může mít místa, která mu předcházejí - množinu vstupních míst ( $\bullet t$ ) a místa, která ho následují - množinu výstupních míst ( $t \bullet$ ). Stejně tak i místa mohou mít přechody, které jim předcházejí ( $\bullet p$ ) a přechody, které je následují ( $p \bullet$ ).

Formalizace jednotlivých částí petriho sítě

#### 1. Základní blok

$$(\exists t \in T)[\bullet t \neq 0][t \bullet \neq 0]$$

Základní blok je tvořen přechodem, který má vstupní i výstupní místa. V aktivním diagramu se tento blok zobrazí jako jedna aktivita. Vstupní objekty jsou tvořeny vstupními místy a výstupní objekty výstupními.



Obrázek 20: Základní blok

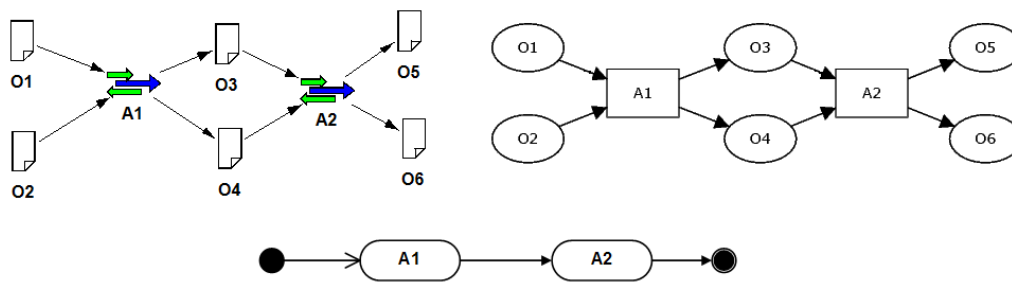
#### 2. Sekvence

$$(\exists t_1, t_2 \in T)[t_1 \bullet \neq 0][\bullet t_2 \neq 0][t_1 \bullet \subseteq \bullet t_2]$$

Sekvence je složením základních bloků za sebe tak, aby proces po sekvenci aktivit skončil. Výstupní objekty první aktivity jsou vstupní objekty pro druhou aktivitu.

#### 3. Rozhodovací blok

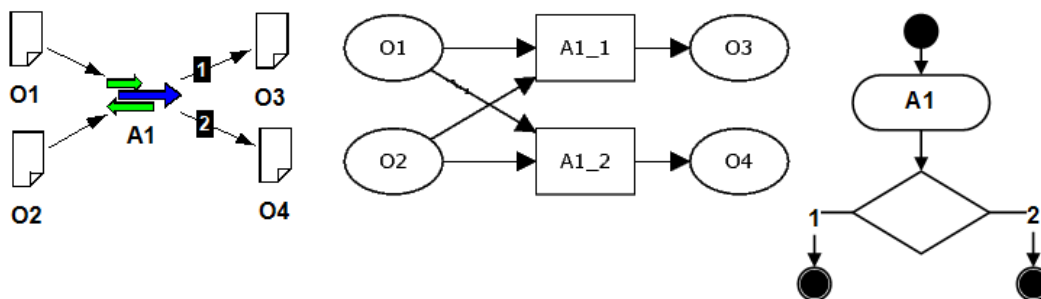
$$(T_v \subset T)(P_v \subset P)(\forall t \in T_v)(\forall p, p' \in P_v)[p \bullet = p' \bullet]$$



Obrázek 21: Sekvence

V petriho síti se jedná o část, která je zachycena volnou sítí - pro všechny přechody v takové síti platí, že mají stejná vstupní místa.[8] Tato část sítě se zobrazí na aktivním diagramu tak, že se vybere jeden reprezentant z množiny přechodů, které takovou síť tvoří a zobrazí se na aktivitu v aktivním diagramu. Na tuto aktivitu se naváže rozhodovací blok a každý přechod, který tvoří volnou síť se zobrazí na alternativní tok z rozhodovacího bloku.

Vstupní objekty pro aktivitu, která byla vytvořena z jednoho zástupce z množiny přechodů, tvoří vlastně všechny vstupní místa tohoto přechodu. Výstupní objekty jsou tvořeny místy, které vystupují z každého přechodu, který tvoří volnou síť. Pokud je některé výstupní místo spojeno s dvěma přechody, bere se pouze jednou.



Obrázek 22: Alternativy

#### 4. Paralelní blok - split

$$(\exists t \in T)(\exists p_1, p_2 \in P)[t^\bullet \ni \{p_1, p_2\}, p_1^\bullet \neq p_2^\bullet]$$

Paralelní zpracování aktivit je v BPM zachyceno tak, že z jedné aktivity jsou výstupní objekty využity dalšími aktivitami tak, že existuje dvojice výstupních objektů aktivity, které vstupují do různých aktivit. Jinými slovy, dvě aktivity nemohou mít některý vstupní objekt stejný. V aktivním diagramu je toto zachyceno tak, že aktivity, ze které výstupní objekty splňují výše popsanou podmínku, je zobrazena před

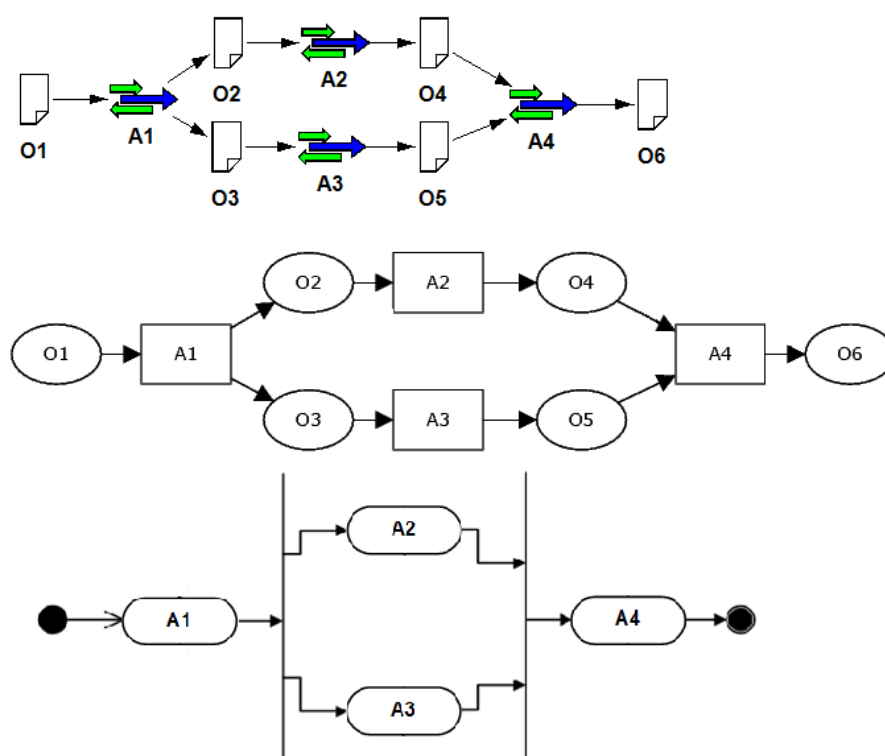


začátkem paralelního bloku a za paralelním blokem jsou aktivity zobrazeny podle jednotlivých částí petriho sítě.

#### 5. Paralelní blok - join

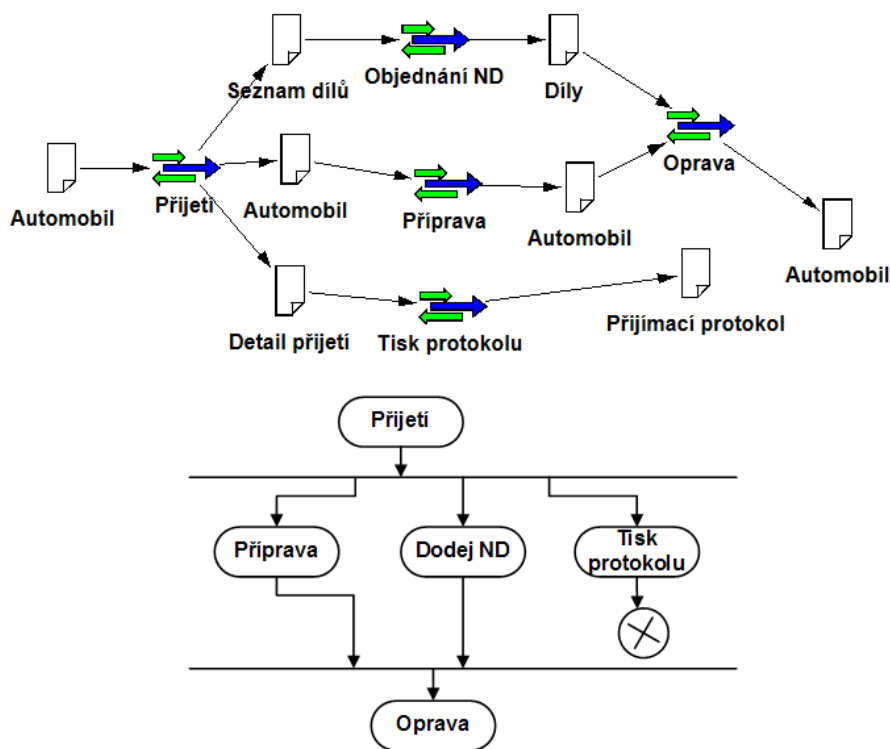
$$(\exists p_1, p_2 \in P)(\exists t \in T)[\bullet p_1 \neq \bullet p_2, \bullet t \ni \{p_1, p_2\}]$$

Ukončení paralelního zpracování je v BPM, potažmo v petriho síti zachyceno tak, že výstupní objekty paralelně probíhajících aktivit tvoří vstupní objekty pro jednu aktivitu. Ta podle pravidel pro dynamiku petriho sítě nemůže proběhnout, pokud nemá dostupné všechny zdroje. Čeká se tedy na ukončení takových paralelních větví, kde jsou výstupní objekty jednotlivých větví jako vstupní pro jednu aktivitu. V aktivním diagramu jsou paralelní větve před koncem paralelního bloku, za kterým následuje aktivita.



Obrázek 23: Paralelismus

Během paralelního zpracování může dojít k situaci, kdy neočekáváme ukončení všech paralelních větví. Na obrázku 24 jsou z aktivity přijetí vytvořeny tři objekty, které vstupují do aktivit, jež probíhají paralelně. Ale aktivita tisk protokolu vytiskne protokol, který se v procesu již dále nepoužije. Z toho je zřejmé, že po vytištění protokolu dojde k ukončení jedné větve paralelismu bez nutnosti ukončit větve ostatní. I tuto možnost lze zachytit aktivním diagramem.



Obrázek 24: Ukončení jedné větve paralelismu bez ukončení ostatních

## 6. Cyklus

Cyklus je konečná posloupnost aktivit, kdy se z počáteční aktivity dostaneme touto posloupností zpět do počáteční aktivity. Protože můžeme koordinační model převést na petriho síť, která je bipartitní orientovaný graf, můžeme využít grafové algoritmy pro detekci cyklu.

Kružnice je takový graf  $C_n = (V, E)$ , kde  $V = \{v_1, \dots, v_n\}$  a  $E = \{e_1, \dots, e_n\}$  a platí:  $e_i = (v_i, v_{i+1}), i = 1, \dots, n - 1$  a  $e_n = (v_n, v_1)$

Definici pro bipartitní graf lze zavést následovně:

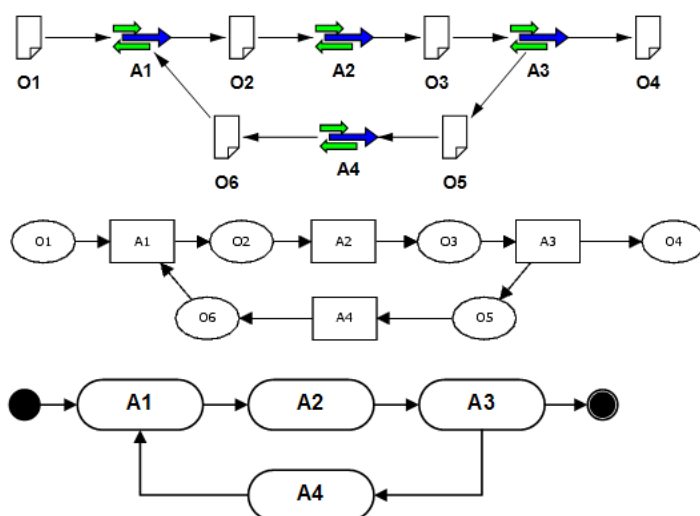
$C_n = (T, P, E)$ , kde  $T = \{t_1, \dots, t_n\}$ ,  $P = \{p_1, \dots, p_n\}$  a  $E = \{e_1, \dots, e_n\}$  a platí:

$e_j = (t_i, p_i), i = 1, \dots, n - 1$ , pro  $j$  lichá

$e_j = (p_{i-1}, t_i), i = 1, \dots, n - 1$ , pro  $j$  sudá

$e_n = (p_n, t_1)$

Převod cyklu z petriho sítě na aktivitní diagram je realizován tak, že pro každý přechod petriho sítě se vytvoří aktivita a všechny takto vytvořené aktivity se spojí orientovanými hranami ve směru cyklu v petriho síti. Zjednodušeně lze říci, že se vytvoří petriho síť, která neobsahuje místa a dovoluje hrany mezi přechody.



Obrázek 25: Cyklus

O převodu mezi modely aktivního diagramu a petriho sítě popisuje více dokument Formalizing activity diagram of uml by petri nets.[9]

### 5.2.2 Objekty v aktivním diagramu

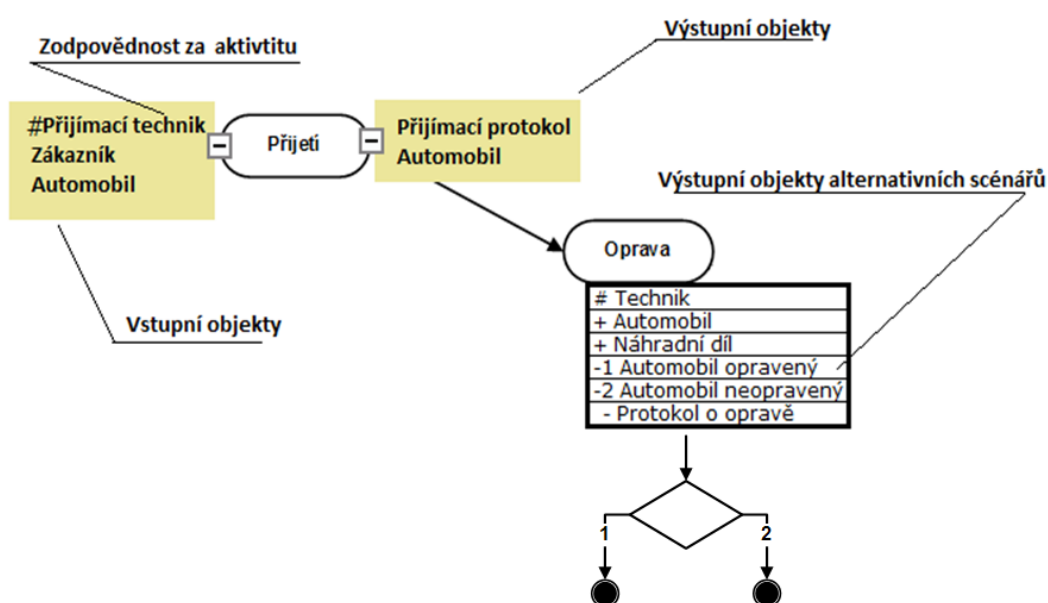
Aktivní diagram umožňuje znázornění objektů, které jsou předávány mezi aktivitami.[6] Ale v koordinačním modelu existují vstupní a výstupní objekty a ne vždy se všechny výstupní objekty musí použít jako vstupní pro další aktivitu. Například v situaci, kdy chceme znát pouze tok procesu, nás nemusí nutně zajímat objekty, které se v tomto toku objevují a zpracovávají. Možností jak zachytit vstupní a výstupní objekty aktivity je tak spousta. Jako nejjednodušší možnost se nabízí objekty přidat jako přídatnou informaci - vstupní objekty na levou stranu, výstupní na pravou. Takové zobrazení ale může být nevýhodné ve větších procesech, do kterých může přinášet nepřehlednost a zmatek. Případně může tato možnost být vhodná, pokud se proces zachytí na papír.

V programech můžeme zvolit úpravu modelovacího prvku. Prvek, který znázorňuje aktivitu upravíme tak, že mu na levou a pravou stranu přidáme tlačítka se symbolem plus. Po rozkliknutí těchto tlačítek program zobrazí vstupní, nebo výstupní objekty dané aktivity. Po rozkliknutí se symbol tlačítka změní na symbol mínus a umožní dalším kliknutím skrytí informace o objektech.

Další možností je upravit modelovací prvek na prvek podobný znázornění třídy v třídním diagramu v UML. Můžeme využít i dalších analogie z třídního diagramu pro označení vstupních a výstupních objektů - označení viditelnosti atributů. Symbolem plus označíme vstupní objekty, symbolem mínus objekty výstupní. V případě alternativních scénářů označíme výstupní objekty symbolem mínus následovaný číslem scénáře.

Mezi objekty patří i zodpovědnosti za aktivitu a proto je nutné zachytit i tuto informaci do aktivního diagramu. V koordinačním modelu může být zachyceno více zodpovědných objektů. Je tak důležité najít vhodnou reprezentaci pro zachycení aktivním diagramem. Protože zodpovědnost je znázorněna v koordinačním modelu objektem, je zřejmé, že bude zahrnuta mezi seznam objektů. V seznamu objektů ji můžeme nějak zvýraznit, nebo před názvem objektu zapsat speciální symbol, například hash (#). Tento speciální symbol můžeme také využít, pokud objekty zapisujeme do upraveného prvku.

Informace o zodpovědnosti však není uvedena v petriho síti, ze kterého se model převede na aktivní diagram. V petriho síti je ale informace o všech objektech, pouze není zachycena vlastnost objektu. Proto je potřeba se na tuto vlastnost podívat zpět do koordinačního modelu. Místo možnosti zahrnout zodpovědnost za aktivitu do seznamu



Obrázek 26: Rozšířený aktivní diagram

vstupních objektů, existuje možnost zachytit zodpovědnosti do tzv. plaveckých dráh, viz. obrázek 4. V aktivním diagramu je možné zachytit aktivitu ve více plaveckých drahách - v koordinačním modelu existuje více zodpovědných objektů. Ne vždy ale mohou být tyto plavecké dráhy vedle sebe a pak může být zachycení zodpovědnosti aktivit složité.

Koordinační model BPM zachycuje provádění jednotlivých aktivit a práci se zdroji v procesu. Nezachycuje však počátek ani konec procesu. Proces může mít více počátečních i koncových aktivit. Proto není možné tímto převodem zachytit v aktivním diagramu tyto údaje. V případě, že by bylo řečeno, kterými aktivitami proces začíná, bylo by možné spuštěním simulace procesu zjistit koncové aktivity a případně posléze tyto informace do aktivního diagramu zaznamenat.

Převod z koordinačního modelu BPM do aktivního diagramu je postaven na určitých konstrukcích petriho sítě, které se zobrazují na určité konstrukce aktivního diagramu.

Pro účely simulace se koordinační model doplňuje o další informace jako jsou například časové intervaly trvání aktivit, čekání na aktivity a pravděpodobnostní určení alternativních scénářů. Převod takto rozšířeného modelu na petriho síť způsobí, že síť obsahuje více míst a přechodů, které zachycují informace rozšířeného modelu. Převod koordinačního modelu na aktivitní diagram by v případě, že by model petriho sítě obsahoval informace rozšiřující model pro potřeby simulace, nebyl tak snadný, protože například u alternativních scénářů by muselo dojít k vynechání přechodů, které obsahují pravděpodobnos vykonání dané alternativy. Proto je převod znázorněn pouze z petriho sítě, která informace pro účely simulací neobsahuje.

Obrázek 26 zachycuje zjednodušený náhled pomocí prvků aktivitního diagramu na proces zachycený na obrázku 10. Vstupní a výstupní objekty jsou zachyceny u jedné aktivity formou prvku s tlačítky plus a minus, u druhé aktivity jsou zobrazeny jako součást prvku a je využito symbolů pro jejich označení.

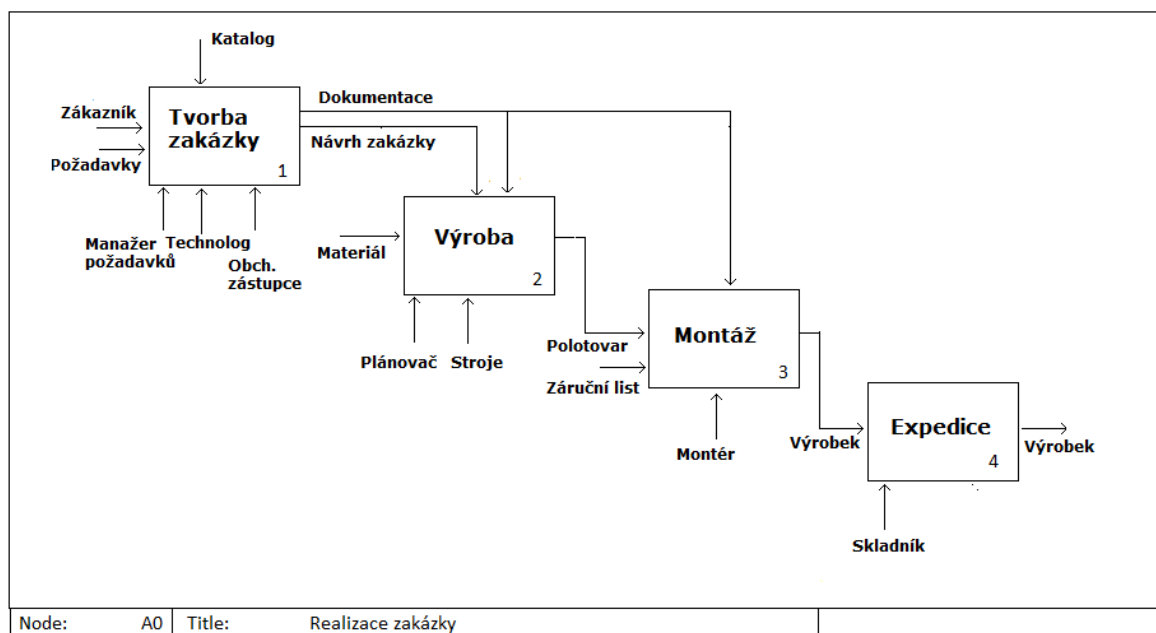
## 6 Byznys proces výroby

Na obrázku 28 je zachycený byznys proces výroby. Je zřejmé, že náhled na proces je složitý, vystupuje v něm spousta zdrojů a mnoho aktivit má alternativní scénáře. Není však zcela zřejmý tok procesu. Pro zjednodušení si můžeme proces rozdělit na několik dílčích kroků.

1. **Tvorba zakázky** - tímto krokem byznys proces začíná. Od zákazníka se přijímají jeho požadavky a následně se vypracuje dokumentace a návrh zakázky. Poté je vedena se zákazníkem konzultace, kde zákazník může předložit, nebo upravit požadavky nebo zrušit zakázku. Jestliže zákazník má další připomínky, vypracuje se opět dokumentace a návrh zakázky. Jestliže se vším souhlasí, přejde se k plánování.
2. **Výroba** - začíná aktivitou plánování, ve které se zakázka zařadí mezi již zpracovávané zakázky tak, aby odpovídala termínu dokončení, který požaduje zákazník. Po naplánování se může rozjet výroba. Ta začíná aktivitou výsek materiálu. V této aktivitě zpracovává odborný pracovník na CNC stroje materiál a vytváří z něj polotovar. S určitou pravděpodobností se stane, že stroj vyrobí zmetek. V několika případech lze ale zmetek po menších úpravách zařadit zpět do výroby. Po výseku materiálu nastává zpracování polotovaru. Tuto činnost provádí odborník na zámečnické práce. Na výsek materiálu a zpracování polotovaru dohlíží mistr výroby. Ten řeší různé problémy, například problémy v dokumentaci, zjišťování stavu nářadí a podobně. Během zpracování polotovarů mohou opět vznikat zmetky, které lze v určitých případech zařadit zpět do výroby. Po zpracování polotovarů následuje lakování, které provádí lakýrník. Na lakování dohlíží mistr lakovny, který řeší podobné problémy jako mistr výroby, například docházející barvu. V lakovně opět může docházet ke vzniku zmetků.
3. **Montáž** - po lakování polotovarů je výrobek smontován na montážní lince, kde pracuje několik montérů. Na montáž dohlíží mistr montáže, který dále zajišťuje pracovní prostředky pro správnou montáž výrobku. Výrobek je po smontování předán na balící linku, kde ještě před zabalením zkontroluje kontrolor, zda nejsou na výrobku vady. Přiloží se uživatelský manuál a záruční list. Poté se výrobek zabalí a uskladní.
4. **Expedice** - skladník přebírá výrobky po kontrole a balení a chystá je na expedici.

Pokud se během výroby a montáže vyrobí, nebo vyskytne zmetek, přijde kontrolor a rozhodne, zda je možné výrobek s takovou vadou pustit dále do výroby. Pokud to možné není, předává zmetek výrobnímu řediteli, který dá nový požadavek na plánování zakázky a zmetek zpracuje, například vyhozením do kontejneru.

Slovní popis procesu zabral spoustu místa a také zobrazení pomocí modelu IDEF0 na obrázku 27 není úplně ideální, protože vůbec nezachytává informaci o možné výrobě zmetků a jejich zpracování. V IDEF0 nejde tyto informace zachytit, protože se jedná o alternativní toky. Pomocí IDEF0 se zachytil akorát základní tok procesu, který vede k



Obrázek 27: Tok byznys procesu výroby

cíli. Proto využijeme konstrukcí a postupů z předchozích kapitol a proces převedeme na zjednodušený náhled pomocí aktivitního diagramu.

Nejprve si převedeme koordinační model na model petriho sítě podle popisu v kapitole 5.1. Mnoho aktivit v procesu má alternativní scénáře a proto se model petriho sítě značně rozroste, protože pro každou aktivitu s alternativními scénáři je nutné zobrazit tolik přechodů, kolik má alternativ. Aktivita tvorba dokumentace a zpracování požadavků na začátku procesu probíhají paralelně. Celý tento paralelní blok tvoří cyklus. Aktivita konzultace obsahuje tři alternativní scénáře. Znamená to tedy tři přechody pro tuto aktivitu a celkem sedm míst. Po aktivitě plánování následuje několik aktivit s alternativními scénáři. Pokud bychom si odmysleli alternativu číslo 2, dostaneme vlastně jen sekvenci aktivit. Alternativa 2 u těchto aktivit způsobí pouze spuštění přidružené aktivity, která taktéž má dva alternativní toky. V alternativě 1 dojde v modelu petriho sítě k vygenerování tokenu do místa zmetek a následnému zpracování přechodem zpracování zmetků. Alternativa 2 vždy bude vracet token do původního toku procesu tak, aby proces mohl dále pokračovat. Celý model petriho sítě je zobrazen na obrázku 30 v příloze. Model je vytvořený v programu CPN Tools, který je určen pro vytváření modelů petriho sítě a jejich simulaci. Pro účely simulace petriho sítě jsou na hrany vloženy závorky, které znamenají násobnost hran. Tento nástroj je také určen pro modelování barevných petriho sítí. Proto je u každého místa uvedený typ místa. Tyto informace jsou ale v tomto případě úplně bezpředmětné.

Model petriho sítě už potom snadno převedeme opět podle konstrukcí v kapitole 5.2 na aktivitní diagram. Všechny aktivity, které mají alternativní toky, se převedou na

---

prvky aktivity následované prvkem rozhodovacího bloku. Na začátku procesu se ještě vytvoří paralelní blok. Tento převod zachycuje obrázek 31 v příloze. Je vidět, jak hodně došlo k zjednodušení náhledu. Oproti koordinačnímu modelu je proces přehlednější a je zřejmý tok procesu. Tento náhled je ještě možné obohatit o vstupní a výstupní objekty, jak zachycuje obrázek 32 v příloze. Na tomto obrázku nejsou objekty u všech aktivit, protože by velikost obrázku přesáhla jednu stranu a velmi by se snížila čitelnost a názornost.

Na obrázku 29 v příloze je zachycen koordinační model obohacený o prvky pro simulační náhled. Už na první pohled je vidět, v jakém stavu jsou které aktivity. Například aktivita zmetky výsek není ve stavu čekání ani ve stavu spuštěno. Při detailnějším pohledu by bylo vidět, že nad touto aktivitou je pozastaven kurzor. Podle popisu chování v kapitole 4 je to přesně situace, kdy se tabulka s informacemi zobrazí u neprováděné aktivity ve chvíli, kdy je nad ní kurzor. Aktivita kontrola a balení je ve stavu spuštěno a provádění aktivity je v jedné třetině svého času. Je zřejmé, že aktivita, která se ještě neprovádí a ani nečeká, nemá stanovený přesný časový okamžik čekání, nebo svého trvání, jak je to vidět u aktivity zmetky výsek. Naopak u aktivity kontrola a balení jsou stanoveny jak časy čekání, tak trvání. Aktivita zpracování zmetků je aktuálně ve stavu čekání. Taktéž zatím nemá určený časový okamžik provádění aktivity. Ten se doplní až při spuštění aktivity. Stejně tak informace o aktuálně vybraném scénáři se vybere až v okamžiku spuštění aktivity.



## 7 Závěr

V práci jsem vytvořil návrh na modelovací prvky pro aktivitní diagram, které rozšiřují možnosti aktivitního diagramu pro zachycení zjednodušeného náhledu na byznys proces modelovaný metodou BPM. Také jsem vytvořil simulační náhled zahrnující nové prvky pro zobrazení informací v koordinačním modelu během simulace v BP studiu.

Práce zároveň ukazuje rozdíly mezi jednotlivými metodami pro modelování a pro každou metodu zobrazuje několik modelů, kterými lze zachytit byznys proces. Taktéž popisuje rozdíly mezi jednotlivými modely, popisuje jejich výhody a nevýhody a možné problémy při modelování byznys procesů.

BP studio neumožňuje dobře sledovat stav aktivit během simulace a proto bylo potřeba simulační náhled rozšířit o prvky, které by umožňovaly zachytit informace o stavu aktivit. Myslím si, že seznam informací, tak jak byl uveden v této práci, je velmi vhodný a snadno rozšiřitelný. Vzhledem k tomu, že simulace v BP studiu lze provádět pouze manuálně, mohlo by určitým rozřízením být vytvoření automatických simulací. S tím by se tabulka informací mohla rozšířit o identifikátor instance a možnost procházet informace u jedné aktivity přes všechny dokončené instance. Také se nabízí možnost vytvořit tabulku informací pro celý proces. V té by mohlo být uvedeno, jaké procento aktivit je dokončeno, jaké je využití zdrojů a podobně. U automatických simulací by se takto dalo sledovat například stresové chování procesu.

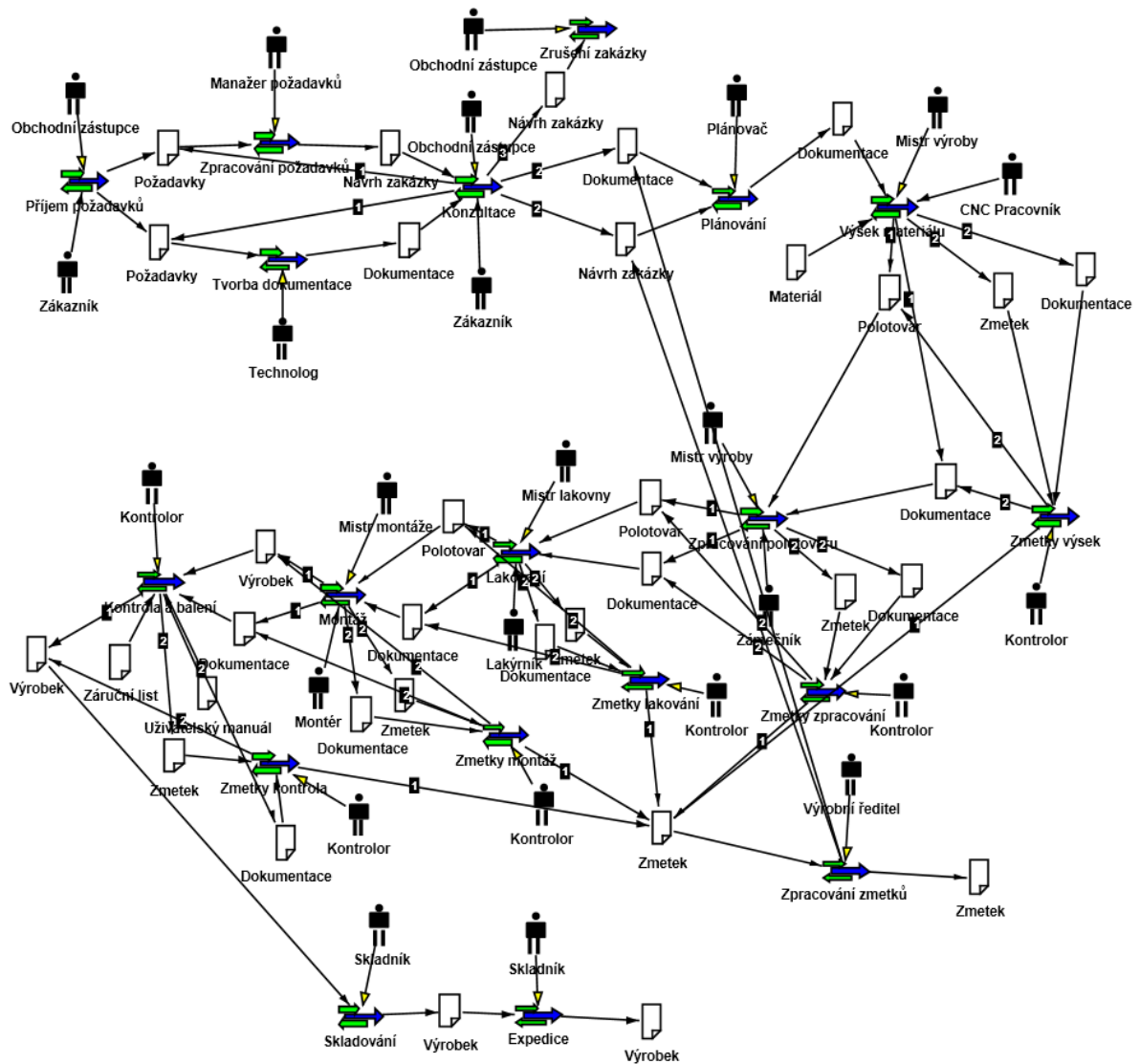
Také zjednodušený náhled dobře ukazuje, jak je možné provázat dva různé modely a využít tak jednoduššího modelu k validaci složitějšího. Práce v tomto směru ukazuje, že je možné modelovat byznys proces složitějším modelem a ten pak převést na model jednodušší, který je názornější, přehlednější a přitom vypovídací hodnota zůstává velmi podobná vypovídací hodnotě složitějšího modelu. Popis pro převod modelů by šel určitě udělat ještě lépe a taktéž se nabízí vytvořit programové vybavení pro automatický převod.

Na práci by mohlo navázat vytvoření programového vybavení v BP studiu pro vytvoření zjednodušeného náhledu na koordinační model. Dále by se v BP studiu mohl vytvořit simulační náhled podle této práce, který by ke každé aktivitě přiřadil informační tabulku. Taktéž by se BP studio mohlo rozšířit o automatizaci simulací nad koordinačním modelem a vytváření statistik o chování procesu v různých stupních zátěže.

## 8 Reference

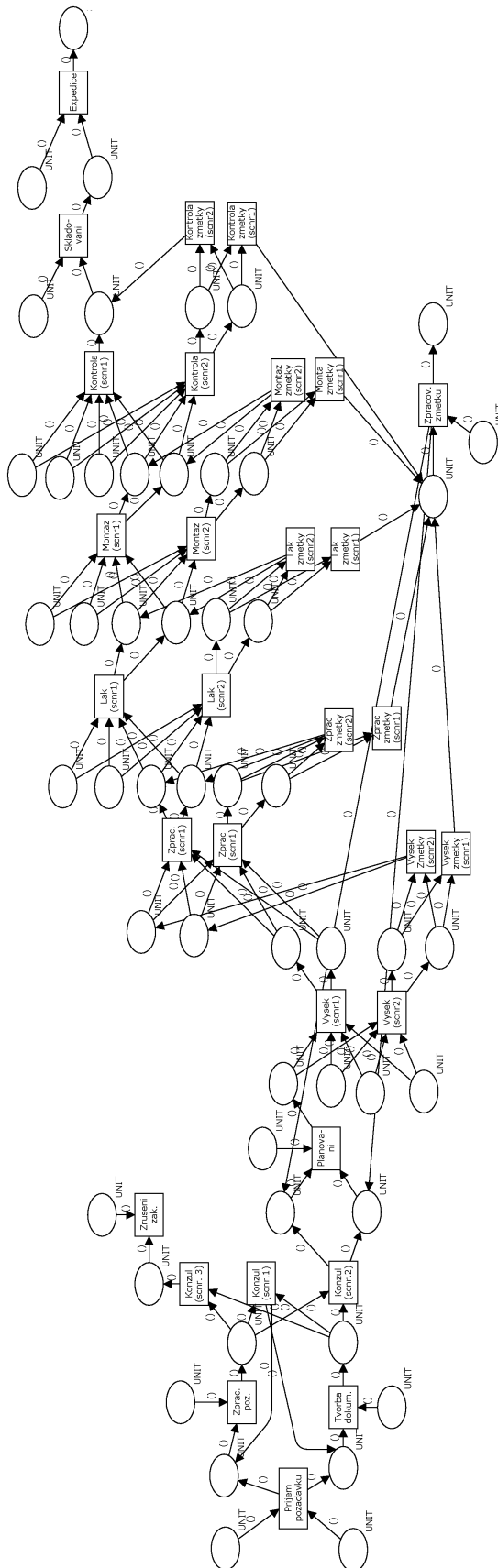
- [1] Vondrák, I.: Metody byznys modelování
- [2] Vondrák, I.: Úvod do softwarového inženýrství
- [3] Kuchař, Š., Kožusznik, J.: BPM METHOD EXTENSION FOR AUTOMATIC PROCESS SIMULATION
- [4] Řepa, V.: Podnikové procesy. 2.vydání, Grada Publishing, 2007
- [5] Dlouhý M., Fábry J., Kuncová M., Hladík T.: Simulace podnikových procesů
- [6] <http://www.omg.org>: Unified Modeling Language: Superstructure  
<http://www.omg.org/spec/UML/2.0/Superstructure/PDF/>, [cit. 2011-04-20]
- [7] <http://www.omg.org>: Business Process Model and Notation (BPMN)  
<http://www.omg.org/spec/BPMN/2.0/PDF/>, [cit. 2011-03-15]
- [8] Markl, J.: Skriptum Petriho sítě 1
- [9] Tričkovic, I.: Formalizing activity diagram of uml by petri nets  
[http://www.dmi.uns.ac.rs/NSJOM/Papers/30\\_3/NSJOM\\_30\\_3\\_161\\_171.pdf](http://www.dmi.uns.ac.rs/NSJOM/Papers/30_3/NSJOM_30_3_161_171.pdf),  
[cit. 2011-04-30]
- [10] Ježek D., Kožusznik J., Kuchař Š., Štolfa S.: Petri Net Based Simulation for SPI, FU-BUTEC'2011 London, 2011

## **A Byznys proces výroby**

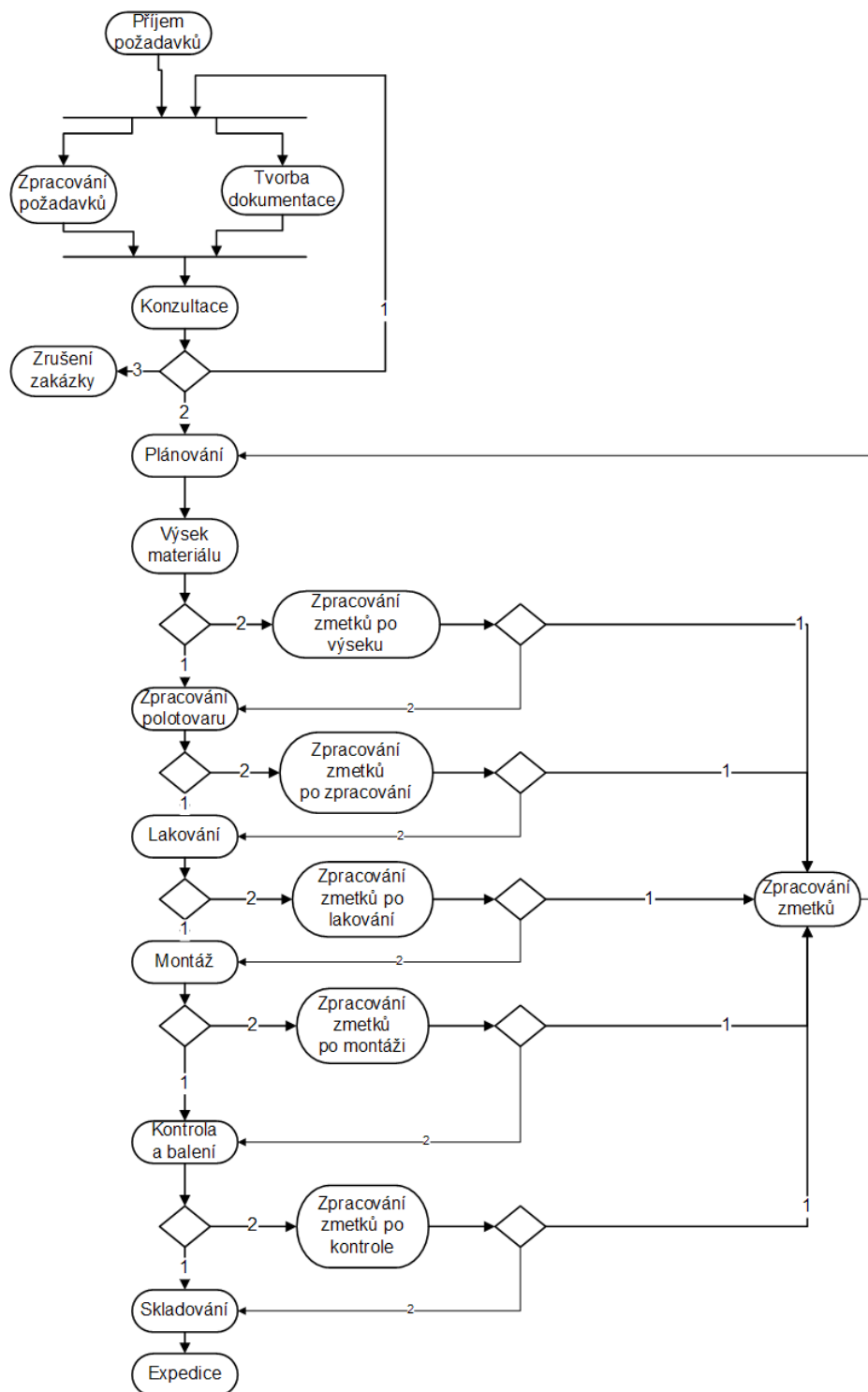


Obrázek 28: Koordinační model BPM zachycující proces výroby

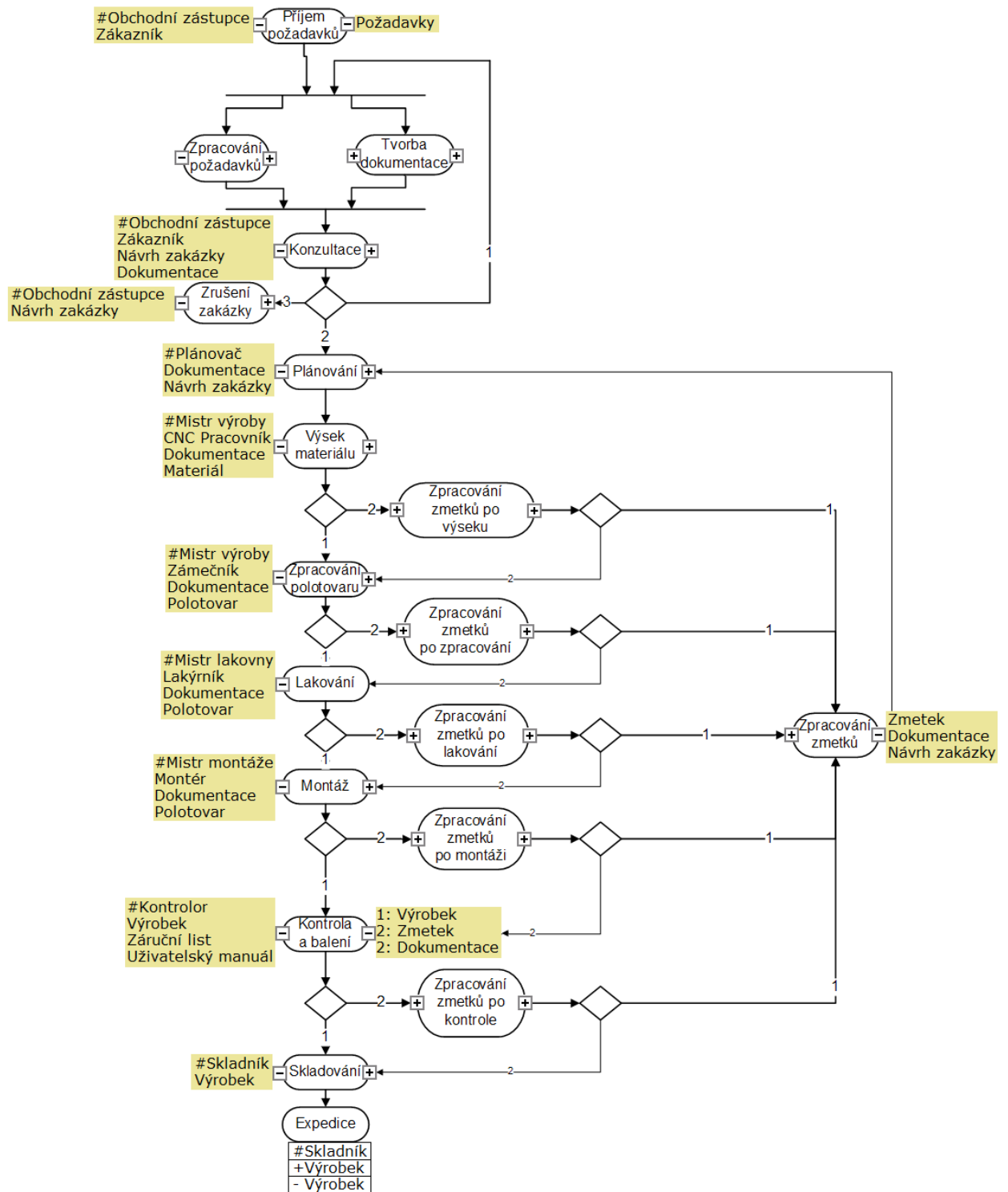




Obrázek 30: Model byznys procesu výroby zachycený petriho sítí podle koordinačního modelu



Obrázek 31: Zjednodušený náhled na byznys proces



Obrázek 32: Zjednodušený náhled na byznys proces s přidávanými objekty