



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Elektronizace procesů pomocí BPM nástroje Camunda
<b>Student:</b>	Dominik Stasiowski
<b>Vedoucí:</b>	Ing. Lukáš Zoubek
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Informační systémy a management
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2020/21

### Pokyny pro vypracování

Fakulta elektrotechnická provozuje aplikaci elektronizující agendu studentských žádostí na studijním oddělení. Aplikace je postavena na procesnímu enginu Camunda a je složena ze dvou různých uživatelských rozhraní. Jedno (volně dostupná nadstavba "Camunda web-app") slouží pro přístup zaměstnanců, druhé, vlastní, rozhraní pro přístup studentů nese název eProcesy. Cílem práce je vytvořit samostatně fungující prostředí eProcesy a zbavit se závislosti na využívané nadstavbě Camunda web-app. Konkrétní cíle:

- Analyzujte filtry a UX funkcionality formulářů v rozhraní Camunda web-app.
- Navrhněte funkcionality filtrů do rozhraní eProcesy.
- Navrhněte definice UX funkcionalit pomocí BPMN.
- Implementujte filtry do rozhraní eProcesy.
- Upravte implementaci automatických formulářů v rozhraní eProcesy tak, aby podporovaly nově vymyšlené definice z BPMN.
- Zhodnoťte časovou a finanční náročnost vytvoření vlastního řešení a porovnejte s cenou licence Camunda.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 23. ledna 2020





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Elektronizace procesů pomocí BPM nástroje Camunda**

*Dominik Stasiowski*

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Lukáš Zoubek

4. června 2020



---

## Poděkování

V první řadě bych rád mnohokrát poděkoval Ing. Lukáši Zoubkovi, vedoucímu mé práce, za jeho cenné rady, pohotovou spolupráci a vstřícnost při tvorbě této závěrečné práce. Důležité poděkování si také zaslouží Ing. Denis Baručić za jeho konzultaci a aktivní spolupráci při tvorbě praktické části této práce. Děkuji mé rodině, která mě nepřestala podporovat napříč celým bakalářským studiem i touto závěrečnou prací. Děkuji i svým přátelům, kteří tuto cestu prožívali a sdíleli společně se mnou.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisu. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisu, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 4. června 2020

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2020 Dominik Stasiowski. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Stasiowski, Dominik. *Elektronizace procesů pomocí BPM nástroje Camunda*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.



---

# Abstrakt

Tato práce se zabývá problematickou implementací procesní aplikace eProcesy/Camunda na Fakultě elektrotechniky ČVUT v Praze. Aktuální implementace přináší komplikovanou rozšiřitelnost a udržitelnost aplikace. Cílem práce je implementovat rozšíření rozhraní eProcesy o funkcionality webové aplikace Camunda Tasklist a odstranit tak závislost na jejím použití. Jedná se o funkcionality filtrování úkolů a zobrazení formulářů reprezentující dané úkoly. Následuje ekonomicko-manažerské vyhodnocení, za jakých podmínek je výhodnější přistoupit k vlastnímu řešení potřebných funkcionalit nebo problémy vyřešit zakoupením licence Camunda Enterprise. Vybrané funkcionality pro tuto práci jsou implementovány v rozsahu, který vyšel z analýzy reálného použití funkcionalit. Implementace je finančně vyčíslena a její výsledek je pak použit jako jeden z aspektů pro ekonomicko-manažerské vyhodnocení. Na základě finanční náročnosti implementovaných funkcionalit a podmínek licence Camunda Enterprise je vyhodnoceno, že varianta vlastního řešení je pro situaci Centra znalostního managementu a Fakulty elektrotechnické ČVUT v Praze výhodnější. Zakoupení licence Camunda Enterprise by bylo vhodné například pro případ automatizace procesů napříč celou univerzitou.

**Klíčová slova** implementace rozšíření, camunda web-app, filtry, formuláře, manažerské zhodnocení, licence Camunda Enterprise, eProcesy, elektronizace procesů

# Abstract

This work deals with the problematic implementation of the process application eProcesy/Camunda at the Faculty of Electrical Engineering at CTU in Prague. The current implementation brings complicated extensibility and sustainability of the application. The aim of the work is to implement and extend the eProcesses interface with the functionality of the web application Camunda Tasklist and thus get rid of the dependence on its use. This is the functionality of filtering tasks and displaying forms representing the tasks. The following is an economic-managerial evaluation of the conditions under which it is more advantageous to approach your own solution of the necessary functionalities or to solve problems by purchasing a Camunda Enterprise license. Selected functionalities for this work were successfully implemented in full. The implementation was financially quantified. The result of the calculation served as one of the aspects for economic-managerial evaluation. Based on the financial demands of the implemented functionalities and requirements contained in the Camunda Enterprise license, it was evaluated that purchasing a Camunda Enterprise license would be recommended for example of process automation in university-wide scope.

**Keywords** extension implementation, camunda web-app, filters, forms, managerial evaluation, Camunda Enterprise license, eProcesy, process electronization

---

# Obsah

Úvod	1
<b>1 Úvod do procesního řízení</b>	<b>3</b>
1.1 Podnikový proces . . . . .	3
1.2 Modelování podnikových procesů . . . . .	4
1.3 Notace BPMN . . . . .	4
1.4 Notace DMN . . . . .	8
<b>2 Elektronizace procesů</b>	<b>13</b>
2.1 Procesně řízený systém . . . . .	13
2.2 Nástroj Camunda BPM . . . . .	14
<b>3 Aplikace eProcesy/Camunda na FEL ČVUT</b>	<b>21</b>
3.1 Motivace a cíle aplikace . . . . .	21
3.2 Aktuální stav aplikace . . . . .	21
3.3 Popis aplikace eProcesy/Camunda . . . . .	22
3.4 Problémy aktuální implementace . . . . .	24
3.5 Řešení problematické implementace . . . . .	24
<b>4 Analýza vybraných funkcionalit</b>	<b>27</b>
4.1 Analýza filtrování úkolů . . . . .	27
4.2 Analýza UX funkcionalit . . . . .	28
<b>5 Návrh a realizace</b>	<b>31</b>
5.1 Funkcionalita filtrů . . . . .	31
5.2 UX funkcionality . . . . .	33
<b>6 Ekonomicko-manažerské zhodnocení</b>	<b>35</b>
6.1 Postup zhodnocení . . . . .	35
6.2 Licence Camunda Enterprise . . . . .	36

6.3	Finanční a časová náročnost vlastního řešení . . . . .	39
6.4	Výhody a nevýhody obou variant řešení . . . . .	40
6.5	Zhodnocení a doporučení . . . . .	42
	<b>Závěr</b>	<b>43</b>
	<b>Bibliografie</b>	<b>45</b>
<b>7</b>	<b>Seznam použitých zkratk</b>	<b>47</b>
<b>8</b>	<b>Obsah příloženého CD</b>	<b>49</b>

---

## Seznam obrázků

1.1	Aspekty modelování v notaci DMN [6] . . . . .	9
1.2	Decision Requirement Diagram[6] . . . . .	11
1.3	Rozhodovací tabulka obsahující jednu výstupní komponentu [6] . .	12
2.1	Přehled komponent nástroje Camunda BPM [11] . . . . .	15
2.2	Architektura procesního enginu Camunda BPM [11] . . . . .	16
5.1	Diagram entitních tříd . . . . .	32



---

## Seznam tabulek

6.1	Porovnání obsahů edice Community a Enterprise . . . . .	37
6.2	Určení vhodné edice pro příslušný projekt . . . . .	39
6.3	Časová náročnost implementace rozhraní eProcesy . . . . .	40





---

# Úvod

## Představení tématu

Na ČVUT FEL v Centru znalostního managementu (dále jen CZM) je provozována webová aplikace, jejíž účelem je elektronická podpora studijního oddělení, konkrétně agendy studentských žádostí. Cílem elektronizace je usnadnění práce zaměstnancům školy a vyřizování žádostí studentům. Jelikož vyřízení žádostí má jasně definovaný průchod a podmínky, tak je na něj pohlíženo jako na určitý proces. Z toho důvodu je aplikace postavena na procesním enginu Camunda a každá z žádostí, která se v aplikaci vyskytuje, je vyjádřena a zpracována pomocí procesního modelu v notaci BPMN 2.0.

Aplikace se skládá ze dvou různých uživatelských rozhraní. Jedno rozhraní slouží pro přístup zaměstnanců, kde je využito volně dostupné nadstavby Camunda web-app (dále jen CW). Druhé rozhraní, které si CZM implementovalo samo, je určeno pro přístup studentů a nese název eProcesy. Tato aktuální implementace za pomoci dvou rozhraní není optimální a způsobuje mnoho omezení, která vedou k náročné udržitelnosti a rozšiřitelnosti aplikace. Nadstavba CW je balíček tří webových aplikací: Tasklist, Cockpit a Admin, které jsou dostupné pod edicí Camunda Community. Hlavní příčinou problémů je aplikace Camunda Tasklist.

## Motivace a cíle práce

Motivace k použití rozhraní Camunda Tasklist vznikla na základě dvou faktorů, prvním je filtrování úloh a druhým je možnost využití funkce tzv. embedded formulářů, pomocí kterých jsou vyjádřeny jednotlivé kroky průchodu elektronizovaných žádostí. Motivací pro vznik rozhraní eProcesy byla chybějící funkcionality historie procesů v aplikaci Camunda Tasklist. CZM se mohlo rozhodnout, zda tento problém vyřeší zakoupením licence Camunda Enterprise,

kteřá tuto funkcionalitu přináší, nebo si ji vlastními silami implementuje. Zvolena byla druhá varianta.

Existence a užívání těchto dvou rozhraní se ale jeví jako problematické. Embedded formuláře aplikace Camunda Tasklist sice umožňují implementovat téměř jakoukoli UX funkcionalitu, která se týká formuláře, což je velikou výhodou, ale takto implementované funkcionality nejsou přenositelné a musí se pro každou žádost, která je do aplikace zavedena, implementovat zvlášť. To vede k opakování stejného kusu kódu. Přístup do dvou různých rozhraní není naopak přívětivý z uživatelského hlediska.

Možnosti, jak může CZM tyto problémy odstranit, jsou dvě. První možností je vlastní implementace potřebných funkcionalit, díky čemuž bude možné přestat používat rozhraní Camunda Tasklist. Druhou variantu tvoří zakoupení licence Camunda Enterprise, díky které naopak nebude zapotřebí rozhraní eProcesy. Cílem této práce je uskutečnit první variantu a připravit rozšíření do rozhraní eProcesy o filtrování úkolů a UX funkcionality formulářů, následně zhodnotit časovou a finanční náročnost výsledného řešení a porovnat ji s cenou licence Camunda Enterprise.

# Úvod do procesního řízení

Procesní řízení je odvětví, které se zabývá problematikou efektivitu a konzistenci práce v organizaci. Toto odvětví je založeno na pozorování, že každý produkt, který je vyprodukován určitou společností, ať už se jedná o vládní orgán, neziskovou organizaci nebo podnik, je výsledkem několika po sobě následujících činností, které mezi sebou zauímají určitý vztah a návaznost. Tyto činnosti a vztah mezi nimi následně tvoří něco, čemu se dnes říká podnikový proces. Podnikový proces je klíčový nástroj pro organizaci těchto činností a zlepšení porozumění jejich vzájemnému vztahu.[1, 2]

## 1.1 Podnikový proces

Podnikový proces, někdy také nazývaný jako byznys proces nebo pouze proces, se skládá ze skupiny činností, které jsou prováděny v určité posloupnosti v nějakém organizačním a technickém prostředí. Tyto činnosti společně dohromady vedou ke splnění určitého stanoveného cíle.[1]

Při zanoření hlouběji do definice procesu je možné zjistit, že se skládá nejenom z několika navazujících činností, ale obsahuje také další složky. Součástí procesu jsou také například události. Za událost jsou považovány věci, které se dějí automaticky a odehrávají se v jednom určitém okamžiku a tedy nemají delší dobu trvání. Příkladem takové události může být doručení zboží na pobočku nějaké firmy. Tato událost pak spustí zástup několika dalších činností a úkolů, například kontrolu doručení zboží skladníkem nebo nutnost fakturovat obdrženu objednávku.

Byl zmíněn pojem úkol. Za úkol je možné považovat takovou činnost, která je poměrně jednoduchá a dá se dokončit v relativně krátkém čase. Například zmíněná kontrola doručení zboží je poměrně snadná činnost a je možné ji považovat za úkol. Činností je pak myšlena série několika úkolů.

Další složkou procesu jsou takzvané momenty rozhodnutí. Momenty rozhodnutí nastávají v určitém čase, kdy je potřeba rozhodnout o tom, jak bude

proces probíhat dále. Mají tedy vliv na průběh procesu a takovým momentem může být například chvíle, když zaměstnanec kontrolující právě převzaté zboží musí rozhodnout o tom, zda je zásilka v pořádku a může být uložena či v opačném případě je potřeba ji reklamovat.

Aby mohl být proces uskutečněn a proveden, musí zahrnovat tzv. aktéry. Aktéři jsou vykonavatelé činností v procesu a mohou jimi být fyzické osoby, organizace nebo i software jednající ve jménu právě fyzické osoby nebo organizace. Proces také ve většině případů obsahuje fyzické či nemateriální objekty, kterými mohou být například právě objednané zboží nebo faktura zadaná v informačním systému podniku.

Souhra všech těchto aspektů vede ke splnění předem vytyčeného cíle, jinými slovy na konci každého procesu je určitý výstup. Výstup procesu může být pozitivní či naopak negativní. Pozitivním výstupem vzniká v případě, kdy ukončení procesu přináší hodnotu všem zúčastněným aktérům, například opět ono zmíněné obdržené zboží pro firmu, která si jej objednala a zaplacená faktura pro dodavatele, jenž zboží doručil. Na druhou stranu, pokud by dodavatel spletl objednávku a dodal jiné zboží, než bylo poptáváno, byl by nucen zákazník zboží vrátit a výstup procesu by ani jednomu z aktérů nepřinesl pozitivní výsledek a jednalo by se o negativní výstup.[2]

### 1.2 Modelování podnikových procesů

V procesním inženýrství je kvůli větší srozumitelnosti potřeba, aby procesy byly jakkoli popsány nebo znázorněny. V dnešní době se procesy nejčastěji vyjadřují vizuálně a to právě kvůli větší přehlednosti a pochopitelnosti. Srozumitelně znázorněné procesy jsou potřeba i v případě jejich optimalizace či změny. O této činnosti, kdy jsou procesy vizualizovány, se hovoří často jako o modelování procesů, jehož výsledkem je model procesu.

Model procesu, obzvláště v grafické podobě, popisuje všechny zmíněné složky procesu, jako jsou úkoly, činnosti, události, momenty rozhodnutí, aktéři, objekty související s procesem a souvislosti mezi nimi.[3]

Mezi nejpopulárnější způsoby znázornění procesu patří grafický model, který může nabývat mnoha podob. Nejzákladnější a nejjednodušší podobou grafického modelu je flowchart, mezi sofistikovanější modely patří například notace BPMN nebo DEMO. Na notaci BPMN, konkrétně na její verzi BPMN 2.0, se zakládají zvolené technologie pro tuto práci a podrobně bude popsána v následující sekci.

### 1.3 Notace BPMN

Business Process Model and Notation, ve zkratce BPMN, je mezinárodně standardizovaná notace, která byla vyvinuta konsorciem Object Management

Group (dále jen OMG) za účelem, který byl konsorciem vystižen takto:

*"Primární cíl BPMN je poskytnutí takové notace, která by byla snadno čitelná a srozumitelná pro všechny relevantní uživatele, od business analytika, který vytváří počáteční návrhy procesů, přes technické vývojáře, kteří jsou odpovědní za implementaci technologií podporující vykonávání těchto procesů až po lidi ve společnosti, kteří je budou řídit a dohlížet nad nimi."*

BPMN je v dnešní době jedna z nejvíce používaných notací pro modelování procesů a to nejen z důvodu čitelnosti a srozumitelnosti modelů, ale také možnosti tyto modely automatizovat. Tímto BPMN pokrývá propast mezi počátečními návrhy procesů a jejich následnou implementací, kterou například velice populární formát flowchart, jenž se stal kvůli své čitelnosti a pružnosti inspirací pro notaci BPMN, není schopen překonat.[4]

### 1.3.1 Rozsah BPMN

BPMN podporuje výhradně modelování podnikových procesů, modelování jakékoliv jiné sféry není touto notací aplikovatelné. Mimo rozsah BPMN patří například následující případy:

- definování organizačních struktur a zdrojů,
- modelování funkčních poruch,
- datové a informační modely,
- modelování strategie organizace,
- modely obchodních pravidel či směrnic.[4]

### 1.3.2 Typy modelů BPMN 2.0

Přesto, že rozsah modelování v BPMN je omezený a je soustředěný pouze na procesy, tak umožňuje sdělovat mnoho různých informací široké škále veřejnosti. Dovoluje nám navrhnout několik různých druhů modelů, kde se každý z nich soustředí na jinou oblast procesu. Je rozlišováno několik základních typů v rámci BPMN modelu:

- procesní model,
- kolaborační model,
- konverzační model,
- choreografický model.

K nejpoužívanějším modelům patří procesní a kolaborační modely, které existují již od první verze BPMN. Konverzační model společně s choreografickým byly do notace přidány až od verze BPMN 2.0.[4]

### 1.3.3 Elementy notace BPMN 2.0

Modely v notaci BPMN 2.0 se vytváří za pomoci definovaných elementů. Dle specifikace standardu OMG se elementy dělí do pěti základních kategorií:

1. Flow objects,
2. Data,
3. Connecting objects,
4. Swimlanes,
5. Artifacts.

Příčemž flow objects jsou považovány za základní stavební kameny modelů podnikových procesů, neboť definují chování procesu.

#### 1.3.3.1 Flow objects

Flow objects jsou děleny do dalších tří podkategorií: Events, Activities, Gateways.

**Events** - Události, jak napovídá název, jsou elementy, které nastávají v průběhu procesu a ovlivňují tok procesu. Mohou být výsledkem nějaké aktivity nebo její příčinou. Existují tři typy událostí podle jejich vlivu na tok procesu: startovací, koncové a intermediate. Startovací události jsou ty, které signalizují začátek procesu, koncové jsou takové, které signalizují naopak konec procesu a za intermediate jsou považovány ty, které nastanou kdykoli mezi začátkem a koncem procesu. Intermediate událostí může být například přijmutí zprávy nebo její odeslání.

**Activities** - Aktivita je vyjádření pro jakoukoli činnost prováděnou v průběhu procesu a dělí se do dalších čtyř kategorií: úkol, podproces, transakce, call aktivita. Úkol je atomická aktivita, kterou již nelze rozpadnout na menší části. Podproces je dále dělitelný a obsahuje skupinu úkolů. Využití podprocesu je zejména kvůli přehlednosti modelu. Transakce je speciální případ podprocesu, kde musí dojít ke splnění všech aktivit najednou nebo nedojde k dokončení ani jedné z nich. Call aktivita je globální proces, který lze kdykoli opakovaně spouštět.

**Gateways** - Účel gateways (nebo také bran) je řízení toku procesu. Je to jediný element, který dovoluje na základě nějaké stanovené podmínky rozdělit nebo sloučit toky procesu. Brány se dělí podle jejich typických vlastností do několika skupin: exkluzivní, inkluzivní, paralelní a speciální skupina bran,

kteře jsou založeny na událostech. Exkluzivní brána umožňuje průchod pouze právě jednou větví procesu. Inkluzivní brána umožňuje rozdělit průtok procesu do jedné či více větví. K paralelní bráně se neváže žádná podmínka, ale zaznamenává souběžně vykonávané aktivity a tedy tok procesu se větví. Každá z větví brány založené na události musí vždy začínat spuštěním určité události, lze je také kombinovat s předchozími třemi skupinami. Je možné například využít bránu, která je exkluzivní a zároveň založená na události.[4, 5]

### 1.3.3.2 Data

Typickým požadavkem v procesním modelování je možnost zachytit věci, se kterými je během procesu manipulováno a nelze je zachytit pomocí jiných elementů. Na mysli jsou jakékoli fyzické objekty nebo informace. Tento požadavek je v BPMN možné realizovat skrze element dat, který se dělí na čtyři různé kategorie. Ty jsou následující:

1. **Data objects** - Základní jednotka pro modelování dat, se kterými se v průběhu procesu pracuje. Data objects jsou dočasné a existují tak dlouho, jak dlouho existuje instance procesu jíž jsou součástí.
2. **Data inputs** - Data, které jsou vyžadována pro spuštění procesu a tedy jsou jeho vstupním parametrem. Znárodnují se stejně jako data objects.
3. **Data outputs** - Data, která jsou naopak procesem produkována a tedy jsou jeho výstupem. Taktěž znárodnovány pomocí data objects.
4. **Data stores** - Na rozdíl od data objects se pomocí data stores modelují data, která jsou dostupná kdykoli a existují i před započnutím nebo po ukončení procesu, například data uložená v databázi.[4]

### 1.3.3.3 Connecting objects

Connecting objects zajišťují propojení mezi jednotlivými elementy flow objects nebo jinými informacemi, tak aby vznikla jasná návaznost od začátku do konce procesu. Existují čtyři druhy connecting objects:

1. **Sequence flow** - Udává pořadí, ve kterém mají být zpracovávány flow objects v procesu. Každá sequence flow ma jasně daný zdroj a cíl.
2. **Message flows** - Vyjadřuje tok zpráv v procesu, který nastává mezi různými organizačními jednotkami nebo participanty, kteří jsou v procesu znárodnění jako Pool.
3. **Associations** - Jsou využívány výhradně pro spojení informací a artefaktů s elementy typu flow objects.
4. **Data associations** - Slouží k vyjádření toku dat mezi jednotlivými aktivitami nebo celými procesy.[4]

### 1.3.3.4 Swimlanes

Swimlanes jsou prvky, které umožňují rozlišit jednotlivé účastníky a odpovědnosti v modelu procesu. Díky tomu je možné seskupovat jednotlivé aktivity procesu do mezi sebou rozdílných funkčních celků. Swimlanes se dělí na dva druhy:

1. **Pool** - Objekt ve tvaru obdélníku, který ohraničuje proces a reprezentuje nějakého účastníka v rámci modelu procesu. Účastníkem může být samostatná osoba, ale i celá organizace nebo nějaké oddělení v rámci organizace. Hranice mezi jednotlivými pooly mohou být překročeny pouze pomocí message flows.
2. **Lanes** - Seskupují aktivity v rámci jednoho poolu a rozdělují je tak na vícero skupin podle kompetencí účastníků dotyčného procesu v konkrétním poolu.[4]

### 1.3.3.5 Artifacts

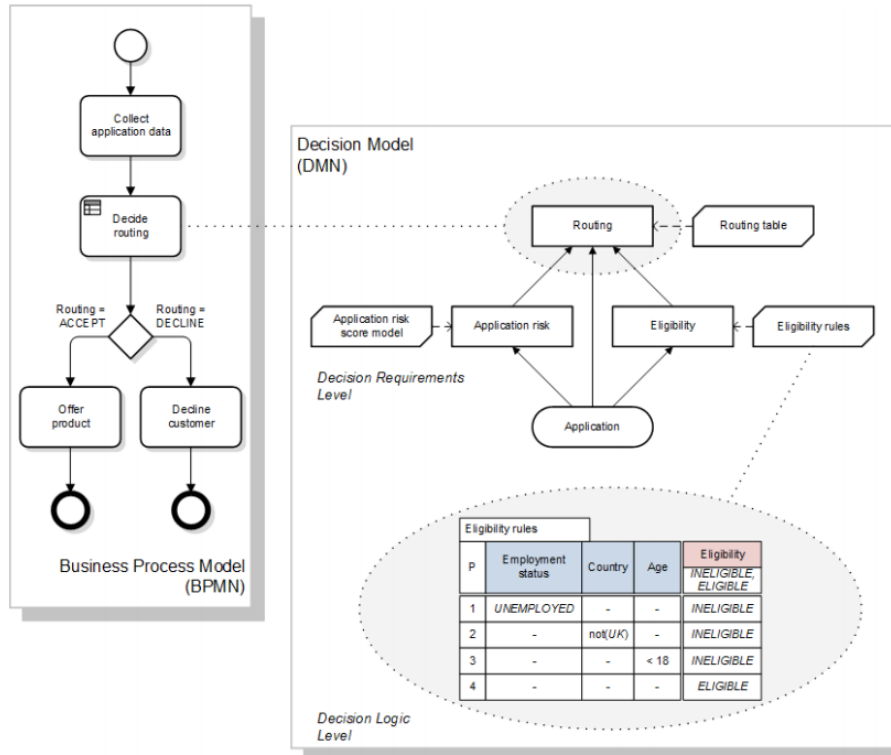
Artifacts přináší do modelu podrobnější informace o procesu, ale nemají žádný vliv na jeho průběh, pouze zlepšují čitelnost modelu a zvyšují úroveň detailu v procesu. Artifacts mohou být do modelu zaneseny jako Text Annotation nebo pomocí Group. Text Annotation v sobě nese dodatečné informace a je spojována s plovoucími objekty pomocí asociace. Group je pouze logické seskupení aktivit stejné kategorie.[4]

## 1.4 Notace DMN

Notace DMN (Decision Model and Notation) byla stejně jako notace BPMN vyvinuta konsorciem Object Management Group. Účelem notace je poskytnout takové konstrukty, které umožní modelovat rozhodování organizací, aby ona rozhodnutí mohla být snadno znázorněna v diagramech, přesně definována byznys analytiky a v případě potřeby automatizována. Notace byla specifikována tak, aby bylo možné DMN modely provázat s procesními modely BPMN. V procesních modelech jsou definovány úkoly, ve kterých je potřeba učinit rozhodnutí. Definice těchto rozhodnutí jsou znázorňovány pomocí Decision Requirements Diagram (DRD). DRD definují která rozhodnutí mají být učiněna, vztahy mezi nimi a nutné požadavky pro jejich uskutečnění. Samotná rozhodnutí jsou pak v patřičném detailu definována pomocí rozhodovací logiky, která umožní, aby rozhodování mohlo být automatizováno a validováno. Rozhodovací logika a DRD tak mohou společně poskytnout kompletní model rozhodování a doplnit modely podnikových procesů o detailní specifikaci úkolů, ve kterých je potřeba učinit příslušná rozhodnutí. Obrázek 1.3. graficky znázorňuje vztah těchto tří popasných aspektů.



Obrázek 1.1: Aspekty modelování v notaci DMN [6]



### 1.4.1 Rozsah DMN

Modelování rozhodnutí je prováděno nejčastěji byznys analytiky za účelem stanovení lepšího porozumění operativním rozhodnutím využívaných v každodenních procesech v jakýchkoliv organizacích a podnicích. V tomto kontextu lze rozlišit několik použití notace DMN:

- modelování lidského rozhodování,
- modelování požadavků pro automatizované rozhodování,
- implementace automatizovaného rozhodování.

V prvním případě jde zejména o popisné modelování za účelem větší srozumitelnosti. Takto modelované rozhodnutí může být rozpadnuto na několik základních, mezi sebou závislých, rozhodnutí pomocí DRD využívající komponenty konceptu Decision Requirements Graph (DRG). Za pomocí těchto konceptů můžeme graficky vyjádřit řízení organizace, její politiku nebo zdroje a znalosti, které jsou potřeba pro vykonávání příslušných rozhodnutí.

V druhém případě se jedná téměř o totožné využití s tím rozdílem, že modelování požadavků pro automatizované rozhodování je spíše normativní než

popisné. Klade se větší důraz na rozhodovací logiku. Pokud vyžadujeme plnou automatizaci rozhodování, tak by rozhodovací logika měla být kompletně zpracována a měla by být schopná poskytnout rozhodnutí za jakýchkoli okolností, v případě automatizovaných rozhodnutí se myslí spíše jakákoliv sada vstupních dat. Avšak běžnější je automatizace částečná, což je kombinace aktivit procesu, které jsou automatizovány a těch, které je potřeba vykonat za pomoci lidského přičinění.

Třetí a poslední případ není potřeba detailněji rozepisovat. V případě, že jsou plně specifikovány všechna rozhodnutí, včetně jejich požadavků, je možné modely těchto rozhodnutí automatizovat a spouštět.[6]

### 1.4.2 Vztah DMN a BPMN

Notace BPMN umožňuje do jisté míry modelovat rozhodnutí pomocí gateways. V realitě mohou podniky a organizace činit velice složité rozhodnutí, která jsou závislá na mnoha vnějších podmínkách, kontrolách nebo na dalších rozhodnutích. V takových případech by modelování rozhodnutí v BPMN vedlo k velice složitým a těžko udržitelným modelům. Z toho důvodu zahrnuje BPMN druh aktivity nazývaný Business rule task, který vyjadřuje rozhodnutí, které je potřeba stanovit a je možné skrze něj odkázat na model rozhodování v notaci DMN. Další přínos notace DMN je, že umožňuje, aby rozhodnutí bylo plně automatizované a mohlo být učiněno bez zásahu uživatele.

### 1.4.3 Základní koncepty DMN

Notaci DMN je možné rozdělit do dvou základních konceptů:

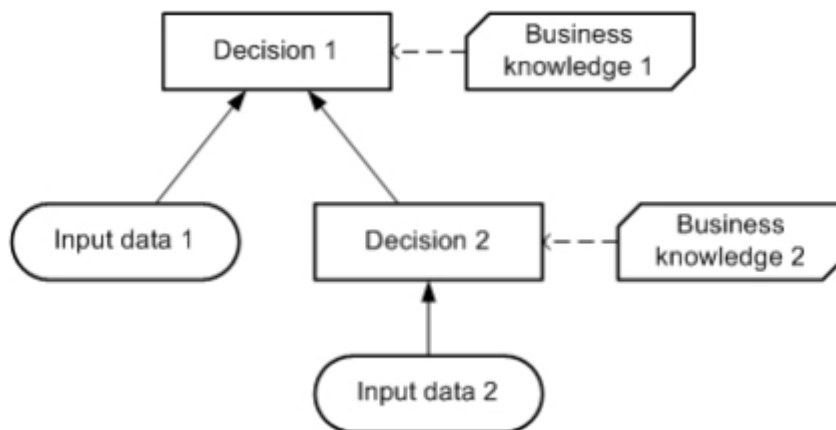
1. Decision requirement level,
2. Decision logic level.

Decision requirement level je úroveň pohledu na DRD, která je zaměřena zejména na požadavky, které jsou potřeba k učinění jednotlivých rozhodnutí. Rozhodnutí je zde chápáno ve smyslu určení výstupní hodnoty (zvolená možnost) z řady vstupních hodnot za pomoci definující logiky, která určuje podmínky, jak se příslušný výstup z konkrétních vstupů určí. Obsah této logiky může být zapouzdřen do tzv. business knowledge models (v překladu modely obchodních znalostí), které utváří know-how organizace ve formě organizačních pravidel, analytických modelů či jiných formalismů. Využití modelů obchodních znalostí není nutností, je to otázka stylu a metodologie.

K učinění rozhodnutí je zapotřebí vstupních dat, které jsou vyžadovány v takovém pořadí, v jakém mají být jednotlivá rozhodnutí uskutečněna, jelikož tyto vstupní data mohou být výstupními daty jiných rozhodnutí. Rozhodnutí tak můžeme znázornit pomocí Decision Requirement Graph, což přináší modelovací konstrukty jako jsou decision, business knowledge, input data,

knowledge source a dále spojovací prvky association, knowledge requirement, authority requirement, information requirement nebo annotation. Za pomoci těchto konstruktů je vytvářen DRD, který znázorňuje jak jednotlivá rozhodnutí na sebe navazují, jak závisí na vstupních datech nebo na modelech obchodních znalostí. Níže máme jednoduchou ukázkou DRD.

Obrázek 1.2: Decision Requirement Diagram[6]



Použití konceptu Decision logic level umožňuje specifikovat komponenty DRD ve větším detailu a zachytit tak kompletní sadu obchodních pravidel a výpočtů. Díky tomu je v případě potřeby možné plně automatizovat rozhodování. Detailnější popis těchto komponent je možné vyjádřit zejména za pomoci výrazů v jazyce FEEL (Friendly Enough Expression Language), dále pomocí tzv. Decision tables (rozhodovací tabulky) nebo jejich kombinací.

#### 1.4.4 Rozhodovací tabulka

Rozhodovací tabulka je možnost zpracování rozhodovací logiky v DRD modelu. Jak napovídá název, jedná se o tabulkovou reprezentaci pravidel, které na základě sady vstupních hodnot určují hodnoty výstupní. Tabulka se skládá z několika prvků:

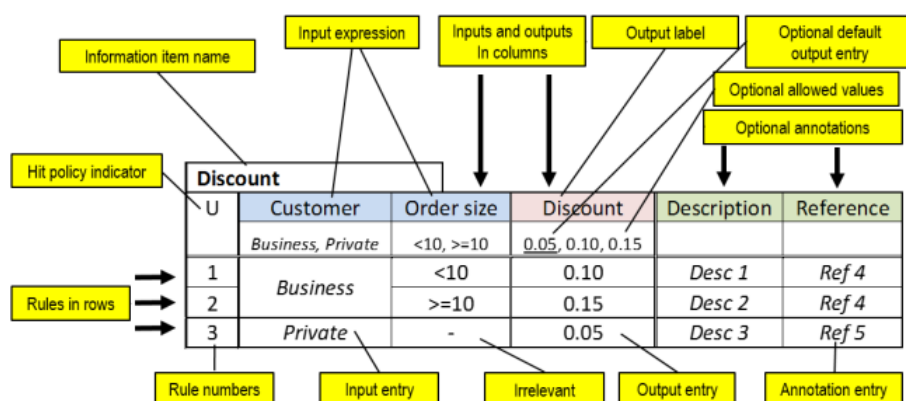
- Název rozhodovacího prvku DRD (Information item name).
- Seznam vstupních klauzulí (Input clauses) - Každá ze vstupních klauzulí je tvořena výrazem a seznamem hodnot, kterých mohou nabývat vstupní hodnoty.
- Seznam výstupních klauzulí (Output clauses) - Každá výstupní klauzule je tvořena názvem a seznamem hodnot, kterých mohou nabývat výstupní komponenty.

## 1. ÚVOD DO PROCESNÍHO ŘÍZENÍ

- Sada výstupních komponent (Output components) - V případě, že výstupní klauzule obsahuje pouze jednu výstupní komponentu, tak ona komponenta nemusí být pojmenovaná a je tvořena pouze výstupní hodnotou. Pokud je výstupní klauzule rozdělena na více komponent, musí mít každá z příslušných komponent své unikátní jméno a hodnotu. Výstupní komponenty určují hodnoty pro každé ze stanovených pravidel.
- Seznam anotací (Annotations) - Každá z anotací má vlastní název a obsahuje vlastní hodnotu pro každé pravidlo. Anotace nehrají roli při vyhodnocování pravidla.
- Seznam pravidel (Rules) - Pravidla jsou v tabulce orientované buď po řádcích nebo po sloupcích. Každé pravidlo je tvořeno specifickými vstupními, výstupními hodnotami a anotacemi. Pokud jsou pravidla zapsána po řádcích, sloupce tabulky jsou tvořeny vstupními a výstupními klauzulemi a naopak.

Pravidla jsou velice jednoduše znázorněná uspořádáním v buňkách tabulky, což přináší výhody snadné čitelnosti a ověřování. Pravidla jsou vyhodnocována stejně jako if/else/then podmínky a mezi jednotlivými vstupními hodnotami je vždy využito logiky AND. Pokud se tedy hodnoty všech vstupních výrazů rovnají předepsaným hodnotám v tabulce, pravidlo je splněno a výstupem bude hodnota výstupní komponenty, která je vázaná k dotčenému pravidlu. Například tedy při pohledu na pravidlo označené číslem 1 v tabulce na obrázku 1.3 je možné vidět, že pokud výraz Customer bude nabývat hodnoty Business a výraz OrderSize bude nabývat hodnoty menší než 10, bude potom výstupní komponenta Discount nabývat hodnoty 0.10.

Obrázek 1.3: Rozhodovací tabulka obsahující jednu výstupní komponentu [6]



---

## Elektronizace procesů

Podnikové procesy bývají často podporovány informačními systémy. Procesní řízení klade důraz na flexibilitu těchto informačních systémů. Podnikové procesy jsou velice proměnlivé a podstatou systémů pro podporu procesů je schopnost se změnám procesu přizpůsobovat.[7] O podpoře procesů informačním systémem je možné mluvit také jako o automatizaci procesů. Proces je nazván automatizovaným ve chvíli, kdy je jeho jakkoli velká část automatizována. Může jít o automatizaci samostatných činností prováděných v rámci samostatné aktivity procesu či celého procesu od začátku do konce. Automatizovaný proces lze také nazvat jako workflow.[2] Pojem workflow byl definován již v 90. letech 20. století institucí Workflow Management Management Coalition:

*”Workflow znamená automatizaci celého nebo části podnikového procesu, během kterého jsou dokument, informace nebo úkoly předávány od jednoho účastníka procesu k druhému podle sady procedurálních pravidel.”*[8]

Jinými slovy lze workflow popsat také jako:

*”Instance procesu či pracovní tok, který probíhá v konkrétním čase, za konkrétních okolností a s konkrétními aktéry procesu v konkrétní variantě.”*[7]

### 2.1 Procesně řízený systém

Dle Václava Řepy[7] musí informační systém vyhovět několika nárokům pro úspěšnou podporu procesního řízení v organizaci:

- systémová podpora veškerých činností procesu,
- správa, monitorování a vyhodnocování probíhajícího workflow procesu,

- podpora veškerých omezení a zákonitosti proces určených příslušným podnikem,
- přizpůsobení jakékoliv koncepční a operativní změně procesu.

Nejčastěji se objevují takové nástroje, které podporují automatizování procesů modelovaných v notaci BPMN 2.0. V dnešní době takových nástrojů existuje již mnoho, například nástroj IBM BPM, Camunda BPM, Activiti, Bizagi BPM suite nebo například ActiveVOS. V této práci bude podrobněji popsán pouze nástroj Camunda BPM, protože projekt, kterého se týká tato práce je postaven na této technologii.

## 2.2 Nástroj Camunda BPM

Nástroj Camunda BPM je open-source projektem společnosti Camunda, která byla v roce 2008 založena Jakobem Freundem a Berndem Rückerem jako konzultační firma v oblasti BPM (Business Process Management).[9, 10]

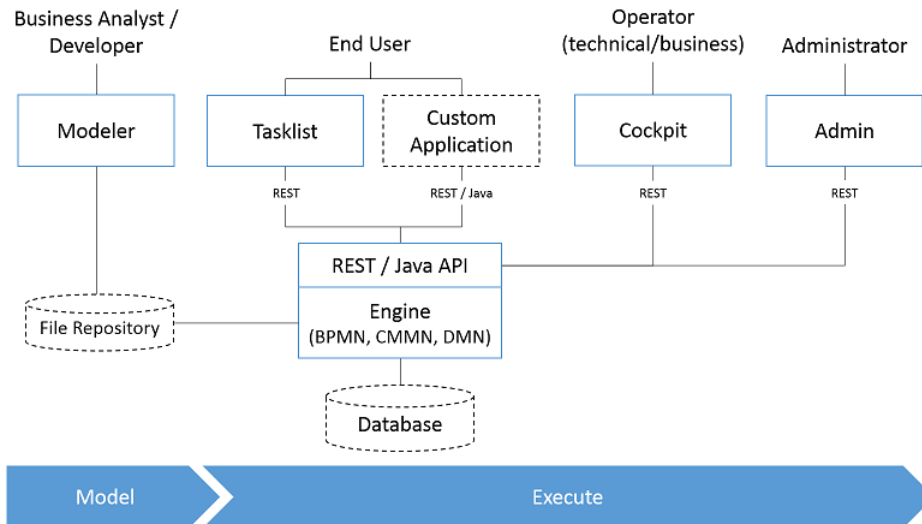
Camunda BPM je platforma zaměřená na automatizaci procesů a jejich workflow. Platformu tvoří zejména procesní engine napsaný v jazyce Java, který umožňuje implementaci a automatizaci procesů modelovaných v BPMN 2.0 a zajišťuje jejich podporu a monitorování v reálném čase. Kdykoli bude zmíněn název Camunda, bude na mysli především právě procesní engine. Camunda BPM také kromě BPMN 2.0 umožňuje modelování v notacích CMMN 1.1, která se využívá k řízení událostí a DMN 1.1., která je využita k řízení rozhodnutí. Součástí celé platformy je dále desktopová aplikace Camunda Modeler určená pro modelování v BPMN, CMMN a DMN. V nabídce je také několik webových aplikací, kde každá z nich má jiné využití, které si představíme v několika podkapitolách níže. Mezi poskytované webové aplikace patří: Camunda Tasklist, Camunda Cockpit a Camunda Admin. Tyto tři webové aplikace jsou distribuovány jako jeden balíček pod názvem Camunda web-app, jenž byl zmíněn již v úvodu této práce.[11]

### 2.2.1 Architektura

Camunda BPM je napsána se záměrem poskytnout vhodný nástroj, jak pro vývojáře, kteří se zabývají jazykem Java, tak i pro ty, kteří preferují jiný programovací jazyk. Z toho důvodu jsou hlavní komponenty napsány v Javě, tak aby poskytované nástroje podporovali modelování, implementování a spouštění procesů v JVM (Java Virtual Machine) a pro přístup k procesnímu enginu poskytují dvě rozhraní.

Prvním rozhraním je Java API a druhým je REST API, díky kterému jsou vývojáři, kteří neprogramují v Javě, schopni vytvářet aplikace připojené ke vzdálenému procesnímu enginu. Obrázek 2.1 demonstruje grafický přehled hlavních komponent celé platformy společně s jejich typickými uživateli.

Obrázek 2.1: Přehled komponent nástroje Camunda BPM [11]



### 2.2.2 Procesní engine

Procesní engine je Java knihovna zodpovědná za spouštění BPMN 2.0 procesů, CMMN 1.1 událostí a DMN 1.1 rozhodnutí. Jádrem engineu je lightweight POJO a z důvodu perzistence využívá relační databázi. Procesy jsou engineu předávány pomocí takzvaných procesních definic, které jsou implementovány pomocí modelů v BPMN 2.0 a jsou engineu předávány v podobě XML souboru. V případě, že chceme procesy spustit, tak jsou engineem z jednotlivých procesních definic vytvářeny procesní instance, které reprezentují běh jednoho konkrétního workflow. Vztah procesní definice a procesní instance se dá připodobnit ke vztahu objektům a tříd v objektově orientovaném programování, kde procesní instance hraje roli objektů a procesní definice roli tříd. Díky engineu máme přístup také k procesním proměnným, což mohou být proměnné týkající se konkrétního spuštěného případu (procesní instance) nebo týkající se procesu samotného, například datum spuštění procesu nebo jméno uživatele, který proces spustil.

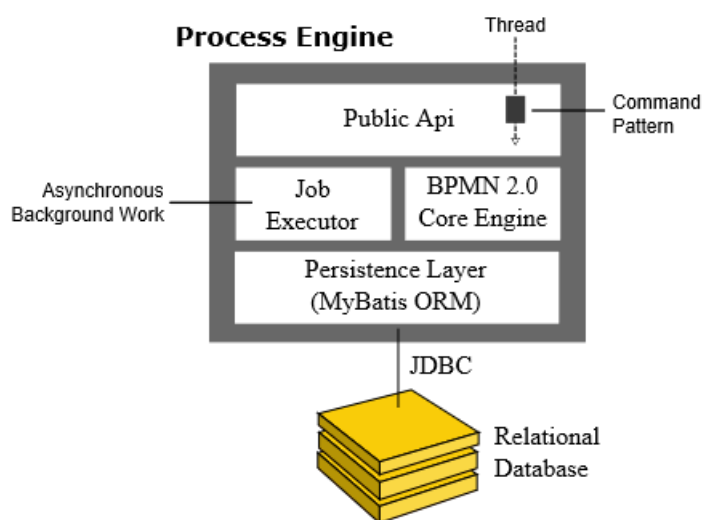
Camunda umožňuje také verzování procesů. Do databáze jsou ukládány informace o tom v jaké verzi byly procesní instance spuštěny a v případě nasazení nové verze příslušného procesu poběží instance stále na verzi, na které byly původně spuštěny. Nově vzniklé instance už ale poběží na nejnovější verzi procesu.

Camunda BPM je flexibilní framework, který je možné zaintegrovat několika různými způsoby. Může být integrován do samotné Java aplikace jako knihovna, je také možné jej spustit uvnitř runtime kontejneru (například Servlet kontejner nebo aplikační server), engine je pak poskytován v podobě kontejnerové služby a je přístupný všem aplikacím, které jsou v daném kontejneru

nasazený. Dále v neposlední řadě je možné engine implementovat jako samostatný procesní server, kde je engine zprostředkováván pomocí síťové služby. Všechny aplikace v rámci příslušné sítě mohou s procesním serverem komunikovat pomocí komunikačních kanálů. Nejjednodušší cesta je například srkze předpřipravené REST API.

Architektura procesního engine je tvořena několika hlavními komponentami, jejich složení je prezentováno na obrázku 2.2.

Obrázek 2.2: Architektura procesního engine Camunda BPM [11]



- **Public API** - servisně orientované API, které umožňuje aplikacím interagovat s procesním enginem.
- **BPMN 2.0 Core Engine** - jádro procesního engine, které poskytuje transformaci BPMN 2.0 XML modelu do Java objektů a implementaci sady behaviorálních konstruktů BPMN 2.0, jako jsou například Gateways.
- **Job Executor** - zodpovědný za asynchronní spuštění a průběh procesů na pozadí. Například časovač.
- **The Persistence Layer** - vrstva, která zodpovídá za setrvání stavů procesů v relační databázi.

### 2.2.3 Camunda Modeler

Camunda Modeler je desktopová aplikace sloužící k modelování a úpravě diagramů v notaci BPMN, CMMN a DMN. Aplikace je velice intuitivní a je uzpů-



sobena k tomu, aby byznys analytici i vývojáři mohli pracovat v rámci jednoho modelu procesu v tom stejném nástroji. Umožňuje ukládat procesní modely ve formě XML souboru přímo do lokálního souborového systému a kromě modelování se v aplikaci konkretizují a nastavují vlastnosti procesu pro jeho spuštění.[11]

### 2.2.4 Camunda REST API

Hlavním přínosem Camunda REST API je možnost zavedení samostatně fungujícího procesního serveru, ke kterému mohou přistupovat vzdálené aplikace. Samozřejmě vzdálené připojení k procesnímu serveru může být implementováno i pomocí webových služeb SOAP nebo JMS, ale ty nejsou předem připraveny a uživatel si je musí naimplementovat sám. Vestavěné Camunda REST API tedy tvoří nejjednodušší cestu, jak se k vzdálenému procesnímu serveru připojit.[11]

### 2.2.5 Camunda Tasklist

Tato aplikace je určena pro správu procesních aktivit, které mají být uskutečněny lidským přičiněním a tedy účastníky daného procesu. Aplikace umožňuje účastníkům procesu spravovat uživatelské úkoly a aktivity procesů pomocí filtrů, formulářů a delegování. Jednotlivým úkolům lze také nastavit datum kontroly, které poslouží jako připomínka či datum splatnosti vyjadřující termín, kdy má být daný úkol splněn. Skrze Tasklist je také možné spouštět procesy. Pro tuto práci jsou stěžejní zejména funkcionality filtrů a embedded formulářů. Obě funkcionality budou podrobněji představeny v následujících dvou podkapitolách.[11]

#### 2.2.5.1 Funkcionalita filtrů

V Tasklistu je možné vytvářet filtry úloh, díky kterým je možné si zobrazit seznam úkolů dle mnoha různých preferencí. Existují čtyři kategorie, podle kterých lze filtry vytvářet a spravovat.

- **General** - nabízí několik základních vlastností, které je možné nastavit. Nastavit lze libovolné jméno, popis a také barva filtru. Dále je možné nastavit, aby se filtr automaticky aktualizoval nebo priorita, podle které se určí pořadí nastavených filtrů.
- **Permissions** - umožňuje nastavit oprávnění pro jednotlivé uživatele nebo celé skupiny uživatelů, zda mohou daný filtr vidět a využívat.
- **Criteria** - pomocí kritérií je určeno, které úkoly se budou v daném filtru zobrazovat. Musí být definován klíč a hodnota, kde jako klíč je zvolena některý z procesních atributů. Existuje mnoho klíčů, které je možné vybírat z několika různých kategorií. Seznam kategorií a do nich

spadajících atributů je následující: Process Instance (ID, Business Key), Process Definition (ID, Key, Name), Case Instance (ID, Business Key), Case Definition (ID, Key, Name), Other (Process Instance state, Activity instance ID, Execution ID), User/Group (Assignee, Owner, Candidate User or Group, Involved user, Unassigned, Delegation State), Task (Definition Key, Name, Description, Priority) and Dates (Created date, Due date, Follow up date).

- **Variables** - nastavení zobrazení proměnných procesu v přehledu jednotlivých úkolů v seznamu příslušného filtru.

Příkladným nastavením filtru může být filtrování úkolů dle přiřazeného řešitele nebo naopak filtrování úkolů, které ještě nikomu přiřazeny nebyly.

### 2.2.5.2 Formuláře

Existuje více možných způsobů, jak implementovat formulář v Tasklistu. Pro implementaci formuláře je potřeba propojit zdroj formuláře s elementem procesu, který příslušný formulář využívá. V BPMN 2.0 existují dva elementy, které mohou využívat formulářů, těmi jsou Start Event a User Task. Podporované způsoby implementace formulářů v Tasklistu:

- **Embedded Task Forms** - formuláře založené na jazyce HTML vkládané do Tasklistu.
- **Generated Task Forms** - formuláře generované na základě XML metadat procesního modelu v BPMN 2.0.
- **External Task Forms** - přesměrování uživatele na User Task do jiné, externí aplikace.
- **Generic Task Forms** - formulář, který je zobrazen v případě, že neexistuje žádná z předchozích variant. Umožňuje editovat podobu formuláře v průběhu spuštěné instance procesu.

Pro tuto práci jsou důležité pouze Embedded formuláře. V následující kapitole budou představeny podrobněji.

### 2.2.5.3 Embedded formuláře

Embedded formuláře jsou založeny na jazyce HTML a JavaScript, které mohou být zobrazeny přímo v aplikaci Tasklist. Pro přidání nového formuláře stačí vytvořit HTML soubor a odkazovat s k němu přes User Task v procesním modelu, což je procesem definovaný úkol, který má být zpracován uživatelem. Na HTML soubor formuláře se lze odkazovat dvěma způsoby.

Prvním způsobem je přidat soubor do vývojového projektu. Soubor formuláře bude zabalen do WAR archivu a při běhu aplikace odtud bude načítán.

Druhá možnost je přidat soubor do procesního archivu, což znamená, že bude uložen v databázi Camundy, odkud jej bude možno načíst. Toto má výhodu, že mohou být formuláře verzovány společně s procesy, kterých se formulář týká.

### 2.2.6 Camunda Cockpit

Camunda Cockpit zajišťuje monitorování právě běžících nebo již ukončených procesních instancí. Toto rozhraní poskytuje operace, pomocí kterých lze vyhledávat konkrétní procesní instance, sledovat jejich stav a v případě selhání je opravovat.[11]

### 2.2.7 Camunda Admin

Aplikace Camunda Admin je určená pro správu jednotlivých uživatelů, uživatelských skupin a autorizací. Lze napojit na LDAP.[11]



---

# Aplikace eProcesy/Camunda na FEL ČVUT

Aplikace eProcesy/Camunda byla začátkem zimního semestru 2019/2020 uvedena do pilotního provozu na FEL ČVUT. Aplikace je určena pro elektronickou podporu studentských žádostí na studijním oddělení.

Žádosti, které studenti mohou podávat do nedávna existovali pouze v papírové podobě. K tomu, aby mohly být žádosti v pořádku podané, vyřízené a uzavřené, je často potřeba mnoho vyjádření a podpisů od různých osob, například od vedoucích kateder, studijních referentek nebo proděkanů fakulty. Student má toto ve své režii a je povinen tyto kroky zajistit. To do nedávna znamenalo, že musel všechny osoby relevantní k dotyčné žádosti postupně obcházet. Centrum znalostního managementu má za cíl za pomoci této aplikace ukázat, že to jde i jednodušším a modernějším způsobem.

## 3.1 Motivace a cíle aplikace

Z úvodu této kapitoly vyplývá jasná motivace i cíl projektu aplikace. Motivací je jednoznačně snaha o modernizaci, usnadnění problémů studentům spojených se studiem a zefektivnění práce zejména studijním referentkám, ale také dalším zaměstnancům fakulty. Cílem projektu je uvést aplikaci do standardního provozu a postupně plně elektronizovat všechny dostupné žádosti. Vedlejším cílem je vyzkoušet technologii Camunda BPM a obeznámit se s tím, jaké přináší možnosti a omezení.

## 3.2 Aktuální stav aplikace

Student má možnost podat patnáct různých žádostí. Aplikace je momentálně ve stavu, kdy podporuje jedenáct z nich, na dvou dalších se pracuje a dvě ještě vůbec nejsou zmapovány. Žádosti, které jsou nyní aplikací podpořeny se

dělí na dva typy. Prvním typem jsou žádosti, jejichž průběh vyřízení je téměř plně elektronizován a druhým typem jsou ty žádosti, které poskytují pouze vygenerování předvyplněného formuláře.

V aplikaci jsou implementovány dvě různá uživatelská rozhraní. Prvním je rozhraní, které nese název eProcesy a je určeno pro přístup studentů. Z rozhraní eProcesy mohou studenti vytvářet nové žádosti a plnit úkoly, které jim jsou přiřazeny, součástí je přehled momentálně aktivních žádostí a historie již uzavřených.

V druhém rozhraní pracují zaměstnanci školy, zejména studijní referentky. Umožňuje spravovat úkoly, které je potřeba v průběhu vyřízení žádosti plnit.

Jak již bylo řečeno v úvodu této práce, tato implementace je v mnohém nevyhovující. Konkrétní problémy, které jsou spojeny se společným využitím těchto dvou rozhraní budou popsány v následující sekci.[12, 13]

## 3.3 Popis aplikace eProcesy/Camunda

V této sekci bude nejdříve popsán způsob, jakým se s aplikací pracuje, následně je představeno technické zpracování a architektura aplikace a v poslední řadě budou vysvětleny problémy a nevýhody, které jsou výsledkem aktuálního stavu implementace.

### 3.3.1 Práce s aplikací

Vyřízení každé z žádostí, které jsou aplikací podporovány, je reprezentováno pomocí modelu procesu v BPMN 2.0. Model procesu žádosti jasně definuje možné průběhy při vyřízení žádosti a obsahuje úkoly, které je potřeba splnit k úspěšnému zpracování žádosti. Jednotlivé úkoly jsou přiřazovány příslušným zodpovědným osobám za pomoci rozhodovacích DMN tabulek, které se váží ke konkrétním procesům. Úkoly, za které je zodpovědný student, jsou spravovány pomocí rozhraní eProcesy a úkoly, které jsou určeny studijním referentkám nebo jiným zaměstnancům fakulty jsou naopak spravovány v rozhraní Camunda.

Každý proces začíná iniciativou studenta. Pokud má student zájem podat žádost, tak má možnost spustit novou instanci procesu žádosti z rozhraní eProcesy, kde se mu okamžitě po vyvolání žádosti otevře první úkol, ve kterém musí vyplnit své osobní údaje, důvod žádosti a přiložit relevantní dokumenty, pokud je tak konkrétním procesem definováno. Po splnění určených úkolů má student zpravidla několik možností jak s žádostí dále naložit. V případě, že ztratil o podání žádosti zájem, tak může proces zrušit, v opačné situaci existuje několik variant, jak pokračovat dále. Pokud se jedná o žádost, která je téměř plně elektronizovaná, tak ji může nechat odeslat na studijní oddělení, kde si ji převezmou na starost studijní referentky nebo jiní, předem určení zaměstnanci fakulty. Pokud je žádost podpořena aplikací pouze formou předvyplněného vygenerovaného formuláře, tak má student možnost si daný formulář stáhnout

a tím následně celou žádost v aplikaci uzavřít. Žádost jako taková ale uzavřena není a student všechny zbývající náležitosti musí obstarat mimo prostředí aplikace.

Ze strany studijních referentek nebo jiných zaměstnanců, kteří jsou součástí procesu, vypadá vyřízení žádosti elektronickou formou následovně. V momentě, kdy student odešle žádost na studijní oddělení, přikročí se dle procesní definice k další aktivitě, kterou mají zpravidla na starosti někteří ze zaměstnanců fakulty. V aplikaci Camunda vznikne nová úloha, kterou je možné spravovat a tedy ji buď zpracovat nebo delegovat. Dotyčná osoba, které se aktivita týká se skrze rozhraní Camundy přihlásí do aplikace, kde uvidí nově vzniknutou úlohu a má možnost ji spravovat.

Proces končí ve chvíli, kdy se uzavře poslední aktivita. Zpravidla je to kontrola vyplněných informací v průběhu celé žádosti a vytisknutí formuláře žádosti studijní referentkou. Vytisknutou žádost následně podepíše, orazítkují a založí do složky studenta.[12, 13]

#### 3.3.2 Rozhraní eProcesy

Prostředí eProcesy, do kterého přistupují zejména studenti FEL, nabízí několik funkcionalit. Hlavní funkcí je spouštět nové instance procesů jednotlivých žádostí a plnit úkoly, které jsou ve studentově odpovědnosti. Studentům dále nabízí přehled o všech procesech, které kdy v aplikaci spustili a přehled o úkolech, které mají na starosti. Pro zaměstnance FEL nabízí historii všech již uzavřených nebo právě běžících procesů, respektive žádostí studentů. Pro administrátory existuje navíc možnost impersonace, která umožňuje přihlásit se za kteréhokoli uživatele.

Úkoly jsou zde reprezentovány takzvanými automatickými formuláři. To jsou elektronické formuláře, které jsou aplikací automaticky generovány podle nastavených tzv. Form Fields v definici procesu. Podle konfigurace příslušného Form Fieldu je vygenerováno výsledné pole formuláře.

Rozhraní eProcesy je implementováno v jazyce Java EE ve verzi 7 a je integrováno s procesním enginem Camundy pomocí vestavěného Java api, které Camunda nabízí. Pomocí Java api se získávají a ukládají veškeré informace týkající se procesů a jejich instancí.[12, 13]

#### 3.3.3 Rozhraní Camunda

Rozhraní Camunda je rozděleno do tří částí. První částí je Tasklist, který je určen uživatelům aplikace, zbylé dvě části Cockpit a Admin jsou pouze pro administrátory a správce aplikace.

Tasklist obdobně jako eProcesy umožňuje spravovat jednotlivé úkoly spuštěných procesů, je to zároveň jediné a hlavní využití této části aplikace. Tasklist uživatelům poskytuje přehled všech úkolů, které je potřeba zpracovat a možnost je filtrovat podle mnoha různých atributů, viz kapitola 2.2.5.1. Taktéž

jako v eProcesech jsou úkoly v Tasklistu reprezentovány pomocí elektronických formulářů. Zde se ale formuláře eProcesů a Tasklistu liší v implementaci. Zatímco formuláře v eProcesech jsou automaticky generovány dle procesní definice, v Tasklistu jsou implementovány pomocí embedded formulářů.[12, 13] Embedded formuláře byly představeny v kapitole 2.2.5.3.

## 3.4 Problémy aktuální implementace

Problémy aktuální implementace tkví v několika hlavních bodech:

- dvě odlišná rozhraní,
- nedostatek vývojářské kontroly nad rozhraním Camunda Tasklist,
- využití embedded formulářů.

Dvě odlišná rozhraní jsou uživatelsky velice nepřívětivá. Studijní referentky jsou nuceny pracovat s dvěma různými rozhraními. Pro splnění úkolů spojených s žádostmi je potřeba pracovat s Camunda Tasklist a pro zkoumání historických žádostí, ale i aktuálně probíhajících, je nutné se přihlásit do rozhraní eProcesy. To může být pro práci uživatele velice rušivé.

Vývojářský tým aplikace eProcesy/Camunda zvolil řešení v podobě vlastního rozhraní eProcesy zejména proto, že nebyl znalý technologií, ve kterých byla aplikace Camunda Tasklist vyvinuta. Z toho důvodu nemá CZM možnost kontroly nad tímto rozhraním, i přesto, že existuje pod licencí Apache License 2.0 a lze jej upravovat dle vlastních potřeb. Důsledkem je, že CZM nemůže pohotově vyhovět případným uživatelským nárokům na toto rozhraní, aspoň co se správy úkolů týče.

Embedded formuláře přinášejí sice velkou variabilitu v zobrazení User Tasků procesů, ale při velkém množství procesů jsou velice těžko udržitelné. Jednotlivé procesy žádostí se mezi sebou ve svém workflow příliš neliší, ale informační a dokumentační nároky mají častokrát velice odlišné. Z toho důvodu je mnohokrát potřeba pro každý proces a pro každý jeho úkol implementovat nový embedded formulář, který není znovupoužitelný. Pro každý takový formulář je následně potřeba implementovat UX funkcionality, které jsou nápomocny k jejich správnému vyplnění uživateli. Z důvodu nedostatečné vývojářské kontroly nad rozhraním Camunda Tasklist je nutné tyto UX funkcionality implementovat pomocí jazyka Javascript a odpovídající kód implementovat do každého formuláře zvlášť. V případě potřeby je nutné provést změny tohoto kódu ve všech formulářích, kterých se to týká.[12, 13]

## 3.5 Řešení problematické implementace

Problematický stav aplikace lze řešit dvěma způsoby. Implementací funkcionalit filtrů a formulářů rozhraní Camunda Tasklist do rozhraní eProcesy nebo



zakoupením licence Camunda Enterprise. První možnost odstraní závislost na použití aplikace Camunda Tasklist společně s omezeními, které pro CZM přináší. Druhá možnost doplní mezeru balíčku Camunda web-app v podobě chybějící funkcionality historie. Důsledkem bude možnost využití pouze aplikací balíčku CW bez rozhraní eProcesy. Omezení spojené s nedostatečnou vývojářskou kontrolou tato možnost ale neodstraní a CZM by se této situaci muselo přizpůsobit jiným způsobem.

Tato práce se zabývá řešením první varianty. Rozhraní eProcesy bude rozšířeno o filtrování úkolů a UX funkcionality embedded formulářů z Camunda Tasklist. Závěrem práce budou obě možnosti řešení finančně a časově zhodnoceny a porovnány.



---

## Analýza vybraných funkcionalit

V této kapitole bude analyzováno reálné použití funkcionality filtrů a UX funkcionalit embedded formulářů v Camunda Tasklist. Analýza je opřena o poznatky CZM z pilotního provozu aplikace.

### 4.1 Analýza filtrování úkolů

Filtrování v Camunda Tasklist nabízí mnoho možností, které byly představeny v kapitole 2.2.5.1, ale pouze některé z nich mají reálné využití. Pohled na filtrování v Camunda Tasklist lze rozdělit na dvě strany. Prvním je strana správců aplikace a druhá koncových uživatelů.

Pro koncové uživatele je důležité zobrazení relevantních úkolů v pro ně dostupných filtrech a seřazení seznamu úkolů dle data vzniku počínaje nejpozdějším záznamem.

Správci aplikace mají rozsáhlejší potřeby. Vyžadují možnost filtry vytvářet, upravovat a mazat. Nastavení filtrů je rozděleno do několika kategorií: General, Permissions, Criteria, Variables.

#### 4.1.1 General

V rámci kategorie General lze pro filtr nastavit název, popis, priorita, barva a automatické obnovení. Účel názvu a popisu je zřejmý, priorita je číselná hodnota, která určuje pořadí zobrazení filtrů, barva nastaví barevné odlišení názvu filtru a automatické obnovení slouží k obnově filtrovaných úkolů v reálném čase. Pro účely správců je důležitý pouze název, popis a priorita.

#### 4.1.2 Permissions

Permissions umožňují nastavit oprávnění pro přístup uživatelů k dotyčnému filtru. Oprávnění lze udělit jednotlivým uživatelům, skupinám uživatelů nebo všem uživatelům naráz. Tato kategorie je vyžadována v plném rozsahu.

### 4.1.3 Criteria

V rámci kategorie Criteria se natavují podmínky, podle kterých se mají úkoly pro uživatele filtrovat. Ze široké škály, která je v nabídce, je pro správce důležité nastavit filtrování dle klíčů: Assignee, Candidate groups, Created Before, Created After. Assignee představuje přiřazeného řešitele k úkolu. Candidate groups jsou skupiny uživatelů, kterým může být úkol přiřazen. Klíč Created After filtruje pouze úkoly vzniklé po zvoleném datu a Created Before úkoly vzniklé před zvoleným datem.

### 4.1.4 Variables

Kategorie Variables není pro účely studijního oddělení a CZM na FEL ČVUT vůbec důležitá, proto kategorii implementace zahrnovat nebude.

### 4.1.5 Zadání k implementaci

Z analýzy výše vzniká zadání pro implementaci. Rozhraní eProcesy bude rozšířeno o filtrování úkolů ve výše popsaném rozsahu.

## 4.2 Analýza UX funkcionalit

V javascriptových kódech z embedded formulářů procesů byly identifikovány jednotlivé funkcionality. Funkcionalita jsou rozděleny na ty, které jsou v automatických formulářích již implementovány a ty, které je potřeba implementovat.

### 4.2.1 UX funkcionality vybrané pro implementaci

Níže následuje seznam identifikovaných funkcionalit které automatické formuláře ještě nepodporují a budou v rámci této práce navrženy a implementovány.

#### Podmíněné zobrazení prvku formuláře

Některá pole formulářů je potřeba zobrazit pouze za určitých podmínek. Podmínkou je ve všech případech určitá hodnota některé z procesních proměnných nastavené v průběhu procesu. Pokud je tato procesní hodnota nastavená dle nastavené podmínky, pole se ve formuláři vykreslí, v opačném případě naopak.

#### Podmíněné potvrzení pro přesun do dalšího kroku procesu

V mnoha formulářích je možné vykonat různé akce, které určují, jak má proces pokračovat. Tato funkcionalita má za cíl pro předem určené typy akcí vyvolat potvrzovací okno s upozorňující otázkou, zda si uživatel opravdu přeje vyvolat příslušnou akci.

### **Překlad hodnoty pole typu boolean**

Některé formuláře obsahují HTML prvky typu boolean a jsou zobrazená jako checkbox. Ve formulářích slouží pro vyjádření ano / ne či doporučuji / nedoporučuji. Na základě vyplnění tohoto pole se uloží boolean hodnota true, respektive false, kterou je poté potřeba promítnout do formulářů v následujících krocích v předem nastaveném znění. Například zaškrtnutý checkbox s názvem "Doporučujete schválit?" se v následujících formulářích vykreslí jako "Studijní referentka doporučuje schválit tuto žádost".

### **4.2.2 Již implementované UX funkcionality**

Všechny níže popsané funkcionality jsou již v automatických formulářích v rozhraní eProcesy podporované.

#### **Povinné pole formuláře**

Chování této funkcionality je velice jednoznačné. Ta pole, která jsou označeny povinným příznakem je nutné před odesláním formuláře vyplnit, v opačném případě jinak nelze přejít do dalšího kroku procesu.

#### **Promítnutí vyplněného pole formuláře v následujících krocích procesu**

Vyplněné pole formulářů je mnohokrát potřeba promítnout do formulářů následujících kroků procesů. Aktuálně je tohoto dosaženo pomocí procesních proměnných, do kterých se ukládají vložené hodnoty a z těch je to poté pomocí javascriptu zobrazeno v HTML prvcích formulářů.

#### **Výběr data pomocí kalendáře**

Při vyplnění formulářového pole typu datum se zobrazí malý kalendář, ve kterém je možné si najít požadované datum a pomocí tlačítka myši vybrat.

#### **Upozornění na nevyplněné povinné pole formuláře**

V případě, že kontrola vyplnění formuláře nebude úspěšná, formulář označí červeně špatně vyplněnou oblast a posune formulář tak, aby uživatel pole viděl ve vrchní části obrazovky.



---

## Návrh a realizace

Aplikace eProcesy/Camunda je napsána v několika technologiích. Pro frontend je zvolena technologie JSF 2.0 (JavaServer Faces), pro backend Java EE 7 komunikující s PostgreSQL databází pomocí Java API vestavěného v Camunda BPM. Obě funkcionality jsou implementovány v rámci nadcházející verze rozhraní eProcesy. Zdrojový kód eProcesů ale není (a ani v blízké době nebude) otevřený, a protože funkcionality nejsou navrženy jako samostatně fungující rozhraní, tak zdrojový kód přiložený v rámci této práce není samostatně spustitelný.

### 5.1 Funkcionalita filtrů

Pro implementaci filtrování úkolů bylo potřeba napsat několik servisních tříd, entit a tříd, které mají za účel držet stav zobrazované stránky aplikace, tzv. backing beans. Myšlenkový postup návrhu jsem zahájil od servisních tříd a jejich metod, které zajistí zpracování dat a byznys logiku funkcionality. Za účelem ukládání filtrů jsem přistoupil k integraci s databázovým modelem filtrů, jež implementuje Camunda.

#### 5.1.1 Servisní třídy

Pro návrh servisních tříd bylo zapotřebí důkladné prostudování Java dokumentace nástroje Camunda BPM a jeho vestavěného Java API. Servisní třídy, které jsem implementoval volají metody servisních tříd zmíněného Java API. Dokumentace popisuje pouze výsledek volání jednotlivých metod, proto bylo zapotřebí volané metody testovat a studovat, jak přesně se chovají ve spolupráci s databází.

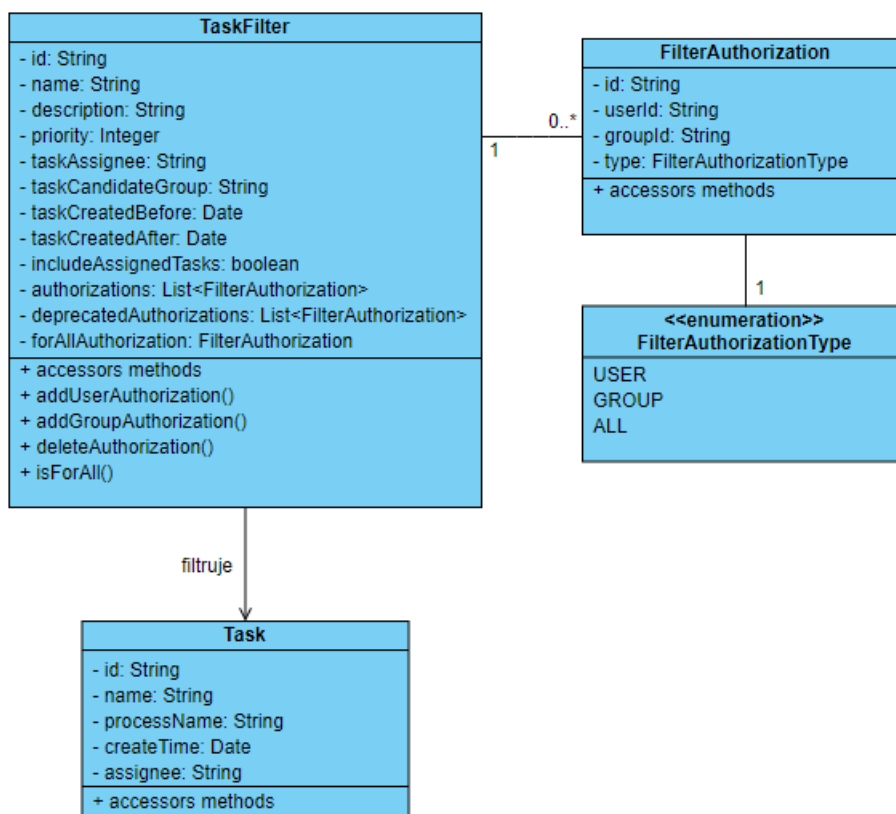
Když bylo plně porozuměno chování jednotlivých servisních tříd a jejich metod poskytující Java API Camunda BPM, tak následovala následovala implementace navržených servisních tříd. Návrh obsahuje čtyři servisní třídy pro práci se třemi hlavními entitami vybrané funkcionality. Tyto třídy obsahují

metody pro uložení záznamů do databáze, jejich odstranění či modifikaci a dále podpůrné metody.

### 5.1.2 Entitní třídy

Entitní třídy mají za cíl reprezentovat jednotlivé prvky ve vybrané funkcionalitě. Těmi jsou: Task (filtrovaný úkol), Authorization (oprávnění filtru), TaskFilter (filtr úkolů). Třídy obsahují atributy jednotlivých entit a metody pro práci s nimi.

Obrázek 5.1: Diagram entitních tříd



### 5.1.3 Backing beans

Backing beans jsou třídy, které mají za cíl držet aktuální stav jednotlivých stránek či komponent aplikace. V tomto případě komponent v rámci jedné stránky. Backing beans jsem navrhl a implementoval celkem čtyři. Je potřeba držet stav seznamu filtrů, seznamu filtrovaných úkolů a dále stav vytváření či editace filtru.



### 5.1.4 Vzhled aplikace

Vzhled aplikace je napsán pomocí technologií XHTML, CSS a frameworků Bootsfaces, Primefaces. Vytvořil jsem čtyři XHTML stránky, jednu pro kostru vzhledu, druhou pro seznam filtrů a úkolů a další dvě pro modální okna sloužící k vytvoření nebo editaci filtru. Vzhled funkcionality je koncipován tak, aby seděl do zbytku aplikace.

## 5.2 UX funkcionality

Úvodem této kapitoly bude seznámení s implementací automatický formulářů v rozhraní eProcesy a bude následovat popis návrhu jejich rozšíření a implementace o jednotlivé funkcionality.

### 5.2.1 Automatické formuláře v rozhraní eProcesy

V rámci procesního modelu lze nastavit pro jednotlivé User Tasky, v jaké podobě se mají vykreslit. V kapitole 2.2.5.2 bylo uvedeno několik způsobů, kterými je možné vykreslení docílit. Rozhraní eProcesy využívá způsob obdobný způsobu, který Camunda nazývá Generated Task Forms. V rámci projektu eProcesy je tento způsob označován jako automatické formuláře.

V definici User Tasku v procesním modelu lze nastavit tzv. Form Fields, jedná se o pole, která se mají ve výsledném formuláři vykreslit. Každému Form Fieldu lze nastavit id, název, typ, validation constraints a properties. Typ je možné vybrat ze základních datových typů string, boolean, long, date a enum. Je možné také zvolit typ *custom type*, který je možné si libovolně přizpůsobit a doprogramovat. Validation constraints jsou podmínky, podle kterých se mají příslušná pole validovat. Tyto validátory lze taktéž programovat dle libovolných potřeb. Takovým validátorem může být například podmínka vyplnění pole ve vzoru telefonního čísla České republiky.

V properties lze nastavit libovolný příznak s názvem a hodnotou. Properties samotné nemají na nic vliv, ale jsou vhodné pro rozlišení, zda se Form Field má generovat jinak než obvyklým nastaveným způsobem. Na základě seznamu Form Fieldů a jejich typu se v eProcesech vygenerují odpovídající JSF komponenty se všemi nastavenými vlastnostmi.

### 5.2.2 Podmíněné potvrzení pro přesun do dalšího kroku procesu

Přesun do dalšího kroku procesu je v automatických formulářích uskutečňován pomocí speciálních tlačítek. Tyto tlačítka nesou název Complete Button. Jsou generovány na základě nastaveného Form Fieldu typu string a s properties obsahující příznak *completeButton*. Pomocí toho eProcesy poznají, že

jde o tlačítko s určitou akcí. Typ akce je poté nastaven taktéž pomocí nastavené property *action* a určitou hodnotou, podle které je proces přesunut do odpovídajícího kroku.”

Funkcionalita byla implementována tak, že v XHTML souboru, který generuje výsledné formuláře, bylo nastaveno rozpoznání, zda tlačítko formuláře je typu *completeButton* a v případě, že ano a bylo toto tlačítko spuštěno, tak dojde k vyvolání potvrzovacího formuláře se zprávou odpovídající akci, která je pro tlačítko nastavená. Pro určování těchto zpráv byla naprogramována třída *ConfirmMessageFactory*, která dle hodnoty *action* nastaví obsah potvrzovacího okna.

### 5.2.3 Podmíněné zobrazení prvku formuláře

Pro implementaci této funkcionality bylo také využito možnosti nastavení properties Form Fieldu. Pro formulářové pole, které mají být zobrazeny, či schovány pouze za určitých podmínek, byla vymyšlena property s názvem *hide* a hodnotou ve formátu JSON, ve které jsou nastaveny podmíněné hodnoty procesních proměnných, při kterých má ke schování formulářového pole dojít. Pro tento případ byla upravena servisní třída *FormService*, která při získávání seznamu Form Fieldu z procesních modelů vyhodnotí, zda daný Form Field obsahuje příznak *hide* a pokud ano, tak vyhodnotí zda jsou splněny nastavené podmínky. V případě, že ano, pole nebude vůbec vykresleno.

### 5.2.4 Překlad pole typu boolean

Form Fieldům typu boolean je možné nastavit v properties příznak *toText*, který značí, že se má daný field přeložit. Properties je potřeba doplnit o příznaky *falseText* a *trueText* a jako jejich hodnotu vyplnit požadovaný textový překlad. V XHTML souboru zodpovědným za vykreslení Form Fieldu typu boolean bylo nastaveno rozpoznání, zda Form Field obsahuje atribut *toText*. V případě, že Form Field atribut obsahuje, tak podle vyplněné hodnoty dotyčného pole se zobrazí jeden z překladů nastavených v attributech *falseText*, respektive *trueText*.

---

## Ekonomicko-manažerské zhodnocení

Aktuální implementace aplikace elektronizující procesy studijního oddělení FEL ČVUT je problematická a není dlouhodobě udržitelná a rozšiřitelná. Dílčím cílem této práce bylo rozšířit rozhraní eProcesy o funkcionality rozhraní Camunda Tasklist a odstranit tak závislost na tomto rozhraní, jež je hlavní příčinou problémů. Druhou možností, jak problémy vyřešit, je odstranění závislosti na rozhraní eProcesy a zakoupení licence Camunda Enterprise, která poskytuje kompletní řešení a služby s ním spojené. Závěrečným cílem této práce a zároveň obsahem této kapitoly je zhodnotit časovou a finanční náročnost obou možností.

### 6.1 Postup zhodnocení

Pro odpovídající zhodnocení a porovnání bylo třeba zajistit veškeré podklady a informace. Pro účely vyčíslení finanční a časové náročnosti vlastního řešení byl veškerý čas strávený implementací měřen a zaznamenáván. Během průzkumu podmínek a náležitostí licence Camunda Enterprise bylo zjištěno, že výsledek srovnání ceny implementovaného řešení v této práci s cenou licence Camunda Enterprise nebude příliš vypovídající vzhledem ke stanovenému problému. Rozsah a přidaná hodnota obou možností se velice liší, tím pádem i okolnosti, za kterých by bylo výhodné k nim přistoupit. Hlavní motivací pro zakoupení licence v tomto případě je rozšíření balíčku CW o funkcionalitu historie, avšak to je pouze zlomek z funkcionalit, které licence přináší. Největším přínosem licence jsou ale doprovodné služby společnosti, motivace pro zakoupení licence by tak měla mít zcela jiný základ. Většina obsahu, který by byl zakoupením licence obdržen, vůbec nemá vliv na stanovený problém.

Na základě tohoto zjištění bylo zhodnocení doplněno o více systémový přístup a vyčíslení finanční a časové náročnosti bylo učiněno nad aktuálním

stavem aplikace eProcesy společně s náročností rozšíření implementovaného v této práci. Porovnání tak bude mezi kompletním řešením vlastního rozhraní a kompletním obsahem licence. Obě varianty budou porovnány kromě finančního hlediska i na základě míry řešení stanoveného problému a výhod, respektive nevýhod, které by přinesli. Výsledkem tak nebude odpověď pouze na otázku, která z těchto dvou možností je finančně výhodnější, nýbrž i na otázku, za jakých podmínek by bylo vhodné zvolit každou z variant. Následovat bude vyhodnocení, která z možností lépe poslouží prostředí a podmínkám Centra znalostního managementu a Fakulty elektrotechnické na ČVUT.

Postup zhodnocení a obsah následujících kapitol:

1. Představení licence Camunda Enterprise
2. Finanční a časová náročnost vlastního řešení
3. Výhody a nevýhody obou variant
4. Porovnání obou variant a vyhodnocení

## 6.2 Licence Camunda Enterprise

Přesné podmínky licence Camunda Enterprise nejsou veřejně dostupné a v případě zájmu o obeznámení s nimi je nutné kontaktovat společnost Camunda Services GmbH. Společnost byla kontaktována a veškeré informace a podklady k licenci, které tato podkapitola shrnuje byly poskytnuty společností prostřednictvím videohovoru s odpovědnou osobou ze zákaznického oddělení.

### 6.2.1 Edice Camunda Community

Společnost Camunda poskytuje pod licencí Apache License 2.0 balíček několika aplikací nesoucí název Camunda Community Edition. Základem jsou BPMN Engine a DMN Engine. Jedná se o frameworky, které umějí spouštět modely v notacích BPMN nebo DMN. Jako nadstavbu pro tyto frameworky nabízí společnost Camunda v rámci této edice několik volně dostupných uživatelských rozhraní:

- Modeler (modelování v notacích BPMN, DMN),
- Tasklist (správa uživatelských úkolů procesů),
- Cockpit (správa procesů a jejich instancí),
- Admin (správa uživatelů a oprávnění).

Aplikace Modeler, Tasklist a Admin jsou volně dostupné v plném rozsahu, aplikace Cockpit je volně dostupná pouze v omezené verzi.

### 6.2.2 Edice Camunda Enterprise

Společnost Camunda nabízí rozšíření pro tento balíček s názvem Camunda Enterprise Edition pod jejich vlastní licencí. Edice Camunda Enterprise nabízí rozšířenou verze rozhraní Cockpit, další dvě aplikace Cawemo, Optimize, a zejména doprovodné služby. Pro lepší představu byl rozdíl mezi volně dostupným a licencovaným balíčkem aplikací a služeb zanesen do tabulky 6.1.

Tabulka 6.1: Porovnání obsahů edice Community a Enterprise

Porovnání obsahů edice Community a Enterprise		
Balíček aplikací	Community	Enterprise
BPMN Engine	Ano	Ano
DMN Engine	Ano	Ano
Modeler	Ano	Ano
Tasklist	Ano	Ano
Cockpit	základní verze	Ano
	rozšířená verze	Ano
Cawemo	Ne	Ano
Optimize	Ne	Ano
Balíček služeb		
Smluvní záruky	Ne	Ano
Podpora založena na SLA (24/7/365)	Ne	Ano
Opravné balíčky pro větší stabilitu	Ne	Ano
Konzultační a tréninkové služby	Ne	Ano

Rozšířená verze Cockpitu přináší funkcionality nasazování procesních modelů a DMN tabulek do aplikace, historii spouštěných procesů, migraci procesů

ních instancí na novou verzi procesního modelu a provádět operace s procesy, které jsou dostupné v rámci základní verzi pouze pro jednotlivé procesy i s celými skupinami procesů naráz.

Cawemo umožňuje spolupráci více lidí při vytváření procesních diagramů a modelů a exportovat modely do externích systémů.

Optimize je nástrojem pro kontinuální optimalizaci automatizovaných procesů. Spouštěné procesy jsou monitorovány, díky čemuž je následně možné vytvářet propracované reporty, analyzovat možné varianty úprav procesů a mnoho dalších optimalizačních nástrojů.

Služby, které společnost nabízí jsou velice bohaté. Podpora je založena na typu smlouvy Service Level Agreement a je pro zákazníky dostupná po celých 365 dní v roce 24 hodin denně online či telefonicky. V případě, že v rámci projektu bude nalezena chyba v poskytovaných aplikacích, je možné ji společnosti nahlásit a chyba bude urgentně řešena a opravný balíček dodán zákazníkům s licencí přednostně. Konzultační a tréninkové služby zahrnují téměř cokoli, co je spjaté s technickými náležitostmi projektu. V případě potřeby je možné konzultovat správnost procesních modelů, navrženou systémovou architekturu nebo dokonce samotný zdrojový kód aplikace. Nabízí tréninkový program pro modelování v notacích BPMN a DMN či zacházení s celou platformou Camunda BPM. Zahrnuty jsou také doporučení vhodných návrhových vzorů nebo vylepšení stávajících. Jednoduše společnost Camunda v rámci licence poskytuje cokoli, co je potřeba k úspěšné integraci a zprovoznění aplikací a projektu celkově.

Na základě tabulky 6.1 a obsahu softwarových rozšíření je možné usoudit, že největší přidaná hodnota licence Camunda Enterprise tkví v poskytovaných službách. Aplikace, které přináší navíc nejsou pro elektronizaci procesů pomocí Camunda BPM klíčové, pouze jej doplňují. Téměř vše potřebné pro úspěšné automatizování a spuštění procesů je již zahrnuto v edici Camunda Community. Funkcionalitou, která v edici Community chybí a dala by se považovat za klíčovou, je historie a audit procesů. Projekt aplikace eProcesy/Camunda toho je příkladem.

### 6.2.3 Podmínky licence

Licence Camunda Enterprise není škálovatelná a nelze ji pořídit pouze na vybrané produkty. Společnost nabízí pouze kompletní balíček aplikací a služeb. Cena není jednotně stanovená a liší se dle rozsahu projektu, pro který by měla být zakoupena. Projekt může spadnout do několika kategorií, které si sama společnost rozlišuje. Její rozsah je vyměřen pomocí několika metrik:

- počet automatizovaných procesů,
- roční objem spuštění všech procesů,

- počet automatizovaných DMN tabulek v rámci procesu,
- počet lidí, kteří budou procesy přímo ovlivňovat.

Projekt aplikace eProcesy/Camunda byl na základě hrubého vyměření daných metrik zařazen do nejnižší cenové kategorie. Tato cenová kategorie začíná na 50 000 € a může se vyšplhat až do 89 000 €, v přepočtu na české koruny to odpovídá rozsahu od 1 345 819,29 Kč do 2 395 558,34 Kč. Trvání zakoupené licence je jeden rok.

#### 6.2.4 Cílová skupina licence

Camunda obecně doporučuje potenciálním zákazníkům, aby konečné rozhodnutí, zda se pro ně vyplatí zakoupení licence Camunda Enterprise nebo postačí edice Community, uskutečnili na základě několika parametrů. Doporučení s parametry a jejich hodnotami shrnuje tabulka 6.2.

Tabulka 6.2: Určení vhodné edice pro příslušný projekt

Parametry projektu	Community	Enterprise
Komplexita hledaného řešení	Nízká	Vysoká
Význam řešení pro organizaci	Nízký	Vysoký
Urgentnost projektu	Nízká	Vysoká
Vývojářská kapacita	Vysoká	Nízká
Existence procesů zahrnujícího zákazníka	Ne	Ano

### 6.3 Finanční a časová náročnost vlastního řešení

Finanční náročnost je postavena na času stráveném implementací funkcionalit, které byly vypracovány v rámci této práce a je doplněn o čas, který byl stráven implementací dosavadní podoby rozhraní eProcesy. Časové údaje poskytlo Centrum znalostního managementu. Časová náročnost jednotlivých funkcionalit byla zanesena do tabulky 6.3.

Tabulka 6.3: Časová náročnost implementace rozhraní eProcesy

Implementovaný rozsah	Časová náročnost (v hodinách)
Filtrování úkolů	94
Rozšíření automatických formulářů	22
Aktuální stav rozhraní eProcesy	452
<b>Celkem</b>	<b>568</b>

Pro získání konkrétní finanční částky bude výsledný čas vynásoben hodinovou mzdou. Pro tuto práci předem nebyla výše hodinové mzdy určena, proto vzhledem ke geografickému umístění řešeného problému, kterým je Praha, je uvažována obvyklá hodinová mzda programátora v této oblasti. Z průzkumu portálu Smitio.cz odpovídá medián hrubého měsíčního platu programátora v Praze výši 65 000 Kč.[14] Výsledný náklad CZM na programátora by byl ale ve výši superhrubé mzdy. Výše superhrubé mzdy odpovídá 86 970 Kč. Pokud budeme počítat se 160 hodinovou měsíční pracovní dobou bude náklad CZM na hodinu práce programátora 543,6 Kč. Vynásobením celkového času hodinovou sazbou bude výsledná částka 308 764,8 Kč. Finanční náročnost rozšíření, které bylo vytvořeno v rámci této práce odpovídá 63 057,6 Kč.

## 6.4 Výhody a nevýhody obou variant řešení

Základem pro stanovení výhod a nevýhod daných řešení byly podmínky a požadavky projektu aplikace eProcesy/Camunda a nelze je tak brát zcela objektivně. U velkého projektu s neomezenými finančními prostředky, kde by se rozhodovalo o výběru i mezi jinými nástroji pro automatizaci procesů by takový výstup vypadal pravděpodobně zcela jinak.

Obě možnosti mají své pro a proti. Každá z variant je vhodná pro trochu odlišné situace. Koupě licence Camunda Enterprise je určena spíše pro rozsáhlejší a komplexnější projekty, jejichž výsledek má veliký dopad na fungování příslušné organizace nebo společnosti. Zatímco využití volně dostupné edice Community společně s vyvinutím vlastního rozhraní je vhodné pro projekty, kde se očekává menší objem spouštěných procesů, potřeby projektu jsou velice specifické a které nemají přidělené mnoho finančních prostředků. Příkladem může být právě projekt elektronizace studentských žádostí na FEL ČVUT. Automatizovaných procesů je celkem 13 a intenzita jejich spuštění nepřesahuje 20 instancí za týden.



Obecné výhody a přínosy licence Community Enterprise:

- poskytnuté komplexní, hotové řešení,
- efektivita vývoje a integrace nástrojů do příslušného technického prostředí,
- podpora ze strany společnosti při jakémkoliv problému v jakýkoliv den v roce,
- nástroje pro průběžného monitorování a optimalizace procesů.

V případě zvolení této varianty by CZM nevyřešilo všechny problémy. Licence Camunda Enterprise by odstranila problém s existencí dvou rozhraní. Díky rozšíření o funkcionalitu historie procesů by nebylo nutné dál používat rozhraní eProcesy a veškerá uživatelská činnost by tak mohla být pohodlně prováděna v rámci jednotného rozhraní. Licence by ale nevyřešila problémy týkající se udržitelnosti embedded formulářů a nedostatečná vývojářská kontrola by také přetrvávala. V takovém případě by CZM muselo hledat jiné, další řešení těchto problémů. Z pohledu CZM přináší řadu nevýhod:

- více než 4x vyšší cena oproti vlastnímu řešení,
- trvání licence pouze jeden rok,
- nemožnost zakoupení vybraných částí licence,
- nedostatečné řešení stávajících problémů.

Varianta vlastního řešení společně s využitím edice Camunda Community vyhovuje podmínkám projektu aplikace eProcesy/Camunda mnohem více z několika důvodů:

- téměř všechny potřebné funkcionality jsou pokryty edicí Camunda Community,
- možnost implementace pouze vybraných a potřebných funkcionalit,
- mnohonásobně nižší výsledná cena,
- řešení všech stávajících problémů,
- modifikovatelnost řešení dle uživatelských potřeb.

Díky rozšíření rozhraní eProcesy nebude nutné použití aplikace Tasklist a funkcionality embedded formulářů. Centrum znalostního managementu tak bude mít pouze jedno uživatelské rozhraní, které bude mít plně pod kontrolou, bude se snadněji udržovat, rozšiřovat a bude uživatelsky přívětivější. Nevýhody tohoto řešení spočívají zejména v tom, že se CZM při implementaci musí spolehnout pouze sama na sebe a v případě problémů se nemůže spolehnout na aktivní podporu ze strany společnosti Camunda.

## 6.5 Zhodnocení a doporučení

Na základě všech podkladů a informací posbíraných o licenci Camunda Enterprise a projektu eProcesy/Camunda vychází pro Centrum znalostního managementu výhodněji implementace vlastního řešení. Cena za licenci by byla velice neúměrná k problémům, které by díky této volbě byly vyřešeny.

Zakoupení licence bude výhodné tehdy, kdy je potřeba komplexního řešení, které nemá příliš specifické požadavky na uživatelské rozhraní, bude počítat s implementací vícero druhů procesů a bude očekávána vysoká intenzita spouštění procesů. Takovým případem by mohl být projekt pro automatizaci procesů v rámci celé Fakulty elektrotechnické nebo celé univerzity ČVUT. Objem procesů celé univerzity by byl enormně vyšší oproti studijnímu oddělení jedné z fakult. Tím by úměrně vzrostl i počet spuštění procesů. Spolu s tím by narostla i komplexita řešení, potřeba monitorovat a optimalizovat procesy. Během stavby a návrhu tohoto řešení by pak své opodstatnění našly zejména tréninkové a konzultační služby společnosti Camunda.

---

## Závěr

Na Fakultě elektrotechniky ČVUT v Praze vyvíjí Centrum znalostního managementu aplikaci eProcesy/Camunda, jejíž účelem je elektronizace procesů studijního oddělení, konkrétně agendy studentských žádostí. Aktuální stav aplikace je pro Centrum znalostního managementu a fakultu problematický. Aplikace využívá dvou uživatelských rozhraní eProcesy a Camunda Tasklist. Přístupovat do dvou různých rozhraní je uživatelsky velice nepřívětivé, avšak to není jediný problém. Centrum znalostního managementu nemá nad rozhraním Camunda Tasklist dostatečnou vývojářskou kontrolu, neboť je vytvořeno v technologiích, ve kterých centrum není zvyklé vyvíjet. Posledním důležitým problémem je obtížná udržitelnost a rozšiřitelnost aplikace a procesních formulářů. Existují dvě varianty řešení. První tvoří implementace funkcionality filtrů a embedded formulářů z rozhraní Camunda Tasklist do aplikace eProcesy. Výsledkem bude odstranění závislosti na použití Camunda Tasklist. Druhou je zakoupení licence Camunda Enterprise, díky které bude naopak možné používat samostatně rozhraní Camunda Tasklist. Tato práce se zabývala vytvořením prvního ze zmíněných řešení. V práci bylo analyzováno, navrženo a implementováno rozšíření do rozhraní eProcesy o funkcionality filtrů a UX funkcionality embedded formulářů z rozhraní Camunda Tasklist. V závěru práce bylo provedeno ekonomicko-manažerské vyhodnocení a porovnání obou variant. První varianta řešení je pro Centrum znalostního managementu finančně i systémově výhodnější. Finanční náročnost varianty byla zhodnocena na 308 764,8 Kč. Cena licence Camunda Enterprise začíná na 1 345 819,29 Kč. To je více jak 4x vyšší částka oproti první variantě. Bylo zjištěno, že srovnávat tyto dvě možnosti pouze na základě ceny není zcela vypovídající. Licence Camunda Enterprise totiž není vůbec vhodná pro řešení tohoto typu problémů a motivace pro její zakoupení by měla mít jiný základ. Přidaná hodnota licence tkví zejména ve službách společnosti, které by při této velikosti projektu nenašly své opodstatnění. Závěrem práce bylo vyhodnoceno, že varianta implementace vlastního řešení je za aktuálních podmínek pro Centrum znalostního managementu výhodnější. Vychází finančně lépe a pomůže oproti druhé

## ZÁVĚR

---

variantě odstranit všechny stávající problémy. Zakoupení licence Camunda Enterprise je vhodné pro projekty, v rámci kterých se očekává velký objem automatizovaných procesů různorodého typu, a intenzita jejich spuštění bude mnohonásobně vyšší, než jak tomu je u aplikace eProcesy/Camunda.

---

## Bibliografie

1. WESKE, Mathias. *Business Process Management: Concepts, Languages, Architecture*. 2. vyd. Berlin Heidelberg: Springer-Verlag, 2012. ISBN 978-3-642-28616-2.
2. DUMAS, Marlon; ROSA, Marcello La; MENDLING, Jan; REIJERS, Hajo A. *Fundamentals of Business Process Management*. 1. vyd. Berlin Heidelberg: Springer-Verlag, 2013. ISBN 978-3-642-33143-5.
3. Business Process Modeling- A Comparative Analysis. *Journal of the Association for Information Systems* [online]. 2006 [cit. 2020-04-10]. Dostupné z: <https://pdfs.semanticscholar.org/cdd5/1a86660b2571254b1b04e9e78e947ed82da2.pdf>.
4. GROUP, Object Management. *Business Process Model and Notation (BPMN)* [online] [cit. 2020-04-10]. Dostupné z: <https://www.omg.org/spec/BPMN/>.
5. INC., Lucid Software. *What is Business Process Modeling Notation* [online] [cit. 2020-04-17]. Dostupné z: <https://www.lucidchart.com/pages/bpmn>.
6. GROUP, Object Management. *Decision Model and Notation (DMN)* [online] [cit. 2020-04-10]. Dostupné z: <https://www.omg.org/spec/DMN/>.
7. ŘEPA, Václav. *Procesně řízená organizace*. 1. vyd. Management v informační společnosti, Grada, 2012, 2012. ISBN 9788024741284. Dostupné také z: [https://books.google.cz/books?id=e\\_JYVujpXK8C](https://books.google.cz/books?id=e_JYVujpXK8C).
8. COALITION, Workflow Management. *The Workflow Reference Model* [online] [cit. 2020-04-17]. Dostupné z: <http://www.wfmc.org/standards/docs/tc003v11.pdf>.

9. Camunda hauls in \$28M investment as workflow automation remains hot. *TechCrunch* [online]. 2018 [cit. 2020-04-17]. Dostupné z: <https://techcrunch.com/2018/12/05/camunda-hauls-in-28m-investment-as-workflow-automation-remains-hot/>.
10. Bossie Awards 2015: The best open source applications. *InfoWorld* [online] [cit. 2020-04-17]. Dostupné z: <https://www.infoworld.com/article/2982622/bossie-awards-2015-the-best-open-source-applications.html>.
11. GMBH, Camunda Services. *Camunda BPM documentation* [online] [cit. 2020-04-17]. Dostupné z: <https://docs.camunda.org/manual/7.5/>.
12. FEL ČVUT, Centrum znalostního managementu. *eProcesy* [software]. 2019 [cit. 2020-04-17]. Dostupné z: <https://www.fel.cvut.cz/eprocesy/>.
13. FEL ČVUT, Centrum znalostního managementu. *Camunda* [software]. 2019 [cit. 2020-04-17]. Dostupné z: <https://www.fel.cvut.cz/camunda/>.
14. Mzdy podle smítia 2.0. Část 1: Srovnání s průzkumem z 2019. *Smitio* [online]. 2020 [cit. 2020-05-20]. Dostupné z: <https://blog.smitio.com/clanek-mzdy-podle-smitia-2-0-srovnani>.

## Seznam použitých zkratek

- BPM** Business Process Management
- BPMN** Business Process Model and Notation
- CW** Camunda web-app
- CZM** Centrum znalostního managementu
- DMN** Decision Model and Notation
- DRD** Decision Requirements Diagram
- OMG** Object Management Group
- UX** User experience
- XML** Extensible markup language





---

## Obsah přiloženého CD

readme.txt .....	stručný popis obsahu CD
impl	
├── src .....	zdrojové kódy implementace
└── example_process .....	ukázková procesní definice
thesis	
├── src .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
└── BP_Dominik_Stasiowski_2020.pdf .....	text práce ve formátu PDF