



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Portál EBIE - nasazení pilotní verze v prostředí VUT
Student:	Bc. Ondřej Batík
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

1. Seznamte se s koncepcí portálu EBIE (Extended BI Encyclopedia) a diplomovými pracemi [1] a [2], které se zabývaly detailními návrhy jeho struktury a obsahu.
2. S využitím metodiky DEMO a případně dalších vhodných postupů navrhnete a dokumentujete procesy údržby, plnění a rozšiřování obsahu portálu.
3. Seznamte se s bakalářskými pracemi [3], [4] a [5], které se vnovaly implementaci částí portálu.
4. Vypracujte plán integrace těchto částí v novém pilotním nasazení portálu EBIE v prostředí VUT, zohledněte též výsledky bodu 2 své práce.
5. V pozici projektového manažera plán integrace a nasazení portálu realizujte.

Seznam odborné literatury

- [1] Václav Jírovský: Konceptuální analýza datových domén Studium a Hodnocení kvality výuky s ohledem na datovou istotu. Diplomová práce VUT FIT. 2015.
- [2] Ladislav Moravec: Ontologická analýza datového skladu VUT. Diplomová práce VUT FIT. 2016.
- [3] David Hajšiar: Návrh a implementace backend projektu EBIE. Bakalářská práce VUT FIT. 2016.
- [4] Cyril Šerný: Návrh a implementace integrace služeb do projektu EBIE. Bakalářská práce VUT FIT. 2016.
- [5] Michal Kopecký: Návrh a implementace klientské části systému EBIE. Bakalářská práce VUT FIT. 2016.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdlík, CSc.
děkan

V Praze dne 23. listopadu 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Portál EBIE – nasazení pilotní verze v prostředí ČVUT

Bc. Ondřej Batík

Vedoucí práce: Ing. Michal Valenta, Ph.D.

3. května 2017

Poděkování

V první řadě bych rád poděkoval vedoucímu Ing. Michalovi Valentovi, Ph.D. a oponentovi Ing. Robertu Perglovi, Ph.D. za cenné rady během tvorby práce. Dále bych chtěl poděkovat členům vývojového týmu EBIE a datového skladu za jejich skvělou spolupráci. Poslední poděkování patří mým nejbližším, kteří mě podporovali v průběhu celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické 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 předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona 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. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 3. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Ondřej Batík. 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, je nezbytný souhlas autora.

Odkaz na tuto práci

Batík, Ondřej. *Portál EBIE – nasazení pilotní verze v prostředí ČVUT*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Práce se zabývá sestavením komplexního plánu obsahujícího plán nasazení pilotního portálu EBIE (Extended Business Intelligence Encyclopedia) a plán integrace, který se skládá z výstupů prací definovaných v zadání. Tento plán je následně realizován autorem práce v roli projektového manažera.

Druhým a hlavním úkolem, který je v práci řešen, je analyzování a namodelování procesů pro údržbu, plnění a rozšiřování obsahu portálu EBIE. Pro modelování procesů je použita metodika DEMO a u některých procesů také notace BPMN. Základní principy těchto modelovacích technik jsou v práci představeny. Práce mimo jiné stručně pojednává o datové čistotě, datovém skladu a portálu EBIE.

Výsledkem práce je souhrnný plán nasazení a integrace, jeho vyhodnocení a především analyzované a namodelované procesy. Na konci samotné práce je provedena simulace procesu pro publikování obsahu na portál EBIE. Výsledné procesy by měly najít patřičné využití.

Klíčová slova Portál EBIE, plán integrace, plán nasazení, DEMO, BPMN, proces

Abstract

The thesis deals with the compilation of a complex plan containing a deployment plan for the pilot portal EBIE (Extended Business Intelligence Encyclopedia) and an integration plan, which consists of the outputs of works defined in the assignment. This plan is subsequently realized by the author of the work in the role of project manager.

The second and the main task, which is being solved in this thesis, is analyzing and modeling processes for the maintenance, fulfillment and dissemination of EBIE content. The DEMO methodology is used for process modeling, and for some processes also the BPMN notation. The basic principles of these modeling techniques are presented in the thesis. The thesis, among the other things, briefly discusses data purity, data warehouse and EBIE.

The outcome of the thesis is comprehensive deployment and integration plan, its evaluation and, above all, analyzed and modeled processes. At the end of the thesis itself, a simulation of the content publishing process on the EBIE portal is performed. The outcoming processes should find appropriate use.

Keywords Portal EBIE, integration plan, deployment plan, DEMO, BPMN, process

Obsah

Úvod	1
Motivace	1
Cíl práce	2
Struktura práce	2
1 Datová čistota a EBIE	3
1.1 Datová čistota	3
1.2 Datový sklad ČVUT	5
1.3 EBIE	8
2 Projektový tým, nasazení a integrace	17
2.1 Projektový tým	17
2.2 Průběh nasazení a integrace	19
2.3 Vyhodnocení plánu nasazení	24
3 Aktuální stav EBIE a analýza procesů	27
3.1 Aktuální stav portálu EBIE	27
3.2 Analýza procesů pro správu obsahu	33
4 Modelování procesů pro správu obsahu	43
4.1 Představení modelovacích notací	43
4.2 Modelování analyzovaných procesů	57
5 Publikování obsahu a vyhodnocení integračního plánu	79
5.1 Ukázka publikování obsahu na testovací server EBIE	79
5.2 Vyhodnocení integračního plánu	82
Závěr	85
Literatura	87

A	Seznam použitých zkratk	91
B	Přílohy	93
B.1	Adresářová struktura portálu	93
B.2	DEMO	96
C	Obsah přiloženého CD	101

Seznam obrázků

1.1	Koncept architektury navrhovaného řešení – EBIE.	4
1.2	Architektura nového datového skladu ČVUT [7].	6
1.3	Informační architektura EBIE [7].	10
1.4	Informační architektura aplikace navržená Ing. Václavem Jirovským [16].	11
1.5	Konceptuální datové schéma EBIE vytvořené Bc. Davidem Hajčirem [18].	13
2.1	Plán popisující integraci a nasazení portálu EBIE.	20
2.2	RACI matice integračního plánu a plánu nasazení portálu EBIE.	24
2.3	Produkční verze portálu EBIE.	25
3.1	Aktuální informační architektura EBIE [9].	28
3.2	Část atributů tabulky t_sppl_studijni_obor.	29
3.3	Ukázka schématu datového tržiště.	30
3.4	Část technických atributů pojmu Klasifikace.	31
3.5	Transakce T01 – uznání předmětu ostatních fakult ČVUT.	34
3.6	Nová položka pro entitu Fakulta v souboru info.json [31].	36
3.7	Ukázka „Merge Requestu“ z větve master_vyvoj do větve master.	36
4.1	Základní elementy BPMN [39].	46
4.2	Specifické elementy BPMN použité při modelování procesů.	46
4.3	Happy flow transakce [43].	49
4.4	Standardní transakční vzor [44].	50
4.5	Distinkční axiom [45].	51
4.6	Tři aspekty organizace [44].	52
4.7	Podmodely ontologického modelu [44].	53
4.8	Publikování obsahu na testovací server (BPMN).	60
4.9	Publikování obsahu na testovací server (ontologický rozbor).	61
4.10	Publikování obsahu na testovací server (OCD).	62
4.11	Publikování obsahu na testovací server (PSD).	63

4.12	Publikování obsahu na produkční server (BPMN).	66
4.13	Publikování obsahu na produkční server (ontologický rozbor).	67
4.14	Publikování obsahu na produkční server (OCD).	68
4.15	Publikování obsahu na produkční server (PSD).	69
4.16	Nahlášení chyby ve zdrojových systémech (BPMN).	70
4.17	Nahlášení chyby ve zdrojových systémech (ontologický rozbor).	71
4.18	Nahlášení chyby ve zdrojových systémech (OCD).	72
4.19	Nahlášení chyby ve zdrojových systémech (PSD).	73
4.20	Zavedení kontroly kvality dat (OCD).	74
4.21	Zavedení kontroly kvality dat (PSD).	75
4.22	Aplikování kontroly kvality dat (OCD).	77
4.23	Aplikování kontroly kvality dat (PSD).	78
5.1	Ukázka vytvořeného souboru info.json.	80
5.2	Ukázka modulu Dokumentace pro vývojáře se sekci Procesy.	81
5.3	Ukázka seznamu procesů v sekci Procesy.	81
5.4	Ukázka procesu Publikování obsahu na testovacím serveru EBIE.	82
B.1	EBIE dirtree návrh 1	94
B.2	EBIE dirtree návrh 2	95
B.3	Kompletní transakční vzor [44].	96
B.4	Elementy OCD diagramu (1) [48].	97
B.5	Elementy OCD diagramu (2) [48].	97
B.6	Elementy OCD diagramu (3) [48].	98
B.7	Elementy PSD diagramu (1) [48].	99
B.8	Elementy PSD diagramu (2) [48].	99

Seznam tabulek

4.1	Publikování obsahu na testovací server (TPT).	62
4.2	Publikování obsahu na testovací server (AOM).	64
4.3	Publikování obsahu na produkční server (TPT).	68
4.4	Publikování obsahu na produkční server (AOM).	69
4.5	Nahlášení chyby ve zdrojových systémech (TPT).	72
4.6	Nahlášení chyby ve zdrojových systémech (AOM).	73
4.7	Zavedení kontroly kvality dat (TPT).	75
4.8	Aplikování kontroly kvality dat (TPT).	77

Úvod

Motivace

Na světě existuje velké množství dat, které postupem času stále narůstá. Se zvětšujícím se objemem dat začaly firmy řešit otázky, jak a kde data uchovávat a především, jak je efektivně využívat.

Na počátku 80. let 20. století začaly vznikat první provozní informační systémy, jejichž hlavním účelem je podpora každodenních operací a činností v dané organizaci, které souvisí s uchováváním dat. Provozní informační systémy ukládají data obvykle v relačních databázích a uživatelé nad nimi mohou provádět jednoduché transakce. Tyto systémy však nejsou vhodné pro vyhodnocování složitějších dotazů, které by nám mohly poskytnout statistické a přehledové výsledky pro budoucí rozhodování [1]. Zmíněné nedostatky vedly k vytváření datových skladů (Data Warehouses), které slouží jako vrstva nad databází/databázemi pro analytické operace a nabízí nám tak možnost zkoumat data z jiného pohledu [2].

Velký nárůst dat za posledních několik let, spojený s pojmem Big Data a IoT (Internet of Things, internet věcí) způsobil, že datové sklady nemusí obsahovat pouze data z různých zdrojových systémů (např. podnikových informačních systémů), ale také z technických přístrojů (chytrých telefonů, domácích spotřebičů atd.) [3].

Nejen všechny uvedené faktory vedou k potřebě dbát větší pozornosti nad datovou kvalitou. Pomocí datové kvality se snažíme říci, zdali jsou sledovaná data kvalitní, což ve skutečnosti nemusí být tak lehké určit, jak se na první pohled může zdát. Kvalitu dat posuzujeme pomocí dimenzí, mezi které patří například dostupnost, věrohodnost, konzistentní reprezentace, relevantnost a další [4]. S datovou kvalitou je provázána datová čistota (čištění dat), která se zabývá hledáním a odstraňováním chyb z dat pro zlepšení datové kvality [5]. Zvýšení datové kvality může zmenšit výskyt duplikátních hodnot, snížit počet nepotřebných dat a ušetřit čas při opravování chyb způsobených nedostateč-

nou datovou kvalitou [6].

Během posledních pár let vznikaly na Českém vysokém učení technickém (dále jen ČVUT) projekty, které se týkaly datového skladu a datové čistoty. Na těchto projektech se podíleli, a stále podílí, bývalí i současní zaměstnanci a studenti ČVUT. Součástí jednoho z nich bylo vytvoření rozšířené BI encyklopedie nesoucí název EBIE (Extended Business Intelligence Encyclopedia), na kterou se zaměřuje hlavní část této práce.

Cíl práce

V práci je kladen důraz na splnění dvou cílů.

Prvním cílem je vytvoření integračního plánu a plánu nasazení portálu EBIE v prostředí ČVUT, který mám v roli projektového manažera realizovat. V integrační části by měly být zohledněny výstupy prací, které se týkaly portálu EBIE.

Druhým cílem a hlavní přidanou hodnotou práce by měly být analyzované a namodelované procesy, které se zaměřují na údržbu, plnění a rozšiřování obsahu portálu. Pro modelování procesů by měla být použita metodika DEMO a případně jiné postupy či technologie. K metodice DEMO tak byla vybrána známá notace BPMN.

Struktura práce

Celá práce je rozdělena do pěti kapitol, z nichž první čtyři jsou obsáhlejší. V první kapitole je provedena rešerše zmiňovaných prací, které souvisejí s portálem EBIE a které byly sepsány jako závěrečné práce studentů ČVUT. V této kapitole jsou mimo jiné rozebrány pojmy datová čistota, datový sklad a EBIE. V druhé kapitole je navržen plán nasazení a integrace a je zde popsán jeho průběh. Ještě před tím jsou v krátkosti představeny týmy pracovníků, které se na projektu podílely. V poslední části kapitoly je vyhodnoceno nasazení portálu. Třetí kapitola popisuje aktuální stav portálu EBIE. V druhé části této kapitoly jsou analyzovány vybrané procesy. Ve čtvrté kapitole jsou uvedeny modelovací techniky DEMO a BPMN. Následně jsou analyzované procesy z předchozí kapitoly s využitím těchto technik namodelovány. Poslední kapitola se zaměřuje na ukázkou vybraného procesu pro publikování obsahu na portál EBIE. Na samotném konci je vyhodnocena integrační část projektu.

Datová čistota a EBIE

Pro zasloužení do problematiky je nutné se seznámit s projekty, které se týkají samotné EBIE a které vedly k jejímu zavedení.

1.1 Datová čistota

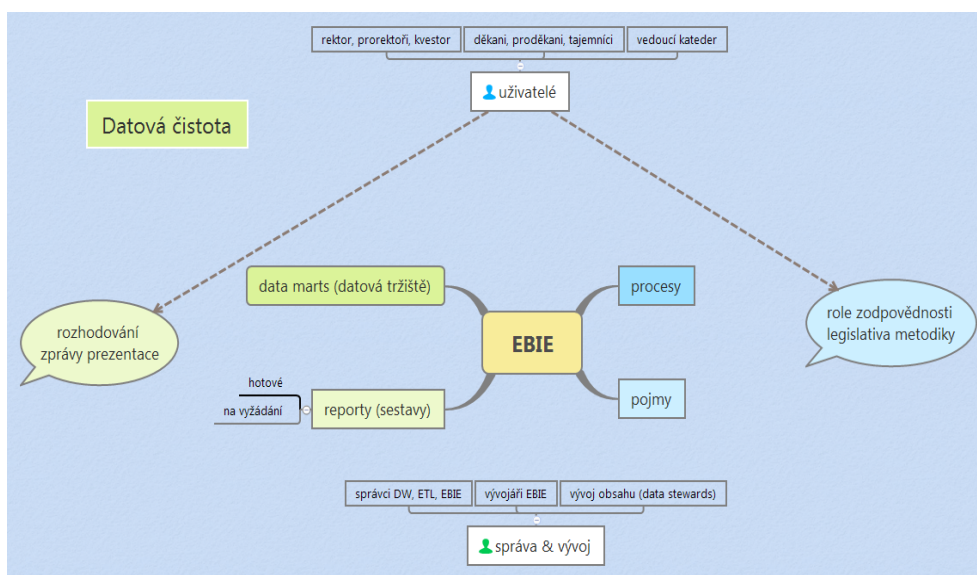
V roce 2015 vznikl projekt v rámci ČVUT IP projektů nazvaný „Datová čistota“ (DU-20). Jedním z hlavních předmětů projektu bylo podle výstupu „IP 2015 DU 20 – Datová čistota – manažerské shrnutí“ [7] analyzování Manažerského informačního systému, který je součástí Portálu ekonomických služeb (dále jen PES/MIS) ¹. Projekt byl zaměřen převážně na doménu Studium a Hodnocení kvality studia. Výsledkem projektu bylo, že stav dat z hlediska kvality, struktury a úplnosti je neuspokojivý. Z manažerského hlediska byly vybrány tři příčiny, které je nutné řešit, má-li se kvalita dat potřebných pro rozhodování změnit k lepšímu:

- **Chybějící metodici** – Chybí role metodika dat („data steward“), který by mohl působit jako garant kvality dat.
- **Děravé systémy zdrojových dat** – Pro zkoumanou doménu Studium například databáze a systém KOS (KOmponenta Studium), u kterého nelze realizovat kompletní kontrolu vkládání dat z jeho rozhraní.
- **Neexistující systém (proces) kontroly kvality dat** – Není dostupný žádný proces pro kontrolu kvality dat spolu s legislativní podporou vylepšení kvality. Proces by měl také obsahovat kompetentní role, ustanovený způsob, jak prezentovat aktuálně platné datové schéma a proces změnového řízení.

Zmíněné příčiny nebyly jedinými problémy, které vyplynuly z komplexní analýzy PES/MIS. Po provedení analýzy se objevily další nedostatky, které se

¹Portál je dostupný na adrese <http://pes.cvut.cz/>

1. DATOVÁ ČISTOTA A EBIE



Obrázek 1.1: Koncept architektury navrhovaného řešení – EBIE.

netýkaly pouze modulu Studium, ale také ostatních modulů PES/MIS². Mezi zmíněné nedostatky patřily například nedostatečná metadata, chyby v datech, nevhodné uživatelské rozhraní, nevhodná prezentace dat a další. Na všechny chyby bylo třeba brát ohled v budoucím návrhu nové koncepce.

Dvě nejdůležitější navrhovaná opatření v budoucí souvislosti s touto prací, která vyplynula z proběhlé analýzy, byla:

1. **Vytvoření nového datového skladu ČVUT** – Nový datový sklad, jehož Access Layer, viz 1.2.2, bude tvořena nezávislými datovými tržišti (data marts) reprezentující jednotlivé moduly a nabídne tak výhody týkající se dokumentace metadat (přes BI encyklopedii) a zajímavější prezentaci dat.
2. **Navrhnutí a implementování rozšířené BI encyklopedie** – BI encyklopedie pro podporu byznys požadavků, kontextu a metodik formou konceptuální vrstvy.

Celkový koncept architektury navrhovaného řešení můžeme vidět na obrázku 1.1, který bude dále rozebrán v podkapitole 1.3.

²Mluvíme o modulech: Ekonomika, Majetek, Mzdové sestavy, Objednávky/Faktura atd.

1.2 Datový sklad ČVUT

Nový datový sklad ČVUT vzniká na základě požadavků projektu Datové čistoty. Při realizaci tohoto projektu byl vytvořen prototyp skladu, jehož kompletní návrh a implementace se přesunuly do IP projektu „Datový sklad ČVUT“ (DU-46). Podobným datovým skladem, akorát z prostředí Fakulty informačních technologií na ČVUT, se ve své diplomové práci „Datový sklad fakulty“ zabývá Ing. Stanislav Kuznetsov [8]. Některé části textu jsou získány právě z této práce a také z dokumentu [9].

1.2.1 Charakteristika datového skladu

V dnešní době v oblasti datových skladů existují dva odlišné přístupy k dané problematice. Jejich autory a zároveň vlastními oponenty jsou William H. Inmon, přezdívaný „otec datových skladů“, a Ralph Kimball. Pro získání kompletního přehledu o dvou přístupech bychom potřebovali více prostoru, který v této práci nemáme. Podrobné charakteristiky datových skladů si můžeme přečíst například ve zdrojích [10] (Inmon) a [11] (Kimball). Zde si pokusíme představit alespoň základní myšlenky těchto přístupů. Následující podkapitola čerpá text převážně ze zdroje [12].

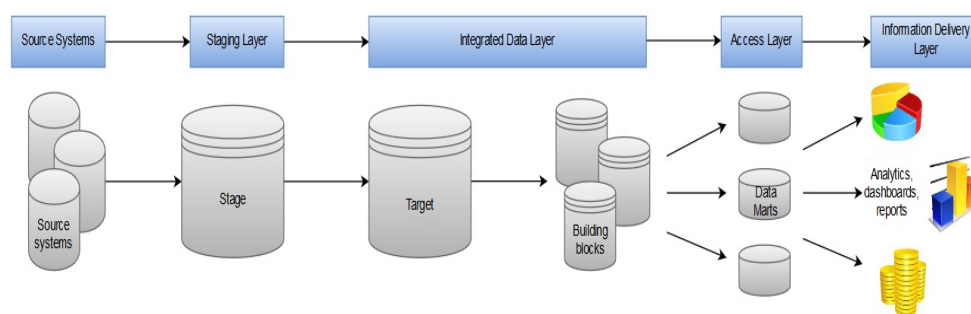
1.2.1.1 Datový sklad podle W. H. Inmona nebo R. Kimballa?

W. H. Inmon i R. Kimball považují datový sklad jako centrální úložiště dat pro organizace. Datový sklad by podle jejich názoru měl primárně sloužit k poskytování reportů a měl by být plněn pomocí ETL ³. Hlavní rozdíly jejich názorů jsou v přístupu k datovým strukturám – jejich modelování, uložení a uchování v datovém skladu.

Inmon přistupuje k vybudování datového skladu nad konceptuálním modelem. Pomocí konceptuálního modelu jsme schopni identifikovat klíčové entity, se kterými byznys pracuje a které ho nejvíce zajímají. Důležité je zmínit, že struktura entit je vybudována v normalizované formě, čímž se snažíme co nejvíce vyhnout redundanci dat. Inmon také navrhuje vytvoření datových tržišť, viz 1.2.3, specifických pro jednotlivá oddělení v organizaci (finance, marketing atd.). Podle Inmona, datová tržiště mohou obsahovat denormalizovaná data pro potřeby reportingu. Všechna data, která se dostanou do datového skladu, jsou integrovaná a datový sklad navíc slouží jako jediný zdroj zmiňovaných datových tržišť. Z těchto faktů vyplývá, že datový sklad slouží jako jediný zdroj pravdy pro organizaci.

Kimballův přístup začíná u identifikování klíčových byznys procesů a byznys otázek, na které musí datový sklad poskytnout odpověď. ETL mechanismus je použit pro přenesení dat ze zdrojových systémů do tzv. „staging

³ETL (Extract, Transform, Load) označuje mechanismus získávání dat z provozních systémů, jejich následné zpracování a nahrání do datového úložiště – např. datového skladu [13].



Obrázek 1.2: Architektura nového datového skladu ČVUT [7].

vrstvy“, odkud jsou data přenášena do dimenzionálního modelu. Tento model však není normalizovaný. Dimenzionální model je tvořen datovými tržišti. Uživatelé přistupují do jednotlivých datových tržišť, která však často mohou obsahovat redundantní data, jelikož datová tržiště nevycházejí ze stejného, centralizovaného zdroje dat.

Pro návrh nového datového skladu na ČVUT byl vybrán přístup podle Inmona, a to nejen ze zmíněných přínosů integrovaného datového skladu. Konceptuální model, na kterém si Inmon zakládá, nám poskytuje další výhody:

- Konceptuální model nám pomůže lépe se zorientovat v dané doméně.
- Z dokumentace konceptuálního modelu (konceptuální dokumentace) můžeme vytvářet různé legislativy a metodiky. Konceptuální dokumentace tedy slouží jako podklad pro vytváření legislativ a metodik, které jsou součástí koncepce návrhu portálu EBIE popsané v části 1.3.1.

1.2.2 Architektura datového skladu

Příklad navržené architektury datového skladu ČVUT je vidět na obrázku 1.2. Architektura se skládá z několika základních vrstev:

- Source Systems – Jedná se vlastně o provozní systémy, které jsou zdrojem dat ukládaných do datového skladu. V prvotním návrhu se počítalo se zdrojovými databázemi KOS, Anketa ČVUT a Závěrečná práce (specifická databáze na FIT). Další přidáné/chystané zdroje: V3S ⁴, USER-MAP ⁵ a další.

⁴Aplikace evidující výsledky vědy a výzkumu a další aktivity vědecko-výzkumných pracovníků ve vědecké komunitě [14].

⁵Systém sloužící pro celouniverzitní evidenci uživatelů (zaměstnanců, studentů atd.) [15].

- Staging Layer – Stejná struktura jako u zdrojů – jedná se o obraz ze zdrojových systémů, respektive jejich relevantních částí, otisk 1:1. Nedochozí k žádným úpravám dat (historizace, čištění apod.).
- Integrated Data Layer – Pro tuto vrstvu je typická historizace a kontrola datové čistoty. Vrstva je tvořena dvěma částmi:
 - Target database – Ukládá data, která prošla transformacemi (validací, přejmenováním a jsou uspořádána do nového databázového schématu).
 - Building blocks – Jedná se o sémantické funkční bloky, které spojují Integrated Data Layer a Access Layer. Tyto bloky vytvářejí entity definované byznys pravidly (např. entita Student) a předpočítávají metriky vztažené k těmto entitám. Bloky jsou tvořeny z jednotlivých entit databázového schématu (blok pro entitu Studium atd.).
- Access Layer – Vrstva je tvořena datovými tržišti, viz podsekcce 1.2.3.
- Information Delivery Layer – Obsahuje reporty/dashboardy přístupné pro uživatele, v našem případě přes portál EBIE, ale také další nástroje.

Výhodou této architektury je v relativně rychlém a jednoduchém budování nových datových tržišť. Jelikož jsou navíc datová tržiště generována z jednotného a integrovaného datového skladu, zůstávají tak velmi konzistentní. Nevýhodou je, že se mnohdy v tomto případě jedná o finančně nákladnou a časově náročnou realizaci.

1.2.3 Datová tržiště

Součástí datového skladu jsou i datová tržiště (data marts), která jsou zastoupena v Access Layer. Datová tržiště jsou základním zdrojem pro reporty a analýzy. V návrhu datového skladu ČVUT jsou rozšířena o další metadata, například datum a čas vytvoření, požadavky autora tržiště garantujícího kvalitu apod.) Každé tržiště reprezentuje specifický pohled na data (fakta), která jsou doplňována různými dimenzemi. Fakta jsou reprezentována tzv. „faktovými tabulkami“ a dimenze tzv. „dimenzionálními tabulkami“.

Faktové tabulky v sobě obsahují analytická data. Každý řádek tabulky by měl obsahovat určitou hodnotu/míru, která by měla vyjadřovat nějakou analytickou událost (počet studentů, známka atd.). Dimenzionální tabulky popisují podniková fakta a obsahují textové atributy. Můžeme si je představit jako podnikové entity zapojené do podnikových procesů, jejichž interakcí vznikají fakta.

1.2.4 Výhody datového skladu

Datový sklad nám může přinést spoustu výhod, jeho realizace však většinou není jednoduchá. K některým výhodám můžeme zařadit například možnost vytváření ad-hoc sestav nad daty z integrovaného datového skladu a rozhodování na základě těchto dat, sledování aktuálního stavu dat, způsob zpětné kontroly kvality/čistoty dat ve zdrojových systémech a další. Integrovanost datového skladu nám zaručuje, že informace uložené v datovém skladu budou správné a jednotné. Do datového skladu se totiž dostávají data z různých systémů a bývají často replikována.

1.2.5 Využití datového skladu v rámci portálu EBIE

Portál EBIE, který si popíšeme v následující podkapitole 1.3, je nadstavbou datového skladu ČVUT a dostává tak z něj data. Data jsou prezentována pomocí datových sestav (reportů) na portálu. EBIE tedy slouží k prezentaci dat a můžeme ji tak brát jako prezentační část pro Integrated Data Layer. Co se týká propojení datového skladu a portálu EBIE, není prezentace dat jedinou vlastností – na portálu EBIE se také zobrazují metadata, popisující data a datové struktury datového skladu (mimo jiné i datová tržiště).

Důležité je upozornit, že na data z datového skladu se můžeme spolehnout tehdy, pokud jim zcela rozumíme, data jsou začištěná, víme, jak a kdy vznikla a máme pod kontrolou datové toky do datového skladu. Portál EBIE nám může pomoci s některými zmiňovanými prvky.

1.3 EBIE

EBIE (Extended Business Intelligence Encyclopedia) je portál (webová aplikace), který vznikl na základě projektu Datové čistoty v prostředí ČVUT, představený v podkapitole 1.1. BI encyklopedie slouží k udržení přehledu o datech používaných systémy v organizaci a poskytuje dokumentaci k procesům a entitám. U BI encyklopedie nejsou důležitá pouze samotná data, ale také jejich souvislosti. EBIE je oproti jiným BI encyklopediím rozšířená o ontologickou a infologickou vrstvu.

1.3.1 Koncepce návrhu

Koncepce návrhu systému obsahující portál EBIE je zobrazena na obrázku 1.1. Zmiňovaný systém je určený převážně pro vedoucí pracovníky školy, fakult, ústavů nebo kateder. Jeho přínos je rozdělen do dvou hlavních bodů:

1. **Poskytování dat pro rozhodování** – Data budou poskytována pro uživatele ve formě reportů, prezentací a zpráv, jejichž smyslem bude nabídnout kvalitní podklady pro rozhodování (operativní a strategické).

2. **Podklad pro vytváření nových legislativ a metodik** – Na základě konzistentního popisu pojmů (entit), procesů, zodpovědností a rolí bude umožněno vytvářet legislativy a metodiky určené pro řízení.

Z druhého pohledu je celé řešení nutné vyvíjet, spravovat a aktualizovat, nejen na základě požadavků od uživatelů zmíněných výše. Pro takové potřeby jsou do systému zařazeny role správců a vývojářů.

Samotná data budou poskytována z datového skladu, který není na obrázku zachycen, ale do kompletního návrhu patří. Jádrem návrhu je portál EBIE.

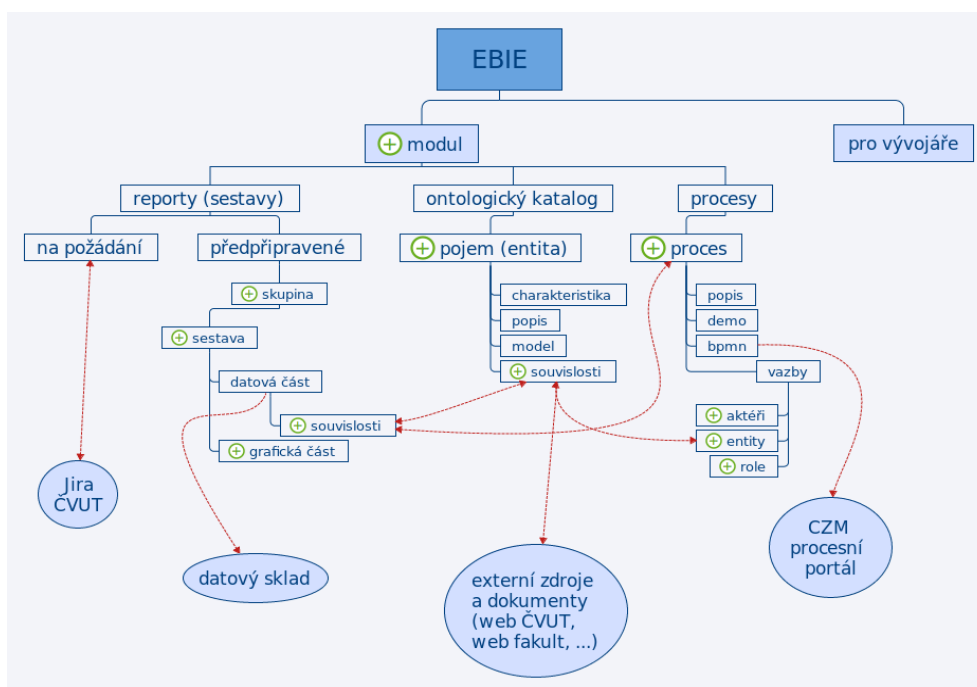
1.3.2 Informační architektura

Hlavní podstatou portálu EBIE je poskytování informací a jejich provázané prezentování uživatelům:

- **Metadata** – Pro uživatele jsou důležitá převážně byznys metadata, která popisují byznys entity seskupené do „ontologického katalogu“. Byznys entity obsahují informace zachycené jak ze strany organizace (byznys procesů), tak z věcného popisu datových entit a mají přímou vazbu na procesy a reporty. Pro uživatele slouží jako pomůcka k pochopení významu sestav, popřípadě jako podklad pro požadavky na sestavy nové či pro úpravy stávajících.
- **Procesy** – Jednotlivé procesy popisují určitou činnost organizace (např. vytvoření a schválení tématu závěrečné práce). Na portálu jsou zobrazeny ve spojitosti s byznys entitami (návaznost, stav), může u nich být přehledně vyznačeno zapojení aktérů/aktorů a jejich namapování na jednotlivé role v organizaci. Při zapojení aktorů do procesu mluvíme o tzv. ontologické transakci, se kterou se seznámíme dále v práci.
- **Reporty (sestavy)** – Jedná se o výstupy portálu, která zobrazují data v textové podobě a nebo s využitím grafických/vizualizačních prvků (tabulky, grafy apod.). Portál by měl disponovat jak jednoduchými reporty (s menším počtem parametrů), tak prostředím pro náročnější datové analýzy. Pro uživatele by tak měly být předpřipravené ukázkové sestavy. Na základě uživatelských potřeb by poté měla být možnost požádat o vytvoření nových reportů.

Prvotní návrh celé informační architektury je znázorněn „adresářovou strukturou“, viz snímek 1.3. Aktuální struktura bude zmíněna až v kapitole 3.1.1, konkrétně na obrázku 3.1, a to z důvodu zachycení provedených změn.

Na nejvyšší úrovni architektury jsou moduly (Studium, Ekonomika atd.) reprezentující specifické agendy organizace. Pod nimi se vyskytují zmiňované procesy, reporty a ontologický katalog, který se rozpadá na jednotlivé byznys



Obrázek 1.3: Informační architektura EBIE [7].

entity (pojmy). Na snímku jsou patrné vazby na externí systémy, které také patří do informační struktury.

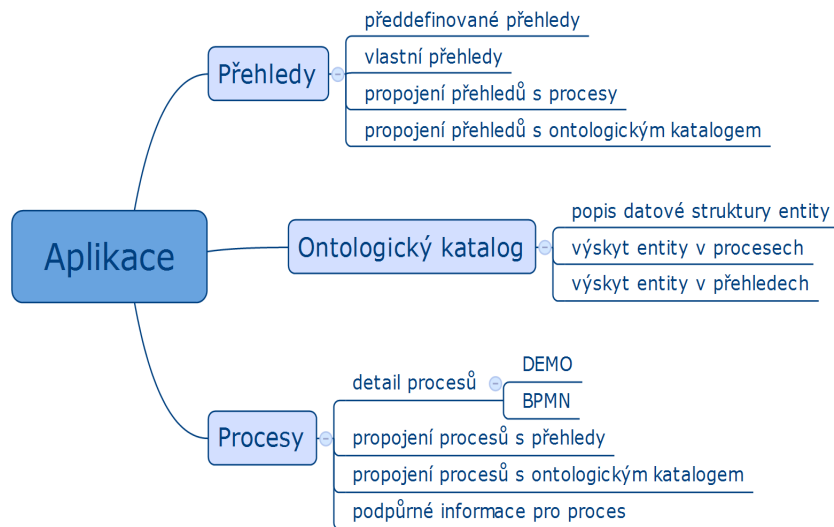
1.3.3 Související práce

V nadcházejících částech textu si popíšeme bakalářské a diplomové práce vzniklé na ČVUT, které se týkaly Datové čistoty a EBIE. Pokusíme se seznámit s základní koncepcí těchto prací a jejich výstupy.

1.3.3.1 Konceptuální analýza datových domén Studium a Hodnocení kvality výuky s ohledem na datovou čistotu

Diplomová práce „Konceptuální analýza datových domén Studium a Hodnocení kvality výuky s ohledem na datovou čistotu“ autora Ing. Václava Jirovského [16], jak již je zřejmé z názvu, mapuje konceptuální model domén Studium a Hodnocení kvality výuky v rámci datové čistoty. Na základě výsledků navrhuje vylepšení tohoto modelu.

Konceptuální model je závislý na legislativních a procesních částech domény, a proto je jedním z cílů práce najít v těchto částech chyby, které vedou ke zhoršení datové čistoty. Autor během analýzy přišel na problém u datové čistoty, kde uživatelé často nerozumí těmto datům a ke své práci tak přidal vy-



Obrázek 1.4: Informační architektura aplikace navržená Ing. Václavem Jirovským [16].

tvoření/navrhnutí rozhraní, které zpřístupní dokumentaci procesů (transakce) a informace o entitách (ontologický katalog) v přehledné podobě. Procesní část byla analyzována pomocí metodiky DEMO.

Navržená informační architektura aplikace pro uživatele na obrázku 1.4 je de facto odpovídající informační architektuře EBIE, i když autor ji v práci konkrétním názvem neoznačuje. Přehledy jsou v tomto případě myšleny reporty/sestavy EBIE. Kromě samotné architektury bylo navržené i uživatelské rozhraní (UI, user interface) aplikace, které je zachováno dodnes. Už při jeho návrhu autor počítal s možností rozšíření obsahu o vývojářskou dokumentaci, například o dokumentaci datových tržišť (data marts), ke které se dostaneme i v této práci.

Výsledky práce potvrdily, že:

- Prototyp datového skladu je navržen správně.
- Existují legislativní a procesní rozpory s konceptuálním modelem (vztahy a atributy entit), které je potřeba vyřešit, jinak by mohly narušit datovou čistotu.
- Je nutné chápat stejným způsobem entity a jejich implementaci v různých systémech, čili konzistentní definice.
- Uživatelům je důležité poskytnout nástroj pro přístup k dokumentaci přehledů (sestav), procesů a entit. Tento nástroj vyústil v portál EBIE.

1.3.3.2 Ontologická analýza datového skladu ČVUT

Druhá diplomová práce „Ontologická analýza datového skladu ČVUT“, jejímž autorem je Ing. Ladislav Moravec [17], měla za cíl ontologickou analýzu agend v rámci ČVUT, konkrétně agendu Anketa ČVUT a agendu Závěrečných prací. Pan Moravec poukázal na fakt, že současné BPMN procesy nejsou úplně přesné, a proto ve své práci použil metodologii DEMO, která by měla prohloubit analýzu o infologickou, datalogickou a samozřejmě ontologickou úroveň. Výsledné DEMO diagramy by měly sloužit jako zdroj pro EBIE.

Autorovi se povedlo zpracovat zvolené agendy na všech třech úrovních a za použití DEMO metodologie docílit výsledku, že vypracované diagramy nejsou nijak platformově závislé a je u nich patrná jejich esence. S rozpoznatelnou esencí není nutné přidávat další upřesňující textové popisky k těmto procesům. U všech procesů agendy Ankety ČVUT a Závěrečné práce byly vytvořeny OCD a PSD diagramy s přidruženými TPT tabulkami (všechny tyto pojmy jsou vysvětleny v kapitole 4). V kontrastu s BPMN přidává vytvoření DEMO diagramů a tabulek zcela nový pohled na procesy.

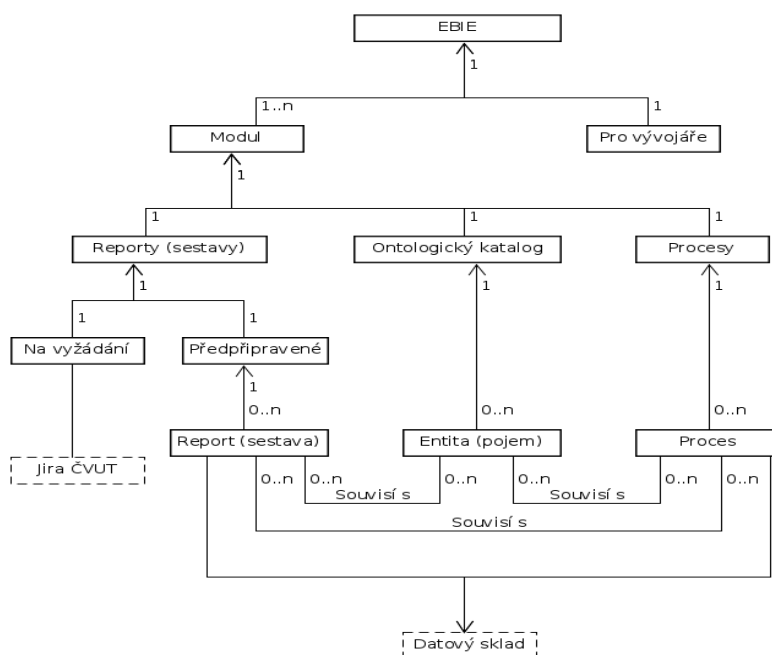
1.3.3.3 Návrh a implementace backend projektu EBIE

Bakalářská práce „Návrh a implementace backend projektu EBIE“ od Bc. Davida Hajčiar [18] se zabývá backend částí portálu EBIE, konkrétně návrhem konceptuálního datového schématu, viz obrázek 1.5, a vyřešením obsahových požadavků (ze seznamu jsou vybrány ty nejdůležitější):

- Úprava architektury EBIE.
- Oddělení aplikační logiky od obsahu EBIE.
- Nalezení vhodné technologie pro správu obsahu EBIE, která by splňovala zadané požadavky (přehlednost, možnost jednoduchého upravování obsahu a přidávání dalšího obsahu apod.).
- Návrh a začlenění technických a business metadat datového skladu do obsahu EBIE.

Autorovi se podařilo všechny tyto požadavky splnit – navrhl a implementoval jejich řešení. Pro celkovou správu obsahu zvolil technologii AsciiDoctor. Jedná se o rychlý textový procesor a publikační nástroj pro konvertování mezi formátem AsciiDoc na HTML5, popřípadě na jiný formát. AsciiDoctor je napsaný v jazyce Ruby [19].

Jelikož AsciiDoctor pracuje s formátem AsciiDoc, pan Hajčiar se rozhodl převést obsah entit (ontologického katalogu) a procesů do tohoto formátu a vyrobil základní šablony pro usnadnění aktualizace obsahu. Oddělení aplikační logiky od obsahu EBIE se autorovi povedlo pomocí technologie git submodule.



Obrázek 1.5: Konceptuální datové schéma EBIE vytvořené Bc. Davidem Hajčiarem [18].

Dalším vypracovaným úkolem bylo vytvoření prototypu pro transformaci metadat z datového skladu do EBIE.

Velmi významným výstupem popisované práce je generátor napsaný v Ruby on Rails ⁶, na kterém je postavena webová aplikace EBIE. Vytvořený generátor umožňuje automaticky vykonat operace pro převod mezi AsciiDoc a HTML, transformaci metadat a uložení času změny obsahu a verzi aplikace.

1.3.3.4 Návrh a implementace integrace služeb do projektu EBIE

„Návrh a implementace integrace služeb do projektu EBIE“ je další bakalářská práce zaměřující se na EBIE, jejímž autorem je Bc. Cyril Černý [21]. Tato práce se zabývá integrací dvou prvků do portálu EBIE:

- Manažerem identit na ČVUT (ČVUT IdM), při kterém se provede i analýza existujících byznys rolí. K přihlašování je použit autentizační server (OAAS) Zuul [22].
- Systémem JIRA, ve kterém autor řeší mimo jiné analýzu poskytovaného API (zde REST JIRA API), aby mohl zajistit přístup k tzv. ticketům ⁷.

⁶Ruby on Rails (Rails) je framework napsaný v jazyce Ruby [20].

⁷Tickety se používají k evidenci požadavků, chyb a problémů při vývoji software.

Výsledkem práce bylo vytvoření prototypové demo aplikace, která demonstruje zmiňované integrace. Pro propojení s ČVUT IdM byly použity již existující struktury rolí a pro autentizaci a získání rolí uživatele byly aplikovány OAAS Zuul a Usermap API. Pro spojení se systémem JIRA byla navržena komunikace s JIRA REST API. Výhodou výsledku je, že portál EBIE může spravovat požadavky uživatelů, aniž by samotní uživatelé museli využívat webové rozhraní pro zpracování požadavků (ČVUT Helpdesk).

Implementovaná aplikace však dle závěrů není určena k nasazení a pouze demonstruje navrhovaná řešení, konkrétně činnosti:

- Integrace s manažerem identit ČVUT:
 - Přihlášení přes ČVUT účet (provedené spojení s OAAS Zuul).
 - Vypsání seznamu rolí přihlášeného uživatele (role jsou získány z Usermap API).
 - Základní autorizace ke zdrojům (použití převzatých rolí).
- Integrace s ticketovacím systémem JIRA:
 - Vypsání seznamu ticketů vytvořených přihlášeným uživatelem.
 - Založení nového ticketu s vlastním nadpisem.
 - Uzavření/vyřešení ticketu.
 - Přejít do systému JIRA k detailům zvoleného ticketu.

Posledním výstupem, který je třeba zmínit, je vytvoření tzv. whitelistu, který obsahuje jméno a seznam přidělených rolí a slouží k udělení výjimek pro přístup do portálu.

1.3.3.5 Návrh a implementace klientské části systému EBIE

Poslední analyzovanou prací je bakalářská práce „Návrh a implementace klientské části systému EBIE“ od Bc. Michala Kopeckého [23] zabývající se vizualizací datových sestav z datového skladu ČVUT v portálu EBIE. Autor využívá k zobrazování sestav nástavbu nad knihovnu D3.js jazyka JavaScript [24], která umožní vykreslovat data s libovolnou strukturou do předpřipravených koster grafů. Datové sestavy jsou získány pomocí API aplikace a pracují s upravenými daty ve formátu JSON. Před tvorbou práce nebyla struktura dat pevně daná, a proto dalším úkolem autora bylo upravení této API aplikace, aby odpovídala struktuře navržených reportů.

Datové sestavy (prototypy) jsou navrženy na základě funkčních a nefunkčních požadavků, analýzy reportů ve smyslu jejich přehlednosti a analýzy možnosti grafového a tabulkového zobrazení. Zmiňované požadavky byly například:

- **Funkční požadavky** – Řazení dat, vyhledávání, zobrazení dat (výběr typu zobrazení), export dat apod.
- **Nefunkční požadavky** – Zobrazení v prohlížeči a responsivita.

Na základě proběhlých analýz autor došel k závěru, že si uživatel bude volit vstupní parametry sám a k zobrazení bude vybrána jen část datové sestavy. Po vybrání typu reportu a konkrétních vstupních parametrů proběhne komunikace s API aplikací, pomocí které se získají data. Po úspěšném získání dat se zobrazí sekce s filtry, které bude možné kombinovat.

Hlavní výstupy práce jsou tedy prototypy reportů, které svým způsobem udávají, jakým způsobem by se mělo v budoucnu při implementaci nových sestav postupovat. Reporty jsou vytvořené nad datovými sestavami „Výsledky předmětů“ a „Počet studií“.

Projektový tým, nasazení a integrace

V této kapitole si představíme projektový tým, který se zaměřoval, a stále zaměřuje, na rozvoj portálu EBIE. Popíšeme si, jak probíhaly schůzky týmu a co se na nich řešilo. V druhé části se zaměříme na průběh nasazení portálu EBIE a integrace jednotlivých částí projektu.

2.1 Projektový tým

Portál EBIE vznikl jako výstup projektu „Datové čistoty“, který jsme si popsali v podkapitole 1.1. Základy portálu byly vytvořeny, na některých částech se podíleli i samotní studenti ČVUT, viz 1.3.3, ale celý projekt zdaleka nebyl dotažen a bylo na něm potřeba udělat další kus práce.

2.1.1 Tým EBIE

Na základě požadavků od Ing. Michala Valenty, Ph.D., který je vedoucím projektu „Datové čistoty“ a portálu EBIE, se utvořil tým, jenž má na starosti další vývoj portálu EBIE. Tým budeme dále nazývat vystihujícím názvem „Tým EBIE“. Tým EBIE tvoří následující členové:

- **Ing. Michal Valenta, Ph.D.** – Zakladatel týmu, zaměstnanec ČVUT a hlavní vedoucí projektu, který má na starosti všechny formální náležitosti, prezentování a propagaci projektu na úrovni vedení ČVUT.
- **Bc. Ondřej Batík** – Autor práce a současný student ČVUT, který v týmu zastává roli projektové manažera. V rámci projektu jsou jeho hlavními prioritami organizace schůzek, tvorba zápisů ze schůzek, podílení se na diskusi o projektu a návrhu změn. Další zaměření na projektu je náplní této práce.

- **Bc. David Hajčiar** – Současný student ČVUT, který je v týmu v roli programátora/testera. Jeho hlavní oblastí je část projektu týkající se obsahu EBIE. Na schůzkách přispívá do diskuse a podílí se na návrhu změn. V minulosti se podílel na tvorbě projektu – jeho výstupem je bakalářská práce popsána v části 1.3.3.3.
- **Bc. Cyril Černý** – Další současný student ČVUT, který je v týmu také v roli programátora/testera. Na rozdíl od kolegy Hajčiara, jeho hlavním zaměřením je v současnosti aplikační část portálu EBIE. V rámci schůzek přispívá do diskuse a podílí se na návrhu změn, stejně jako ostatní členové. Na projektu již spolupracoval a vytvořil bakalářskou práci, která byla představena v části 1.3.3.4.

2.1.2 Tým datových skladníků

Důležité je zmínit i druhý tým, konkrétně „Tým datových skladníků“, který má na starosti správu a vývoj datového skladu ČVUT, představeného v podkapitole 1.2. Portál EBIE je nadstavbou nad tímto skladem a čerpá z něj data nejen pro datové sestavy. V současnosti mají někteří členové na starosti také částečné plnění obsahu EBIE v oblasti datových tržišť, metadat a dalších náležitostí vztahujících se k datovému skladu. Součástí Týmu datových skladníků jsou:

- Ing. Magda Friedjungová
- Ing. Stanislav Kuznetsov
- Bc. Jakub Krejčí
- Bc. Robert Kotlář

2.1.3 Schůzky

V průběhu školního roku 2016/2015 na ČVUT probíhaly schůzky týmu/týmů, na kterých se členové zabývali především dalším vývojem portálu EBIE. Ke každé schůzce byly vytvořeny formální zápisy, popisující rozebraná témata a návrhy všech změn, přítomnost jednotlivých účastníků a jejich případné úkoly. Některé úkoly se řešily/demonstrovaly rovnou na místě schůzek (kancelář A-927, areál ČVUT).

K únoru roku 2017 se uskutečnily tyto schůzky:

- **14. listopadu 2016** – První společná schůzka, na které byli přítomni všichni členové Týmu EBIE. Začala se řešit integrace a nasazování portálu na server (viz 2.2), whitelist a práva, struktura EBIE a další.

- **6. prosince 2016** – Druhá schůzka celého Týmu EBIE, na které se pokračovalo v integraci projektů, řešila se správa těchto projektů a testovalo se publikování/měnění obsahu portálu EBIE.
- **13. prosince 2016** – Schůzka Týmu EBIE s Týmem datových skladníků, které se účastnili jen někteří členové. Hlavním tématem bylo prezentování postupu úpravy obsahu portálu Týmu datových skladníků. Dále se probírala témata datových tržišť, datových zdrojů, reportů, úpravy entit a metadata.
- **18. ledna 2016** – Aktuálně poslední schůzka Týmu EBIE. Na tomto sezení se převážně řešila práva přístupu do portálu, rozdělení serverů pro nasazení, aktuálně použitelné reporty a technologie používané (navrhované k použití) na portálu.

2.2 Průběh nasazení a integrace

Jedním z cílů této práce bylo vytvořit plán integrace a realizovat pilotní nasazení portálu EBIE z pozice projektového manažera. V následujících částech textu si tyto úkoly popíšeme.

2.2.1 Integrační plán a plán nasazení

Plán integrace a nasazení je znázorněn souhrnným Ganttovým diagramem, který můžeme vidět na snímku 2.1. Ganttův diagram (Gantt Chart) slouží pro grafické znázornění naplánované posloupnosti činností v čase, které se využívá při řízení různých projektů [25]. Pro tvorbu návrhu byl použit program GanttProject, dostupný na adrese <http://www.ganttproject.biz/>.

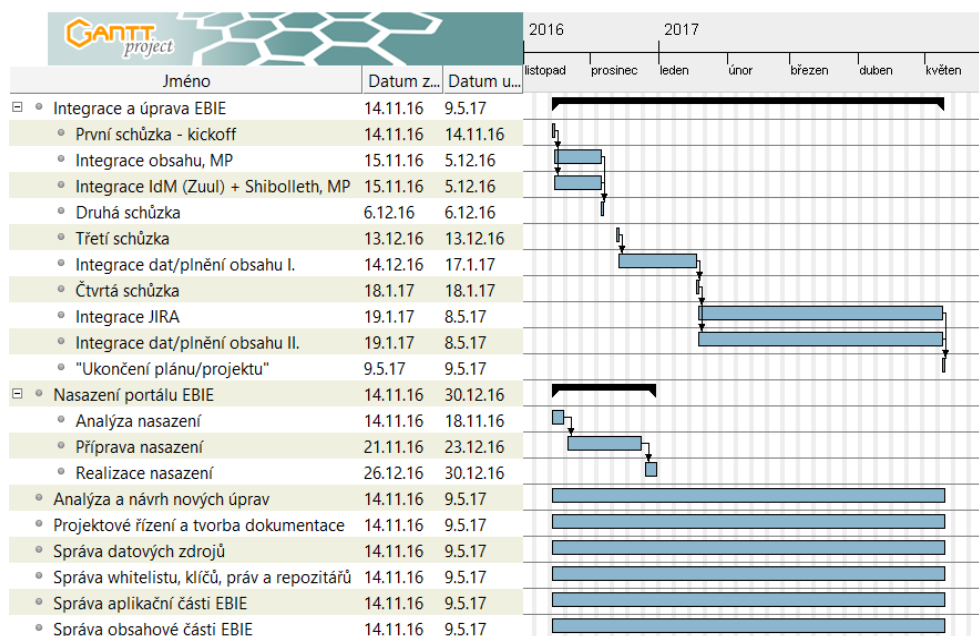
Na realizaci projektu se podílel celý Tým EBIE 2.1.1 a částečně i Tým datových skladníků 2.1.2, který měl na starosti převážně propojení datových zdrojů a plnění obsahu.

2.2.1.1 Rozbor jednotlivých částí

Pro integrační část byly vybrány činnosti, které byly relevantní pro aktuální situaci stavu EBIE – například některé části z prací byly již do portálu propojeny a nebylo je potřeba zohledňovat při navrhovaném plánu, případně byly součástí integrace jiných částí. Analýza souvisejících prací (podsektce 1.3.3) z hlediska integrace je shrnuta v seznamu:

- **Práce Ing. Václava Jirovského analyzovaná v části 1.3.3.1** – Návrh UI zohledněn při vytváření prototypu EBIE a importu dat (procesů, entit atd.) vzešlých z autorovy práce proběhl již v minulosti, nicméně v plánu bude součástí integrace obsahu.

2. PROJEKTOVÝ TÝM, NASAZENÍ A INTEGRACE



Obrázek 2.1: Plán popisující integraci a nasazení portálu EBIE.

- **Práce Ing. Ladislava Moravce analyzovaná v části 1.3.3.2** – Procesy, které byly vytvořeny v práci a počítá se s jejich přítomností v portálu EBIE, je nutné zapojit do integrační části projektu.
- **Práce Bc. Davida Hajčiara analyzovaná v části 1.3.3.3** – Velkou část integračních požadavků provedl autor při tvorbě práce, a proto není nutné zařazovat do plánu. Některá řešení je však nutné lehce upravit.
- **Práce Bc. Cyrila Černého analyzovaná v části 1.3.3.4** – Hlavním cílem plynoucí z autorovi práce je provést propojení portálu s ČVUT IdM a systémem JIRA.
- **Práce Bc. Michala Kopeckého analyzovaná v části 1.3.3.5** – Datové sestavy byly v minulosti importovány, reporty byly zprovozněny. V souvislosti zavedení do nového stavu budou zahrnuty v části integrace obsahu. Je důležité se zaměřit také na datové zdroje těchto sestav.

Jak již bylo zmíněno v prvním odstavci této podkapitoly, do integračního plánu a plánu nasazení bylo potřeba zahrnout především relevantní činnosti, které vyplynuly z aktuálních požadavků shromážděných na proběhlých schůzkách. Z důvodu větší přehlednosti byly některé úkoly přidány do souhrnných

činností. Všechny schůzky zmiňované v návrhu odpovídají schůzkám z podkapitoly 2.1.3. Ostatní části plánu, které můžeme vidět na snímku 2.1, jsou stručně popsány v následujících bodech:

- **Integrace obsahu, MP** – Součástí integrace obsahu je obecně spojení obsahové části portálu EBIE. Jedná se například o datové sjednocení reportů, entit, procesů, které jsou zaneseny do kompletní adresářové struktury (JSON soubory, AsciiDoc soubory apod.), generátoru a další. MP značí „merge projektů“, čili sloučení projektů do takové podoby, aby měly požadovanou strukturu.
- **Integrace IdM (Zuul, whitelist), MP** – Na základě výstupu práce rozebrané v části 1.3.3.4 bylo rozhodnuto, že se provede propojení Zuul (server pro autorizaci uživatelů, který využívá připojení přes LDAP ⁸ a řeší integraci s Shibboleth ⁹). Správa uživatelských rolí/uživatelů bude prozatím řešena přes whitelist, dokud se nevyřeší ostatní náležitosti (práva atd.).
- **Integrace dat/plnění obsahu I.** – Jedná se o plnění obsahu novými daty – seznam target tabulek, datamarty, technické atributy tabulek a další metadata doplněná z datového skladu ČVUT.
- **Integrace JIRA** – Propojení systému JIRA pro podávání žádosti o novou sestavu jednotlivými uživateli s portálem EBIE. Aktuálně je konec úkolu nastaven na den ukončení plánu, viz dále.
- **Integrace dat/plnění obsahu II.** – Jedná se o plnění obsahu novými daty – relevantní procesy pro portál vytvořené v práci Ing. Ladislava Moravce, představené v části textu 1.3.3.2, a procesy vzniklémi v této práci, více v kapitole 4. Konec úkolu je nastaven stejně jako v předchozím případě na den ukončení plánu.
- **„Ukončení plánu/projektu“** – Symbolický milník ukončení plánu, který byl vybrán na datum 9. 5. 2017, kdy má dojít k řádnému odevzdání této práce. Pro některé úkoly to však nemusí být nutný bod ukončení, jelikož celý projekt vývoje portálu bude pokračovat i po vypracování práce.
- **Analýza nasazení** – První část nasazení, ve které je hlavním úkolem analyzovat a navrhnout průběh nasazení – jak a kde nasazení proběhne, co je k tomu potřeba a další.

⁸Lightweight Directory Access Protocol je otevřený protokol pro uchovávání a získávání dat z adresářové struktury [26].

⁹Shibboleth je open-source systém, který umožňuje ověřeným uživatelům přístup pomocí HTTP protokolu k poskytovatelům služeb [27]

- **Příprava nasazení** – Ve skutečnosti se tato činnost prolínala s jinými činnostmi ve smyslu, že některé úkoly z ostatních částí spadaly i do této části a naopak. Příprava nasazení byla určena pro instalování technologií na server/servery, předpřípravení konfiguračních souborů, vytváření potřebných účtů, nastavení repozitářů a další potřebné úkony s GitLab apod.
- **Realizace nasazení** – Poslední úsek nasazení, ve kterém jde o samotnou realizaci nasazení portálu EBIE na server/servery – nastavení obou serverů, viz sekce 2.3, do stejného a funkčního stavu a dořešení patřičných úkolů.
- **Analýza a návrh nových úprav** – První činnost z činností (tato a všechny následující), které jsou zobrazeny/nastaveny po celou dobu průběhu plánu. Do analýzy a návrhů nových úprav jsou zařazeny všechny myšlenky, které se týkaly nápadů o rozšíření a upravení portálu EBIE.
- **Projektové řízení a tvorba dokumentace** – Pro plán/projekt je důležité mít projektového manažera, který se stará o jeho úspěšné dokončení. Nemálo důležitá je i průběžná tvorba dokumentace proběhlých změn v projektu, výsledného řešení a vytváření zápisů ze schůzek, které spadají do projektového řízení.
- **Správa datových zdrojů** – Pro portál EBIE jsou důležitá data z datového skladu ČVUT, která jsou používána v reportech. V plánu musíme počítat se správou těchto zdrojů.
- **Správa whitelist, klíčů, práv a repozitářů** – Aktuální přístup do portálu je řešen přes whitelist, který je nutné udržovat. Podobnou pozornost musíme věnovat také SSH klíčům, právům pro GitLab a správě vytvořených repozitářů.
- **Správa aplikační části EBIE** – Úkoly spojené s aplikační částí portálu EBIE se budou vyskytovat během celého integračního plánu. Konkrétně mluvíme o malých úkolech, které přímo nesouvisí s nasazením, ani integrací a zaměřují se spíše na vývoj portálu, popřípadě je jejich provedení nutné pro další úkoly zabývající se integrací či nasazením.
- **Správa obsahové části EBIE** – Stejně jako v případě předchozí aplikační části je také důležitá správa u obsahového segmentu portálu EBIE.

Časový úsek jednotlivých činností je vidět z plánu zobrazeném na snímku 2.1, kde druhý sloupec značí datum začátku a třetí sloupec datum ukončení činnosti. Pro lepší představu těchto úseků je plán graficky znázorněn v pravé části zmiňovaného snímku.

Důležité je znovu zdůraznit, že po „Ukončení plánu/projektu“ nedochází k úplnému konci projektu, jelikož vývoj portálu EBIE bude pokračovat i v budoucnu.

2.2.1.2 Rozdělení činností

Pro jednotlivé činnosti bychom měli vybrat odpovědné osoby, čili osoby, kterých se daná činnost bude nějakým způsobem týkat. K takovým účelům je vhodné využít tzv. RACI matice. Následující úryvky textu o RACI maticích jsou převzány ze zdroje [28].

Matice odpovědnosti RACI (RACI matice) je metoda pro přiřazení a zobrazení odpovědností jednotlivých osob či pracovních míst v nějakém úkolu (např. projektu) v organizaci. RACI je akronym z počátečních velkých písmem slov:

- **R – Responsible** – Osoba je odpovědná za vykonání přiděleného úkolu.
- **A – Accountable** – Osoba je odpovědná za celý úkol - je odpovědná za to, co je vykonáno.
- **C – Consulted** – Osoba, která může poskytnout cennou radu či konzultaci k úkolu.
- **I – Informed** – Osoba, která má být informována o průběhu úkolu či o rozhodnutích v úkolech.

RACI maticí tedy vhodně přiřadíme k jednotlivým činnostem odpovědnosti členů týmu. Metoda RACI je jednoduchou formou modelu kompetencí. Na obrázku 2.2 je zobrazena RACI matice pro náš plán/projekt – řádky odpovídají jednotlivým činnostem, sloupce jednotlivým členům (první písmeno z jména, první písmeno z příjmení) kromě posledního, ke kterému je přiřazen celý Tým datových skladníků představený v části 2.1.2. Znak „-“ vyznačuje, že pro danou činnost a osobu není přidělena žádná odpovědnost z odpovědností vyskytujících se v RACI matici.

U RACI matice pro integrační plán a plán nasazení byl u všech schůzek zvolen stav Responsible v případě, že se daná osoba schůzky účastnila. Pro jednu činnost může mít více osob stav Responsible, nicméně nemůže nastat situace, kdy mají dvě osoby pro stejnou činnost stav Accountable. Jedna osoba může navíc zastávat více stavů pro jednu konkrétní činnost (v našem případě A/R). Přiřazení odpovědností u ostatních činností odpovídá skutečnosti, jak plán probíhal/probíhá.

2. PROJEKTOVÝ TÝM, NASAZENÍ A INTEGRACE

	MV	OB	CČ	DH	Tým d.s.
První schůzka - kickoff	R	A/R	R	R	I
Integrace obsahu, MP	I	I	C	A/R	-
Integrace IdM (Zuul, whitelist), MP	I	I	A/R	C	-
Druhá schůzka	R	A/R	R	R	I
Třetí schůzka	R	A/R	I	I	R
Integrace dat/plnění obsahu I.	I	I	I	I	A/R
Čtvrtá schůzka	R	A/R	R	R	I
Integrace JIRA	I	I	A/R	C	-
Integrace dat/plnění obsahu II.	I	A/R	I	I	R
"Ukončení plánu/projektu"	R	A/R	R	R	I
Analýza nasazení	R	A/R	R	R	-
Příprava nasazení	C	A	R	R	-
Realizace nasazení	C	A	R	R	I
Analýza a návrh nových úprav	A/R	R	R	R	-
Projektové řízení a tvorba dokumentace	R	A/R	-	-	-
Správa datových zdrojů	I	I	I	I	A/R
Správa whitelist, klíčů, práv a repozitářů	A/R	R	R	I	-
Správa aplikační části EBIE	I	I	A/R	C	-
Správa obsahové části EBIE	I	I	C	A/R	-

Obrázek 2.2: RACI matice integračního plánu a plánu nasazení portálu EBIE.

2.3 Vyhodnocení plánu nasazení

Nasazování portálu EBIE probíhalo bez větších problémů. V „Analýze nasazení“ bylo rozhodnuto, že pilotní nasazení bude provedeno na dva servery VICu¹⁰:

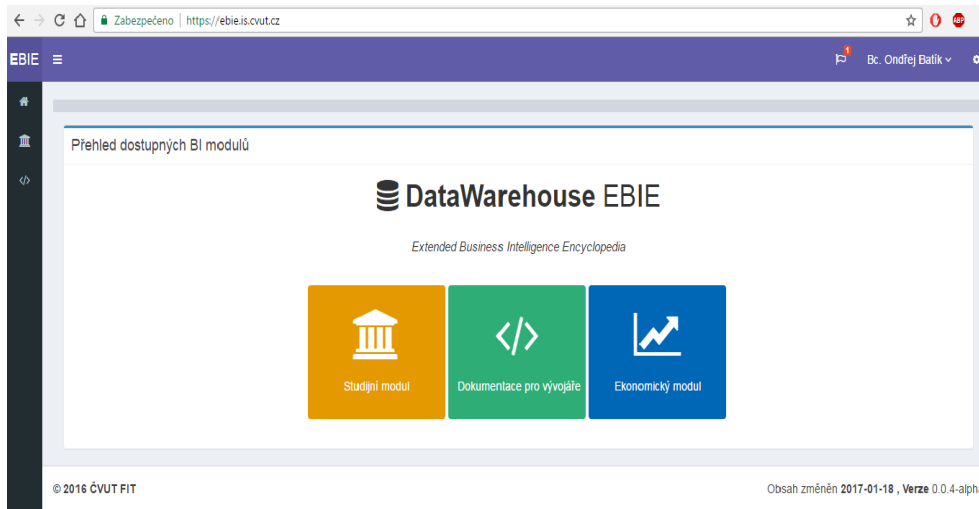
- **Server „EBIE-vývoj“** – Testovací server portálu EBIE, který je dostupný na webové adrese <https://ebie-vyvoj.is.cvut.cz/>.
- **Server „EBIE“** – Produkční server portálu EBIE, který je dostupný na webové adrese <https://ebie.is.cvut.cz/>, viz obrázek 2.3.

Portál EBIE se podařilo úspěšně nasadit na oba zmiňované servery. Při přípravě serverů bylo provedeno větší množství menších úkolů, především zaměřených na instalaci potřebných technologií, například: Ruby/Ruby on Rails, Nginx, Thin, PostgreSQL a Git. Dále bylo nutné vytvořit nové účty, zaregistrovat distribuční klíče do všech repozitářů, zapnout protokol SSL/TLS, předpřipravit konfigurační soubory a další. Všechny činnosti spojené s integrační částí proběhly prozatím v pořádku (k únoru roku 2017).

Během nasazování portálu EBIE a jeho integrační části došlo k několika změnám, které se týkaly zejména architektury EBIE. Aktuální informační

¹⁰Výpočetní a informační centrum ČVUT.

2.3. Vyhodnocení plánu nasazení



Obrázek 2.3: Produkční verze portálu EBIE.

strukturu, adresářovou/obsahovou strukturu a strukturu portálu na GitLabu si popíšeme v následující kapitole.

Aktuální stav EBIE a analýza procesů

Od původního návrhu portálu EBIE rozebraném v podkapitole 1.3 uběhla poměrně dlouhá doba a došlo k několika změnám. Hlavním podnětem těchto změn byly požadavky na rozšíření portálu, ale také požadavky nutné k pilotnímu nasazení portálu a zavedení integračních částí, viz část textu 2.2.1.1. V první podkapitole se zaměříme na tyto změny portálu EBIE, které se týkají zejména architektury. Následně se podíváme, jak vypadá struktura stránky popisující proces. V druhé podkapitole provedeme analýzu procesů spojených se správou obsahu portálu EBIE.

3.1 Aktuální stav portálu EBIE

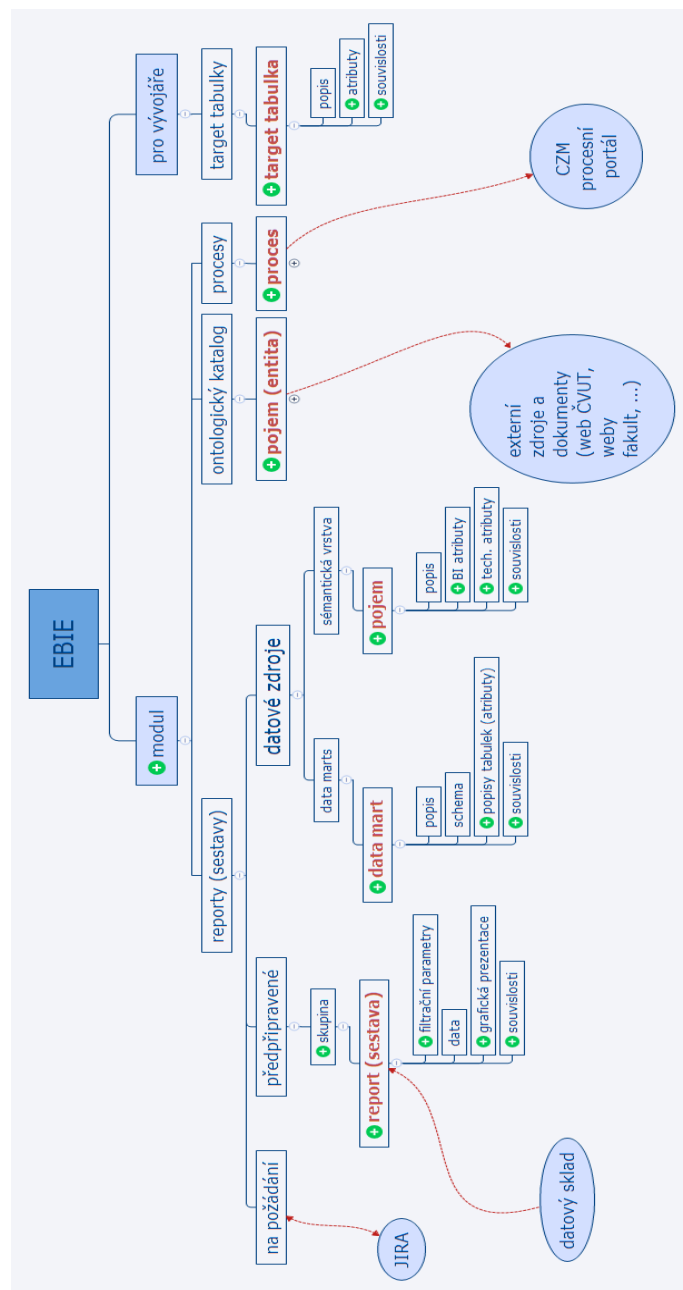
V následujících odstavcích poukážeme na nejdůležitější změny portálu EBIE, které se udály v období několika posledních měsíců.

3.1.1 Informační struktura portálu

Původní návrh informační struktury je znázorněn na snímku 1.3. Pokud porovnáme původní strukturu s aktuální, která je zobrazena na obrázku 3.1, lehce poznáme vytvořené změny. Na obou snímcích označuje ikonka znaku „+“ v kroužku více výskytů daného typu uzlu.

První viditelnou změnou je přidání sekce „target tabulek“ do části pro vývojáře. Druhou odlišností je rozšíření sekce reportů (sestav) o dokumentaci datových zdrojů. Datové zdroje se dále rozpadají na části „data marts“ a „sémantickou vrstvu“.

3. AKTUÁLNÍ STAV EBIE A ANALÝZA PROCESŮ



Obrázek 3.1: Aktuální informační architektura EBIE [9].

 t_stpl_studijni_obor

Název atributu	Datový typ	Zdrojový systém	Popis
studijni_obor_bk	bigint	KOS	identifikátor studijního oboru
nazev_cs	varchar(100)	KOS	název studijního oboru (česky)
nazev_en	varchar(100)	KOS	název studijního oboru (anglicky)

Obrázek 3.2: Část atributů tabulky t_stpl_studijni_obor.

3.1.2 Target tabulky

Target tabulky (na portálu „Seznam target tabulek“) je nová sekce určená pro vývojáře, která v sobě obsahuje jednotlivé target tabulky vyskytující se v Target databázi v datovém skladu ČVUT. Obsah stránky na portálu popisující target tabulky je dále strukturován do následující podoby:

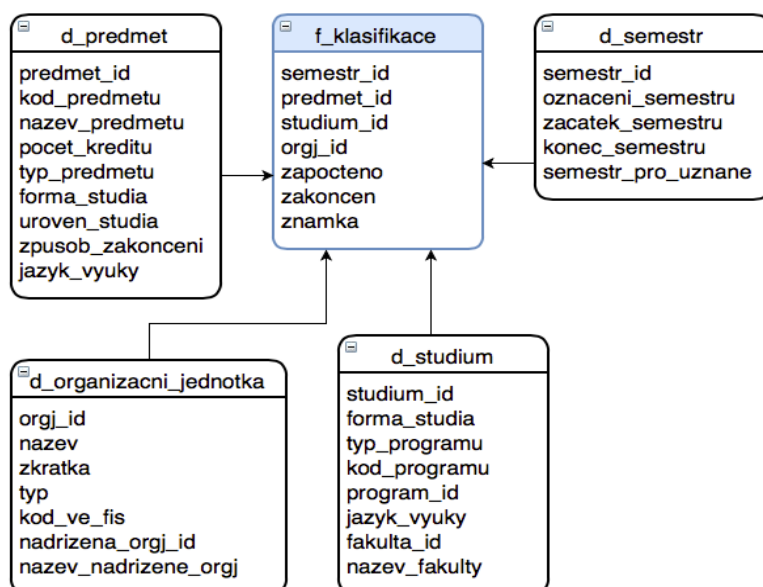
- **Popis** – Slovní popis tabulky – stručné popsání vztahujících se entit.
- **Atributy** – Specifické atributy (sloupce), které popisují target tabulky. Aktuálně jsou zobrazovány sloupce „název atributu“ (název konkrétního atributu target tabulky), „datový typ“, „zdrojový systém“ a „popis“ (vysvětlení daného atributu target tabulky), viz snímek 3.2.
- **Souvislosti** – Do souvislostí spadají názvy jiných tabulek, se kterými daná tabulka souvisí, a pokud jsou tyto tabulky do portálu EBIE zaneseny, pak také patřičné odkazy. Ostatní souvislosti jsou zmíněny v části popisu tabulky – odkazy na související entity.

3.1.2.1 Data marts

Složka datová tržiště (data marts) v sobě obsahuje jednotlivá datová tržiště, která jsou důležitá nejen pro tvorbu reportů na portálu EBIE. Datová tržiště můžeme brát jako produkt datového skladu, který může být dále nabízen třetím stranám, které by o ně projevíly zájem. Výhodou portálu EBIE je, že dává kontext o datovém skladu a tedy rovněž o datových tržištích. Třetí strana si na portálu přečte dokumentaci o datových tržištích a ví, co může očekávat.

Na stránce popisující datové tržiště jsou poskytovány informace:

- **Popis** – Krátký slovní popis datového tržiště – jaká data poskytuje.



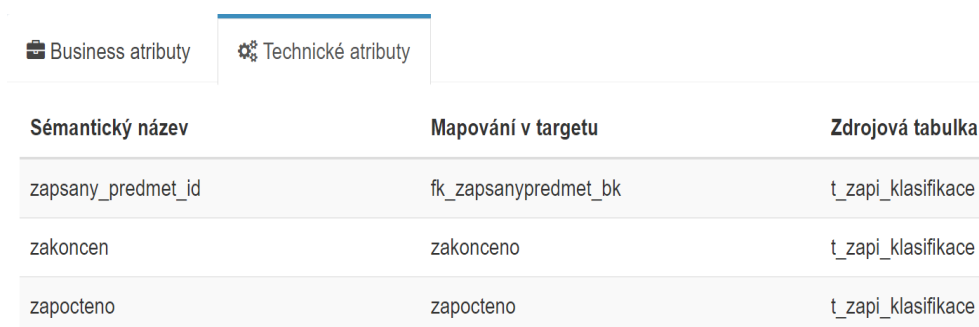
Obrázek 3.3: Ukázka schématu datového tržiště.

- **Schéma** – Schéma dimenzionálních a faktových tabulek datového tržiště, například ukázka na snímku 3.3.
- **Popisy tabulek (atributy)** – Specifické atributy (sloupce) které popisují každou tabulku ze schématu. Každý atribut tabulky je popsán „názevem“ (konkrétní název atributu) a „popisem“ (slovní popis/význam atributu).
- **Souvislosti** – Patří sem například odkazy na reporty, které využívají dané datové tržiště a příslušející pojmy ze sémantické vrstvy (odkazy na pojmy).

3.1.2.2 Sémantická vrstva

Ontologický katalog poskytuje pojmy, které jsou skutečné (vytváření skutečnost) a důležité pro organizaci, ale datový sklad o nich nemusí poskytovat žádné informace. Pod sémantickou vrstvou spadají vybrané pojmy, které jsou detailněji popsány z pohledu datového skladu.

- **Popis** – Krátký slovní popis pojmu – jeho vysvětlení.
- **BI atributy** – Pod BI (zde business/byznys) atributy patří „sémantický název“ (atribut tabulky daného pojmu) a „business popis“ (business význam atributu).



Sémantický název	Mapování v targetu	Zdrojová tabulka
zapsany_predmet_id	fk_zapsanypredmet_bk	t_zapi_klasifikace
zakoncen	zakonceno	t_zapi_klasifikace
zapocteno	zapocteno	t_zapi_klasifikace

Obrázek 3.4: Část technických atributů pojmu Klasifikace.

- **Technické atributy** – Do technických atributů spadá „sémantický název“, „mapování v targetu“ (příslušný název atributu v target tabulce) a „zdrojová tabulka“ (název target tabulky, do které atribut patří).
- **Souvislosti** – Odkazy na vztahující se pojmy (entity) a datová tržička, pod která pojem náleží.

3.1.3 Adresářová struktura portálu

Návrh nové adresářové struktury od Bc. Davida Hajčiara, který je uveden v jeho bakalářské práci analyzované v části textu 1.3.3.3, již není aktuální, a proto si popíšeme největší změny, ke kterým došlo. Kompletní adresářová struktura je z důvodu velikosti rozdělena do dvou snímků B.1 a B.2 zobrazených v Přílohách.

Od navrhované verze z bakalářské práce došlo k těmto nejvýznamnějším změnám:

- V kořenovém adresáři `htmls` došlo k přidání adresáře `sources`, který obsahuje HTML soubory potřebné k testovacímu generování vytvořených souborů.
- Generování je vykonávané nově pomocí souboru `generate.ruby` (dříve `generate.sh`).
- Do `ebie-content` přibyl modul dokumentace se sekci `table` (pro target tabulky).
- Do modulu `studium`, konkrétně do adresáře `reports`, byla přidána složka `data_sources`.

3.1.4 Struktura GitLab z hlediska portálu

Ke správě verzí a vývoje se pro portál EBIE používá GitLab. GitLab je software, který poskytuje management repozitářů, podporu pro sledování chyb a dokumentaci wiki [29]. V současnosti z pohledu portálu EBIE máme tři hlavní repozitáře:

- **ebie-content** – Obsahuje kompletní obsah portálu, viz podsekcce 3.1.3, který je zobrazován. Správci obsahu tak mohou upravovat obsah bez nutnosti znát, jak celý portál EBIE funguje. Tento repozitář obsahuje definice jednotlivých stránek ve formě AsciiDoc souborů. Reporty od Bc. Michala Kopeckého, viz část 1.3.3.5, byly přesunuty do této části, stejně jako data (entity, procesy atd.) od Ing. Václava Jirovského, viz část 1.3.3.1.
- **ebie-data-api** – Propojuje webovou aplikaci EBIE s datovým skladem ČVUT – obsahuje mimo jiné datové connectory na tento sklad. Ebie-data-api data zpracuje a následně je poskytuje dále přes REST API. Ebie-data-api se dá spustit jako samostatná aplikace.
- **ebie-source-code** – První a původní repozitář, který obsahuje celou aplikační logiku (přihlašování, zobrazování obsahu z ebie-content atd.) portálu EBIE. Stejně jako ebie-data-api je i ebie-source-code samostatně spustitelná aplikace, která k plné funkčnosti potřebuje právě ebie-data-api.

Původně nebyly repozitáře v této podobě. Pomocí git submodule došlo k rozdělení repozitářů na další „podrepozitáře“ – git submodule tak umožňují uchovávat repozitář jako podadresář jiného repozitáře. Nejprve tak bylo učiněno především k oddělení obsahu od aplikační logiky, aby měli správci portálu ulehčenou editaci. Aplikováním submodule byla také oddělena historie úprav obou repozitářů a vývoj se tak pro tyto části stal nezávislý [18].

V průběhu nasazování bylo potřeba vyřešit požadavek na rozdělení vývoje na testovací a produkční (hlavní) verzi. K takovému úkolu je vhodné použít tzv. branches. Branches (větve) se běžně používají pro vývoj software a dají se představit jako „pracovní verze“. Commit (nahrání změny) se do různých větví provádí odděleně. Ke sloučení různých větví se používají tzv. merge requesty. Po provedení tohoto sloučení se sloučí určitý repozitář v obou větvích – dojde vlastně k přehraní změn z jednoho repozitáře jedné větve do repozitáře v druhé větvi [30].

V našem případě byly vytvořeny dvě větve:

- **master_vyvoj** – Je nasazena na testovacím serveru „EBIE-vývoj“. Její hlavní využití je ke zkoušení nových vývojových prvků. Pokud se prvky osvědčí, dojde k jejich nahrání do master větve pomocí merge request.

- **master** – Je nasazena na produkčním serveru „EBIE“. Tato větev v sobě udržuje verzi, kterou chceme poskytovat uživatelům.

3.1.5 Struktura stránky popisující proces

Jelikož se ve zbylých částech práce budeme věnovat převážně procesům zaměřených na správu obsahu portálu EBIE, bude vhodné si ukázat, jak vypadá ukázková stránka portálu zobrazující právě procesy. Rozvržení stránky pro procesy bylo navrženo Ing. Václavem Jirovským a je viditelné v diplomové práci představené v části 1.3.3.1. Jedinou změnou v původním návrhu je přidání sekce: „Podpůrné informace pro transakci“, viz obrázek 3.5. Kompletní rozvržení stránky procesu:

- Název procesu – Vystihující název procesu.
- Popis popisu – Slovní popis procesu – co znázorňuje.
- Popis procesu (DEMO metodika) – Obrázek procesu namodelovaném pomocí DEMO metodiky.
- Transakce procesu znázorněné tabulkami – Tabulka obsahující informace o iniciátorovi, exekutorovi a produktu.
 - Podpůrné informace pro transakci – Rozšiřující informace k transakci, například odkazy na legislativu, reporty atd.
- Vazby aktorů v procesu na ontologický katalog – Tabulka popisující vztah jednotlivých aktorů k entitám z ontologického katalogu.
- Popis procesu (notace BPMN) – Obrázek procesu namodelovaném pomocí BPMN a odpovídající zdroj procesu.

3.2 Analýza procesů pro správu obsahu

V nadcházející části se zaměříme na zmiňované procesy, které se týkají správy obsahu portálu EBIE – plnění/rozšiřování a částečné údržby, provedeme jejich analýzu a v následující kapitole 4 procesy namodelujeme pomocí metodiky DEMO a BPMN. Tyto procesy byly po konzultaci s vedoucím práce Ing. Michalem Valentou, Ph.D. vybrány jako nejdůležitější pro naše účely.

3.2.1 Publikování obsahu

Publikování obsahu je nejdůležitější proces zabývající se plněním obsahu portálu, který spadá částečně do rozšiřování portálu, jelikož při naplnění portálu novým obsahem dojde de facto k jeho rozšíření z obsahového pohledu. Některé úryvky textu o tomto procesu jsou čerpány ze zdroje [31].

3. AKTUÁLNÍ STAV EBIE A ANALÝZA PROCESŮ

Transakce T01 - uznání předmětu ostatních fakult ČVUT	
Iniciátor	A01 - žadatel o uznání předmětu
Exekutor	A02 - uznavatel předmětu
Produkt	předmět je uznán

Podpůrné informace pro transakci

- [Seznam ekvivalentních předmětů FIT a FEL ČVUT \(report\)](#)
- [Studijní a zkušební řád pro studenty ČVUT \(8. 7. 2015\)](#)
- [\(FIT\) případně Směrnice děkana FIT ČVUT č. 13/2015 pro realizaci BSP a MSP](#)

Obrázek 3.5: Transakce T01 – uznání předmětu ostatních fakult ČVUT.

3.2.1.1 Analyzování procesu

Proces publikování obsahu můžeme rozdělit do několika základních částí/kroků:

1. **Příprava lokálního prostředí pro úpravy – stažení aktuálního obsahu** – Provedeme stažení aktuálního obsahu z repozitáře ebie-content do nově vytvořeného lokálního adresáře pomocí příkazu:

```
git clone git@gitlab.fit.cvut.cz:hajcidav/ebie-content.git
```

Zmíněným příkazem dostaneme kompletní kopii aktuálního obsahu portálu EBIE (z testovací větve master_vyvoj) do našeho adresáře.

2. **Provedení požadovaných úprav** – V tomto případě záleží na zvolené úpravě, jelikož při rozdílných typech úprav musíme provést některé kroky navíc, případně zvolit kroky zcela odlišné.
 - a) **Úprava existujícího souboru/stránky** – Pro úpravu existující stránky na portálu, respektive .adoc souboru v repozitáři, postačí nalézt tento soubor ve správném adresáři a následně v něm provést změny v souladu s jazykem AsciiDoc.
 - b) **Vytvoření nového souboru/stránky** – V určitých podadresářích ebie-content se nachází soubory info.json, podle kterých EBIE načítá a zobrazuje svůj obsah. Při vytváření nového souboru .adoc je nutné zapsat potřebné informace do odpovídajícího info.json souboru, který je ve stejném adresáři, jako přidávaný .adoc soubor. Příklad vytvořeného souboru fakulta.adoc je viditelný na snímku 3.6. Pokud chceme mít položky ve výčtu na portálu seřazeny, musíme je ručně abecedně zařadit v souboru info.json.

- c) **Vytvoření nové sekce** – Pro novou sekci musíme vytvořit nový podadresář na zvoleném místě ve struktuře, přidat požadované .adoc soubory s obsahem jednotlivých stránek a soubor info.json. Důležité je vytvořit nový atribut v souboru logic.adoc v adresáři ebie-content. O attributech a interních odkazech dále v textu 3.2.1.4.
- d) **Vytvoření nového modulu** – Každý modul se skládá z těchto složek/souborů, které musíme do modulu zařadit, abychom docílili jeho úspěšného zavedení do portálu EBIE:
- Složka html5, obsahující soubory pomocí kterých Ascidoctor provádí převod .adoc souborů do souborů .html.
 - Složka lib, ve které se nacházejí šablony, viz 3.2.1.5.
 - Složky sekcí, ve kterých jsou soubory uvedené v předešlém bodě.
3. **Kontrola provedených změn** – Vytvořené .adoc soubory jsou převáděny do souborů jazyka HTML. Pro tyto účely se používá skript generate.ruby, který je umístěn v kořenovém adresáři ebie-content. Po spuštění příkazu `ruby generate.ruby` se vygenerují .html soubory, ve kterých je možné si výsledek zkontrolovat. Tato část tedy není povinná, nicméně se doporučuje její provedení.
4. **Nahrání změn do ebie-content** – Po provedení úprav potřebujeme změny dostat zpátky na GitLab do repozitáře ebie-content a větve master_vyvoj. Nahrání učiníme například pomocí sekvence těchto příkazů:
- ```
git add .
git status
git commit -m "*zpráva o provedených změnách*"
git push
```
5. **Zavedení změn na server EBIE** – Po aplikování předchozí sekvence příkazů jsme docílili k nahrání změn do ebie-content, nicméně je musíme zavést na samotný server. V prostředí serveru „EBIE-vývoj“ v kořenovém adresáři ebie-web za uživatele ebie provedeme příkazy:
- ```
git submodule update --remote
bundle exec rails generate content
```
- Prvním příkazem stáhneme obsah a druhým vygenerujeme výsledné HTML stránky.

Po provedení popsaného postupu dojde k publikování změn na server „EBIE-vývoj“, pokud budeme brát v potaz současný stav rozdělení větví (branches), viz podsekcce 3.1.4, a serverů pro portál EBIE uvedených v podkapitole 2.3. Pokud bychom chtěli promítnout nový obsah i na produkční server EBIE, museli bychom provést dodatečné úkony:

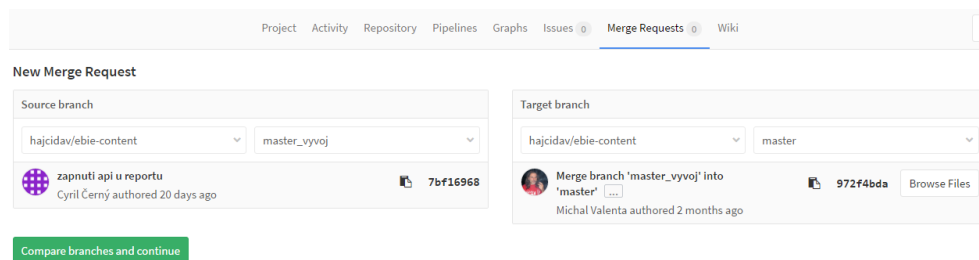
3. AKTUÁLNÍ STAV EBIE A ANALÝZA PROCESŮ

```
{
  "title" : "Fakulta",
  "description" : "Organizační jednotka univerzity.",
  "path" : ".",
  "filename" : "fakulta.html",
  "assets" : {
    "css" : [],
    "img" : [],
    "js" : []
  }
},
```

Obrázek 3.6: Nová položka pro entitu Fakulta v souboru info.json [31].

6. **Vykonání „Merge Request“** – Naším cílem je provést merge request¹¹ z větve `master_vyvoj` do větve `master` pro repozitář `ebie-content`. Příklad můžeme vidět na obrázku 3.7. Merge request se vykonává v prostředí GitLab a je k němu nutné mít patřičná oprávnění. Jako parametry musíme zvolit název repozitáře, který chceme integrovat, název výchozí a cílové větve.
7. **Zavedení změn na server EBIE** – Stejně jako v předchozím případě s rozdílem, že zde provádíme příkazy na serveru „EBIE“.

V budoucnu by mohly kroky „Zavedení změn na server EBIE“ z procesu zcela odpadnout, jelikož existují speciální nástroje a prostředky, které sledují změny z prostředí Git (u nás GitLab) a dokáží automaticky aktualizovat server/servery.



Obrázek 3.7: Ukázka „Merge Requestu“ z větve `master_vyvoj` do větve `master`.

¹¹Merge request se používá k integrování změn projektu v odlišných větvích [32].

3.2.1.2 Potřebné role

- Repozitář `ebie-content` je privátní a proto uživatel (autor nového obsahu), který chce provádět změny, musí mít práva k přístupu do tohoto repozitáře, aby si mohl stáhnout jeho obsah do lokálního adresáře.
- Pro přihlášení na server musí mít autor nového obsahu zajištěný přístup k tomuto serveru. Přes vzdálené SSH připojení je vhodné mít uložený veřejný klíč na serveru, aby správce nemusel zadávat heslo.
- U kroků „Zavedení změn na server EBIE“ musí být příkazy prováděny pod uživatelem `ebie`.
- Pro merge request z větve `master_vyvoj` do větve `master` musí mít uživatel na integrovaný projekt/repozitář udělena práva k této akci. Podobné podmínky platí také u schvalování žádosti o sloučení. Aby patřičná osoba mohla schvalovat merge request u repozitáře `ebie-content`, musí mít přidělenou roli `Master` k tomuto repozitáři.

Všechna vypsaná oprávnění/role by měla být přidělena správcům portálu EBIE, aby jim bylo umožněno měnit obsah portálu. Musíme však počítat se situací, kdy autorem nového obsahu je osoba, která nemá zajištěna tato oprávnění. Autor nového obsahu tak musí před stažením repozitáře požádat o přístup odpovědnou osobu, kterou je `Master` repozitáře `ebie-content`, stejně jako v případě přidělení role `Master`. U „Zavedení změn na server EBIE“ je osobou poskytující oprávnění správce daného serveru EBIE. Správce serveru může navíc provést zavedení změn za samotného autora.

3.2.1.3 Systémové závislosti

Pro provádění jednotlivých příkazů při publikování obsahu musíme mít nainstalovaný `Git`. Není to však jen `Git`, který potřebujeme k práci s `ebie-content`. Ostatní systémové závislosti pro generování nového obsahu jsou:

- Ruby 2.2+ ¹²
- AsciiDoctor 1.5+ (`install asciidoctor -v '<1.6'`)
- Tilt 2+ (`gem install tilt`)

3.2.1.4 Tvorba interních odkazů v souborech

V adresáři `ebie-content` se nachází soubor `logic.adoc`, viz struktura na snímku B.1, který obsahuje tzv. atributy ¹³. Tyto atributy jsou dostupné pro všechny

¹²Ke stažení například z <https://www.ruby-lang.org/en/documentation/installation/>

¹³Atributy využíváme například k aktivování různých stylů, chování, případně dalších funkcí [19].

.adoc soubory v EBIE. Pro nás je důležité, že s využitím atributů můžeme do souborů zanést interní odkazy na jiné soubory ve stylu:

link:nazev_modulu-nazev_sekcenazev_souboru.html[zobrazený text]

Dalším důvodem, proč jsou zde atributy uváděny, je nezbytnost zapsání nového atributu při vytvoření nové sekce nebo modulu. Ukázkou budiž následující atribut pro sekci ontologický katalog:

:studium-ontology: /modules/studium/ontology/view?path=&filename=

3.2.1.5 Šablony pro publikování obsahu

Při publikování obsahu je možné využít AsciiDoc šablony, umístěné ve složkách s názvem lib. Tyto šablony jsou vytvořené pro uživatele, kteří neznají jazyk AsciiDoc v takové míře, aby pomocí něho mohli sami tvořit obsah. V těchto šablonách se můžeme setkat s těmito sekvencemi znaků:

- // – Komentář, podle kterého jsou vyznačena v šabloně místa, kde bychom měli doplnit nějaký text. Samotný komentář pak vysvětluje, co bychom měli napsat.
- ++++ – Označení pro HTML kód, který bude přímo vložen do stránky. Tato místa nemusíme upravovat.
- +++ – Označení pro HTML kód, který obsahuje komentář určený k přepsání.

3.2.2 Nahlášení chyby ve zdrojových systémech

Druhým procesem je proces pro nahlášení chyby ve zdrojových systémech. Na rozdíl od prvního procesu zde nelze podle názvu zcela poznat, že se jedná o proces související s portálem EBIE. Podle následujícího textu zjistíme, že vazba mezi procesem a portálem existuje.

3.2.2.1 Datová dokumentační pravda

V kapitole 1 jsme se seznámili s pojmy datový sklad ČVUT a EBIE. Zjistili jsme, že portál EBIE je nadstavbou nad tímto skladem. O datovém skladu ČVUT víme přímo z podsekcce 1.2.2, že zdrojem ukládaných dat jsou zdrojové systémy (Source Systems). Zdrojové systémy mají své správce, kteří se starají o poskytování datových extraktů, které přicházejí do Stage databáze. Ze Stage se data dále dostávají do Target databáze, kterou můžeme považovat za hlavní databázi pro popis dat a ve které jsou target tabulky. V target tabulkách jsou uchovávána data, která již prošla různými transformacemi. Struktura target tabulek a jejich vzájemné vazby více odpovídají struktuře organizace, než v dílčích provozních databázích (zdrojové systémy), a proto jsou vhodné ke zdokumentování. Zde se znovu dostáváme k W. H. Inmonovi, jehož přístup

vzhledem k navržení datového skladu si zakládá na jasné identifikaci byznys konceptů na úrovni datového skladu, konkrétně na úrovni Integrated Data Layer obsahující Target databázi.

Dokumentace target tabulek je vytvářena Týmem datových skladníků. Nedávno proběhl první import target tabulek, představených v textu 3.1.2, do portálu EBIE, který byl realizován na základě integračního plánu popsáném v podkapitole 2.2. Hlavním cílem prezentování target tabulek na portálu EBIE je nabídnutí dokumentace dalším uživatelům.

V popisu architektury datového skladu ČVUT jsme se také dočetli, že Stage databáze je ve skutečnosti obraz ze zdrojových systémů, čili otisk 1:1. Popis datových struktur ve Stage databázi nám dává přehled o struktuře dat v těchto systémech. Při ETL transformaci ze Stage do Target databáze může také dojít k chybám, a proto je vhodné data kontrolovat, viz proces z podsekcce 3.2.3, a mít správně zdokumentovaná.

Stage a Target databáze vytvářejí datovou dokumentační pravdu, která je základem pro proces nahlášení chyby ve zdrojových systémech.

3.2.2.2 Analýza procesu

Zdrojem dat datového skladu jsou zdrojové systémy. Nabízející se otázkou je, co se stane, když se ve zdrojových systémech vyskytne „chyba“, související s datovými strukturami nebo se samotnými daty? Zmíněná „chyba“ nemusí být skutečnou chybou. Můžeme si představit situaci, kdy správci zdrojového systému KOS naplánovali změnu v systému a provedli ji, nicméně pro ni neposkytli žádnou dokumentaci. Z pohledu správců zdrojových systémů se o chybu nejedná, avšak z pohledu datového skladu o chybu jde, jelikož se do datového skladu promítne.

Závažnost chyby ve zdrojových systémech se může diametrálně lišit. Pro nás, jakožto „stranu datového skladu“, mohou nastat dvě situace:

1. **Chybu poznáme** – Detekujeme chybu v datovém skladu. Následně postupujeme ve dvou krocích:
 - a) Nejprve provedeme kontrolu, zdali nedošlo k chybě na naší straně. Předpokladem této kontroly je přítomnost dokumentace všech datových struktur, pomocí které bychom mohli poznat, o jak závažnou chybu se jedná a nebo jestli nejsme viníci chyby:
 - Pokud ano, nemusíme posílat upozornění na „stranu zdrojového systému“ a proces tak končí.
 - Pokud ne, musíme pokračovat podle části b)
 - b) Pokud jsme chybu nezavinili my, musí řešitel chyby kontaktovat osobu spravující zdrojový systém o vyskytující se chybě v datovém skladu. Od „strany zdrojových systémů“ budeme žádat vysvětlení této chyby nebo její zdokumentování. Na základě odpovědi budeme

případně pokračovat patřičnými úpravami, které povedou k vyřešení dané chyby.

2. **Chybu nepoznáme** – Mohou nastat situace, kdy chybu ve zdrojových systémech v datovém skladu nepoznáme. Příkladem by mohlo být například zmenšení velikosti datového typu VARCHAR nebo jiného. V tomto případě nemůžeme spoléhat na okamžité odhalení chyby, můžeme jen doufat, že se chyba v budoucnu někde objeví, popřípadě na ni upozorní „strana zdrojového systému“ či jiná třetí strana – chyba se například dostane až do portálu EBIE, kde ji někdo odhalí.

3.2.2.3 Potřebné role

- O kontrolu chyb, jejich případné řešení a reportování druhé straně se musí starat oprávněná osoba z oblasti datového skladu ČVUT. Není však podmínkou, aby to byla jedna osoba.
- Zdokumentovaná/vysvětlená chyba by měla přijít od kontaktní a odpovědné osoby spravující zdrojový systém, kterou by měl být pracovník Helpdesk přijímající zprávy/reporty o chybách, či správce zdrojového systému.

3.2.3 Kontrola kvality dat

Třetím a posledním procesem je proces pro kontrolu kvality dat. Než přejdeme k analýze procesu, zaměříme se na kvalitu dat obecně. V následující podsekcí jsou popsány některé informace z dokumentu [33].

3.2.3.1 Kvalita dat

V samotném úvodu práce jsme se dozvěděli, že kvalitní data není lehké určit, ale z pohledu organizace jsou velice důležitá. Na kvalitu dat se můžeme dívat z více perspektiv:

- **Kvalita byznys metadat** – U kvality byznys metadat se jedná o správné zavedení pojmů na úrovni byznys entit a procesů. Špatné či nejasně zavedené pojmy často vedou k nedorozumění během komunikace mezi zadavatelem a řešiteli/dodavatelem datových sestav. Za zadavatele požadavků na sestavy považujeme především vedoucí pracovníky a za řešitele poté návrháře sestav. Špatně nebo nejasně zavedené pojmy nevedou jen k zmíněným komunikačním problémům mezi stranami, ale také k možné dezinterpretaci dat, protichůdné interpretaci sestav a podobným nedostatkům.
- **Syntaktická kvalita dat** – Datový sklad by měl disponovat kvalitními daty ve smyslu jejich syntaxe, zajistit určení oboru hodnot pro všechny

datové položky, organizaci datových komponent do struktur, správné doménové hodnoty a profilování dat.

- **Sémantická kvalita dat** – Kvalita dat v rámci jejich sémantiky, čili významu těchto dat. Sémanticky kvalitní data nejsou nesmyslná a neobsahují tedy „zjevné nesmysly“, které vyvstanou v datech sestav, popřípadě v obsahu datových tržišť. Tyto „nesmysly“ jsou typicky způsobené nedůsledností vkládání dat do zdrojových systémů či neexistencí pravidel (metodik a postupů) a mohou způsobovat „chyby“ připomínané v předchozím procesu z části 3.2.2.

Pokud bychom měli uvést konkrétní příklad nekvalitních dat z prostředí ČVUT, můžeme se zaměřit například na systém KOS. Nedostatků v rámci datové kvality je v KOSu mnoho – data se v něm nehlídají a může docházet k časté nekonzistenci dat. My se zaměříme na atributy předmětů:

- **Atribut zakončení předmětu** – Kromě očekávaných, ale zároveň nejasně zavedených kontextů (ZK, Z+ZK, Z, KZ a variací) se v atributu zakončení předmětu objevují hodnoty jako „NIC“, „?“ a „101“ a další.
- **Rozsah předmětu** – Podobné jako v předchozím případě se jedná o textové pole s nedefinovaným obsahem – v kontextu FIT nacházíme vymezení v týdenním režimu „2p+2s“ (dvě hodiny přednášek a dvě hodiny seminářů týdně). Další zmatky způsobilo zavedení proseminářů, například „2p+1k+2s“. Pro předměty kombinovaného studia jsou uváděny hodnoty rozsahu v kontextu délky celého semestru, například „14+6“. Ostatní fakulty navíc mohou mít jiné formy výuky (např. ateliéry), které způsobují ještě větší nejasnosti v interpretaci položky.

Nejen představené problémy chybějící sémantiky jsou důvodem k stále rostoucí potřebě zavedení kontroly kvality dat.

3.2.3.2 Analyzování procesu

Pokud bychom chtěli kontrolovat například sémantické nedostatky ze systému KOS, nemůžeme po jeho správcích požadovat, aby provedli implementaci procesu pro celé ČVUT. Na úrovni organizační jednotky (zde vybrané fakulty) si však můžeme zavést proces pro kontrolu kvality dat, který bude podobné problémy řešit. Tento proces je vhodné umístit na úroveň mezi Stage a Target databázi. V datovém skladu následně můžeme implementovat proces jako ETL job, popřípadě provést jeho implementaci do již existujícího ETL jobu.

Jednotlivé kroky procesu kontroly kvality dat jsou:

1. **Zvolení kontrolovaného obsahu/položek** – Vybereme si, co je pro nás důležité, nad čím chceme provádět kontrolu dat a podrobně to popíšeme.

3. AKTUÁLNÍ STAV EBIE A ANALÝZA PROCESŮ

2. **Kontrola kvality dat se začne provádět** – Vybrání vhodného ETL jobu/procesu, do kterého kontrolu kvality dat začleníme, popřípadě vytvoření zcela nového. Tento proces by měl fungovat na úrovni datového skladu:

- Mezi Stage a Target databází v ETL jobu.
- Kontrola nad Target databází.

Součástí zavedení kontroly kvality dat je její technická implementace a dokumentace.

3. **Vytváření reportů/logů** – Výstupem kontroly kvality dat jsou výsledné reporty, které určí, jestli došlo k narušení datové integrity (chybě), případně její typ a další přidané informace ke kontrolovaným položkám.

4. **Zpracování reportů, odeslání reportů a jejich případná publikace** – Výsledky musíme oznámit zainteresovaným (třetím) stranám, kterých se daný report týká. Samotné reporty/logy můžeme případně publikovat na portál EBIE.

3.2.3.3 Potřebné role

- Pro zavedení kontroly kvality dat by měla mít organizační jednotka odpovědnou osobu, která bude zavedení a celý proces řídit.
- Výběr kontrolovaných položek, implementace kontroly kvality dat, zpracování reportů a publikování reportů na portál EBIE by měly vykonávat osoby z prostředí datového skladu a EBIE, které dané problematice rozumí.

Modelování procesů pro správu obsahu

V předchozí kapitole jsme analyzovali procesy, které se dále pokusíme namodelovat. Pro modelování procesů byly vybrány notace BPMN a metodika DEMO. Abychom procesním modelům správně porozuměli, musíme se s modelovacími technikami BPMN a DEMO alespoň ve stručnosti seznámit. Při rešerši těchto modelovacích technik si mimo jiné uvedeme jejich základní výhody a nevýhody, které se nám ve většině případů potvrdí i ve výsledných modelech. U všech procesů budeme vycházet z proběhlé analýzy z kapitoly 3.

4.1 Představení modelovacích notací

V této sekci se seznámíme se základními prvky notace BPMN a metodiky DEMO, které jsou použity při modelování analyzovaných procesů. O něco více prostoru věnujeme metodice DEMO, jelikož pro mnohé čtenáře nemusí být tolik intuitivní jako BPMN.

4.1.1 BPMN

BPMN (Business Process Model and Notation) je notace, která slouží k modelování podnikových procesů. BPMN byla vytvořena iniciativou BPMI (Business Process Management Initiative), která je v současnosti udržována organizací OMG (Object Management Group), s níž se spojila v roce 2005 [34]. V roce 2006 přijalo konsorcium OMG standard BPMN 1.0. Následovaly další verze BPMN, které neobsahovaly velké změny. Největší posun nastal v roce 2011, kdy vyšla verze BPMN 2.0, ve které došlo k několika úpravám a která je navíc také verzí poslední. BPMN od tohoto momentu neoznačovala zkrácený název pro Business Process Modeling Notation, ale pro Business Process Model and Notation [35]. Od doby svého vzniku se BPMN stala de facto standardem pro modelování podnikových procesů [36].

V BPMN aktuálně (BPMN 2.0) definujeme tři typy diagramů:

- Diagram spolupráce (procesu)
- Diagram choreografie
- Diagram konverzace

Diagram spolupráce, někdy nazývaný jako diagram procesu, je nejpoužívanějším diagramem ze tří jmenovaných – v BPMN se vyskytuje již od standardu 1.0 [37]. V práci, kde je dále nazýván obecně BPMN diagramem, se setkáme pouze s ním, a proto si ho blíže představíme.

4.1.1.1 BPMN diagramy

BPMN diagram je tvořen sítí grafických objektů/elementů, jejich aktivitami a zobrazením toku informací mezi nimi. Každý element můžeme jednoznačně rozlišit. U těchto elementů je důležité dodržovat jejich tvar, volnost poté máme při jejich barevném označení. Proto se můžeme například u různých modelovacích nástrojů podporující tvorbu BPMN diagramů potkat s odlišně barevnými reprezentacemi. V určitých případech můžeme použít v diagramu vlastní grafické objekty, které by však neměly nikterak narušovat ostatní objekty, pouze je doplňovat či upřesňovat [36].

4.1.1.2 Základní elementy BPMN

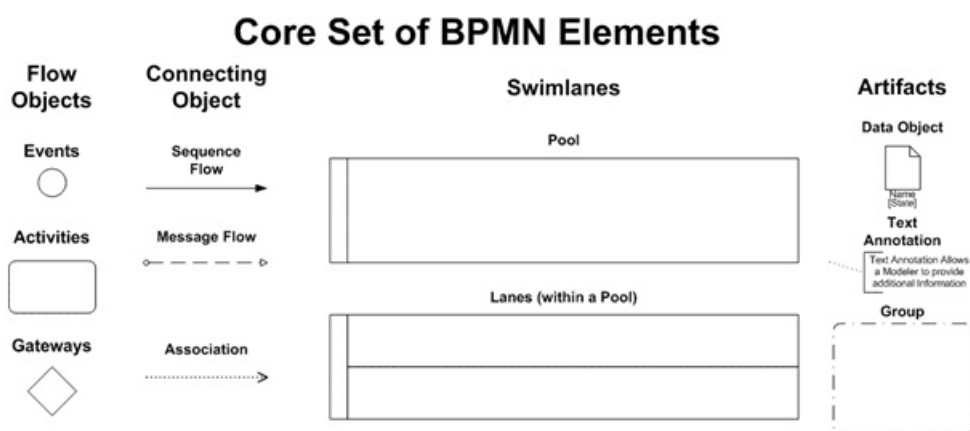
Informace o elementech BPMN jsou brány z publikací [37] a [38]. Grafické elementy jsou základním stavebním kamenem každého BPMN diagramu. V této části si popíšeme pouze základní elementy, které jsou důležité pro porozumění vytvořených diagramů. Jejich grafické znázornění můžeme vidět na obrázku 4.1. Mezi základní BPMN elementy tedy řadíme:

- **Plavecké dráhy (Swimlanes)** – Používají se k zobrazování účastníků nebo k uspořádání činností v procesu, například podle rolí.
 - **Bazén (Pool)** – Ohraničuje proces a v jeho záhlaví se nachází název bazénu. Bazén reprezentuje účastníka procesu. Komunikace mezi jednotlivými bazény probíhá pomocí zpráv.
 - **Dráha (Lane)** – Bazén se může rozdělovat na jednotlivé dráhy, které slouží k uspořádání a kategorizaci aktivit. Dráha může značit např. roli, organizaci či funkce organizace. Komunikace mezi jednotlivými dráhami probíhá pomocí sekvenčního toku.
- **Tokové objekty (Flow Objects)** – Hlavní objekty související s tokem informací v procesu.

- **Událost (Event)** – Události, jimiž proces začne, skončí, či které nastanou v jeho průběhu a které přímo ovlivňují tok procesu. Značí se kroužkem, ve kterém mohou být ikony upřesňující událost.
- **Aktivita (Activity)** – Znázorňuje určitou činnost či práci v procesu. Aktivita může být atomická (úloha, task) nebo v sobě může obsahovat nový proces (podproces, subprocess). Značí se obdélníkem s kulatými rohy.
- **Brána (Gateway)** – Rozděluje tok procesu na více toků, nebo slučuje více toků do jednoho. Brána se tedy používá například k rozhodování či paralelnímu zpracování. Značí se stojícím kosočtvercem s případnou ikonou uvnitř, která specifikuje typ brány.
- **Spojovací objekty (Connecting Objects)** – Propojují mezi sebou ostatní elementy do určitého celku.
 - **Sekvenční tok (Sequence Flow)** – Používají se ke spojení tokových objektů a určují, v jakém pořadí (sekvenci) dojde k jejich vykonání. Značí se nepřerušovanou čarou s plnou šipkou na konci.
 - **Tok zpráv (Message Flow)** – Znázorňuje tok zpráv mezi dvěma účastníky procesu. K vizualizaci se používá přerušovaná čára s prázdnou šipkou na konci.
 - **Asociace (Association)** – Umožňuje spojit vybraný objekt s dodatečnou informací. Označujeme ji přerušovanou čarou.
- **Artefakty (Artifacts)** – Využívají se k přidání upřesňujících informací do procesu a zároveň neovlivňují jeho tok.
 - **Datový objekt (Data Object)** – Reprezentuje data, se kterými pracují aktivity a jsou buď nezbytná pro danou aktivitu, popřípadě z aktivity vycházejí. Znak pro datový objekt je ve většině případů obdélník s přehnutým rohem (list papíru).
 - **Seskupení (Group)** – Seskupení vybraných objektů pro analytické či dokumentační důvody. Skupina je ohraničena obdélníkem značeným přerušovanou čarou.
 - **Poznámka (Annotation)** – Poskytuje dodatečnou informaci k jinému objektu ve formě textu. Text je k objektu připojen asociací.

4.1.1.3 Specifické elementy BPMN použité v diagramech

Následující seznam specifických elementů použitých při modelování procesů pomocí BPMN nám pomůže k úplnému pochopení jednotlivých diagramů. Grafické znázornění si můžeme prohlédnout na snímku 4.2. Čísla seznamu odpovídají číslům elementům z obrázku.



Obrázek 4.1: Základní elementy BPMN [39].

1. **Začátek procesu.**
2. **Signál pro začátek procesu** – Vyvolání nějaké události z jiného procesu způsobí start tohoto procesu.
3. **Konec procesu.**
4. **Ukončení procesu s odesláním zprávy** – Proces končí, ale účastník ještě odešle zprávu jinému účastníkovi.
5. **Událost pro přijetí zprávy** – Zpráva je přijata od komunikujícího účastníka.
6. **Exkluzivní (XOR) brána** – Při rozdělení brány směřuje tok dále podle jedné z odchozí části v závislosti na výchozí podmínce.
7. **Aktivita s odesláním zprávy** – Vykonání dané aktivity a odeslání zprávy.
8. **Aktivita s přijetím zprávy** – Přijetí zprávy a vykonání dané aktivity.



Obrázek 4.2: Specifické elementy BPMN použité při modelování procesů.

4.1.2 DEMO

DEMO je akronymem pro Design & Engineering Methodology for Organizations a jedná se o metodiku pro modelování procesů, která je podložena silným teoretickým základem. DEMO zkoumá procesy z pohledu celé organizace a poskytuje tak celkový pohled nad organizací i v rámci jejího vnitřního chování [40].

DEMO je tedy metodika sloužící k návrhu, (re)inženýringu a implementaci pro organizace, případně sítě organizací. Začátky metodiky DEMO, jejímž zakladatelem je prof. Jan Dietz, sahají do 80. let 20. století do Nizozemí, konkrétně do Technické Univerzity v Delfu [41]. Většina textu o metodice DEMO je dále čerpána ze zdrojů [42], [43] a [44].

Jedním z hlavních důvodů vzniku DEMO byl podnět pro vytvoření metodiky, ze které bude možné zrekonstruovat ontologický model podniku pomocí Enterprise Ontology.

4.1.2.1 Enterprise Ontology

Enterprise Ontology (podniková ontologie) je jedním z pilířů Enterprise Engineering¹⁴ sloužící k porozumění esence organizací. Enterprise Ontology je kompletně oproštěná od způsobu, jak je tato esence realizována a implementována. Hlavním cílem Enterprise Ontology je vytvoření ontologického modelu organizace – zabývá se tím, jak organizace funguje uvnitř (jak probíhá proměna vstupů na výstupy). Podniková ontologie je vhodná k pochopení konstrukce a operací podniku a je založena na PSI teorii.

4.1.2.2 PSI teorie

DEMO oproti jiným modelovacím metodám nenahlíží na procesy jako na činnosti, ale jako na síť komunikace, která je zachycena v PSI teorii. Cílem PSI teorie je vyextrahovat jádro konkrétní organizace tak, aby nebylo závislé na konkrétní implementaci. PSI teorie se zabývá fungováním organizace a hledí na ni jako na skupinu objektů, které vstupují do vzájemné interakce. Tuto interakci nazýváme tzv. transakcí, která je určena k vytvoření nějakého produktu. Každá transakce probíhá mezi dvěma stranami – strana vyvolávající transakci (iniciátor) a strana vykonávající transakci. PSI teorie se skládá celkově z jednoho teorému a čtyř axiomů:

- **Operační axiom** – Podle operačního axiomu je chod organizace složen z aktivit, které jsou prováděny tzv. aktory. Aktoři provádí dva typy aktů, přičemž akt je atomická jednotka akce v organizacích:

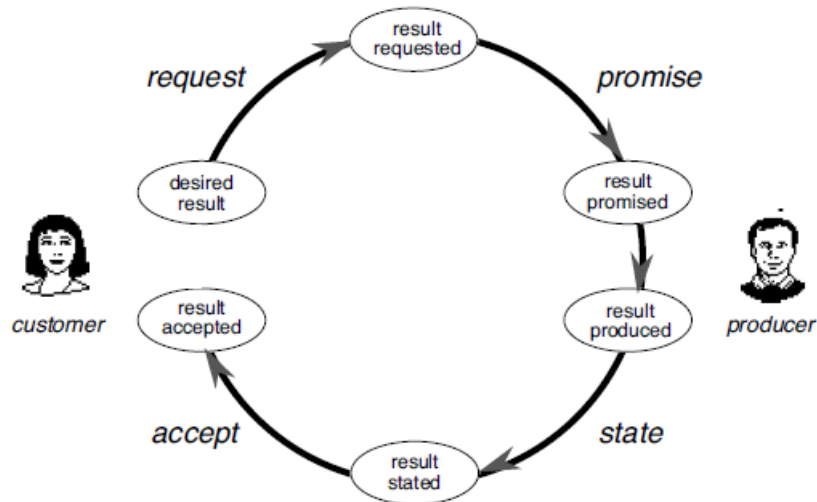
¹⁴Dalšími pilíři jsou: Enterprise Architecture, Enterprise Design, Enterprise Management a Enterprise Governance.

1. **Koordinální akt (C-act)** – Akce, pomocí níž vzniká v tzv. koordinačním světě (C-world) koordinační fakt (C-fact). Koordinální akt je prováděn aktorem nazývaným performer a adresován aktorovi nazývaným addressee. Aktoři vykonáváním těchto aktů vstupují do vztahů/závazků, které mají za cíl vytvoření produkčního aktu. Můžeme tvrdit, že se prakticky jedná o komunikaci o výsledném produkčním faktu.
 2. **Produkční akt (P-act)** – Akce, pomocí níž vzniká v tzv. produkčním světě (P-world) produkční fakt (P-fact). Při provádění produkčních aktů vytváříme nové produkty a nebo služby, které mohou být jak hmotné, tak nehmotné. Příkladem hmotného P-actu může být výroba nové součástky, jeho opravení či prodej. Příkladem nehmotného P-actu může být zvolení hráče kapitánem týmu a další.
- **Transakční axiom** – Řeší záležitosti ohledně propojenosti koordinačních a produkčních aktů a míry jejich vzájemné závislosti. V transakčním axiomu se objevují univerzální vzory (šablony), podle kterých jednotlivé C-acty probíhají v přesně určeném pořadí. Tento vzor představuje transakci, která probíhá mezi dvěma aktory jejichž cílem je dosáhnout P-factu. První z aktorů je vždy tzv. iniciátor, který žádá o vytvoření produktu druhého aktora, tzv. exekutora, čili vykonavatele transakce. Každá transakce se skládá ze tří fází:
 1. Order fáze – Pomocí vzájemné komunikace se snaží aktoři dojít k dohodě na výsledku transakce (P-factu). Iniciátor žádá exekutora.
 2. Execution fáze – Exekutor vytvoří daný výsledek (P-fact).
 3. Result fáze – Aktoři diskutují o „správnosti“ výsledku, čili jestli výsledek odpovídá představám. Pokud ano, je P-fact přijat. Po jeho přijetí začne P-fact existovat, do té doby neexistuje.

Transakční vzory dělíme do tří kategorií:

1. **Základní transakční vzor** – Je to základní transakční vzor, který reprezentuje pouze tzv. Happy path (Happy flow). Happy path označuje transakci, ve které jsou všechny akty přijaty a nedochází k žádným odmítnutím. Příklad Happy flow můžeme vidět na snímku 4.3. Happy flow (základní transakční vzor) se skládá z čtyř základních C-(f)actů (request, promise, state, accept), které vedou ke vzniku P-(f)actu v následujícím pořadí:
 - a) Request – Iniciátor vytvoří požadavek na nový P-fact.
 - b) Promise – Exekutor slíbí iniciátorovi vytvoření P-factu.
 - c) Execution – Exekuce/vytvoření P-factu (je prováděn P-act).

- d) State – Požadovaný P-fact je vytvořen.
- e) Accept – Přijetí P-factu iniciátorem.



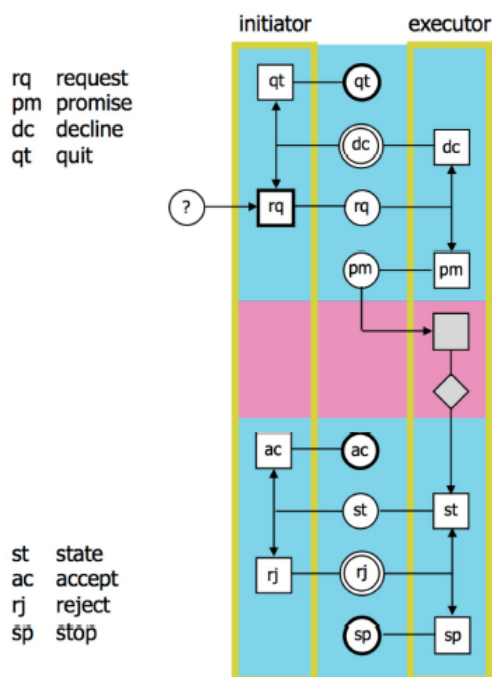
Obrázek 4.3: Happy flow transakce [43].

2. **Standardní transakční vzor** – Vychází ze základního transakčního vzoru, ke kterému přidává možnosti, kdy v transakci dojde k odmítnutí ve dvou případech:
 - a) Decline – Exekutor odmítne vytvoření P-factu. Pokud dojde k decline, iniciátor může vytvořit nový request, případně celou transakci ukončit (quit).
 - b) Reject – Iniciátor odmítne výsledek (P-fact) a nastane tzv. reject. Exekutor posoudí, zdali je odmítnutí oprávněné. Jestliže ano, je transakce ukončena (stop). Pokud není, transakce se vrací do stavu state.

Standardní transakční vzor je zobrazen na obrázku 4.4.

3. **Kompletní transakční vzor** – Standardní transakční vzor neuvazuje situace, kdy chce aktor vzít zpátky některé akty a vrátit se do předchozího bodu. To vedlo k vytvoření kompletního transakčního vzoru, který lépe modeluje situace z reálného života a obsahuje tzv. odvolávací vzory (revoky). Tyto vzory vrací transakci do požadovaného stavu (request, promise, state, accept), pokud návrat druhá strana povolí. Pro případný zájem čtenářů je kompletní transakční vzor zobrazen v závěrečných Přílohách, konkrétně na obrázku B.3.

- **Kompoziční axiom** – Podle transakčního axiomu dojde k vytvoření určitého P-factu. Kompoziční axiom oproti tomu definuje, jak jsou jed-



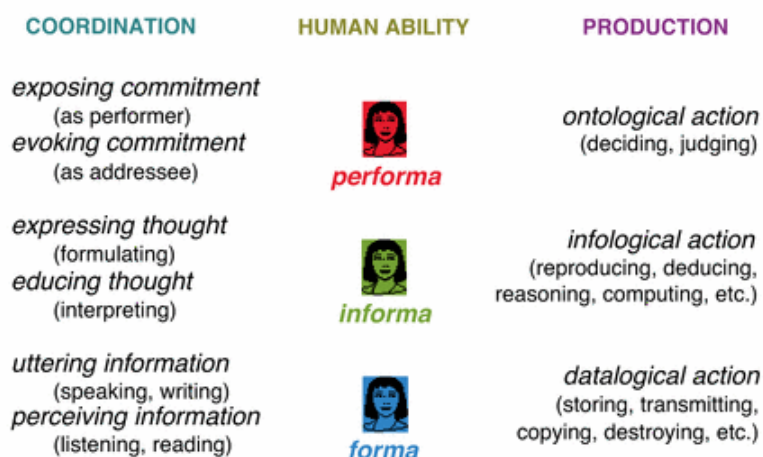
Obrázek 4.4: Standardní transakční vzor [44].

notlivé P-facty propojené. Aby například mohl vzniknout produkční fakt jedné transakce, musí nejprve vzniknout produkční fakt druhé transakce a podobně.

V rámci PSI teorie existují tzv. self-activated transakce, které jsou iniciovány jednotlivými aktory v momentě, kdy je potřeba vyrobit určitý P-fact. K jeho vytvoření může být požadováno vytvoření jiných P-factů. Dohromady je vytvořen tzv. strom transakcí. S takovými transakcemi se setkáme u našich procesů.

- **Distinkční axiom** – Na základě distinkčního axiomu mohou aktoři při svých úkonech provádět tři schopnosti vztahující se k produkčním aktům nazývané performa, informa a forma:
 - **Performa** – Schopnost, pomocí které se vytvářejí nové originální věci. Využívá se při tom buď přímé, nebo nepřímé komunikace.
 - **Informa** – Zabývá se obsahem informace a jak o informaci probíhá komunikace mezi lidmi. Obsah informace je zcela abstrahován od její formy.
 - **Forma** – Schopnost týkající se formy, ve které jsou informace přijímány, předávány atd.

Na distinkční axiom se můžeme podívat na snímek 4.5. Obecně platí, že pro vykonání performy musíme provést více činů informy. To samé můžeme říci také pro informu a formu.



Obrázek 4.5: Distinkční axiom [45].

- **Organizační teorém** – Předchozí čtyři axiomy PSI teorie přináší různé pohledy na fungování a chod organizace. K vytvoření konzistentního, stručného, koherentního a všeobecného modelu podniku musíme definovat tzv. organizační teorém. Organizační teorém říká, že podnik či organizace je heterogenní systém, který je tvořen vrstevnou integrací třech heterogenních systémů [45]:

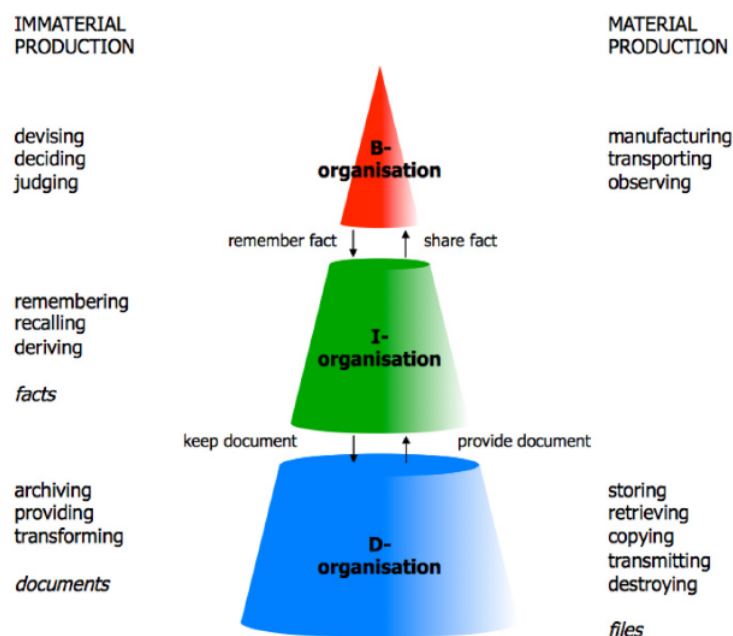
1. B-organizace (B = Business) s ontologickým výstupem.
2. I-organizace (I = Intellect) s infologickým výstupem.
3. D-organizace (D = Document) s datalogickým výstupem.

Na snímku 4.6 je zobrazena struktura zmiňovaných systémů. Je zde vidět, že jsou systémy propojené – D-organizace podporuje I-organizaci a I-organizace B-organizaci. Pro potřeby metodiky DEMO je nejdůležitější vrchol pyramidy, čili B-organizace.

4.1.2.3 Modelování pomocí DEMO, DEMO modely

V následujících částech práce si ukážeme modely, které používá metodika DEMO. Konkrétní příklady elementů dvou diagramů OCD a PSD (vysvětlení diagramů dále) jsou uvedeny v Přílohách v sekci B.2. Některé informace o modelech jsou získány z diplomové práce [46].

V DEMO jsou dva nejdůležitější elementy pro modelování ontologické transakce a aktoři:

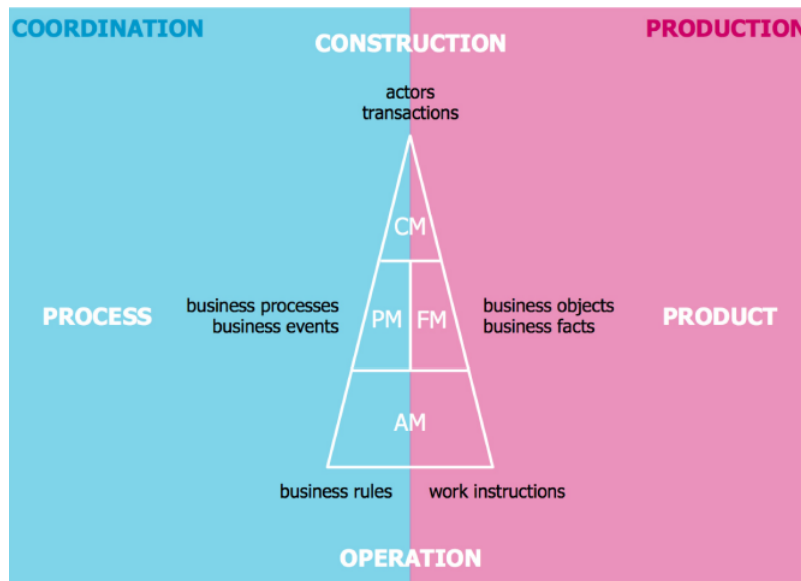


Obrázek 4.6: Tři aspekty organizace [44].

- **Ontologická transakce** – Jsou to transakce probíhající na ontologické úrovni (performa), viz Distinkční axiom, jejichž výsledkem je vytvoření něčeho zcela nového, hmotného či nehmotného.
- **Aktor** – Transakce má vždy dva aktory: iniciátora a exekutora (vykonavatele) transakce. V rámci implementace modelů může být v DEMO jedna osoba iniciátorem více transakcí, nicméně exekutorem pouze jedné transakce.

Pro modelování procesů v metodice DEMO se používají čtyři druhy modelů, které se mohou rozdělovat na další podmodely. Z obrázku 4.7 můžeme odvodit, že každý model se dívá na organizace z různých úrovní abstrakce – Construction Model (CM) má nejvyšší úroveň abstrakce, zatímco Action Model (AM) popisuje organizaci o mnoho detailněji. Dohromady tvoří všechny modely kompletní ontologickou znalost organizace [47].

- **Construction Model (CM)** – Znázorňuje konstrukci organizace. CM zobrazuje nalezené druhy jednotlivých transakcí, jejich produkty a odpovídající aktory, kteří tyto transakce provádí. CM dále obsahuje:
 - **Organization Construction Diagram (OCD)** – Ukazuje jednotlivé transakce, aktory, jejich vztahy na transakce a případné



Obrázek 4.7: Podmodely ontologického modelu [44].

další vztahy na transakční banky (viz dále). Tento diagram znázorňuje tzv. Scope of Interest ¹⁵, neboli zkoumanou oblast zájmu, pomocí níž dokážeme vyčíst interní a externí role diagramu. Elementy, které se používají pro modelování OCD, jsou zobrazeny v Přílohách v podsekcí B.2.2.

- **Transaction Product Table (TPT)** – Tabulka znázorňující slovní popis transakcí a jejich výsledných produktů. Občas se tabulka obohacuje o informace iniciátora a exekutora dané transakce, nicméně ve spojení s OCD se jedná o redundantní informace.
- **Bank Contents Table (BCT)** – Reprezentuje skutečnou banku faktů ve formě tabulky, která obsahuje informace o závislých a nezávislých faktech všech analyzovaných transakcí. V tabulce se mohou objevit také informace o faktech vnějších transakcí.
- **Process Model (PM)** – Ukazuje posloupnost a návaznosti transakcí, čili jejich průběh. Každá transakce probíhá podle transakčního vzoru. Podle PM se dozvíme, jak je jedna transakce závislá na produktu jiné transakce. Process Model obsahuje dva diagramy:
 - **Process Structure Diagram (PSD)** – Popisuje strukturu jednotlivých procesů a jednotlivé kroky pro všechny transakce v procesu. Z diagramu je zřejmé, ve kterém C-factu se používá podřízená

¹⁵Scope of Interest (SoI) – Zkoumaná oblast zájmu. Ve většině případů se jedná o zkoumanou organizaci, nebo její část.

transakce a zároveň v jakém C-actu se čeká na dokončení podřízené transakce. PSD mimo jiné definuje, kolik a jaké podřízené transakce daná transakce spouští [48]. Elementy, které se používají pro modelování PSD, jsou zobrazeny v Přílohách v podsekcí B.2.3.

- **Transaction Pattern Diagram (TPD)** – Jedná se o konkrétní implementaci kompletního transakčního vzoru, viz obrázky B.3. Do transakčního vzoru jsou zaneseny konkrétní cesty, které reprezentují konkrétní situace v procesu. Nevýhodou diagramu může být použitelnost pouze pro jednu transakci – jeden TPD odpovídá jedné transakci.
- **Fact Model (FM)** – Můžeme si ho představit jako „stavový prostor P-worldu“, kde jednotlivé stavy tvoří produkty transakcí. V diagramu jsou zmíněna závislá i nezávislá fakta, která jsou navzájem propojena business pravidly. Fact Model je reprezentován jedním diagramem:
 - **Object Fact Diagram (OFD)** – Jediný diagram FM, který obsahuje business objekty, fakta a jejich vzájemný vztah, případně jejich atributy. OFD deklaruje typy výsledků.
- **Action Model (AM)** – Nejobsáhlejší model ze všech zmiňovaných, ze kterého bychom mohli všechny ostatní modely odvodit. Action Model neobsahuje žádný diagram ani tabulku, je tvořen pomocí tzv. akčních pravidel (action rules), které jsou zapsány v pseudo-algoritmickém jazyce. Pomocí nich jsou textově popsány akce, které nastanou ve stavech requested, promised, stated a accepted.

4.1.3 Rozdíly BPMN a DEMO

V této podsekcí si představíme základní rozdíly mezi modelovacími technikami BPMN a DEMO, jejichž základní koncept byl rozebrán v předchozích částech textu. Kromě samotných rozdílů také uvedeme silné a slabé stránky BPMN a DEMO. Kompletní tematika porovnání BPMN a DEMO byla rozebírána již v několika pracích, na což upozorňuje ve své diplomové práci s názvem „Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů“ Ing. Ondřej Mráz [42]. Pan Mráz ve své práci tyto práce rozebírá a na základě provedených analýz uvádí výsledky, ze kterých budeme v podsekcí čerpat informace. Dalším zdrojem některých informací týkajících se zejména výhod a nevýhod BPMN a DEMO je další diplomová práce „Porovnání způsobu modelování podnikových procesů pomocí BPMN a DEMO z pohledu běžných uživatelů“ [46], jejímž autorem je Ing. Michal Radvan.

4.1.3.1 Výhody a nevýhody BPMN

BPMN je notací pro modelování procesů, která se ve světě IT stala velmi populární. Její hlavní výhody (silné stránky) jsou:

- Vytvoření modelů pro obvyčejné procesy je poměrně jednoduché.
- Porozumění modelům není nikterak složité ani pro méně zasvěcené osoby do této notace – BPMN je intuitivní.
- Existuje velká řada podpůrných editorů, ve kterých můžeme procesy namodelovat a případně odsimulovat.
- Pro popis procesu se často používá jediný diagram (diagram spolupráce), podle kterého pochopíme fungování procesu.

BPMN má také své nevýhody (slabé stránky):

- Absence teoretického základu, která vyústila v neexistenci přesného návodu na vytváření BPMN modelů. Neexistující návod může způsobit nekompletnost modelů, případně chybějící či naopak přebytné informace. Pokud bychom navíc nechali namodelovat stejný proces více uživatelům, je velice pravděpodobné, že bude vytvořeno více rozdílných modelů – BPMN je benevolentní (jedná se o notaci, kde nemáme přesný návod).
- Jednotlivé editory mohou rozdílně interpretovat modely, což může způsobit, že model vytvořený v jednom editoru nemusí jít odsimulovat v jiném.
- Těžce zachycují konkrétní pravidla pro určité aktivity nebo události.
- Pokud bychom chtěli u větších procesů zaznamenat všechny reálné situace, které by mohly v procesu nastat, výsledný model bude velmi obsáhlý a pravděpodobně také nečitelný.

4.1.3.2 Výhody a nevýhody DEMO

DEMO je metodikou pro modelování procesů, o které naopak mnozí lidé z oblasti IT stále nevědí. Mezi její největší výhody řadíme:

- DEMO je postaveno na silném teoretickém základu, kterým je PSI teorie. Pokud aplikujeme všechny pravidla, která nám teorie dává a které slouží k tvorbě nových modelů, měli bychom dostat vždy stejný model, ať ho vytvořil kdokoliv – DEMO je striktní a neumožňuje přílišnou benevolenci (jedná se o metodiku, kde máme přesný návod).
- Na základě pravidel z předchozího bodu je vytvořený model kompletní a nechybí v něm žádné důležité prvky.
- Přesná reprezentace procesů na všech úrovních.
- Poměrně dobře použitelné na modelování větších celků a pro zachycení byznys pravidel v celé organizaci.

Nevýhody metodiky DEMO jsou následující:

- Vytvoření modelů může být náročné, jelikož je nutné mít pokročilé teoretické znalosti PSI teorie, což můžeme tvrdit také o porozumění jednotlivým modelům.
- Neexistence podpůrných nástrojů, pomocí kterých bychom byli schopni namodelovat všechny typy modelů. Chybí navíc nástroje, pomocí kterých bychom mohli DEMO modely odsimulovat. V současnosti se stále rozvíjí aplikace Demoworld (<http://www.demoworld.nl/>), ve které můžeme provádět simulaci, nicméně pomocí aplikace lze vyrobit pouze OCD a PSD diagramy ¹⁶.
- Pro porozumění procesu nám většinou nestačí pouze jeden diagram.

4.1.3.3 Porovnání BPMN a DEMO

Obecně lze říci, že silné a slabé stránky BPMN a DEMO se rozcházejí, čili kde je „silné“ BPMN, je naopak „slabé“ DEMO. V rámci diplomové práce Ing. Ondřeje Mráze [42] byly zjištěny odlišnosti a podobnosti metodiky DEMO a notace BPMN, z nichž si představíme všechny, které jsou důležité i pro naše modely:

- Construction Model (Konstrukční model) DEMO nemá v BPMN přesné vyjádření, slouží totiž k zaznamenání struktury jednotlivých transakcí, jejichž posloupnost je dále rozebrána v PSD diagramu spadajícím do Process Modelu (Procesní model). Procesní model je navíc nejvíce podobný BPMN diagramu spolupráce.
- Fact Model (Skutečný model/Model fakt) nemá v BPMN žádné zastoupení. Action Model (Akční model) může sloužit jako příručka, pomocí které uživatelé například poznají, podle jakého toku mají na základě podmínky v BPMN diagramu pokračovat.
- K porovnání BPMN a DEMO není potřeba vytvářet Fact Model a Action Model. K pochopení procesu stačí Construction Model a Process Model, konkrétně diagramy OCD a PSD. Pro pochopení některých dalších souvislostí jsou v našich procesech doplněny o TPT a AOM tabulky (viz dále).
- BPMN diagram standardně obsahuje všechny typy transakcí (ontologické, infologické a datalogické). DEMO se naopak zaobírá převážně ontologickými transakcemi.

¹⁶Seznam ostatních nástrojů pro DEMO modelování můžeme najít na adrese <http://www.ee-institute.org/en/demo/tools>

- Ve většině případů se v BPMN při vynechání „nepotřebných“ infologických a datalogických transakcí význam procesu výrazně nezmění. Infologické a datalogické transakce se používají převážně ke konkrétní implementaci procesu. K jeho dokumentaci stačí využít ontologické transakce.
- V DEMO jsou u ontologických transakcí aplikovány všechny stavy transakčního axiomu (request, promise, state, accept, decline, reject), zatímco v BPMN většinou nenajdeme všechny tyto stavy. Proto byly pro porovnání k jednotlivým BPMN diagramům vytvořeny BPMN diagramy s ontologickým rozbohem (viz dále). Využití ontologického konceptu navíc vede k lepší konzistenci modelu, analýze popisovaného problému a optimalizaci podnikových procesů [46].
- V BPMN se většinou nevyznačují tzv. unhappy path (unhappy flow), tedy cesty v procesu, u kterých dojde k nečekaným situacím. Pokud bychom chtěli vyjádřit v BPMN také tyto situace, mohl by být výsledný diagram velmi obsáhlý a nečitelný.
- V BPMN se ve většině případů neuvazuje o odvolání části procesu (revoke) a jeho návrat do předchozího stavu/místa. V DEMO naopak odvolávací vzory používáme.
- Produkční akty se v BPMN definují pomocí aktivit a koordinační akty jsou znázorněny buď pomocí aktivit, nebo pomocí událostí. Tok procesu je v BPMN vyjádřen sekvenčním tokem. V DEMO používáme k popisu toku PSD diagram, ze kterého však není zcela patrné, v jakém kroku se proces nachází.
- V DEMO mohou aktoři vykonávat pouze jednu transakci, na rozdíl od BPMN, kde může jeden člověk (reprezentovaný plaveckou drahou v bázenu) provádět více činností. Kvůli tomu je vhodné vytvořit mapovací tabulku (AOM) mezi aktory jednotlivých transakcí (DEMO) a specifickými účastníky (BPMN). Koncept aktora v DEMO je podobný konceptu plavecké dráhy v BPMN.

4.2 Modelování analyzovaných procesů

V následujících částech si ukážeme vytvořené diagramy a tabulky jednotlivých procesů. K procesům byly namodelovány diagramy a vytvořeny tabulky, které se zdály být dostačující a relevantní vzhledem ke struktuře procesů. Tento fakt byl odsouhlasen na základě schůzky s vedoucím a oponentem práce.

Z důvodu velikosti a jeho lepšího zachycení byl proces „Publikování obsahu“ rozdělen na dva menší procesy:

- **Publikování obsahu na testovací server** – Publikování nového obsahu na testovací server „EBIE-vývoj“.

- **Publikování obsahu na produkční server** – Publikování nového obsahu z testovacího serveru „EBIE-vývoj“ na produkční server „EBIE“.

Proces „Kontroly kvality dat“ byl podobně jako předchozí proces rozčleněn na dva:

- **Zavedení kontroly kvality dat** – Proces se zabývá zavedením kontroly kvality dat na úrovni organizační jednotky, čili výběrem kontrolovaných položek, tvorbou dokumentace a implementací.
- **Aplikování kontroly kvality dat** – Proces se zaměřuje na aplikaci kontroly kvality dat jako již zavedeného procesu, jehož výstupem jsou reporty, které je nutné dále zpracovat.

U procesů týkajících se kontroly kvality dat nebyl vyroben BPMN diagram společně s ontologickým rozbohem a to především z důvodů (zejména u Zavedení kontroly kvality dat) větší složitosti a pravděpodobné nečitelnosti diagramu, která byla rozebírána v části textu 4.1.3.1 a na kterou upozornil také Ing. Ondřej Mráz ve své diplomové práci [42]. U Aplikování kontroly kvality dat se navíc vyskytuje transakce typu self-activated, kterou by mohlo být složité v BPMN namodelovat. Nebylo proto nutné vyrábět ani matici mapování (AOM) mezi osobami z organizace (účastníky BPMN) a aktorskými rolemi, viz dále.

Pro modelování BPMN diagramů byl použit online editor „BPMN.IO“, dostupný na adrese <http://bpmn.io/>, který umožňuje mimo jiné export a import BPMN diagramů. Pro modelování DEMO diagramů byl vybrán nástroj Modelworld, dostupný na adrese <http://modelworld.nl/>, který také umožňuje export a import DEMO modelů vytvořených pomocí tohoto nástroje. Oba tyto nástroje splňovaly všechny autorovi požadavky na vytvoření modelů. Zdrojové soubory stejně jako obrázky všech diagramů jsou poskytnuty na přiloženém CD.

Elementy upřesňující jednotlivé DEMO diagramy (OCD a PSD) jsou k nahlédnutí v Přílohách (podsekce B.2.2 a B.2.3).

4.2.1 Publikování obsahu na testovací server

4.2.1.1 BPMN diagram a diagram ontologického rozboru

Na prvním diagramu (obrázek 4.8) je znázorněno publikování obsahu na testovací server pomocí notace BPMN. Na základě analýzy procesu jsme vybrali tři účastníky: Autor nového obsahu, Master repozitáře ebie-content a Správce testovacího serveru EBIE.

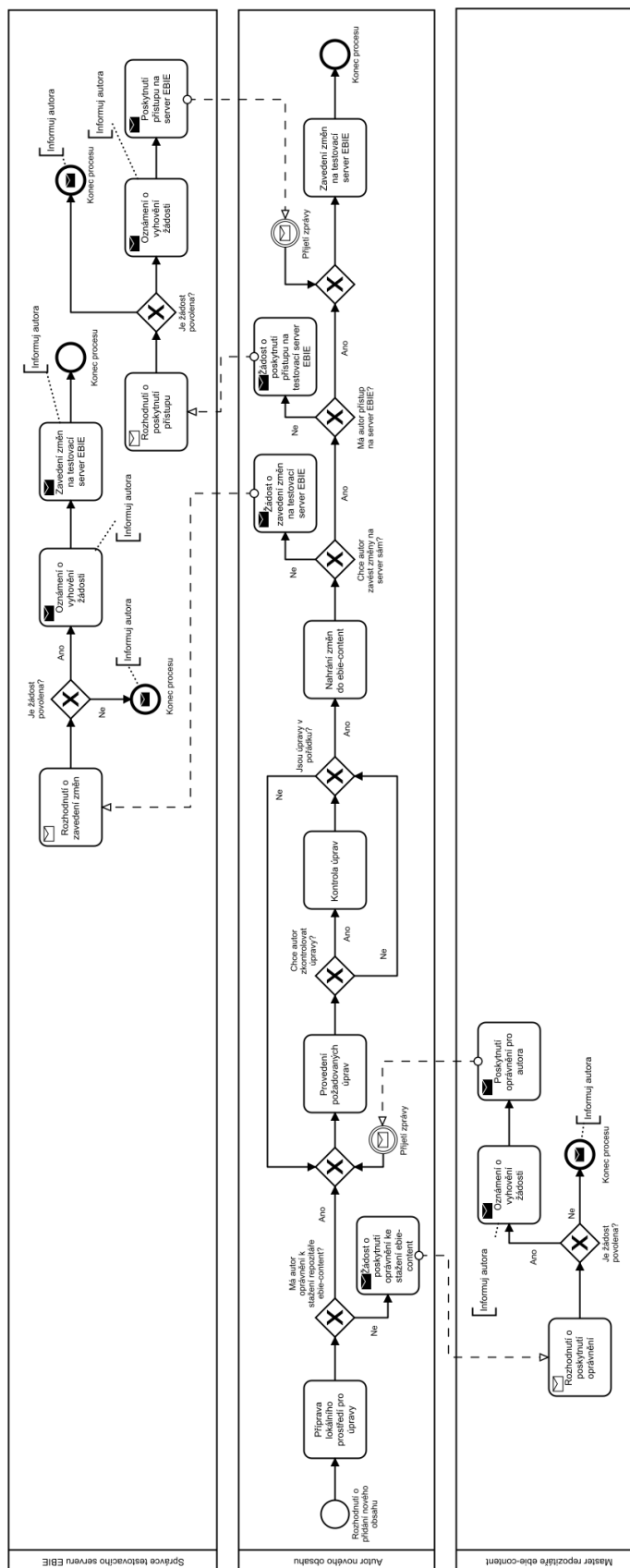
Můžeme zahlédnout, že v diagramu se nachází několik rozvětveních toku, která v těchto případech většinou odpovídají možnostem, kdy autor nového obsahu má/nemá patřičná oprávnění k vykonání činnosti a následně o ně může požádat, případně požádá jiného účastníka o provedení úkolu (Zavedení změn na testovací server EBIE). Pokud má autor nového obsahu všechna oprávnění a rozhodne se zavést obsah na testovací server sám, postupuje po pomyslné „rovne čáře (přímém toku)“ a nepotřebuje komunikovat s jinými účastníky pro dokončení procesu. Jak již bylo zmiňováno u samotné analýzy procesu, správci portálu EBIE budou většinou postupovat tímto způsobem, jelikož by jim měly být přiděleny všechny přístupy.

Druhý diagram (obrázek 4.9) ukazuje ontologický rozbor prvního BPMN diagramu. Ontologický rozbor je důležitý pro potřeby metodiky DEMO, jelikož pomocí něho můžeme najít v procesu jednotlivé transakce. Ontologický rozbor je ve skutečnosti aplikace performa-informa-forma analýzy, čili použití distinkčního axiomu, popsáno v části textu 4.1.2.2. Pomocí této analýzy jsou jednotlivé aktivity rozděleny na:

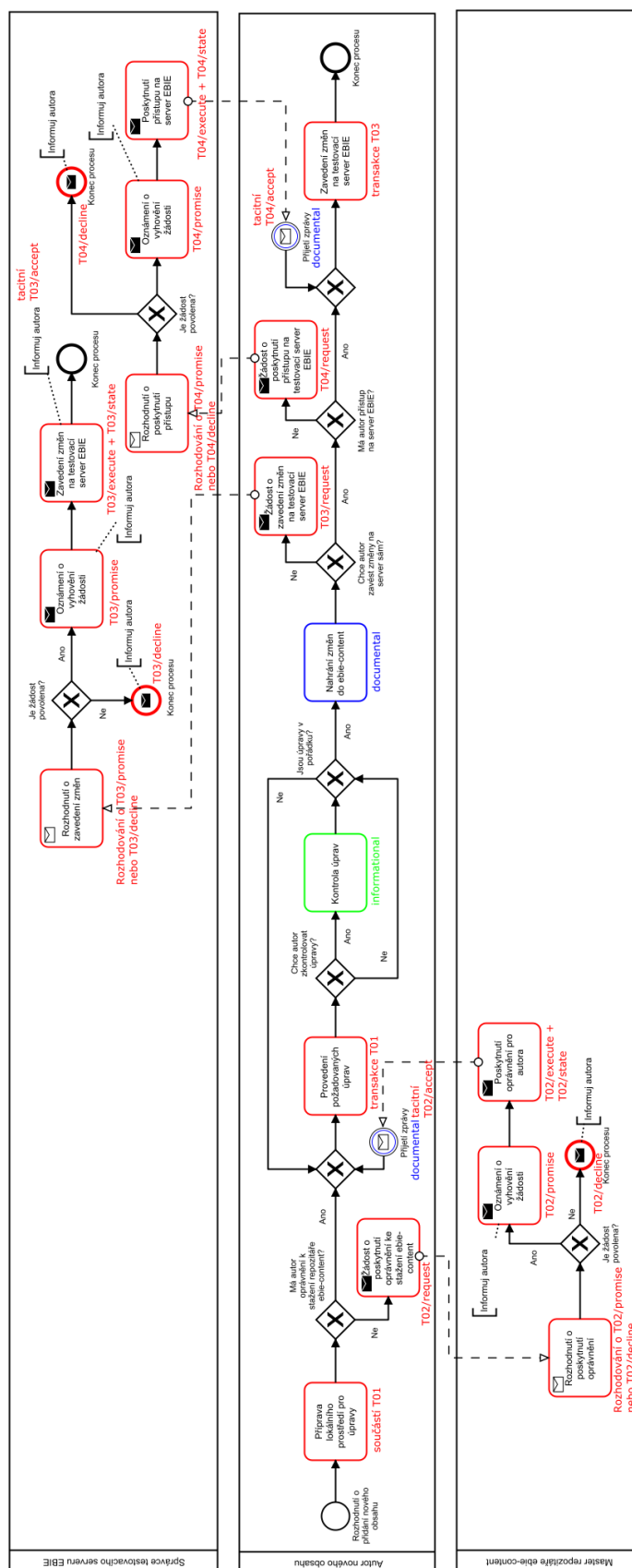
- **Ontologické** – V diagramu červená barva ohraničení elementu a označení stavu transakce, případně celé transakce.
- **Infologické** – V diagramu zelená barva ohraničení elementu a označení „informational“.
- **Datalogické** – V diagramu modrá barva ohraničení elementu a označení „documental“.

Všimněme si, že například všechny žádosti (aktivity s tmavou obálkou znázorňující odeslání) účastníků z diagramu, jsou označeny červenou barvou a spadají tak do kategorie „ontological“, i když odeslání této žádosti odpovídá akci „documental“. Pro nás je však důležitější, že se jedná o koordinační akt na úrovni performa – je to žádost o nový (originální) produkt, např. žádost o poskytnutí oprávnění ke stažení ebie-content. Tento případ nastává u více aktivit nejen v diagramech tohoto procesu.

4. MODELOVÁNÍ PROCESŮ PRO SPRÁVU OBSAHU



Obrázek 4.8: Publikování obsahu na testovací server (BPMN).



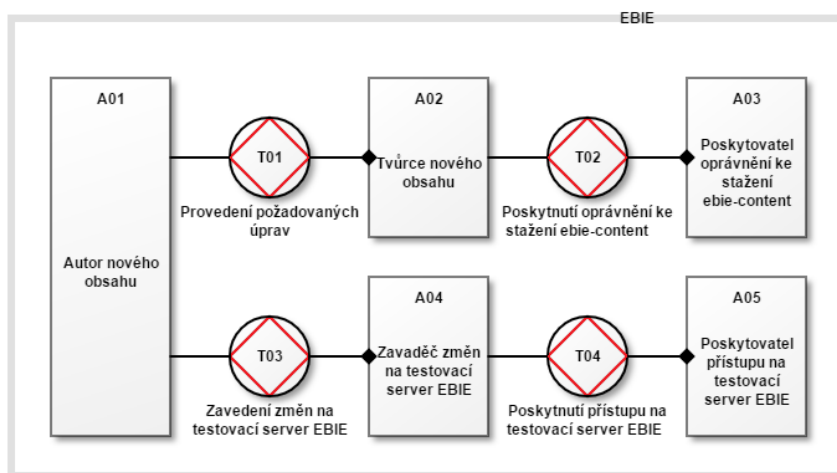
Obrázek 4.9: Publikování obsahu na testovací server (ontologický rozbor).

4.2.1.2 OCD a TPT

Prvním vytvořeným modelem je Konstrukční model, který je v našem případě reprezentován diagramem OCD (obrázek 4.10) a tabulkou TPT (4.1).

Z diagramu je patrné, že byly v procesu identifikovány čtyři ontologické transakce. K rozpoznání transakcí jsme provedli ontologický rozbor, viz diagram 4.9. Za zkoumanou oblast zájmů (Scope of Interest) byla zvolena „EBIE“. Posledním důležitým elementem v diagramu jsou aktorské role, kterých bylo definováno pět: A01 Autor nového obsahu, A02 Tvůrce nového obsahu, A03 Poskytovatel oprávnění ke stažení ebie-content, A04 Zavaděč změn na testovací server a A05 Poskytovatel přístupu na testovací server.

V tabulce jsou vyjmenovány jednotlivé názvy transakcí (transaction kind), stejně jako jejich výsledné produkty (product kind).



Obrázek 4.10: Publikování obsahu na testovací server (OCD).

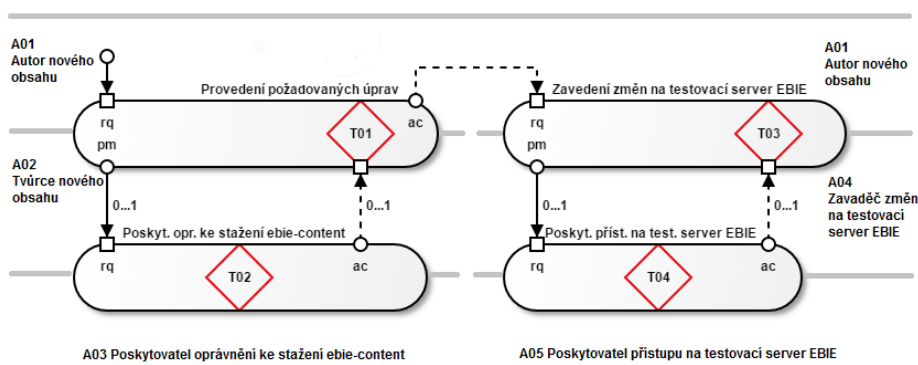
Transaction kind	Product kind
T01 Provedení požadovaných úprav	P01 Úpravy jsou provedeny
T02 Poskytnutí oprávnění ke stažení ebie-content	P02 Oprávnění je poskytnuto
T03 Zavedení změn na testovací server EBIE	P03 Změny jsou zavedeny
T04 Poskytnutí přístupu na testovací server EBIE	P04 Přístup je poskytnut

Tabulka 4.1: Publikování obsahu na testovací server (TPT).

4.2.1.3 PSD a AOM

Procesní model jsme znázornili pomocí PSD (obrázek 4.11). Z diagramu je patrná návaznost jednotlivým transakcím:

- Transakce T01 může být závislá na produktu transakce T02 – např.: Dokud není vytvořen P02 Oprávnění je poskytnuto (oprávnění ke stažení ebie-content), nemůže se provést spuštění (execute) transakce T01. „Může být“ je v diagramu znázorněno textem: „0...1“ vedle šipky značící request. Transakce T02 při případném request na „Poskytnutí oprávnění ke stažení ebie-content“ čeká na vytvoření produktu P02 (může i nemusí čekat, text „0...1“ vedle šipky značící accept). To odpovídá skutečnosti z BPMN diagramu (viz 4.8), kdy Autor nového obsahu může i nemusí žádat o oprávnění – pokud ho nemá, musí o něj požádat a počkat na jeho přidělení, jinak nemůže vytvořit nový obsah (provést úpravy).
- Stejný případ platí také pro transakci T03 a transakci T04, kdy aktorská role A04 Zavaděč změn na testovací server EBIE může i nemusí požádat o „Poskytnutí přístupu na testovací server EBIE“ v závislosti na faktu, zdali již má, nebo nemá přístup přidělen.



Obrázek 4.11: Publikování obsahu na testovací server (PSD).

Z předchozího popisu návaznosti transakcí jsme si mohli všimnout, že při vyjmenovávání aktorů není zcela zřejmé, jestli se jedná o aktory (aktorské role) z prostředí DEMO nebo účastníky (skutečné aktory) z prostředí organizace (v našem případě účastníky BPMN). Ke zjištění návaznosti mezi aktorskými rolami a skutečnými aktory je vhodné vyrobiť matici mapování.

Actor Organization Matrix (AOM) – Někdy nazývána Actor Function Matrix, je matice, která ukazuje rozdělení aktorů (aktorských rolí) na reálné pozice v podniku (v našem případě účastníky z BPMN). Aktor je v rámci DEMO kombinací zodpovědnosti a autority k provádění daných C-actů a

P-actů, z čehož plyne, že pro každý C-act a P-act musí být jasné, kdo ho vykonává. Pomocí plaveckých drah z BPMN, které představují určitou roli, nějakého člověka či nějaké oddělení v rámci organizace, je to možné provést. V BPMN můžeme navíc pojmenovávat plavecké dráhy podle reálných názvů a člověk uvedený v plavecké dráze může provádět více transakcí, což jsme mohli zaznamenat už v prvním ontologickém rozboru BPMN diagramu (viz 4.9). V DEMO se naopak aktoři většinou pojmenovávají podle akce/výkonu, kterou provádějí (např. Poskytovatel oprávnění ke stažení ebie-content) a mohou vykonávat právě jednu transakci [42].

Matice (tabulka 4.2) může obsahovat dvě hodnoty, nepočítáme-li prázdné pole (citováno ze zdroje [43]):

- **A** – Aktor je zodpovědný za určitou aktorskou roli.
- **D** – Aktor je (může být) delegovaný k vykonání dané aktorské role.

Například první řádek této matice reprezentuje aktora „Autor nového obsahu“, u kterého můžeme vyzorovat, že je zodpovědný za role A01 Autor nového obsahu a A02 Tvůrce nového obsahu. Autor nového obsahu může být navíc delegovaný k vykonání role A04 Zavaděč změn na testovací server EBIE.

	A01	A02	A03	A04	A05
Autor nového obsahu	A	A		D	
Master repozitáře ebie-content			A		
Správce testovacího serveru EBIE				A	A

Tabulka 4.2: Publikování obsahu na testovací server (AOM).

4.2.2 Publikování obsahu na produkční server

Obecně je proces Publikování obsahu na produkční server podobný svou strukturou prvnímu procesu Publikování obsahu na testovací server, což by pro nás nemělo být po prostudování analýzy žádným překvapením. Podobnost procesů se nám promítne také v samotných diagramech.

4.2.2.1 BPMN diagram a diagram ontologického rozboru

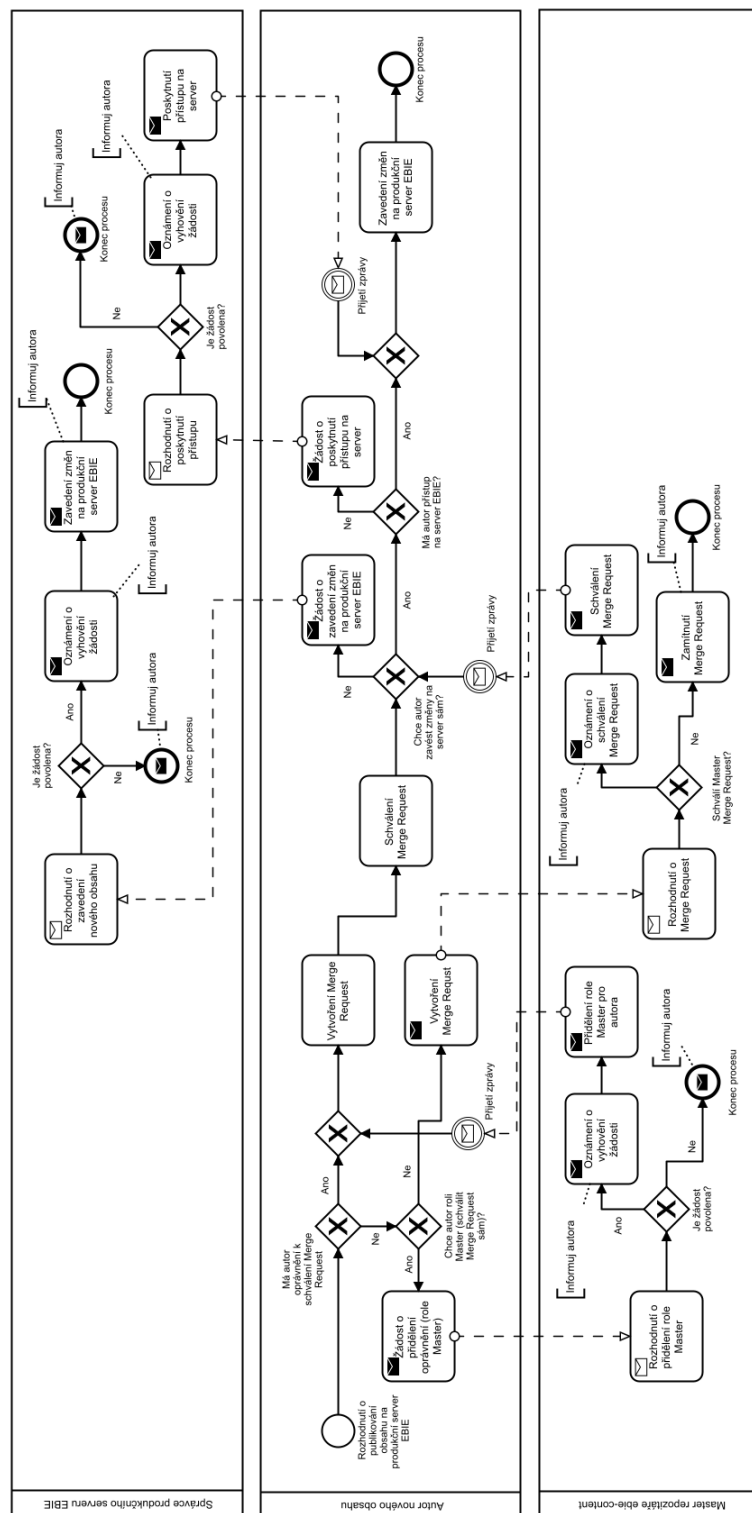
Na diagramu BPMN (obrázek 4.12) je tok procesu předáván mezi třemi účastníky (aktory): Autor nového obsahu, Master repozitáře ebie-content a Správce produkčního serveru EBIE.

Z diagramu můžeme vypočítat, že proces začíná od kroku číslo 6 z analýzy procesu v části textu 3.2.1.1 a tedy vytvořením Merge Request. Merge Request vytváří Autor nového obsahu a může ho schválit buď on sám, pokud na to má oprávnění, nebo musí požádat Mastera repozitáře ebie-content o jeho přidělení, případně jen o samotné schválení. Na základě rozhodnutí autora nového obsahu se tedy tok procesu může různě přelévat. Po schválení Merge Request je tok procesu téměř identický s předchozím procesem, nicméně Autor nového obsahu může požádat o přístup/zavedení Správce produkčního serveru EBIE, nikoliv Správce testovacího serveru EBIE, což ve skutečnosti může být jedna osoba, ale nemusí.

Stejně jako v procesu Publikování obsahu na testovací server může Autor nového obsahu zavést nový obsah/změny na server sám, pokud má přidělena všechna práva a přístupy.

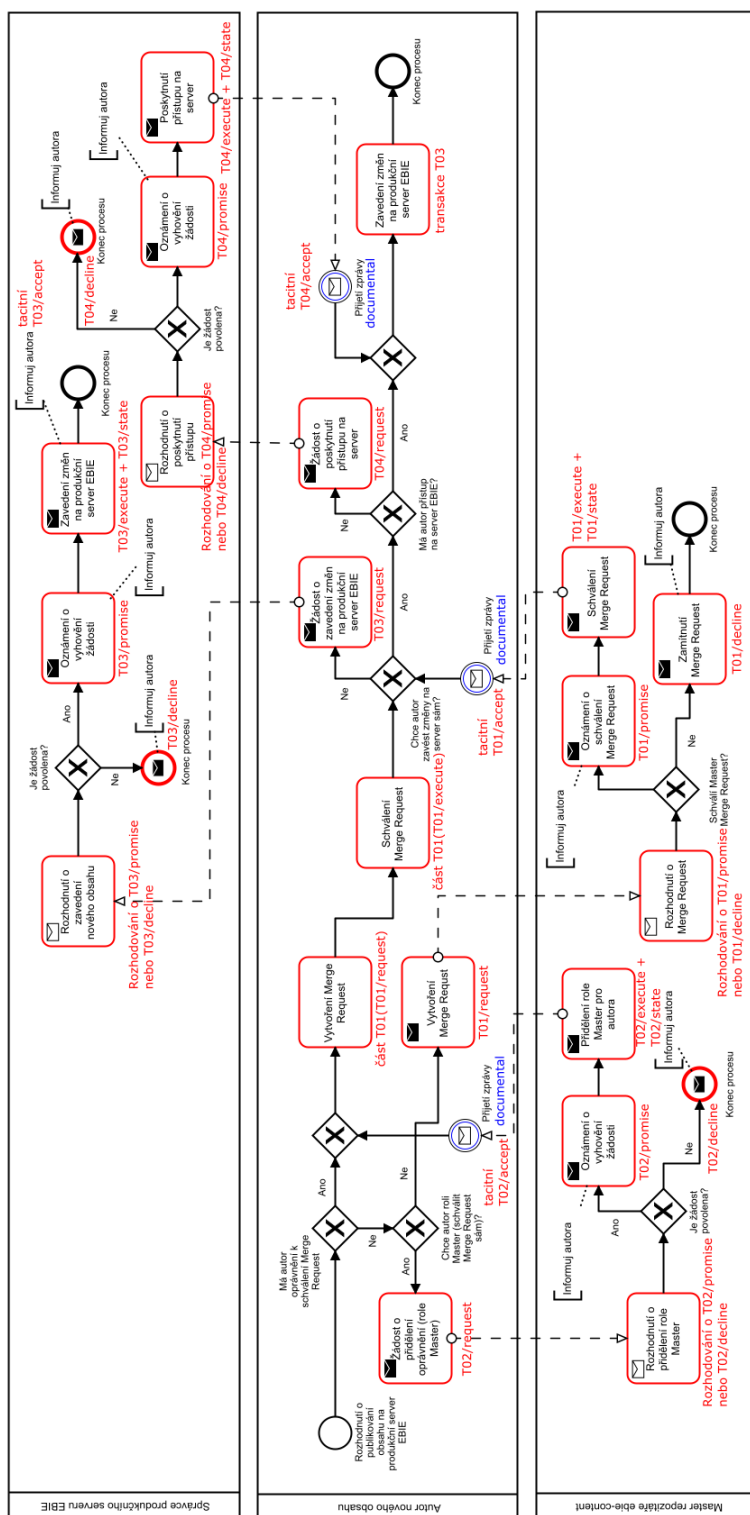
Druhý diagram (obrázek 4.13) znovu reprezentuje ontologický rozbor procesu.

4. MODELOVÁNÍ PROCESŮ PRO SPRÁVU OBSAHU



Obrázek 4.12: Publikování obsahu na produkční server (BPMN).

4.2. Modelování analyzovaných procesů

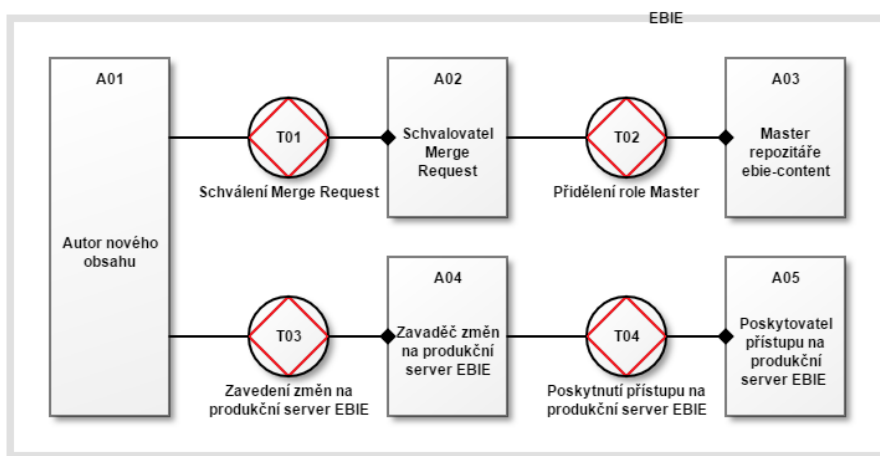


Obrázek 4.13: Publikování obsahu na produkční server (ontologický rozbor).

4.2.2.2 OCD a TPT

Konstrukční model pro proces Publikování obsahu na produkční server znovu popisuje diagram OCD (obrázek 4.14) a tabulka TPT (4.3).

V diagramu jsou stejně jako v případě předchozího procesu celkem čtyři ontologické transakce a pět aktorských rolí: A01 Autor nového obsahu, A02 Schvalovatel Merge Request, A03 Master repozitáře ebie-content, A04 Zavaděč změn na produkční server EBIE a A05 Poskytovatel přístupu na produkční server EBIE. Zkoumaná oblast zájmu byla vybrána znovu „EBIE“.



Obrázek 4.14: Publikování obsahu na produkční server (OCD).

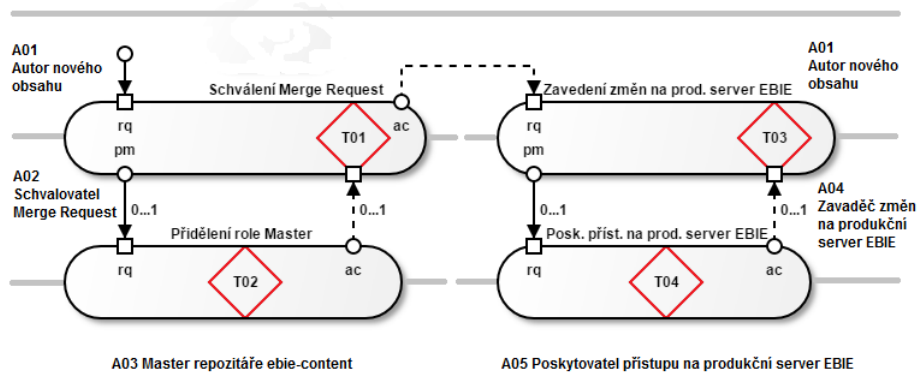
Transaction kind	Product kind
T01 Schválení Merge Request	P01 Merge Request je schválen
T02 Přidělení role Master	P02 Role Master je přidělena
T03 Zavedení změn na produkční server EBIE	P03 Změny jsou zavedeny
T04 Poskytnutí přístupu na produkční server EBIE	P04 Přístup je poskytnut

Tabulka 4.3: Publikování obsahu na produkční server (TPT).

4.2.2.3 PSD a AOM

Procesní model procesu tvoří diagram PSD (obrázek 4.15). Stejně jako v případě prvního procesu se zde setkáváme s „podmíněnými“ transakcemi:

- Transakce T01 může být závislá na dokončení transakce T02 – čekání na případné přidělení role Master, pokud ještě není udělena Schvalovateli Merge Request.
- Transakce T03 může být závislá na dokončení transakce T04 – čekání na případné přidělení přístupu na produkční server EBIE pro Zavaděče změn na produkční server EBIE.



Obrázek 4.15: Publikování obsahu na produkční server (PSD).

Mapování aktorů na aktorské role je vyjádřené maticí AOM (tabulka 4.4. Pro Autora nového obsahu je zodpovědný za roli, nesoucí stejný název (A01 Autor nového obsahu). Zajímavějším faktem je, že mu mohou být delegovány tři další role, konkrétně A02, A03 a A04. Autor nového obsahu tak v rámci tohoto procesu v určitých případech může být delegován do rolí Schvalovatele Master Request, Master repozitáře ebie-content (byla mu přidělena role Master) a Zavaděče změn na produkční server.

Master repozitáře ebie-content zastává dvě role, stejně jako Správce produkčního serveru.

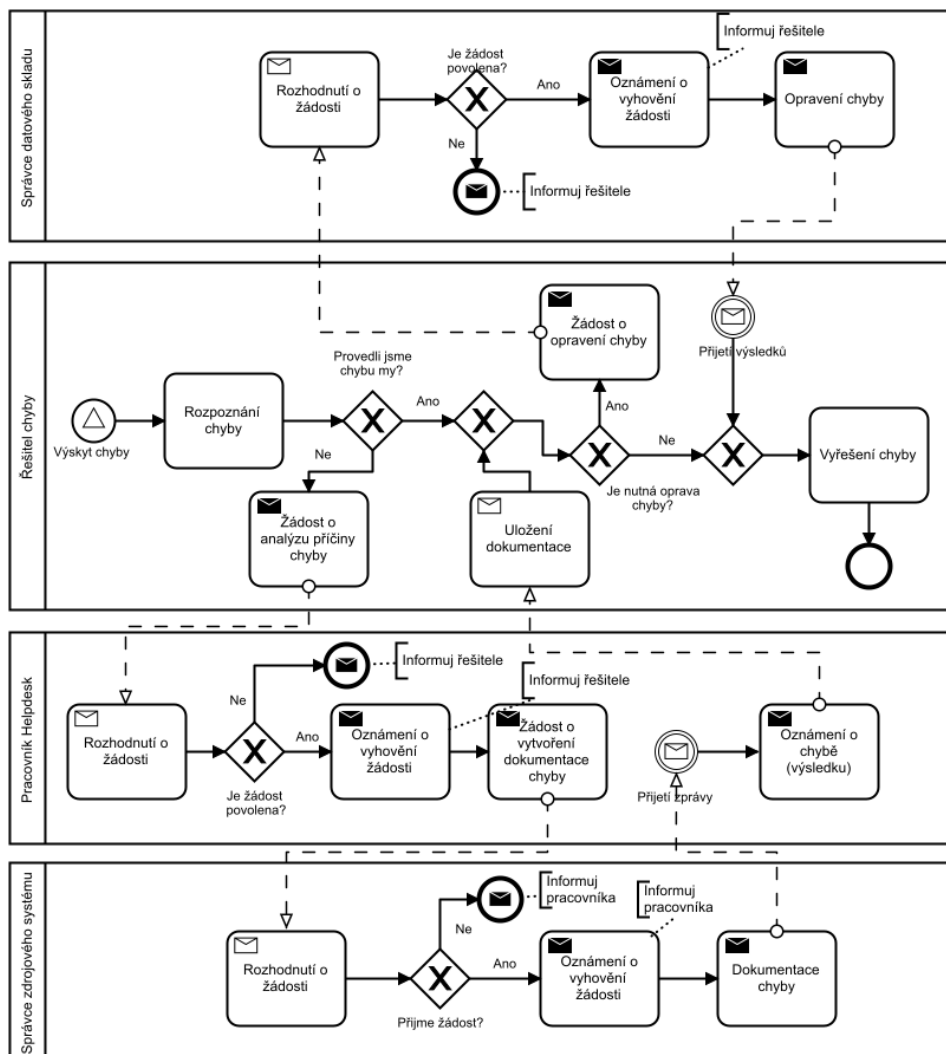
	A01	A02	A03	A04	A05
Autor nového obsahu	A	D	D	D	
Master repozitáře ebie-content		A	A		
Správce produkčního serveru EBIE				A	A

Tabulka 4.4: Publikování obsahu na produkční server (AOM).

4.2.3 Nahlášení chyby ve zdrojových systémech

4.2.3.1 BPMN diagram a diagram ontologického rozboru

Na snímku 4.16 je namodelovaný BPMN diagram procesu Nahlášení chyby ve zdrojových systémech. Součástí tohoto procesu je také určení postupu, jak danou chybu vyřešit.



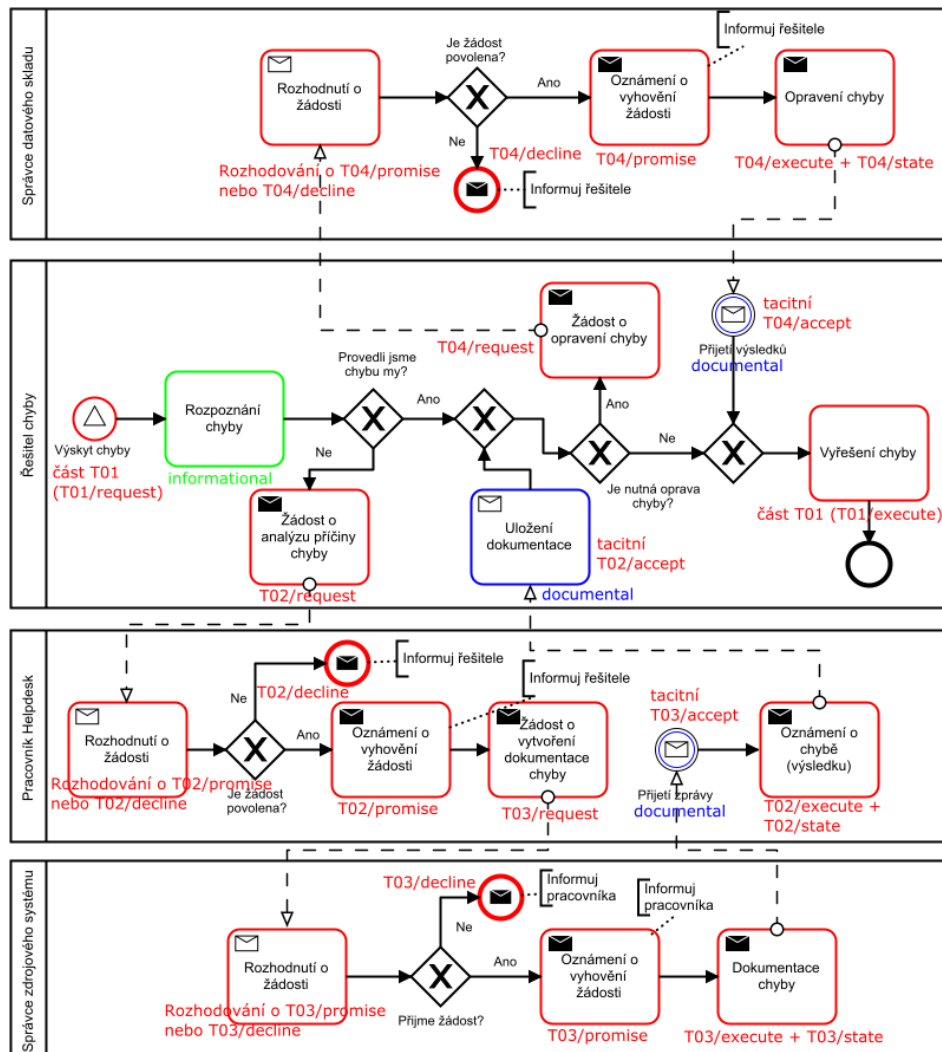
Obrázek 4.16: Nahlášení chyby ve zdrojových systémech (BPMN).

Na diagramu jsou čtyři účastníci: Řešitel chyby, Pracovník Helpdesk, Správce zdrojového systému a Správce datového skladu. Řešitel chyby po rozpoznání chyby určuje další kroky procesu (posloupnost aktivit a událostí), které byly analyzovány v části 3.2.2.2.

4.2. Modelování analyzovaných procesů

Namodelovaný postup je následující: Jestliže Řešitel potřebuje zjistit příčinu chyby, tok se přelévá k Pracovníkovi Helpdesk (zde se skrývá nahlášení chyby), který žádost o analýzu chyby zpracuje a sám požádá Správce zdrojového systému, aby chybu zdokumentoval/vysvětlil. Na základě odpovědi od Správce posílá Pracovník Helpdesk odpověď s případnou dokumentací zpátky Řešiteli. Pokud je nutné opravení chyby na úrovni datového skladu, pošle Řešitel žádost o opravení a počká na výsledek. Potom může chybu označit jako vyřešenou.

Na diagramu (obrázek 4.17) je ontologický rozbor procesu Nahlášení chyby ve zdrojových systémech.



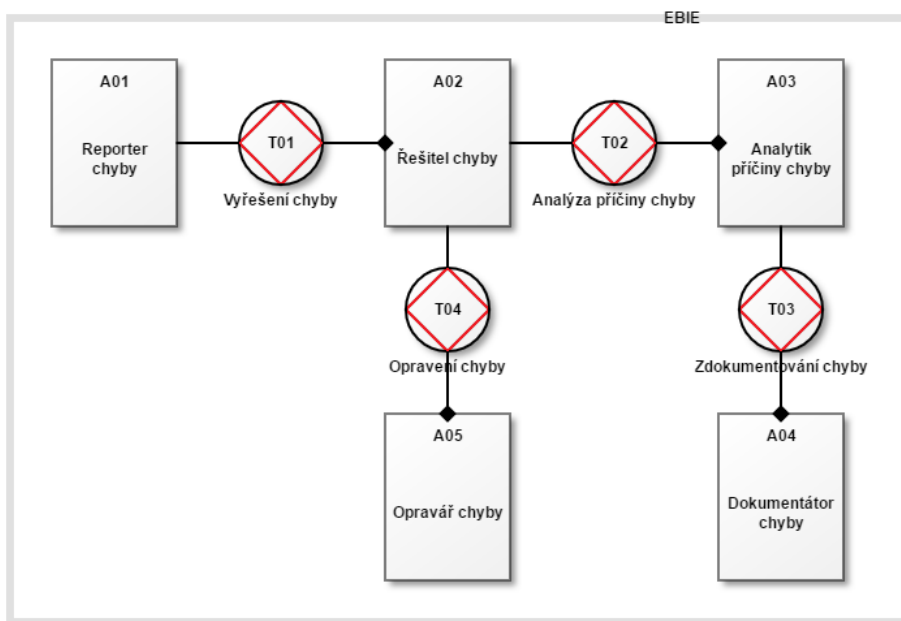
Obrázek 4.17: Nahlášení chyby ve zdrojových systémech (ontologický rozbor).

4.2.3.2 OCD a TPT

Konstrukční model pro proces Nahlášení chyby ve zdrojových systémech se skládá z OCD (obrázek 4.18) a TPT (4.5).

Struktura diagramu je znovu rozdělena na čtyři ontologické transakce a pět rolí aktorů: A01 Reporter chyby, A02 Řešitel chyby, A03 Analytik příčiny chyby, A04 Dokumentátor chyby a A05 Opravář chyby. Zkoumaná oblast zájmů je „EBIE“.

V tabulce jsou vypsány názvy transakcí a jejich produkty.



Obrázek 4.18: Nahlášení chyby ve zdrojových systémech (OCD).

Transaction kind	Product kind
T01 Vyřešení chyby	P01 Chyba je vyřešena
T02 Analýza příčiny chyby	P02 Příčina chyby je analyzována
T03 Zdokumentování chyby	P03 Chyba je zdokumentována
T04 Opravení chyby	P04 Chyba je opravena

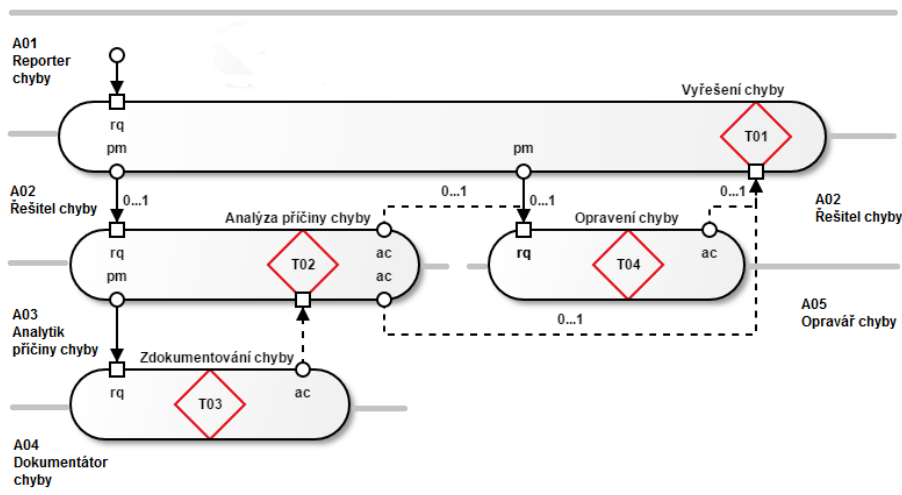
Tabulka 4.5: Nahlášení chyby ve zdrojových systémech (TPT).

4.2.3.3 PSD a AOM

Diagram PSD (obrázek 4.19) obsahuje návaznosti transakcí procesu:

- Transakce T01 Vyřešení chyby může být v určitých případech vyřešena až tehdy, pokud je zjištěna příčina chyby (P02 Příčina chyby je analyzována), případně až když je chyba opravena (P04 Chyba je opravena). Pokud nahlašujeme chybu a následně chybu v datovém skladu opravujeme, musí transakce T01 čekat na oba tyto výsledky/produkty.
- Transakce T02 Analýza příčiny chyby nemůže být vykonána do té doby, dokud není chyba zdokumentována (proběhne transakce T03 Zdokumentování chyby a je vyroben produkt P03 Chyba je zdokumentována).

Matrice AOM (tabulka 4.6) definuje mapování aktorů organizace na aktorské role. Řešitel chyby zastává v tomto případě roli A01 Reporter chyby a A02 Řešitel chyby. V určitých případech mu může být delegována i role A05 Opravář chyby. Podobné příklady platí také pro ostatní aktory (viz tabulka).



Obrázek 4.19: Nahlášení chyby ve zdrojových systémech (PSD).

	A01	A02	A03	A04	A05
Řešitel chyby	A	A			D
Pracovník Helpdesk			A	D	
Správce zdrojového systému			D	A	
Správce datového skladu	D	D			A

Tabulka 4.6: Nahlášení chyby ve zdrojových systémech (AOM).

4.2.4 Zavedení kontroly kvality dat

Proces Zavedení kontroly kvality dat je první částí analyzovaného procesu, viz analýza v 3.2.2.2. Proces se zabývá kompletním nasazením kontroly kvality dat nad vybranými položkami, pro které je vytvořena dokumentace a provedena implementace kontroly kvality dat na vybraném místě (ETL jobu). K procesu jsou vytvořeny DEMO diagramy a tabulky.

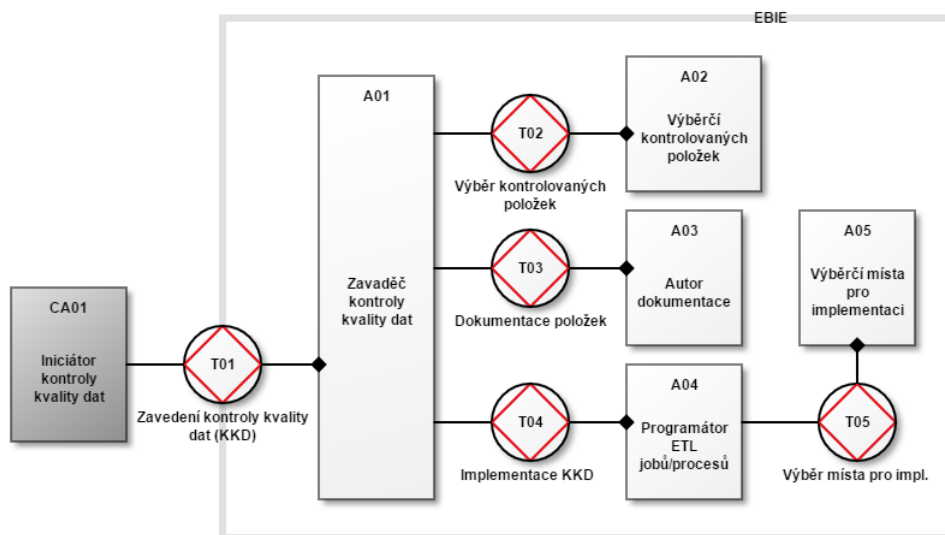
4.2.4.1 OCD a TPT

Diagram OCD (obrázek 4.20) a tabulka TPT (4.7) popisují základní strukturu procesu.

V diagramu se vyskytuje šest aktorských rolí, z nichž jedna je tzv. kompozitní externí role. Externí role je zde kvůli tomu, že role vstupuje do procesu zvenčí (je mimo Scope of Interest, kterým je „EBIE“). Slovo kompozitní z důvodu, že se pod touto rolí mohou skrývat další (pod)role [46]. Tuto roli poznáme v diagramu podle šedé výplně a označení prvním písmenem „C“ (composite, kompozitní).

Aktorské role: CA01 Iniciátor kontroly kvality dat, A01 Zavaděč kontroly kvality dat, A02 Výběrčí kontrolovaných položek, A03 Autor dokumentace, A04 Programátor ETL jobů/procesů a A05 Výběrčí místa pro implementaci.

V tabulce je vypsán seznam všech pěti ontologických transakcí a jejich produktů.



Obrázek 4.20: Zavedení kontroly kvality dat (OCD).

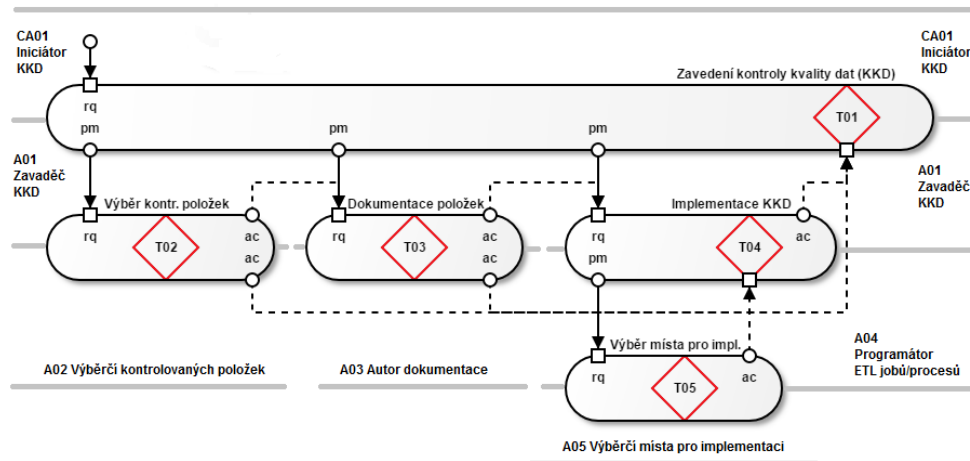
Transaction kind	Product kind
T01 Zavedení kontroly kvality dat	P01 Kontrola kvality dat je zavedena
T02 Výběr kontrolovaných položek	P02 Položky jsou vybrány
T03 Dokumentace položek	P03 Položky jsou zdokumentovány
T04 Implementace kontroly kvality dat	P04 Kontrola kvality dat je implementována
T05 Výběr místa pro implementaci	P05 Místo pro implementaci je vybráno

Tabulka 4.7: Zavedení kontroly kvality dat (TPT).

4.2.4.2 PSD

Z diagramu PSD (obrázek 4.21) je vidět, že vždy musí proběhnout všechny transakce pro dokončení procesu. Většina transakcí na sebe navazuje a znovu čekají na vytvoření produktu z jiné transakce (dokončení jiné transakce):

- Transakce T01 čeká na dokončení transakcí T02, T03 a T04.
- Transakce T03 čeká na dokončení transakce T02.
- Transakce T04 čeká na dokončení transakce T03 a je navíc vykonána až po vytvoření/přijmutí produktu P05 (viz TPT 4.7).



Obrázek 4.21: Zavedení kontroly kvality dat (PSD).

4.2.5 Aplikování kontroly kvality dat

Posledním namodelovaným procesem je proces Aplikování kontroly kvality dat, který řeší druhou analyzovanou část, viz 3.2.3.2, zaobírající se samotnými reporty chyb. Pro tento proces byly vyrobeny pouze diagramy a tabulky pomocí metodiky DEMO.

4.2.5.1 OCD a TPT

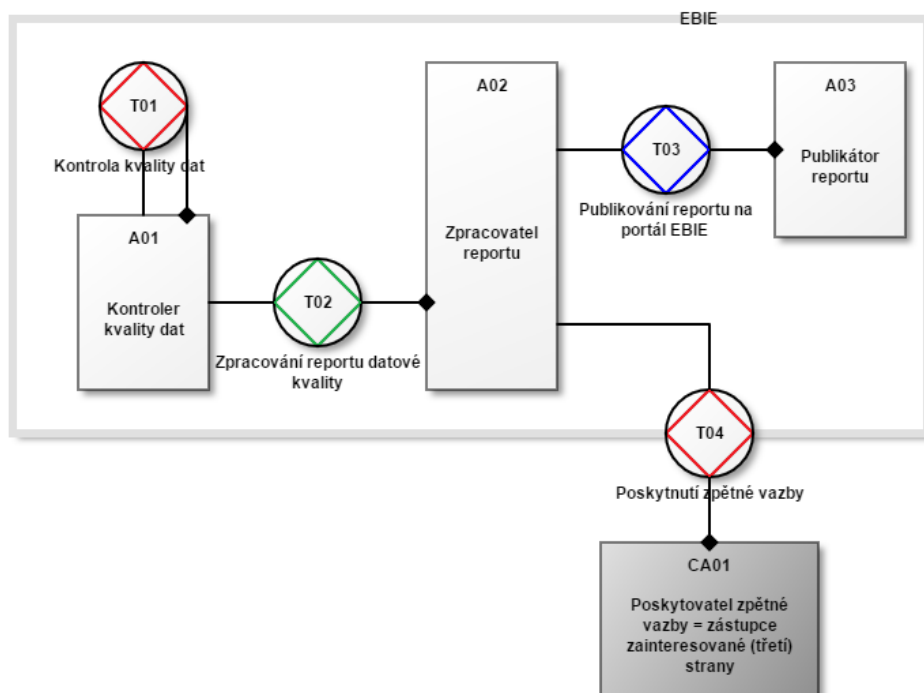
Na rozdíl od všech předchozích, neobsahuje diagram OCD (obrázek 4.22) procesu Aplikování kontroly kvality dat pouze ontologické transakce, ale také jednu infologickou a datalogickou transakci, které pracují se zmiňovanými reporty. Celkově se v diagramu vyskytují čtyři transakce (dvě ontologické):

- Transakce T01 Kontrola kvality dat reprezentuje samotný „proces“ kontroly kvality dat, který je spouštěn periodicky. Transakce T01 se řadí mezi self-activated transakce, o kterých jsme si mohli něco přečíst v popisu metodiky DEMO, viz podsekcce 4.1.2, a má tak stejného iniciátora i vykonavatele (A01 Kontroler kvality dat).
- Transakce T02 Zpracování reportu datové kvality pracuje s vytvořeným reportem z první transakce. K tomuto zpracování jsou potřebné nějaké konkrétní informace – proto je označena jako infologická transakce.
- Transakce T03 Publikování reportu na portál EBIE pracuje jako předchozí transakce s reportem a spadá do datalogické úrovně (nahrání reportu na server).
- Exekutorem transakce T04 Poskytnutí zpětné vazby je kompozitní aktorská role CA01 Poskytovatel zpětné vazby, která odpovídá Zástupci zainteresované (třetí) strany zmiňovaném v analýze procesu.

Zkoumanou oblastí zájmu je stejně jako ve všech procesech „EBIE“. V OCD můžeme vyzorovat čtyři aktorské role, které byly pro proces definovány: CA01 Poskytovatel zpětné vazby = zástupce zainteresované (třetí) strany, A01 Kontroler kvality dat, A02 Zpracovatel reportu a A03 Publikátor reportu.

V TPT (4.8) jsou k vyjmenovaným názvům transakcí popsány jejich produkty.

4.2. Modelování analyzovaných procesů



Obrázek 4.22: Aplikování kontroly kvality dat (OCD).

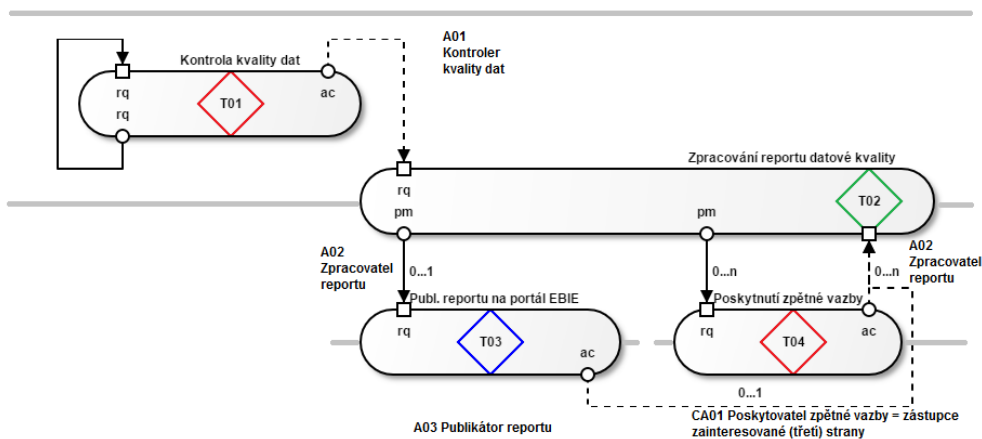
Transaction kind	Product kind
T01 Kontrola kvality dat	P01 Report je vytvořen
T02 Zpracování reportu datové kvality	P02 Report datové kvality je zpracován
T03 Publikování reportu na portál EBIE	P03 Report je publikován na portál EBIE
T04 Poskytnutí zpětné vazby	P04 Zpětná vazba je poskytnuta

Tabulka 4.8: Aplikování kontroly kvality dat (TPT).

4.2.5.2 PSD

Procesní model je opakovaně ztvárněn diagramem PSD (obrázek 4.23) u kterého oproti jiným PSD diagramům můžeme zaregistrovat dvě největší odlišnosti:

- První transakce je self-activated (iniciována a vykonávána stejným aktorem), čímž je znázorněna pomocí šipky tvořící „cyklus“ v pomyslném tvaru obdélníku.
- Transakce T02 Zpracování reportu datové kvality čeká na výsledné produkty transakcí T03 a T04. Právě pro transakci T04 může a nemusí dojít k jejímu provedení. K tomuto provedení může dojít vícekrát (o chybě může být informováno více třetích stran a my než vyřešíme chybu, čekáme na jejich vyjádření). Tento fakt (více „spuštění“ transakce) je v diagramu vyznačen textem „0...n“.



Obrázek 4.23: Aplikování kontroly kvality dat (PSD).

Publikování obsahu a vyhodnocení integračního plánu

V poslední kapitole provedeme simulaci prvního procesu na konkrétním případě, kdy se pokusíme publikovat nový obsah na testovací server EBIE. Tímto obsahem budou namodelované procesy z předchozí kapitoly. Ve druhé části vyhodnotíme integrační plán, který byl sestaven v sekci 2.2.

5.1 Ukázka publikování obsahu na testovací server EBIE

Názorná ukázka publikování obsahu na testovací server EBIE se udála v režii autora práce a publikovaných procesů.

Prvním krokem bylo připravení lokálního prostředí. Na své instalaci operačního systému Ubuntu 12.04 jsem si připravil prázdný adresář, provedl jsem instalaci potřebných technologií, viz seznam v části 3.2.1.3. Jelikož jsem už měl nastavený přístup do repozitáře ebie-content, mohl jsem provést jeho stažení pomocí příkazu:

```
git clone git@gitlab.fit.cvut.cz:hajcidav/ebie-content.git
```

Dalším postupem bylo vytvoření požadovaných úprav. V rámci konzultace s vedoucím práce jsme dospěli k rozhodnutí, že nejvhodnější lokace pro umístění nových procesů bude modul „Dokumentace pro vývojáře“, ve kterém bych měl vytvořit novou sekci „Procesy“ a následně ji naplnit obsahem a provést tak integraci těchto procesů. Prvním krokem v úpravách bylo vytvoření nové sekce (podadresáře v adresáři „dokumentace“). Do založeného adresáře „processes“ jsem doplnil další podadresáře „assets/img“ pro diagramy procesů, kam jsem je také vložil, a soubor info.json s vyplněným obsahem pro strukturu nové sekce. Část souboru (definování prvního procesu v sekci) si můžeme prohlédnout na snímku 5.1. Do souboru logic.adoc, který je umístěn v kořenovém adresáři repozitáře, jsem přidal nový atribut:

:dokumentace-processes: /modules/dokumentace/processes/view?path=&filename=

Dále jsem vytvořil nové .adoc soubory, do kterých jsem vyplnil obsah jednotlivých procesů podle jejich šablony z adresáře „lib“, která už se v repozitáři nacházela. Kontrolu jednotlivých úprav v lokálním prostředí jsem neprováděl a tak jsem mohl přistoupit k nahrání změn zpátky na GitLab, čehož jsem docílil pomocí sekvence příkazů uvedené ve čtvrtém bodu seznamu: „Nahrání změn do ebie-content“ z části textu 3.2.1.1.

```

1 {
2   "title" : "Procesy",
3   "description" : "Procesy",
4   "item_node" : false,
5   "list" : [
6     {
7       "title" : "Publikování obsahu na testovací server EBIE",
8       "description" : "Proces popisující publikování nového obsahu na testovací server EBIE.",
9       "path" : ".",
10      "filename" : "pub_obsahu_test_server.html",
11      "assets" : {
12        "css" : [],
13        "img" : [],
14        "js" : []
15      }
16    },

```

Obrázek 5.1: Ukázka vytvořeného souboru info.json.

Dalším krokem, který bylo potřeba vykonat, bylo zavedení nových změn na testovací server EBIE. K tomuto úkolu jsem využil technologie VPN a SSH, pomocí kterých jsem se připojil na požadovaný server VICu, na který jsem měl již zajištěný přístup, a proto jsem o něj nemusel žádat. Příkaz ke vzdálenému připojení vypadal následovně:

```
ssh root@ebie-vyvoj.is.cvut.cz
```

Na serveru jsem postupoval podle návodu – přihlásil jsem se pod uživatelem ebie, dostal se do adresáře ebie-web a spustil dva příkazy:

```
git submodule update --remote
```

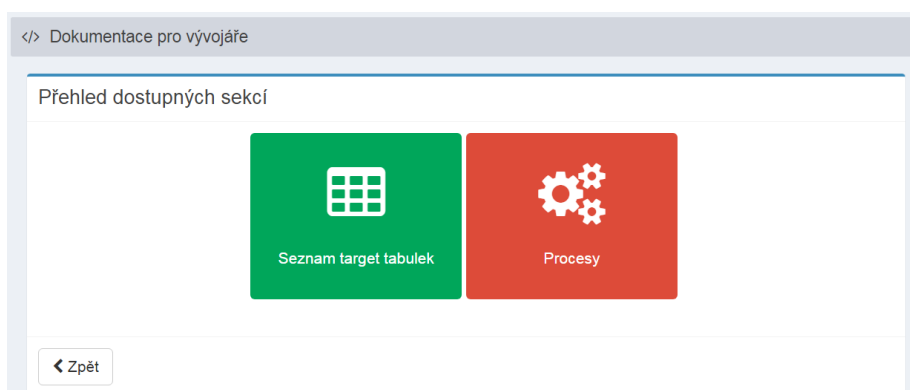
```
bundle exec rails generate content
```

Nový obsah byl tímto momentem zaveden na testovací server EBIE. Po připojení na server pod svým uživatelským jménem, přes které mám umožněn přístup (aktuálně přes zmiňovaný whitelist), jsem se podíval na výsledek a pořídil pro ilustraci tři snímky:

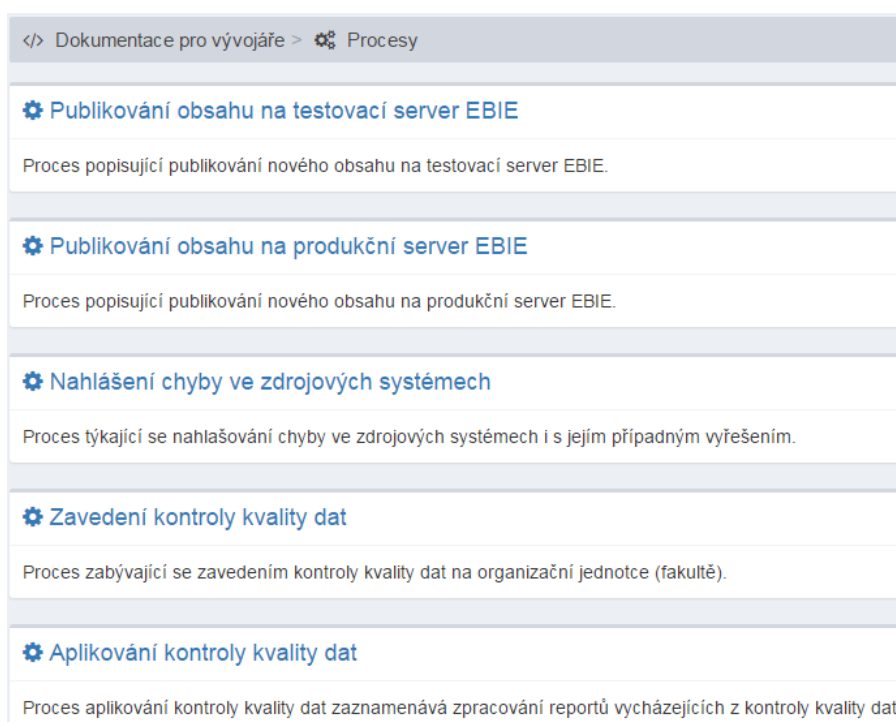
- Snímek 5.2 – Ukázka nově vytvořené sekce Procesy v modulu Dokumentace pro vývojáře.
- Snímek 5.3 – Vizualizuje seznam všech integrovaných procesů této práce do portálu EBIE.

5.1. Ukázka publikování obsahu na testovací server EBIE

- Snímek 5.4 – Část stránky procesu Publikování obsahu na testovací server EBIE. K nahlédnutí je stručný popis procesu a jeho OCD diagram.

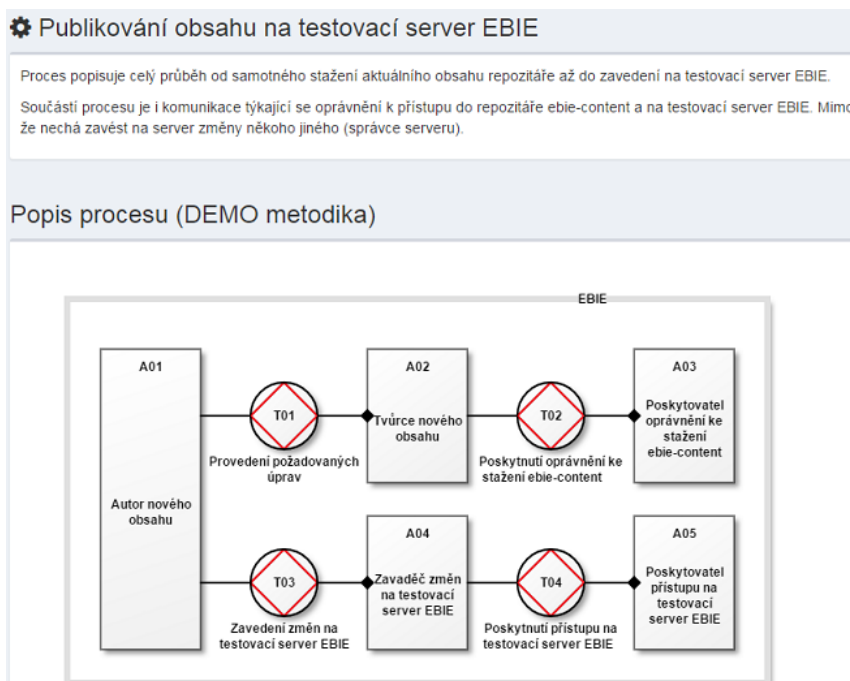


Obrázek 5.2: Ukázka modulu Dokumentace pro vývojáře se sekci Procesy.



Obrázek 5.3: Ukázka seznamu procesů v sekci Procesy.

5. PUBLIKOVÁNÍ OBSAHU A VYHODNOCENÍ INTEGRAČNÍHO PLÁNU



Obrázek 5.4: Ukázka procesu Publikování obsahu na testovacím serveru EBIE.

5.2 Vyhodnocení integračního plánu

Jedním z cílů práce bylo vypracovat plán pilotního nasazení portálu EBIE a plán integrace částí rozebraných v podsekcí 1.3.3 se zohledněním výstupů této práce, čili namodelovanými procesy. Tyto úseky, které byly shrnuty do jednoho komplexního plánu, byly navrženy v sekci 2.2. Výsledný projekt je zobrazen na obrázku 2.1. Vyhodnocení části nasazení bylo zpracováno již v podkapitole 2.3. Návrh integrace, který byl v komplexním plánu nazván „Integrace a úprava EBIE“ a v době vyhodnocení nasazení stále probíhal, bude vyhodnocen v této podkapitole.

5.2.1 Rozhodnutí o zbývajících integrovaných částech

Integrační plán měl být podle projektu a časového plánu dokončen 9. května 2017. Od doby vyhodnocení plánu nasazení probíhaly z pohledu integrace dva úkoly:

- **Integrace JIRA** – Propojení systému JIRA pro podávání žádosti o novou sestavu jednotlivými uživateli.
- **Integrace dat/plnění obsahu II.** – Jedná se o plnění obsahu novými daty – relevantní procesy pro portál vytvořené v práci Ing. Ladislava

Moravce, představené v části textu 1.3.3.2, a procesy vzniklémi v této práci.

S ohledem na integraci výstupů analyzovaných prací byly tyto úkoly posledními, které bylo ještě potřeba splnit.

Co se týká Integrace JIRA, tato činnost byla přidělena členovi Bc. Cyrilovi Černému, který je součástí „Týmu EBIE“, představeného v podsekcí 2.1.1. Námět na přidělení úkolu tomuto členovi vznikl na základě faktu, podle kterého sahá úkol mimo rámec této práce a faktu, že byl sám pan Černý osobou, která měla za cíl ve své bakalářské práci [21] tuto integraci navrhnout a implementovat. Na přenechání odpovědnosti propojení systému JIRA s nasazeným portálem EBIE se shodl celý tým v čele s vedoucím této práce panem Ing. Michalem Valentou, Ph.D., který také souhlasil. V současné době integrace ještě neproběhla, a to především z důvodu rozvržení přidělování práv. Nový termín prozatím není stanoven a pravděpodobně bude naplánován někdy v průběhu tohoto roku.

První část Integrace dat/plnění obsahu II. měla najít relevantní procesy, které navrhl Ing. Ladislav Moravec, a následně je do portálu EBIE importovat. Po konzultaci s vedoucím práce jsme dospěli k závěru, že žádné procesy nejsou vhodné pro publikování do portálu.

Druhou částí Integrace dat/plnění obsahu II. a posledním úkolem integračního plánu bylo provést nahrání vytvořených procesů pro správu portálu EBIE, které vzešly z této práce, na portál EBIE. Integrace byla provedena, viz předchozí podkapitola 5.1, a shodou okolností se také jednalo o simulaci jednoho z vytvořených procesů Publikování obsahu na testovací server EBIE.

Závěr

Hlavním úkolem pro tuto práci bylo splnění dvou cílů.

Prvním z cílů bylo vytvoření komplexního plánu zahrnujícího plán pilotního nasazení portálu EBIE a plán integrace, který se skládal z výstupů prací definovaných v zadání. Tento plán jsem měl v roli projektového manažera realizovat. Druhým cílem bylo analyzování a následné namodelování procesů pro údržbu, plnění a rozšiřování obsahu portálu EBIE. K namodelování procesů mělo být využito metodiky DEMO a případně dalších zvolených postupů na vlastním uvážení. Jako druhá modelovací technika tak byla vybrána notace BPMN.

Práce byla systematicky rozdělena do pěti kapitol. V první kapitole byla provedena rešerše zmiňovaných prací, které souvisely s portálem EBIE. Ještě před tímto seznámením byly v kapitole vysvětleny pojmy datová čistota, datový sklad a EBIE, které měly čtenáře uvést do problematiky. V druhé kapitole byly představeny dva týmy, které se podílely na projektu a byly popsány zdokumentované schůzky, které probíhaly během jeho realizace. V další části kapitoly byl navržen souhrnný plán integrace a nasazení portálu. Tento návrh byl následně rozebrán a jeho část nasazení vyhodnocena. Ve třetí kapitole byl sepsán aktuální stav portálu EBIE do doby jeho nasazení. Následně byly analyzovány procesy, které byly vybrány na základě konzultace s vedoucím práce. V předposlední kapitole byly popsány základy DEMO a BPMN, jejich silné a slabé stránky a poté byly uvedeny rozdíly těchto dvou modelovacích technik. Po představení DEMO a BPMN byly namodelovány samotné procesy. Pátá kapitola slouží především jako ukázka publikování nového obsahu. Na samotném konci práce byla vyhodnocena integrační část projektu.

Plán nasazení a integrace proběhl podle časového harmonogramu. Portál EBIE byl nasazen na dvou serverech pro testovací a produkční verzi. Nejdůležitější části integrace proběhly bez problémů. Za všechny úkoly, za které jsem byl zodpovědný, byly v pořádku dokončeny a můžu tak konstatovat, že byl tento cíl splněn. Vytvořený plán však nenajde další využití a byl účelně navržen pouze pro nasazení portálu a jeho rozšíření pomocí integrovaných částí.

Portál EBIE naopak bude mít velké uplatnění nejen pro vedoucí pracovníky, které zde není potřeba více rozebírat.

Během tvorby práce bylo celkově vytvořeno pět procesů, z nichž všechny byly namodelovány pomocí diagramů a tabulek metodiky DEMO a tři z nich také pomocí notace BPMN. Tyto procesy dávají největší přidanou hodnotu práce. První dva procesy se zabývají publikováním nového obsahu na testovací a produkční server EBIE. Jejich využití je zřejmé už z jejich názvu. Procesy budou sloužit například jako návod správcům portálu EBIE i dalším zájemcům pro publikaci obsahu. Třetí proces pro nahlašování chyb ve zdrojových systémech definuje, jak nahlásit chybu, která se vyskytla v datovém skladu, a jak ji případně na základě poskytnuté dokumentace opravit. Poslední dva procesy pro kontrolu kvality dat ukazují, jak kontrolu kvality dat zavést a následně aplikovat, čili jak pracovat s jejími výstupy (reporty). Také druhý cíl byl tedy splněn.

Kromě dosažení zmíněných cílů byl v práci zdokumentován aktuální stav portálu EBIE, shrnuta metodika DEMO a notace BPMN a provedena názorná ukázka procesu pro publikování obsahu na testovací server EBIE.

Na práci lze navázat vytvořením dalších procesů pro správu portálu EBIE. Podle procesů kontroly kvality dat by bylo vhodné na úrovni fakulty tuto kontrolu zavést nad vybranými položkami pro otestování funkčnosti a sledovat její aplikování v praxi. Pokud by se kontrola kvality dat osvědčila, mohl by se v portálu vytvořit modul či sekce, do které by byly automaticky importovány výsledné reporty, aby se jejich nahrávání nemuselo provádět „ručně“. Dalším nápadem je zautomatizování dokumentace datového skladu ČVUT do portálu EBIE. V obou případech by měly zainteresované strany přístup do portálu, kde by měly reporty a dokumentaci samy k nahlédnutí. K umožnění těchto přístupů by mělo být dořešeno přidělování práv v portálu, a to zejména kvůli citlivosti některých dat. Tím je míněno, že by měla být vyřešena autorizace nad jednotlivými moduly a sekcemi v portálu EBIE.

Literatura

- [1] Vrána, J.: *Datové sklady a jejich optimalizace [online]*. 2001, [cit. 2017-02-05]. Dostupné z: <https://www.systemonline.cz/clanky/datove-sklady-a-jejich-optimalizace.htm>
- [2] Cardon, D.: *Database vs Data Warehouse: A Comparative Review [online]*. 2014, [cit. 2017-02-05]. Dostupné z: <https://www.healthcatalyst.com/database-vs-data-warehouse-a-comparative-review>
- [3] Polák, P.: *Internet věcí skýtá velké příležitosti [online]*. 2015, [cit. 2017-02-05]. Dostupné z: <https://www.systemonline.cz/it-pro-logistiku/internet-veci-skyta-velke-prilezitosti.htm>
- [4] Zýka, O.: *Datová kvalita [online]*. 2015, [cit. 2017-02-05]. Dostupné z: https://profinit.eu/wp-content/uploads/2015/12/02_Datova_kvalita.pdf
- [5] Rahm, E.; Do, H. H.: *Data Cleaning: Problems and Current Approaches [online]*. University of Leipzig, Germany, 2000, [cit. 2017-02-23]. Dostupné z: http://betterevaluation.org/sites/default/files/data_cleaning.pdf
- [6] Loshin, D.: *Master Data Management*. Boston: Elsevier/Morgan Kaufmann, 2009, ISBN 978-0-12-374225-4.
- [7] Valenta, M.; kolektiv: *IP 2015 DU 20 - Datová čistota - manažerské shrnutí*. 2016, dílčí výstupní zpráva projektu na ČVUT, [cit. 2017-02-15].
- [8] Kuznetsov, S.: *Datový sklad fakulty*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.
- [9] Valenta, M.; Holý, R.: *Datový sklad ČVUT a datová kvalita*. 2017, prezentace v rámci projektů na ČVUT, [cit. 2017-02-23].

- [10] Inmon, W. H.: *Building the Data Warehouse*. 3rd ed, New York: J. Wiley, 2002, ISBN 0-471-08130-2.
- [11] Kimball, R.: *The Data Warehouse Toolkit : Practical Techniques for Building Dimensional Data Warehouses*. 1st ed, New York: J. Wiley, 1996, ISBN 0-471-15337-0.
- [12] Rangarajan, S.: *Data Warehouse Design – Inmon versus Kimball [online]*. 2016, [cit. 2017-03-09]. Dostupné z: <http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>
- [13] Schiller, M.: *Co se skrývá pod zkratkou ETL? [online]*. 2003, [cit. 2017-03-09]. Dostupné z: <https://www.systemonline.cz/clanky/co-se-skryva-pod-zkratkou-etl.htm>
- [14] Výpočetní a informační centrum ČVUT: *V3S [online]*. 2015, [cit. 2017-02-24]. Dostupné z: <https://v3s.cvut.cz>
- [15] Výpočetní a informační centrum ČVUT: *USERMAP [online]*. 2011, [cit. 2017-02-24]. Dostupné z: <https://usermap.cvut.cz>
- [16] Jirovský, V.: *Konceptuální analýza datových domén Studium a Hodnocení kvality výuky s ohledem na datovou čistotu*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.
- [17] Moravec, L.: *Ontologická analýza datového skladu ČVUT*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [18] Hajčiar, D.: *Návrh a implementace backend projektu EBIE*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [19] AsciiDoctor: *AsciiDoctor.org [online]*. 2016, [cit. 2017-02-19]. Dostupné z: <http://asciidoctor.org>
- [20] Kehoe, D.: *What is Ruby on Rails? [online]*. 2013, [cit. 2017-02-19]. Dostupné z: <http://railsapps.github.io/what-is-ruby-rails.html>
- [21] Černý, C.: *Návrh a implementace integrace služeb do projektu EBIE*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [22] Jirůtka, J.: *Zuul OAAS [online]*. 2016, [cit. 2017-02-20]. Dostupné z: <https://github.com/cvut/zuul-oaas>
- [23] Kopecký, M.: *Návrh a implementace klientské části systému EBIE*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

-
- [24] Bostock, M.: *D3 Data-Driven Documents [online]*. 2015, [cit. 2017-02-21]. Dostupné z: <https://d3js.org/>
- [25] ManagementMania: *Ganttův diagram (Gantt Chart) [online]*. 2015, [cit. 2017-02-25]. Dostupné z: <https://managementmania.com/cs/ganttuv-diagram>
- [26] Ellingwood, J.: *Understanding the LDAP Protocol, Data Hierarchy, and Entry Components [online]*. 2015, [cit. 2017-02-28]. Dostupné z: <https://www.digitalocean.com/community/tutorials/understanding-the-ldap-protocol-data-hierarchy-and-entry-components>
- [27] Výpočetní a informační centrum ČVUT: *Shibboleth [online]*. [cit. 2017-02-28]. Dostupné z: <https://www.civ.cvut.cz/info/info.php?id=207>
- [28] ManagementMania: *Matice odpovědnosti RACI (RACI Responsibility Matrix) [online]*. 2016, [cit. 2017-03-01]. Dostupné z: <https://managementmania.com/cs/matice-odpovednosti-raci>
- [29] GitLab Inc.: *GitLab.com [online]*. 2012, [cit. 2017-03-05]. Dostupné z: <https://gitlab.com>
- [30] Chacon, S.; Straub, B.: *Pro Git: everything you need to know about the Git*. 2st ed, New York: Apress, 2014, ISBN 978-1484200773. Dostupné z: <https://progit2.s3.amazonaws.com/en/2016-03-22-f3531/progit-en.1084.pdf>
- [31] Hajčiar, D.: *Obsah EBIE [online]*. 2017, [cit. 2017-03-11]. Dostupné z: https://gitlab.fit.cvut.cz/hajcidav/ebie-content/blob/master_vyvoj/README.adoc
- [32] GitLab Inc.: *How to create a merge request [online]*. [cit. 2017-03-12]. Dostupné z: <https://docs.gitlab.com/ee/gitlab-basics/add-merge-request.html>
- [33] Valenta, M.; kolektiv: *Zpráva o stavu dat a obecná návrh metodiky*. 2016, dílčí výstupní zpráva projektu na ČVUT, [cit. 2017-03-19].
- [34] Brookshier, D.: *BPMN 2.0 Tutorial [online]*. 2011, [cit. 2017-04-14]. Dostupné z: http://www.omg.org/news/meetings/workshops/SOA-HC/presentations-2011/14_MT-2_Brookshier.pdf
- [35] Chinosi, M.: *Why to Consider BPMN 2.0? [online]*. 2009, [cit. 2017-04-14]. Dostupné z: <https://www.slideshare.net/mchinosi/why-to-consider-bpmn-20>
- [36] Vašíček, P.: *3. část: Úvod do BPMN [online]*. 2008, [cit. 2017-04-14]. Dostupné z: <http://bpm-sme.blogspot.cz/2008/03/3-uvod-do-bpmn.html>

- [37] OMG: *Business Process Model and Notation (BPMN) Version 2.0 [online]*. 2011, [cit. 2017-04-14]. Dostupné z: <http://www.omg.org/spec/BPMN/2.0/PDF/>
- [38] Camunda: *BPMN Modeling Reference [online]*. 2016, [cit. 2017-04-14]. Dostupné z: <https://camunda.org/bpmn/reference/>
- [39] OMG: *Core set of BPMN elements [online]*. 2008, [cit. 2017-04-14]. Dostupné z: https://www.researchgate.net/figure/229048697_fig3_Figure-6-Core-set-of-BPMN-elements-OMG-2008/
- [40] Enterprise Engineering Institute: *Enterprise Engineering And DEMO [online]*. [cit. 2017-04-16]. Dostupné z: <http://www.ee-institute.org/en/demo>
- [41] CCMi: Centrum pro konceptuální modelování a implementace: *DEMO [online]*. 2016, [cit. 2017-04-16]. Dostupné z: <https://ccmi.fit.cvut.cz/metodiky/demo/>
- [42] Mráz, O.: *Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [43] Naplava, P.; Pergl, R.: Empirical Study of Applying the DEMO Method for Improving BPMN Process Models in Academic Environment. In *2015 IEEE 17th Conference on Business Informatics*, ročník 2, Červenec 2015, s. 18–26, doi:10.1109/CBI.2015.12.
- [44] Dietz, J. L. G.: *Perinforma, A.P.C.: The essence of organisation*. 1st ed, Sapio Enterprise Engineering, 2015, ISBN 978-90-815449-4-8.
- [45] Dietz, J. L. G.: *Enterprise Ontology: theory and methodology*. 1st ed, Berlin; New York: Springer, 2006, ISBN 978-3-540-33149-0.
- [46] Radvan, M.: *Porovnání způsobu modelování podnikových procesů pomocí BPMN a DEMO z pohledu běžných uživatelů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.
- [47] Heller, .: *Možnosti využití metodiky DEMO pro tvorbu BPMN modelů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [48] Dietz, J. L. G.: *DEMO-3: Models and Representations [online]*. 2013, [cit. 2017-04-18]. Dostupné z: http://www1.osu.cz/~hunka/vyuka/javaOOP/poont/poont_08/DEM0%20Models&representations%2036c%20201303.pdf

Seznam použitých zkratk

EBIE Extended Business Intelligence Encyclopedia

DEMO Design & Engineering Methodology for Organizations

BPMN Business Process Model and Notation

ČVUT České vysoké učení technické

IP Institucionální plán

PES Portál ekonomických služeb

MIS Manažerský informační systém

KOS KOmponenta Studium

BI Business Intelligence

ETL Extract Transform Load

FIT Fakulta informačních technologií

V3S Výsledky vědy a výzkumu

OCD Organization Construction Diagram

PSD Process Structure Diagram

TPT Transaction Product Table

HTML HyperText Markup Language

IdM Identity Manager

OAAS OAuth 2.0 Authorization Server

REST Representational State Transfer

A. SEZNAM POUŽITÝCH ZKRATEK

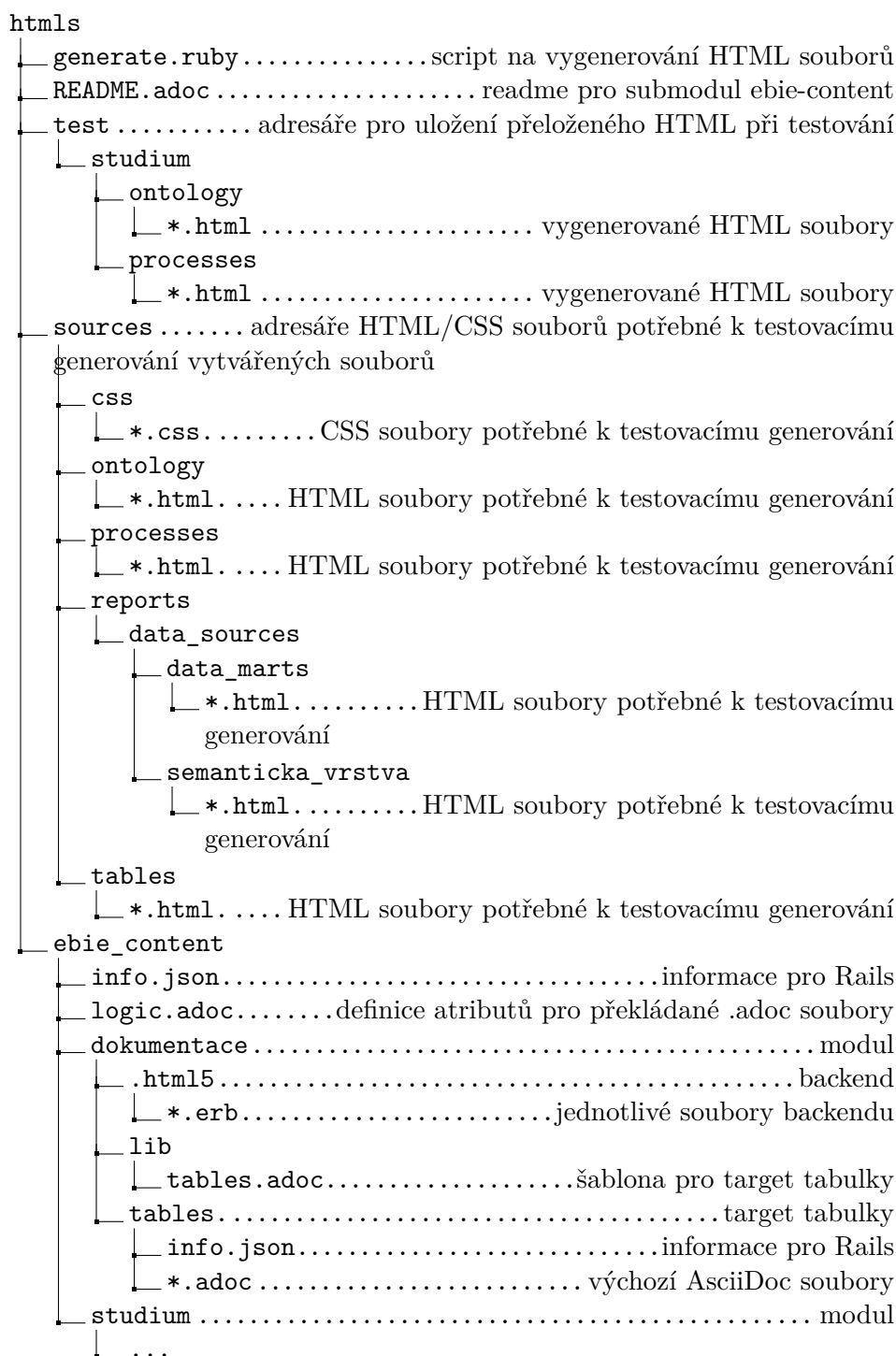
API Application Programming Interface
JSON JavaScript Object Notation
UI User Interface
LDAP Lightweight Directory Access Protocol
HTTP Hypertext Transfer Protocol
VIC Výpočetní a informační centrum
RACI Responsible, Accountable, Consulted, Informed
SQL Structured Query Language
SSL Secure Sockets Layer
TLS Transport Layer Security
SSH Secure Shell
VARCHAR Variable Character Field
Z, ZK, KZ Zápočet, Zkouška, Klasifikovaný zápočet
KZ Klasifikovaný zápočet
OMG Object Management Group
XOR Exclusive OR (exkluzivní disjunkce)
PSI Performance in Social Interaction
CM Construction Model
PM Process Model
BCT Bank Contens Table
TPD Transaction Pattern Diagram
FM Fact Model
OFD Object Fact Diagram
AM Action Model
IT Information Technology
AOM Actor Organization Matrix
VPN Virtual Private Network
CSS Cascading Style Sheets

Přílohy

B.1 Adresářová struktura portálu

Aktuální adresářová struktura portálu EBIE je zobrazena na snímcích B.1 a B.2. Tato struktura je znázorněna před generováním HTML souborů (spuštěním `generate.ruby`), které se vygenerují v adresářích s výchozími AsciiDoc soubory.

B. PŘÍLOHY



Obrázek B.1: Aktuální adresářová struktura s obsahem EBIE (1)

```

studium..... modul
├── .html5..... backend
│   └── *.erb.....jednotlivé soubory backendu
├── lib
│   ├── ontology.adoc.....šablona pro ontologický katalog
│   └── processes.adoc.....šablona pro procesy
├── ontology
│   ├── info.json.....informace pro Rails
│   ├── *.adoc.....výchozí AsciiDoc soubory
│   └── assets
│       └── img
│           └── images.....obrázky zobrazované v HTML souborech
├── processes
│   ├── info.json..... Informace pro Rails
│   ├── *.adoc.....výchozí AsciiDoc soubory
│   └── assets
│       └── img
│           └── images.....obrázky zobrazované v HTML souborech
├── reports
│   ├── info.json.....informace pro Rails
│   ├── data_sources
│   │   ├── info.json.....informace pro Rails
│   │   ├── assets
│   │   │   └── img
│   │   │       └── images.....obrázky zobrazované v HTML souborech
│   │   └── data_marts
│   │       ├── info.json.....informace pro Rails
│   │       └── *.adoc.....výchozí AsciiDoc soubory
│   │       └── lib
│   │           ├── data_marts.adoc..... šablona pro data marts
│   │           └── data_marts.adoc..... šablona pro sémantickou vrstvu
│   └── semanticka_vrstva
│       ├── info.json.....informace pro Rails
│       └── *.adoc.....výchozí AsciiDoc soubory
├── dirs.....složky s jednotlivými kategoriemi pro reporty
│   ├── info.json.....informace pro Rails
│   ├── dirs.....složky obsahující HTML soubory a assets
│   │   ├── *.html.....HTML soubory, které Rails renderují a zobrazí
│   │   ├── info.json.....informace pro Rails
│   │   └── assets
│   │       ├── css
│   │       │   └── .css..... stylování pro HTML
│   │       └── js
│   │           └── .js.....JavaScript pro HTML

```

Obrázek B.2: Aktuální adresářová struktura s obsahem EBIE (2)

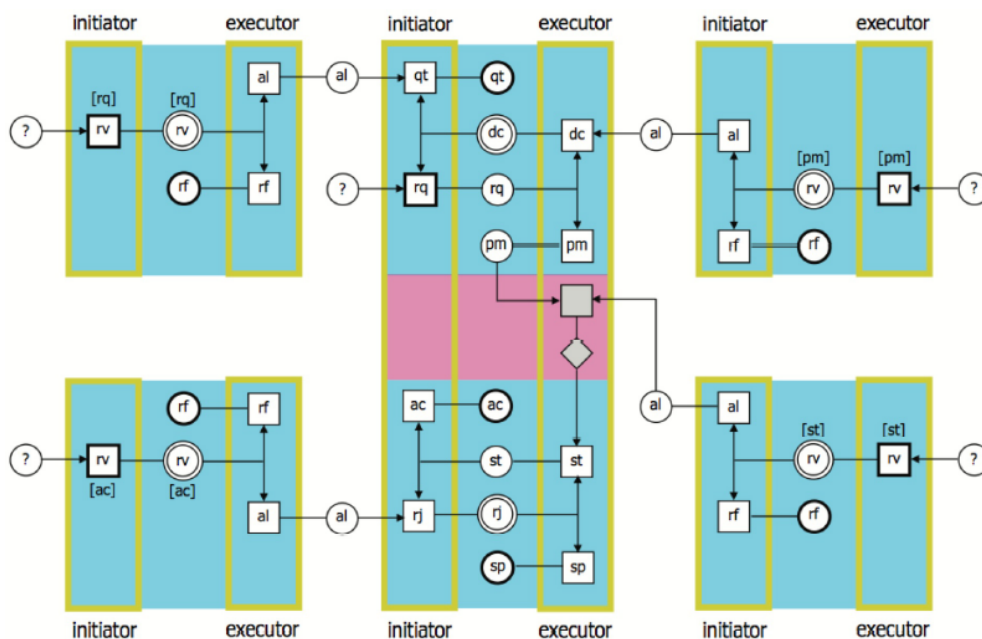
B.2 DEMO

B.2.1 Kompletní transakční vzor

Kompletní transakční vzor oproti standardnímu transakčnímu vzoru přidává tzv. odvolávací vzory (revoky). Tyto vzory musí být schváleny druhým aktorem. Jsou umožněny čtyři druhy revoke:

1. Revoke request
2. Revoke promise
3. Revoke state
4. Revoke accept

Kompletní transakční vzor je přikládán zde:



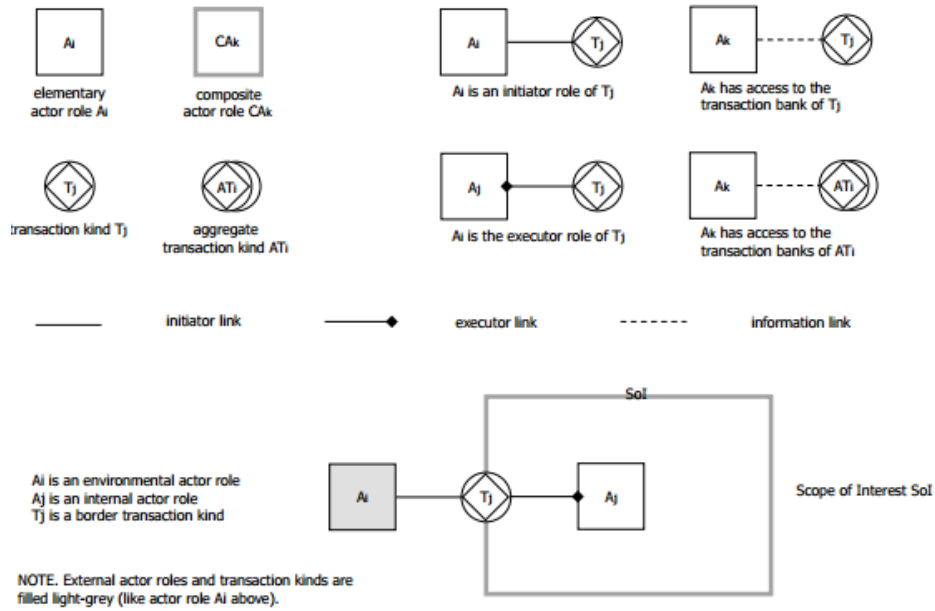
Obrázek B.3: Kompletní transakční vzor [44].

Na transakčním vzoru můžeme vidět zkratky aktů. U revoke najdeme tyto zkratky:

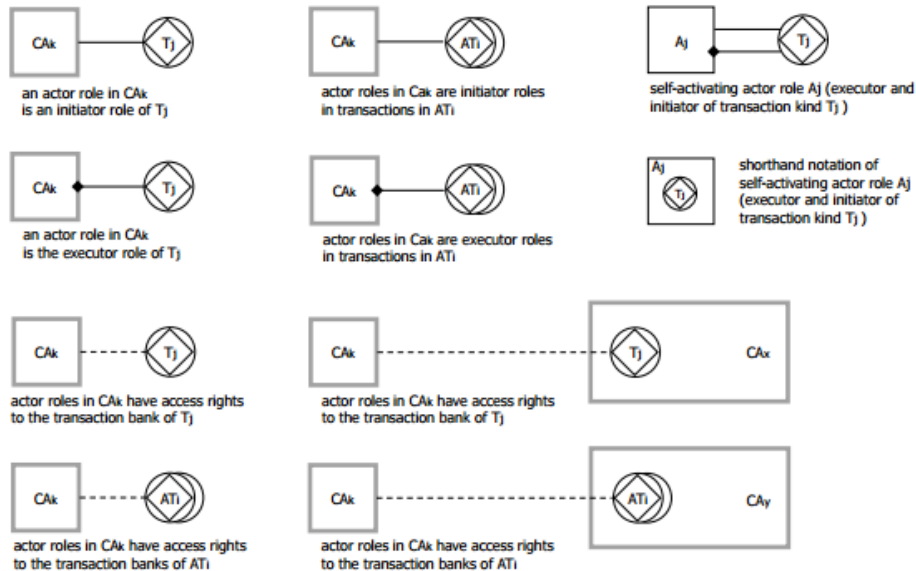
- al (allow, allowed) – Povolení revoke.
- rf (refuse, refused) – Zamítnutí revoke.

B.2.2 Elementy pro modelování OCD

Pro modelování OCD diagramů se používají následující elementy, které byly získány ze zdroje [48].

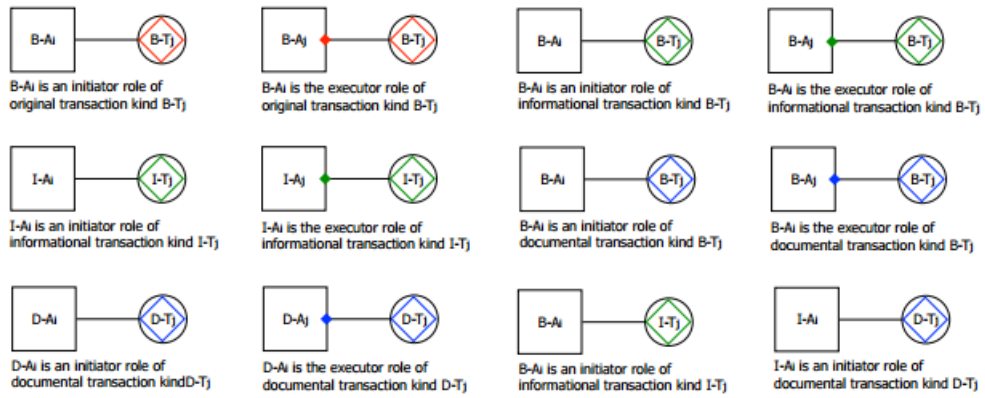


Obrázek B.4: Elementy OCD diagramu (1) [48].



Obrázek B.5: Elementy OCD diagramu (2) [48].

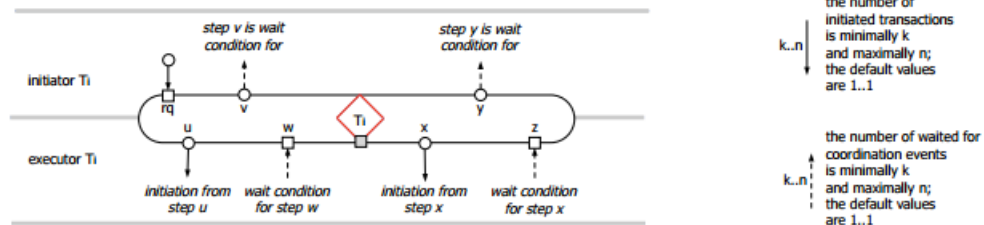
B. PŘÍLOHY



Obrázek B.6: Elementy OCD diagramu (3) [48].

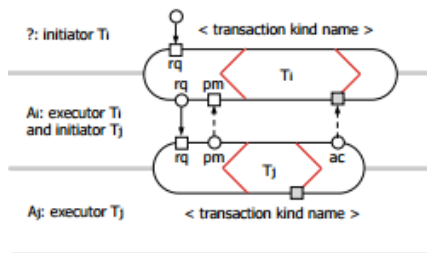
B.2.3 Elementy pro modelování PSD

Pro modelování PSD diagramů se používají následující elementy, které byly získány ze zdroje [48].

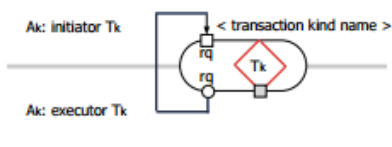


Obrázek B.7: Elementy PSD diagramu (1) [48].

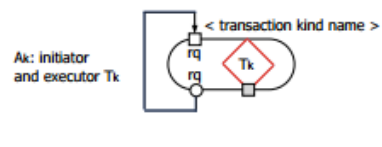
Example of transaction kind T_i with enclosed transaction kind T_j



Example of a self-initiating transaction kind T_k



Alternative notation:



Obrázek B.8: Elementy PSD diagramu (2) [48].

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
models	
BPMN.....	BPMN modely
img.....	obrázky modelů
*.png.....	diagramy procesů ve formátu PNG
*.svg.....	diagramy procesů ve formátu SVG
source.....	zdrojové soubory modelů
*.bpnm.....	diagramy procesů ve formátu BPMN
DEMO.....	DEMO modely
img.....	obrázky modelů
*.png.....	diagramy procesů ve formátu PNG
source.....	zdrojové soubory modelů
demo_models.xml.....	diagramy procesů v Modelworld XML formátu
DP_Batík_Ondřej.....	zdrojová forma práce ve formátu L ^A T _E X
text.....	text práce
DP_Batík_Ondřej.pdf.....	text práce ve formátu PDF
DP_Batík_Ondřej.ps.....	text práce ve formátu PS