



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název: Analýza a optimalizace proces životního pojištění
Student: Bc. Miroslav Šiagi
Vedoucí: Ing. Robert Pergl, Ph.D.
Studijní program: Informatika
Studijní obor: Webové a softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: do konce zimního semestru 2016/17

Pokyny pro vypracování

Cílem práce je provedení analýzy a optimalizace proces zajišťujících vznik a správu produktu životního pojištění nabízeného pojišťovnou.

Podrobnou analýzu proveďte metodou DEMO, zaměřte se na oddělení ontologické, infologické a datalogické úrovně. Po validaci modelu proveďte návrh optimalizace procesů a informační infrastruktury pro vybrané klienty (pojišťovny) firmy Trask Solutions. Porovnejte se stávajícím stavem a formulujte doporučení a závěry.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 14. února 2015

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



Diplomová práce

Analýza a optimalizace procesů životního pojištění

Bc. Miroslav Šiagi

Vedúci práce: Ing. Robert Pergl, Ph.D.

8. mája 2016

Pod'akovanie

Touto cestou chcem vyjadriť vďaku všetkým, ktorí mi počas obdobia písania tejto práce pomohli a ma podporili. Obzvlášť moja vďaka patrí môjmu vedúcemu práce Ing. Robertovi Perglovi, PhD. za pomoc pri realizácii tejto práce, za jeho rady a vedenie. Tiež sa chcem poďakovať kolegom a pracovníkom zúčastnených na procesoch životného poistenia za oboznámenie sa s ich prácou a postupmi. Nesmierna vďaka za podporu patrí i mojej rodine a známym.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. Ďalej prehlasujem, že som s ČVUT uzavrel dohodu, na základe ktorej sa ČVUT vzdalo práva na uzavrenie licenčnej zmluvy o používaní tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona. Táto skutočnosť nemá vplyv na ust. § 47b zákona č. 111/1998 Sb. o vysokých školách.

V Prahe 8. mája 2016

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2016 Miroslav Šiagi. Všetky práva vyhradené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Šiagi, Miroslav. *Analýza a optimalizace procesů životního pojištění*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Dôležitý aspekt procesov v organizáciách, ktorému sa práca venuje, je uvedenie si ich podstaty. Vzhľadom na veľké množstvo modelovacích prístupov, problémovosť vstupného riešenia BPM (*Business Process Management*) systémom, zložité a dynamické procesy vo firmách, sa vynára potreba vhodnej teórie. Nastolíme preto disciplínu *Enterprise Engineering* a metódu DEMO (*Design and Engineering Methodology for Organizations*) postavenú na jej teóriách. Cieľom práce je aplikovanie danej metódy na vybrané procesy životného poistenia, následne vyhodnotenie dôsledkov aplikácie, porovnanie prístupov metódy DEMO s BPM riešením a podanie návrhov. Vyhodnotenie ukázalo, že BPM riešenie nie je vhodné, čo vychádza z podstaty BPM a vo veľkej miere tomu prispieva komplexita procesov. Aplikovanie metódy DEMO nás naviedlo na správne vnímanie procesov spolu s účastníckymi rolami. Okrem iného metóda poukázala na možnosti implementácie. Vďaka jej prístupom bude ďalej možné spracovanie i zložitých funkcionalít.

Kľúčová slova BPM, *Enterprise Engineering*, DEMO, organizácia, proces

Abstract

The thesis deals with business process understanding problem. Due to large number of modeling approaches, problems in given BPM solution, complicated and dynamic business processes, the proper theory is required. Discipline Enterprise Engineering and DEMO offer a way to solve the problem. We focus on the application of DEMO to the given life insurance processes. We evaluate the results of the application, compare the application with the given solution and make propositions. Obtained results signify untowardness of BPM solution. This is mainly based on the nature of BPM and complexity of processes adds to this largely. The DEMO method application has led us to correct perception of processes along with the participant roles. Moreover, this method pointed at implementation possibilities. Thanks to its approaches, processing of more complicated functionalities will be possible.

Keywords BPM, Enterprise Engineering, DEMO, organization, process

Obsah

1	Úvod	1
1.1	Ciel práce	1
1.2	Metodika práce	2
2	Enterprise engineering a jeho teórie	5
2.1	Enterprise Engineering	5
2.2	Enterprise Ontology	8
2.3	ϕ teória	9
2.4	τ teória	13
2.5	ψ teória	15
3	DEMO	25
3.1	Organizačný teorém	25
3.2	Realizácia organizácie	26
3.3	Implementácia organizácie	28
4	Analýza procesov	31
4.1	Úvod k analýze procesov	31
4.2	Procesné modely	32
4.3	Transakčné tabuľky	35
4.4	Role naprieč procesmi	49
4.5	Transaction Product Table (TPT)	52
4.6	Construction model (CM)	54
4.7	Process Model (PM)	56
4.8	Fact Model (FM)	60
4.9	Action Model (AM)	63
4.10	Storno životného poistenia	66
5	Vyhodnotenia	67
5.1	Výhody metódy DEMO	67

5.2	Dôsledky aplikácie DEMa	68
5.3	Dôsledky BPM riešenia	70
	Záver	73
	Literatúra	75
	A Zoznam použitých skratiek	77
	B Syntax PSD	79
	C Obsah priloženého CD	81

Zoznam obrázkov

1.1	Identifikácia transakcií v analýze	2
2.1	Klasifikácia tried podnikového inžinierstva	7
2.2	Semiotický trojuholník (zdroj: [10])	10
2.3	Ontologický paralelogram (zdroj: [10])	11
2.4	Vzťahy medzi kategóriami systémov (zdroj: [7])	12
2.5	Generický proces tvorby systému (zdroj: [7])	13
2.6	Grafická reprezentácia úkonového axiómu (zdroj: [7])	16
2.7	Štandardný transakčný vzor a jeho skrátaná verzia (zdroj: [7])	18
2.8	Kompletný transakčný vzor (zdroj: [7])	19
2.9	Príklad vlozenej transakcie do inej transakcie (zdroj: [7])	20
2.10	Príklad transakčného samospúšťania (zdroj: [7])	20
2.11	Prehľad úrovňového axiómu (zdroj: [7])	22
2.12	Koordinácia a jej úrovne (zdroj: [7])	23
3.1	Reprezentácia organizačného teorému (zdroj: [7])	26
3.2	Integračné vrstvy organizácie (zdroj: [7])	27
4.1	Procesný BPM model procesu Vznik ŽP – pohľad z vrchu	33
4.2	Procesný BPM model procesu Vznik ŽP – fáza oceňovania	33
4.3	Procesný BPM model procesu Zmena ŽP – pohľad z vrchu	34
4.4	Procesný BPM model procesu Zmena ŽP – netechnická zmena	34
4.5	Procesný BPM model procesu Zmena ŽP – technická zmena	35
4.6	CM určenia typu vzniku ŽP	56
4.7	CM určenia typu vzniku ŽP	57
4.8	CM určovania typu zmeny a doplňovania atribútov	58
4.9	CM spracovania netechnickej zmeny	59
4.10	FM procesu vzniku ŽP	60
4.11	FM procesu zmena ŽP	62
5.1	Kompletný OCD procesu vzniku životného poistenia	70

5.2	Kompletný OCD procesu zmena životného poistenia	71
B.1	Syntax pre PSD, prvá časť (zdroj: [17])	79
B.2	Syntax pre PSD, druhá časť (zdroj: [17])	80

Zoznam tabuliek

4.1	Transakcia T01 – Proces Vzniku ŽP – hrubý pohľad	36
4.2	Transakcia T02 – Určenie typu zmluvy	36
4.3	Transakcia T03 – Overenie zmluvy vzniku životného poistenia . . .	37
4.4	Transakcia T04 – Podanie návrhu na storno	38
4.5	Transakcia T05 – Ocenenie zmluvy	39
4.6	Transakcia T06 – Oprava zmluvy	39
4.7	Transakcia T07 – Vykonanie rozhodnutia o úplnosti podkladov . .	40
4.8	Transakcia T08 – Doplnenie dokumentácie	40
4.9	Transakcia T09 – Rozhodnutie o výsledku podmieneného prijatia zmluvy	41
4.10	Transakcia T01_ZM – Zahájenie spracovania zmeny ŽP	42
4.11	Transakcia T02_ZM – Určenie typu zmeny ŽP	43
4.12	Transakcia T03_ZM – Doplnenie atribútov žiadosti	44
4.13	Transakcia T04_ZM – Spracovanie NTZ	44
4.14	Transakcia T05_ZM – Odoslanie informácie klienta o vykonaní zmeny	45
4.15	Transakcia T06_ZM – Určenie typu do ZIS	45
4.16	Transakcia T07_ZM – Zamietnutie TZ	46
4.17	Transakcia T08_ZM – Vytvorenie protinávrhu	47
4.18	Transakcia T09_ZM – Overenie zmeny	47
4.19	Transakcia T10_ZM – Ocenenie zmeny	48
4.20	Transakcia T11_ZM – Vytvorenie návrhu na protinávrh	48
4.21	Mapovanie DEMO rolí k firemným tímom	49
4.22	Mapovanie firemných tímov k rolám	51
4.23	Transaction Product Table	53
4.24	Action model - A05 - T03/rq	63
4.25	Action model - A06 - T03/pm	64
4.26	Action model - A06 - T03/st	64
4.27	Action model - A22 - T04/pm	65
4.28	Action model - A22 - T04/st	66

Úvod

Disciplíny zaoberajúce sa firemnými procesmi sú známe už veľa rokov a za posledné tri dekády upútali veľa pozornosti – z časti vďaka pokroku v informatike a komunikačných technológiách [1]. Informačné technológie hrajú významnú rolu v manažmente organizácií a hrajú rolu iniciátora pre mnoho procesných disciplín ako napríklad *Business Process Management* (BPM).

Vzniklo mnoho nových disciplín, ktoré presvedčajú o ich používaní, no značná časť z nich časom zlyhala, nakoľko predstavujú často už konkrétny nástroj alebo systém. V takých prípadoch, kedy sa na proces pozeráme napr. pomocou BPMN, už nemôžeme vidieť samotnú podstatu podniku a procesov.

Preto sme si položili otázku, či môže existovať vhodná teória, ktorá by popisovala skutočný charakter podnikov a procesov. Dôležitosť vhodnej teórie citoval už Albert Einstein: „To, či vec pozorovať môžeme alebo nie, závisí na teórii, ktorú používame. Je to práve teória, ktorá určuje, čo môže byť pozorovateľné.“

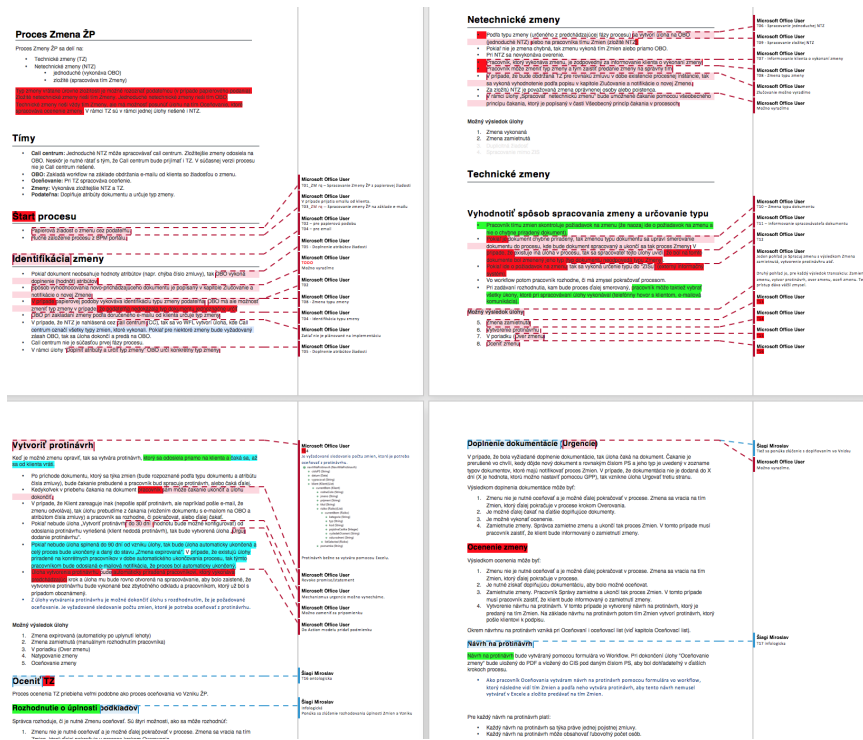
Odpoveďou by mohla byť práve disciplína *Enterprise Engineering* (EE), ktorú si v predstavíme v prvej časti práce. Spolu s ňou si predstavíme teórie, ktoré pod EE spadajú a následne sa presunieme ku konkrétnej metodike DEMO (*Design & Engineering Methodology for Organizations*). Metodika DEMO je založená práve na teóriách disciplíny EE.

1.1 Cieľ práce

Hlavným cieľom našej práce je vykonanie analýzy procesov životného poistenia a následné aplikovanie metódy DEMO. V analýze rozlíšime úrovne jednotlivých identifikovaných transakcií, určíme účastnícke role, namodelujeme konštrukčné diagramy a popíšeme pravidlá jednotlivých akcií pre jednotlivých účastníkov v rámci určitého úkonu.

Navrhnuté modely a dôsledky ich použitia porovnáme s aktuálnym riešením pomocou BPM systému. Hlavne sa zameriame na výhody získané apliká-

1. ÚVOD



Obr. 1.1: Identifikácia transakcií v analýze

ciou metódy DEMO a nedostatky BPM riešenia. Spolu s nimi podáme možné návrhy, čo by šlo zlepšiť alebo nahradiť.

1.2 Metodika práce

Na začiatku práce si predstavíme disciplínu EE a cez predstavenie teórií prejdeme ku popisu metodiky DEMO.

Na samotnú analýzu procesov sme použili analýzu vykonanú pre BPM riešenie spolu so získanými informáciami od zamestnancov, ktorí dané procesy vykonávajú. Načrtne si procesné *flow* jednotlivých procesov pomocou nástroja *draw.io* [2].

Vytvoríme dokument, ktorý popisuje základný priechod procesom i ďalšie požiadavky a podrobnejšie informácie. Zameriame sa najprv na základ procesu a jeho hlavných častí. Pri prechádzaní dokumentu budeme jednoducho ofarbovať vety príslušnou farbou, ktoré popisujú časti alebo celú transakciu 1.1.

Na základe identifikovaných transakcií, postupne vytvoríme transakčné tabuľky (používaný *framework* je $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$), pri ktorých určíme účastnícke role a ďalej sa zamyslíme nad i vo vstupnej analýze nepopisovanými úkonami a do-

plníme ich. Po vytvorení tabuliek vytvoríme zoznam transakcií s odpovedajúcimi produktmi. Vytvoríme mapovanie účastníckych rolí k existujúcim tímom a naopak.

Na základe zistenia procesného charakteru, transakcií a rolí vytvoríme konštrukčné diagramy pomocou nástroja UMLet [3] a skončíme pri vytvorení pravidiel jednotlivých úkonov pre účastnícke role.

Na záver práce spíšeme dôsledky použitia metódy DEMO najprv všeobecne a následne vyvodíme dôsledky aplikovania metódy na naše procesy. Súčasne porovnáme stávajúce riešenie, ktorému popíšeme výhody i nevýhody.

Enterprise engineering a jeho teórie

V tejto kapitole si predstavíme hlavné teórie, na ktorých sa DEMO zakladá. Najskôr si avšak predstavíme disciplínu *Enterprise engineering*, ktorá tieto teórie obsahuje a v rámci nej si popíšeme základné ciele a vlastnosti.

Ďalej si predstavíme *Enterprise ontology*, ktorá pokrýva niektoré ciele *Enterprise engineeringu* a ktorá bude pre nás jedným z odrazových mostíkov

Základné teórie, na ktorých je DEMO postavené, nesú názvy písmen gréckej abecedy – ϕ , τ a ψ . Tieto teórie predstavujú základy využívajúce v *Enterprise engineering*. Začal ich predstavovať p. Dietz a to v 90. rokoch minulého storočia ([4], [5], [6] a [7]). V rámci ψ teórie si predstavíme základné axiómy, ktoré sú silným základom pre DEMO.

2.1 Enterprise Engineering

Podnik je zámerne vytvorené združenie ľudí, ktoré má určitý spoločný sociálny cieľ. Tvorba takého podniku vyžaduje adekvátny dizajn. Pod týmto pojmom často vidíme rádobý mechanický prístup k tvorbe dizajnu a môžeme vidieť spojitost s vytváraním dizajnu nejakého stroja. Pre tento mechanický pohľad na organizáciu a manažment je niekedy použitý pojem sociálne inžinierstvo (*social engineering*) [8]. Sociálne inžinierstvo prirovnáva manažovanie podniku s ovládaním stroja smerom zhora nadol a z tohto hľadiska je podnik vnímaný ako objektívna entita, pre manažment nezávislá komponenta, ktorá takmer vôbec nepotrebuje byť kontrolovaná [6].

Časom a množstvom výskumov toho, ako by sa taký dizajn mal vykonávať, sa zistilo, že dizajn je veľmi dôležitým prvkom majúci vplyv na:

1. efektívnosť, účinnosť podniku,
2. efektívne zaistenie kvality, služieb, zákazníckej orientácie a

3. strategické smerovanie.

Môžeme zdefinovať, že podnikové inžinierstvo je vedecká disciplína, v ktorej sú podniky považované za navrhnuté systémy (*designed systems*), ktorých dizajn sa môže meniť a redefinovať.

Predstavme si nasledujúce všeobecné ciele podnikového inžinierstva :

2.1.1 Bystrá ovládateľnosť

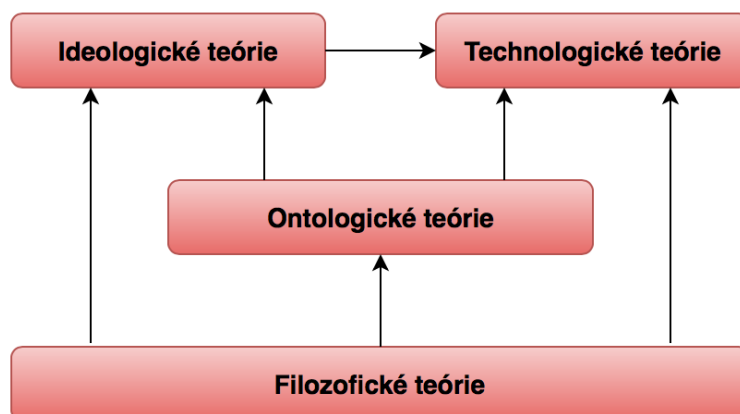
Na to, aby sme získali, udržali si prehľad podniku a zmien v rámci neho, a zvládli jeho komplexitu, potrebujeme vhodné teórie o konštrukcii a aktivitách podniku. Podniky, ktoré nie sú komprehenzívne chápané (pokrývajú všetky problémy a otázky spolu s faktami), nemôžu byť adekvátne riešené. Preto charakter prípadných potrebných zmien by nemohol byť určený jednoznačne a v dôsledku toho by zmeny nemohli byť aplikované efektívne. Do toho v dnešnú dobu, kedy je ICT už takmer neoddeliteľnou súčasťou podnikov, by bola ich implementácia pomalá a prakticky neovládateľná. V konečnom dôsledku na to, aby sme mohli dané zmeny aplikovať rýchlo a mať ich pod kontrolou, potrebujeme spôsoby, ktoré by implikovali dobre navrhnutú systematickú redukciu komplexity podniku [6]. Týmto cieľom (bystrá ovládateľnosť – anglicky *intellectual manageability*) sa zaoberá tzv. *enterprise ontology*, nazveme ju podniková ontológia [9].

2.1.2 Organizačná súhra

Na to, aby sme zmeny implementovali optimálne a úspešne, podniky musia fungovať ako zjednotený a integrovaný celok a to zo všetkých relevantných aspektov. Mnoho prístupov vývoja podnikov obsahuje slabiny a nekompletnosť z hľadiska teórie a metodológie prístupov. Evidentne, nie je dostatočné považovať procesy, informácie k nim, software a ich infraštruktúru ako domény, na ktorých by sme mali založiť dizajn podniku. Potrebujeme životaschopnú teóriu a metodológiu, ktorá je schopná riešiť všetky relevantné aspekty – a to aj tie, ktoré v súčasnosti nemôžu byť predpovedané. Potrebujeme ich riešiť správnym uceleným spôsobom, aby funkčná stránka podniku bola vždy koherentná a konzistentná. Jej časti tak musia byť adekvátne a harmonicky usporiadané – organizačná súhra (anglicky *organisational concinnity*) je práve jedným z cieľom podnikového inžinierstva [6] a týmto cieľom sa zaoberá podniková architektúra (anglicky *enterprise architecture*).

2.1.3 Sociálna oddanosť

V podnikovom inžinierstve musíme zdôrazniť dôležitosť zapojenia zamestnancov a ostatných účastníkov, ktorí sa podieľajú na vytváraní podnikových produktov, majú vplyv na kvalitu služieb a produktov, na kvalitu orientácie smerom k zákazníkovi, na kvalitu výuky a inovácií, na zvládanie podnikovej dy-



Obr. 2.1: Klasifikácia tried podnikového inžinierstva

namiky, komplexite spojenjej s neistotou vedúcej k (naliehavému) vývoju podniku. Oproti mechanistickému pohľadu na organizácie, podnikové inžinierstvo považuje ľudí za základnú entitu a centrum diania. Preto by mali mať zamestnanci plnú moc a kompetenciu v tom, čo vykonávajú. To musí byť schválené transparentnou autoritou a zamestnanci musia mať tiež prístup ku všetkým informáciám, ktoré potrebujú na zodpovedné vykonávanie svojich úloh [6].

2.1.4 Teórie podnikového inžinierstva

Pre disciplínu existuje množstvo teórií, ktoré sú fundamentálne pre podnikové inžinierstvo ako disciplínu. Mnoho z nich je vyvinutých už veľmi dobre, niektoré potrebujú ďalší vývoj a vylepšenie. Rozlišujeme štyri základné teórie [6]:

filozofické,
ontologické,
technologické a
ideologické.

Tieto štyri teórie sú navzájom prepojené a podľa obrázku 2.1 zobrazujúceho vzťahy, kde šípka medzi jednotlivými triedami znamená, že každá teória v danej triede na strane šípky je založená na teóriách triedy na druhej strane.

Filozofické teórie popisujú úplne základné koncepcie. Zahrňujú filozofické odvetvia ako epistemológiu, fenomenológiu, logiku a matematiku. Tieto teórie sú oceňované vďaka ich pravdivosti v daných oblastiach, ktorá je založená na argumentácii a posudzovaní reality. Čo sa týka teórií týkajúcich sa logiky a matematiky, tak tu prevláda exaktnosť, pričom u iných teóriách to v takej miere nie je. Do triedy filozofických teórií patrí ϕ , δ a τ teória.

Ontologické teórie sa zaoberajú povahou vecí, vysvetľovaním alebo predpovedaním vzťahov medzi nimi na základe pozorovaných javov. V rámci disciplíny podnikového inžinierstva sa zameriavajú obzvlášť na dôsledky vzťahov v systémoch. Tieto vzťahy vedia vysvetliť pozorované správanie a tiež vedia do istej miery predpovedať správanie – to je celé založené na ontologickom chápaní, ktoré ontologické teórie popisujú. Ontologické teórie sú cenené vďaka ich platnosti a vhodnosti. Platnosť je zdedená práve z filozofických teórií a vhodnosť je zavedená vyhodnocovaním praktickým využitím. Do triedy ontologických teórií patrí ψ a π teória.

Technologické teórie sú teórie, ktoré sa zaoberajú koncovými vzťahmi medzi javmi, čo predstavuje jadro všetkých druhov inžinierstva. Technologické teórie sú základom pre navrhovacie metódy. Metóda, ktorý je pevne zakomponovaná v technologickej teórii, sa často nazýva metodológia. Navrhovací proces je v podstate proces analyzovanie nejakého problému, po ktorom nasleduje vytvorenie nejakého riešenia. Akonáhle máme riešenie pokrývajúce všetky potrebné detaily, môžeme ho implementovať. Implementácia v rámci technologických teórií znamená priradenie konkrétneho významu elementom implementačného modelu. Technologické teórie sú oceňované ich presnosťou a relevantnosťou, kde presnosť je zakorenená v ontologických teóriách a relevantnosť je zavedená vyhodnotením ich praktickým využitím. Do triedy technologických teórií patrí σ teória.

Ideologické teórie sa zaoberajú cieľmi, ktoré ľudia smú kedykoľvek v spoločnosti dosiahnuť a najmä ktoré môžu dosiahnuť v podnikoch. Ideologické teórie sú poháňané víziami, presvedčeniami a vierou. Preto na rozdiel od objektívnych ontologických a technologických teórií zohľadňujú podstatu vecí – také aké sú a smú byť. Rola ideologických teórií pri vývoji podniku je viesť plánovanie a vyberať potrebné zmeny. Ideologické teórie nemôžu byť z podstaty považované za pravdivé, platné, presné ani relevantné aj napriek tomu, že sú zakorenené vo vyššie spomínaných teóriách. Môžeme hovoriť len o ich spoločenskom význame. Do triedy ideologických teórií patrí β a ν teória.

2.2 Enterprise Ontology

Jeden z cieľov podnikového inžinierstva – bystrá ovládateľnosť (*intellectual manageability*) – je krytým podnikovou ontológiou. Začnime uvedením, že manažovanie podniku, využívanie podnikových služieb ako klient alebo spolupráca s podnikom ako partner je čím ďalej zložitejšia a komplexnejšia. Súčasný problémy v podnikoch bývajú dobre vyskúmané a zdokumentované, avšak na ich vyriešenie už nebýva vynaložená taká snaha. Spoločný znak problémov vo firme býva komplexita, ktorá môže byť zvládnutá len za splnenia dvoch podmienok. Prvá podmienka je, že potrebujeme mať k dispozícii kompletnú teóriu o veciach, ktorých komplexitu chceme zvládnuť. Druhá podmienka je, že potrebujeme mať k dispozícii vhodné analytické metódy a techniky, ktoré

by na spomínanej teórii boli založené [7].

Na to, aby sme zvládli súčasné a budúce výzvy, potrebujeme konceptuálny model, ktorý spĺňa tzv. $4C - 4$ kritéria a to:

koherentnosť,
komprehenzívnosť,
konzistentnosť a
koncíznosť.

Koherentným modelom máme na mysli, že aspekt modelov ktorý tvorí logický a pravdivý celok. Komprehenzívnym modelom máme na mysli, že pokrývame všetky relevantné problémy a otázky. Konzistentným máme na mysli, že pohľad na model je bez rozporov alebo nezrovnalostí. Koncízny máme na mysli, že model neobsahuje žiadne prebytočné fakty a náležitosti a že celý model je kompaktný a stručný. Najdôležitejšia vlastnosť je avšak tá, aby tento konceptuálny model bol esenciálny – aby predstavoval len esenciu podniku. Esenciu podniku oddeľujeme od akejkoľvek formy realizácie alebo implementácie [7].

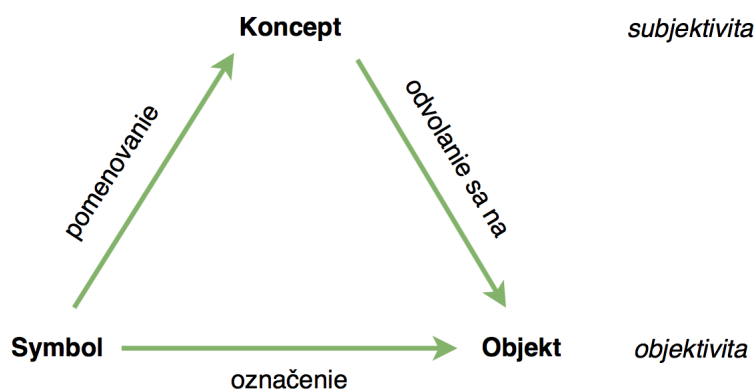
2.3 ϕ teória

ϕ teória sa zaoberá tvorbe koncepcií z faktických znalostí. Vyslovenie znaku ϕ – „fi“ – znamená pre našu doménu záujmu iniciály pre *Fact and Information*. Teória je zakorenená v semiotike, v ontológii a v logike. V rámci disciplíny semiotika si predstavíme semiotický trojuholník 2.2. Ukazuje, ako ľudia používajú symboly ako reprezentáciu objektov, aby o objektoch mohli komunikovať bez toho, aby boli objekty prítomné [10].

Základné pojmy v trojuholníku sú znak, objekt a koncept. Myšlienka (koncept) sa nachádza v mysli subjektu (ľudskej bytosti), pojem alebo symbol (znak) určuje koncept a skutočná vec (objekt) je objekt odvolávajúci sa na koncept a označený znakom.

Prvá vec, ktorú by sme mali ešte zmieniť je, že pod objektom by sme mali mať na mysli identifikovateľnú vec. Je to napríklad nejaká osoba, auto, obchodná dohoda, nájomná zmluva atď. Hoci objekt nemusí byť vždy niečo, čo môžeme vidieť (napr. dohodu), môže byť prítomný a definovaný ako vec v objektívnom svete – a môže byť objavený pomocou vlastností daného objektu.

Odhliadnuc od prítomnosti objektov v objektívnom svete, máme ich obraz, ktorý existuje v mysli pozorovateľa. Väčšina komunikácii medzi ľuďmi je o spoločných obrazoch nejakých objektov. Obraz objektu v mysli pozorovateľa nazveme koncept (ako koncept v semiotickom trojuholníku). Tento koncept je vec existujúca v subjektívnom svete. Tu vzniká otázka ako komunikovať s ostatnými o koncepte, ktorý je časťou subjektívneho sveta. Odpoveď je taká, že



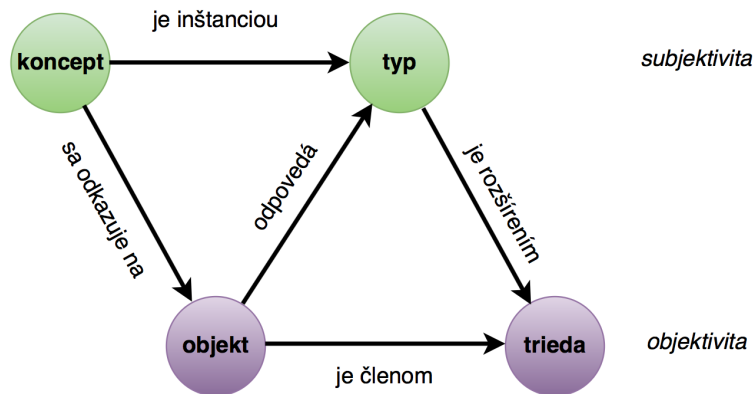
Obr. 2.2: Semiotický trojuholník (zdroj: [10])

Ľudia majú tendenciu mať v mysli vytvorený koncept, ktorý reflektuje koncept toho druhého, pričom ide o ten istý objekt.

Avšak korektná komunikácia medzi ľuďmi je nemožná, pokiaľ nie sú použité žiadne znaky. Znakmi máme na mysli zvuky, slová, diagramy, kresby a ďalšie symboly, ktoré musia byť zrozumiteľné a byť formou nejakého konceptu. Forma konceptu musí byť oddeliteľná od jeho fyzickej podoby, ktorá sama o sebe nie je tak podstatná. Nie je to ani časť znaku, je iba jeho nositeľom. Napríklad, znak „článok“ je jedným znakom a nezáleží, či je článok napísaný pomocou *Microsoft Office Word* alebo perom na papier. To by sme rozprávali už o jeho podobe.

Ďalej vzniká otázka ako to vyzerá so vzťahom medzi objektom a informáciou. Informácia sa odkazuje na objekt a môže byť vnímaná ako neoddeliteľná dvojica znak – koncept, ktorú pomenováva. Znak je nazývaný forma informácie a koncept je nazývaný obsahom informácie. Neskôr môžeme v tejto práci použiť výraz dokument, pod ktorým máme na mysli už formu nejakej informácie o veci.

Ďalšia vec týkajúca sa objektu je, že pojem objekt je veľmi široko definovaný. Pokrýva nielen fyzické objekty (stromy, stroje či hodiny) ale aj abstraktné objekty (projekt, pôžička alebo svadba). V tomto kontexte potrebujeme zaujať filozofický postoj. Začnime tým, že si predstavíme tri typy ľudí - objektivistov, subjektivistov a konštruktivistov. Objektivisti veria, že svet, v ktorom žijú, existuje sám o sebe a je od nich plne nezávislý. Prosto veria v pravdivú objektívnu realitu. Subjektivisti naopak veria, že neexistuje žiadna realita mimo subjekt (ľudí) a v extrémnom prípade veria, že každý subjekt má svoju vlastnú predstavu o realite. Niekde medzi sa nachádzajú konštruktivisti, ktorí súhlasia so subjektivistami v tom, že neexistuje žiadna absolútne objektívna realita, ale tiež veria, že existuje nejaká semi-objektívna realita, ktorú nazývajú intersubjektívna realita. Taká realita je vytvorená a kontinuálne prispôbovaná pomocou vyjednávania a vznikom súhlasov medzi



Obr. 2.3: Ontologický paralelogram (zdroj: [10])

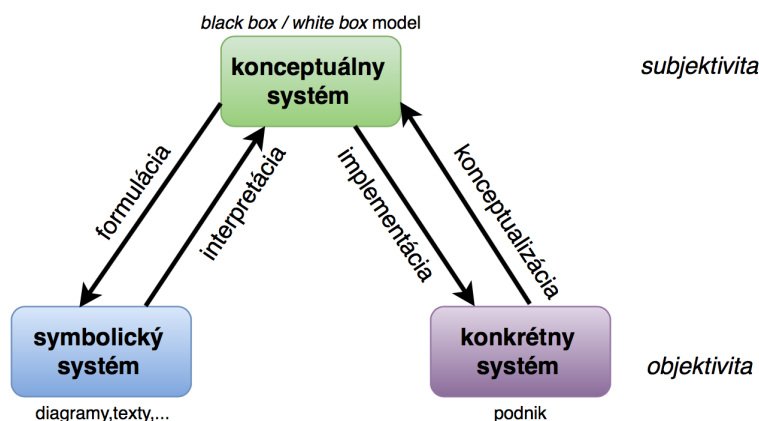
subjektami [7]. Príkladom môže byť leasingová zmluva, ktorá je abstraktný objekt, ktorý je výsledkom vyjednávania medzi dvoma subjektami. Dohoda existuje v momente, kedy ju obe strany považujú za dohodu. V našej doméne využívania podnikového inžinierstva budeme v tejto práci zaujímať pohľad konštruktivistov.

Semiotický trojuholník nie je aplikovateľný na jednotlivé koncepty ale skôr na generické koncepty. Jan Dietz definuje generický koncept ako typ a generický objekt ako triedu. Trieda je rozšírením (*extension*) typu a typ je vyjadrenie obsahu triedy (*intension*). Pojem typ môže byť najlepšie chápaný ako predpis formy, kde forma je nejaká kolekcia relevantných vlastností. Potom jeho prislúchajúci koncept nazývame inšinciou typu. Objekt bude členom triedy, ktorá je rozšírením typu, ktorému sa podriaďuje.

Predstavme si nasledujúci príklad pre proces uzavretia nájomnej zmluvy. Existuje trieda ukončených dôhod o nájme, ktoré sú semi-objektívne pozorovateľné pre každého. Tieto dohody o nájme majú spoločnú množinu relevantných vlastností. Vyjadrenie obsahu (*intension*) triedy dohôd o nájme je typu *nájomná zmluva*. Semi-objektívne pozorovateľná dohoda o nájme je vyjadrená na ontologickom paralelograme 2.3 ako objekt a odpovedá typu *nájomná zmluva*. Konkrétna nájomná zmluva by ďalej znamenala koncept a teda by bola inšinciou typu *nájomná zmluva*.

Reprezentácia konceptu označená znakom nie je na ontologickom paralelograme zobrazená, nakoľko znak nie je relevantný z ontologického hľadiska. Ontológia je totiž o esencii vecí a nie o tom, ako ich voláme. Ontologický paralelogram bude pre nás základňou pre vytvorenie ontológie sveta. V akýkoľvek moment sa svet nachádza v konkrétnom stave a je definovaný množinou objektov. Pokiaľ tento stav prevláda, objekty sú zachované. Svet podnikov môžeme chápať ako príklad spomínaného sveta.

Presuňme sa ďalej na úroveň podniku a chodu organizácií. Definujme si prechod, resp. premenu ako zmenu stavu sveta v konkrétnom čase. Jan Dietz

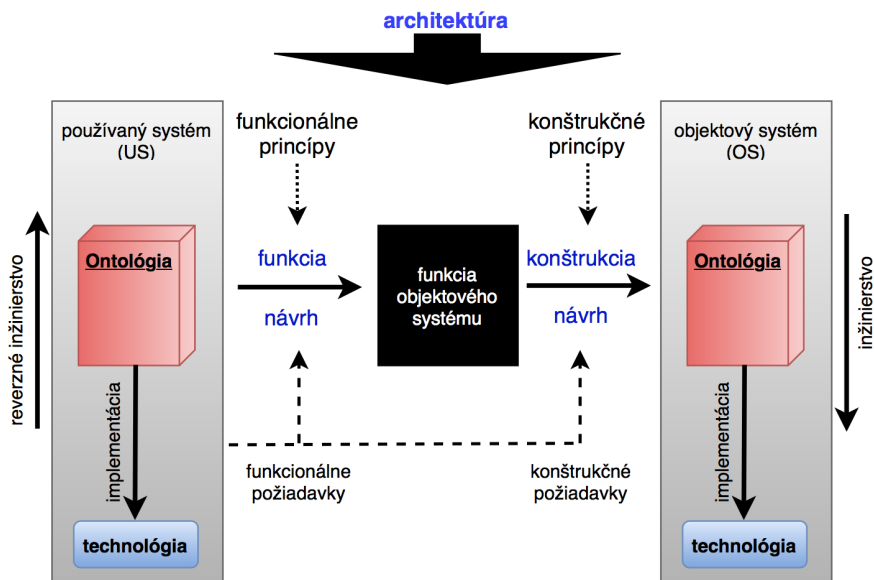


Obr. 2.4: Vzťahy medzi kategóriami systémov (zdroj: [7])

[7] v tomto kontexte hovorí, že stav sveta objektov je reflektovaný efektmi produkčných aktov, ktoré sú vykonávané systémovými elementmi. Jan Dietz považuje organizáciu za systém, ktorý je charakteristický určitou produkciou.

Organizácie sú komplexné a všeobecný spôsob ako skúmať komplexný systém je postaviť model daného systému. Pre našu definíciu modelu sa odkazujeme na Apostela [11]: akýkoľvek subjekt používajúci systém A, ktorý ani priamo či nepriamo interaguje so systémom B, na získanie informácie o systéme B použije A ako model pre B. Podľa tejto definície je systém považovaný za model, ak je systém použitý ako model. Existujú tri rôzne kategórie systémov, ktoré budeme rozlišovať: konkrétny systém, symbolický systém a konceptuálny systém. Vzťahy medzi týmito kategóriami popisujeme na obrázku 2.4. Konceptuálny systém je považovaný za konceptuálny model konkrétneho systému prostredníctvom konceptualizácie. Obrátene, konkrétny systém je považovaný za konkrétny model konceptuálneho modelu prostredníctvom implementácie konceptuálneho systému. Symbolický systém považujeme za symbolický model konceptuálneho systému po tom, čo vyformulujeme konceptuálny systém.

Jan Dietz opisuje dva existujúce rôzne typy konceptuálnych modelov nejakého konkrétneho systému: *black box* a *white box* model. *Black box* model definujeme ako konceptuálny model funkcie nejakého konkrétneho systému. *White box* model je zas definovaný ako konceptuálny model konštrukcie konkrétneho systému. *White box* model je vhodný za účelom tvorby alebo zmeny nejakého konkrétneho systému a preto je dominantným typom modelu vo všetkých inžinierskych odvetviach.



Obr. 2.5: Generický proces tvorby systému (zdroj: [7])

2.4 τ teória

Vyslovenie τ teórie ako „tao“ predstavuje inicály troch slov: technológia, architektúra a ontológia. Je to teória o vývoji (tvorbe) artefaktov. Keďže je podnik považovaný tiež za artefakt, τ teóriu by sme mali vnímať ako teóriu zaoberajúcu sa tvorbe organizácií. Podobne ako pri tvorbe organizácii, budeme do nej zahrňovať aj tvorbu systémov – pre prípad nového systému alebo zavádzania zmeny existujúceho systému. Proces tvorby systému pokrýva všetky aktivity, ktoré majú byť vykonané, aby bol systém kompletne implementovaný. Predstavíme si podstatu tohto procesu, ktorý nás prinesie ku koncepcii generického modelu procesu tvorby systému 2.5. Akýkoľvek proces tvorby zahŕňa dva systémy, ktoré nazývame používaný systém a objektový systém. Pod objektovým (OS) systémom rozumieme systém, ktorý bude vytvorený a pod používaným systémom (US) rozumieme systém, ktorý bude využívať služby (funkcionality) objektového systému. Ako exemplárny príklad používaného systému uvažujme oddelenie predaja nejakého podniku. Jedna zo služieb, ktorú používaný systém využíva z objektového systému je vyrátanie mesačného obratu podniku. Pre príklad objektového systému si vezmeme napríklad informačný systém obsahujúci informácie z predajov a nákupov. Jedna zo služieb, ktorú produkuje, je súčet množstva numerických hodnôt. Tento súčet je interpretovaný používaným systémom ako mesačný obrat [10].

Kompletný proces tvorby systému rozdelíme na tri hlavné sub-procesy, resp. fázy: návrh, inžinierstvo a implementáciu.

Najprv si rozoberieme fázu návrhu. Rozlišujeme dve hlavné aktivity pri di-

zajne objektového systému. Prvou je funkčný návrh a druhou je konštrukčný návrh. Funkčný návrh začína na konštrukcii používaného modelu a končí pri funkcii objektového modelu. Toto znamená, že špecifická funkcia objektového systému neobsahuje žiadnu informáciu o konštrukcii objektového systému. A teda funkcia objektového systému musí byť plne špecifikovaná len z hľadiska konštrukcii používaného systému (US). Konštrukčný návrh je dosť rozličný od funkčného návrhu. Ten začína so špecifikáciou funkcie objektového systému a končí konštrukciou objektového systému. Konštrukčný návrhári musia prepojiť mentálnu medzeru medzi funkciou a konštrukciou. Je to tým pádom len o prepojení medzi systémami rôznych druhov: medzi druhom používaného systému a druhom objektového systému. Výsledok aktivity funkčného návrhu je funkcionálny model objektového modelu, ktorý z podstaty predstavuje *black box* model. Zahrňuje všetky funkcionálne špecifikácie pre objektový systém. Tie zahrňujú špecifikácie pre rozhrania, ktoré sú službami objektového systému poskytované a nadobudnuté používaným systémom (US).

Hlavný vstup pre aktivitu funkčného návrhu je množina funkcionálnych *požiadavkov* získaných používaným systémom (tejto práci budeme nazývať žiadosť požiadavkom). Tieto nemusia byť podobné funkcionálnym špecifikáciám obsiahnutým funkcionálnym modelom objektového systému a to z niekoľkých dôvodov. Napríklad kvôli tomu, že požiadavky môžu byť neopodstatnené. Ďalší dôvod môže byť, že implementácia funkcionálnych a konštrukčných špecifikácií musí byť uskutočniteľná – pomocou dostupných technológií a na základe daného rozpočtu.

Celý návrh systému môžeme chápať ako proces meniacej sa analýzy a vzniknutých krokov. Jeden z analytických krokov je taký, kde danému problému začíname chápať lepšie. Vzniknutý krok chápeme ako krok bližší k jasnejšiemu riešeniu. Podľa Christophera Alexandra [12] v konečnom dôsledku proces návrhu nie je jedným (veľkým) funkčným krokom nasledovaným jedným veľkým konštrukčným krokom v návrhu, ale je to sekvencia menších meniacich sa analytických a vzniknutých krokov. Výsledok analytického kroku je lepšie chápanie požiadavkov používaného systému (zahrňujúci funkcionálne princípy). Tento výsledok tak vylepšuje predošlé výsledky rozšírením alebo zlepšením *black box* modelu objektového systému. Výsledok vzniknutého kroku je dosiahnutie lepšieho pochopenia, ako by mal byť systém vytvorený. Tento výsledok zas rozširuje alebo zlepšuje špecifikácie týkajúce sa konštrukcie a operácií objektového systému.

Vo všeobecnosti návrh teda obsahuje vyjednávanie za účelom, aby bol výsledok vyrovnaným kompromisom medzi (rozumnými) požiadavkami a (uskutočniteľnými) špecifikáciami. Toto sa podobne týka i fázy konštrukčného návrhu. Konštrukčné požiadavky, ktoré sú poskytované používaným systémom, musia byť chápané ako hlavný vstup pre túto fázu. Zvyknú sa nazývať nefunkcionálnymi požiadavkami. Konštrukčné požiadavky berú do úvahy výkonové charakteristiky, s ktorými objektový systém bude počítat pri poskytovaní jeho služieb.

V dnešnú dobu je dôležitá ontológia aj technológia a ontológia resp. ontologický model nejakého systému je modelom jeho konštrukcie, ktorý je úplne nezávislý od spôsobu, ktorým je implementovaný. Z definície ontologický model predstavuje konštrukčný model systému z najširšieho uhlu pohľadu. Systémové inžinierstvo je aktivita, v ktorej sú produkované série konštrukčných modelov. Každý model je plne založený na predošlých a dostupných špecifikáciách. Na rozdiel od navrhovania je inžinierstvo skôr záležitosťou remeselníctva než kreativity. Inžinierstvo sa odráža od ontologického modelu a končí pri implementačnom modeli. To je dôvod, prečo je inžinierstvo nazývané tiež implementácia návrhu. Implementáciou návrhu chápeme priradenie technologických prostriedkov do konštrukčných prvkov v implementačnom modele. Akonáhle je návrh implementovaný korektne, systém môže byť uvedený do prevádzky.

2.5 ψ teória

Teória ψ – tiež skratka pre *Performance in Social Interaction* pozostáva z dvoch častí: zo všeobecnej ψ teórie a zo špeciálnej ψ teórie.

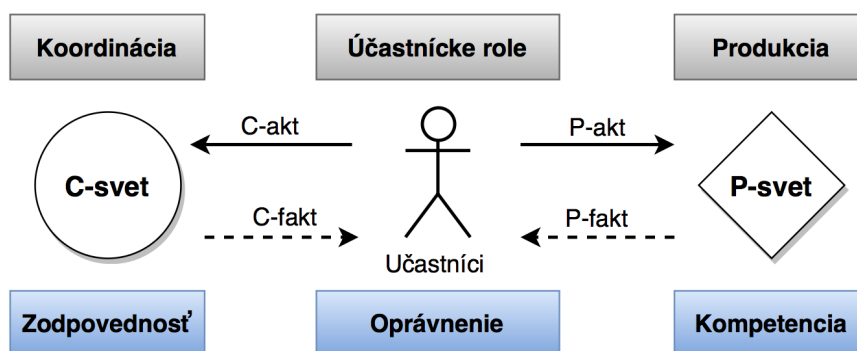
Všeobecná ψ teória je teória o ľudskej spolupráci. Tiež je „ľudskou tvárou“ alebo vonkajšou vrstvou ψ teórie. Je založená na teórii komunikačnej akcii (*Theory of Communicative Action*) a teórii rečových aktov (*Speech Act Theory*).

Špeciálna ψ teória objasňuje dôsledky všeobecnej ψ teórie systémomv využívanými organizáciami. Preto je tiež nazývaná „systémovou tvárou“ alebo akýmsi vnútrom ψ teórie. Je založená na systémovej ontologickej teórii (*Systemic Ontology Theory*) a na δ teórii.

ψ teória pozostáva z organizačného teorému a zo štyroch axiómov: úkonového axiómu (*operation axiom*), transakčného axiómu (*transaction axiom*), kompozičného axiómu (*composition axiom*) a úrovňového axiómu (*distinction axiom*).

2.5.1 Úkonový axióm (*the operation axiom*)

Úkonový axióm hovorí, že prevádzka organizácie je tvorená aktivitami (úkonami) účastníkov. Tieto aktivity predstavujú základné autoritatívne jednotky, ktoré sú zodpovedne naplňované ľuďmi. Účastníci vykonávajú dva druhy aktov: produkčný akt (skrátene P-akt) a koordinačný akt (skrátene C-akt). Vykonaním P-aktu, účastník pomáha vytvárať tovar alebo službu, alebo informáciu, alebo dáta, ktoré majú byť predané (doručené) ďalším účastníkom. Daný P-akt môže byť ako hmotný tak aj nehmotný. Príklady nehmotných aktov sú napr. rozhodnutie súdu odsúdiť niekoho, poskytnutie poistnej zmluvy či predaja tovaru. Vykonaním C-aktu, účastníci uzatvárajú a dodržujú záväzky voči sebe, aby sa mohol následne vyvolať P-akt. C-akt je definovaný jeho návrhom a jeho zámerom. Jeho zámer pozostáva z P-faktu (napríklad „Objednávka



Obr. 2.6: Grafická reprezentácia úkonového axiómu (zdroj: [7])

č. 101 je doručená“) a z dátumu doručenia. P-fakt (produkčný fakt) pritom chápeme ako výsledok úspešného vykonania produkčného aktu. Podobne tak chápeme C-fakt (koordinačný fakt): jedná sa o výsledok úspešne vykonaného koordinačného aktu. Zámer reprezentuje účel vykonávateľa, kde zámer môže byť „požiadavok“ – *request*, „sľub“ – *promise* a „odmietnutie“ – *decline*. Dôsledok vykonania C-aktu je, že vykonávateľ aj adresát aktu sa zapojujú do záväzku spojeného s P-aktom.

Treba spomenúť, že tak ako C-akty, tak P-akty majú efekt na tzv. koordinačný svet – C-svet, resp. na produkčný svet – P-svet. Stav P-sveta predstavuje množinu P-faktov a stav C-sveta je množinou C-faktov. Presnejšie vravíme, že stav P-sveta je v určitý čas tvorený množinou P-faktov, ktoré boli vytvorené do daného času – podobne takto opisujeme C-svet. Takto sme si schopní udržovať históriu oboch svetov [7]. Všetko, čo do neho vložíme, už nejde odvrátiť, resp. zmazať – fakt nemôže byť zničený. Jediný spôsob je vytvorenie tzv. anti-faktu, ktorý obnovuje situáciu pred vytvorením inkriminovaného faktu – túto procedúru nazývame odvolanie (*cancellation*).

Na grafickej reprezentácii úkonového axiómu 2.6 znázorňujeme C-svet ako kruh, postavu človeka ako role účastníkov a P-svet ako diamant. Celá šípka znázorňuje situáciu, kedy účastník vykonáva C-akt, resp. P-akt. Prerušovaná šípka vyjadruje, že účastníci berú do úvahy aktuálny stav C, resp. P sveta. Na danom obrázku vidíme ešte ďalšie tri okruhy pridané k úkonovému axiómu: zodpovednosť, oprávnenie a kompetencia. **Kompetenciu** chápeme ako schopnosť subjektu vykonať určitý P-akt spolu s korešpondujúcim C-aktom. Táto schopnosť sa vyznačuje najmä v produkčnej fáze. Na to, aby akt daný subjekt mohol vykonať, potrebuje mať získané **oprávnenie**. Keď už daný subjekt disponuje kompetenciou a oprávnením pre vykonanie aktu, očakáva sa, že ho vykoná **zodpovedne**.

2.5.2 Transakčný axióm (*the transaction axiom*)

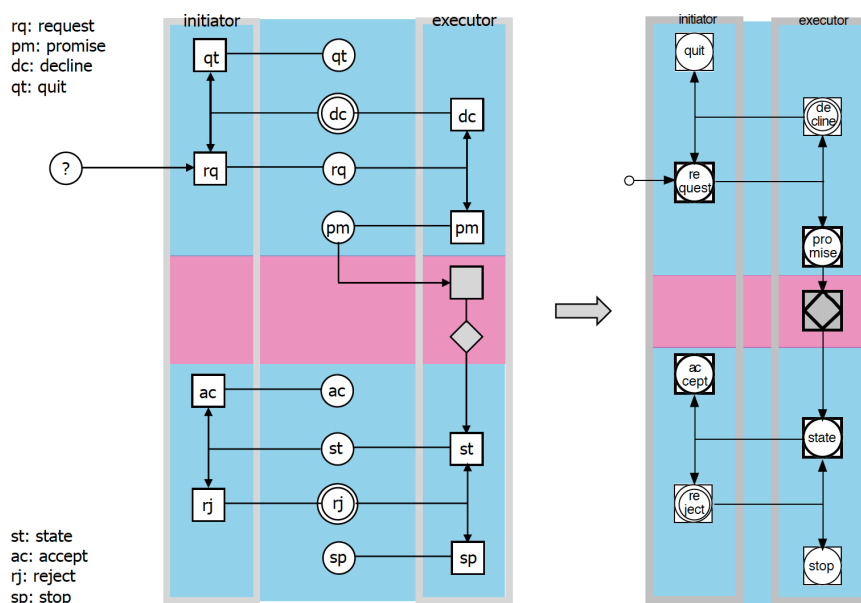
Transakčný axióm hovorí, koordinačné akty sú vykonávané v krokoch podľa univerzálnych vzorov. Pre vykonanie aktu tieto vzory – tiež nazývané transakciami – zahrňujú vždy dve účastnícke role. Transakcia pozostáva z dvoch konverzácií: „príkazovej“ a „výslednej“. Konverzácia je definovaná ako sekvencia koordinačných aktov medzi dvoma účastníckymi rolami, ktoré majú za cieľ dosiahnutia príslušného P-aktu, resp. P-faktu.

Transakcia zahrňuje tri fázy: príkazovú *order* – O-fázu, vykonávaciu *execution* – E-fázu a výslednú *result* – R-fázu. Jedna z dvoch rolí sa nazýva iniciátor transakcie a druhá vykonávateľ transakcie.

Predstavme si štandardný transakčný vzor 2.7: pozostáva z troch horizontálnych častí prislúchajúcich transakčným fázam: hornej časti – príkazová fáza, stredná časť – vykonávací fáza a dolná časť – výsledná fáza. V príkazovej časti sa iniciátor a vykonávateľ snažia vytvoriť konsenzus o P-fakte, ktorý následne vytvorí vykonávateľ v nejaký čas. Vo vykonávacej časti vykonáva vykonávateľ jeho P-fakt a vo výslednej časti iniciátor a vykonávateľ sa snažia vytvoriť konsenzus o tom, že P-fakt bol naozaj vytvorený v daný čas. Na to, aby P-fakt mohol vzniknúť, musí vzniknúť medzi nimi i spomínaný konsenzus. Keď si v štandardnom vzore vezmeme do úvahy cestu *request* -> *promise* -> *execute* -> *state* -> *accept* – požiadavok, sľub, vykonanie/produkcia, podanie a prijatie, pôjde o základný transakčný vzor. Základný transakčný vzor odpovedá *happy path* – situácii, kedy sa iniciátor a vykonávateľ zakaždým zhodnú. To ale často nemusí tak bývať. Vo vzore máme dva stavy, v ktorých môže nastať iná situácia – v stave *requested* a v stave *stated*. Miesto sľubu môže účastník odpovedať na požiadavok odmietnutím (*decline*) a miesto prijatia môže odpovedať zamietnutím (*reject*) – tieto úkony dostanú proces do stavu *declined*, resp. *rejected*. V takom prípade sa musia účastníci procesu dohodnúť a vyjednať výsledok danej situácie. Možný výsledok je obnoviť požiadavok (*request*) alebo podanie výsledku (*state*) s pravdepodobnou modifikáciou produktu, alebo neúspech, čomu odpovedajú stavy *quit* alebo *stop*. Existujú tri tzv. *validity claims* – *claim to truth*, *claim to justice* a *claim to sincerity* – na ktorých je postavená komunikačná teória (*Theory of Communicative Action*).

Relevantnosť transakčného axiómu leží na zistení všetkých C-aktov vrátane tých, ktoré boli vykonané implicitne. Akty sú vytvárané postupne podľa toho ako procesom putujeme, napr.: iniciátor požiada (prebehne vytvorenie C-aktu do C-sveta) vykonávateľa o vytvorenie nejakého produktu, vykonávateľ sľúbi jeho vytvorenie (vytvorenie C-aktu do C-sveta), vykonávateľ vytvorí produkt (vytvorenie P-aktu do P-sveta) vyústení do P-faktu, vykonávateľ následne ponúkne svoj produkt s informáciou, že je hotový (vytvorenie C-aktu do C-sveta) a iniciátor ho prijme (vytvorenie C-aktu do C-sveta).

Štandardný transakčný vzor môžeme ďalej rozšíriť na kompletný transakčný vzor znázornený na obrázku 2.8. Predstavuje mechanizmus, vďaka ktorému v akomkoľvek stave procesu, môžeme odvolať jednu zo štyroch základ-



Obr. 2.7: Štandardný transakčný vzor a jeho skrátaná verzia (zdroj: [7])

ných krokov (požiadavok – *request*, sľub – *promise*, predanie – *state* a prijatie – *accept*) štandardného transakčného vzoru. Tým pádom akoby resetoval proces do nejakého špecifického stavu.

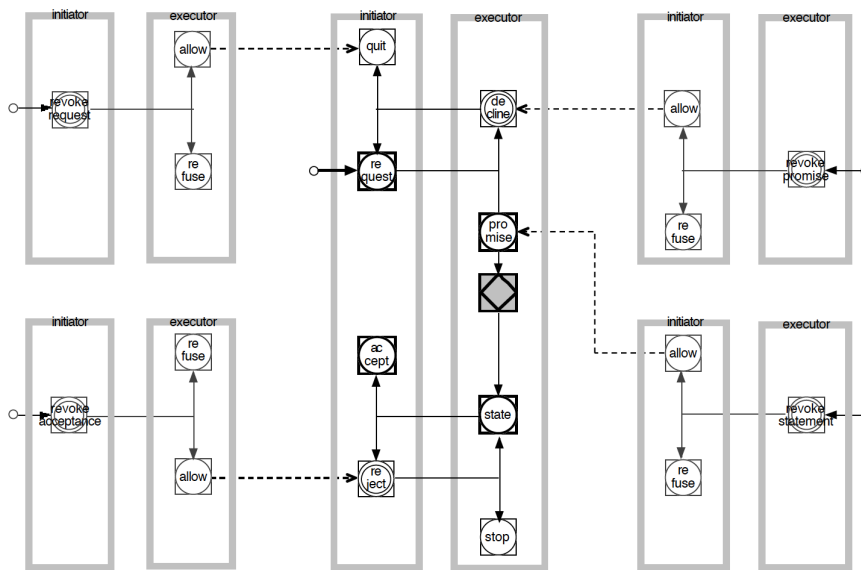
Tzv. *success path* transakcie predstavuje úspešné vykonanie krokov: *request*, *promise*, *execution*, *state* a *accept*. Na základe požiadavku vykonávateľ vykoná sľub, alebo tiež môže odmietnuť požiadavku (vykoná *decline*) a v ďalšom kroku buď iniciátor zmení svoj požiadavok a vytvorí nový, alebo transakciu ukončí a dôjde k vykonaniu kroku *quit*. Podobným spôsobom funguje pri predaní produktu – iniciátor ho môže odmietnuť (*reject*) a buď sa transakcia vykonávateľom ukončí, alebo vykonávateľ vytvorí nový produkt.

Ad-hoc krok *revoke request* funguje tak, že ho môže kedykoľvek vyvolať iniciátor (rozhodol sa vziať späť svoj požiadavok) a ostáva na vykonávateľovi, či mu ho prijme alebo zamietne. Ak by mu požiadavok na odvolanie požiadavku zamietol, transakcia pokračuje. V prípade schválenia odvolania požiadavku sa transakcia dostáva ku kroku *quit*, po ktorom končí.

2.5.3 Kompozičný axióm (*the composition axiom*)

Kompozičný axióm hovorí, že transakcia môže byť vložená do inej, ktorá napr. čaká na dokončenie vlozenej. Ďalej môže byť transakciou pod správou zákazníka (externá – mimo organizáciu), alebo môže ísť o samo-aktivujúcu transakciu. Platí, že jedna transakcia teda môže pozostávať z viacerých transakcií.

Keď vytvárame nejaký produkt, napríklad bicykel, potrebujeme najprv vyrobiť časti, z ktorých ho následne zložíme. Je jasné, že bicykel musíme zosťa-

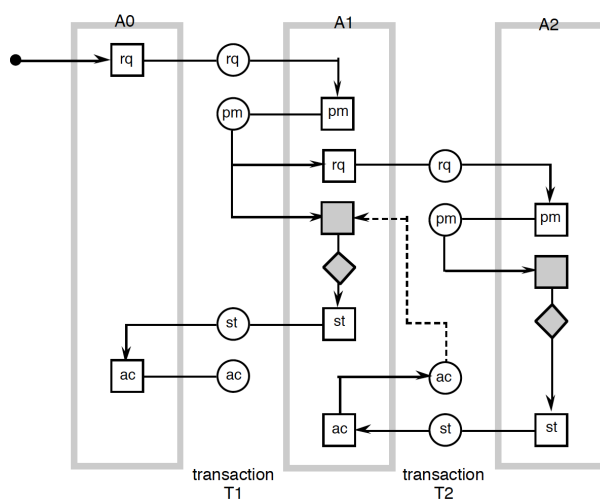


Obr. 2.8: Kompletný transakčný vzor (zdroj: [7])

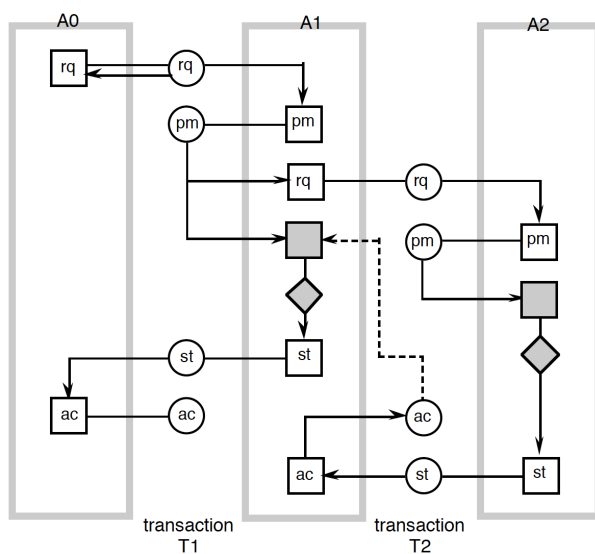
viť v nejakom logickom poradí – napr., že najprv potrebujeme mať zhotovený rám, aby sme naň následne mohli nasunúť kolesá. Vo význame P-faktov tak musíme mať niektoré P-fakty vytvorené skôr (predstavujú prerekvizitu) pre vytvorenie iných P-faktov. Toto docielime použitím vlozenej transakcie – príklad vlozenej transakcie vidíme na obrázku 2.9. Vo vstave *promised* transakcie T1 vykonávateľ (A1) vykonáva dva akty. Prvý predstavuje vykonanie produkčného aktu (šedý štvorec) a druhý je zase vykonanie požiadavku pre transakciu T2, v ktorej vykonávateľ predstavuje účastnícku rolu A2. Toto zaisťuje, že v rámci každej transakcii T1 je transakcia T2 začatá. Prerušovaná čiara zo stavu *accepted* transakcie T2 do produkčného aktu transakcie T1 znamená, že vykonanie produkčného aktu musí čakať na dokončenie transakcie T2. Vďaka tomu sa transakcia T2 stáva vloženou transakciou do transakcie T1 [7].

Na obrázku 2.9 ešte vidíme, že transakcia T1 je spustená externe (mimo organizáciu) – to indikuje šípka s bodkou. Tú môže spustiť napr. zákazník alebo iný subjekt mimo organizáciu. Rozšírením hraníc (*Scope of Interest*) by sa transakcia T1 mohla stať internou s iniciátorom s internou účastníckou rolou. Podľa obrázku iniciátor transakcie T1 hrá tzv. environmentálnu účastnícku rolu.

Existuje ešte ďalší spôsob ako spustiť transakcie a ten nazývame samo-spustenie alebo samo-aktivovanie. Obrázok 2.10 ukazuje príklad takej transakcie využívajúcej spomínaný spôsob. Zo stavu *requested* transakcie T1 sa vykonávajú dva akty. Prvý je sľub transakcie T1 vykonaný vykonávateľom transakcie T1 (A1). Druhý akt je požiadavok pre ďalšiu transakciu T1 vykonanej iniciátorom transakcie T1 (A0). Táto štruktúra samo-aktivácie je generickým



Obr. 2.9: Príklad vloženj transakcie do inej transakcie (zdroj: [7])



Obr. 2.10: Príklad transakčného samospúšťania (zdroj: [7])

riešením pre najmä periodické a kontrolné aktivity.

Kompozičný axióm nás donucuje sa zamyslieť nad definíciou biznisového procesu, nad transakciami a ich druhmi v ňom a nad internými a externými rolami.

2.5.4 Úrovňový axióm (*the distinction axiom*)

Úrovňový axióm slúži pre rozlíšenie záujmov, resp. úrovní v biznisovom procese. Hovorí, že existujú tri rôzne ľudské spôsobilosti či schopnosti hrajúce rolu pri konaní aktérov. Nazývame ich *performa*, *informa* a *forma*. Tieto schopnosti sa týkajú komunikácii, vytvárania vecí, odôvodňovania a spracovania informácií [7]. Ich prehľad s príkladmi sumarizujeme na obrázku 2.11.

Forma (latinsky *form*) sa zaoberá formami komunikácie a formami manipulovania s informáciami. Máme teda na mysli vyslovenie a vnímanie viet v nejakom jazyku, ich syntaktickú analýzu, ich kódové schémy, prenosy dát, ukladanie a získavanie dát alebo dokumentov. *Forma* by mala tiež zahŕňať fyzické uloženie objektov predstavujúcich zakódované informácie.

Informa (latinsky *in-forma* znamená, čo je vo forme) sa zaoberá obsahovými aspektmi komunikácie a informácie. Zaoberáme sa teda tiež komunikáciou aj informáciou, ale na tejto úrovni úplne abstrahujeme aspekty *formy*. Zaoberáme sa s vecami ako zdieľaním myšlienok medzi ľuďmi, pamätaním alebo získavaním vedomostí a odôvodňovaním.

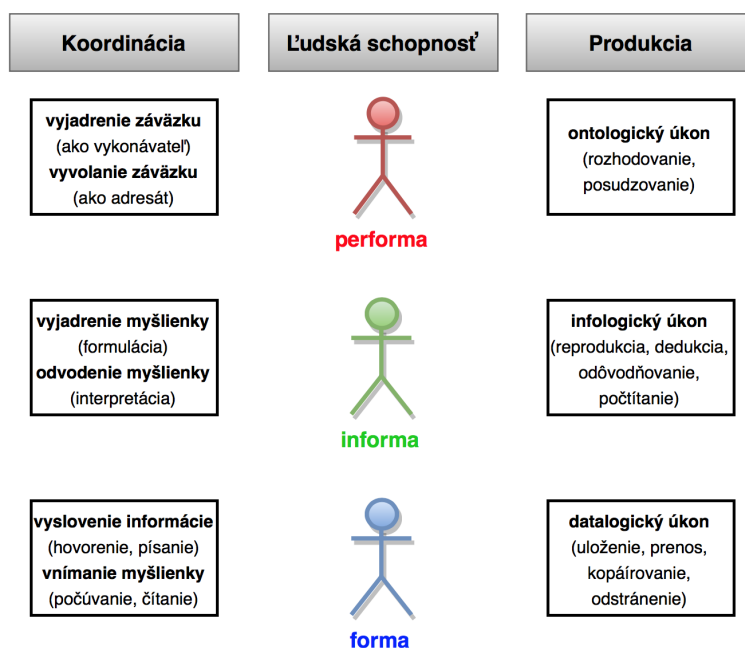
Performa (latinsky *per-forma* znamená skrz myšlienku) je o prinášaní nových, **originálnych** vecí pomocou priamej či nepriamej komunikácie. Máme na mysli zapojenie sa do záväzku a rozhodovanie, ohodnocovanie. Túto úroveň, resp. schopnosť považujeme za **esenciálnu** ľudskú schopnosť pre vykonávanie biznisu rôzneho druhu.

Typicky pre jeden ontologický úkon musíme vykonať viacero infologických úkonov a pre jeden infologický úkon podobne viacero datologických úkonov. Tieto úrovne nám tak poskytujú významnú redukciu komplexity procesného modelu, kde sú modelované iba ontologické úkony.

2.5.5 Koordinácia

V tejto sekcii si popíšeme proces vykonávania koordinačného aktu. Obrázok 2.12 nižšie znázorňuje typické druhy komunikačných aktov, ktoré musia byť vykonané, aby sme mohli úspešne dokončiť koordinačný akt. Existujú tri druhy komunikačných aktov korešpondujúcich s tromi ľudskými schopnosťami, resp. úrovňami abstrakcie znázornenými na obrázku so znázorneným úrovňovým axiómom 2.11 [7].

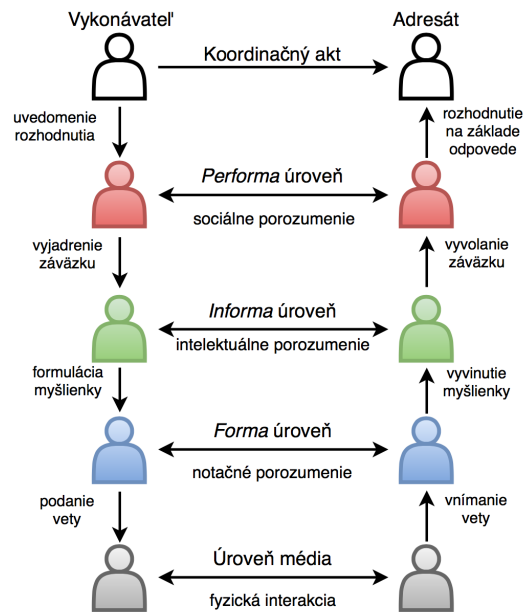
Aby sme úspešne vykonali koordinačný akt smerom od vykonávateľa k adresátovi, vykonávateľ musí vyjadriť záväzok vo vykonaní aktu adresovanom adresátovi, ktorý musí vyvolať adekvátnu odpoveď na základe korešpondujúceho vykonaného aktu. Tento akt je časťou výmeny na úrovni *performy*. Väčšinou ide o jediný akt v rámci výmeny, no dodatočné akty sa zvyknú týkať objasňovania spoločného sociálneho porozumenia koordinačného aktu, resp. zámeru koordinačného aktu. To, že úspešne dospejeme ku spoločnému sociálnemu porozumeniu koordinačného aktu, nazývame *performa condition*.



Obr. 2.11: Prehľad úrovňového axiómu (zdroj: [7])

Jediná cesta akou má vykonávateľ jeho záväzok vyjadriť a dať adresátovi najavo jeho záväzok, je vyjadrenie záväzku pomocou jeho *informa* schopnosti. To je nasledované vynútením ekvivalentnej myšlienky v mysli adresáta. Používame pojem myšlienky ako všeobecné označenie stavu mysle, nakoľko sme na úrovni úrovni *informy*. Vykonávateľ je s adresátom súčasťou *informa* výmeny. Takáto výmena pozostáva z *informa* aktov, ktoré majú za cieľ intelektuálne porozumenie koordinačného aktu medzi vykonávateľom a adresátom. Existujú aspoň dva *informa* akty. Jeden je taký, kde vykonávateľ je rozprávač a adresát je načúvač, v ktorom vykonávateľ informuje adresáta o obsahu koordinačného aktu. Druhý akt je taký, kde adresát je rozprávač a vykonávateľ zas načúvač a v ktorom adresát potvrdzuje vykonávateľovi, že jeho intelektuálnemu zámeru rozumie. Aktov môže samozrejme prebehnúť viac – napríklad v prípade, že vykonávateľ nešpecifikoval svoj zámer v prvom *informa* akte. To, že úspešne dospejeme k spoločnému intelektuálnemu porozumeniu koordinačného aktu, nazveme *informa condition* – ide o podmienku pre úspešnú koordináciu.

Vyjadrenie myšlienky môže byť vykonané len jej formuláciou vo vete v nejakom jazyku a jej vyslovením alebo napísaním atď. Podobne to bude fungovať pri adresátovi pri vyvinutí, resp. vytvorení nejakej myšlienky. To vyžaduje, aby existovalo niečo, čo sa dá vnímať - pomocou počúvania alebo čítania. Existuje teda ďalšia úroveň, v ktorej sa musí vykonávateľ a adresát spoločne porozumieť a to úroveň *forma*. Hovoríme teraz o notačnom porozumení (*significational understanding*), čo znamená, že vykonávateľ používa



Obr. 2.12: Koordinácia a jej úrovne (zdroj: [7])

jazyk, ktorý adresát ovláda a je schopný rozoznať, čo od neho vykonávateľ žiada. Spomínané porozumenie prichádza spolu s výmenou na úrovni *forma*. Daná výmena pozostáva z aspoň dvoch *forma* aktov. Prvý predstavuje vyslovenie (vyjadrenie) vety vykonávateľom adresátovi a druhý akt predstavuje potvrdenie adresáta, že vnímal danú vetu. Pre každý *informa* akt je potrebné, aby nastala nejaká *forma* výmena. Úspešné dosiahnutie notačného porozumenia koordinačného aktu nazveme *forma condition*.

Nakoniec, existuje výmena na fyzickej úrovni, kde vyslovenie vety vykonávateľom je nejakým spôsobom zakódované a prenesené adresátovi. Predpokladáme, že podmienka korektného fyzického presunu je zahrnutá v podmienke pre úspešné vykonanie *forma*. Všeobecne budeme hovoriť o veciach, ktoré sú vyslovené vykonávateľom, prenesené adresátovi a vnímané adresátom ako o dátach.

DEMO

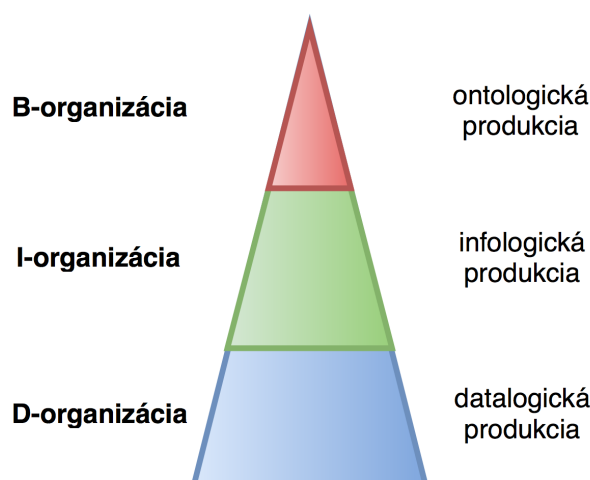
V nasledujúcej kapitole si predstavíme poslednú časť ψ teórie – organizačný teorém, ktorý s neskôr popisovanou realizáciou organizácie a implementáciou organizácie, predstavuje posledný krok k predstaveniu jednotlivých modelov DEMA.

3.1 Organizačný teorém

Posledná časť ψ teórie sa zaoberá organizačným teorémom, ktorý hovorí, že organizácia nejakého podniku je sociálny systém, ktorý je tvorený integračnými vrstvami troch homogénnych systémov: B-organizáciou (biznis), I-organizáciou (intelekt) a D-organizáciou (dokument). D-účastník v D-organizácii podporuje I-účastníkov I-organizácie a I-účastníci v I-organizácii podporujú B-účastníkov v B-organizácii [10].

Obrázok 3.1 zobrazuje grafickú reprezentáciu organizačného teorému. Ukazuje vzťahy medzi tromi systémovými aspektami. Všetky tri homogénne systémy sú v kategórii sociálnych systémov, čo znamená, že sú si podobné z hľadiska koordinácie: elementy sú subjekty, ktoré vstupujú do súladu so záväzkami medzi sebou. Líšia sa len v jednom druhu produkčného aktu: produkcia v B-organizácii je ontologická, produkcia v I-organizácii je infologická a na úrovni D-organizácie je datalogická. Toto delenie je dôvod, prečo považujeme organizáciu podniku za heterogénny systém. B-organizáciu, I-organizáciu a D-organizáciu nazývame aspektové systémy organizácie podniku [7].

Trojuholníkový tvar obrázku s grafickou reprezentáciou organizačného teorému 3.1 vyjadruje, že nad ontologickou úrovňou sa už nenachádza nič. Inými slovami tiež povedané, že poznanie, resp. znalosti B-organizácie podniku sú kompletným poznaním a znalosťami esencie podniku. Zvyšok predstavuje už len realizáciu a implementáciu. Rozdelenie obrázka na červenú, zelenú a modrú oblasť pomerovo tak, že obsah červenej oblasti je najmenší a obsah modrej časti je najväčší, znamená niekoľko vecí. Po prvé to znamená, že v rámci koordinácie to predstavuje približné podiely množstiev úsilia vynaložených na



Obr. 3.1: Reprezentácia organizačného teorému (zdroj: [7])

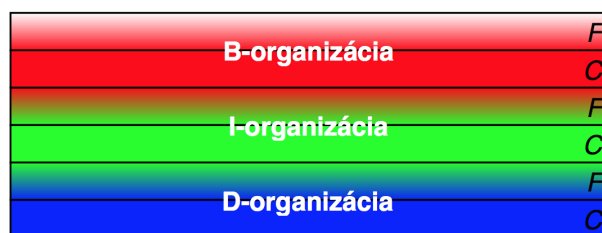
performa výmenách, *informa* výmenách a *forma* výmenách, resp. sprostredkovaníach pri vykonávaní koordinačných aktov. Po druhé to znamená, že v rámci účastníckych rolí to predstavuje približné podiely počtov účastníckych rolí v B-organizácii, I-organizácii a D-organizácii. Po tretie to znamená, že v rámci produkcie to predstavuje približné podiely veľkostí snahy vynaloženej v ontologickej produkcii, infologickej produkcii a datologickej produkcii.

Samozrejme, že podnik je viac než len dobre vyvinutá integrácia troch organizačných aspektov. Ľudské bytosti sú tiež biologické bytosti a to znamená, že potrebuje určité prostredie, v ktorom žijú. Podobne potrebujú špecifické zariadenia, aby ich život bol pohodlnejší. Bytím biologickým individuum zahrňuje aj to, že sme fyzikálnou vecou a teda musíme naplňovať aj niektoré fyzikálne požiadavky – napríklad priestor (kanceláriu) pre prácu, služby pre transport (MHD, auto). Tiež sme i emocionálnou bytosťou, psychologickou bytosťou atď. V našej práci budeme predpokladať, že všetky požiadavky týkajúce sa týchto aspektov budú splnené.

3.2 Realizácia organizácie

Realizáciou podniku chápeme ako dôkladnú integráciu všetkých troch organizačných aspektov. Zakladá sa na vzájomnom spojení všetkých troch organizačných aspektov, pričom niektorý systém z niektorej vrstvy môže podporovať systém z vyššej vrstvy. Opačne tiež niektorý systém môže byť podporovaný systémom nižšej vrstvy. Teda B-organizácia využíva I-organizáciu a I-organizácia používa D-organizáciu.

Pripomeňme si ďalej rozdiel medzi funkciou a konštrukciou nejakého systému. Spomínali sme, že zaobchádzanie s funkciou nejakého systému a jeho



Obr. 3.2: Integrované vrstvy organizácie (zdroj: [7])

konštrukciou predstavujú dve úplne rozdielne veci. Uvažujme napríklad hodiny – funkcia hodín je jednoduchá a každý ju pozná: vraví nám čas, meraný štandardným spôsobom. Spôsoby, ktorými sú hodnoty (dni, minúty a sekundy) zobrazované, sa môžu líšiť – máme napríklad analógový displej s ručičkami alebo digitálny s ciframi desiatkovej sústavy. Hoci je ich funkcia identická, hodiny sa môžu veľmi líšiť v spôsobe, ktorým boli skonštruované. Existujú napr. mechanické hodiny, elektrické hodiny, vodné hodiny, pneumatické hodiny atď. Na tomto príklade vidíme, že funkcia hodín nemá prirodzene daný vzťah s konštrukciou. Funkcia je popisovaná z pohľadu užívateľa hodín tak, že vidí na nich čas, hodiny, minúty a pod. Konštrukciu môžeme zas popisovať pomocou komponent a elementov, ktoré navzájom interagujú. Tieto komponenty a elementy „nevedia“ o hodinách, minútach ani sekundách. Miesto toho „vedia“ len o ich mechanických silách, rýchlosti, elektrickom napätí a pod. Je to práve remeselnícke umenie konštruktéra hodín, ktorý prepája funkciu s konštrukciou, čo spôsobí, že činnosť (prevádzka) konštrukcie vytvorí požadované funkčné správanie.

Podobné dedukcie ako pri hodinách platia i pre organizácie. Obrázok 3.2 vyjadrujúci jednotlivé vrstvy v sebe zahrňuje ešte aj funkčnú perspektívu (F) aj konštrukčnú perspektívu (C). Vidíme, že funkcia I-organizácie podporuje konštrukciu B-organizácie a funkcia D-organizácie podporuje konštrukciu I-organizácie.

Na to, aby sme si objasnili význam pojmu integrácie v rámci daných troch organizačných aspektov, uvažujme najprv I-organizáciu. Keď vezmeme do úvahy funkčnú perspektívu, vidíme, že I-organizácia poskytuje informačné služby konštrukcii B-organizácii, t.j. B-účastníkom. Je veľmi dôležité si uvedomiť, že poskytovaná informačná služba je určená týmito B-účastníkmi a preto musí byť (funkčne) popísaná z hľadiska činnosti, resp. prevádzky B-organizácie. Napríklad, nejaký B-účastník v spoločnosti môže chcieť vedieť na konci každej pracovnej doby, aký bol denný obrat. Obrat je pojem, ktorý pre neho niečo znamená a spadá do rámca ostatných ekonomických pojmov, ktoré sú pre neho základom a sú nevyhnutné pre jeho rolu v B-organizácii. Oproti B-účastníkom I-účastníci už nepoznajú pojem obrat. I-účastník nebude nervózny, ak obrat rapídne klesne. Pojem obrat používajú len v rámci dodávanej

3. DEMO

služby. Nie je to pre nich nič viac ako len označenie pre výdaj konkrétnej kalkulácie. No keďže sú I-účastníci časťou konštrukcie I-organizácie, sú zodpovední za vykonanie I-transakcií, v ktorých sa produkujú I-produkty. Ako príklad I-transakcie uveďme denný proces spočítania dobre nadefinovaných výpočtov, ktorých výsledok je označený ako „obrat“. Výsledok je doručený iniciátorovi I-transakcie a vykonávajúci I-účastník (vykonávateľ) pozná obrat ako sumu niečoho. Musíme si uvedomiť, že toto celé je ale vyvolané B-účastníkom, ktorý potrebuje vedieť, aký je obrat – ako môže byť ale B-účastník iniciátorom I-transakcie? Je to vďaka tomu, že subjekt, ktorý spĺňa rolu B-účastníka, disponuje všetkými troma schopnosťami. Pri vykonávaní ontologického produkčného aktu, účastník využíva predovšetkým svoju *performa* schopnosť. Na druhú stranu, iniciátor alebo vykonávateľ B-transakcie potrebuje všetky tri schopnosti pre vykonanie koordinačných aktov (viď 2.12).

Podľa všetkého nie je ťažké striedať schopnosti pri ich používaní. Subjekt pri používaní dočasne preberie podobu I-účastníka, ktorá perfektne sedí pre zapojenie sa I-transakcie. Po tom, čo prijme výsledok I-transakcie, subjekt sa vráti k používaniu svojej *performe* a teda preberie späť podobu B-účastníka, ktorý ďalej pracuje už s náležitou znalosťou o obratoch a pod.

Pozrime sa lepšie na I-účastníka, ktorý produkuje hodnotu obratu ako výsledok vykonanej I-transakcie. Kompetencia tohto účastníka je jasná – počítať súčty alebo vykonávať iné matematické operácie. Na to, aby spočítal sumu jednotlivých vzorcov, potrebuje mať k dispozícii samotné vzorce. Dostávame sa k podobnej situácii ako o odstavce vyššie a nastáva otázka, ako sa I-účastník dozvie o týchto vzorcov. Subjekt vystupujúci v role I-účastníka prevezme podobu role D-účastníka a v tejto podobe iniciuje D-transakciu, ktorej výsledok je doručenie dokumentu, ktorý obsahuje požadované informácie. Po tom, čo výsledok prijme, subjekt sa vracia do svojej podoby I-účastníka, v ktorej je už schopný interpretovať získanú informáciu a vykonať potrebné infologické úkony [7].

3.3 Implementácia organizácie

Implementáciou organizácie alebo akéhokolvek systému rozumieme vytvorenie fungujúcej realizácie organizácie z pohľadu technológie [7]. Podľa ϕ teórie implementačný model organizácie musí odpovedať konceptuálnemu modelu fungujúcej (*operational*) organizácie. Konceptuálny model je reprezentovaný diagramami, popismi procedúr, inštrukciami k práci, software kód atď. Časť implementačného modelu môže byť vykonávaná počítačovými systémami a z tohto ohľadu vhodné účastnícke role môžu byť naplňované software programami. V takom prípade hovoríme o automatizovaných účastníkoch alebo tiež agentoch. Rozlišujeme dva typy agentov: ľudskú bytosť (subjekt) a agenta. Subjekt samozrejme môže využívať agenta [10]. Pri implementácii sa snažíme myslieť na sociálne systémy okrem len racionálnych – tieto dva druhy ne-

spadajú do rovnakej kategórie [13]. Jan Dietz [7] totiž všetky tri homogénne systémy (B-organizáciu, I-organizáciu a D-organizáciu) považuje za sociálne systémy. Napriek tomu, že úkony v I-organizácii a D-organizácii sú vo veľa prípadoch automatizované, Dietz jasne ukazuje, že hlavne v rámci produkčných aktov v B-organizácii a koordinačných aktov naprieč všetkými tromi organizáciami sú na vykonávanie určené ľudským bytostiam. Minimálne úkony, pri ktorých používame *performa* schopnosť, nemôžu vykonávať stroje. Od roku 2000 bolo zverejnených niekoľko štúdií, ktoré tvrdia, že informačný systém je v podstate racionálny systém, ktorý pracuje ako objektový systém B-organizácie. Tieto štúdie ignorujú to, že informačný systém sám o sebe je tiež racionalizáciou sociálneho systému – I-organizácie a D-organizácie.

Podľa Mallensa a Dietza [14] je informačný systém založený na B-organizácii a je kategorizovaný ako racionálny systém. Jednotlivé elementy systému nazývajú racionálnymi individuami (*rational individuals*), ktoré vykonávajú racionálne úkony ako získavanie, načítavanie, poskytovanie znalostí, výpočty a vytváranie logických dedukcií. Diskutujú o aplikovaní určitých spôsobov modelovania biznisových procesov pomocou rôznych techník pomocou UML, aby tak vytvorili model informačných systémov za účelom návrhu systému. Využitie UML na celý návrh má tri problémy. Prvým je vymedzenie problému – určenie rozsahu informačného systému, ktorý má byť vyvinutý. Druhý problém sa týka identifikácie všetkých relevantných informačných systémových konceptov v rámci určitého rozsahu a tretím problémom je špecifikácia konceptov informačného systému.

Pri DEME budeme implementáciou organizácie rozumieť: alokáciu subjektov na účastnícke role v B-organizácii; alokáciu subjektov alebo artefaktov na účastnícke role v I-organizácii; alokáciu subjektov alebo artefaktov na účastnícke role v D-organizácii; alokáciu vhodných technologických prostriedkov pre koordinačné akty a nakoniec alokáciu vhodných technologických prostriedkov pre produkčné akty. Výsledok by mal byť podnikový informačný systém (EIS – *enterprise information system*), ktorý je skutočnou **implementáciou realizácie esenciálneho modelu**.

Kompletnosť a relevantnosť požiadavkov pre EIS môžu byť garantované len v prípade, že sú postavené na esenciálnom modeli podniku. Iný prístup je bohužiaľ odsúdený k zániku. Vyprodukovanie esenciálneho modelu a určenie požiadavkov na ňom založených je zodpovednosťou biznisových znalcov ako architektov, analytikov atď. Biznisová časť podniku by si nemala myslieť, že ak niečo oni neurobia, urobia to za nich ľudia z IT.

Vezmime si príklad – ERP systémy. ERP systém by mal byť nejakou implementáciou nejakej realizácie esenciálneho modelu organizácie. Keďže poskytovatelia ERP systémov o esenciálnych modeloch nevedia nič, ERP systém je odsúdený k zániku.

Analýza procesov

V nasledujúcej kapitole si predstavíme procesný model a analýzu procesov životného postenia pomocou DEMO modelov. Dané bude slúžiť na pochopenie a ako podklad, resp. vstup pre metódu DEMO.

Jedná sa o nasledujúce procesy:

- Vznik životného poistenia (Vznik ŽP),
- Zmena životného poistenia (Zmena ŽP),
- Storno životného poistenia (Storno ŽP).

4.1 Úvod k analýze procesov

Pri analýze sa snažíme dívať na procesy z pohľadu biznisu. Je to pre nás jadro, z ktorého sa budeme odrážať. Musíme začať s čistým pieskom, na ktorom postavíme DEMO model a následne na tom navrhujeme informačné systémy. Musíme byť teda nezávislí od riešenia v BPM.

Nakoľko ale podstata BPM vo veľkej miere reflektuje samotný priebeh procesov, sa môže niekedy zdať, že popisujeme a analyzujeme už stávajúce BPM riešenie. Pri návrhu DEMO modelu sa snažíme zamerať hlavne na biznisovú časť procesov a zväziť filtráciu pridaných hodnôt implementovaného BPM riešenia. Je dobré rozlíšiť optimalizáciu samotného biznisového procesu, resp. popis samotného biznisového procesu od vlastnosti alebo pridanej funkcionality BPM riešenia.

Čo je to ale pridaná hodnota BPM riešenia? Predstavme si funkcionality F , ktorú BPM riešenie pridáva. Túto funkcionality odoberme a spýtajme sa, či daná funkcionality bude chýbať niektorej z troch organizačných aspektov. Ak by daná funkcionality chýbala – vo výsledku najmä pre B-organizáciu – tak o pridanú hodnotu naozaj ide. To, že ale implementované riešenie pomocou BPM systému pridáva nejakú hodnotu, DEMU vôbec nevedí.

Môžeme teda vykonávať analýzu už nad stávajúcim BPM riešením. Otázkou by možno bolo, ku ktorej verzii BPM riešenia sa máme prikláňať - na ktoré verzie chodu firmy. Bolo by ťažké odpovedať, ktorá z nich je najlepšia a najvhodnejšia pre stavbu DEMO modelu. Každopádne ak si predstavíme, že daný firemný proces prešiel svojim evolučným vývojom, množstvami optimalizácií, tak môžeme predpokladať, že posledná verzia je pre nás najvhodnejšia. Tiež argument, že sa na nej zhodol celý biznis, ktorý danú verziu zanalyzovali a schválili, nesie značnú váhu. Vznikla na základe podnetov pracovníkov jednotlivých oddelení a prešla si svojím vývojom.

4.2 Procesné modely

4.2.1 Vznik životného poistenia

Zjednodušený proces Vzniku ŽP je zložený z piatich fáz:

- určovania,
- oceňovania,
- overovani,
- korekcie a
- storna.

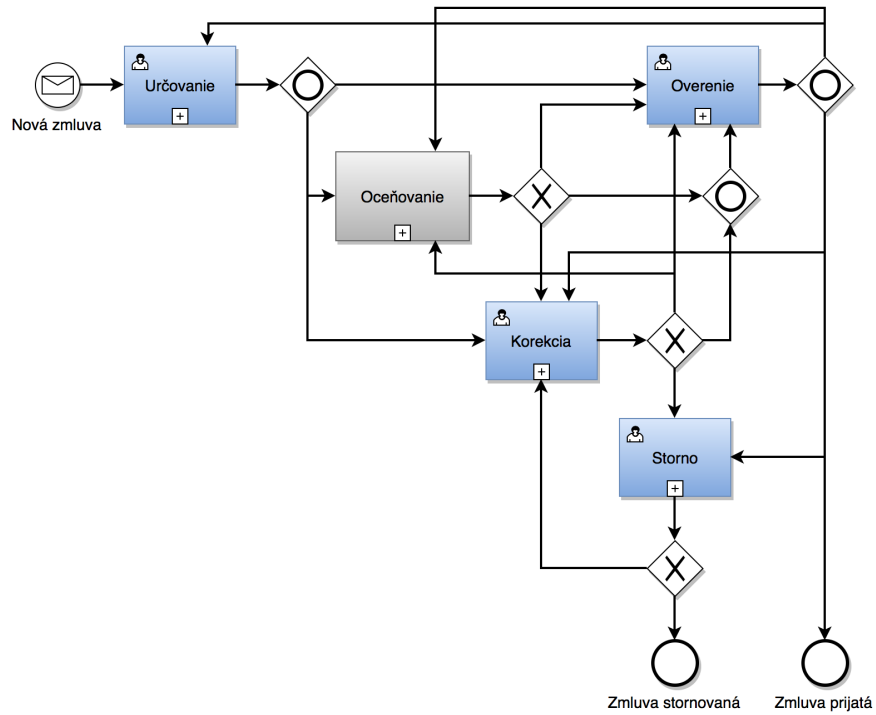
Tento zjednodušený proces vidíme na obrázku 4.1, v ktorom sú prechody medzi fázami definované. Charakter procesu je taký, že medzi danými fázami platí viac-menej to, že z každej fázy sa dá presť do inej fázy. Kvôli tomuto charakteru sa nám BPM(N) diagram javí ako spleť prechodov – šípok. DEMO nám takúto spleť odstraňuje a objasňuje úroveň jednotlivých úloh, takže v konečnom dôsledku je pohľad na proces zhora pre biznis jasnejší a vypovedajúcejší.

Fáza oceňovania – ako vidíme na obrázku 4.2 je zložená ešte z troch úloh. Samotné úlohy si popíšeme pri samotných transakciách v ďalšej sekcii.

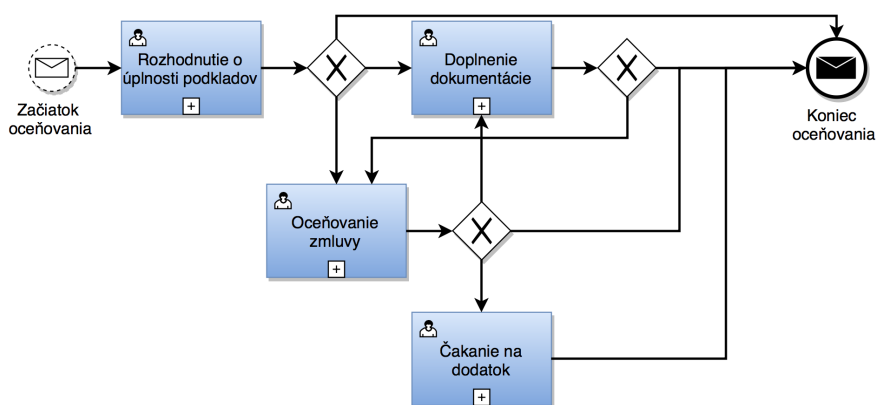
4.2.2 Zmena životného poistenia

Proces Zmena životného poistenia sa delí na:

- technické zmeny (TZ),
- netechnické zmeny (NTZ), ktoré delíme ešte na
 - jednoduché NTZ (vykonáva tím OBO) a
 - zložité NTZ (vykonáva tím Zmeny).

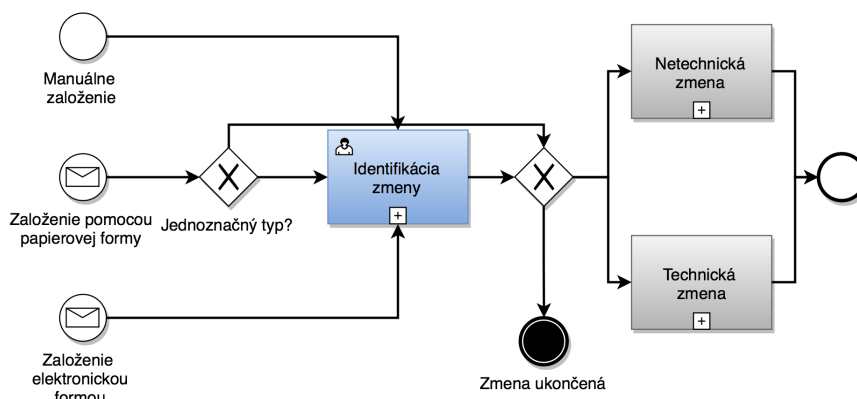


Obr. 4.1: Procesný BPM model procesu Vznik ŽP – pohľad z vrchu

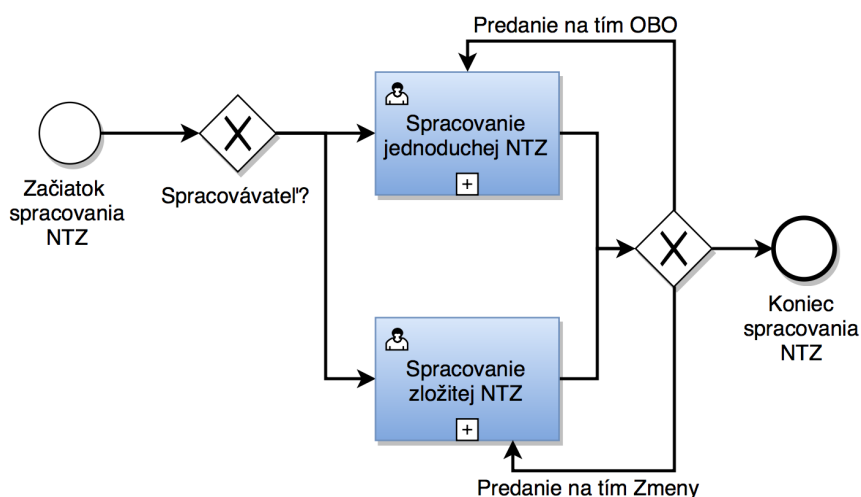


Obr. 4.2: Procesný BPM model procesu Vznik ŽP – fáza oceňovania

4. ANALÝZA PROCESOV



Obr. 4.3: Procesný BPM model procesu Zmena ŽP – pohľad z vrchu

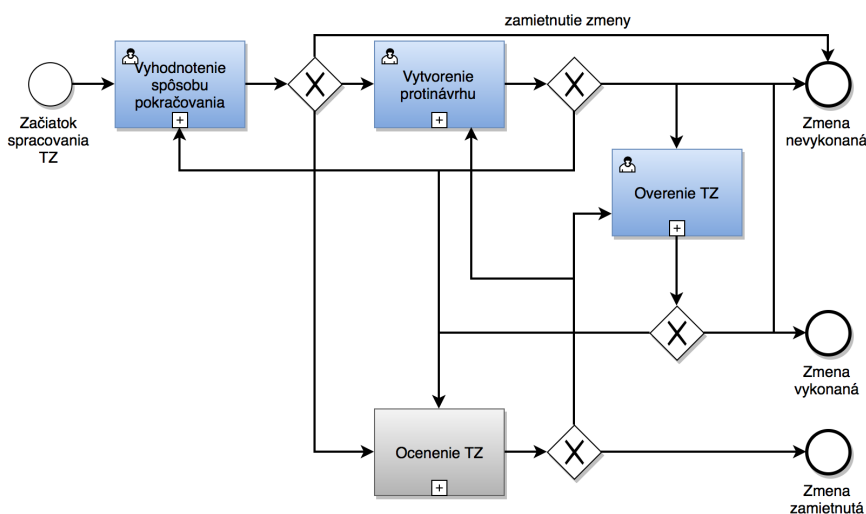


Obr. 4.4: Procesný BPM model procesu Zmena ŽP – netechnická zmena

Tím OBO (jedná sa v podstate o *back-office* zakladá workflow na základe prijatia e-mailu od klienta so žiadosťou o zmenu. Tím Oceňovanie spracováva ocenenie TZ, tím Zmeny spracováva zložité NTZ a TZ a tím Podateľňa doplňuje atribúty dokumentu (číslo poisťnej zmluvy, rodné číslo klienta a pod.) a určuje typ zmeny.

Na obrázkoch môžeme vidieť procesné *flow* pre: celý proces – pohľad z vrchu 4.3, netechnickú zmenu 4.4 a technickú zmenu 4.5.

Jednotlivé úlohy popíšeme v ďalšej sekcii, kde vydefinujeme tabuľky ontologických, infologických a datalogických procesov.



Obr. 4.5: Procesný BPM model procesu Zmena ŽP – technická zmena

4.3 Transakčné tabuľky

4.3.1 Vznik životného poistenia

Ako prvú transakciu sme pre ilustráciu zvolili transakciu, ktorá predstavuje hrubý pohľad na celý proces vzniku. Iniciátora považujeme za žiadateľa spracovania vzniku životného poistenia a za vykonávateľa všeobecne – spracovávateľa vzniku ŽP.

Prvá úloha, ktorá sa má po príchode novej zmluvy vykonať, je určenie typu zmluvy 4.1. Obyčajne žiadateľ býva podateľňa, ktorá svojou žiadosťou označí zmluvu k určeniu typu. Určiť typ jej môže ona sama, alebo účastník tímu Určovanie. V tabuľke už nerozlišujeme, o aký tím ide, ale píšeme, že vykonávateľ je určovateľ typu. Slub pri takomto označení zmluvy vzniká automaticky, avšak môžeme označenie odmietnuť, ak na prvý pohľad vidíme, že zmluva, resp. dokument zastupujúci zmluvu, nepatrí do procesu pre vznik ŽP alebo nepatrí na určenie typu (napr. typ už mohol byť určený). Transakciu sme určili, že patrí medzi I-organizáciu a teda ide o infologickú transakciu. Pri tejto činnosti vykonávateľ vykonáva určenie typu zmluvy na základe pohľadu na dokument - na hlavičku a jeho celkový obsah, podľa ktorého je schopný vcelku jasne určiť typ.

Transakciu overenia zmluvy vzniku životného poistenia 4.3 považujeme za ontologickú, pretože overenie, resp. verifikácia zmluvy, je pre nás dôležitý úkon – v čase jedinečný produkt – a predstavuje už len krok k dokončeniu celého procesu. Prebieha celková kontrola dokumentov a dochádza k **rozhodnutiu**, že sú všetky úkony a údaje, ktoré prešli spracovaním predošlých účastníkov procesu, sú v poriadku. Príchodom zmluvy ako *request* máme na mysli to, že

4. ANALÝZA PROCESOV

ID transakcie	T01_VZ
Názov transakcie	Spracovanie vzniku životného poistenia
Produkt	Vznik životného poistenia je spracovaný
Iniciátor	A01 – Žiadateľ o spracovanie vzniku životného poistenia
Vykonávateľ	A02 – Spracovávateľ vzniku ŽP
<i>Request</i>	Príchod novej zmluvy z podateľny
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Zmluva je spracovaná
<i>Accept</i>	Zmluva je vyhodnotená v poriadku
<i>Decline</i>	–
<i>Reject</i>	–
<i>Revoke request</i>	–
<i>Revoke promise</i>	–
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.1: Transakcia T01 – Proces Vzniku ŽP – hrubý pohľad

ID transakcie	T02_VZ
Názov transakcie	Určenie typu zmluvy
Produkt	Typ zmluvy je určený
Iniciátor	A03 – Žiadateľ o určenie typu zmluvy
Vykonávateľ	A04 – Určovateľ typu zmluvy
<i>Request</i>	Príchod novej zmluvy z podateľny
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: overenie, ocenenie, korekcia, overenie zároveň s ocenením alebo storno
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav zmluvy nie je označený ako Typ zmluvy určený
<i>Revoke request</i>	–
<i>Revoke promise</i>	Zmluva nie je určená na určenie typu
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.2: Transakcia T02 – Určenie typu zmluvy

buď si vykonávateľ sám pozrie, či sa v dokumentovej službe nachádza zmluva

určená k overeniu, alebo sa môže stať, že príde za ním iný účastník, ktorý ho požiada o overenie zmluvy. Odvolanie požiadavku môže nastať, ak iniciátor by na dokumente videl pri získaní dokumentu označenie, že výsledok ocenenia alebo korekcie zmluvy bol storno – táto situácia platí pre prípad, kedy je vykonávateľ a iniciátor fyzicky rovnaký subjekt. Z rovnakého dôvodu môžeme odvolať sľub (platí pre prípad, kedy iniciátor a overovateľ nie je fyzicky rovnaká osoba) s tým, že ďalší dôvod môže byť aj iný – zmluva nie je ešte vhodná na overenie, čo môže byť vidno prvý pohľad. Pridávame tiež ošetrenie situácie, kedy môžeme odmietnuť (*revoke*) výsledok overenia zmluvy, ak by iniciátor videl napr., že vykonávateľ neoznačil zmluvu ako „overená“.

ID transakcie	T03_VZ
Názov transakcie	Overenie zmluvy vzniku životného poistenia
Produkt	Zmluva životného poistenia je overená
Iniciátor	A05 – Žiadateľ overenia zmluvy
Vykonávateľ	A06 – Overovateľ zmluvy
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: zmluva je prijatá, určovanie, ocenenie, korekcia, overenie zároveň s ocenením alebo storno
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav zmluvy nie je označený ako Zmluva overená
<i>Revoke request</i>	Výsledok ocenenia alebo korekcie zmluvy bol storno
<i>Revoke promise</i>	Výsledok ocenenia alebo korekcie zmluvy bol storno, alebo zmluva nie je určená k overeniu
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.3: Transakcia T03 – Overenie zmluvy vzniku životného poistenia

Transakcia podania návrhu na storno 4.4 je infologická, nakoľko podľa jasných indikácií sa dá zhodnotiť, či zmluvu zamietneme alebo ju ďalej pošleme na korekciu.

Transakciu Ocenenie zmluvy 4.5 považujeme za veľmi dôležitú, pretože pri nej vzniká rozhodnutie, na základe ktorého sa v konečnom dôsledku zmluva prijme alebo zamietne. Na transakciu nadväzujú transakcie s rozhodnutím, či sú všetky podklady dodané ?? a prípadná transakcia doplnenia dokumentácie 4.8. Tu jasne vidíme, že ontologická transakcia môže vyvolať niekoľko info-logických transakcií, ako sme popisovali v sekcii s organizačným teóromom

4. ANALÝZA PROCESOV

ID transakcie	T04_VZ
Názov transakcie	Podanie návrhu na storno
Produkt	Návrh na storno je podaný
Iniciátor	A07 – Žiadateľ storna zmluvy
Vykonávateľ	A08 – Spracovávateľ storna zmluvy
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: zamietnutie zmluvy alebo korekcia
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav zmluvy nie je označený ako Storno rozhodnuté
<i>Revoke request</i>	–
<i>Revoke promise</i>	Zmluva nie je určená na stornovanie
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.4: Transakcia T04 – Podanie návrhu na storno

zobrazeným na obrázku 3.1.

Transakcia Oprava zmluvy je pre nás ontologická, nakoľko vytvárame de-facto novú zmluvu, ktorá je už opravená a je pre nás ontologicky novým produktom.

Transakciu o vykonaní rozhodnutí o úplnosti podkladov ?? považujeme za infologickú, nakoľko danému vykonávateľovi stačí vedieť, aké dokumenty sa k danému typu viažu. To má typicky popísané v nejakej dokumentácii.

ID transakcie	T05_VZ
Názov transakcie	Ocenenie zmluvy
Produkt	Zmluva je ocenená
Iniciátor	A09 – Žiadateľ ocenenia zmluvy
Vykonávateľ	A10 – Oceňovateľ zmluvy
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: storno, overenie, oceňovanie alebo podmienené prijatie
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav zmluvy nie je označený ako Zmluva ocenená
<i>Revoke request</i>	Výsledok korekcie je ocenenie a súčasne už iná transakcia pre ocenenie zmluvy prebieha.
<i>Revoke promise</i>	Zmluva nie je určená pre ocenenie
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.5: Transakcia T05 – Ocenenie zmluvy

ID transakcie	T06_VZ
Názov transakcie	Oprava zmluvy
Produkt	Zmluva je opravená
Iniciátor	A11 – Žiadateľ o opravu zmluvy
Vykonávateľ	A12 – Opravár zmluvy
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: storno, ocenenie alebo overenie
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav zmluvy nie je označený ako Zmluva opravená
<i>Revoke request</i>	–
<i>Revoke promise</i>	Zmluva nie je určená na opravu
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.6: Transakcia T06 – Oprava zmluvy

4. ANALÝZA PROCESOV

Tabuľka 4.7: Transakcia T07 – Vykonanie rozhodnutia o úplnosti podkladov

ID transakcie	T07_VZ
Názov transakcie	Vykonanie rozhodnutia o úplnosti podkladov
Produkt	Rozhodnutie o úplnosti podkladov je vykonané
Iniciátor	A13 – Žiadateľ o vykonanie rozhodnutia o úplnosti podkladov
Vykonávateľ	A14 – Rozhodovateľ o úplnosti podkladov
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: storno, overenie, vyžiadanie doplnenia alebo oceňovanie
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Chýba rozhodnutie o úplnosti podkladov
<i>Revoke request</i>	Výsledok korekcie je ocenenie a súčasne už iná transakcia v rámci fázy ocenenia zmluvy prebieha.
<i>Revoke promise</i>	Zmluva nie je určená pre rozhodovanie o úplnosti podkladov
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Doplnenie dokumentácie 4.8 považujeme tiež za infologickú transakciu, nakoľko vykonávateľ má k dispozícii z dokumentácie, ako a ktoré dokumenty má dopĺňať, prípadne o ktoré si požiadať.

Tabuľka 4.8: Transakcia T08 – Doplnenie dokumentácie

ID transakcie	T08_VZ
Názov transakcie	Doplnenie dokumentácie
Produkt	Dokumentácia je doplnená
Iniciátor	A15 – Žiadateľ o doplnenie dokumentácie
Vykonávateľ	A16 – Doplnovateľ dokumentácie
<i>Request</i>	Príchod zmluvy
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: storno, overenie, oceňovanie alebo podmienené prijatie
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	–

<i>Revoke request</i>	Výsledok korekcie je ocenenie a súčasne už iná transakcia v rámci fázy ocenenia zmluvy prebieha.
<i>Revoke promise</i>	Zmluva nie je určená pre doplnenie dokumentácie
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Transakcia o rozhodnutí výsledku podmieneného prijatia zmluvy 4.9 je ontologicky dôležitá, nakoľko vzniká produkt – rozhodnutie – či je všetko týkajúce sa zmluvy, jej ocenenia, jej opráv, jej podkladov a údajov na nich, v poriadku. Na základe toho klientovi životné poistenie vzniká alebo nevzniká. Ide o finálne rozhodnutie. Po tom, čo životné poistenie klientovi vznikne, sa môže založiť proces zmeny životného poistenia.

Tabuľka 4.9: Transakcia T09 – Rozhodnutie o výsledku podmieneného prijatia zmluvy

ID transakcie	T09_VZ
Názov transakcie	Rozhodnutie o výsledku podmieneného prijatia zmluvy
Produkt	Výsledok podmieneného prijatia zmluvy je rozhodnutý
Iniciátor	A38 – Žiadateľ o podmienené prijatie
Vykonávateľ	A39 – Rozhodovateľ o výsledku podmieneného prijatia
<i>Request</i>	Existuje zmluva pre vykonanie rozhodnutia podmieneného prijatia alebo bola taká zmluva prijatá
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Podanie rozhodnutia s jedným z nasledujúcich výsledkov: zmluva prijatá, alebo zmluva neprijatá
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Rozhodnutie o výsledku prijatia nie je označené
<i>Revoke request</i>	–
<i>Revoke promise</i>	Zmluva nie je určená pre rozhodnutie o výsledku podmieneného prijatia
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

4.3.2 Zmena životného poistenia

Prvou transakciou 4.10 poukazujeme na zahájenie spracovania procesu zmeny životného poistenia. Samotnú transakciu vynecháme v ďalších analýzach. Proces sa začína v momente, keď príde od klienta žiadosť o zmenu životného poistenia – buď v podobe papierovej, alebo elektronickej (e-mailom). Podľa spôsobu prijatia tejto žiadosti sa zvolí príslušný tím, ktorý vykonáva prvý krok a to určenie typu zmeny, ku ktorému prislúcha transakcia T02_ZM 4.11.

ID transakcie	T01_ZM
Názov transakcie	Zahájenie spracovania zmeny ŽP
Produkt	Spracovanie zmeny ŽP je zahájené
Iniciátor	A17 – Žiadateľ o spracovanie zmeny životného poistenia
Vykonávateľ	A18 – Zahajovateľ spracovania zmeny ŽP
<i>Request</i>	Príchod papierovej alebo elektronickej (e-mail) žiadosti
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Žiadosť je v poriadku a pripravená na spracovanie
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	–
<i>Revoke request</i>	–
<i>Revoke promise</i>	Žiadosť nie je určená pre spracovanie zmeny ŽP
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.10: Transakcia T01_ZM – Zahájenie spracovania zmeny ŽP

Transakcia určenia typu zmeny životného poistenia je infologická, nakoľko je účastník vie, na ktoré časti žiadosti (dokumentu) sa má dívať, na ktoré údaje, na základe ktorých určí typ. Táto jedna transakcia nám pokrýva dve možnosti, resp. nerozlišuje spôsob, akým prišla žiadosť o zmenu ŽP. Toto je jedna z výhod DEMA, kedy takýto detail ďalej špecifikujeme v *action modeli*.

Transakciu Doplnenie atribútov žiadosti 4.12 považujeme za infologickú, nakoľko podobne ako pri transakcii vyhodnotenia typu zmeny ŽP 4.11, pre vykonávateľa existuje nejaká dokumentácia, ktorá mu vraví, ktoré atribúty (číslo poistnej zmluvy, rodné číslo klienta, ID zmeny životného poistenia, atď.) majú byť pre úspešné dokončenie zmeny životného poistenia vyplnené. Do tejto transakcie sme doplnili situácie, kedy by mohol nastať *reject* alebo *revoke promise*, čo v analýze pre BPM riešenie nebolo zahrnuté.

Transakcia spracovania netechnickej (jednoduchšej alebo zložitejšej) zmeny 4.13 je dôležitá transakcia, kde sa určuje, či zmena je akceptovaná a prijatá,

ID transakcie	T02_ZM
Názov transakcie	Určenie typu zmeny ŽP
Produkt	Typ zmeny ŽP je určený
Iniciátor	A36 – Žiadateľ o určenie typu zmeny
Vykonávateľ	A37 – Určovateľ typu zmeny
<i>Request</i>	Pokus o určenie typu ako prvý krok spracovania zmeny
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Typ je určený
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Je potrebná zmena typu zmeny alebo chýba označenie, že typ zmeny bol určený
<i>Revoke request</i>	–
<i>Revoke promise</i>	Žiadosť nie je určená pre proces spracovania zmeny ŽP alebo žiadosť už má určený typ
<i>Revoke statement</i>	Je potrebná zmena typu zmeny
<i>Revoke acceptance</i>	Je potrebná zmena typu zmeny

Tabuľka 4.11: Transakcia T02_ZM – Určenie typu zmeny ŽP

alebo sa žiadosť o zmenu zamietajú. Ide v rámci procesu o najdôležitejšie rozhodnutie vykonávateľa transakcie – vzniká produkt zmeny ŽP v danom čase unikátny.

V analýze pre stávajúce BPM riešenie bol spomínaný krok odoslania informácie klientovi o vykonaní zmeny. My tento krok popisujeme datalogickou transakciou 4.14. Datalogickú úroveň sme vybrali z dôvodu, že ide o jednoduché posunutie existujúcej informácie a nie sú potrebné žiadne iné úkony.

Určenie typu do ZIS je transakcia 4.15 popisujúca infologický úkon, kedy vykonávateľ vykonáva vloženie informácií na základe žiadosti do externého systému – ZISu. BPM riešenie v rámci tejto úlohy zahŕňa i krok, kde v BPM aplikácii užívateľ musí ešte dohľadať vloženú informáciu o zmene zo ZISu a priradiť elektronickej žiadosti v BPM aplikácii ID zmeny nájdenej v ZISE. Tieto ďalšie kroky majú za následok vytvorenie ďalších infologických transakcií, ktoré normálne nie sú potrebné na to, aby bola zmena spracovaná. To môžeme vnímať ako nevýhodu stávajúceho BPM riešenia. Tieto kroky by totiž nemuseli existovať, ak by sme sa dívali na proces pomocou DEMA.

Transakciu zamietnutia zmeny 4.16 považujeme za ontologickú, nakoľko vykonávateľ nesie veľkú zodpovednosť a rozhoduje sa na základe všetkých dodaných podkladov, informácií, histórie atď. Vznikne teda produkt so zamietnutím, kde výsledok je, že žiadosť je neplatná alebo nevyhovujúca.

Vytvorenie protinávru popisujúce transakciou 4.17 je ontologické, nakoľko

4. ANALÝZA PROCESOV

ID transakcie	T03_ZM
Názov transakcie	Doplnenie atribútov žiadosti
Produkt	Doplnenie atribútov žiadosti je dokončené
Iniciátor	A19 – Žiadateľ o doplnenie atribútov žiadosti alebo zmluvy
Vykonávateľ	A20 – Doplnovateľ atribútov žiadosti alebo zmluvy
<i>Request</i>	Pokiaľ dokument neobsahuje hodnoty atribútov, vyplníme
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Hodnoty sú vyplnené
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Hodnoty už boli korektne vyplnené alebo chýba označenie, že je atribúty boli doplnené
<i>Revoke request</i>	–
<i>Revoke promise</i>	Hodnoty už boli korektne vyplnené
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.12: Transakcia T03_ZM – Doplnenie atribútov žiadosti

ID transakcie	T04_ZM
Názov transakcie	Spracovanie NTZ
Produkt	NTZ je spracovaná
Iniciátor	A21 – Žiadateľ o spracovanie NTZ
Vykonávateľ	A22 – Spracovávateľ NTZ
<i>Request</i>	Spracovanie jednoduchej a otypovanej žiadosti
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Spracovanie vykonané a zmena vykonaná
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Zmena nie je jednoduchá alebo je chybná, alebo nie je označená ako spracovaná
<i>Revoke request</i>	Žiadosť nie je otypovaná
<i>Revoke promise</i>	Zamietnutie žiadosti o zmenu
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.13: Transakcia T04_ZM – Spracovanie NTZ

ID transakcie	T05_ZM
Názov transakcie	Odoslanie informácie klienta o vykonaní zmeny
Produkt	Informácia o vykonaní zmeny klientovi odoslaná
Iniciátor	A22 – Spracovávateľ NTZ
Vykonávateľ	A23 – Informovateľ klienta o spracovaní NTZ
<i>Request</i>	Informovanie zmeny klienta po vykonaní zmeny
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Informovanie vykonané
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Odoslanie informácie klientovi sa nepodarilo
<i>Revoke request</i>	Zmena nie je vykonaná
<i>Revoke promise</i>	–
<i>Revoke statement</i>	Odoslanie informácie klientovi sa nepodarilo
<i>Revoke acceptance</i>	–

Tabuľka 4.14: Transakcia T05_ZM – Odoslanie informácie klienta o vykonaní zmeny

ID transakcie	T06_ZM
Názov transakcie	Určenie typu do ZIS
Produkt	Určenie typu do ZIS je dokončené
Iniciátor	A24 – Žiadateľ o určenie typu do ZIS
Vykonávateľ	A25 – Určovateľ typu do ZIS
<i>Request</i>	Určenie typu TZ zmeny do ZIS
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Zobrazil sa výsledok určovania typu do ZIS
<i>Accept</i>	Typ sa do ZIS podarilo určiť
<i>Decline</i>	–
<i>Reject</i>	Typ sa do ZIS určiť nepodarilo
<i>Revoke request</i>	–
<i>Revoke promise</i>	–
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.15: Transakcia T06_ZM – Určenie typu do ZIS

ide o dôležitý produkt popri produktu samotnej zmeny. Protinávrh pracovník vytvára pomocou programu Excel. Ide o dokument, ktorý okrem základných údajov obsahuje zoznam klientov, ktorých sa protinávrh týka, obsahuje riziká a čiastky, ktoré pracovník určuje. Tento protinávrh sa neskôr klientovi odosiela a čaká, až sa vráti späť. Protinávrh je v podstate návrh zmeny klientovej

ID transakcie	T07_ZM
Názov transakcie	Zamietnutie TZ
Produkt	Zamietnutie TZ je dokončené
Iniciátor	A26 – Žiadateľ o zamietnutie zmeny
Vykonávateľ	A27 – Zamietateľ zmeny
<i>Request</i>	Zamietnutie TZ v akomkoľvek bode spracovania
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Žiadosť o zmenu je neplatná alebo nevyhovujúca
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav žiadosti nie je označený ako Žiadosť zamietnutá
<i>Revoke request</i>	–
<i>Revoke promise</i>	Daná žiadosť o zmenu nie je určená pre zamietnutie
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.16: Transakcia T07_ZM – Zamietnutie TZ

žiadosti o zmenu s vysvetlením.

Podobne ako pri sme transakcii overenia zmluvy v procese vzniku životného poistenia 4.3 určili produkt vznikajúci overením, krok overenia zmeny určujeme za ontologický 4.18. Vykonávateľ sa musí na zmenu pozrieť ako na celok – na údaje na žiadosti, dodané podklady a údaje na nich, na históriu, nahliadnúť do ZISu a vyhodnotiť.

Ontologická transakcia ocenenia zmeny 4.19 predstavuje podobne ako pri oceňovaní zmluvy v procese vzniku ŽP dôležitý krok. Okrem iného, ak je potrebné zmenu nejako upraviť, tak sa vytvára návrh na protinávrh – transakcia je popisovaná tabuľkou nižšie 4.20. V tom prípade transakcia oceňovania čaká na dokončenie transakcie vytvorenia návrhu na protinávrh.

Ocenenie zmeny nevznikne v prípade, ak ju navrhujeme na zamietnutie alebo vyžiadanie dokumentácie – tento prípad popisujeme úkonom *revoke request*.

Transakcia vytvorenia návrhu na protinávrh predstavuje krok, kedy vykonávateľ na základe danej šablóny vytvorí návrh na protinávrh, ktorý sa následne pošle na tím Zmeny, ktorých pracovníci vytvárajú finálny protinávrh. Vytvorenie návrhu na protinávrh predstavuje infologickú transakciu, nakoľko užívateľ ma k dispozícii šablónu, v ktorej len pred-pripravy protinávrh.

ID transakcie	T08_ZM
Názov transakcie	Vytvorenie protinávrhu
Produkt	Protinávrh je vytvorený
Iniciátor	A28 – Žiadateľ o vytvorenie protinávrhu
Vykonávateľ	A29 – Tvorca protinávrhu
<i>Request</i>	Vytvorenie protinávrhu, keď stav spracovania TZ zmeny tomu vyhovuje
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Protinávrh je vyrobený s výsledkom Zmenu je potrebné oceniť alebo Zmenu je potrebné overiť, alebo je potrebné určiť typ zmeny znova
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	–
<i>Revoke request</i>	–
<i>Revoke promise</i>	–
<i>Revoke statement</i>	Zmena je expirovaná alebo zamietnutá
<i>Revoke acceptance</i>	–

Tabuľka 4.17: Transakcia T08_ZM – Vytvorenie protinávrhu

ID transakcie	T09_ZM
Názov transakcie	Overenie zmeny
Produkt	Zmena je overená
Iniciátor	A30 – Žiadateľ overenia zmeny
Vykonávateľ	A31 – Overovateľ zmeny
<i>Request</i>	Zmenu je potrebné overiť
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Zmena je overená a schválená (vykonaná)
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	Stav žiadosti nie je označený ako Overená
<i>Revoke request</i>	Zmenu je potrebné oceniť alebo zmeniť typ, alebo opraviť. Zmenu môžeme tiež zamietnuť
<i>Revoke promise</i>	–
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.18: Transakcia T09_ZM – Overenie zmeny

4. ANALÝZA PROCESOV

ID transakcie	T10_ZM
Názov transakcie	Ocenenie zmeny
Produkt	Zmena je ocenená
Iniciátor	A32 – Žiadateľ ocenenia zmeny
Vykonávateľ	A33 – Oceňovateľ zmeny
<i>Request</i>	Zmenu je potrebné oceniť
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Zmena je ocenená s hotovým návrhom na protinávrh
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	–
<i>Revoke request</i>	Zmenu nie je nutné oceňovať alebo zmenu zamietneme, alebo vyžiadame doplnenie dokumentácie
<i>Revoke promise</i>	–
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.19: Transakcia T10_ZM – Ocenenie zmeny

ID transakcie	T11_ZM
Názov transakcie	Vytvorenie návrhu na protinávrh
Produkt	Návrh na protinávrh je vytvorený
Iniciátor	A34 – Žiadateľ o vytvorenie návrhu na protinávrh
Vykonávateľ	A35 – Tvorca návrhu na protinávrh
<i>Request</i>	Oceňovateľ rozhodol, že je potrebné vytvoriť návrh
<i>Promise</i>	<i>Implicitný</i>
<i>State</i>	Protinávrh je vytvorený a pripravený pre tím Zmien
<i>Accept</i>	<i>Implicitný</i>
<i>Decline</i>	–
<i>Reject</i>	–
<i>Revoke request</i>	–
<i>Revoke promise</i>	–
<i>Revoke statement</i>	–
<i>Revoke acceptance</i>	–

Tabuľka 4.20: Transakcia T11_ZM – Vytvorenie návrhu na protinávrh

4.4 Role naprieč procesmi

V tejto sekcii sme zostavili zoznam všetkých rolí použitých v transakciách popisujúcich dané procesy a k nim sme priradili tímy predstavujúce základnú jednotku členenia pracovníkov 4.21. Vidíme, že rolí je niekoľko násobne viac oproti tímom, do ktorých sa pracovníci zaraďujú. Vidíme tak, že čo jedna rola, to viac menej jedna kompetencia.

Tabuľka 4.21: Mapovanie DEMO rolí k firemným tímom

DEMO rola	Tímy
A01 – Žiadateľ o spracovanie vzniku životného poistenia	Podateľňa
A02 – Spracovávateľ vzniku ŽP	Účastníci procesu Vzniku ŽP
A03 – Žiadateľ o určenie typu zmluvy	Podateľňa
A04 – Určovateľ typu zmluvy	Určovanie a Podateľňa
A05 – Žiadateľ overenia zmluvy	Určovanie, Oceňovanie, Korekcie a Zmeny
A06 – Overovateľ zmluvy	Overovanie a Zmeny
A07 – Žiadateľ storna zmluvy	Určovanie, Overovanie, Oceňovanie a Korekcie
A08 – Spracovávateľ storna zmluvy	Storno
A09 – Žiadateľ ocenenia zmluvy	Určovanie, Overovanie, Korekcie a Správca
A10 – Oceňovateľ zmluvy	Oceňovanie
A11 – Žiadateľ o opravu zmluvy	Určovanie, Overovanie a Storno
A12 – Opravár zmluvy	Korekcie
A13 – Žiadateľ o vykonanie rozhodnutia o úplnosti podkladov	Určovanie, Overovanie a Korekcie
A14 – Rozhodovateľ o úplnosti podkladov ŽP	Správca
A15 – Žiadateľ o doplnenie dokumentácie	Správca
A16 – Doplnovateľ dokumentácie	Správca
A17 – Žiadateľ o spracovanie zmeny životného poistenia	Klient
A18 – Zahajovateľ spracovania zmeny ŽP	Podateľňa a OBO
A19 – Žiadateľ o doplnenie atribútov žiadosti alebo zmluvy	OBO
A20 – Doplnovateľ atribútov žiadosti alebo zmluvy	OBO
A21 – Žiadateľ o spracovanie NTZ	OBO, Podateľňa a Zmeny (zložitá NTZ)

4. ANALÝZA PROCESOV

A22 – Spracovávateľ NTZ	OBO a Zmeny (zložitá NTZ)
A23 – Informovateľ klienta o spracovaní NTZ	OBO a Zmeny (zložitá NTZ), no datalogická
A24 – Žiadateľ o určenie typu do ZIS	Zmeny
A25 – Určovateľ typu do ZIS	Zmeny
A26 – Žiadateľ o zamietnutie zmeny	Zmeny
A27 – Zamietateľ zmeny	Zmeny
A28 – Žiadateľ o vytvorenie protinávrhu	Zmeny
A29 – Tvorca protinávrhu	Zmeny
A30 – Žiadateľ overenia zmeny	Zmeny
A31 – Overovateľ zmeny	Zmeny
A32 – Žiadateľ ocenenia zmeny	Správca
A33 – Oceňovateľ zmeny	Oceňovanie zmien
A34 – Žiadateľ o vytvorenie návrhu na protinávrh	Oceňovanie zmien
A35 – Tvorca návrhu na protinávrh	Oceňovanie zmien
A36 – Žiadateľ o určenie typu zmeny	Podateľňa pre papierovú formu, OBO pre e-mail
A37 – Určovateľ typu zmeny	Podateľňa pre papierovú formu, OBO pre e-mail
A38 – Žiadateľ o podmienené prijatie	Oceňovanie zmien a Správca
A39 – Rozhodovateľ o výsledku podmieneného prijatia	Správca

Pozreli sme sa na mapovanie aj z opačnej strany a priradili sme firemným tímom role. Z takejto tabuľky môžeme identifikovať počet rolí, resp. i kompetencií, ktorými musí subjekt (pracovník) disponovať, aby sa mohol v danom tíme nachádzať. Môžeme identifikovať nevyváženosť medzi tímami. Vidíme, že počet rolí, ktoré zastáva tím Zmeny, je oproti tímu Storno príliš vysoký. Nemusí to hneď znamenať nevyváženosť, ale i dôsledok kompetencie alebo náročnosti daných úkonov. Avšak tím s príliš veľa kompetenciami alebo rolami môže znamenať, že na tím je kladená veľká záťaž a bolo by ho napr. rozčleniť na viac tímov. Vidíme, že napr. tím Oceňovanie zahrňuje len tri role, avšak pracovníci z tímu Oceňovanie vykonávajú ontologickú transakciu oceňovania zmluvy, ktorá je ontologická a vyžaduje sa od pracovníka jeho spôsobilý posudok.

Tabuľka 4.22: Mapovanie firemných tímov k rolám

DEMO rola	Tímy
Korekcie	A05 – Žiadateľ overenia zmluvy, A07 – Žiadateľ storna zmluvy, A09 – Žiadateľ ocenenia zmluvy, A12 – Opravár zmluvy, A13 – Žiadateľ o vykonanie rozhodnutia o úplnosti podkladov
OBO	A18 – Zahajovateľ spracovania zmeny ŽP, A19 – Žiadateľ o doplnenie atribútov žiadosti alebo zmluvy, A20 – Doplnovateľ atribútov žiadosti alebo zmluvy, A21 – Žiadateľ o spracovanie NTZ, A22 – Spracovávateľ NTZ, A23 – Informovateľ klienta o spracovaní NTZ, A36 – Žiadateľ o určenie typu zmeny, A37 – Určovateľ typu zmeny
Oceňovanie	A05 – Žiadateľ overenia zmluvy, A07 – Žiadateľ storna zmluvy, A10 – Oceňovateľ zmluvy
Oceňovanie Zmien	A33 – Oceňovateľ zmeny, A34 – Žiadateľ o vytvorenie návrhu na protinávrh, A35 – Tvorca návrhu na protinávrh, A38 – Žiadateľ o podmienené prijatie
Overovanie	A06 – Overovateľ zmluvy, A07 – Žiadateľ storna zmluvy, A09 – Žiadateľ ocenenia zmluvy, A11 – Žiadateľ o opravu zmluvy, A13 – Žiadateľ o vykonanie rozhodnutia o úplnosti podkladov

4. ANALÝZA PROCESOV

Podateľňa	A01 – Žiadateľ o spracovanie vzniku životného poistenia, A03 – Žiadateľ o určenie typu zmluvy, A04 – Určovateľ typu zmluvy, A18 – Zahajovateľ spracovania zmeny ŽP, A21 – Žiadateľ o spracovanie NTZ, A36 – Žiadateľ o určenie typu zmeny, A37 – Určovateľ typu zmeny
Správca	A09 – Žiadateľ ocenenia zmluvy, A14 – Rozhodovateľ o úplnosti podkladov ŽP, A15 – Žiadateľ o doplnenie dokumentácie, A16 – Doplnovateľ dokumentácie, A32 – Žiadateľ ocenenia zmeny, A38 – Žiadateľ o podmienené prijatie, A39 – Rozhodovateľ o výsledku podmieneného prijatia
Storno	A08 – Spracovávateľ storna zmluvy, A11 – Žiadateľ o opravu zmluvy
Určovanie	A04 – Určovateľ typu zmluvy, A05 – Žiadateľ overenia zmluvy, A07 – Žiadateľ storna zmluvy, A09 – Žiadateľ ocenenia zmluvy, A11 – Žiadateľ o opravu zmluvy, A13 – Žiadateľ o vykonanie rozhodnutia o úplnosti podkladov
Zmeny	A05 – Žiadateľ overenia zmluvy, A06 – Overovateľ zmluvy, A21 – Žiadateľ o spracovanie NTZ, A22 – Spracovávateľ NTZ, A23 – Informovateľ klienta o spracovaní NTZ A24 – Žiadateľ o určenie typu do ZIS, A25 – Určovateľ typu do ZIS, A26 – Žiadateľ o zamietnutie zmeny, A27 – Zamietateľ zmeny, A28 – Žiadateľ o vytvorenie protinávruhu, A29 – Tvorca protinávruhu, A30 – Žiadateľ overenia zmeny, A31 – Overovateľ zmeny

4.5 Transaction Product Table (TPT)

V nasledujúcej tabuľke 4.23 vidíme súhrn transakcií, ich prislúchajúce produkty a farbou znázornené ich organizačné úrovne. Pozorujeme, že zo základnej analýzy procesov sme identifikovali (nepočítajúc prvé ilustratívne transakcie z oboch procesov) 4 ontologické transakcie z procesu vzniku životného

poistenia a 4 infologické transakcie z celkových. Z procesu zmien životného poistenia sme z analýzy pre BPM riešenie identifikovali 5 ontologických, 4 infologické a jedno datalogickú transakciu.

Pri prechádzaní spísanej analýzy sme brali do úvahy podstatné informácie, podstatné transakcie, no dané počty z vyššieho odseku môžu znamenať, že analýza pre BPM nemusela byť kompletná. Očividne, v analýze chýbajú niektoré infologické transakcie a datalogických transakcií bude chýbať ešte viac. DEMO nám pomáha tieto transakcie objaviť. Ďalšie si uvedomujúce transakcie zobrazíme v ďalšej sekcii s niekoľkými *construction* modelmi.

Tabuľka 4.23: Transaction Product Table

Transaction kind	Product kind
T01_VZ – Spracovanie vzniku životného poistenia	P01_VZ – Vznik životného poistenia je spracovaný
T02_VZ – Určenie typu zmluvy	P02_VZ – Typ zmluvy je určený
T03_VZ – Overenie zmluvy vzniku životného poistenia	P03_VZ – Zmluva vzniku životného poistenia je overená
T04_VZ – Podanie návrhu na storno	P04_VZ – Návrh na storno je podaný
T05_VZ – Ocenenie zmluvy	P05_VZ – Zmluva je ocenená
T06_VZ – Oprava zmluvy	P06_VZ – Zmluva je opravená
T07_VZ – Fáza oceňovania – Vykonanie rozhodnutia o úplnosti podkladov	P07_VZ – Rozhodnutie o úplnosti podkladov je vykonané
T08_VZ – Fáza oceňovania – Doplnenie dokumentácie	P08_VZ – Dokumentácia je doplnená
T09_VZ – Rozhodnutie o výsledku podmieneného prijatia zmluvy	P09_VZ – Výsledok podmieneného prijatia zmluvy je rozhodnutý
T01_ZM – Zahájenie spracovania zmeny ŽP	P01_ZM – Spracovanie Zahájenie spracovania zmeny ŽP zmeny ŽP je zahájené
T02_ZM – Určenie typu zmeny ŽP	P02_ZM – Typ zmeny ŽP je určený
T03_ZM – Doplnenie atribútov žiadosti	P03_ZM – Doplnenie atribútov žiadosti je dokončené
T04_ZM – Spracovanie NTZ	P04_ZM – NTZ je spracovaná
T05_ZM – Odoslanie informácie klientovi o vykonaní zmeny	P05_ZM – Informácia o vykonaní zmeny klientovi odoslaná
T06_ZM – Určenie typu do ZIS	P06_ZM – Typ do ZIS je určený je dokončené
T07_ZM – Zamietnutie TZ	P07_ZM – Zamietnutie TZ je dokončené

T08_ZM – Vytvorenie protinávru	P08_ZM – Protinávvrh je vytvorený
T09_ZM – Overenie zmeny	P09_ZM – Zmena je overená
T10_ZM – Ocenenie zmeny	P10_ZM – Zmena je ocenená
T11_ZM – Vytvorenie návrhu na protinávvrh	P11_ZM – Návrh na protinávvrh je vytvorený

4.6 Construction model (CM)

V tejto sekcii si popíšeme pár spôsobov využitých pri modelovaní a predstavíme si samotné *construction* modely popisujúce niekoľko transakcií. Doteraz vymenované transakcie, ktoré boli vytvorené na základe analýzy pre stávajúce BPM riešenie, tím pádom rozšírime o ďalšie, ktoré sa často v analýze nevyskytujú kvôli i nedostatočnej metodike.

4.6.1 Spôsoby modelovania

Model I-organizácie obsahuje i infologické role účastníkov, ktorý podporuje role O-účastníkov v B-organizácii. V ψ teórii sme sa mohli dozvedieť, že subjekt s nejakou účastníckou rolou vykonáva svoj produkčný akt použitím jedným z troch schopností (*performa*, *informa* alebo *forma*) a že je schopný presunúť svoju schopnosť na inú. V konečnom dôsledku tak O-účastník, ktorý iniciuje infologickú transakciu, môže tiež interpretovať dokumenty, formulovať fakty, vykonávať logické operácie v rámci *action rules* v jeho *informa* schopnosti.

C-fakty a P-fakty sú vytvárané O-účastníkmi a sú pamätané I-účastníkmi. Pamätaním faktu máme na mysli infologický produkčný úkon, ktorý nám slúži pre získanie náležitého faktu. O-účastníka v modele značíme ako O-A_i a v rámci hraníc B-organizácie vytvára nové P-fakt a C-fakty. Tieto fakty sú zapamätané I-účastníkom I-A_{ij}[0,1], kde i je číslo účastníckej role, j je číslo typu služby, 0 hovorí, že ide o C-fakt a 1, že ide o P-fakt.

Ďalej si popíšeme spôsob, pri ktorom využívame agendu. Agendou máme na mysli situáciu, kedy na C-fakt vytvorený v nejaký čas nejakým účastníkom A_j musí účastník A_i reagovať. Terminológiou B-organizácie a I-organizácie, O-A_j vytvára C-fakty. Tieto fakty sú pamätané účastníkom I-A_j10 a sú zdieľané pomocou zdieľača agendy I-A0020. Ak sa účastník O-A_i pýta, či existuje pre neho nejaká agenda, má sa pýtať samotnej agendy I-A0020.

4.6.2 Vznik ŽP

Predstavme si *construction* model 4.6 pre transakciu určenia typu zmluvy procesu vzniku ŽP 4.2. Iniciátor v skutočnosti môže iniciovať dve infologické

transakcie. Prvou je označenie zmluvy stavom (I-T0210) pre potrebu určenia typu. Pri tomto označení sa ďalej vykoná aktualizácia stavu zmluvy (dokumentu) do informačného systému, čo odpovedá datalogickej transakcii D-T0210.

Ďalej môžu nastať dve varianty. Prvou je, že Žiadateľ o určenie typu iniciuje transakciu I-T02 (z analýzy označovanú ako T02_VZ), čo donúti vykonávateľa učiť typ zmluvy. Druhou možnosťou je, že Určovateľ typu iniciuje sám transakciu I-T0020 a pomocou zdieľania agendy sa sám „spýta“, či daná zmluva so stavom „Je potreba určiť typ“, existuje. Zdieľač agendy iniciuje infologickú transakciu I-T0230, čo spôsobí načítanie zmluvy (dokumentu) z informačného systému ďalšou datalogickou transakciou D-T0230. Tento zdieľač agendy slúži pre celú I-organizáciu.

Keď určovateľ typu zmluvy dokončí určovanie typu zmluvy, spustí infologickú transakciu I-T0211, ktorá zas analogicky vyvolá datalogickú transakciu pre uloženie zmluvy ako dokumentu do informačného systému. Daná zmluva už bude mať stav Typ určený.

Všimnime si tiež datalogickú transakciu D-T10, s ktorou sa v analýze pre BPM riešenie ani nepočíta a tak proces nie je kompletne zmapovaný.

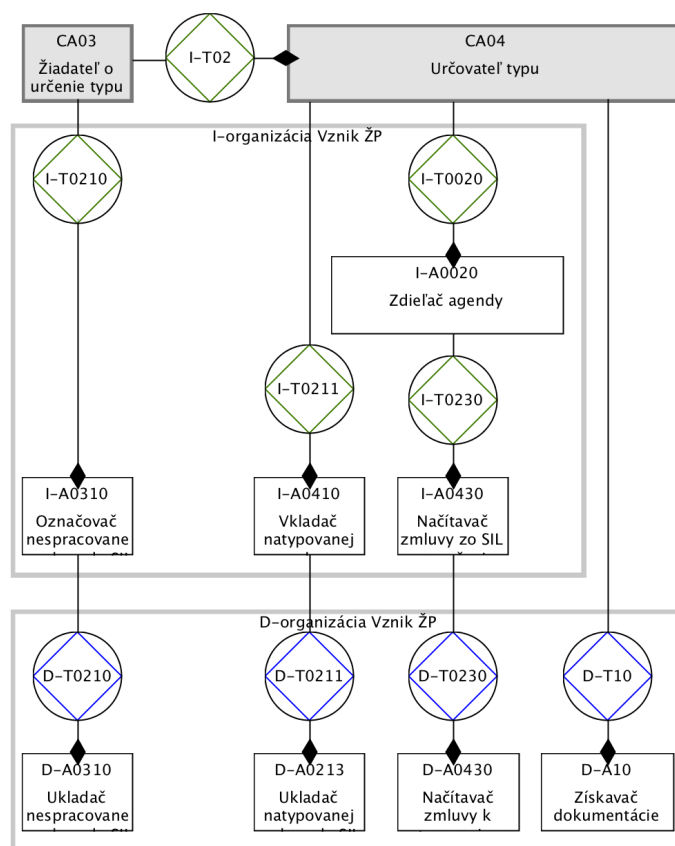
Konštrukčný model pre transakciu T03 (overenie zmluvy) 4.7 sa trochu podobá na predošlý model s určením typu zmluvy. Okrem toho, že iniciátor a vykonávateľ vystupuje ako B-účastník, nám navyše pribudla transakcia pre zdieľanie P-faktov (podobne ako transakcia pre zdieľanie agendy) a *transaction bank* AT03, do ktorej môžu pristupovať obaja B-účastníci pre prípad získania minulých faktov. Okrem *transaction banky* AT03 môže overovateľ využiť i fakty a informácie z ostatných transakcií a to z AT05 a AT06 (ocenenie zmluvy a overenie zmluvy).

4.6.3 Zmena ŽP

Popíšme si ďalej konštrukčné modely pre proces zmeny životného poistenia. Svojou podstatou sú podobné tým modelom pre proces vzniku.

Konštrukčný model zahrňujúci transakcie určenia typu zmeny a doplnenia atribútov žiadosti 4.8 odhaľuje ďalšie transakcie zahrnuté pri týchto činnostiach. Ide hlavne o infologické transakcie, pri ktorých sa zmluva označuje k určeniu, alebo k doplneniu, alebo sa žiadosť s daným stavom načítava, alebo sa s vykonanými potrebami vkladá do SILu. Pod nimi vidíme datalogické transakcie vykonávajúce samotné načítanie, resp. vloženie zo, resp. do informačného systému – SILu. V strede konštrukčného diagramu máme ako pri procese vzniku ŽP zdieľač agendy, ktorý slúži pre získanie informácie, či sa nejaká zmena k určeniu typu, resp. k doplneniu atribútov, nachádza.

V modeli taktiež vidíme dve datalogické transakcie slúžiace pre získanie dokumentácie alebo podkladov, na podľa ktorých tak vykonávatelia vykonávajú určenie typu, resp. doplnenie atribútov. S týmito transakciami BPM riešenie tiež nepočíta.

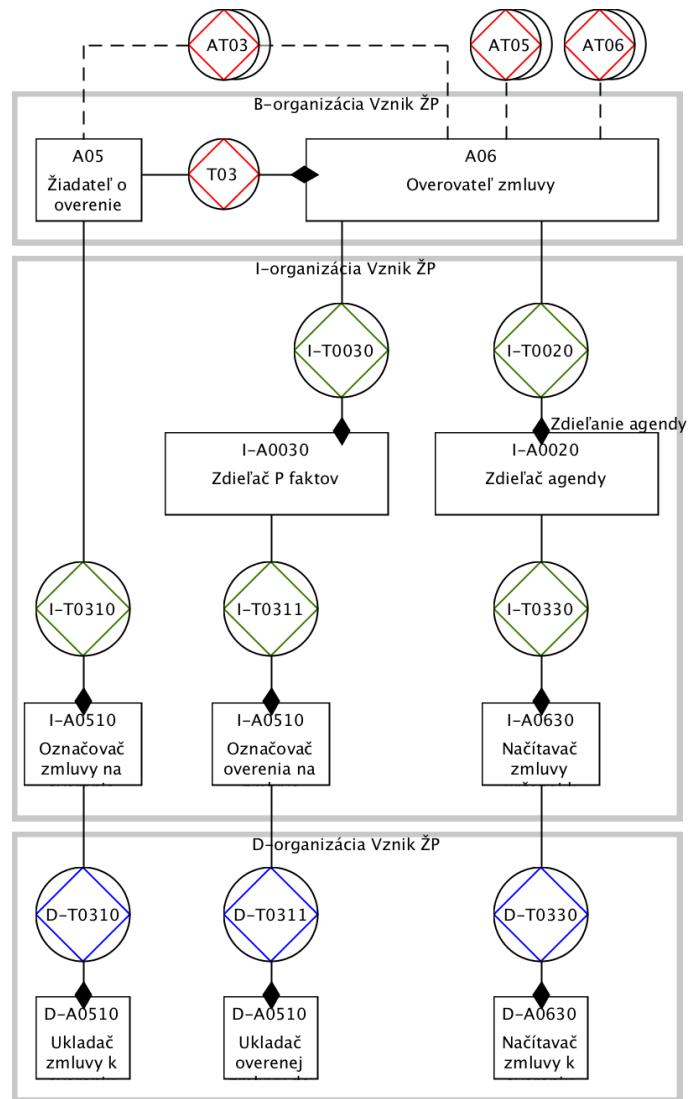


Obr. 4.6: CM určenia typu vzniku ŽP

Konštrukčný model transakcie spracovania NTZ zmeny 4.9 je stavbou takmer totožný s konštrukčným modelom overenia zmluvy vzniku životného poisťenia 4.7. Navyše pozorujeme *transaction bank* AT03, čo znamená, že spracovávateľ zmluvy a žiadateľ v určitých prípadoch majú otázku jeden na druhého – napr. v prípade, že spracovávateľ sa spýta žiadateľa na nejaké vysvetlenie niektorých údajov v žiadosti o zmenu. Spracovávateľa tiež môžu zaujímať fakty z (ak boli vykonané) ostatných transakcií ako T07 – zamietnutie TZ (k tomu slúži *transaction bank* AT07), T08 – vytvorenie protinávhrhu (*transaction bank* AT08), T09 – overenie zmeny (*transaction bank* AT09) a T10 – ocenenie zmeny (*transaction bank* AT10).

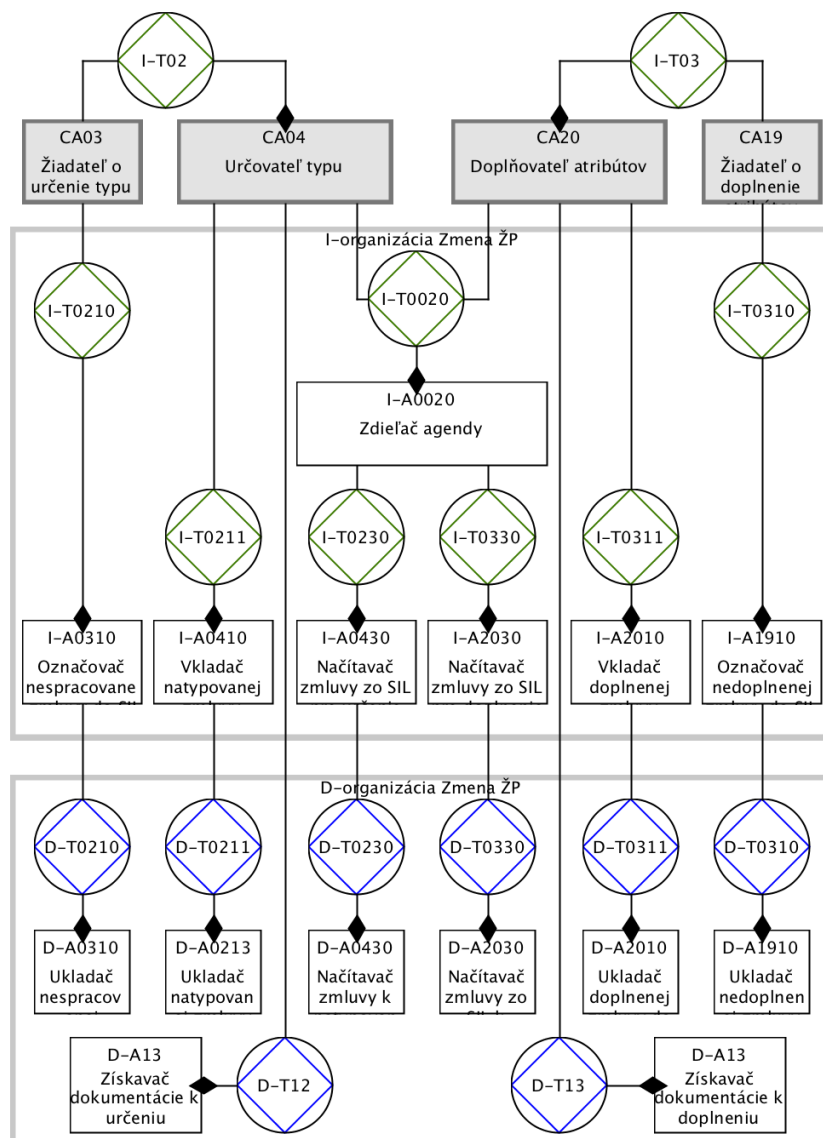
4.7 Process Model (PM)

Process model reprezentovaný *process structure diagram* obsahuje špecifický transakčný vzor pre každý transakčný typ konštrukčného modelu. Zobrazuje klasické a podmienené vzťahy medzi transakciami definovanými konštrukč-

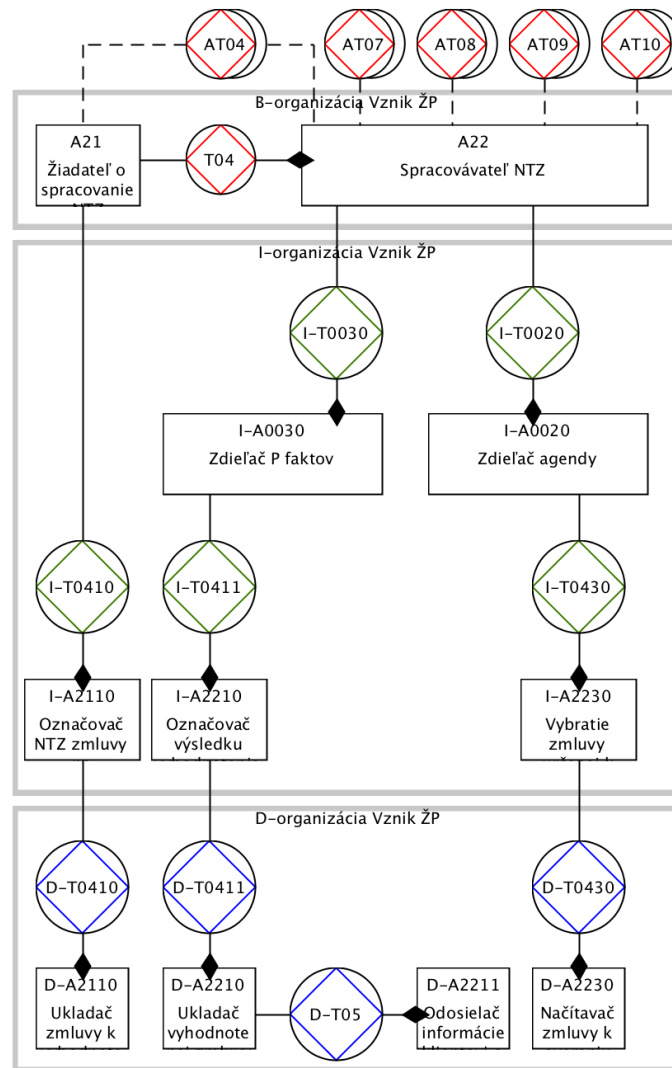


Obr. 4.7: CM určenia typu vzniku ŽP

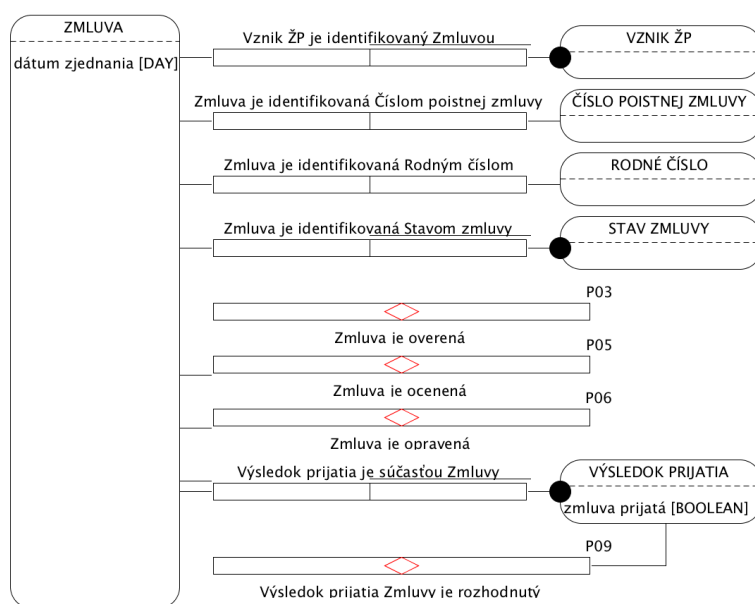
4. ANALÝZA PROCESOV



Obr. 4.8: CM určovania typu zmeny a doplňovania atribútov



Obr. 4.9: CM spracovania netechnickej zmeny



Obr. 4.10: FM procesu vzniku ŽP

ným modelom. Procesný model predstavuje špecifikáciu stavového priestoru a prechodového priestoru C-sveta. Jeho syntax nájdeme v prílohe B.

V našich procesoch čo sa týka ontologických transakcií sme priamo nenašli príklad napr. vložených ontologických transakcií, ktorú sme si popisovali v prvej kapitole – viď obrázok 2.9. Podobne ontologické transakcie pri daných procesoch majú charakter, že z každej ontologickej transakcie sa môžeme dostať do inej a to ľubovoľne krát iterovať. Ak bol charakter definície procesu sekvenčný, tvorba PSD by mala väčší zmysel.

4.8 Fact Model (FM)

Fact model organizácie predstavuje ontologický model stavového a prechodového priestoru produkčného sveta. Čo sa týka stavového priestoru, FM obsahuje všetky identifikované druhy faktov (deklarované a odvodené) a existenčné pravidlá (*existence laws*). Existujú tri druhy existenčných pravidiel: referenčné pravidlá, pravidlá jednoznačnosti a zvyšné špecifikované textovo. FM obsahuje tiež výsledky transakcií. FM je typicky reprezentovaný ODF (*Object Fact Diagram*).

4.8.1 Vznik ŽP

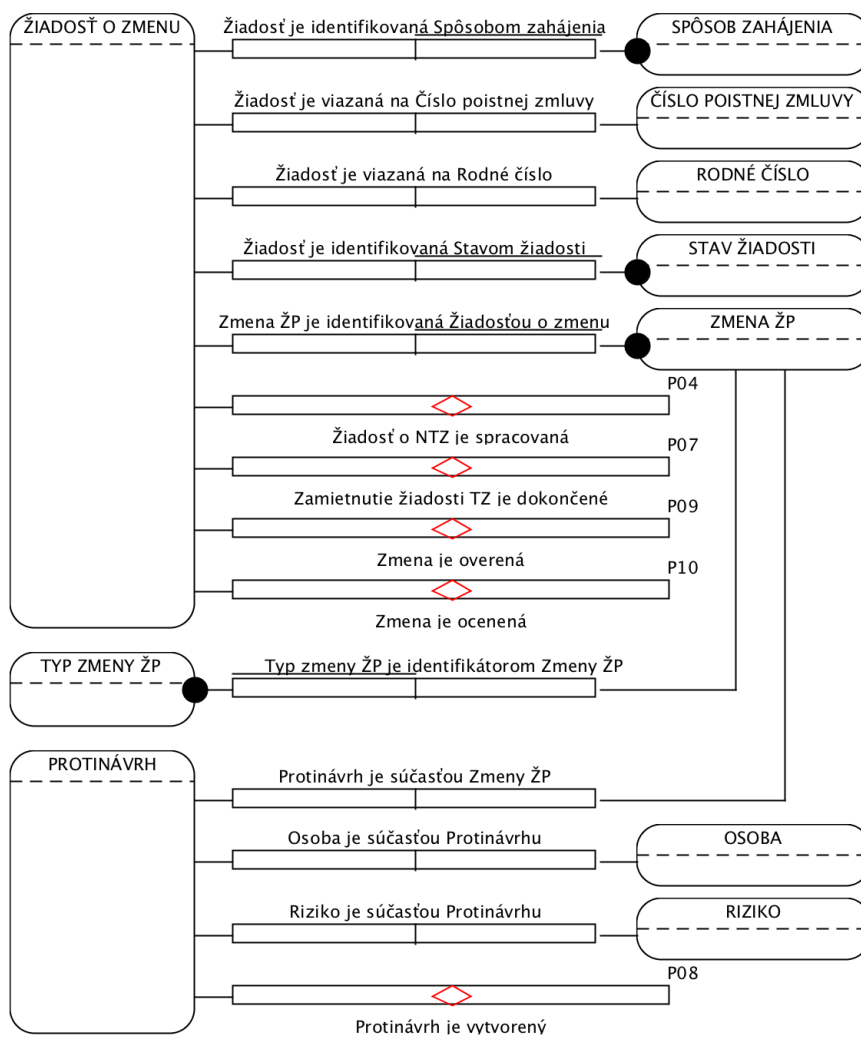
Predstavme si FM procesu vzniku ŽP 4.10. Základnou triedou objektov, ktorá sa podriaďuje príslušnému druhu faktu, je ZMLUVA. ZMLUVA má jedno-

značne určený dátum zjednania (tu sa uplatňuje *unicity law* – pravidlo jednoznačnosti). VZNIK ŽP je jednoznačne určený triedou ZMLUVA – vznik ako produkt je určený práve jednou zmluvou (na základe práve jednej zmluvy). Podobne riešime stav zmluvy, ktorý má zmluva práve jeden. Ďalej je ale dovolené, že zmluva môže obsahovať viac rodných čísel alebo čísel poistných zmlúv (čo ale typicky nebýva). Na zmluvu ďalej napájame jednotlivé ontologické transakcie, ktoré zmluvu posúvajú do iného stavu.

4.8.2 Zmena ŽP

Podobne ako pri procese vzniku ŽP sme vytvorili FM pre proces zmeny ŽP 4.11. Tu je zas základnou triedou objektov ŽIADOSŤ O ZMENU, ktorá má jednoznačný a práva jeden spôsob zahájenia, stav žiadosti, odpovedajúcej zmene ŽP. Žiadosť o zmenu sa môže týkať viacero rodných čísel a mať väzbu na viacero čísel poistných zmlúv. Trieda objektov ZMENA ŽP má zas jednoznačne určený jeden typ zmeny ŽP, no môže sa na ňu viazať viacero protinávrhov, ktorý obsahuje ľubovoľné množstvo osôb a rizík.

4. ANALÝZA PROCESOV



Obr. 4.11: FM procesu zmena ŽP

4.9 Action Model (AM)

Action model organizácie pozostáva z množiny tzv. *action rules* – z množiny pravidiel popisujúcich jednotlivé úkony. Pre každý druh úkonov a pre každú internú účastnícku rolu existuje jedno úkonové pravidlo (*action rule*). Úkonové pravidlo špecifikuje úkony, ktoré musia byť vykonané a musí byť preverená absencia alebo existencia potrebných stavov koordinačného, resp. produkčného sveta.

Úkonový model slúži najmä ako popis toho, čo má daný účastník robiť – syntax nemusí byť pevne daná a model je tým pre vykonávateľov úkonov zrozumiteľný. V našich príkladoch nebudeme uvádzať úkony pre datalogické transakcie.

4.9.1 Vznik ŽP

Uvedme si *action model* pre účastnícku rolu A05 (žiadateľ overenia zmluvy) a úkon požiadavku (*request*) transakcie T03 (overenie zmluvy vzniku životného poistenia) 4.24. Pri požiadaní overujeme, že vykonávateľ požiadavku (úkonu *request*, nie celej transakcie) je skutočne žiadateľom o overenie. Týmto vykonáme *de-facto* autorizáciu. Iné podmienky splnené byť nemusia – pri ďalšom úkonovom modeli. V poslednej časti úkonového modelu píšeme, čo sa má ďalej stať. V prípade, že je všetko v poriadku, vykonáme v našom prípade *promise* v rámci transakcii T03 pre danú zmluvu. V prípade, že podmienky splnené nie su, tak vykonáme úkon *decline*.

Tabuľka 4.24: Action model - A05 - T03/rq

keď	overenie zmluvy pre Zmluvu je požiadané
posúď	<i>oprávnenia:</i> daný vykonávateľ daného požiadavku je Žiadateľ o overenie <i>úprímnosť:</i> < bez podmienky > <i>pravdivosť:</i> < bez podmienky >
ak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i>
potom	<u>sľubu</u> overenie zmluvy pre Zmluvu
inak	<u>decline</u> overenie zmluvy pre Zmluvu

V prípade, že je vykonaný úkon *promise*, vykonávateľ transakcie sa riadi nasledujúcim úkonovým modelom 4.25. V časti oprávnenia vykonávame znova autorizáciu ako pri vykonaní požiadavku. V časti „pravdivosť“ sme zakomponovali zavolanie infologickej transakcie pre získanie agendy. V prípade, že sa všetko podarí, tak v poslednej časti postupne voláme úkony *execute*, *state* transakcie T03 a nakoniec vykonáme *request* pre zápis overenej zmluvy so stavom overená, pričom do P-sveta I-organizácie zapíšeme (zazdieľame) P-fakt o overenej zmluve.

4. ANALÝZA PROCESOV

Tabuľka 4.25: Action model - A06 - T03/pm

keď	overenie zmluvy pre Zmluvu je slúbené
posúdiť	<i>oprávnenia:</i> daný vykonávateľ daného <u>slubu</u> je Overovateľ zmluvy <i>úprimnosť:</i> < bez podmienky > <i>pravdivosť:</i> <u>request</u> zdieľanie agendy pre existenciu Zmluva s daným stavom zmluvy typu Stav zmluvy rovným „Na overenie“ keď zdieľanie agendy pre existenciu Zmluvy je <u>podané</u> <u>accept</u> zdieľanie agendy pre existenciu Zmluvy
ak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i>
potom	<u>execute</u> overenie zmluvy pre Zmluvu <u>state</u> overenie zmluvy pre Zmluvu <u>request</u> zdieľanie P-faktov pre Zmluvu s daným stavom zmluvy typu Stav zmluvy je „Overená“ <u>request</u> vloženie overenej zmluvy do SIL pre Zmluvu s daným stavom zmluvy typu Stav zmluvy je „Overená“ keď vloženie overenej zmluvy do SIL pre Zmluvu <u>je podané</u> <u>accept</u> vloženie overenej zmluvy do SIL pre Zmluvu keď zdieľanie P-faktov pre Zmluvu je <u>podané</u> <u>accept</u> zdieľanie P-faktov pre Zmluvu

Pri predávaní overenej zmluvy 4.26 okrem autorizácie vykonávame ešte kontrolu formátu, resp. existenciu jednotlivých položiek na zmluve ako je dátum zjednania, rodné číslo a číslo poisťnej zmluvy. Ak výsledok kontroly (nazývame ju pravdivosťou) je pozitívny, vykonáme úkon *accept* transakcie T03.

Tabuľka 4.26: Action model - A06 - T03/st

keď	overenie zmluvy pre Zmluvu <u>is</u> <u>podané</u> s daným dátumom zjednania nejakej Zmluvy je typu Day dané rodné číslo typu Rodné číslo predstavuje Number dané číslo poisťnej zmluvy typu Číslo poisťnej zmluvy predstavuje Number
posúdiť	<i>oprávnenia:</i> daný <u>vykonávateľ</u> aktu <u>state</u> je Overovateľom zmluvy <i>úprimnosť:</i> < bez podmienky > <i>pravdivosť:</i> dĺžka daného rodného číslo je desať cifier dané rodné číslo je deliteľné číslom 11 dĺžka daného čísla poisťnej zmluvy je desať cifier prvým znakom daného čísla poisťnej zmluvy je číslo 1
ak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i>
potom	<u>accept</u> overenie zmluvy pre Zmluvu

4.9.2 Zmena ŽP

Action model úkonu *promise* transakcie T04 procesu zmena ŽP týkajúca sa roli A22 (Spracovávateľa NTZ) 4.27 je svojou podstatou podobná ako pri popisovaní úkonu *promise* transakcie overenia zmluvy v procese vzniku ŽP 4.25. Čo je pre spracovanie NTZ dôležité, že v poslednej časti rozlišujeme spôsob zahájenia procesu zmeny ŽP. Na základe spôsobu, tak špecifikujeme, ktorý tím bude samotný úkon *execute* vykonávať. S porovnaním s BPM riešením tu tak nepotrebujeme dva objekty *User task*. Pre ontologickú časť procesu toto rozlišovať nepotrebujeme a stačí to vydefinovať tu v *action* modeli.

Tabulka 4.27: Action model - A22 - T04/pm

keď	spracovanie NTZ pre Žiadosť o zmenu je slúbené
posúď	<i>oprávnenia:</i> daný vykonávateľ <u>slubu</u> je Spracovávateľ NTZ <i>úprimnosť:</i> < bez podmienky > <i>pravdivosť:</i> <u>request</u> zdieľanie agendy pre existenciu Žiadosti o zmenu s daným typom zmeny ŽP typu Typ zmeny je „NTZ“ a s daným stavom žiadosti typu Stav žiadosti rovným „Na spracovanie“ keď zdieľanie agendy pre existenciu Žiadosti o zmenu je <u>podané</u> <u>accept</u> zdieľanie agendy pre existenciu Žiadosti o zmenu
ak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i> a daný spôsob zahájenia typu Spôsob zahájenia je „Papierový“
potom	<u>execute</u> spracovanie NTZ pre Žiadosť o zmenu tímom Podateľňa
inak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i> a daný spôsob zahájenia typu Spôsob zahájenia je „Elektronický“
potom	<u>execute</u> spracovanie NTZ pre Žiadosť o zmenu tímom OBO <u>state</u> spracovanie NTZ pre Žiadosť o zmenu <u>request</u> zdieľanie P-faktov pre Žiadosť o zmenu s daným stavom žiadosti typu Stav žiadosti rovným „Spracovaná“ <u>request</u> označenie výsledku spracovania žiadosti o zmenu pre Žiadosť o zmenu s daným stav žiadosti typu Stav žiadosti rovným „Spracovaná“ keď označenie výsledku spracovania žiadosti o zmenu pre Žiadosť o zmenu je <u>podaná</u> <u>accept</u> označenie výsledku spracovania žiadosti o zmenu pre Žiadosť o zmenu keď zdieľanie P-faktov pre Žiadosť o zmenu je <u>podané</u> <u>accept</u> zdieľanie P-faktov pre Žiadosť o zmenu

Uvedme si úkonový model aj pre úkon *state* transakcie spracovania NTZ 4.28. Pri tomto úkone vykonávame o.i. validáciu formátu a existencie požadovaných atribútov žiadosti.

Tabuľka 4.28: Action model - A22 - T04/st

keď	spracovanie NTZ pre Žiadosť o zmenu je podané s daným rodným číslom typu Rodné číslo predstavuje Number daným číslom poistnej zmluvy typu Číslo poistnej zmluvy predstavuje Number daným stavom žiadosti typu Stav žiadosti predstavuje Nominal
posúď	<i>oprávnenia:</i> daný vykonávateľ úkonu <u>state</u> je Spracovávateľ NTZ <i>úprimnosť:</i> < bez podmienky > <i>pravdivosť:</i> dĺžka daného rodného čísla je desať cifier dané rodné číslo deliteľné číslom 11 dĺžka daného čísla poistnej zmluvy je desať cifier daný stav žiadosti je rovný „Spracovaná“ prvý znak daného čísla poistnej zmluvy je číslo 1
ak	<i>v súlade s tvrdeniami považujeme za odôvodniteľné</i>
potom	accept spracovanie NTZ pre Žiadosť o zmenu

4.10 Storno životného poistenia

Proces Storno životného poistenia nám najviac pripomína proces zmeny ŽP. Podobne na začiatku procesu dochádza k identifikácii typu storna – rozlišujeme typ storno s odbytným a typ storno bez odbytného. Pokiaľ ide o storno bez odbytného, nasleduje samotné spracovanie storna, čo je podobne ako pri procese zmeny a úlohe Spracuj NTZ, ontologická transakcia. Čo sa týka storna s odbytným, v rámci neho existujú hlavné dve ontologické transakcie ukončenia likvidácie storna a nasledovné overenie. Oceňovacia časť v sa prípade storna nevyskytuje, tým pádom ide o jednoduchší proces.

V rámci daného procesu sa vyskytujú podobné infologické transakcie ako pri procese zmeny životného poistenia a to najmä doplnenie dokumentácie, resp. ďalších podkladov k spracovaniu.

Čo v procese storna ŽP pribudlo, je, že storno počas spracovania môžeme kedykoľvek odvolať. Tým pádom vykonávame úkony typu napr. *revoke request* – odvolávame požiadavky. Ďalší rozdiel je, že po dokončení úlohy spracovania storna s odbytného a úlohy verifikácie storna s odbytným, sa čaká na vykonanie storna definované konkrétnym dátumom. Toto čakanie by sme mohli popísať ako datalogickú transakciu Čakanie na vykonanie storna.

Charakter procesu, doťahovanie jednotlivých dokumentov z predošle spomínaných systémov je rovnaké ako pri procese zmeny ŽP. Tým pádom modely, role, popisy jednotlivých akcií pre jednotlivé role a pozorovania by boli podobného charakteru.

Vyhodnotenia

V predošlých kapitolách sme si predstavili disciplínu *enterprise engineering* s náležitými teóriami, metódu DEMO s následnou aplikáciou. V tejto kapitole vyhodnotíme poznatky z aplikácie DEMA, navrhujeme možné zlepšenia, popíšeme nedostatky aktuálneho BPM riešenia a prediskutujeme ďalšie možné kroky, ktoré by mohli na túto prácu nadväzovať.

5.1 Výhody metódy DEMO

Hlavná výhoda DEMA oproti iným populárnym spôsobom modelovania spočíva v zvládnutí komplexity súčasných podnikov a podnikových zmien [15]. To sa dá zvládnuť redukciami veľkostí modelov so súčasným zameraním na esenciu podniku, ktorú získame odstránením technológie a zo všetkých infologických a datologických transakcií [16].

Podľa Landa a Dietza [16] v kapitole „Benefits of Enterprise Ontology in Governing Complex Enterprise Transformations“ transakčný a úrovňový axióm v ψ teórii ponúka každý aspoň 70% redukciiu komplexity konštrukčného modelu v DEMA. Zameraním sa len na ontologickú úroveň a začlenením ontologických transakcií, kde používame vždy rovnakú postupnosť krokov (transakčný vzor), dosiahneme i viac než 90% redukciiu veľkosti modelu pričom zobrazená ontologická esencia podniku obsahuje všetky relevantné elementy.

So silným teoretickým základom DEMO zachytáva sociálnu interakciu medzi ľudskými bytosťami pričom následne vytvorený ontologický model je koherentný, komprehenzívny, konzistentný, koncízny a esenciálny [7]. Prakticky to znamená, že celá esenciálna znalosť o organizácii, jej výkonov a konštrukcii, je zahrnutá v DEMO modeli. Na základe popisu organizácie je vytvorený DEMO model (samotný výsledok) jednoznačný. Transakčný axióm DEMA pomáha zachytiť všetky komunikačné kroky a to i tie, ktoré sa vykonávajú implicitne. Vytvorený model nám poskytuje holistický a implementačno-nezávislý pohľad na podnik.

Dôležitá výhoda je, že jediná fáza, pri ktorej zapojujeme kreativitu, je fáza modelovania – vytvárania DEMO modelov. Implementácia software nám tak dáva menej priestoru pre kreativitu a menšiu slobodu v rozhodovaní pri programovaní. Tým pádom programátori nebudú určovať, ako sa má proces chovať, ale analytici, architekti a ľudia z biznisu. Vedľajší dôsledok je i menšie množstvo chýb na strane software.

Ďalšie dôsledky, ktoré z DEMA plynú sú napríklad:

- tvorba agilnej organizácie a agilných ICT aplikácií,
- prehľad naprieč organizáciou do šírky i do hĺbky,
- transparentný prehľad autorizácií a zodpovedností,
- jednovýznamové a čisté procesné a dátové vlastníctvo,
- zamestnanci sa stávajú esenciou podniku,
- udržanie kontroly nad komplikovanými zmenami v organizácii,
- vyriešenie vyváženia rovnováhy (*alignment*) medzi ICT a biznisom, a
- vyriešenie inžinierskych požiadavkov.

5.2 Dôsledky aplikácie DEMA

V prvej fáze – pri skúmaní analýzy daných procesov – sme mali za úlohu identifikovať jednotlivé transakcie, ich časti, ich druh a účastnícke role, ktoré sme im priradili. Podarilo sa nám identifikovať niekoľko ontologických transakcií, niektorým krokom sme prideliť infologickú úroveň a v základnej analýze sme natrafili i na datalogické transakcie.

Kompletné OCD (*organization construction diagram*) pre proces vzniku a zmeny zobrazujeme nižšie obrázkami 5.1 a 5.2. Dva obrázky nám dávajú informáciu o spolu 18 rolách na úrovni B-organizácie, to môže byť spolu až 19 druhov potrebných kompetencií, resp. zodpovedností; ďalej spolu v nich máme zobrazených 9 ontologických transakcií, ktoré sú práve esenciou daných procesov a nie je potrebné v nich mať ďalšie; a ďalej vidíme transakčné banky, ktoré znamenajú, že jednotliví účastníci majú prístup k informáciám vzniknutých z iných transakcií. Jednoduchosť a kompletnosť je charakteristická pre tieto diagramy.

Pri každej identifikovanej transakcii sme si ujasnili, ktoré kroky nastávajú implicitne, no pridali sme i niektoré možné situácie, ktoré by mohli nastať a v pôvodnej analýze zachytené nie sú – šlo hlavne o úkony typu *revoke*, *reject*.

Niektoré dvojice transakcií sme zlúčili do jednej – napríklad išlo o transakciu určenia typu zmeny, kedy na základe toho, akým spôsobom bola žiadosť podaná, ju vykonával určitý tím. Miesto dvoch transakcií pre každý tím sme

identifikovali len jednu transakciu a spomínané rozlíšenie sme vyšpecifikovali až v úkonovom modeli. Podobne sme zlúčili transakcie spracovania netechnických zmien, pri ktorých sme (v BPM) rozlišovali typ netechnickej zmeny (jednoduchý alebo zložitý). Tieto transakcie sme teda zlúčili opäť do jednej a daný rozdiel sme rozlišovali až na úrovni úkonového modelu.

Z analýzy sme úlohu „Zmena typu zmeny“ určili, že o samotnú transakciu nepôjde, ale že ide v skutočnosti len o časť už existujúcej transakcie – konkrétne *reject*, *revoke statement* a *revoke acceptance* transakcie určenia typu zmeny.

Z úlohy informovania klienta o výsledky zmeny sme vytvorili datalogickú transakciu Odoslanie informácie klientovi o výsledku zmeny, nakoľko sme zistili, že danú akciu človek nutne robiť nemusí a tým si šetríme prácu.

Vytvorili sme prehľad základných transakcií a ich náležité produkty a v rámci účastníckych rolí sme vytvorili prehľad, ktoré tímy do ktorej účastníckej role spadajú pre lepšiu orientáciu (prehľad sme vytvorili z pohľadu účastníckych rolí i z pohľadu existujúcich tímov).

Navrhli sme základný objektový model pre oba procesy, od ktorého sa môžeme odraziť pri návrhu jednotlivých podporných infologických či datalogických funkcií a služieb a s ktorými môžeme rátať na úrovni B-organizácii, kde nám v základe nič viac netreba.

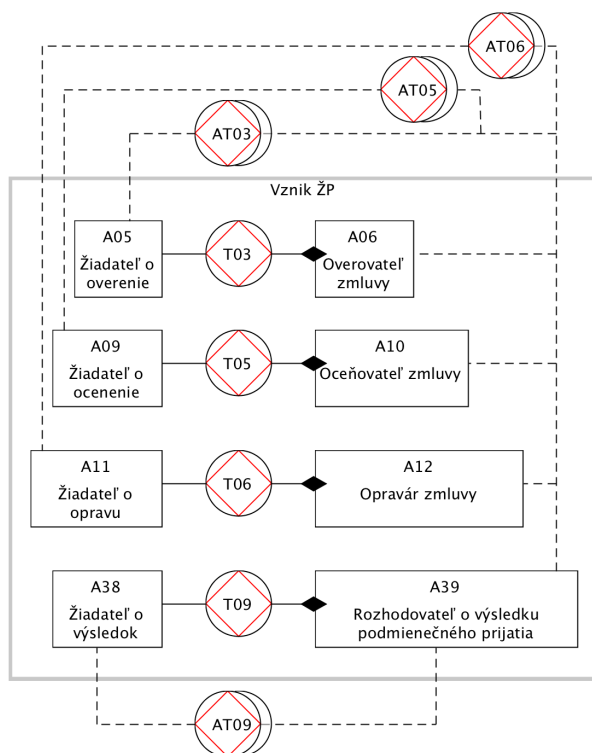
Úkonové modely (*action models*) nám predstavujú popis toho, čo má jednotlivý účastník robiť. Predstavuje to pre neho popis jeho práce a úkonov, ktoré má vykonávať a akým spôsobom. To platí pre subjekty (ľudí) i agentov (ICT službu).

5.2.1 Ďalšie návrhy

Okrem toho, že si uvedomíme skutočnú „tvár“ daných procesov, môžeme navrhnúť prípadnú zmenu charakteru procesu, aby sa stal viac sekvenčným a zamedzili situácii, kedy „každý môže všetko“. Kvôli tomu i PSD by bolo podľa aktuálneho stavu ťažké namodelovať a nebol by prehľadný.

Každopádne potrebujeme si uvedomiť úroveň jednotlivých úkonov a na základe toho aktuálne úkony tomu prispôbiť. Napr. určovateľ typu, doplnovateľ typu a ostatní aktéri z infologickej úrovne nemusia disponovať kompetenciami a nie sú zodpovední za úlohy z B-organizácie. Takto sa môže vykonať validácia a prípadná reštrukturalizácia ľudských zdrojov.

Na základe konštrukčných diagramov môžeme prispôbiť jednotlivé informačné systémy – aby zodpovedali skutočnej potrebe ako napríklad pri dotiahnutí žiadosti zmeny v podobe elektronického dokumentu, aby zdieľač agenty predstavoval službu, ktorá zisťuje aktuálne žiadosti pre určité spracovanie a podobne pre všetky ďalšie infologické a datalogické transakcie. Miesto toho, aby sme vytvárali veľké informačné systémy, pri ktorých je snaha, aby sa tváрили generalizovane, disponovali veľkým počtom funkcií, tak sa radšej pozerať na skutočné potreby od hora a tým vytvoríme služby na mieru, ktorých imple-



Obr. 5.1: Kompletný OCD procesu vzniku životného poistenia

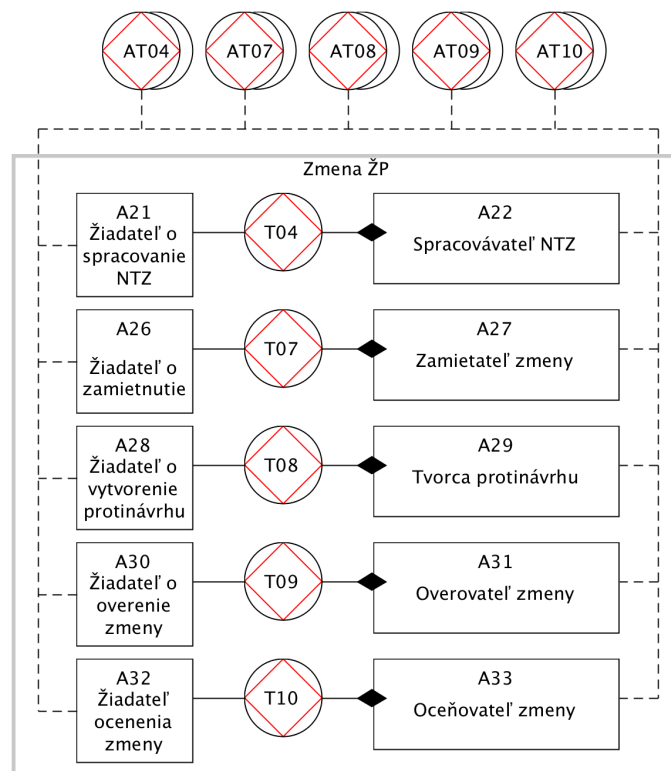
mentácia potom v konečnom dôsledku je veľmi jednoduchá. Následne zmena jednoduchých služieb by bola jednoduchá tiež – to už píšeme ale o fázy, kedy by už boli služby naimplementované.

5.3 Dôsledky BPM riešenia

V tejto sekcii si popíšeme niektoré výhody existujúceho BPM riešenia a k nim následne skúsime namietat a tým poukážeme na limit BPM riešenia a následný prechod do nevýhody.

Jednou z prvých výhod BPM riešenia je, že procesy sa dajú relatívne rýchlo naimplementovať a sú relatívne čitateľné – i pre biznis. No jedným z nepriaznivých dôsledkov BPM implementácie je to, že pri komplexnejších procesoch sa stáva proces neprehľadný, strácajú sa pre biznis podstatné úlohy, nakoľko jednou zo základných komponent v BPM je práve úloha.

Pri skúmaní implementácie v BPM systéme často narazíme na situácie, kedy je potrebné zistiť, ako funguje nejaká funkcionalita. To získame preklikávaním do rôznych úrovní procesu a skúmaním rôznych skriptov. BPM zastupuje v našom prípade riešenie na „všetko“ okrem systémov disponujúcich



Obr. 5.2: Kompletný OCD procesu zmena životného poistenia

dokumentami (žiadosťami, zmluvami alebo inými podkladmi) a informáciami o zmenách ŽP a vznikoch ŽP.

Druhá výhoda BPM riešenia, ktorú si spomenieme, je, že i analytik je schopný postaviť a navrhnuť proces – nepotrebuje špeciálne znalosti o programovaní. V súčasnej dobe je to lákadlom pre veľa firiem, nakoľko je nedostatok programátorov. To môže ale vyústiť do nestability danej aplikácie a neošetrením technických dopadov. I v našom BPM riešení bolo bežné, že analytik bol začlenený do vývoja, ale nakoľko je komplexita procesu (keď rátame aj najnižšie úrovne a ďalšie pridané funkcie) veľká, často si už nevie rady a v konečnom dôsledku práca i tak ostáva na programátora.

Treťou výhodou BPM riešenia je, že naimplementovaný proces je jednoduché spustiť a skúšať si tak jednotlivé úlohy a úkony. Poskytuje nám rôzne funkcionality, ktoré ale ešte donedávna neboli úplne funkčné. Chyby v samotnom BPM systéme spôsobili takmer ročný odklad nasadenia aplikácie. Síce sme boli pomocou BPM systému schopní naimplementovať proces rýchlo, no ladenie chýb zabralo mnoho času a mnohokrát bola aplikácia nasadzovaná. Biznis sa následne zameral viac na BPM systém, ako sa môže vylepšiť, modifikovať, no zabúda sa na vylepšovanie samotného procesu bez systému.

Veľkou nevýhodou, ktorá je spojená v aktuálnom riešení doťahovaním do-

5. VYHODNOTENIA

kumentov a práce s nimi v BPM je, že dokument, ktorý sa pohybuje naprieč procesom, je veľký objekt a BPM systém nie je vhodný pre veľké objekty. Tu sa stretávame so situáciou, kedy na prácu s dokumentami bol použitý nevhodný systém. To je cenou za to, že sme videli možnú rýchlu implementáciu daných procesov.

Nároky biznisu sú väčšinou, že chcú vidieť čo najviac informácií na jednej obrazovke, čo je tiež v protiklade s predpokladmi aktuálneho BPM riešenia. Na základe DEMA si môžeme technológiu zvoliť a vyhnúť sa takémuto systému.

Záver

V práci sme si v úvode načrtli dôležitosť a potrebu vhodných teórií pre popis procesov a podniku ako sociálneho systému. Išlo o problém, kedy existujúce nástroje a metodiky určujú, ako má proces vyzerieť, ako a čím ho implementovať. Obchádzali avšak podstatu podnikov a procesov.

Predstavili sme si disciplínu *Enterprise Engineering* predstavujúcu problematiku od základov. Následne sme pokračovali v predstavení jednotlivých pre nás relevantných teórií a niekoľkých axiómov. Ďalej sme prešli k popisu metódy DEMO, kde sme si teórie spresnili a na danú metódu sa zamerali.

V ďalšej časti práce sme vykonali samotnú analýzu procesov, vytvorili sme jednotlivé modely a tabuľky podľa metódy DEMO. Čo sa týka rozsahu, tak sme identifikovali základné časti jednotlivých procesov nutné k ich fungovaniu. Čo sa týka konštrukčných modelov, uviedli sme dva druhy ako ukážkové, ktoré by sa ďalej opakovali len s drobnými zmenami. Podobne sme tak uvideli úkonový model (AM) a urobili súhrn charakteru procesov v sekcii pre procesný diagram.

V časti práce s vyhodnotením sme spísali všeobecné výhody metódy DEMO. Ďalej sme si popisovali výhody vychádzajúce z našej aplikácii metódy DEMO na dané procesy, ktorých základom bola hlavne kategorizácia a identifikácia ontologických transakcií spolu s infologickými a datalogickými vrátane úkonov k nim patriacim. Už priamo pri identifikácii sme vykonali niektoré optimalizácie – hlavne zlúčenie niekoľko transakcií. Ďalej sme uviedli ďalšie možné návrhy, medzi ktoré patria napr. zmena charakteru procesu, validácia zodpovedností či kompetencií pracovníkov a návrh na charakter potrebných informačných služieb. V dôsledkoch BPM riešenia sme uvádzali jednak výhody BPM riešenia a k nemu následne uviedli limit, alebo protipríklad, kedy sa z výhody stáva nevýhoda. Väčšinou bol dôvod komplexita daného riešenia v BPM a veľkosť objektov „žijúcich“ naprieč BPM systémom. Uvedme ale, že pokiaľ ide o jednoduché procesy, je BPM systém vhodný – za predpokladu, že súhlasíme s nákladmi na licencie alebo kúpu daného systému.

V tejto práci by sa dalo ďalej pokračovať a premyslieť si zhotovenie DEMO

modelov na ďalšie požadované funkcionality. Medzi nimi patrí napríklad mechanizmus urgovania jednotlivých pracovníkov – ide o mechanizmus, ktorý pracovníkovi povie, že sa blíži termín spracovania napríklad úlohy spracovania NTZ. Dôvod môžu byť SLA ale aj prosté pripomínanie rozpracovaných úloh. Ďalší mechanizmus, ktorý by ďalej stálo za to implementovať pomocou DEMA je napríklad odkladanie rozpracovanej práce – typicky, keď pracovník zistí, že k napríklad overeniu zmluvy mu chýba dokument, na ktorý mu stačí počkať (avšak čakanie by malo byť pasívne zo strany pracovníka). Ako tretí príklad pre ďalšiu prácu uvedieme synchronizovaný mechanizmus nad oceňovaním TZ, kedy prípadné zmeny nad dokumentmi by mal vykonávať maximálne len jeden pracovník v určitý čas. Okrem týchto príkladov by sa ďalej dala rozšíriť implementačná časť a navrhnuť už priamo systém s vybranými technológiami, ktoré by sa už ale správali podľa navrhnutých DEMO modelov.

Literatúra

- [1] Lindsay, A.; Downs, D.; Lunn, K.: Business processes - attempts to find a definition. *Information and Software Technology*, ročník 45, č. 15, 2003: s. 1015–1019.
- [2] Ltd., J.: draw.io. April 2016. Dostupné z: <https://www.draw.io>
- [3] Team, T. U.: Free UML Tool for Fast UML Diagrams. April 2016. Dostupné z: <http://www.umlet.com>
- [4] Dietz, J. L. G.: A Communication Oriented Approach to Conceptual Modelling of Information Systems. In *Lecture Notes in Computer Science*, Berlin, Springer Verlag, 1990, s. 441–460.
- [5] Dietz, J. L. G.; Widdershoven, G. A. M.: Speech Acts or Communicative Action. In *The Second European Conference on Computer Support Cooperative Work*, Amsterdam, 1991.
- [6] Dietz, J. L. G.: DEMO: Towards a discipline of organisation engineering. *European Journal of Operational Research*, , č. 128, 2001: s. 351–363.
- [7] Dietz, J. L. G.: *Enterprise Ontology – theory and methodology*. Springer Verlag, 2006.
- [8] Tsoukas, H.: Refining common sense: types of knowledge in management studies. *Journal of Management Studies*, , č. 6, 1994: str. 761–780.
- [9] Dietz, J. L. G.; Hoogervorst, J.; et al.: The Discipline of Enterprise Engineering. *IJODE*, , č. 1, 2013.
- [10] de Jong, J.: *A Method for Enterprise Ontology based Design of Enterprise Information Systems*. SIKS Dissertatiereeks, 2013.
- [11] Apostel, L.: Towards the formal study of models in the non-formal sciences. *Synthese*, , č. 12, 1960.

- [12] Alexander, C.: *Notes on the Synthesis of Form*. Harvard University Press, 1964.
- [13] Bunge, M. A.: *Treatise on Basic Philosophy*. Dordrecht, The Netherlands: D. Reidel Publishing Company, 1979.
- [14] Mallens, P. J. M.; Dietz, J. L. G. e. a.: The Value of Business Process Modeling with DEMO prior to Information Systems Modeling with UML. In *International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, Interlaken, 2001.
- [15] Barjis, J.: Enterprise Modeling and Simulation Within Enterprise Engineering. *Journal of Enterprise Transformation*, ročník 1, č. 3, 2011: s. 185–207.
- [16] Albani, A.; Aveiro, D.; Barjis, J.: *Advances in Enterprise Engineering VI*, ročník 110. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-29902-5.
- [17] Dietz, J. L. G.: Models and Representations, 2014, presentation slides.

Zoznam použitých skratiek

AM *Action Model*

BPM *Business Process Management*

CM *Construction Model*

C-svet Koordinačný svet

C-akt Koordinačný akt/úkon

C-fakt Koordinačný fakt

DEMO *Design & Engineering Methodology for Organizations*

EE *Enterprise Engineering*

EIS *Enterprise Information System*

FM *Fact Model*

OCD *Organization Construction Diagram*

OFD *Object Fact Diagram*

P-svet Produkčný svet

P-akt Produkčný akt/úkon

P-fakt Produkčný fakt

PM *Process Model*

PSD *Process Structure Diagram*

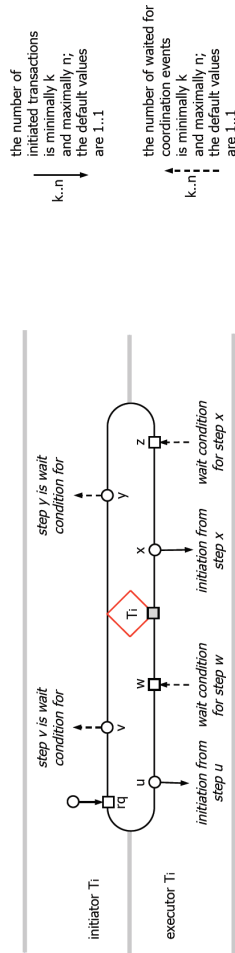
TPT *Transaction Product Table*

US *Using System* – používaný systém

A. ZOZNAM POUŽITÝCH SKRATIEK

ŽP Životné poistenie

Syntax PSD

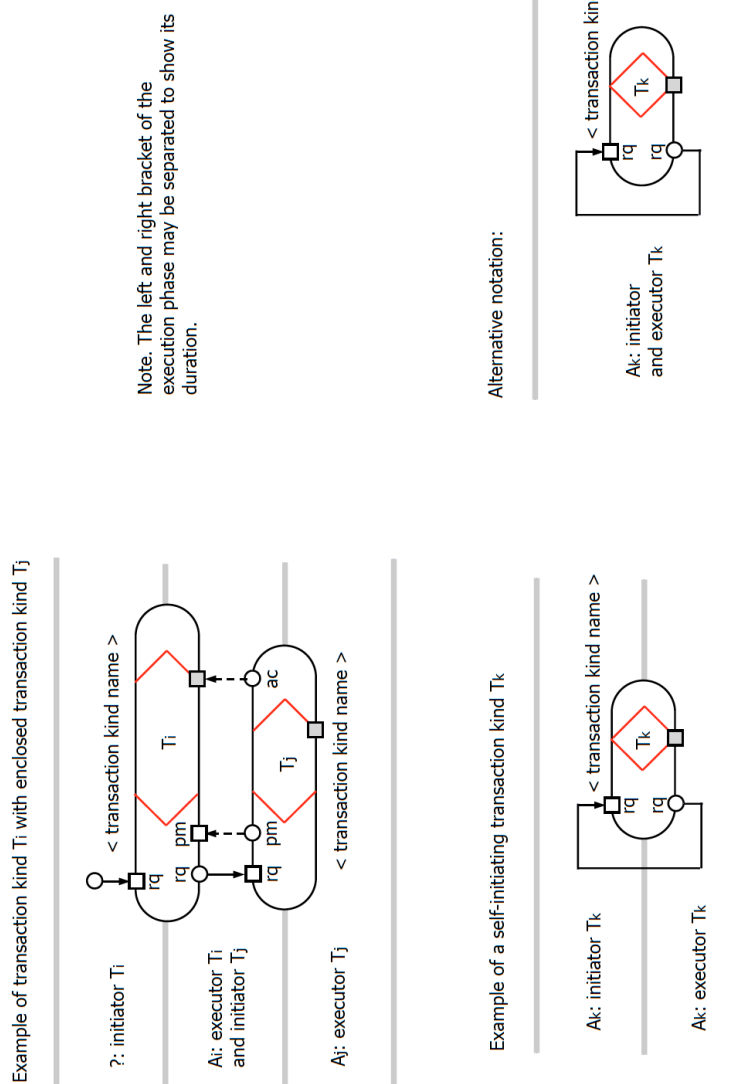


Note 1. There is a (non-proportional) linear time axis from left to right.

Note 2. A transaction proceeds in three phases: the *proposition phase* (left from the diamond), the *execution phase* (between the left and the right bracket of the diamond), and the *result phase* (right from the diamond).

Note 3. The exhibited transaction kinds can also be of the sorts informational (green diamond) en documental (blue diamond)

Obr. B.1: Syntax pre PSD, prvá časť (zdroj: [17])



Obr. B.2: Syntax pre PSD, druhá část (zdroj: [17])

Obsah priloženého CD

modely..	adresár s modelmi vo formáte PNG a UXF pre vstup do UMLet
text	adresár s textom a zdrojovou formou práce
├─ siagi_diplomova_praca.pdf	text práce
├─ siagi_zdroj.zip	zdrojová forma práce vo formáte \LaTeX
└─ umlet.zip....	archív s nástrojom UMLet obsahujúcim palety pre DEMO