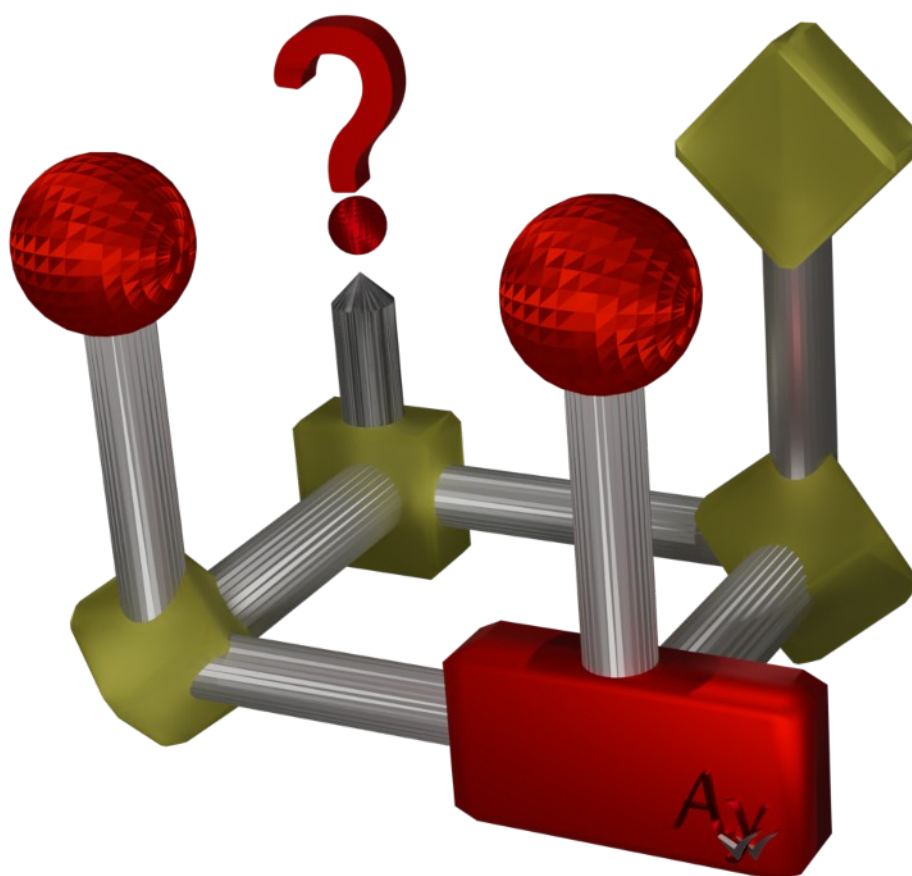


Jazyk ArchiMate® v3.0

Referenční příručka



PDQM

© 2007 – 2026

Obsah

0. ÚVODNÍ SLOVO.....	5
0.1. K PŘEKLADU.....	5
0.2. POUŽITÍ REFERENČNÍ PŘÍRUČKY.....	5
0.3. CO JE ARCHIMATE.....	5
0.4. ARCHIMATE - UKÁZKY.....	5
0.5. ARCHIMATE – ŠKOLENÍ.....	5
0.6. GRAFICKÁ REPREZENTACE.....	6
1. ARCHIMATE – BUSINESS ÚROVEŇ.....	7
1.1. STRUKTURÁLNÍ ELEMENTY JAZYKA – AKTIVNÍ ČINITELÉ.....	7
1.2. STRUKTURÁLNÍ ELEMENTY - PASIVNÍ OBJEKTY.....	7
1.3. PŘÍKLADY BUSINESS VRSTVY.....	9
1.4. ELEMENTY JAZYKA POPISUJÍCÍ CHOVÁNÍ.....	9
1.5. METAMODEL BUSINESS VRSTVY.....	10
2. APLIKAČNÍ VRSTVA.....	12
2.1. ELEMENTY APLIKAČNÍ VRSTVY.....	13
2.2. METAMODEL APLIKAČNÍ VRSTVY.....	13
3. TECHNOLOGICKÁ A FYZICKÁ VRSTVA.....	15
3.1. ELEMENTY TECHNOLOGICKÉ VRSTVY.....	15
3.2. METAMODEL TECHNOLOGICKÉ VRSTVY.....	17
3.3. ELEMENTY MIMO VRSTVY.....	20
4. VAZBY V JAZYCE.....	21
4.1. STRUKTURÁLNÍ VAZBY.....	21
4.2. DYNAMICKÉ VAZBY.....	22
4.3. DALŠÍ VZTAHY.....	23
4.4. VERTIKÁLNÍ VAZBY - PROVÁZÁNÍ VRSTEV.....	24
5. POPIS MOTIVACE.....	26
5.1. VYLOUČENÉ ELEMENTY STRATEGICKÉHO PLÁNOVÁNÍ.....	26
5.2. ELEMENTY POPISU MOTIVACE.....	26
5.3. METAMODEL ROZŠÍŘENÍ MOTIVACE.....	29
6. POPIS PROJEKTŮ A PROGRAMŮ – IMPLEMENTAČNÍ ROVINA.....	30
6.1. PŘEHLED ELEMENTŮ POPISU ČINNOSTÍ.....	30
6.2. METAMODEL PLÁNU PROJEKTU.....	31
6.3. PŘÍKLAD POPISU PROJEKTU.....	31
7. K ČEMU ARCHIMATE.....	34
7.1. ZHODNOCENÍ ARCHIMATE.....	34
7.2. JAK S ARCHIMATE PRACOVAT.....	35

Přehled obrázků

Obrázek 1 Příklad strukturálního modelu – použití business rozhraní.....	8
Obrázek 2 Příklad popisu produktu.....	9
Obrázek 3 Metamodel business vrstvy.....	11
Obrázek 4 Příklad agregovaných vazeb.....	11
Obrázek 5 Ukázka hierarchických vztahů elementů stejného typu.....	12
Obrázek 6 Metamodel aplikační vrstvy.....	14
Obrázek 7 Příklad aplikační vrstvy.....	14
Obrázek 8 Ukázka dynamických vazeb - vzájemného volání systémů.....	15
Obrázek 9 Metamodel technologické vrstvy - informační model.....	17
Obrázek 10 Příklad implementace prostředí pro interní systém a propojení do internetu.....	18
Obrázek 11 Příklad modelu fyzické výroby (spalovna odpadu).....	19
Obrázek 12 Ukázka technologického procesu (spalování odpadu).....	20
Obrázek 13 Zakreslení kompozice produktu ze služeb.....	21
Obrázek 14 Kompozice zakreslená explicitními vazbami.....	21
Obrázek 15 Vertikální vazby business vrstvy na nižší vrstvy.....	24
Obrázek 16 Vertikální vazby implementační vrstvy na technologickou.....	24
Obrázek 17 Business elementy používají elementy aplikační vrstvy.....	25
Obrázek 18 Metamodel rozšíření <i>Motivace</i>	28
Obrázek 19 Vertikální vazby strategie a vize do business vrstvy.....	29
Obrázek 20 Metamodel rozšíření <i>Implementace</i> s vazbami na Enterprise architekturu.....	30
Obrázek 21 Příklad rozdělení projektu na 2 etapy.....	31
Obrázek 22 Příklad: Plán realizace etapy projektu.....	31
Obrázek 23 Ukázka plánu úkolu projektové etapy.....	32
Obrázek 24 Ukázka zjednodušeného modelu projektu.....	33

Tento materiál vytvořili v PDQM s využitím školících materiál.

ArchiMate® je registrovanou známkou The Open Group. Samotný standard je veřejně publikován

Základem materiálu je veřejně dostupná specifikace jazyka ArchiMate od The Open Group. Materiál obsahuje výtahy ze specifikace (překlady) a vlastní poznámky autorské organizace a to včetně doslovných pasáží určitých částí originální specifikace.

Obsah tohoto dokumentu nelze považovat překlad uvedených dokumentací, nečiní si nároky na úplnost ani doslovnou správnost. Současně obsahuje poznámky, komentáře a jiné texty, které z uvedených publikací nepochází. Části voleného překladu u komentářů nejsou od sebe odděleny.

Grafické elementy jsou vytvořeny pomocí freeware grafických prvků společnosti Orbus software.

Jak zpracovávat business model

AyMINE – nástroj pro správu business modelu a spolupráci napříč celou společností



V AyMINE® zpracujete celý projekt, požadavky i analýzu. Díky jeho snadné dostupnosti můžete spolupracovat se všemi, kteří mají k modelu co říct.

www.aymine.cz

0. Úvodní slovo

ArchiMate je otevřeným standardem tvořeným the Open Group, organizací stojící i za standardem TOGAF. The Open Group vydává volně dostupné specifikace, na jejichž základě je tento souhrnný materiál vytvořen.

Cílem tohoto souhrnu je poskytnout esenci klíčových informací, která je určena analytikům a architektům, kteří mají zkušenost s jinými modely a ArchiMate / TOGAF berou do ruky za účelem rozšíření svých teoretických znalostí.

0.1. K překladu

Překlad využívá obvyklého názvosloví používaného v IT sféře. Materiál se snaží v co možná největší míře využívat české výrazy, ale současně se neodchylovat od výrazů anglických, které v běžné řeči zdomácněly. Pro udržení vztahu mezi anglickým standardem jsou často uváděny i originální anglické názvy a označení.

Základním dokumentem, ze kterého vychází popis, je technický standard jazyka verze 3.0

0.2. Použití referenční příručky

Materiál vznikl pro vnitřní potřebu PDQM a jako doplňkový materiál ke školením, které PDQM poskytuje svým klientům.

Materiál reflektuje zkušenosti školitele a jeho znalosti o tom, co analytikům činí potíže a co se obtížně dostává do běžné praxe.

Dokument je určen pro business analytiku a zaměřuje se proto na business úroveň EA a jazyka ArchiMate.

0.3. Co je ArchiMate

ArchiMate je jazykem pro kompletní popis Enterprise architektury (EA). Jazyk vznikl na základě potřeby modelovacího prostředí pro celou architekturu v rozsahu a způsobem definovaným standardem TOGAF.

TOGAF definuje 3 úrovně enterprise architektury, které ArchiMate sleduje:

1. **Business úroveň** – popisuje produkty a služby, které jsou organizací poskytovány externím subjektům nebo od nich přijímány a struktury, které tyto služby realizují, činitele, role a procesy.
2. **Aplikační úroveň** – popisuje aplikační služby sloužící pro realizaci business služeb a samotné aplikace, které služby zajišťují a poskytují. Zahrnuje business aplikace.
3. **Technologická úroveň** – popis technologického prostředí zajišťují potřebné technické služby pro fungování aplikací. Zahrnuje hardware, sítě a systémový software.

0.4. ArchiMate - ukázky

Ukázky dokumentace v ArchiMate jsou např. dokumentace AyMINE na stránkách technické dokumentace – viz www.aymine.org.

Technická dokumentace je současně ukázkou, kde jsou hranice ArchiMate. Jazyk jednoduše není vhodný na všechno a pro detailní úrovně modelu není postačující. Proto jej kombinujeme s dalšími jazyky – BPMN, E-R modely, UML (příp. SysML).

0.5. ArchiMate – školení

Nejenom ArchiMate, ale i další jazyky školíme v rámci [cyklu školení pro business analytiku](#). Podrobnosti jsou na [stránkách PDQM](#). Rádi pomůžeme zlepšit i váš tým.

0.6. Grafická reprezentace

0.6.1. Tvary elementů

Většina vizuálních prvků v modelu ArchiMate má dvojí grafickou reprezentaci - tzv. box – obdélník a vizuálně názornější grafickou. Účelem dvojí reprezentace je umožnit snadné hierarchické zakreslení vztahu *náležení do sebe*. Obdélníková reprezentace se tak typicky používá v situaci, kdy se na schématu zakresluje i co daný element interně obsahuje.

Jsou ale možné i opačné přístupy - zobrazit obdélník i když se nerozpadá a naopak, požit grafické zobrazení i s interními detaily.

0.6.2. Používání barev

ArchiMate nepředepisuje používání barev v modelu nicméně je jejich použití v praxi ve většině případů nezbytné.

V rámci standardu ArchiMate jsou používány barvy pro rozlišení popisující informace, struktury a chování elementů architektury. Tato příručka tuto logiku určenou pro metamodel¹ nedodrží a drží se obvyklejší formy, kdy barvy jsou používány k rozlišení příslušnosti k vrstvě modelu.

Barevné rozlišení vrstev je v praxi velmi důležité, protože používání tvarově shodných elementů pro business i aplikační vrstvu znemožňuje snadné rozlišení, o které vrstvy daný element patří. ArchiMate navrhuje i alternativní rozlišení pomocí písmen označujících danou vrstvu v levém horním rohu elementu, ale většina modelovacích tuto syntaxi nepodporuje.

Na druhou stranu je vhodné využívat barvy v širším rozsahu, než jenom na odlišení vrstev. Užitečné je např. odlišovat elementy interní a externí ve vztahu k projektu / organizaci / části modelu, odlišovat standardní a nestandardní běh apod.

Používání barev, stejně jako jiných volitelných nástrojů jazyka, pomůže přehlednosti pouze v případě, že jsou používány konzistentně a všem čtenářům srozumitelně.

¹ Model popisující, jak modelovat.






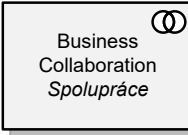


1. ArchiMate – business úroveň

Kapitola popisuje business úroveň modelu, ale vzhledem k tomu, že logika aplikační a technologické úrovně je shodná a shodný je i význam většiny elementů, je kapitola současně popisem i pro ostatní úrovně. Oblasti, které jsou shodné pro ostatní úrovně, nejsou popisovány opakovaně.

1.1. Strukturální elementy jazyka – aktivní činitelé

Strukturální popis zaměřuje na popis statických struktur organizace resp. modelovaného business prostředí.

Aktivní činitelé jsou subjekty, které mohou něco vykonávat – tedy lidé (ať už jako aktéři nebo role) nebo na nižších úrovních moduly systémů, které mohou pracovat autonomně

Business Actor Činitel	Organizační jednotka schopná vykonávat činnost ² Pracovní pozice		
Business role Role	Pojmenování pro konkrétní činnost, kterou činitel (actor) v určitém pracovním kontextu vykonává Pracovní role		
Business collaboration Spolupráce	Součinnost dvou či více rolí při realizaci konkrétní činnosti (Spolupráce je „virtuální role“ zastupující skupinu rolí - strukturální element) Spolupráce není oddělení, nejde o reprezentaci organizační jednotky Spolupráce v rámci modelu může zůstat bezejmenná		
Business interface Rozhraní	Rozhraní popisuje, jak je business služba dostupná prostředí (těm, kdo ji chtějí využívat) Business rozhraní může být propojeno s jednou nebo i více business službami, čímž se ukazuje, že tyto služby jsou prostřednictvím rozhraní dostupné.		

1.2. Strukturální elementy - pasivní objekty

Pasivní objekty, na rozdíl od aktivních, samy žádnou činnost (aktivitu) nevykonávají, mohou být předmětem činností, které iniciují aktivní subjekty.

² Poznámka k implementaci v Enterprise Architect (EA) v13: Podpora ArchiMate v2 měla velmi užitečný příznak, že aktér je skupina (tzv *compound*), ale tato možnost v implementaci v3 už není. Tento příznak nepodporuje ani standard jazyka, takže i když byl užitečný, bohužel nelze zrušení podpory aktéra, na kterém byla jasné, že jde o oddělení, divizi nebo tým považovat za nedostatek nástroje.

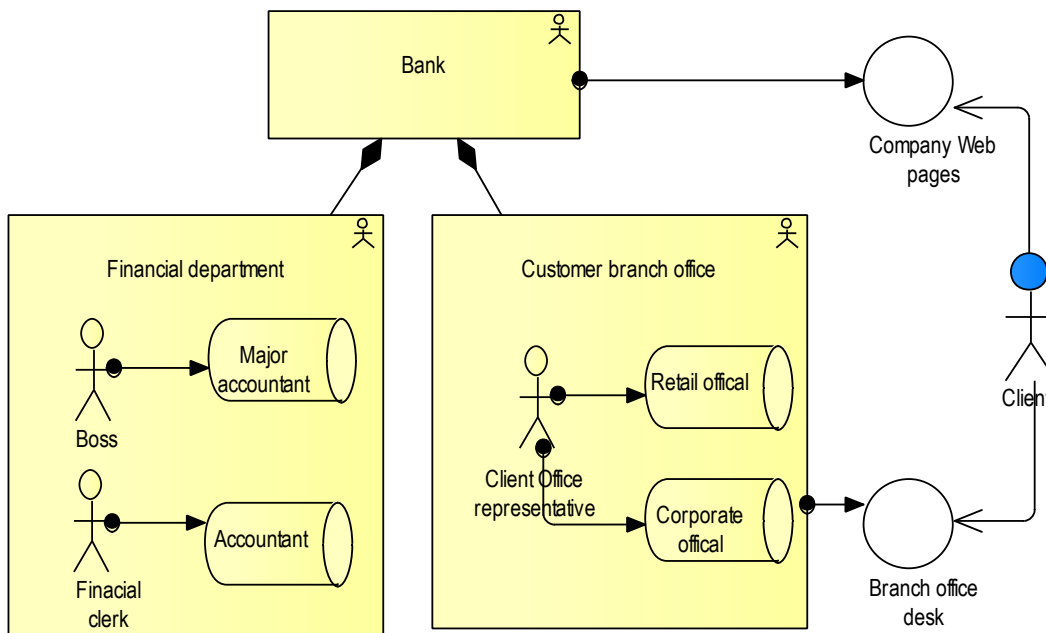
<p>Business object Objekt</p>	<p>Business objekt, je obecně jakákoli pasivní entita, kterou business využívá. Typicky jde o jednotku informace (např. smlouva, faktura), ale obecně jde o cokoli, co je předmětem business modelu, tedy např. publikace v modelu nakladatelství,</p>	
<p>Representation Reprezentace</p>	<p>Vnímatelná (zřetelná) forma reprezentace informace (typicky dokument, ale též např. přehled na webové stránce)</p> <p>Reprezentace může vyjadřovat jeden či více business objektů.</p>	
<p>Product Produkt</p>	<p>Balíček služeb a výrobků, které jsou společně jako produkt nabízeny business jednotkou externím subjektům.</p> <p>Striktně vzato je produkt kolekce objektů, která může zahrnovat pasivní i aktivní objekty, které dohromady produkt tvoří.</p> <p>Produkt se typicky jmenuje názvem komunikovaným s klientem.</p>	
<p>Contract Kontrakt, Smlouva</p>	<p>Popis podmínek (formální i neformální), které se vztahují k produktu.</p> <p>Kontrakt je typicky nějak zaznamenan a proto je element kontrakt speciálním případem business objektu (proto s ním má i velmi podobnou vizuální formu)</p>	

Poznámky

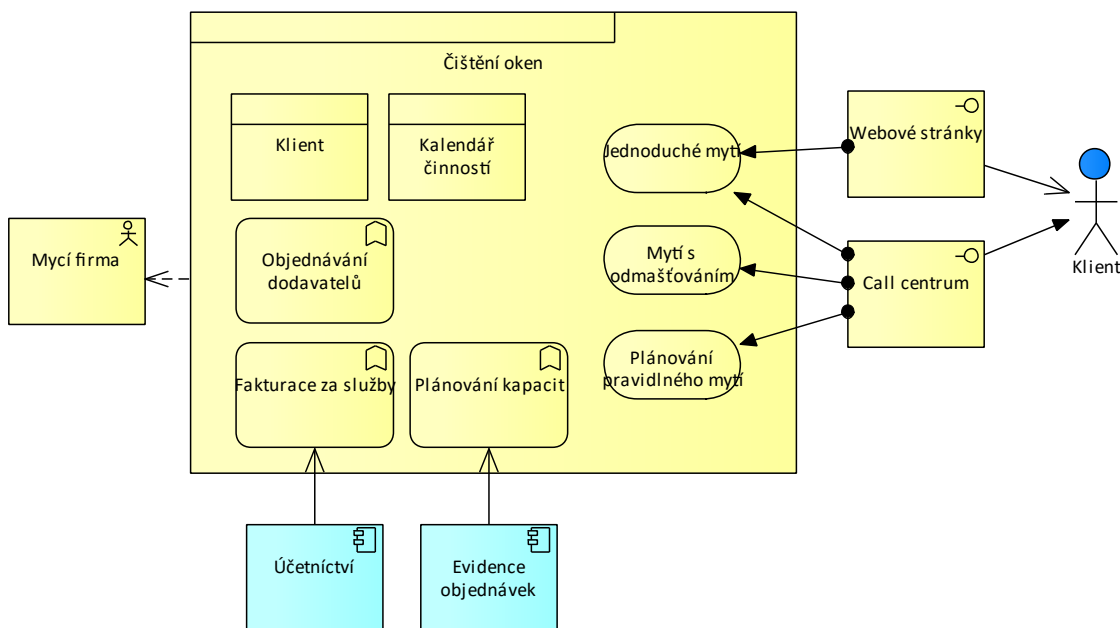


Rozhraní je nástrojem, který modeluje konkrétní vlastnosti a schopnosti určité role, které jsou významné pro plnění popisovaného businessu. De facto tak vyjadřují význam role pro popisované činnosti.

1.3. Příklady business vrstvy



Obrázek 1 Příklad strukturálního modelu – použití business rozhraní



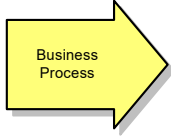
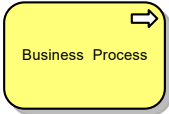



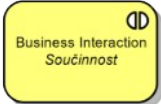
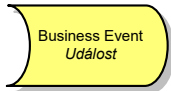

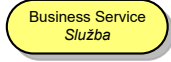
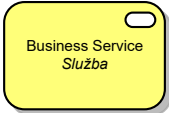
Obrázek 2 Příklad popisu produktu

1.4. Elementy jazyka popisující chování

Elementy popisující chování jsou v angličtině nazývané Behavioural elements.

Z vnějšího pohledu na organizaci jsou z chování patrné především business služby, které organizaci navenek poskytuje. Z obecnějšího pohledu jsou business funkce viditelnými schopnostmi (capabilities) organizace. Příkladem business služby je i výrobek, který je hmotným výstupem.

Schopnosti organizace jsou vnitřně zajištěny procesy. V procesech jsou realizovány konkrétní business funkce.

Business Process <i>Business proces</i>	Interní činnost nebo sada činností realizující business službu Proces je přiřazen konkrétní roli, která ji vykonává Proces se v detailu může členit na dílčí procesy, události, funkce, interakce, které jsou přiřazeny rolím podrobněji popisujícím roli procesu ³		
Business Function <i>Business funkce</i>	Interně vykonávaná skupina činností, kterou vykonává jedna role ⁴ . Seskupuje činnosti (popsané procesy) do logických celků		
Business interaction <i>Součinnost</i>	Činnost, kterou vykonávají spolupracující role. Součinnost popisuje děje, které vykonává virtuální role spolupráce nebo na které se podílí více rolí; ekvivalent činnosti v situaci, kdy se na ní podílí více subjektů ⁵ . Součinnost je popis činnosti-funkce, kterou společně vykonávají role sdružené v elementu spolupráce; nemůže být proto přiřazena jedné roli, ale výhradně spolupráci.		
Business event <i>Událost</i>	Jakákoli interní nebo externí událost, která ovlivňuje průběh business procesů, funkcí nebo spolupráce		
Business service <i>Služba</i>	Z vnějšího prostředí viditelná „funkčnost businessu“ (schopnost produkovat výstup). Interně je služba typicky realizována procesem		

1.5. Metamodel business vrstvy

Poznámka: Stejná pravidla rozšiřitelnosti metamodelu, která jsou uvedena pro business vrstvu, platí i pro všechny ostatní vrstvy a rozšíření modelu.

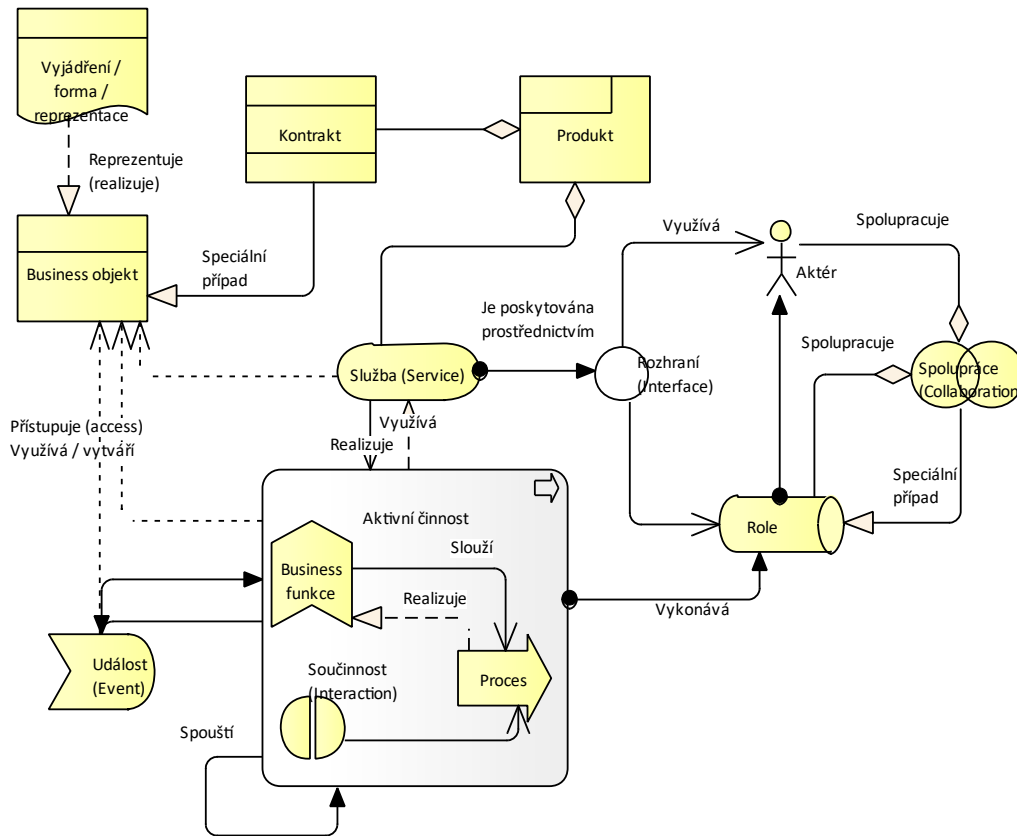
³ Koncept hierarchie je zde důležitý: Pokud se proces dělí na podprocesy, každý z nich může vykonávat jiná role, ale tyto role musí být členěním role, která má přiřazen hlavní proces. Příklad: Obchodní úsek vlastní proces prodeje, který mohou vykonávat jednotliví pracovníci z oddělení. Do procesu ale nesmí patřit činnosti pracovníků jiného oddělení. Hierarchie je důležitá i v tom, že stejný element procesu fakticky zahrnuje tři elementy: Procesní oblast (která se dále štěpí na procesy), celý proces i jednotlivé aktivity (tasks). ArchiMate tak na nejnižší úrovni popisuje proces jako posloupnost činností a spoluprací.

⁴ Business funkce je velmi úzce definována a díky tomu ji nelze zaměňovat s pojmem *Task* / úkol, jak je třeba v BPMN. Do definice úkolu by spadala i interakce, která ovšem dle definice nevyžaduje pouze interní, ale umožňuje i externí spolupracující subjekty, čímž by překračovala hranice procesu.

V jazyce Enterprise Architektury funkci odpovídá schopnost – capability.

ArchiMate nemá nástroje na detailní popis procesu a předpokládá, že pro detailní popis posloupnosti činností bude použit jiný modelovací nástroj (doporučené je BPMN).

⁵ Interakce v ArchiMate je významným aspektem modelování, který je v ostatních jazycích přehlížen. Součinnost (Interakce) je fakticky úkol realizovaný spoluprací více rolí – ty jsou typicky společně definovány spoluprací (*Collaboration*) – a tak je možné přímo vyznačit, že na plnění úkolu se podílí více subjektů a to interních i externích. Např. BPMN nemá přímé přiřazení role úkolu (používání *swimlanes* je čistě věc úzusu), ale a priori neumožňuje definovat aktivitu v procesu jako formu spolupráce, protože musí vždy patřit určitému subjektu – do konkrétního bloku (poolu).

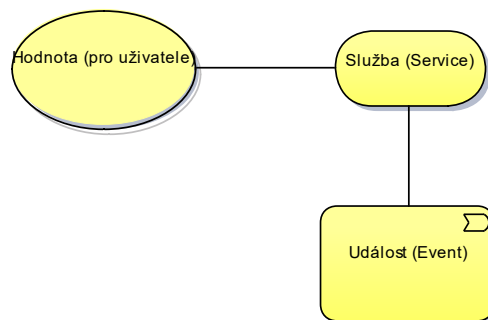


Obrázek 3 Metamodel business vrstvy

Metamodely uvedené v této příručce jsou rozšířením schémat standardu ArchiMate 2.1, protože kromě vazeb uvedených přímo v metamodelu zahrnují i důležité vztahy vyplývající z definice elementů a vzorových příkladů standardu.

1.5.1. Odvozené vztahy

Metamodel vyjadřuje primární vazby mezi elementy vrstvy. Nejsou v něm vyjádřeny vazby do jiných vrstev (zejména aplikační a motivace) a také neobsahuje agregované vazby, které jsou také povolené.



Obrázek 4 Příklad agregovaných vazeb

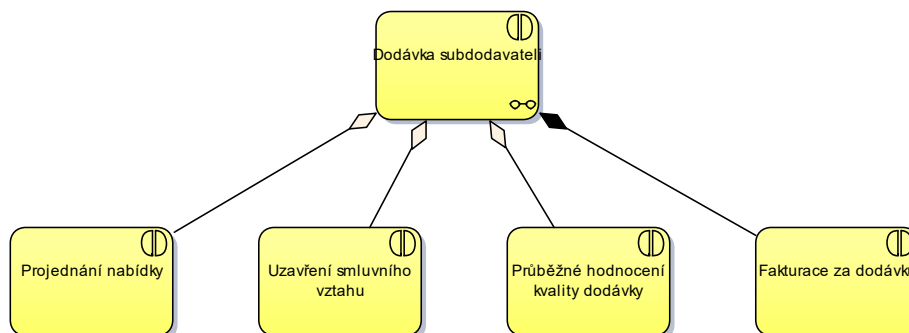
Agregované vazby jsou vždy strukturální a mohou být nejméně tak silné, jako jsou vazby primární. V příkladu na obrázku je služba asociována s hodnotou, kterou má pro uživatele, i s událostí, která potřebu využít službu aktivuje. Asociace je nejslabší vazba, takže jejímu použití mezi libovolnými elementy v podstatě nic nebrání.

Agregovaná vazba neznámá, že by model musel obsahovat elementy, které jsou v metamodelu mezi propojenými objekty. Konkrétně v příkladu na Obrázek 1 je vazba mezi službou a hodnotu, ale to neznámá, že musíme mít v modelu povinně i produkt.

Strukturální vazby jsou pro účely možné agregace seříděny podle jejich síly. Tabulka v oddílu Strukturální Vazby (strana 21) popisuje vazby od "nejsilnější" po nejslabší.⁶

1.5.2. Vazby nezaznamenané v metamodelu

Metamodel pro jednoduchost nezaznamenává vazby hierarchického členění modelu – agregace, kompozice a specializace. Obecně platí, že každý element jazyka ArchiMate může být kompozitní, a uvedené vztahy jsou mezi elementy stejného typu povoleny vždy.



Obrázek 5 Ukázka hierarchických vztahů elementů stejného typu

2. Aplikační vrstva

Modely aplikační vrstvy vyplývají z modelů jazyka UML.

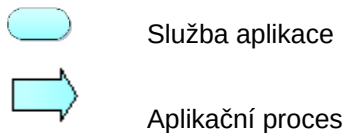
Řada elementů business vrstvy mají podobný význam i v aplikační vrstvě. Konkrétně se to týká elementů:



⁶ Z definice odvozených vazeb plyne mnoho důsledků, z nichž řada je prakticky užitečná (v závorce je vždy uveden důvod platnosti):

- Mezi elementy stejného typu může být teoreticky jakákoli strukturální vazba (je povolena kompozice, která je nejsilnější vazbou, a tedy jsou povoleny i všechny slabší vazby)
- Mezi jakýmkoli elementy je povolena asociace. (Asociace je nejslabší vazba a mezi každými elementy v praxi existuje v metamodelu nějaká cesta)
- Kde je to pro srozumitelnost rozumné, vazba realizace může nahradit vazbu asociace (realizace je slabší než asociace)

Je ale třeba upozornit, že samotný standard uvádí omezené možnosti vazeb, např. nepodporuje, aby jedna business funkce realizovala jinou. Stejně tak je výrazně omezené např. vazby Access, i když jde o "slabou" vazbu.



V následujícím přehledu jsou uvedeny nové elementy a elementy funkce a služby s přesně definovaným významem⁷. Význam ostatních elementů je analogický jejich významu v business vrstvě.

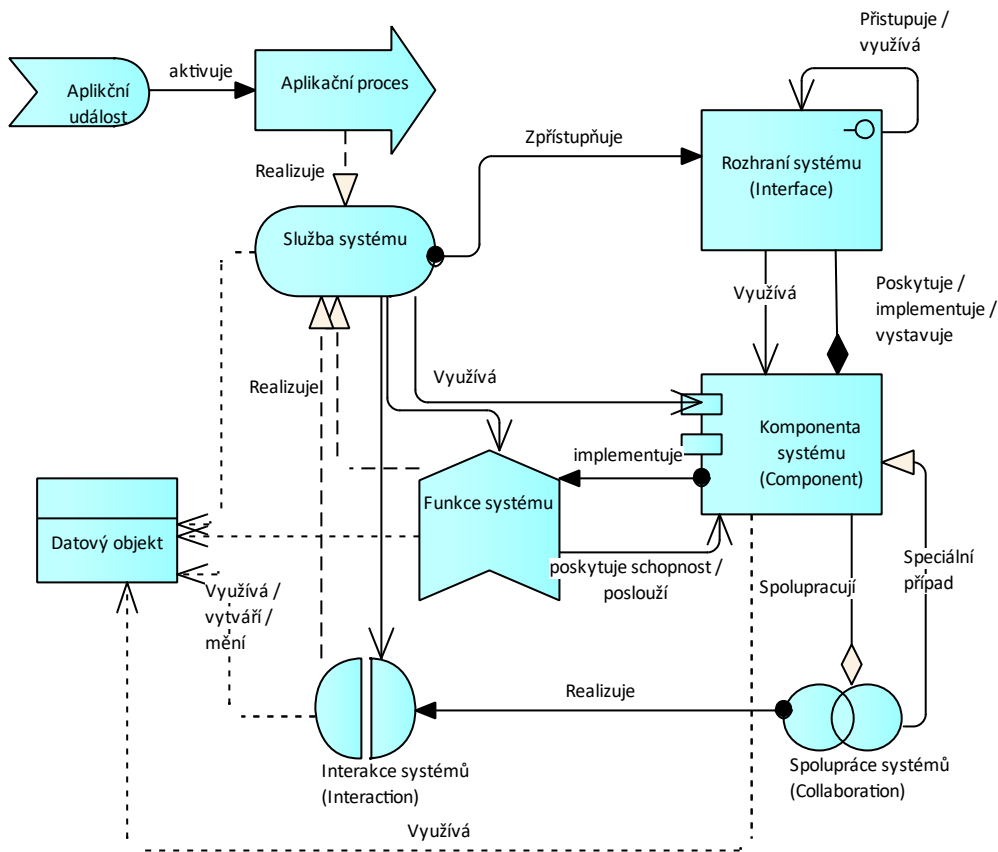
2.1. Elementy aplikační vrstvy

Funkce aplikace	Konzistentní skupina vnitřních kroků aplikace nebo komponenty. Funkce aplikace nejsou využívány přímo businessem (nemají vazbu na business vrstvu), ale interně realizují aplikační služby. Funkce systému může realizovat více služeb.		
Aplikační proces	Posloupnost systémových kroků, které realizují konkrétní činnost, typicky aplikační službu. Z uživatelského hlediska je aplikační proces neviditelný, neměl by obsahovat kroky uživatelské obsluhy.		
Aplikační událost	Událost, která nastává v rámci zpracování procesů a činností na úrovni aplikací Např. změna stavu entity, příchod nového podnětu; z aplikačního rozhraní. Událost může vyvolat jakýkoli aktivní činitel - proces nebo funkce, spolupráce nebo rozhraní.		
Služba aplikace	Z vnějšího prostředí viditelná „funkčnost aplikace pro business“ zajišťována jednou nebo více komponentami, dostupná přes definované rozhraní a užitečná pro okolí aplikace.		
Rozhraní aplikací	Definice, jak je aplikace propojena se svým okolím.		
Application component Komponenta aplikace	Samostatně použitelný a řízený modul aplikace (potenciálně nahraditelný jinou částí), kterou zapouzdřuje určitou funkcionalitu, kterou poskytuje prostřednictvím služeb		

2.2. Metamodel aplikační vrstvy

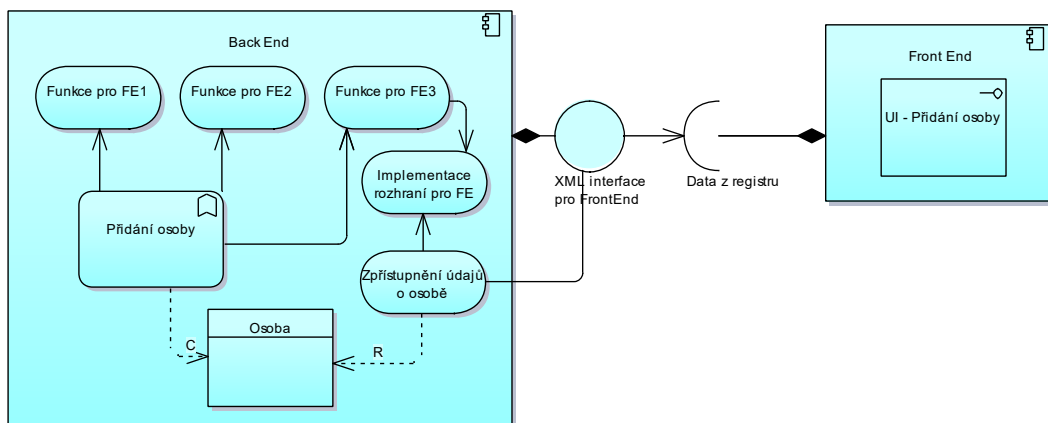
Pro povolené vazby aplikační vrstvy platí stejná pravidla, která jsou uvedena u business vrstvy.

⁷ Z hlediska business analýzy je významný element služby aplikace, protože business analytik se standardně nezabývá interní funkcionalitou aplikací. Pokud se interní funkcionalita dostává do analytických schémat, měla by to být schémata určená pro vývojový tým nikoli uživatele nebo management.

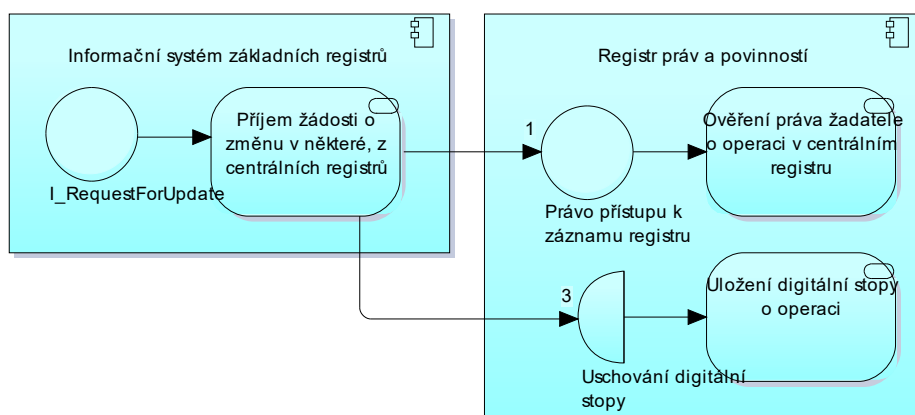


Obrázek 6 Metamodel aplikační vrstvy

Poznámka: Datový objekt (stejně jako řada jiných objektů) nemá v rámci své vrstvy objekt, který by jej implementoval, ale vazbou agregace může být přiřazen do komponenty systému, což odpovídá logickému vnímání aplikace, která obsahuje např. databázi s objekty. Datový objekt realizuje artefakt na úrovni technologické vrstvy (pro většinu datových objektů ale nemá reálný význam technologický artefakt a příslušnou realizační vazbu modelovat.)



Obrázek 7 Příklad aplikační vrstvy



Obrázek 8 Ukázka dynamických vazeb - vzájemného volání systémů

Poznámka ke schématu: Vazby volání mohou být libovolně pojmenovány, zde se používá čísla pro označení pořadí volání (schéma je podmnožinou složitějšího modelu a některé vazby v něm chybí)

Poznámky k dynamickému modelu:

ArchiMate nepodporuje vazbu *Triggering* (volání) mezi funkcí a rozhraním nebo komponentou, protože na úrovni vzájemné spolupráce komponent by se neměly funkce systému vyskytovat (ty implementují služby). V praxi to ale vyžaduje, aby každá funkčnost byla popsána službou, má-li být možné ji modelovat. Nicméně funkce se mezi sebou volat mohou. To vytváří zmatek, protože např. realizace složené funkce funkcemi ve více komponentách je teoreticky obtížně možné modelovat – vazby volání by musely být zobrazeny přímo z komponenty, i když fakticky ji vždycky realizuje nějaká konkrétní funkce.

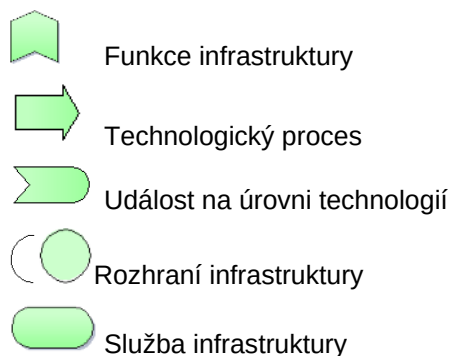
Pro podrobnější popis vnitřního chování komponenty není ArchiMate vhodným jazykem. Je určen pro popis architektury a chybí mu dostatečný vyjadřovací aparát – je proto vhodnější použít UML. Je ale také důležité si uvědomit, že vnitřní chování systému spadá do jejich designu a přesahuje rozsah toho, co by měla enterprise architektura obsahovat.

3. Technologická a fyzická vrstva

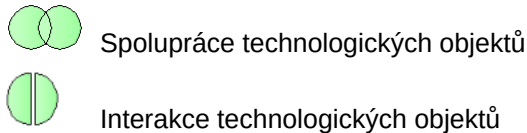
Technologická vrstva popisuje prostředí, ve kterém jsou provozovány aplikace organizace.⁸

3.1. Elementy technologické vrstvy

Technologická vrstva obsahuje některé elementy významově analogické shodně pojmenovaným elementům business a aplikační vrstvy:



⁸ Technologická vrstva v ArchiMate je inspirována UML 2.0, takže její forma je technickým pracovníkům IT snadno srozumitelná.



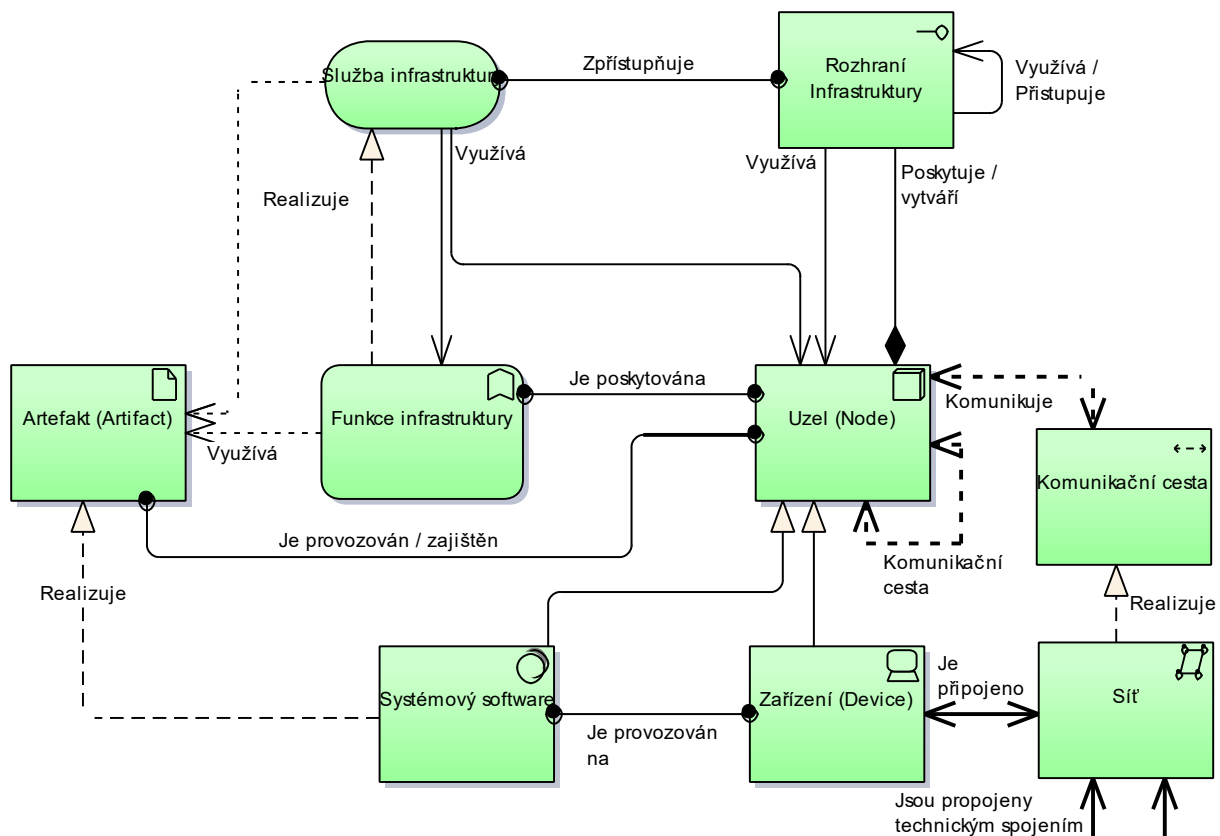
Význam uvedených elementů je totožný s jejich významem v business a aplikační vrstvě a jistě není potřebné je znovu popisovat.

V technologické vrstvě jsou zajímavější elementy, které popisují fyzickou informační architekturu a fyzickou architekturu obecně

3.1.1. Elementy informačních technologií

Node <i>Uzel</i>	Výpočetní prostředí, které zpracovává a vytváří artefakty (data) Typicky jde o servery (aplikační, fyzické), pracovní stanice nebo složitější celky (datová centra)		
Device <i>Zařízení</i>	Fyzické zařízení, které zajišťuje výpočetní kapacitu. Jde o speciální případ uzlu.		
Infrastructure interface <i>Fyzické rozhraní</i>	Místo, kde je funkčnost uzlu dostupná pro uživatele nebo jiné uzly		
Network <i>Sít'</i>	Fyzické propojení uzlů nebo zařízení		
Communication path <i>Komunikační cesta</i>	Propojení uzlů, přes která si mohou vyměňovat informace		
Infrastructure service <i>Služba infrastruktury</i>	Zvenku viditelné a využitelné služby infrastruktury. Příkladem jsou služby SOA bus (předávání zpráv, pojmenovávání služeb), adresářové služby apod.		
System software <i>Systémový software</i>	Software prostředí, který je potřebný pro fungování aplikací, zpracování, uschovávání a výměnu informací / artefaktů. Jde o speciální případ uzlu Příkladem jsou databázový server, Corba server ale i balíkové software jako CRM, ERP nebo procesní server		
Artifact <i>Artefakt</i>	Fyzický kus informace, který vzniká nebo je používán v rámci procesu. Zahrnuje dokumenty, tabulky, skripty, zprávy apod.		

3.2. Metamodel technologické vrstvy



Obrázek 9 Metamodel technologické vrstvy - informační model

3.2.1. Popis fyzické infrastruktury

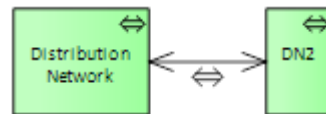
Technologická vrstva neslouží pouze k popisu IT struktur, ale k popisu fyzického prostředí obecně obsahuje proto elementy, které umožňují modelovat i prostředí továrny a materiálových toků.

Facility <i>Prostor</i>	Prostor je obecně jakýmkoli prostorem sloužícím k umístění techniky. Příkladem je továrna, dílna, kancelář Prostor je specifickým případem uzlu (node)	
Equipment <i>Vybavení</i>	Vybavení je jakákoli věc sloužící k práci - nástroj, výrobní jednotka, přístroj aj. Příkladem mohou být scanner, rentgen, svářečka, NC řezačka, extruder atd. Vybavení je specifickým případem uzlu (node) Vybavení je typicky aktivním činitelem, který se podílí na realizaci technologických procesů	
Material <i>Materiál</i>	Materiál je hmotná věc (tedy nikoli informace), která je zpracovávána. Používá se pro modelování vstupů i výstupů procesů, tedy jak vstupního materiálu, tak hotových výrobků. Materiál není pouze surovina, ale i výrobek, který se typicky skládá z dílčích součástí. Materiál je	

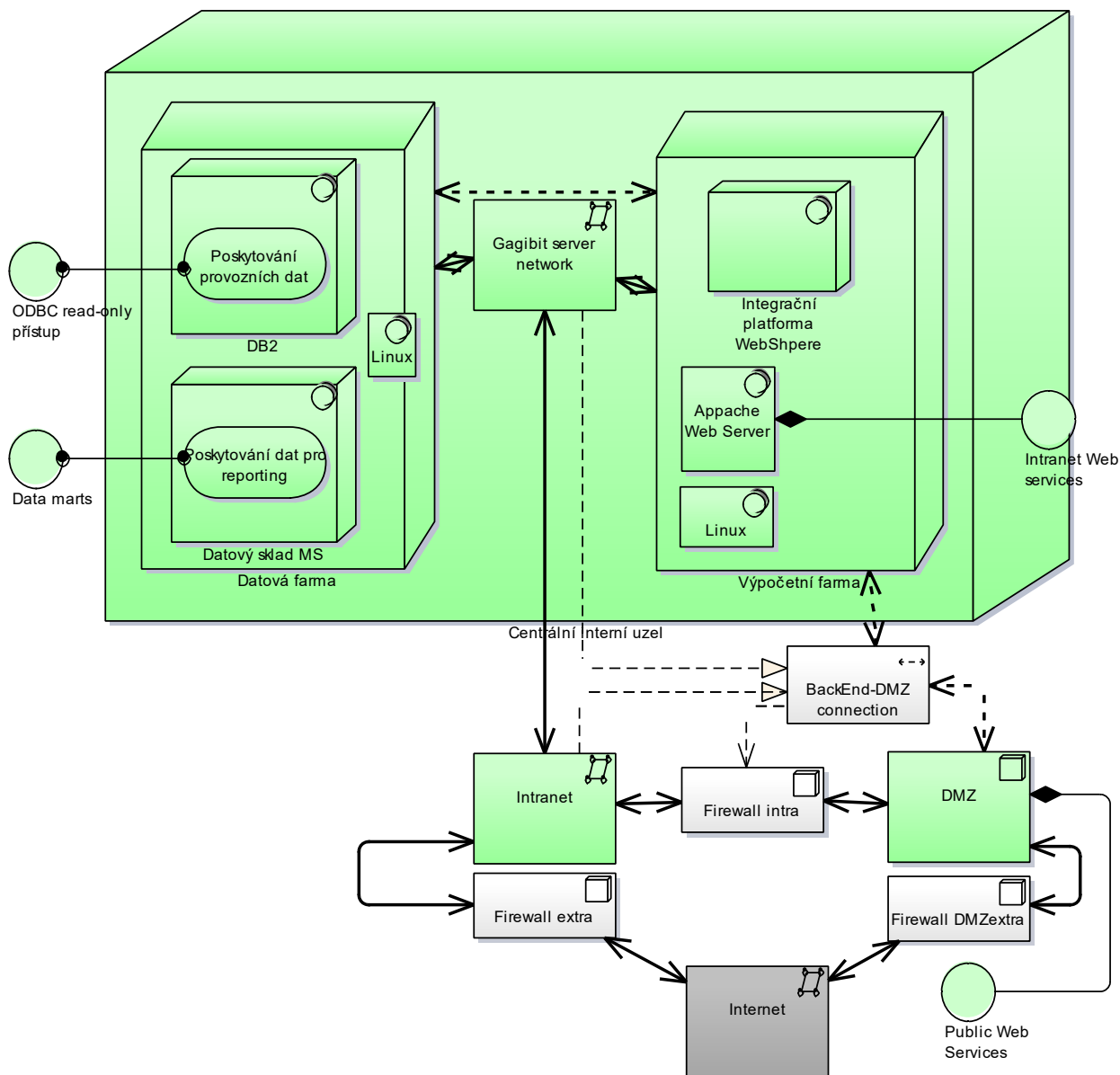
proto často složeným elementem a dílčími součástmi jsou zase elementy typu materiál.

Distribution network
Distribuční síť

Distribuční síť je fyzická síť, která slouží k přesunu materiálu nebo energie.
Distribuční síť má formu elementu nebo vazby mezi elementy. Síť může být na jedné úrovni abstrakce složený element, který se v podrobnějších schématech rozpadá na dílčí síť.

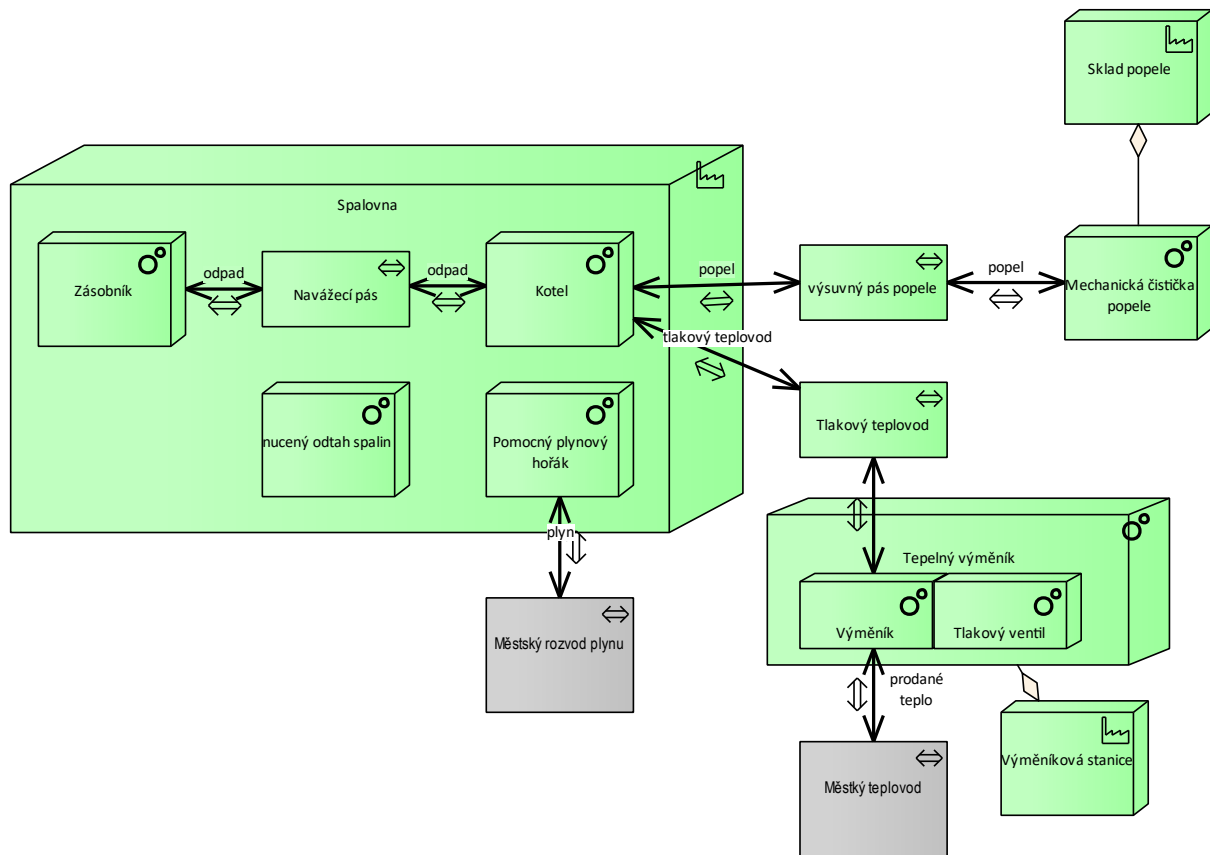


3.2.2. Příklady technologické vrstvy



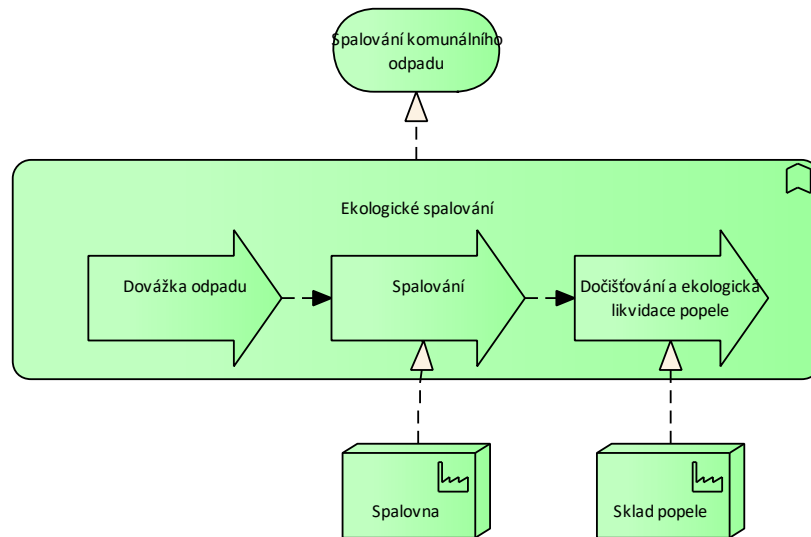
Obrázek 10 Příklad implementace prostředí pro interní systém a propojení do internetu

Poznámka: Komunikační cesta je implementována fyzickými sítěmi. Pro její provoz mezi DMZ a interními aplikacemi je nutný firewall, který ji fakticky realizuje. Protože ale ArchiMate vazbu realizace mezi uzlem a konektivitou nepředpokládá, je použita obecná vazba závislosti (Dependency) elementu na jiném elementu (z hlediska ArchiMate speciální typ vazby asociace).



Obrázek 11 Příklad modelu fyzické výroby (spalovna odpadu)

Příklad spalovny odpadů ukazuje, jak je ArchiMate připraven na modelování fyzických struktur a tak – i když na logické úrovni – vytvářet modely až ne úroveň fyzických zařízení.



Obrázek 12 Ukázka technologického procesu (spalování odpadu)

Příklad technologického procesu demonstruje, že modelování na úrovni technologií je velmi podobné modelování na úrovni businessu. Je třeba zmínit jedinou odlišnost, ukázkový technologický proces je v principu kontinuální činností, kde všechny kroky probíhají současně a to po většinou doby běhu zařízení. Proto mezi součástmi procesu nejsou vazby *triggerring*, ale vazby toku (*flow*).

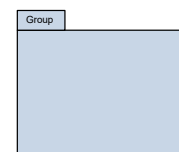
3.3. Elementy mimo vrstvy

ArchiMate definuje pár elementů, které nejsou zařazeny do žádné vrstvy a logicky mohou být používány ve spojitosti s modely každé vrstvy

Location	Prostorové umístění
Lokalita	Popisuje, kde se konkrétní element vyskytuje (při spojení se strukturálním elementem) nebo kde k určitému ději dochází (při spojení v elementem popisujícím chování)



Grouping	Sdružuje objekty s nějakou společnou charakteristikou (bez jejího bližšího určení).
Seskupení	Jde o element sloužící zlepšení přehlednosti modelu bez ekvivalentního elementu v reálném businessu ⁹



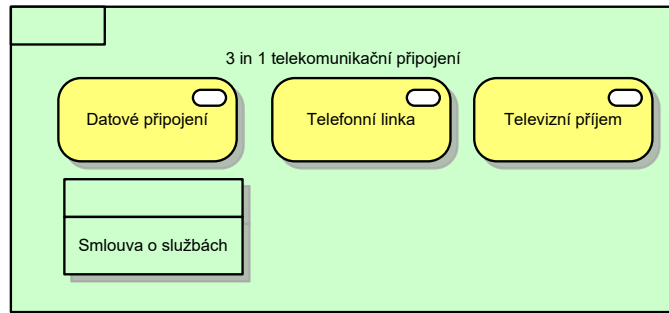
⁹ Poznámka k implementace ArchiMate v3 v nástroji Enterprise Architect (EA): EA podporuje ArchiMate 3 od verze 13. Verze 3 je v řadě případů vylepšením oproti předchozí verzi 2, ale v principu jde o jinak definovaný jazyk a není vhodné kombinovat oba jazyky společně.

Bohužel, implementace skupiny (grouping) se v EA mezi verzemi ArchiMate 2 a 3 výrazně liší, protože ve verzi 2 se s nimi dalo pracovat jako s elementem, kdežto ve verzi 3 jde o vizuální ohraničení, ale ne o element. Tím se použitelnost skupiny výrazně zredukovala (např. k ní nejde přidat popis). Její využitelnost je proto dosti omezená (týká se EA, verze 13). Přesto není vhodné využívat skupinu z jazyka ArchiMate v2, protože s tím v celém modelu vznikají zmatky ohledně rozlišování, v kterém jazyce je model vytvářen.

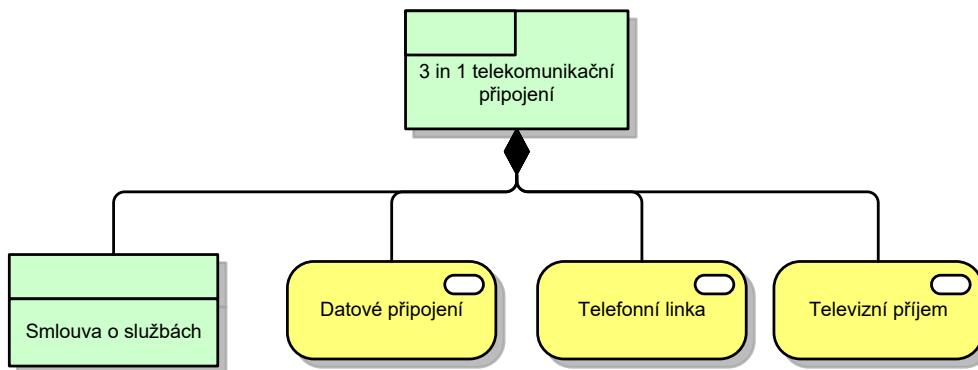
4. Vazby v jazyce

4.1. Strukturální Vazby

Vazby vyjadřují vztahy mezi různými objekty. Standardně se vyjadřují čarami spojujícími dva elementy ve vzájemném vztahu. ArchiMate umožňuje i intuitivní zakreslení přímým vnořením jednoho objektu do jiného



Obrázek 13 Zakreslení kompozice produktu ze služeb



Obrázek 14 Kompozice zakreslená explicitními vazbami

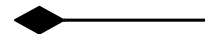
Nevýhodou zápisu zapouzdřením je, že nevyjadřuje explicitně, v jakém vztahu je nadřizený a podřizené objekty. Zápis může vyjadřovat jak kompozici, tak agregaci.

Composition
Kompozice, složení

Vztah vyjadřující, že objekt se skládá z jiných objektů.

Jeden objekt může být součástí kompozice pouze jednoho nadřizeného objektu. Kompozice je v modelech většinou implicitně vyjádřena vnořením objektu do nadřizeného objektu (ať už viditelně na schématu nebo v rámci hierarchie elementů v modelu)

Kompozice může seskupovat objekty různého typu.



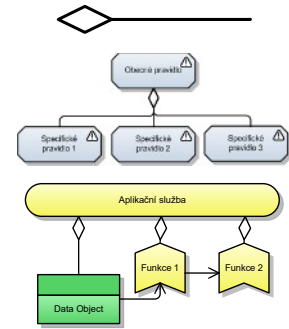
Aggregation
Agregace, Seskupení

Vztah vyjadřuje, že konceptuální třída sdružuje více jiných tříd.

Jeden objekt může být ve více různých seskupeních.

Agregace není ekvivalentní s obecnou třídou v UML. V ArchiMate agregace má obecnější význam a může být použita na popis vazby mezi vnořenými elementy (které popisují, jak je obecnější služba realizována) a jejich nadřazeným elementem, který jejich společný význam zapouzdřuje do jediného elementu vyšší úrovně obecnosti.

Agregace může seskupovat elementy různého typu – povolené agregace jsou uvedeny v meta-modelu.



Assignment
Přirazení

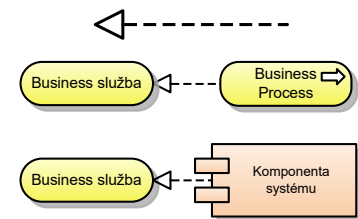
Vazba mezi aktivními prvky a činnostmi, které něco vykonávají (např. mezi rolí a funkcí)



Realization
Realizace

Vazba mezi entitou, která fyzicky realizuje jinou entitu. (např. mezi procesem a funkcí, kterou proces naplňuje, datovým objektem z datového modelu, který realizuje business objekt z business modelu)¹⁰

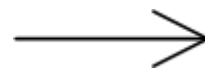
Realizovány jsou zejména služby (business i aplikační) a procesy



Serving
Slouží

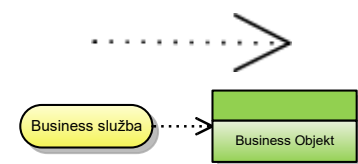
Vazba mezi elementy, které se vzájemně využívají (typicky mezi službami, které využívají funkce, nebo procesy využívajícími rozhraní rolí).

Pozor na orientaci. Šipka míří od objektu, který je využíván (used by), k objektu, který jej využívá



Access
Přístupuje

Vyjadřuje, že určitý prvek chování (proces, funkce apod.) přistupuje a využívá informační nebo business objekt.



Association
Spojení

Obecné spojení vyjadřující jakýkoli vztah, který není postižitelný jiným typem vazby

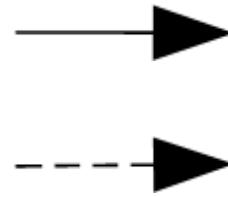


4.2. Dynamické vazby

Dynamické vazby vyjadřují souvislosti, které nastávají v průběhu běhu businessu. Logicky odpovídají vazbám procesního modelu, který je základním nástrojem popisu průběhu. Proto jde o vazby mezi funkčními entitami – procesy, funkcemi, spoluprací a událostmi.

¹⁰ Vazba se často obsahově překrývá s vazbou Trace a je třeba se rozhodnout, která se v modelech používá. Trace v ArchiMate není definováno, takže v rámci jazyka mezi vazbami rozpor není. Ten ale nastane ve chvíli, kdy se míchá více různých jazyků v rámci modelu.

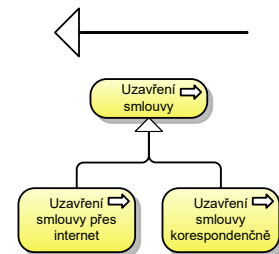
Triggering	Vyjadřuje příčinnou nebo kauzální souvislost
Spouštění	Vazba je základní vazbou procesního toku (workflow)
Flow	Vyjadřuje informační nebo hodnotový tok
Tok	Odpovídá toku v pojetí BPMN jazyka



4.3. Další vztahy

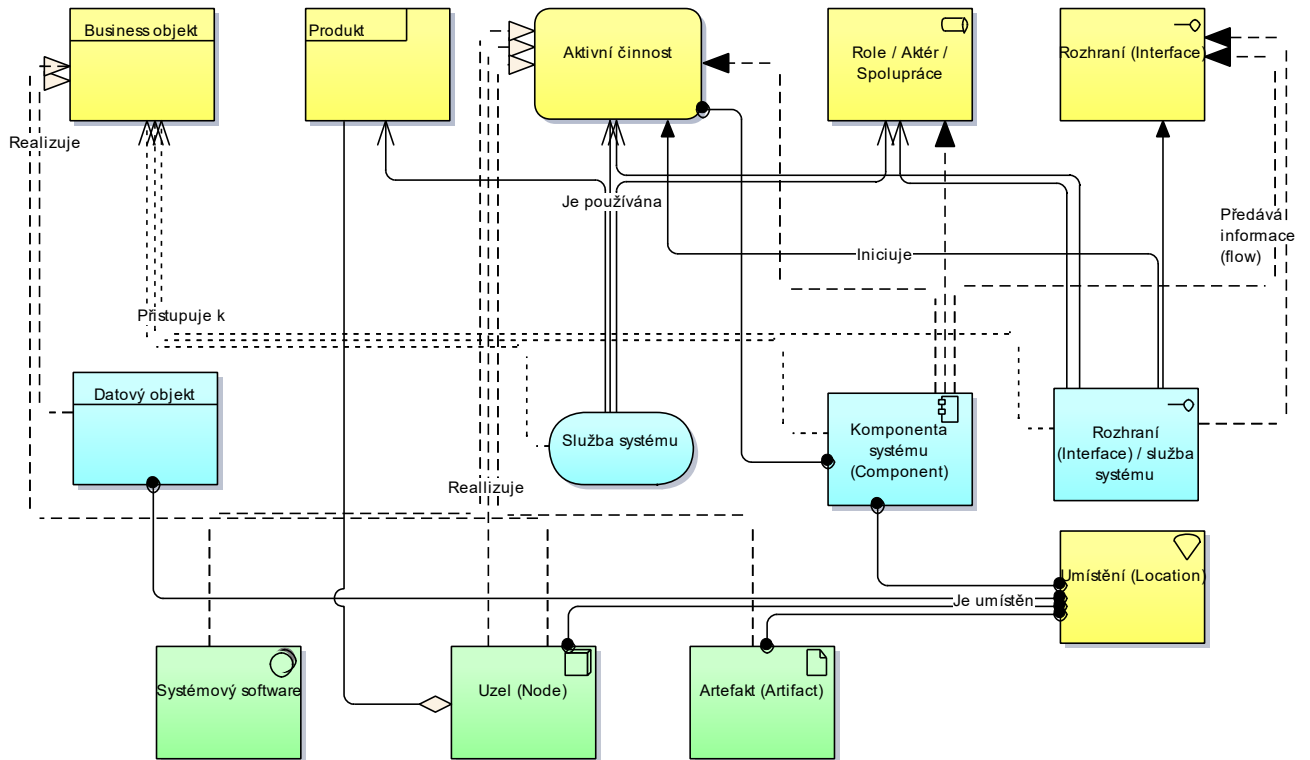
Oddíl shrnuje vazby, které nespádají logicky do předchozích skupin, i když vzájemně nemají společného nic.

Junction <i>Spojení</i>	<p>Spojení dynamických vazeb vyjadřující synchronizaci nebo rozdělení (join, split)</p> <p>Spojení se používá pouze mezi dynamickými vazbami</p> <p>Spojení má obecně buď význam rozhodnutí / NEBO / OR (kroužek) nebo sjednocení / A / AND (plné kolečko)</p>
Specialization <i>Specializace</i>	<p>Vazba vyjadřující, že jeden element je speciálním případem jiného elementu.</p> <p>Vazba je možná pouze mezi elementy stejného typu.</p>

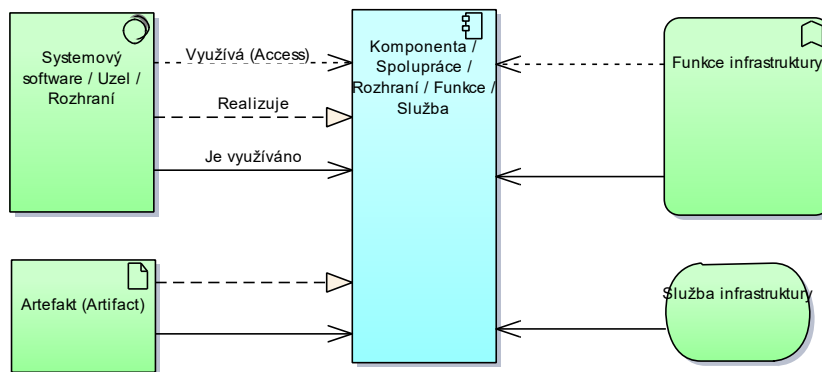


4.4. Vertikální vazby - provázání vrstev

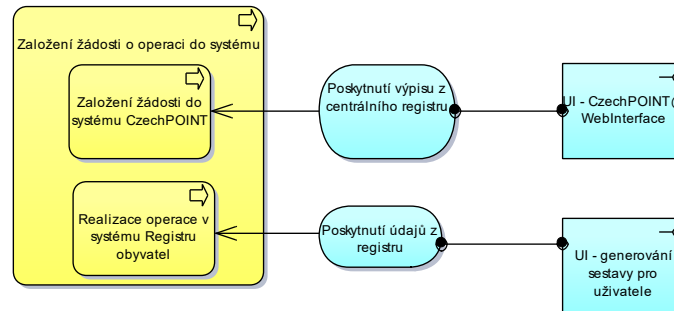
4.4.1. Implementace business a technologické vrstvy



Obrázek 15 Vertikální vazby business vrstvy na nižší vrstvy



Obrázek 16 Vertikální vazby implementační vrstvy na technologickou



Obrázek 17 Business elementy používají elementy aplikační vrstvy

Pro provázání vrstev mezi sebou se používají zejména vazby:

<p>Used By <i>Je používán</i></p>	<p>Element vyšší vrstvy využívá element nižší vrstvy, který jej implementuje</p> <p>Typicky aplikační služba nebo rozhraní je používáno business operací nebo procesním krokem</p>	
<p>Realization <i>Realizuje</i></p>	<p>Element nižší vrstvy realizuje objekt vyšší vrstvy</p> <p>Je třeba konzistentně používat Used By a Realization; Standard předpokládá použití realizace mezi business a datovým objektem</p> <p>Realizována může být business služba, pokud aplikační služba / funkce ji realizuje plně</p>	
<p>Assignment <i>Přiřazení</i></p>	<p>Vazba elementu vyšší vrstvy k elementu nižší vrstvy, který element vyšší vrstvy realizuje</p>	
<p>Aggregation <i>Agregace</i></p>	<p>Vyšší element je agregován elementem nižší vrstvy, když jej tento element plně realizuje</p> <p>Příkladem smysluplné agregace jsou business služby, kterou plně poskytuje např. webové rozhraní aplikace; Tato aplikace pak může agregovat přímo business služby a není třeba vytvářet stejnojmenné aplikační služby, které realizují business služby</p>	

5. Popis motivace

Motivace je třeba chápat ve více rovinách, všechny jsou pro úspěšný projekt důležité:

- Business motivace - cíle a strategii organizace; její sestavení je v kompetenci vrcholového vedení, analytik ji ovšem potřebuje znát; její popis v rámci Enterprise analýzy slouží pro udržení vazeb mezi projektovými a business cíli a strategií
- Motivaci samotného projektu; analyzovat a řádně dokumentovat motivaci projektu je plně v kompetenci business analytika a standardně je součástí *Project Charter*, nebo-li zadání projektu.
- Třetí relevantní oblastí je IT - ta je v kompetenci CIO, vedení IT týmu. Z hlediska business analýzy projektu z IT strategie typicky vyplývá řada funkčních i mimofunkčních požadavků a omezení a v rámci návrhu nového řešení je třeba s ní aktivně pracovat. Popis IT strategie by proto měl být nedílnou součástí Enterprise Architektury.


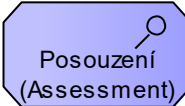
Důležité je uvědomit si jiný význam popisu motivace oproti klíčovým vrstvám. Zatímco klíčové vrstvy (business, aplikační a technologická) popisují enterprise architekturu, která buď je nebo se má vytvořit, motivace popisuje, proč se má vůbec nová architektura (nebo také nové změny) dělat a jaké jsou pro to důvody.

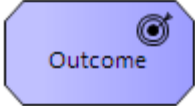
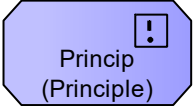
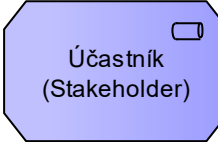
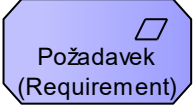

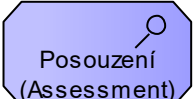
5.1. Vyloučené elementy strategického plánování

Poznámka k rozsahu: Definice jazyka kromě motivačních elementů popisuje i elementy strategického modelování, konkrétně zaměření, schopnost a zdroj. Strategické modelování má umožnit vytvářet strategickou mapu, která modeluje, jaké schopnosti organizace budou změnou posíleny, jaké zdroje jsou potřeba pro kterou schopnost a jak je projekt vytváří nebo uvolňuje a směřování organizace popisuje dlouhodobou strategii firmy, jakým způsobem se chce rozvíjet. Vzhledem k tomu, že analytici, pro které je příručka určena se strategickým plánování na úrovni stanovování korporátních strategií nejen, že standardně nezabývají, ale ani nejsou odpovědní za to, aby analyzované změny s nimi byly v souladu, nejsou elementy prakticky využívány. Proto nejsou ani součástí této příručky.

5.2. Elementy popisu motivace

Elementy z oblasti motivace jsou určeny pro obě zmíněné roviny.

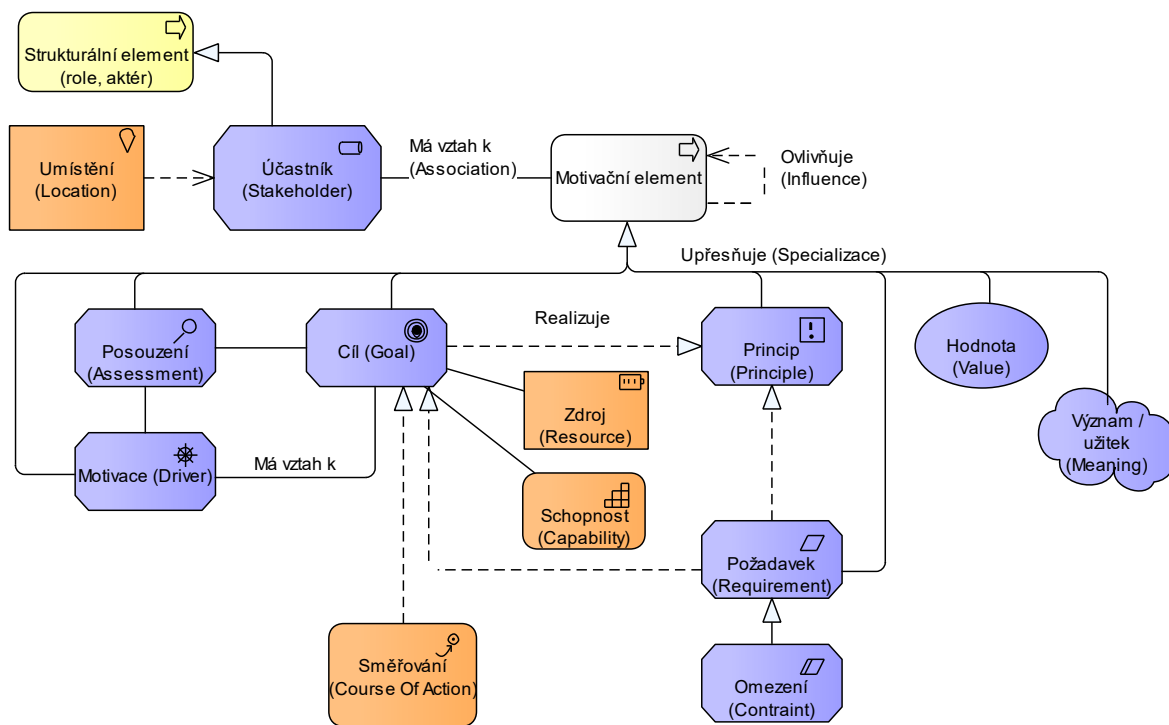
Goal <i>Cíl</i>	Cíl, kterého má být dosaženo nějakým úsilím. Může jít o projektový cíl, nebo business cíl	
Driver <i>Důvod změny</i>	Důvod změny je fakt, který vede k organizaci k tomu, aby změnu (potažmo projekt) realizovala Důvod změny je typicky spojen s rolí - někým, pro koho je změna důležitá Driver zahrnuje i pojem <i>concern</i> , tedy interní záměr, Některé implementace jazyk podporují <i>concern</i> jako samostatný element, což může umožnit jemnější rozlišení, na druhou stranu je	

	odlišnost natolik malá, že spíše vznikají pochybnosti, jestli využít zájem coby specializaci specializaci důvodu změny.	
Outcome Výstup	<p>Výstup je výsledek činnosti, kterého má být dosaženo. Výstupy jsou realizovány na základě požadavků, které definují, jaký výstup má být.</p> <p>Využívá se pouze ve vztahu k projektu</p> <p><i>I když tomu název může nahrávat, výstup není produkt na technologické úrovni, něco, co by bylo na vstupu technologického procesu. Výstupem je myšleno to, co účastníci modelovaného projektu zamýšlejí jako výstup daného projektu.</i></p>	
Principle Princip, pravidlo	<p>Obecná vlastnost, kterou musí určitá skupina objektů nebo činností splňovat.</p> <p>Může jít o princip chování v businessu (<i>např. korektnost poskytovaných informací</i>), princip designu softwarového řešení (<i>principy SOA architektury</i>) nebo principy projektové práce (<i>nepřekročení rozpočtu</i>)</p>	
Stakeholder Účastník	<p>Aktér (osoba, tým nebo organizace) s konkrétním zájmem k projektu.</p> <p>Typicky se využívají aktéři, rozlišení Stakeholder a Actor umožňuje lepší přehlednost schémat, pro téhož aktéra se ale pak používají různé symboly, což nevádí, pokud modelovací prostředí toto umožňuje</p>	
Requirement Požadavek	<p>Vyjadřuje konkrétní vlastnost, kterou musí některý element architektury splňovat</p> <p>Používá se pouze v kontextu projektu</p>	
Constraint Omezení	<p>Překážka nebo limit, který musí být brán v potaz při řešení, protože jej není možné ignorovat. Příkladem omezení je rozpočet projektu, schopnosti lidí, omezení výběru technologie apod.</p> <p>Rozdíl proti požadavku je v tom, že požadavek vznáší nějaké předpoklady o výstupu projektu, co má výstup splňovat, kdežto omezení říká, co musí být zohledněno v samotném procesu tvorby výstupu.</p>	
Assessment Posouzení	<p>Výstup z analýzy určitého systému nebo elementu. Posouzení shrnuje silné a slabé stránky, hrozby a příležitosti, které z oblasti vyplývají</p>	

<p>Meaning <i>Význam</i></p>	<p>Význam určité reprezentace informací pro konkrétní roli. Určitá reprezentace může mít (teoreticky) různé významy pro jednotlivé role</p>	
<p>Value <i>Hodnota</i></p>	<p>Hodnota reprezentuje smysl elementu (i výrobku), pro který si externí subjekt danou službu pořizuje¹¹.</p>	
<p>Resource <i>Zdroj</i></p>	<p>Zdroj je aktivum organizace, Zdroje jsou v rámci strategie vytvářeny; Pro projekty jsou potřebné, aby mohly být projekty úspěšně dokončeny. Před zdroje jsou často projekty provazovány, protože je sdílejí, nebo jeden projekt čeká, až jiný zdroj vytvoří.</p>	
<p>Capability <i>Schopnost</i></p>	<p>Schopnost organizace dosáhnout vytyčených cílů nebo poskytovat hodnotu (službu). Projekty jsou typicky zaměřeny na budování nových schopností nebo na jejich <i>perfekcionalizaci</i> (snižování nákladů, zlepšování kvality – hodnoty měřené assessmentem)</p>	
<p>Course of action <i>Směrování</i></p>	<p>Směrování (přístup) k dosažení nějaké schopnosti či zdroje. Vyjadřuje, co se organizace rozhodla udělat pro to, aby zvoleného cíle dosáhla. Typicky zadání projektu.</p>	

¹¹ Je třeba nezaměňovat business hodnotu s informační hodnotou určité entity (např. číselným vyjádřením počtu kusů). Hodnotou např. auta je schopnost přemístit majitele, reprezentovat jej nebo (v případě veterán) uchovat jeho majetek. Jestliže se na obchod díváme jako na směnu hodnot, tak služba či reprezentace entity má svou hodnotu v tom, co příjemce očekává jako reciprocitu za peníze, které za tyto služby zaplatil.

5.3. Metamodel rozšíření motivace

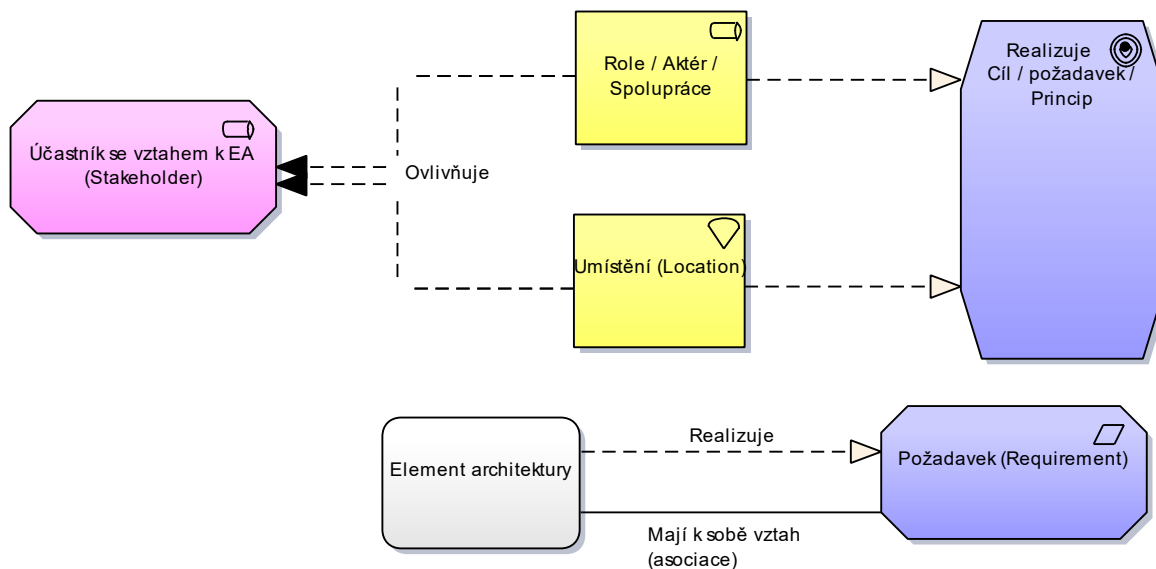


Obrázek 18 Metamodel rozšíření *Motivace*

Základními dvěma vztahy mezi elementy motivace jsou **Vzájemné ovlivňování**, které může být pozitivní nebo negativní.

Jak je z metamodelu patrné, principy jsou nadřazeny cílům a požadavkům - dají se přirovnat k přírodním zákonům, tedy něčemu, co se dodržuje a v rámci architektury uznává.

5.3.1. Realizace strategie a vize



Obrázek 19 Vertikální vazby strategie a vize do business vrstvy

Požadavky, strategie a vize jsou nehmotné elementy – ideje, které na své naplnění čekají (v případě vizí) nebo musí být splňovány (požadavky). Naplnit vizi může pouze člověk, proto vize a postupy jsou realizovány pouze elementy popisujícími lidi.

Požadavek je konkrétním popisem toho, co musí některý z elementů architektury naplňovat. Může se proto vztahovat k libovolnému elementu. Požadavek je buď naplněn (realizován) nebo má obecnější vztah asociace, kterým se vyjadřuje, že daný element je požadavkem nějak ovlivněn, i když jej přímo nerealizuje.

Vertikální vazby realizace projektu jsou popsány v rámci metamodelu rozšíření plánu projektu na Obrázek 20.

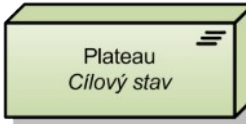

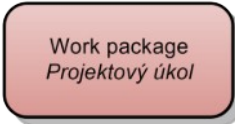
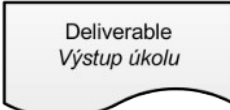
6. Popis projektů a programů – implementační rovina

Plánování postupu prací v čase a jejich rozfázování do etap je typicky jednou z kompetencí projektového řízení, ale business a funkční analýza dává klíčové podklady, ze kterých plánování vychází.

ArchiMate nehovoří o elementech a úhlu pohledu jako o popisu projektů, ale hovoří o implementaci a migraci. Reálně jsou ale všechny činnosti realizovány v rámci projektů a plánování přechodů z jednoho stavu architektury do jiného plně odpovídá plánování projektu změny organizace. Proto tuto oblast označujeme v souladu se zažitou terminologií jako popis projektů.

6.1. Přehled elementů popisu činností

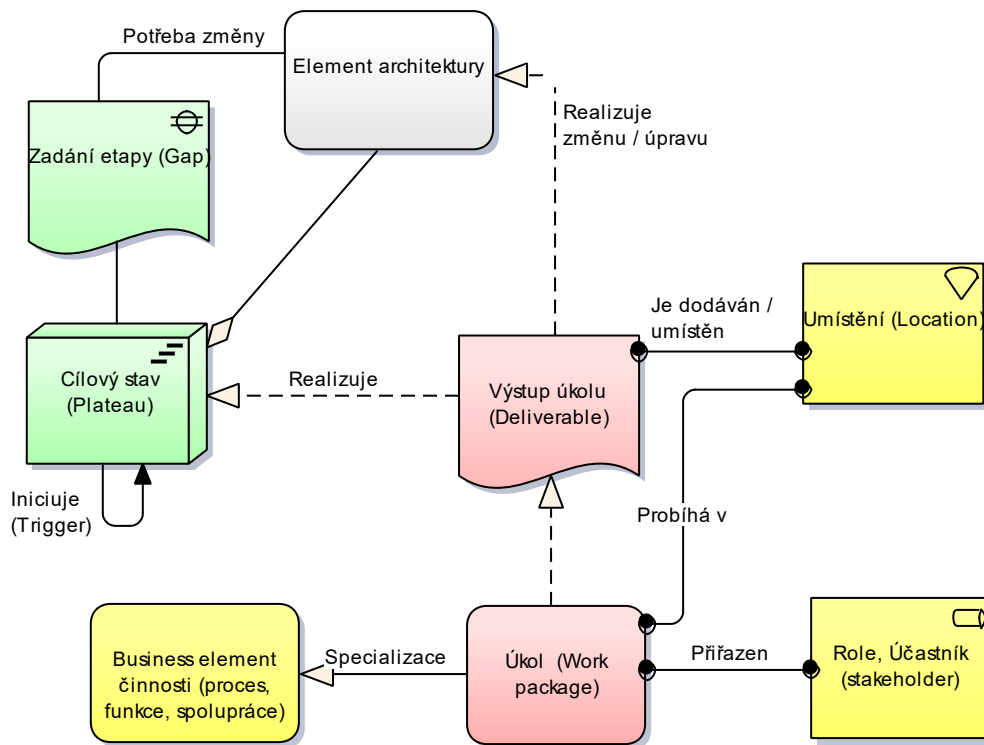
Elementy, o které se ArchiMate rozšiřuje pro oblasti projektového modelování:

Plateau Cílový stav etapy Stabilní stav řešení	Stabilní stav systémů po realizaci etapy projektu. Obecně nemusí být cílové stavy vázány na etapy, ale definují přechodové stavy, na kterých je možné stavět dále - může jít proto např. o etapu analýzy, prototypu apod.	
Gap Etapa (Zadání etapy)	Výstup z analýzy, co je třeba udělat pro přechod mezi dvěma cílovými stavy (plateau)	
Work Package Projektový úkol	Sada aktivit, které vedou k dosažení cíle projektu v daném čase. Projektový úkol má blízko k procesu ale s tím, že typicky jde o proces jednorázový. Pokud existuje např. projektová metodika, je úkol specializací procesu z metodiky upraveném pro daný projekt (tailoring).	
Deliverable Výstup projektu	Výstup z pracovního úkolu ¹²	

¹² Je nasnadě otázka, jak se vlastně liší element *cíle* z popisu motivace od elementu *dodávky* ve vrstvě implementace. Odpověď je právě v rozlišení vrstev resp. úhlů pohledu. Cílem se popisuje, kam projekt směřuje předtím, než je realizován a plánován.

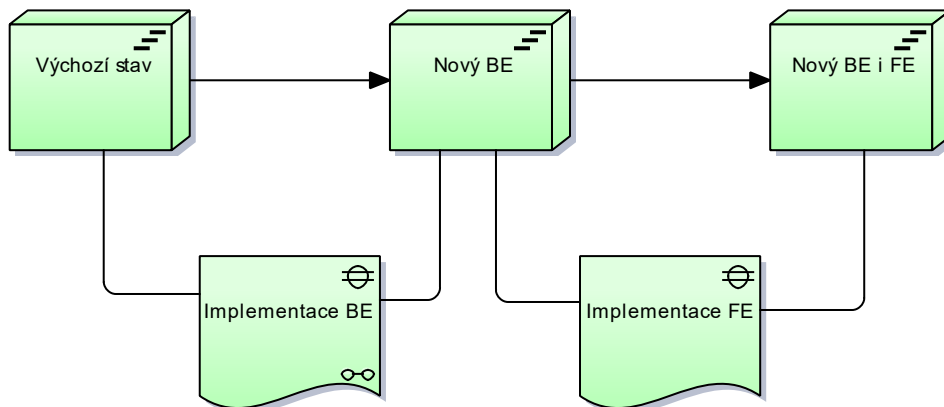
Výstupy a pracovní činnosti jsou praktickým plánem, jak dosáhnout (vzdáleného) cíle etapou praktických, realizovatelných kroků.

6.2. Metamodel plánu projektu

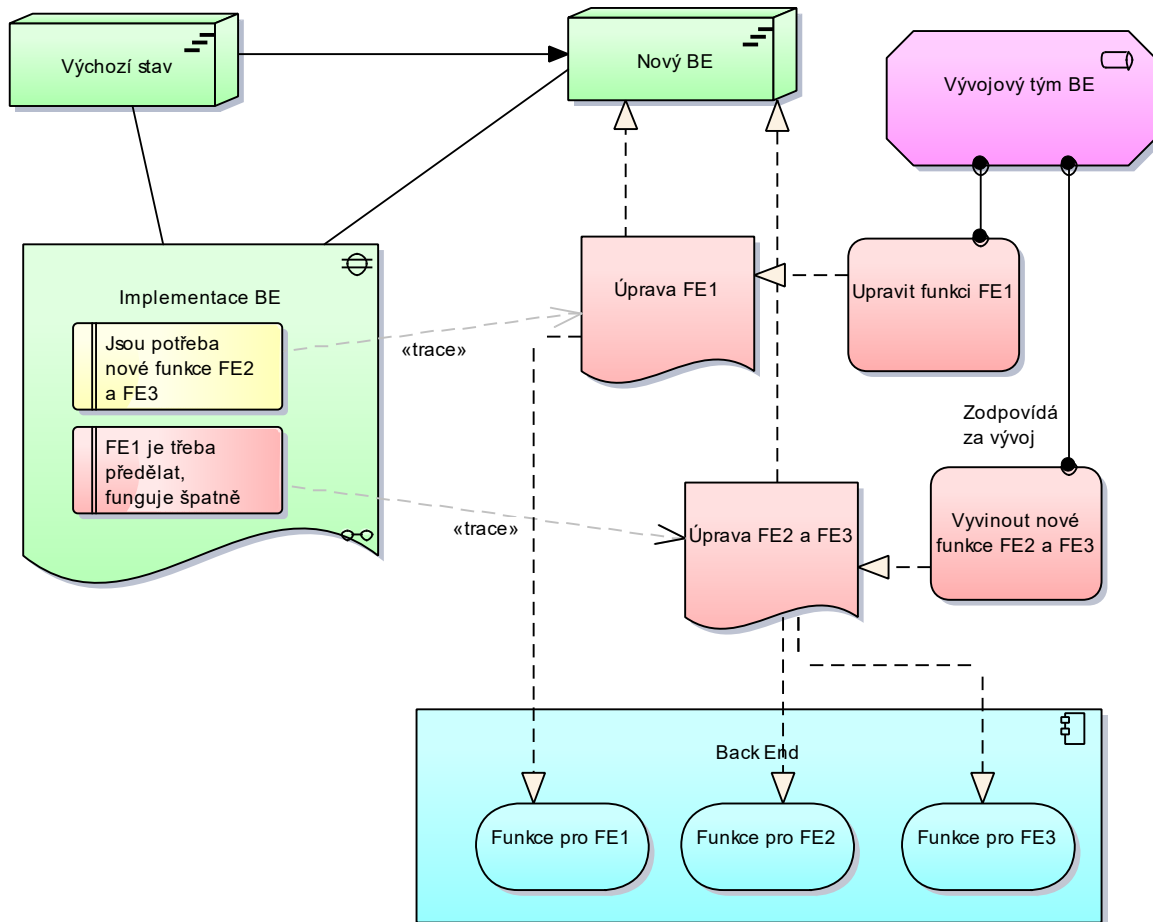


Obrázek 20 Metamodel rozšíření Implementace s vazbami na Enterprise architekturu

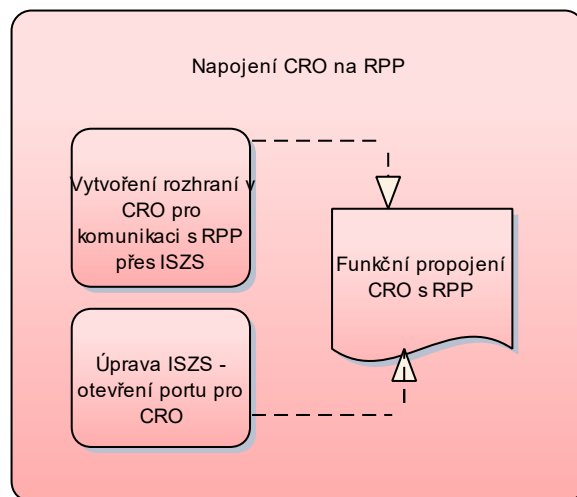
6.3. Příklad popisu projektu



Obrázek 21 Příklad rozdělení projektu na 2 etapy



Obrázek 22 Příklad: Plán realizace etapy projektu



Obrázek 23 Ukázka plánu úkolu projektové etapy

Poznámky ke schémátům:

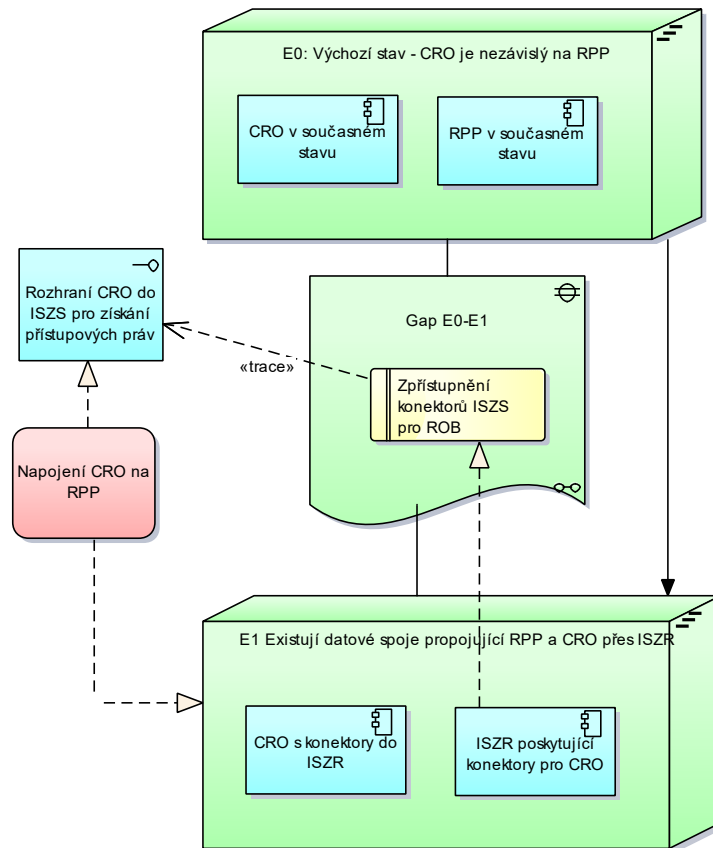
Gap analýza v ArchiMate nemá definované elementy, na které by se dále členila. Typickým výstupem Gap analýzy je identifikace toho, co je třeba změnit (change) a co je špatně (issue), proto jsou také tyto elementy

použity v ukázkovém schématu. Vazba Trace, stejně jako elementy Gap analýzy nevyplývá a ArchiMate, ale je fakticky asociací.

V praxi se trojice elementů – Change/Issue, Delivery (dodávka) a Work Package (úkol) – jeví jako nadbytečná. Element Change plně postačí pro popis, co je třeba změnit a může být přímo spojen s elementem architektury, který se bude měnit.

Úkoly pro jednotlivé změny má smysl definovat, pokud se přenášejí do systému úkolů.

Následující obrázek je zjednodušujícím modelem, který ale nesplňuje syntaxi ArchiMate. Komponenty v plateau jsou instance komponent z architektury, což umožňuje přesnější popis jejich stavu.



Obrázek 24 Ukázka zjednodušeného modelu projektu



7. K čemu ArchiMate

ArchiMate, podobně jako třeba UML, SysML je „jenom“ jazykem. Nástrojem pro zaznamenání skutečnosti i návrhu. Každý z jazyků je dobrý na něco, na něco jiného naopak ne.

7.1. Zhodnocení ArchiMate

7.1.1. V čem je ArchiMate silný

Jeho nejsilnější stránkou je vrstva business a aplikační analýzy a to zejména pro „high level“ pohled. Je relativně dobře srozumitelný, takže s troškou vysvětlení je možné schémata předložit i laikovi.

Na strategické a business úrovni, pokud pracujete na návrhu firemních procesů, systémů a fungování firmy obecně, je ArchiMate stále tím nejlepším, po čem můžete sáhnout. Zejména entity jako cíle, business služby nemají v žádném jiném rozšířeném jazyce tak jasné postavení.

7.1.2. V čem je průměrný

Detaily návrhu a přesnost

Z hlediska business úrovně narazíte na limity jazyka, když se budete dělat detailní analýzu. Typicky je to vidět na procesních modelech. Pohled na základní průběh procesu navrhnete v ArchiMate snadno, ale když budete mít za úkol cokoli s procesem dělat, narazíte na malou přesnost jazyka. Pokud můžete, asi sáhnete pro BPMN, který je přesnější

Na aplikační i systémové úrovni je ArchiMate stále docela dobrým nástrojem. Má na těchto úrovních stejné omezení, jako pro business úroveň. Zatímco ale v business úrovni to tak zásadně nevádí, protože lidé z businessu s analytikem v debatě do detailů buď nepůjdou, nebo si popis raději přečtou v textu, s návrhem aplikací to tak není. Jako analytik budete diskutovat s technicky myslícími lidmi a ti chtějí přesnost a detail, které ArchiMate nejsou úplně vlastní.

I když můžete v ArchiMate popisovat i detaily jako rozhraní, aplikační procesy a scénáře komunikace, jazyk a schémata vám nepomohou být dostatečně precizní. Brzy tak budete muset model doplňovat např. SysML nebo UML.

Modelování projektu

Modely projektu a přiřazení, která činnost se provede v kterém projektu, je konceptuálně zajímavé téma, ale v praxi v podstatě nemá využití. Mnohem důležitější je kvalitně fungující projektová databáze, správně balíkové požadavky zařazené v etapách.

Logiku vedení projektu a zpracování požadavků má lépe zpracováno model BaBOK¹³, kterému logicky přísluší se tím více zabývat..

7.1.3. V čem je ArchiMate slabý

Svět mimo ERP

ArchiMate rozhodně není nástroj pro design, ani detailní analýzu. Má v názvu „architecture“, ještě by se hodilo říct „business architecture“. Rozhodně se nehodí pro architekturu něčeho, co s není business. Technické výrobky, hry a jiné aplikace, které nejsou podnikovým softwarem, tam se ArchiMate nehodí.

¹³ BaBOK = Business Analysis Body Of Knowledge. Srovnávání BaBOK a ArchiMate by bylo mícháním hrušek s jablky, protože ArchiMate je jazykem, kdežto BaBOK metodikou.

Požadavky

Poměrně výraznou a z hlediska použití překvapivou slabinou ArchiMate jsou požadavky. V ArchiMate požadavky jsou, ale stojí vedle všeho ostatního. Přitom pro business požadavky jsou a i do budoucna budou klíčové. Jsou přetavením *use case* a *user stories* do toho, co je potřeba udělat.

Řešit požadavky, to není jenom o tom je zaznamenat. K požadavkům je klíčový proces, jak s nimi pracovat a jak s nimi také dlouhodobě udržovat celý model aktuální. A v tom není slabý jenom ArchiMate, ale v zásadě všechny analytické systémy. ArchiMate má ale v hodnocení pomyslné záporné body ze dvou důvodů:

- U nástroje pro enterprise analýzu je problém zásadnější, protože enterprise modely se běžně udržují i 20 let a během té doby se business velmi změní.
- Modely projektu a entita pro implementaci dávají tušit, že ArchiMate správu modelu uspokojivě řeší, ale *implementace* k tomu nijak nepomáhá.

Jak se starat o analýzu přesahuje záběr referenční příručky a věnujeme se mu na školeních,

7.2. Jak s ArchiMate pracovat

Sdílení modelu

ArchiMate je jazykem modelování, ale ještě důležitější, než mít jak modelovat, je model komunikovat. Zpracování analýzy do šuplíku bez toho, že se všechny jeho části sdílejí, revidují a aktualizují, je práce sice pro mnohé zábavná, ale málo efektivní. Modelování má smysl jen pokud můžete informace z modelu sdílet se všemi, kterých se týkají – manažery, vlastníky procesů, vedoucí týmů, které zajišťují každodenní fungování firmy.

Ke sdílení potřebujete, abyste model měli dostupná v nástroji, který sdílení umožní. [AyMINE](#) je k tomu připraven. Pomocí Enterprise Architect můžete modelovat v ArchiMate i jiných jazycích a AyMINE vám zajistí, aby se k informacím skutečně všichni dostali. Brzy zjistíte, že nad samotným modelem trávíte méně času, ale díky spolupráci je výsledek lepší.

Metodika analýzy a architektury

Celou metodiku práce analytika se snaží postihnout model BaBOK. ArchiMate proto standardně školme práce jako součást cyklu analýzy, který je postaven na BaBOK modelu (odkaz je v patičce).

ArchiMate se snaží postihnout i enterprise architekturu. Metodikou práce s ní se zabývá TOGAF, ale také ITIL, Cobit a dílem i starší Zachman Framework. Podrobnosti o všech i možnost spolupráce na architektuře najdete na [stránce o enterprise architektuře zde](#).

