

Ekonomická  
fakulta  
Faculty  
of Economics

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích  
Ekonomická fakulta  
Katedra aplikované matematiky a informatiky

Bakalářská práce

# Využití modelovacích jazyků UML a ArchiMate při analýze IT projektu

Vypracoval: Zdeněk Holeček, DiS.

Vedoucí práce: Mgr. Radim Remeš, Ph.D.

České Budějovice 2024

# JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Ekonomická fakulta  
Akademický rok: 2022/2023

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Zdeněk HOLEČEK  
Osobní číslo: E21295  
Studijní program: B0688A140010 Podniková informatika  
Téma práce: Využití modelovacích jazyků UML a Archimate při analýze IT projektu  
Zadávající katedra: Katedra aplikované matematiky a informatiky

### Zásady pro vypracování

Cílem práce je představit možnosti využití různých modelovacích jazyků při analýze zadání, definici požadavků, návrhu konceptuálního řešení a funkčních vlastností cílové aplikace v rámci středně rozsáhlého IT projektu. Demonstrace bude provedena na příkladu IT projektu pro různé fáze přípravy projektu.

#### Metodický postup:

1. Rešerše a studium postupů při analýzách IT projektů.
2. Příprava obecného popisu modelovacích jazyků Archimate a UML.
3. Vypracování zásad při tvorbě modelu požadovaného IT řešení.
4. Demonstrace na vybraném ukázkovém IT projektu.
5. Zhodnocení použitelnosti modelovacích jazyků pro nasazení v reálném prostředí.
6. Zhodnocení a závěr.

Rozsah pracovní zprávy: 40 – 50 stran  
Rozsah grafických prací: dle potřeby  
Forma zpracování bakalářské práce: tištěná

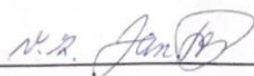
#### Seznam doporučené literatury:

1. Bahri, M., & Williams, J. (2022). *Practical Model-Driven Enterprise Architecture*. Packt.
2. Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide*. 2nd Edition. Addison-Wesley Professional.
3. Fernando, C. (2022). *Solution Architecture Patterns for Enterprise: A Guide to Building Enterprise Software Systems*. Apress.
4. Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture*. O'Reilly Media.
5. Weilliens, T., Lamm, J. G., Roth, S., & Walker, M. (2022). *Model-Based System Architecture*. 2nd Edition. Wiley.

Vedoucí bakalářské práce: Mgr. Radim Remeš, Ph.D.  
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: 20. ledna 2023

Termín odevzdání bakalářské práce: 12. dubna 2024



doc. RNDr. Zuzana Dvořáková Lišková, Ph.D.  
děkanka

JIHOČESKÁ UNIVERZITA  
V ČESKÝCH BUDEJOVICÍCH  
EKONOMICKÁ FAKULTA  
Studentská 13  
370 05 České Budějovice



doc. RNDr. Jana Klícnarová, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 13. března 2023

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum 8. dubna 2024

Podpis studenta

## **Poděkování**

Mé upřímné poděkování panu Mgr. Radimu Remešovi, Ph.D., za jeho podporu a cenné připomínky, které byly zásadní pro formování a finální podobu mé bakalářské práce. Jeho schopnost poskytovat konstruktivní kritiku mi byla během tohoto procesu nesmírně cenná. Rovněž děkuji všem osobám, s nimiž jsem v průběhu tvorby práce spolupracoval, za jejich rady a informace vedoucí ke splnění práce.

Zdeněk Holeček

## Obsah

1	Úvod.....	1
2	Analýza IT projektu .....	2
2.1	Proces analýzy .....	2
3	Nástroje analýzy.....	6
4	Význam modelování v procesu analýzy IT projektu .....	9
5	Modelovací jazyky .....	11
5.1	ArchiMate.....	11
5.1.1	Struktura jazyka .....	11
5.1.2	Vrstvy jazyka .....	12
5.1.3	Aspekty jazyka.....	15
5.1.4	ArchiMate metamodel.....	17
5.2	Unified Modeling Language .....	19
5.2.1	Struktura jazyka .....	20
5.2.2	Diagramy struktury .....	21
5.2.3	Diagramy chování .....	25
5.2.4	Způsob používání UML .....	27
5.3	Možnosti propojení jazyka UML a ArchiMate .....	28
6	Provedení analýzy v konkrétním IT projektu.....	29
6.1	Základní popis prostředí.....	29
6.2	Motivační vrstva.....	30
6.3	Problém a popis současného stavu (As-Is).....	31
6.4	Cílový stav a popis žádoucího stavu (To-Be).....	32
6.5	Business požadavky .....	32
6.6	Business procesy .....	33
6.6.1	Hlavní proces .....	33
6.6.2	Podproces svolání členů POHO.....	34
6.7	Business role.....	35

6.8	Business objekty .....	36
6.9	Požadavky na systém.....	38
6.9.1	Metodologie sběru požadavků použítá na projektu .....	38
6.9.2	Funkční požadavky .....	38
6.9.3	Nefunkční požadavky.....	41
6.9.4	Architektonické požadavky.....	41
6.9.5	Omezení .....	42
6.10	Postup návrhu architektury.....	42
6.10.1	Požadované aplikační služby .....	42
6.10.2	Potřebné aplikační komponenty.....	43
6.10.3	Konceptuální návrh architektury.....	44
6.11	Případy užití .....	45
6.11.1	Přehled případů užití .....	45
6.11.2	Detail konkrétního případu .....	46
6.12	Datový model .....	47
6.12.1	Konceptuální datový model .....	47
6.12.2	Logický datový model.....	48
6.13	Implementace projektu .....	49
6.13.1	Detailní popis případů užití.....	50
6.13.2	Detail komponenty .....	54
7	Závěr .....	55
8	Seznam zkratk .....	56
9	Seznam použitých zdrojů .....	57
10	Seznam obrázků .....	58
11	Seznam tabulek .....	59
12	Seznam příloh.....	60
13	Přílohy.....	61

# 1 Úvod

Cílem této bakalářské práce je ukázat možnosti využití modelovacích jazyků Unified Modeling Language (UML) a ArchiMate při analýze projektů informačních technologií (IT). Snahou je na příkladu z praxe popsat, jak tyto jazyky mohou být efektivně použity při identifikaci potřeb a požadavků, zachycení business procesů a objektů, a jak mohou pomoci při návrhu architektury technického řešení.

V obecné části bude stručně popsána struktura a základní elementy obou jazyků. Budou uvedeny základní možnosti nasazení včetně výhod jejich propojení. Část praktická se bude věnovat ukázce provedení business analýzy konkrétního případu IT změny ve velkém podniku. Nejprve bude ukázáno, jak pomocí modelovacího jazyka zachytit identifikaci prostředí, zúčastněných stran, jejich motivace a hlavní business potřeby. To je klíčovým krokem pro další fáze projektu při zjišťování detailních potřeb a požadavků, protože umožňuje analytikům lépe porozumět kontextu projektu a zajistit, že navrhované řešení bude skutečně relevantní pro všechny zúčastněné strany. Další kapitoly se budou věnovat zachycení business procesů a objektů, identifikace požadovaných aplikačních služeb a případů užití. Všechny tyto kroky jsou nezbytné pro úspěšnou formulaci požadavků a návrh architektury řešení.



## 2 Analýza IT projektu

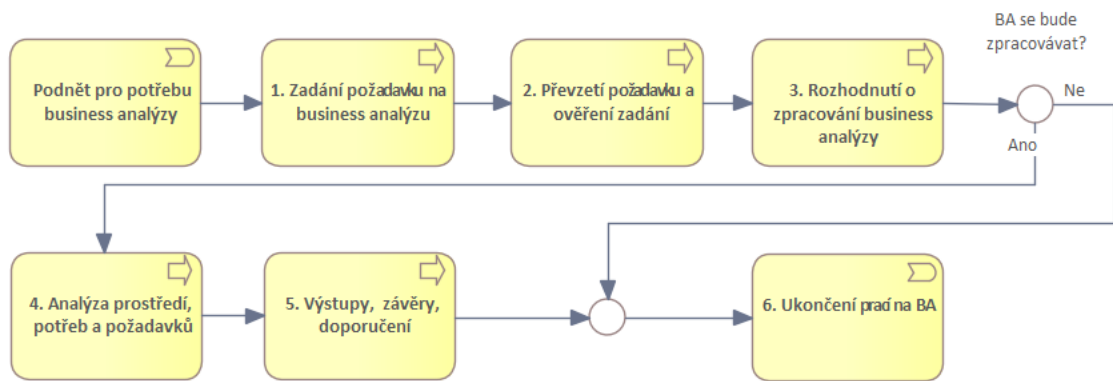
Business analýza a modelování jejích výstupů jsou klíčovými prvky podporující úspěšný průběh a realizaci IT projektů. Hrají zásadní roli při zjišťování potřeb a požadavků zúčastněných stran, identifikaci procesů, objektů při navrhování architektury technického řešení. V kontextu stále komplexnějších IT projektů jsou modelovací jazyky, jako je UML a ArchiMate, klíčovými nástroji pomáhající analytikům a architektům přesně dokumentovat a komunikovat o různých aspektech. (McConnell, 2006)

Podle významného analytika Steva McConnella (2006) lze prostřednictvím analýzy a přesné specifikaci požadavků predikovat a následně navrhnout efektivní model aplikace, což má za následek signifikantní úspory zdrojů. McConnell (2006) uvádí, že výdaje na opravu chyb objevených během fází implementace nebo údržby technického řešení mohou být až stokrát vyšší ve srovnání s náklady na odhalení těchto problémů v analytické fázi.

Z toho vyplývá, že podceňování nebo dokonce vypuštění analytické fáze ve snaze ušetřit čas a náklady často vede k opačnému výsledku.

### 2.1 Proces analýzy

Proces business analýzy je obvykle strukturován do šesti zásadních kroků viz **obr. 1 Proces business analýzy**. Každý krok má definované úkoly a cíle, které zajišťují prozkoumání a pochopení podnikových potřeb a stanovení cesty k jejich efektivnímu naplnění. (BA Proces, 2020)



**Obrázek 1 Proces business analýzy**

*Zdroj: vlastní zpracování*

Obsah jednotlivých kroků je podle autorů BA Proces (2020) následující:

### 1. Zadání požadavku na business analýzu

Tento krok obsahuje zejména:

- Vytvoření podnětu pro business analýzu
- Identifikaci hlavních osob ze strany zadavatele (stakeholderů), osoby zastávající roli business garanta
- Základní charakteristiku projektu(změny), pro který se analýza bude zpracovávat

### 2. Převzetí požadavku a ověření zadání

Z pohledu analytika zahrnuje zejména:

- Identifikaci prostředí zadavatele požadavků
- Definici problému, který vyvolal potřebu změny
- Určení podnikového kontextu požadavku a motivace pro jeho zadání
- Definice přínosů, jak finančních, tak nefinančních, které má projekt přinést
- Identifikace rizik spojených s neuskutečněním změny
- Odhad nákladů a časové náročnosti provádění analýzy pro jednotlivé fáze projektu

- Přezkoumání zadání s ohledem na dosažitelnost a relevantnost v rámci podnikového kontextu
- Kontrolu, zda je požadavek v souladu s podnikovou strategií a cíli
- Identifikace omezení a překážek, které mohou ovlivnit splnění požadavků

### 3. **Rozhodnutí o zpracování business analýzy**

Tento krok většinou úzce souvisí s rozhodnutím o zahájení projektu a zahrnuje:

- Stanovení důvodů a cílů požadavku na zpracování projektu a jasná definice kritérií pro jeho splnění.
- Potvrzení omezení, která je potřeba zohlednit při zpracování požadavků

### 4. **Analýza prostředí, potřeb a požadavků**

Tento krok je fakticky nejdůležitější fází analýzy, a zahrnuje zejména:

- Definice kritérií pro úspěch a očekávané výsledky analýzy
- Identifikace business potřeb a určení způsobu jejich naplnění
- Analýza současných business procesů a identifikace mezery mezi současným a požadovaným stavem (GAP analýza)
- Identifikace rolí a procesů zúčastněných na změně a popis vztahů mezi business objekty
- Příprava požadavků na funkcionality připravovaného řešení
- Přesnou identifikaci požadavků architektonických, legislativních, případně omezení, které mohou mít vliv na návrh technického řešení
- Příprava případů užití uvažovaného IT řešení
- Způsob ověření naplnění požadavků projektu

### 5. **Výstupy, závěry, doporučení**

Závěrečná fáze zahrnuje doladění výstupů fází předchozích, jejich formální potvrzení a odsouhlasení stakeholdery, a doporučení či zamítnutí přechodu projektu do dalších fází.

## 6. Ukončení práce na business analýze

Tento krok završuje práce na analytické části projektu a jeho hlavních obsahem je:

- Uzavření dokumentace a zajištění, že výsledky analýzy jsou v souladu s podnikovými standardy, výstupy analýzy jsou řádně uloženy, zabezpečeny a předány pro následné kroky a implementaci

Každý z těchto kroků je nedílnou součástí pro zajištění, že business analýza je relevantní, akční a přináší hodnotu pro podnikání. Proces se zaměřuje na přesné pochopení potřeb podniku, komunikaci s klíčovými zainteresovanými stranami a na přípravu podkladů pro informované rozhodování a úspěšnou implementaci změn. (BA Proces, 2020)

### 3 Nástroje analýzy

V rámci business analýzy je nutné pochopit a efektivně využívat různé nástroje a metody. Tyto nástroje a metody nejen pomáhají v identifikaci a řešení problémů, ale také v komunikaci s klíčovými stakeholdery a při shromažďování a konsolidaci požadavků pro budoucí vývoj a inovace. Mezi hlavní nástroje analýzy patří metodiky, příručky, best praxe, apod. Pro zachycení výstupů se používají modelovací software nástroje, textové editory, nástroje pro zachycení průběhu diskusí se zadavateli atd.

BABOK (Business Analysis Body of Knowledge) je uznávaný standard pro business analýzy, který je vyvíjen a udržován mezinárodním institutem business analytiků (IIBA). BABOK popisuje soubor znalostí, dovedností, nástrojů a technik, které jsou nezbytné pro efektivní vykonávání business analýzy. Příručka poskytuje rámec, který pomáhá business analytikům a ostatním profesionálům lépe porozumět své roli, zlepšit své pracovní postupy a zvyšovat hodnotu, kterou přinášejí organizacím.

BABOK definuje šest klíčových oblastí znalostí, které pokrývají celé spektrum kompetencí potřebných pro business analýzu:

- Plánování a monitorování business analýzy: Zabývá se plánováním přístupů k analýze, komunikací se zúčastněnými stranami, určováním požadavků, správou a monitorováním aktivit business analýzy.
- Vyvolání a spolupráce: Zaměřuje se na techniky a metody pro získávání informací od zúčastněných stran a pro práci s nimi k dosažení konsenzu o potřebách a řešeních.
- Řízení životního cyklu požadavků: Popisuje, jak spravovat a udržovat požadavky od jejich zavedení až po jejich dodání a následné vyřazení.
- Analýza strategie: Zahrnuje identifikaci business potřeb, definování řešení a posuzování rizik a příležitostí spojených s těmito řešeními.
- Analýza požadavků a definice návrhu: Zaměřuje se na strukturování a organizaci požadavků, ověřování a ověřování informací a identifikaci potenciálních řešení.
- Hodnocení řešení: Zabývá se hodnocením implementovaných řešení pro zajištění, že splňují business potřeby a že jsou efektivně využívána. (OpenAI, 2024)

BABOK je považován za průmyslový standard pro business analýzu. OpenAI (2024) uvádí význam v následujících aspektech:

- Univerzálnost: BABOK poskytuje srozumitelný a komplexní rámec, který je aplikovatelný v různých průmyslových odvětvích a organizačních strukturách.
- Profesionalita: BABOK definuje standardy a očekávání pro business analytiku, což pomáhá zvyšovat profesionalitu a uznání oboru business analýzy.
- Zlepšení praxe: BABOK nabízí návod, jak zlepšit procesy business analýzy, zvyšovat efektivitu a dodávat větší hodnotu organizacím.
- Certifikace: BABOK slouží jako základ pro certifikační programy, jako je CBAP (Certified Business Analysis Professional), což umožňuje analytikům demonstrovat své znalosti a zkušenosti v oblasti business analýzy.

### **Porozumění business modelům**

Prvním krokem v business analýze je pochopení stávajících business modelů organizace. Podnikové procesy zachycené nástroji, jako je například ARIS, BPMN, apod., poskytují analytikům vizualizovanou představu o fungování a architektuře podniku. Tyto nástroje poskytují pohled na business modely, včetně klíčových procesů, zdrojů, zákazníků a dalších prvků, které ovlivňují fungování firmy.

### **Průzkum řídicích dokumentů a příruček**

Průzkum řídicích dokumentů, jako jsou interní předpisy, směrnice a příručky pro výkonné pracovníky je nezbytný pro porozumění stávajícím postupům a pravidlům.

### **Spolupráce se stakeholdery**

Efektivní business analýza vyžaduje pravidelnou interakci s klíčovými stakeholdery, včetně výkonných pracovníků, zákazníků, dodavatelů a jiných zainteresovaných stran. Workshopy, konzultace a meetingy jsou nástroje, které umožňují analytikům získat hluboký vhled do potřeb a očekávání stakeholderů na navrhované změny a inovace.

## **Modelovací nástroje a techniky**

Pro modelování a vizualizaci procesů se často využívají nástroje, jako je Enterprise Architect, které podporují různé modelovací notace. Nástroje tohoto typu nejen usnadňují porozumění a komunikaci procesů a analýzu scénářů pro optimalizaci procesů.

## **Nástroje pro sběr a konsolidaci požadavků**

Sběr a konsolidace požadavků od stakeholderů je zásadní pro úspěšnou realizaci jakýchkoliv změn v organizaci. Nástroje pro správu požadavků, jako jsou JIRA, Trello, nebo speciální software pro správu požadavků, pomáhají v této fázi tím, že poskytují strukturované a sledovatelné prostředí pro zaznamenávání, prioritizaci a sledování stavu požadavků. (OpenAI, 2024)

## 4 Význam modelování v procesu analýzy IT projektu

Modelování je zásadní technika v rámci business analýzy a systémového inženýrství, která přináší hlubší porozumění a přesnější reprezentaci zkoumaných jevů, systémů, nebo procesů.

Přístup k popisu a pochopení komplexních systémů nebo situací může být realizován na třech základních úrovních: prostřednictvím psaného textu, vizuálního obrázku, nebo strukturovaného modelu. Každá z těchto úrovní poskytuje odlišný stupeň detailu a porozumění. (Wiegers, 2008)

### **Zachycení výstupů analýzy pouze psaným textem**

Psané slovo je základním nástrojem pro komunikaci a popis myšlenek. Přestože může být extrémně podrobné, často se ztrácí v abstraktnosti a může být pro čtenáře obtížně představitelné, obzvláště v případech složitých technických nebo procesních detailů. Zjednodušeně pro představu můžeme charakterizovat takový výstup jako 1D nebo jednodimenzionální.

### **Zachycení výstupů analýzy psaným textem a doprovodnými obrázky**

Vizuální reprezentace, jako jsou kresby, poskytují rychlejší a intuitivnější porozumění a jsou výborné pro ilustraci myšlenek a konceptů. Obrázky však často postrádají hloubku a nezahrnují vazby a parametry, které jsou klíčové pro plné pochopení systému. Proto je potřeba je doplnit doprovodným textem v duchu předchozího bodu. Zjednodušeně pro představu můžeme pak charakterizovat takový výstup analýzy jako 2D nebo dvoudimenzionální, druhou dimenzi přidává grafická reprezentace výstupů.

### **Zachycení výstupů analýzy pomocí modelu**

Model představuje nejvyšší úroveň popisu, protože integruje vizuální prvek s analytickým. Modely jsou schopny zachytit složité systémy s vysokým stupněm přesnosti, včetně vztahů mezi jednotlivými prvky a jejich parametry. Tímto způsobem poskytují komplexní rámec pro analýzu, simulaci a předpověď chování systému. Když je model



správně konstruován, jsme ho schopni přeměnit na obrázky, diagramy doplněné o popisný text, nebo text samotný. Modelování je proto preferováno v situacích, kde je požadována vysoká úroveň pochopení a přesnosti, a kde je cílem nejenom popsat, ale i analyzovat a předpovídat. Klíčovým rozdílem mezi modelem a prostým grafickým zachycením výstupů je v tom, že v modelovacím nástroji má element modelu sadu atributů, které umožňují k němu navázat informace využitelné pro generování textové dokumentace modelu i grafické reprezentace. Podle jednoho nebo více atributů elementů můžeme v rozsáhlých modelech vyhledávat potřebné elementy, jejich vzájemné vazby, používat je na více diagramech apod. Analogicky s předchozím srovnáním můžeme charakterizovat model jako multi-D nebo vícedimenzionální.

Modelování přináší větší hodnotu při řešení složitých problémů, protože umožňuje analytikům a ostatním zúčastněným stranám lépe pochopit a pracovat s komplexními systémy. Umožňuje simulace a predikce, které jsou zásadní pro strategické plánování a rozhodování.

Modelování také umožňuje detailně zaznamenat rozdíl mezi současným "AS-IS" stavem a budoucím "TO-BE" stavem, což poskytuje jasný přehled o tom, co se má změnit. Tato schopnost detailně zachytit a vizualizovat změny je klíčová pro strategické plánování, řízení změn a zajištění, že všechny zainteresované strany mají jasnou a srozumitelnou představu o předpokládané podobě a možném vývoji systému.

## 5 Modelovací jazyky

Modelovací jazyky jako jsou UML a ArchiMate umožňují týmům překlenout komunikační bariéry mezi IT a business stranou. To vede k efektivnějšímu řešení problémů a vývoji systémů.

### 5.1 ArchiMate

ArchiMate je standardizovaný modelovací jazyk, který se využívá pro dokumentaci, analýzu a vizualizaci podnikové architektury.

Základní principy ArchiMate mají kořeny v jazyce Unified Modeling Language, ale ArchiMate je speciálně upraven a rozšířen tak, aby vyhověl specifickým potřebám modelování architektur, případně podnikových strategií nebo dalších oblastí, kde nasazení jazyka UML není vhodné. Jazyk kombinuje prvky z různých oblastí podnikové architektury včetně procesů, aplikací a technologické infrastruktury, čímž umožňuje integrovaný pohled na podnik.

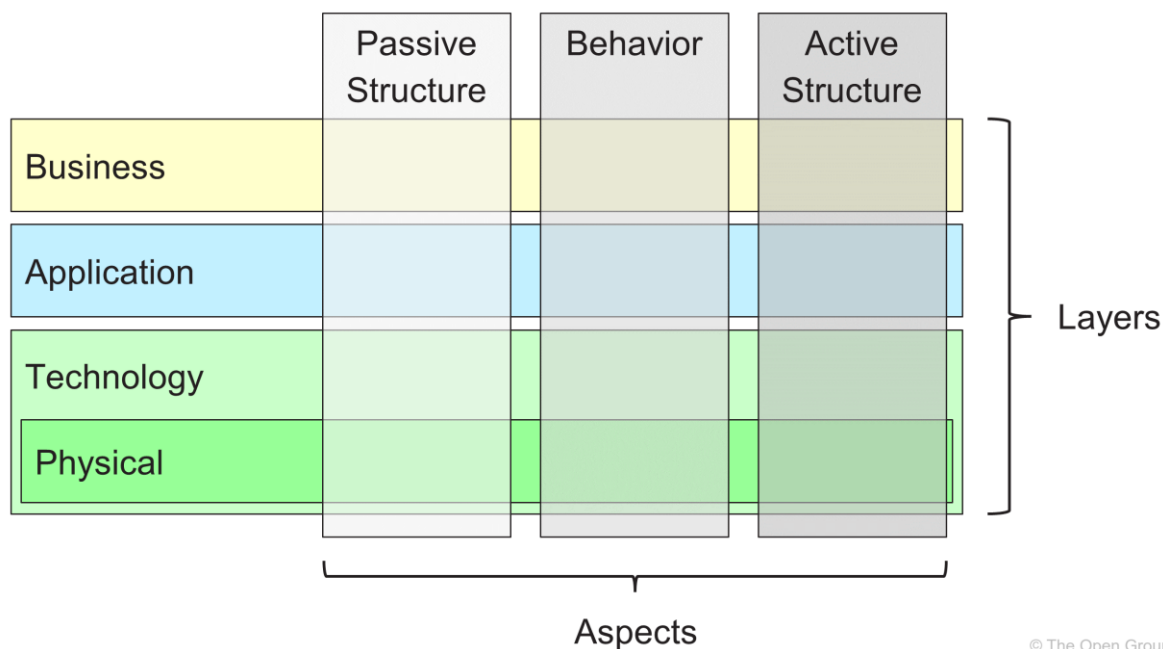
Terminologie ArchiMate je úzce spjata s frameworkem TOGAF, který poskytuje metodologii pro plánování, implementaci a správu podnikové architektury. ArchiMate a TOGAF se vzájemně doplňují, kde ArchiMate poskytuje jazyk pro vizualizaci a analýzu architektonických návrhů a TOGAF nabízí metodologii a procesní rámec pro architektonické plánování a implementaci.

Vztah mezi ArchiMate a TOGAF lze charakterizovat tak, že ArchiMate poskytuje mechanismy a nástroje pro detailní popis a vizualizaci architektury (tedy „jak“), zatímco TOGAF definuje strukturu a požadavky pro architektonický proces a jeho výstupy (odpovídá na otázku „co“). (ArchiMate, 2024)

#### 5.1.1 Struktura jazyka

Jak popisuje Hosiaisuoma (2019) pro představu o základní struktuře jazyka ArchiMate slouží tzv. ArchiMate Core Framework. Ten představuje systém tvořený devíti buňkami, které rozdělují elementy jazyka do kategorií podle jejich funkcí v modelu (pasivní, aktivní, elementy popisující chování) a do vrstev modelu, které slouží

k rozlišení úrovní modelu. Framework naznačuje možnosti modelování podnikové architektury z různých úhlů pohledu viz **Obrázek 2 ArchiMate – Core Framework**.



**Obrázek 2 ArchiMate – Core Framework**

*Zdroj: (The Open Group, 2023)*

Klasifikace prvků je založena na aspektech a vrstvách pouze globálně. Prvky reálné architektury nemusí být přísně omezeny na jeden aspekt, nebo vrstvu. Například, business role fungují jako prostředník mezi prvky, které jsou čistě behaviorální a prvky, které jsou čistě strukturální, a v závislosti na kontextu může být určitý software považován za součást aplikační vrstvy, nebo technologické vrstvy. Struktura tohoto rámce umožňuje modelování podniku z různých pohledů. (Hosiaislouma, 2019)

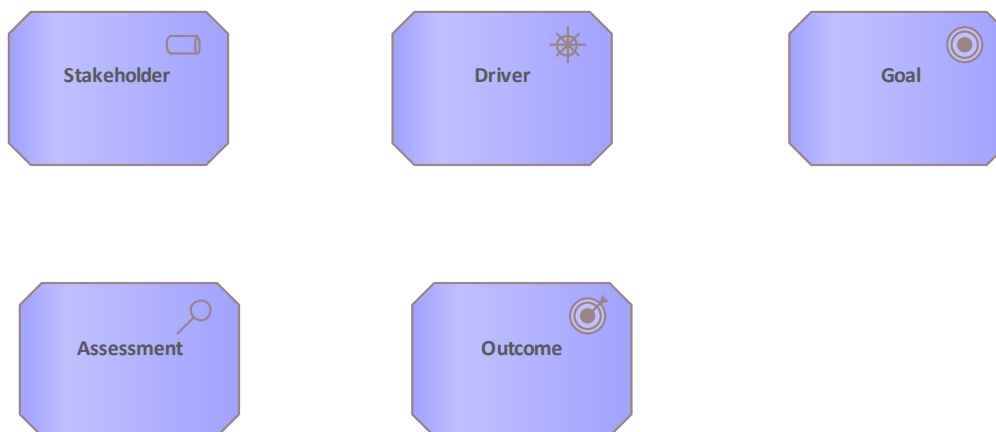
### 5.1.2 Vrstvy jazyka

Základní vrstvy dle The Open Group (2023) jsou realizovány třemi úrovněmi, na kterých lze modelovat podnik v ArchiMate – business, aplikační a technologická. Motivační vrstva byla doplněn v novějších verzích jazyka je používána jako doplňující vrstva.

## Motivační vrstva

Má za úkol zachytit záměry, motivace či hlavní strategické důvody vedoucí k zadání projektu či IT změny.

Hlavní elementy jsou zobrazeny níže na *obr. 3 ArchiMate – Hlavní elementy motivační vrstvy*.



**Obrázek 3 ArchiMate – Hlavní elementy motivační vrstvy**

*Zdroj: vlastní zpracování*

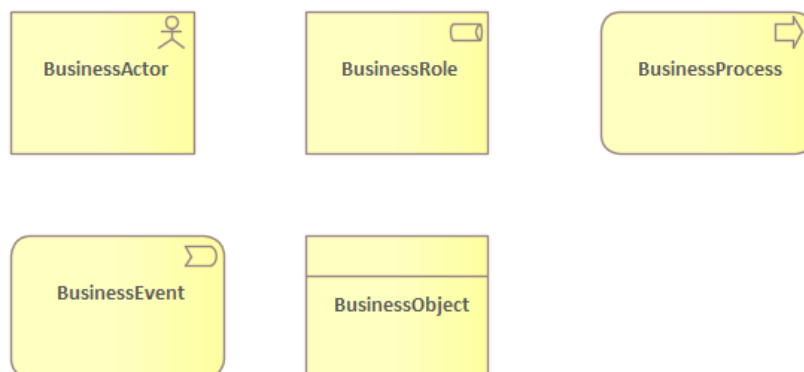
Stakeholder představuje vrcholné zástupce zadávající (business) strany, driver označuje důvod pro spuštění projektu z pohledu vedení společnosti či firmy. Assessment může představovat nějaký dokument, zjištění, auditní výstup, doporučení vedoucí ke snaze zlepšit fungování procesů společnosti pomocí IT projektu. Goal je obecné vyjádření cíle/cílů projektu, outcome potom jeho přesnější, cílový stav.

## Business vrstva

Zaměřuje se na modelování business procesů a funkcionality organizace. V této vrstvě jsou reprezentovány business role, procesy a služby. Dohromady jsou základní strukturou podnikového modelu. *obr. 4 ArchiMate – Hlavní elementy business vrstvy*, zobrazuje hlavní elementy pro business vrstvu.

Business actor představuje jednotku nebo entitu, která provádí business aktivity. Může se jednat o fyzickou osobu, organizaci, nebo její část. Business role definuje sadu odpovědností, chování a oprávnění přidělených podnikovému aktérovi. Role může být přiřazena jednomu nebo více aktérům a odráží jejich funkci v rámci podnikových

procesů. Business event je situace, která vyvolává nebo ukončuje business procesy. Ty reprezentují sled činností organizovány za účelem dosažení konkrétního cíle. Business objekt je objektem pro zaznamenání informací v jakékoliv formě.



**Obrázek 4 ArchiMate – Hlavní elementy business vrstvy**

*Zdroj: vlastní zpracování*

### **Aplikační vrstva**

Soustředí se na aplikační hardware a softwarové komponenty, které podporují business procesy organizace. Aplikační vrstva poskytuje přehled o aplikacích, jejich komponentách, komponentách hardware a jejich propojení s business vrstvou. Příklad hlavních elementů viz *obr. 5 ArchiMate – Hlavní elementy aplikační vrstvy*.

Aplikační služba definuje, jaké aplikační funkcionality jsou nabízeny, aniž by specifikovala, jaké konkrétní aplikační komponenty, nebo konfigurace jsou pro tyto služby použity. Aplikační komponenta představuje modulární část informačního systému, která poskytuje aplikační funkcionality.



**Obrázek 5 ArchiMate – Hlavní elementy aplikační vrstvy**

*Zdroj: vlastní zpracování*

## Technologická vrstva

Zaměřuje se na hardware a infrastrukturu, které jsou základem pro realizaci aplikační vrstvy. Obsahuje fyzické komponenty, jako jsou servery, databáze, nebo síťové zařízení použité pro provoz softwarových aplikací. Příklad hlavních elementů viz **obr. 6 ArchiMate – Hlavní elementy technologické vrstvy**.



**Obrázek 6 ArchiMate – Hlavní elementy technologické vrstvy**

*Zdroj: vlastní zpracování*

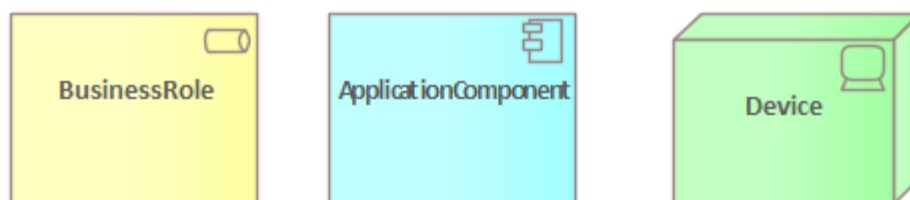
### 5.1.3 Aspekty jazyka

Aspekty dle The Open Group (2023) byly inspirovány jazykem, kde věta má subjekt (aktivní struktura), sloveso (behaviorální aspekt) a objekt (pasivní struktura). Použitím stejných konstrukcí, na které jsou lidé ve svých vlastních jazycích zvyklí, je jazyk ArchiMate snadnější na učení a čtení.

Vzhledem k tomu, že notace ArchiMate je grafický jazyk, kde jsou prvky organizovány prostorově, pořadí v modelování není důležité.

### Aktivní strukturální aspekt

Reprezentuje strukturální prvky, které jsou schopny vykazovat chování, jako jsou například business role, aplikační komponenty a fyzická zařízení (viz. příklad prvků níže na **Obr. 7 ArchiMate – Příklad prvků aktivního strukturálního aspektu**). Těmto prvkům lze přiřadit konkrétní úkoly a odpovědnosti v rámci business, aplikačních či technologických procesů.



**Obrázek 7 ArchiMate – Příklad prvků aktivního strukturálního aspektu**

*Zdroj: vlastní zpracování*

### Behaviorální aspekt

Zaměřuje se na modelování chování organizace prostřednictvím procesů, funkcí, událostí a služeb (viz. příklad prvků níže na *obr. 8 ArchiMate – Příklad prvků behaviorálního aspektu*). Prvky slouží pro pochopení operací a služeb poskytovaných organizací. Behaviorální aspekt zahrnuje vazby mezi aktivními strukturálními prvky a definuje, jakými způsoby jsou tyto prvky využívány či interagují.

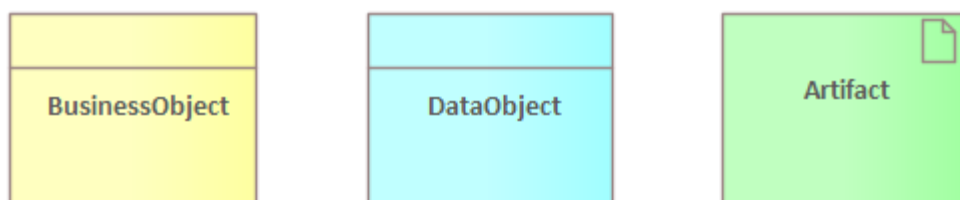


**Obrázek 8 ArchiMate – Příklad prvků behaviorálního aspektu**

*Zdroj: vlastní zpracování*

### Pasivní strukturální aspekt

Představuje objekty, na kterých je chování vykonáváno, například data, nebo informace (viz. příklad prvků níže na *obr. 10 ArchiMate – Příklad prvků pasivního strukturálního aspektu*). Pasivní strukturální prvky jsou manipulovány, nebo změněny prostřednictvím behaviorálních procesů.



**Obrázek 9 ArchiMate – Příklad prvků pasivního strukturálního aspektu**

*Zdroj: vlastní zpracování*

#### 5.1.4 ArchiMate metamodel

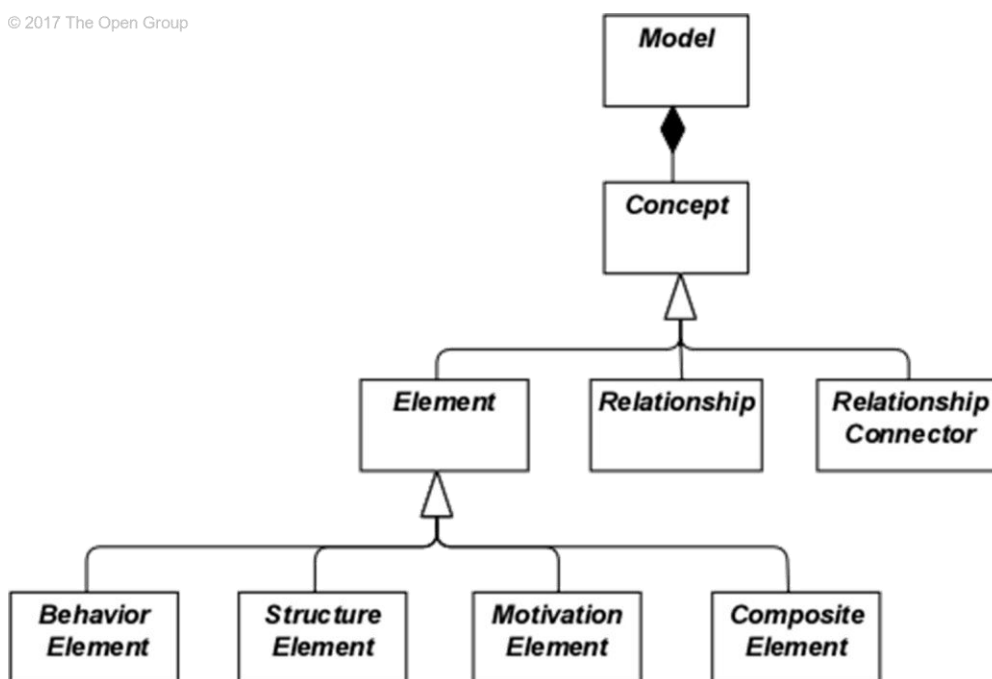
Metamodel modelovacího jazyka ArchiMate představuje strukturovaný způsob, jakým jsou definovány a vzájemně propojeny elementy a koncepty používané v ArchiMate. Poskytuje konzistentní rámec, který umožňuje analytikům a architektům podniku efektivně a srozumitelně modelovat, analyzovat a vizualizovat složité struktury a vztahy pomocí tohoto modelovacího jazyka v rámci organizace. (Hosiaislouma, 2019)

#### **Metamodel struktury jazyka**

U jazyka ArchiMate se setkáváme s řadou metamodelů, které poskytují strukturovaný přehled o tom, jak je jazyk postaven a jak se používá. Jedním z klíčových konceptů je metamodel top-level struktury jazyka, viz *obr. 11 ArchiMate – Top level metamodel*. Tento metamodel je základní průvodce, který pomáhá pochopit základy jazyka ArchiMate a jeho strukturu. Vysvětluje, že každý model v ArchiMate je sestaven z konceptů, přičemž každý koncept může být buď prvkem, nebo vztahem (resp. konektorem vztahu).

Prvky i vztahy mohou být dále kategorizovány do specifických tříd nebo typů. Kategorizace pomáhá přiřazovat prvky a vztahy do logických skupin a poskytuje kontext pro jejich pochopení a správné využití v rámci modelu. (The Open Group, 2023)





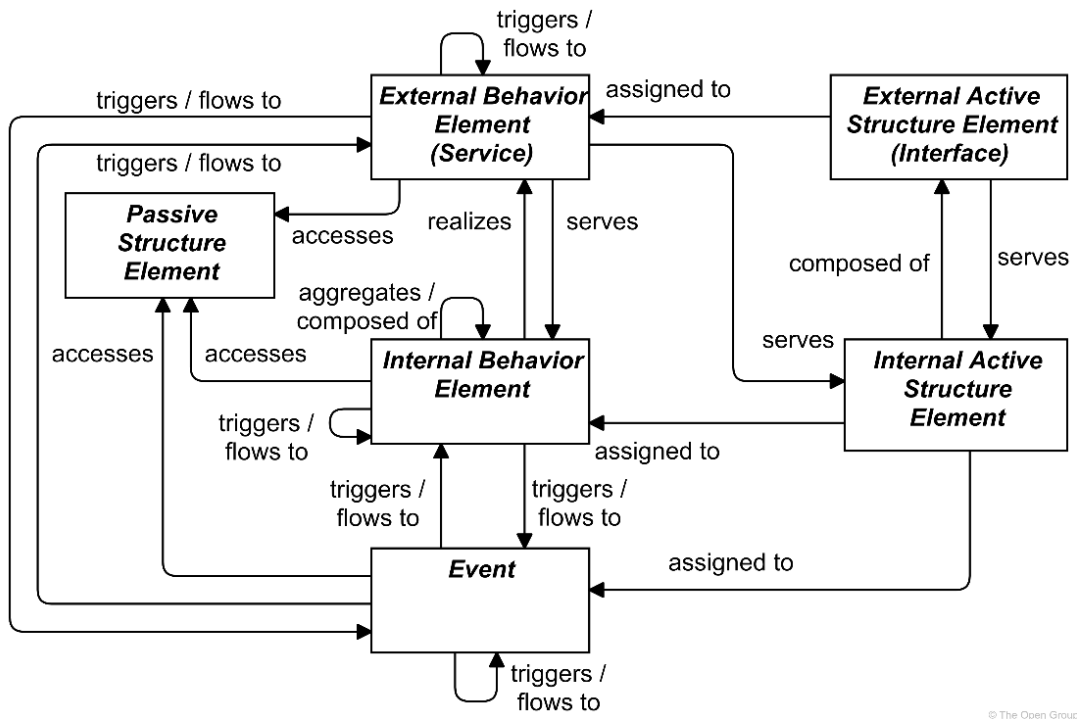
**Obrázek 10 ArchiMate – Top level metamodel**

Zdroj: (The Open Group, 2023)

### Generický metamodel

Generický metamodel, zobrazený na *obr. 12 ArchiMate – Generický metamodel*, je považován za základní metamodel v modelovacím jazyce ArchiMate. Znázorňuje, jak jsou strukturální a behaviorální prvky propojeny, a jaká jsou základní pravidla pro jejich vzájemné působení. Nabízí ucelený pohled, jak je možné různé aspekty podnikové architektury reprezentovat a analyzovat. Pravidla definovaná v generickém metamodelu jsou aplikována a rozšířena do všech vrstev, čímž se zajišťuje konzistence a srozumitelnost modelů napříč celou architekturou. (The Open Group, 2023)

Kompletní vizualizace metamodelu modelovacího jazyka ArchiMate, která je konstruována s použitím modelovacích elementů, je představena v příloze viz *Příloha 1 Kompletní metamodel*.



**Obrázek 11 ArchiMate – Generický metamodel**

Zdroj: (The Open Group, 2023)

## 5.2 Unified Modeling Language

Jednotný modelovací jazyk (Unified Modeling Language – UML) je v oblasti softwarového inženýrství grafický jazyk, který se využívá pro vizualizaci struktur a procesů při vývoji softwaru. Jazyk poskytuje univerzální formát pro popis architektury systému i jeho specifických komponent.

Jednou z hlavních výhod UML je jeho schopnost prezentovat složité informace o systému dekomponované do přehledných a standardizovaných diagramů. Zobrazují dynamiku a hierarchii komponent systému i abstraktnější elementy, jako jsou procesy a funkce, které v systému interagují.

Flexibilita jazyka umožňuje uživatelům přizpůsobit si ho pro různé druhy aplikací a neomezuje se pouze na jednu programovací platformu nebo databázové schéma. Univerzální povaha UML jej činí nepostradatelným nástrojem pro vývoj softwarových komponent, které lze efektivně znovu využívat.

UML samo o sobě nepředepisuje konkrétní metodiku pro analýzu, nebo návrh systémů. Nabízí sadu nástrojů, které mohou být použity v rámci různých metodik softwarového inženýrství.

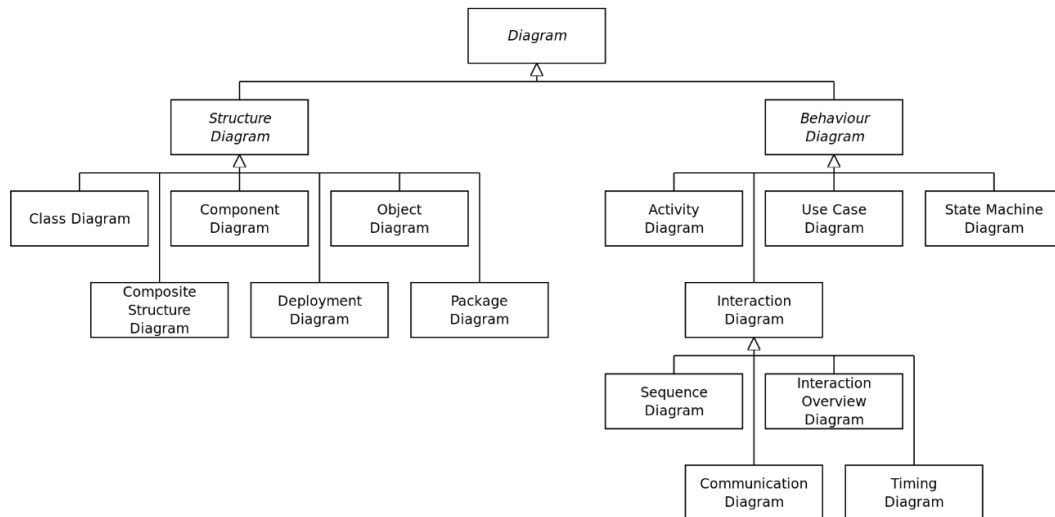
Za standardizaci UML zodpovídá skupina Object Management Group, která se věnuje vytváření a podpoře standardů v oblasti softwarového inženýrství. Skupina má snahu zajistit, aby UML zůstal aktuální a relevantní ve světě rychle se vyvíjejících technologií. (Unified Modeling Language, 2024)

### 5.2.1 Struktura jazyka

Na rozdíl od Archimate, kdy elementy nemají striktně určeno, v jakých typech diagramů se mají vyskytovat, lze elementy UML klasifikovat právě z pohledu typů diagramů. Unified Modeling Language (2024) rozděluje diagramy do dvou základních skupin. Jedná se o strukturální a chovací diagramy. Diagramy interakce se obvykle kategorizují do podskupiny diagramů chování. Struktura UML je zobrazena na *obr. 13 UML – Typy UML diagramů*.

**Strukturální diagramy** poskytují statický pohled na systém. Prezентují architekturu softwaru a jsou zaměřeny na elementy, které jsou často považovány za „kosterní systém“. Tyto diagramy detailně popisují, jaké datové struktury jsou používány, jak jsou tyto struktury uloženy a jakými vztahy jsou propojeny.

**Diagramy chování** se věnují způsobu, jak instance struktur spolu navzájem komunikují. Ilustrují, jak jednotlivé části systému interagují a reagují na různé události v čase. (Rydval, 2019)



**Obrázek 12 UML – Typy UML diagramů**

*Zdroj: (Unified Modeling Language, 2024)*

## 5.2.2 Diagramy struktury

Podle Rydvala (2019) rozlišujeme následující diagramy struktury.

### Diagram tříd

Diagram tříd popisuje třídy a vztahy mezi nimi. Je základem pro každého analytika, který potřebuje zachytit pojmy specifikované zadavatelem, doménové modely, analytické modely, nebo databázová schémata. Tento diagram je významný pro objektivě orientované návrhy a analýzy.

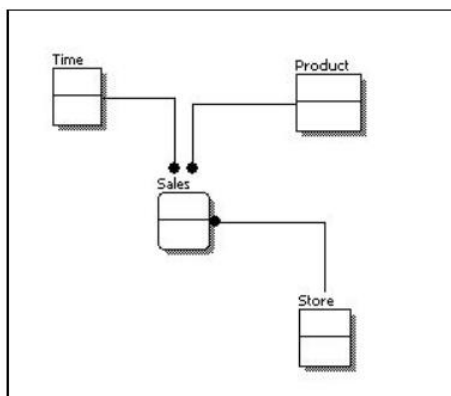
V rámci projektů je diagram tříd využíván na různých úrovních a v různých fázích, aby bylo možné efektivně a srozumitelně předávat informace různým zainteresovaným stranám.

### Koncepční Úroveň

Na této úrovni se diagram tříd soustředí na základní pojmy a vztahy v doméně viz **obr. 14 UML – Konceptuální model**.

Cílem je zachytit a zobrazit hlavní entity a jejich vztahy, jak jsou chápány zadavatelem a uživateli, bez přílišného zaměření na technické detaily.

Koncepční model pomáhá všem zúčastněným stranám (včetně ne-technických) pochopit strukturu a dynamiku domény.



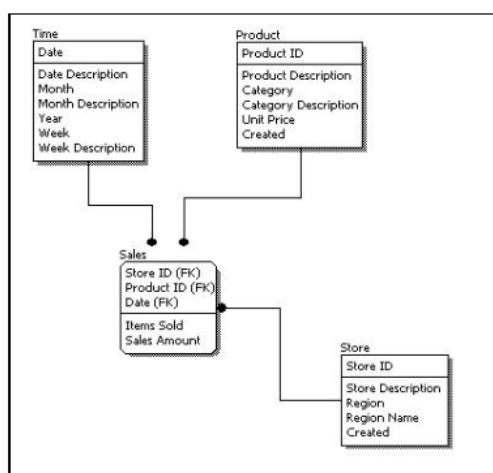
**Obrázek 13 UML – Koncepční model**

*Zdroj: (BA Proces, 2020)*

### Logická Úroveň

Logický model přidává více detailů, včetně specifikace atributů tříd, metod a detailnějších vztahů mezi třídami viz *obr. 15 UML – Logický model*.

V této fázi se model začíná více orientovat na implementaci, ale stále zůstává nezávislý na konkrétních technologiích a platformách.



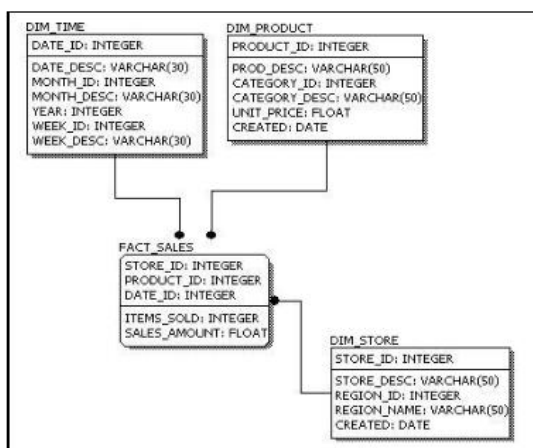
**Obrázek 14 UML – Logický model**

*Zdroj: (BA Proces, 2020)*

## Fyzická Úroveň

Fyzický model zobrazený na *obr. 16 UML – Fyzický model* se zaměřuje na implementační detaily, včetně konkrétních technologií, frameworků a databázových schémat.

Na této úrovni se diagram tříd stává konkrétním plánem pro vývojáře, jak struktury a vztahy implementovat v kódu.



*Obrázek 15 UML – Fyzický model*

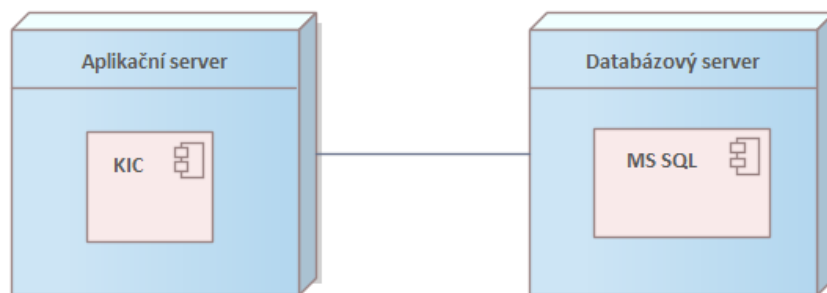
*Zdroj: (BA Proces, 2020)*

## Objektový diagram

Objektové diagramy ukazují, jak mohou vznikat instance tříd, tedy konkrétní objekty. Často se nejprve zachycuje objekt a poté jsou z nich tvořeny třídy. Tyto diagramy jsou užitečné pro konkrétní ilustraci a pochopení vztahů a interakcí mezi objekty v systému.

## Diagram nasazení

Tento typ diagramu ukazuje, jak je systém umístěn na softwarové a hardwarové infrastruktuře. Zobrazuje fyzické rozložení systémových komponent a jejich vzájemné propojení. Příklad diagramu viz *obr. 17 Diagram nasazení*



**Obrázek 16 UML – Diagram nasazení**

*Zdroj: vlastní zpracování*

### **Diagram balíků**

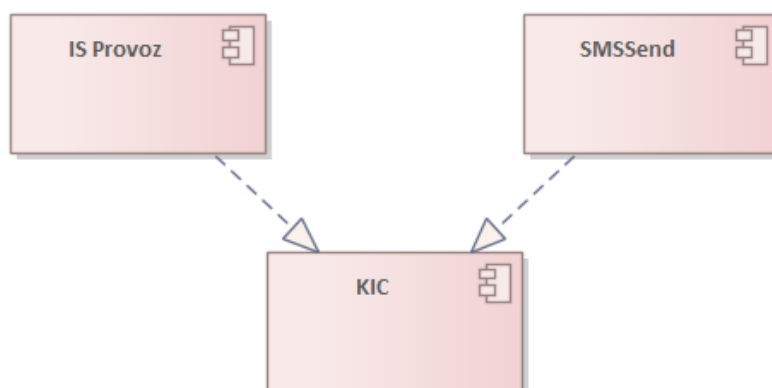
Diagram balíků zobrazuje organizaci a závislosti mezi balíky (packages) v modelu. Balíky skupují třídy a další typy elementů do vyšších logických celků.

### **Diagram vnitřních struktur**

Tento diagram je pokročilým nástrojem pro popis vnitřní struktury tříd nebo komponent. Zobrazuje interní součásti a jejich vzájemné vztahy.

### **Komponentový diagram**

Popisuje softwarové části řešení a jejich vzájemné vztahy. Umožňuje analytikům pochopit, jak jsou systémové komponenty organizovány a jak spolu komunikují. Příklad diagramu viz *Obr. 18 UML – Komponentní diagram*.



**Obrázek 17 UML – Komponentní diagram**

*Zdroj: vlastní zpracování*

## Profily

Umožňují rozšířit standardní UML o další prvky specifické pro konkrétní doménu nebo metodologii. Profily umožňují definovat nové stereotypy, omezení a další prvky.

### 5.2.3 Diagramy chování

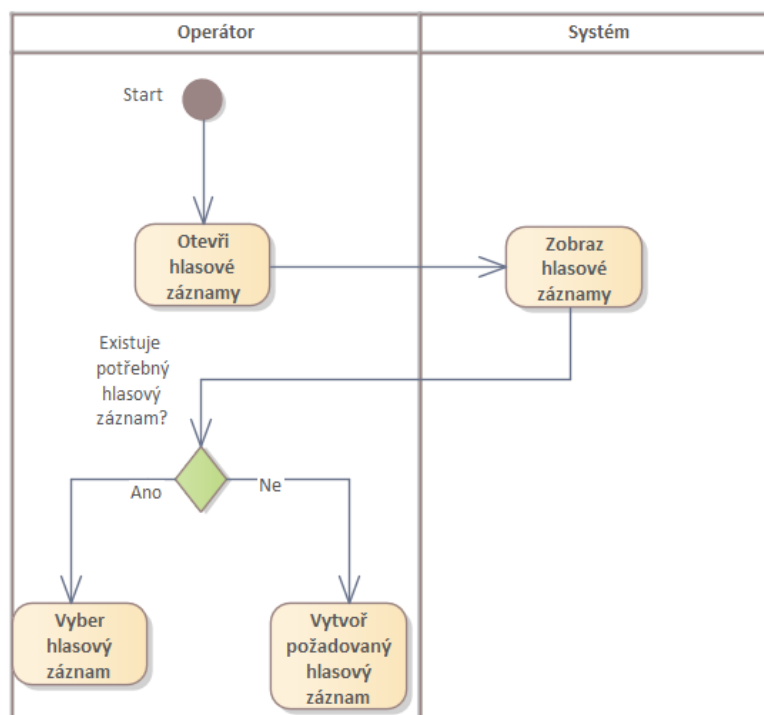
Rydval (2019) popisuje následující diagramy chování.

#### Stavový diagram

Každý objekt může procházet různými stavy v rámci svého životního cyklu. Stavový automat zaznamenává nejen množinu stavů, ale i podmínky pro přechody mezi stavy.

#### Diagram aktivit

Zachycuje průběh aktivit pomocí hran a uzlů, které můžeme chápat jako kroky v dané aktivitě. Je užitečný pro modelování pracovních postupů a algoritmů. Příklad diagramu viz *obr. 19 UML – Diagram aktivit*



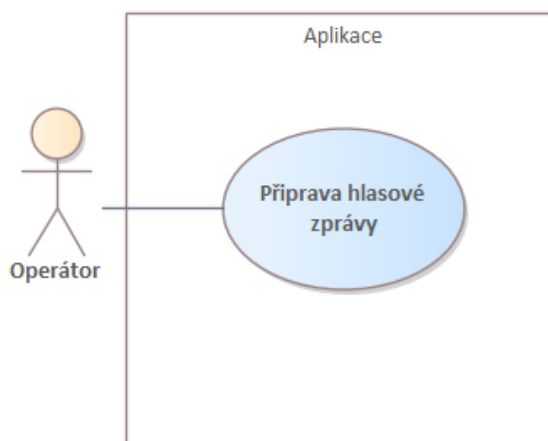
**Obrázek 18 UML – Diagram aktivit**

*Zdroj: vlastní zpracování*



## Diagram případů užití

Zobrazuje případy užití daných systémů, jejich aktéry a vzájemné vztahy. Tento diagram je klíčový pro zachycení funkčních požadavků a interakcí mezi systémem a jeho uživateli nebo dalšími systémy. Příklad diagramu viz *obr. 20 UML – Diagram případů užití*



*Obrázek 19 UML – Diagram případů užití*

*Zdroj: vlastní zpracování*

## Sekvenční diagram

Sekvenční diagram ilustruje interakce mezi objekty v určitém pořadí, které odpovídá průběhu času. Zobrazuje objekty, jejichž životní cyklus a zprávy, které si vyměňují.

## Diagram časování

Tento diagram je specifický tím, že ukazuje změny stavů objektů v čase. Bohužel pro běžného uživatele není dostatečně intuitivní a tím se stává nečitelný.

## Komunikační diagram

Komunikační diagram prezentuje interakce mezi objekty podobně jako sekvenční diagram, ale klade větší důraz na strukturu vztahů mezi objekty než na časovou posloupnost.

## Diagram přehledu interakcí

Kombinuje prvky z diagramu aktivit a sekvenčního diagramu, poskytuje vysokou úroveň abstrakce interakcí v systému. Zobrazuje tok interakcí a kontrolní struktury, jako jsou rozhodovací uzly nebo smyčky. Je vhodný pro zobrazení přehledu interakcí v komplexních systémech, kde je potřeba porozumět celkové struktuře a toku zpráv nebo událostí.

Závěrem této části lze říci, že na většině projektů se jen zřídka využívají všechny typy diagramů UML. Záleží vždy na konkrétních potřebách.

### 5.2.4 Způsob používání UML

UML lze podle Martina Fowlera (2009) rozdělit do třech skupin používání:

- UML pro náčrt (UML as a sketch): Tento přístup je neformální a často se používá během schůzek. Záznamy se obvykle dělají na tabuli nebo na papír. Při této metodě není klíčové striktně dodržovat pravidla UML – důležitější je zachytit a sdílet myšlenku nebo koncept.
- UML pro podrobný návrh (UML as a blueprint): Používá se pro detailní návrh systému nebo jeho části. Zde se již pravidla UML dodržují přísněji, ale ne vždy se využívají všechny prvky UML. Cílem je navrhnout rozhraní systému, zatímco jeho vnitřní strukturu může každý řešit individuálně, opět třeba s využitím UML.
- UML jako programovací jazyk: V tomto případě se UML překládá do spustitelného tvaru nebo je celý model interpretován. Toto se realizuje v rámci přístupů jako je Model Driven Architecture (MDA) nebo Model Driven Development (MDD).

Fowler (2009) také zdůrazňuje jedno klíčové doporučení: „Buďte připraveni porušit pravidla UML, kdykoliv vám to pomůže zlepšit komunikaci. Není cenu vytvářet diagramy, kterým nikdo nerozumí.“ Toto doporučení připomíná, že hlavním cílem UML je usnadnit komunikaci a porozumění systému a někdy může být pro dosažení tohoto cíle vhodné odchýlit se od standardních pravidel.

### 5.3 Možnosti propojení jazyka UML a ArchiMate

ArchiMate poskytuje komplexní pohled na organizaci zahrnující business procesy, informační systémy a technologickou infrastrukturu. Jeho silná stránka je schopnost zobrazit a analyzovat vzájemné vztahy a závislosti mezi různými prvky na vysoké úrovni.

UML je zase ideální pro detailnější znázornění, specifikaci a design softwarových systémů.

Propojení UML a ArchiMate umožňuje organizacím zachytit a spravovat strategický přehled celé enterprise architektury prostřednictvím ArchiMate, tak i detailní technické aspekty jednotlivých systémových komponent pomocí UML.

Názorný příklad propojení těchto dvou jazyků je znázorněn na **obr. 36 IT Projekt – Příklad znázornění detailu komponenty**.

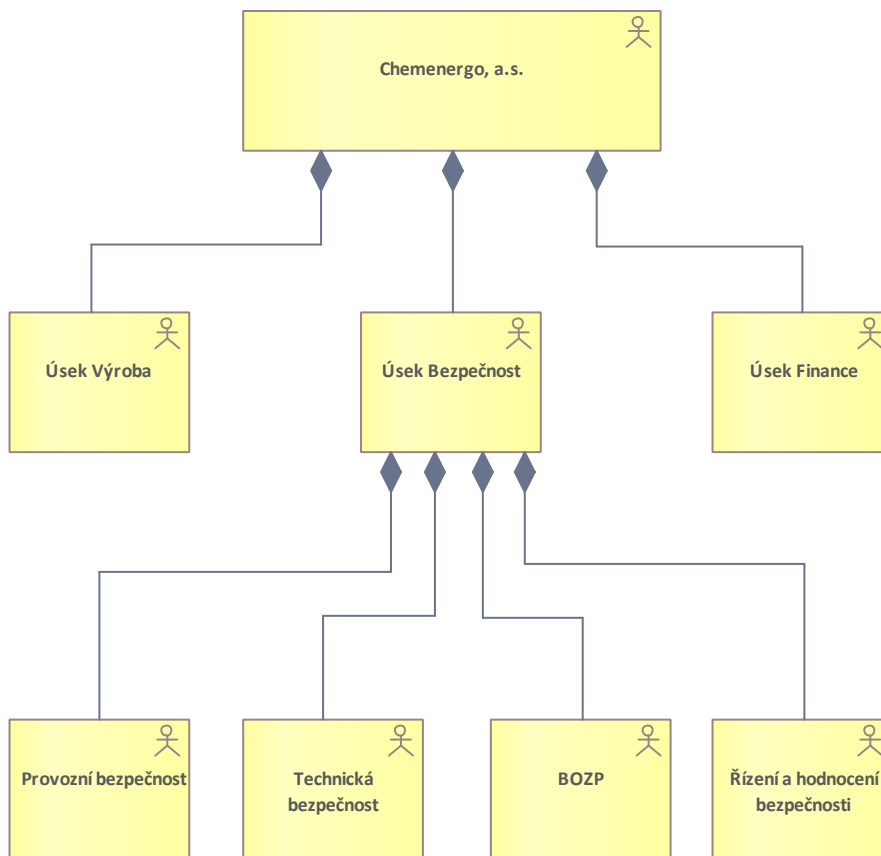
Přestože spojení obou jazyků při zachycení výstupů analýzy a návrhu výsledného technického řešení poskytuje zřetelné výhody, v praxi se nevyskytuje příliš často. Jedním z důvodů subjektivních může být skutečnost, že není mnoho analytiků, kteří by byli ochotni zvládnout oba jazyky. Většinou každý preferuje a zná dostatečně jen jeden z nich. Dalším důvodem, objektivním, je fakt, že neexistuje přesná a jednotná metodika a postup, jak propojení obou jazyků realizovat. O jeden z možných přístupů se pokouší i tato práce.

## 6 Provedení analýzy v konkrétním IT projektu

V této kapitole je představen reálný IT projekt, který byl pro účely práce anonymizován. Některé informace a výstupy projektu jsou fiktivní a byly vytvořeny pouze pro demonstrační účely. Základní struktura a přístupy použité v analýze odpovídají standardním postupům v oblasti IT analýzy.

### 6.1 Základní popis prostředí

Jedná se o velký palivoenergetický komplex (organizační struktura viz *obr. 21 IT Projekt – Základní organizační schéma podniku*). Komplex produkuje chemické látky, paliva a jako vedlejší produkt také vyrábí elektrickou energii pro vlastní potřebu a částečně jí zásobuje okolní obce. Vzhledem k povaze látek a procesům, které je zpracovávají, je nezbytné zajištění bezpečnosti provozu komplexu a minimalizace škod při případných haváriích. Za tuto oblast je odpovědný ředitel úseku Bezpečnost. Ten má připraveny havarijní plány pro zvládnutí nebezpečných situací a pravidelně ověřuje jejich funkčnost při nácviku havarijní připravenosti, a také při haváriích simulovaných. Tým, který zabezpečuje bezpečnost a organizuje nácviky zvládnutí nebezpečných situací, má označení Organizace Havarijní Odezvy (OHO). Jeho členové pracují v důležitých funkcích napříč úseky, disponují velkými zkušenostmi a mají i organizační schopnosti. Pro podporu činnosti tohoto týmu je určený další tým Podpora Organizace Havarijní Odezvy (POHO), který má za úkol zajistit potřebné technické prostředky pro podporu zajištění havarijních situací, zejména výpočetní a komunikační techniku. Podpůrný tým je svoláván při skutečných haváriích a také při nácviku haváriích simulovaných, které mají za úkol zjistit stav připravenosti a odhalit případné nedostatky. Vedení úseku Bezpečnost dostalo za úkol připravit IT Systém, který bude sloužit při svolání týmu, analýze výsledků svolání, bude zachycovat a prokazatelně uchovávat reakce a dobu odezvy jeho jednotlivých členů, poskytne souhrnné výsledky úspěšnosti svolání a podklady pro nápravu nedostatků.



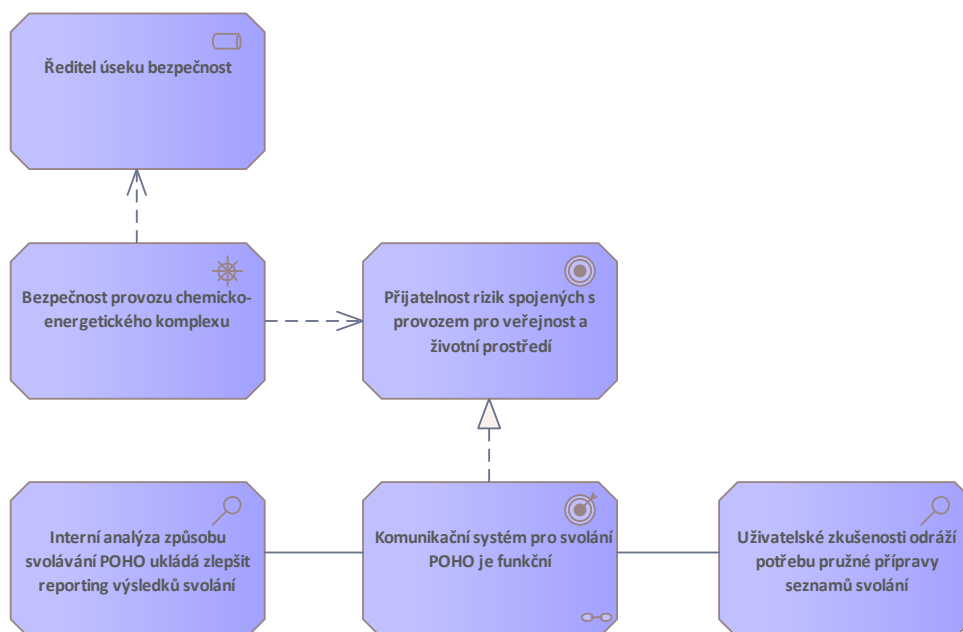
**Obrázek 20 IT Projekt – Základní organizační schéma podniku**

*Zdroj: vlastní zpracování*

## 6.2 Motivační vrstva

IT projekt se zaměřuje na podporu při zajištění připravenosti a efektivní reakce personálu v krizových situacích. Dotýká se business procesu a technologických aspektů IT infrastruktury. Cíle bezpečnostní strategie podniku zahrnují dosažení vysoké úrovně přístupu k výcviku personálu, vytváření procedur pro krizovou komunikaci a vývoj mechanismů pro rychlou mobilizaci zdrojů pro odvrácení, zmírnění následků a odstranění škod při případných havarijních stavech. Pro zajištění připravenosti a efektivity personálu v krizových situacích je potřeba úsilí, které vyžaduje koordinaci mezi různými odděleními, dotčených funkcí a využití pokročilých technologií pro sledování, analýzu a reakci v čase.

Hlavním výsledkem (outcome) projektu je zajištění schopné a efektivní reakce personálu na krizové situace. Diagram motivační vrstvy je znázorněn na **obr. 22 IT Projekt – Motivační vrstva**.



**Obrázek 21 IT Projekt – Motivační vrstva**

*Zdroj: vlastní zpracování*

### 6.3 Problém a popis současného stavu (As-Is)

Stávající stav prostředků technické podpory OHO je závislý na více systémech a telekomunikačních komponentách z let 2002 až 2010. Nyní provozované ICT řešení vykazuje nedostatky způsobené zastaralou technologií, zejména málo pružnou práci se seznamy svolávaných osob, nedostatečně zachycení výsledků svolání a nepřívětivé uživatelské rozhraní. To vyvolává potřebu důkladné modernizace.

Jedním z hlavních problémů je, že původní systém je navržen tak, aby komunikoval se specifickým hardwarem, především technologickými počítači s komunikačními kartami, které jsou již zastaralé a bez podpory ze strany výrobce.

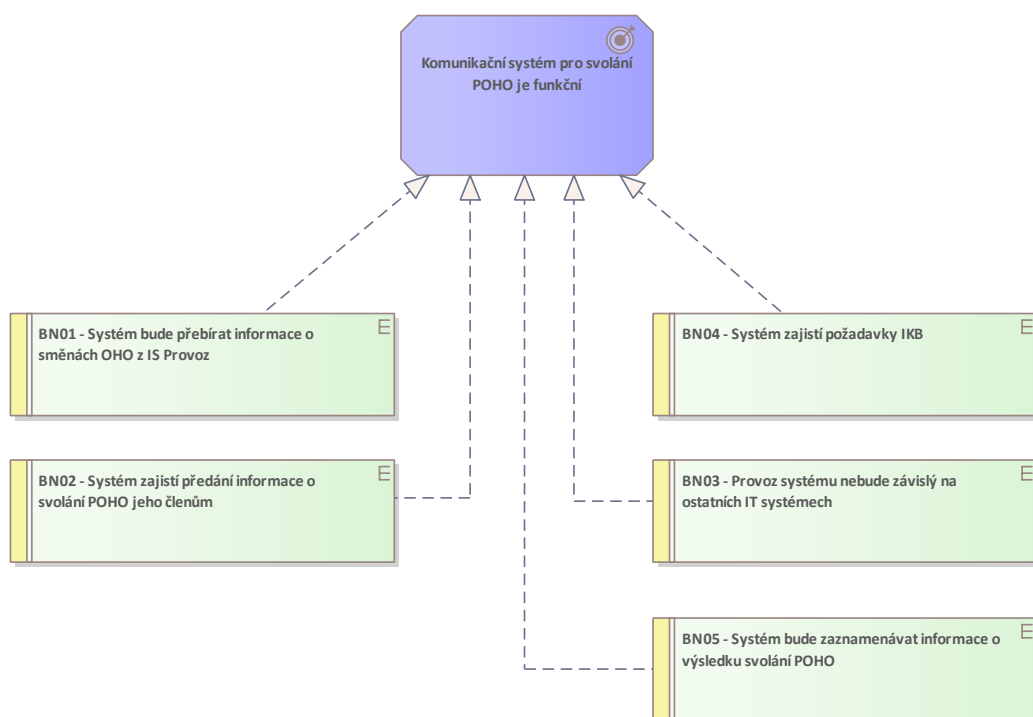
## 6.4 Cílový stav a popis žádoucího stavu (To-Be)

Cílem je získat inovovaný, moderní systém podporujícího přípravu a koordinaci svolání směn POHO, přípravu jejich seznamů, prokazatelné zachycení výsledků svolání s využitím perspektivní technologie. Součástí nového systému bude i plná podpora legislativních požadavků.

## 6.5 Business požadavky

Hlavním cílem dodaného systému je zajistit připravenost a efektivní reakci personálu v kritických situacích tím naplní business potřeby a zajistí plnou a efektivní podporu příslušných business procesů.

Pro dokumentování business potřeb existují různé přístupy. Dva základní způsoby jsou formou klasické tabulky a formou diagramu. Praktické provedení diagramu business potřeb na projektu je znázorněna na **obr. 23 IT Projekt – Business potřeby**.



**Obrázek 22 IT Projekt – Business potřeby**

*Zdroj: vlastní zpracování*

Někdy se používají pro tentýž účel i záznamy ve formě tabulek. Tabulkové záznamy poskytují strukturovaný přehled business potřeb. Na druhou stranu, diagramy nabízí výhody oproti tabulkám, zejména pokud jde o vizualizaci komplexních vztahů a jejich návaznost v procesu. Diagram umožňuje intuitivnější a dynamické zobrazení business potřeb. Pomáhá identifikovat vzory, závislosti a potenciální problémy, které nemusí být v tabulkovém formátu zřejmé.

Praktické provedení seznamu business potřeb na projektu formou seznamu je znázorněno v následující tabulce:

**Tabulka 1 IT Projekt – Business potřeby**

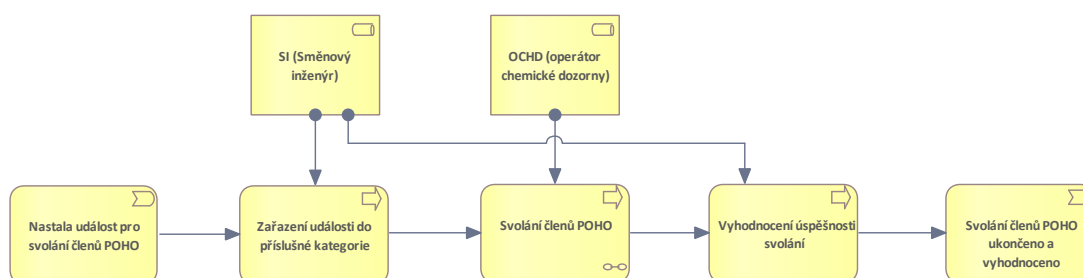
ID	Specifikace potřeby
BN01	Systém bude přebírat informace o směnách OHO z IS Provoz
BN02	Systém zajistí předání informace o svolání POHO jeho členům
BN03	Provoz systému nebude závislý na ostatních IT systémech
BN04	Systém naplní požadavky IKB
BN05	Systém bude zaznamenávat informace o svolání POHO

## 6.6 Business procesy

Hlavním cílem procesu je úspěšně svolat všechny členy POHO dle funkcí, kterých se dotýká vzniklá událost podle kategorie události.

### 6.6.1 Hlavní proces

Základní proces se skládá ze tří po sobě následujících podprocesů (viz *obr. 24 IT Projekt – Základní proces*).



**Obrázek 23 IT Projekt – Základní proces**

*Zdroj: vlastní zpracování*



## 6.6.2 Podproces svolání členů POHO

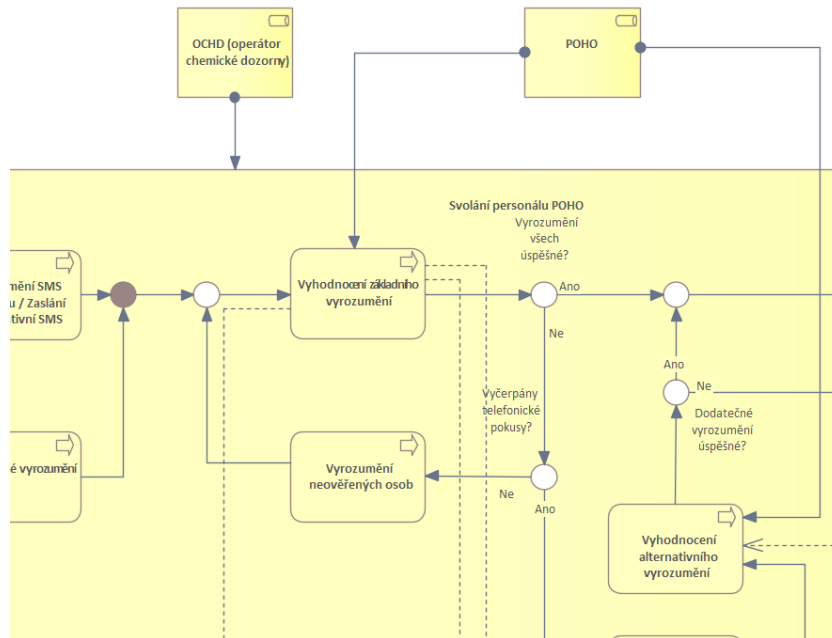
Proces zachycuje průběh vyrozumění všech členů POHO, kterých se týká vzniklá událost (ukázka procesu viz *obr. 25 IT Projekt – Ukázka procesu svolání personálu*). Celý proces je pak zachycen v *příloze 2 Kompletní proces*.

Na začátku procesu vybere operátor seznam dotčených osob pro vyrozumění. Základem pro seznam vyrozumívaných je aktuální přehled členů jednotlivých provozních směn, který je udržován v systému IS Provoz. Uživatel vybere seznam příslušné směny a může provést potřebné úpravy (přidání, vypuštění člena apod.), tím vznikne seznam vyrozumívaných osob. Následně pro tento seznam spustí vyrozumívající proces. Pokud základní typ vyrozumění (tel., SMS) nebude úspěšný, proběhne vyrozumění skrze alternativní variantu. V případě neúspěšnosti vyrozumění skrze základní i alternativní variantu je na dotčenou funkci vybrán náhradní člen.

Do protokolu jsou zaznamenány informace o události, jako jsou informace o kategorii události, datum a čas vzniku pokynu o svolání, jaká směna je aktivovaná a jaké její funkce jsou aktivovány.

Pro základní vyrozumění jsou do protokolu zaznamenány informace o jednotlivých členech na svolaných funkcích. U vyrozumívaných členů je veden záznam obsahující jeho jméno, příjmení, telefonní číslo, označení směny a funkce. Pro každého člena je veden záznam o průběhu vyrozumění – počet pokusů o telefonické vyrozumění, datum a čas spuštění vyrozumění a datum a čas přijetí vyrozumění. Pokud člen nebyl úspěšně vyrozuměn skrze základní vyrozumění, protokol obsahuje záznam v případě přijetí a potvrzení pagingové zprávy.

Po úspěšném vyrozumění dotčených členů POHO finalizuje OCHD protokol o vyrozumění. V této fázi doplní do protokolu, kdy byla předána informace o svolání HŠ a SI. Pokud byl některý ze svolávaných členů POHO vyrozuměn skrze alternativní formu, nebo byl vyrozuměn náhradní člen, zaznamená tuto skutečnost do poznámky.

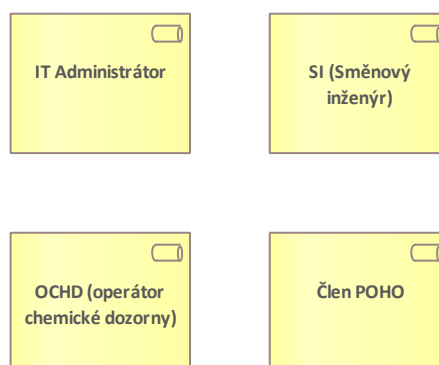


**Obrázek 24 IT Projekt – Ukázka procesu svolání personálu**

*Zdroj: vlastní zpracování*

## 6.7 Business role

Ve výše popsaném procesu figurují role, jejichž přehled je zobrazen na **Obr. 26 IT Projekt – Business role**.



**Obrázek 25 IT Projekt – Business role**

*Zdroj: vlastní zpracování*

## Směnový inženýr (SI)

Směnový inženýr stojí na začátku procesu svolávání. Role iniciuje celý proces a je schopen posoudit situaci a určit kategorii události.

## Operátor chemické dozorny (OCHD)

Po iniciaci procesu směnovým inženýrem má hlavní koordinační úlohu operátor. Role je zodpovědná za provedení potřebných úkonů pro svolání dotčených členů. Operátor používá předem stanovené postupy a komunikační nástroje pro distribuci informací a pokynů.

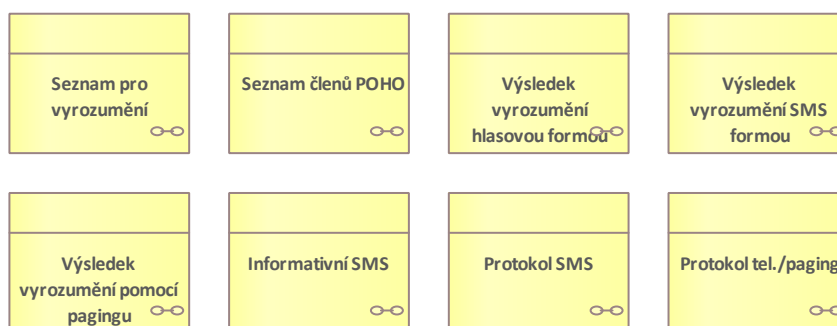
## POHO

Role člena POHO v procesu svolávání představuje koncového uživatele komunikačního řetězce. Členové jsou ty osoby, které mají být o události informovány a jsou přímo zapojeni do reakce na událost. Úkolem člena v rámci procesu svolání je potvrzení přijetí vyzoomění.

## 6.8 Business objekty

V rámci procesu se využívá několik business objektů pro správu a koordinaci komunikačních aktivit.

Business objekty (viz *obr. 27 IT Projekt – Výčet business objektů*) lze rozdělit do tří kategorií, seznamy členů, záznamy o vyzoomění a výstupní protokoly.



**Obrázek 26 IT Projekt – Výčet business objektů**

*Zdroj: vlastní zpracování*

## **Seznamy členů**

- Seznam pro vyrozumění – Objekt obsahuje seznam kontaktů, které uživatel vytvořil a spravuje lokálně. Umožňuje uživatelům přidávat, upravovat, nebo odstraňovat kontakty a je základem pro cílenou komunikaci s týmy, nebo specifickými zaměstnanci mimo standardně vyrozumívané osoby.
- Seznam členů POHO – Je výsledkem integrace s externí aplikací. Obsahuje kontakty, které byly synchronizovány z vnějšího zdroje. Integrace umožňuje automatické aktualizace a zajišťuje, že seznam zůstává aktuální s minimální potřebou manuálního zásahu.

## **Záznamy o vyrozumění**

- Telefonické vyrozumění – Zobrazuje záznamy a průběh telefonických vyrozumění. Poskytuje podrobné informace o každém spuštěném vyrozumění, včetně času, trvání a výsledku.
- SMS vyrozumění – Zobrazuje odeslané SMS zprávy v rámci vyrozumění SMS formou, včetně obsahu zprávy, času odeslání a stavu doručení. Umožňuje rychlou kontrolu úspěšnosti komunikačních kampaní a identifikaci případných problémů s doručením.
- Pagingové vyrozumění – Objekt pagingových vyrozumění umožňuje sledování přijetí pagingových zpráv.

## **Výstupní protokoly**

- Protokol SMS vyrozumění – Detailně dokumentuje veškeré informace týkající se odeslaných SMS, včetně data, času, příjemců a obsahu zprávy.
- Protokol telefonického vyrozumění – Obsahuje záznamy o telefonních hovorech realizovaných v rámci vyrozumění, včetně údajů o volajícím, příjemci, času hovoru a jeho výsledku.
- Protokol pagingového vyrozumění – Zaznamenává výstup z použití pagingové služby pro vyrozumění.

## 6.9 Požadavky na systém

Při sběru požadavků na IT systém bylo postupováno metodicky a s cílem získat co nejhlubší porozumění potřeb zákazníka. Metodologie byla rozdělena do fází s komplexním přístupem k analýze a definici požadavků.

### 6.9.1 Metodologie sběru požadavků použitá na projektu

První fáze byla zaměřena na porozumění stávajícím business modelům organizace. K tomu bylo využito nástroje pro vizualizaci a analýzu podnikových procesů ARIS, který mapuje klíčové procesy, zdroje a zákaznické interakce.

Následně byla provedena analýza existujících řídicích dokumentů a příruček, včetně interních předpisů a směrnic. Průzkum nám umožnil pochopit stávající postupy a pravidla, která musí být při návrhu nového systému zohledněna.

Nejdůležitější součástí metodologie byla pravidelná spolupráce se stakeholdery. Efektivní business analýza vyžaduje porozumění potřebám, očekáváním a preferencím klíčových stakeholderů, včetně výkonných pracovníků, zákazníků a dodavatelů. Pro tento účel byly organizovány workshopy, konzultace a meetingy. Vzájemná interakce umožnila identifikovat a dokumentovat specifické požadavky a zajistit, že navržené změny odpovídají potřebám stakeholderů.

Celý proces sběru požadavků byl dokumentován pro základ další fáze projektu. Požadavky byly pravidelně revidovány a validovány se stakeholdery, aby bylo zajištěno, že zůstávají aktuální a v souladu s business cíli.

### 6.9.2 Funkční požadavky

Při přípravě funkčních požadavků byly základem business potřeby. Byla snaha, aby každá business potřeba byla reprezentována určitou skupinou funkčních požadavků. Při definování požadavků je důležité, aby žádný z nich svou pracností výrazně nepřevyšoval požadavky ostatní. V takovém případě byl požadavek rozdělen na sdružené požadavky, případně jiné samostatné. Seznam sesbíraných požadavků se uveden v *tabulce 2 IT Projekt – Funkční požadavky*.

**Tabulka 2 IT Projekt – Funkční požadavky**

ID	Název požadavku	Parametry požadavku (Customer Satisfaction Criteria)	Číslo UC
RQF01	Aktuálnost barev směn OHO	Systém umožní přenesení seznamu směn POHO podle barev směn OHO IS Provoz.	UC01
RQF01.1	Přenos barev směn	Systém bude přenášet seznamy formou REST API služby.	UC01
RQF02	Odbornosti směn OHO	Systém přeneše seznam odborností směn OHO z IS Provoz.	UC01
RQF02.1	Přenos odborností	Systém bude přenášet seznamy formou REST API služby.	UC01
RQF03	Funkce pro odbornosti OHO	Systém přeneše seznam profesí pro odbornosti směn OHO z IS Provoz.	UC01
RQF03.1	Přenos funkcí pro odbornosti	Systém bude přenášet seznamy formou REST API služby.	UC01
RQF04	Seznamu členů OHO	Systém zajistí přenos seznamu členů směn OHO z IS Provoz.	UC01
RQF04.1	Přenos seznamu členů	Systém bude přenášet seznamy formou REST API služby.	UC01
RQF05	Plánovaný seznam členů směn OHO	Systém poskytne seznam všech plánovaných členů směn OHO na konkrétní den(dny) z IS Provoz.	UC01
RQF05.1	Přenos plánovaného seznamu členů	Systém bude přenášet seznamy formou REST API služby.	UC01
RQF06	Aktuálnost směn	Systém zobrazí seznam aktuálně sloužících členů a rozděljuje členy podle barvy směny.	UC01
RQF07	Vlastní seznam členů	Systém umožní vytvoření vlastního seznamu členů. Členy je možné libovolně zařazovat do vlastních skupiny.	UC01
RQF07.1	Editace vlastního seznamu	Systém umožní editaci vlastního seznamu.	UC01
RQF08	Vyrozumění členů POHO hlasovou formou	Systém umožní kontaktování vybraných členů a skupin přenesených z IS Provoz (webová služba) či vlastních skupin hlasovou formou.	UC04
RQF9	Editace seznamu pro vyrozumění hlasovou formou	Systém umožní editovat seznam členů. Umožní přidávat, nebo odebrat dalšího libovolného člena i tel. číslo, které není vedeno v žádném seznamu členů. Tento seznam bude dočasný a nebude ukládán. Slouží pouze pro finální úpravu příjemců vyrozumění.	UC04a
RQF9.1	Hromadné úpravy seznamu pro vyrozumění	Systém umožní hromadné přidání, nebo odebrání členů seznamu pro vyrozumění. Umožní to podle názvu směn (seznam členů) a funkcí příslušných dané směně.	UC04a
RQF10	Paralelní odeslání SMS zprávy	Systém umožní souběžné odeslání SMS zprávy a hlasové výzvy.	UC04
RQF11	SMS zpráva při neúspěšném vyrozumění	Systém umožní odeslání druhé zprávy při vyčerpání pokusů o telefonické vyrozumění.	UC04
RQF12	Výsledek vyrozumění hlasovou formou	Systém zaznamená výsledky jednotlivých vyrozumění hlasové formy. Výsledek bude obsahovat informace o počtu úspěšných či neúspěšných telefonických vyrozumění.	UC05a
RQF12.1	Fronta vyrozumění hlasovou formou	Systém umožní zobrazení probíhající fronty telefonického vyrozumění.	UC05a
RQF12.2	Doba uložení výsledku vyrozumění hlasovou formou	Systém zajistí uložení výsledků vyrozumění po dobu 6ti měsíců.	UC05a
RQF13	Vyhodnocení úspěšnosti kontaktování	Systém po přijetí vyrozumění a přehrání hlasové zprávy vyhodnotí, pomocí manuálního potvrzení vyrozumívaného (netýká se zvednutí hovoru), zda bylo vyrozumění úspěšné.  Podbarvení jmen členů v seznamu vyrozumívaných se	UC04

		budou měnit v návaznosti na úspěšnost či neúspěšnost vyrozumění.	
RQF14	Zaznamenání totožnosti	Systém po přijetí vyrozumění a přehrání hlasové zprávy zaznamená totožnost vyrozumívaného (formou uložení hlasové zprávy) – jméno, příjmení, funkce.	UC04
RQF15	Vzorové hlasové zprávy	Systém umožní při vyrozumění využít vzorové hlasové zprávy.	UC07
RQF15.1	Příprava vzorové hlasové zprávy	Systém umožní vytvoření a uložení nové vzorové hlasové zprávy.	UC07
RQF16	Vlastní hlasové zprávy	Systém umožní při vyrozumění využít vlastní hlasovou zprávu. Zpráva bude použita pouze pro jedno vyrozumění a nebude uložena.	UC04
RQF17	Generování protokolu pro členy POHO – tel.	Systém po dokončení vyrozumění umožní vygenerovat protokol obsahující informace dle BO.  Tento formulář je využitý pro všechny členy součástí skupiny POHO.	UC03
RQF18	Generování protokolu pro členy mimo POHO – tel.	Systém po dokončení vyrozumění umožní vygenerovat protokol obsahující informace dle BO.  Tento formulář je využitý pro všechny členy mimo skupinu POHO.	UC03
RQF19	Opakované telefonické vyrozumění	Systém umožní systémově nastavit množství opakování telefonických vyrozumění.	UC09
RQF20	Vyrozumění formou SMS	Systém umožní kontaktování vybraných členů a skupin přenesených z IS Provoz (webová služba) či vlastních skupin SMS formou.	UC04
RQF21	Editace seznamu vyrozumívaných SMS	Systém umožní editovat seznam vyrozumívaných. Umožní přidávat, nebo odebírat dalšího libovolného příjemce i tel. číslo, které není vedeno v seznamu členů.  Tento seznam bude dočasný a nebude ukládán. Slouží pouze pro finální úpravu příjemců vyrozumění.	UC04a
RQF21.1	Hromadné úpravy seznamu pro vyrozumění	Systém umožní hromadné přidání, nebo odebrání členů pro vyrozumění. Umožní to podle barvy směn a funkcí příslušných dané směně.	UC04a
RQF22	Vzorové SMS zprávy	Systém umožní při vyrozumění využít vzorové SMS zprávy.	UC08
RQF22.1	Příprava vzorové SMS zprávy	Systém umožní vytvoření a uložení nové vzorové SMS zprávy.	UC08
RQF23	Vlastní SMS zprávy	Systém umožní pro vyrozumění připravit vlastní SMS zprávu. Zpráva bude použita pouze pro jedno vyrozumění a nebude uložena.	UC08
RQF24	Fronta zpráv	Systém umožní zobrazení a úpravu probíhající fronty odesílaných zpráv.	UC05b
RQF25	Výsledek vyrozumění SMS formou	Systém zaznamená výsledky jednotlivých vyrozumění SMS formou. Výsledek bude obsahovat informace o počtu úspěšných či neúspěšných, čekajících SMS vyrozumění.	UC05b
RQF25.1	Doba uložení výsledku vyrozumění SMS formou	Systém zajistí uložení výsledků vyrozumění po dobu 6ti měsíců.	UC05b
RQF26	Informativní SMS	Systém umožní odeslání informativní SMS, která obsahuje přístup k nahranému hlasovému záznamu.	UC04
RQF27	Seznam informativních SMS	Systém umožní zobrazit seznam hlasových záznamů. Po uplynutí 24 hodin je PIN zneplatněn.	UC05
RQF27.1	Doba uložení informativních SMS	Systém zajistí uložení výsledků vyrozumění po dobu 24 hodin.	UC05
RQF28	Generování protokolu – SMS	Systém po vyrozumění umožní vygenerovat protokol obsahující informace dle BO.	UC03
RQF29	Monitoring stavu funkcí GSM brány	Systém zobrazí stav GSM brány.	UC06
RQF30	Monitoring stavu telefonních linek	Systém zobrazí stav telefonních linek.	UC06

RQF31	Zobrazení stavu importu kontaktů	Systém zobrazí čas posledního importu seznamu kontaktů z IS Provoz.	UC06
RQF32	Kontrola vyznění pomocí pagingu	Systém zobrazí seznam potvrzených či nepotvrzených svolání pagingu.	UC05b
RQF33	Generování protokolu – pager	Systém po vyznění umožní vygenerovat protokol obsahující informace dle BO.	UC03
RQF34	Hlasový vstup do systému	Systém umožní vytvořit hlasovou zprávu a následně distribuovat tuto zprávu pomocí hlasového volání nebo formou informační sms. Tento požadavek je realizován hlasovým vstupem do systému z telefonní linky.	UC04b

### 6.9.3 Nefunkční požadavky

Nefunkční požadavky nemusí nutně vždy vycházet z business potřeb. Může se jednat o obecné požadavky na vzhled, prostředí, lokalizaci a další parametry platné pro všechny IT nástroje v dané organizaci. Seznam nefunkčních požadavků konkrétního projektu je uveden v *tabulce 3 IT Projekt – Nefunkční požadavky*.

**Tabulka 3 IT Projekt – Nefunkční požadavky**

ID	Specifikace/název požadavku	Parametry požadavku (Customer Satisfaction Criteria)
RQN01	Získání informací o směnách POHO	Systém bude získávat informace o členech směn POHO z IS provoz.  Integrace bude řešena formou REST API služeb vystavených na API Gateway.
RQN02	Podpora použitého HW	Použitý HW bude mít servisní podporu na více než 4 roky.
RQN03	Shodný použitý HW na lokalitách	Použitý HW bude na obou lokalitách stejný.
RQN04	AD + lokální ověření přihlášení	Systém umožní ověření přihlašovaného uživatele standardně přes AD, ale umožní i ověření přímo v aplikaci.
RQN05	Logování systému	Systém umožní logování základních aktivit uživatele. Dle kapitoly: 3.3.11
RQN06	Přihlášení na lokality	Systém umožní přihlášení z jedné lokality do druhé a plnohodnotné ovládání OKS POHO v druhé lokalitě.
RQN07	Jednoduchost ovládání	Systém zajistí maximální jednoduchost pro svolání POHO. (max 10 kliknutí, bez editace dat a vyznívaných)
RQN08	Dostupnost systému	Systém bude navržen tak, aby byl dostupný 24/7, s minimálními přerušeními služby

### 6.9.4 Architektonické požadavky

Architektonické požadavky vychází z potřeby začlenit a integrovat IT řešení do prostředí organizace tak, aby bylo možno jej snadno udržovat, provozovat a řešit poruchové stavy. Architektonické požadavky pro konkrétní projekt jsou uvedeny v *tabulce 4 IT Projekt – Architektonické požadavky*.



**Tabulka 4 IT Projekt – Architektonické požadavky**

ID	Specifikace/název požadavku	Parametry požadavku (Customer Satisfaction Criteria)
AQ01	Integrace na externí systém	Systém bude schopen se integrovat s aplikací IS Provoz a bude z něj získávat a naplňovat směny POHO.
AQ02	Nezávislost lokalit	V případě výpadku jedné lokality bude možné vzájemná zastupitelnost lokalit.
AQ03	SMS gateway	Systém musí využívat existující SMS gateway

### 6.9.5 Omezení

Systém nesmí vyžadovat trvalé připojení k síti Internet.

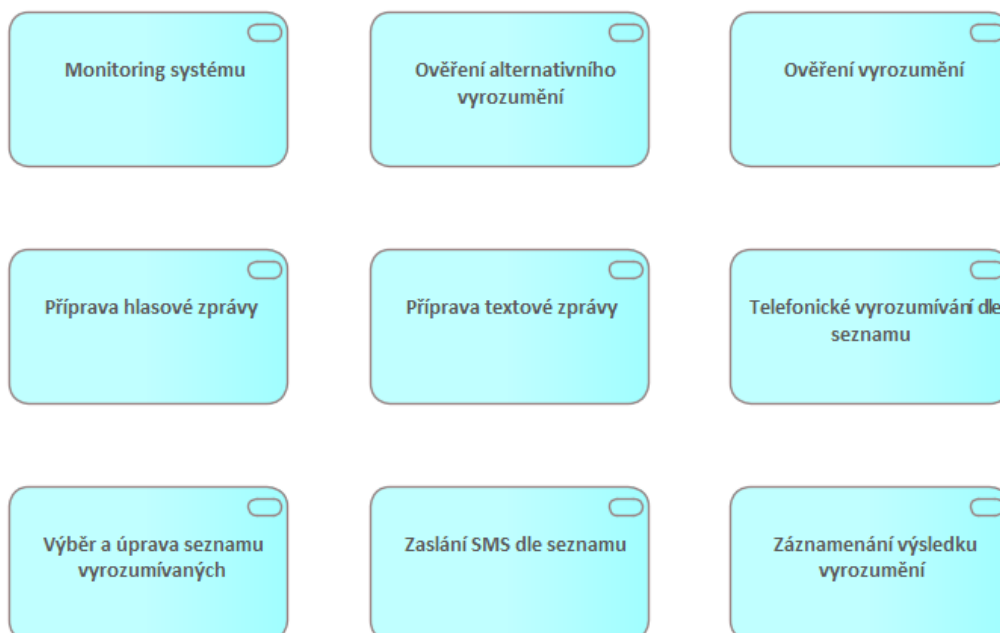
Systém nesmí vyžadovat trvalé propojení obou lokalit.

## 6.10 Postup návrhu architektury

Při návrhu architektury začíná role analytik spolupracovat s rolí solution architekta. Architekt přebírá analytikem požadované aplikační služby pro podporu business procesu, navrhuje komponenty pro jejich realizaci. Při návrhu se snaží maximálně využít již existujících služeb a zamezit redundanci nezbytných komponent.

### 6.10.1 Požadované aplikační služby

V analytické fázi IT projektu byly identifikovány aplikační služby (viz *obr. 28 IT Projekt – Aplikační služby*). Služby představují funkční jednotky nezbytné pro naplnění požadavků uživatelů.



**Obrázek 27 IT Projekt – Aplikační služby**

*Zdroj: vlastní zpracování*

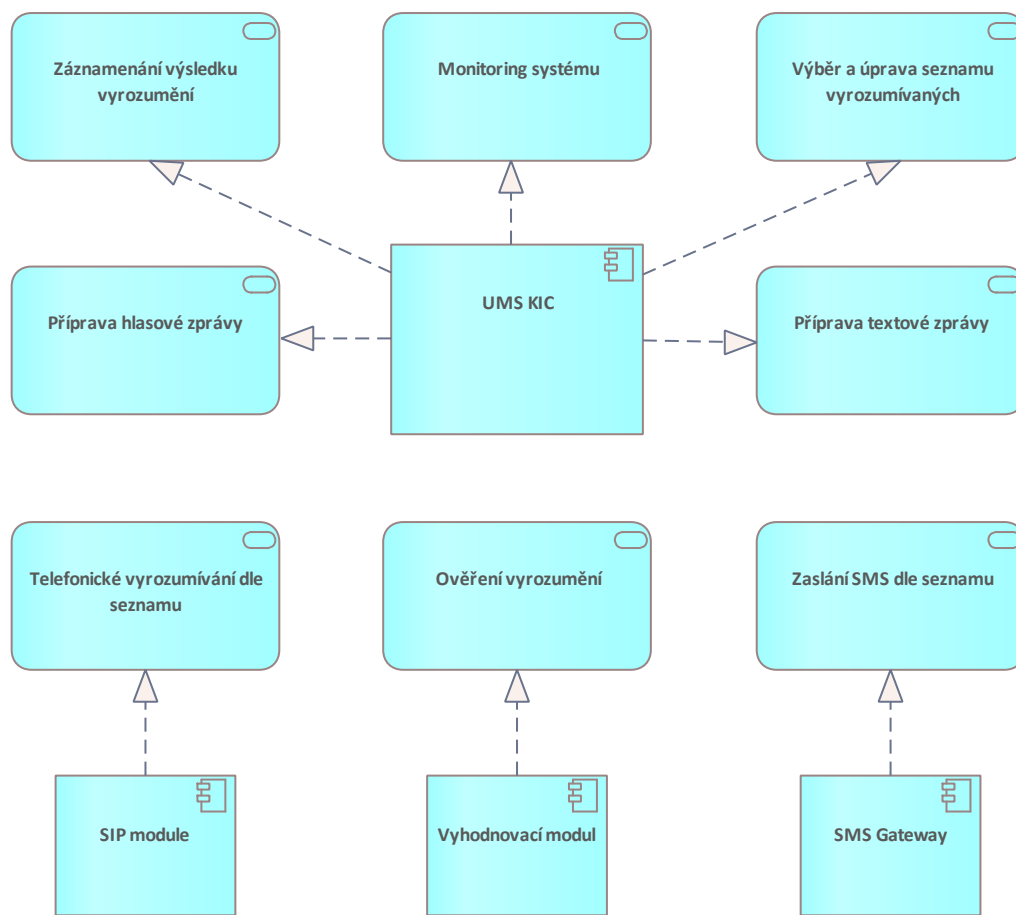
### 6.10.2 Potřebné aplikační komponenty

Požadované aplikační služby posoudí další člen projektového týmu, solution architekt, který je zodpovědný za konceptuální návrh technického řešení.

Při konceptuálním návrhu solution architekt projektu hledá aplikační komponenty naplňující svou funkčností požadované aplikační služby. Potřebné komponenty mohou být součástí již existující podnikové architektury.

Například existující podniková SMS Gateway bude odpovědná za zaslání SMS dle seznamu. Služba na ověření vyznění se bude muset realizovat nově, a to v komponentě KS POHO.

Navržené naplnění požadované funkčnosti aplikačních služeb je zachyceno na **obr. 29 IT Projekt – Aplikační Komponenty**.



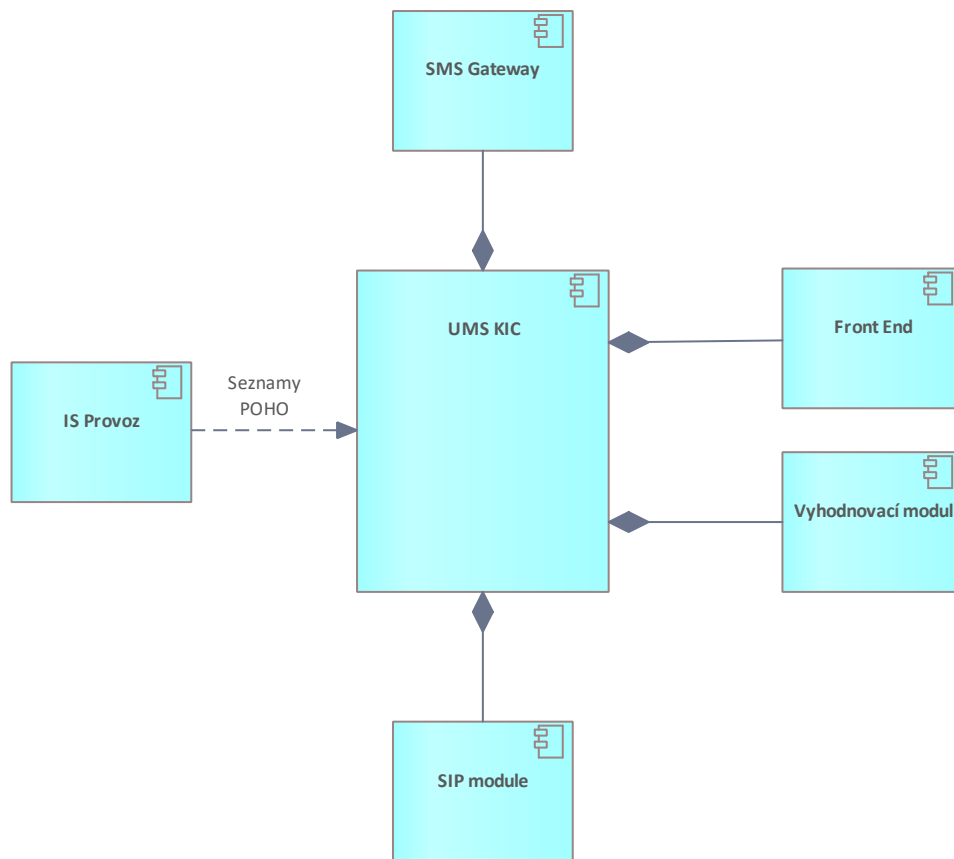
**Obrázek 28 IT Projekt – Aplikační komponenty**

*Zdroj: vlastní zpracování*

### 6.10.3 Konceptuální návrh architektury

Konceptuální návrh architektury vychází z návrhu aplikačních komponent, přidává k nim vazby mezi nimi, potřebná napojení další podpůrné systémy či další integrace.

Představuje základní návrh řešení, který je v dalších fázích projektu po potvrzení dále rozvíjen a rozpracován do detailu v rámci cílového konceptu. Může sloužit jako podklad pro konzultaci s firmami, které budou spolupracovat na jeho realizaci. Je znázorněn na **obr. 30 IT Projekt – Konceptuální návrh architektury**



**Obrázek 29 IT Projekt – Konceptuální návrh architektury**

*Zdroj: vlastní zpracování*

## 6.11 Případy užití

Při návrhu případů užití analytik vychází z jednotlivých kroků business procesů, snaží se najít naplnění funkčních požadavků a připravuje tím základ pro testovací scénáře.

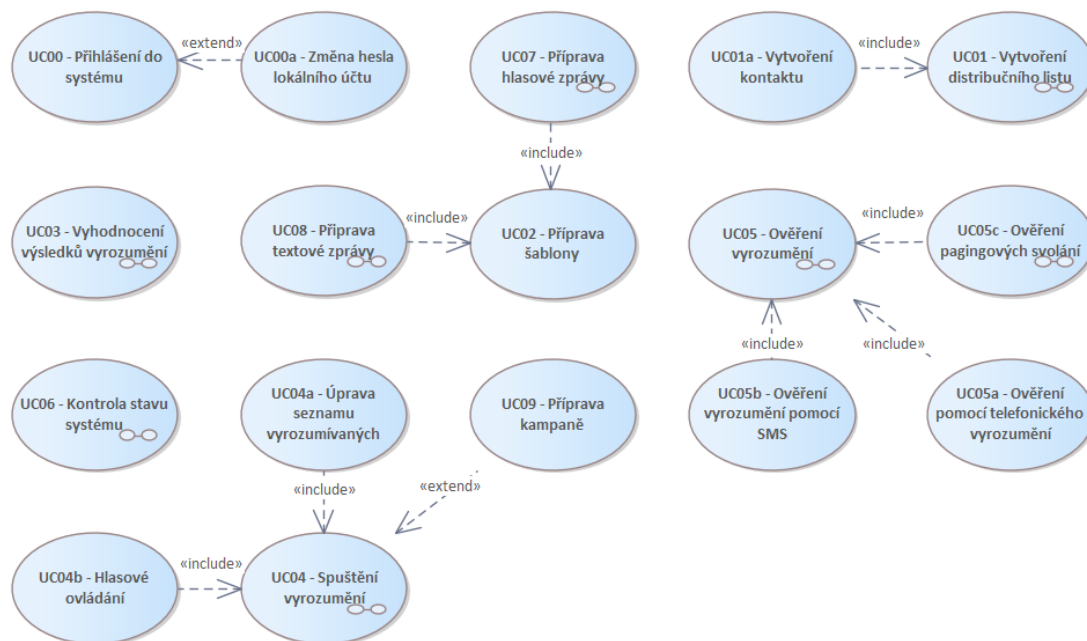
### 6.11.1 Přehled případů užití

Případy užití jsou tvořeny s cílem naplnit uživatelské funkční požadavky (viz **obr. 32 IT Projekt – Vazba požadavků na případ užití**).

Jejich přehled odpovídá předpokládanému rozsahu interakcí uživatele IT systémem v rámci naplňování jednotlivých business kroků.

Přehled očekávaných případů užití na projektu je uveden na následujícím schématu

**obr. 31. IT Projekt – Přehled případů užití:**



**Obrázek 30 IT Projekt – Přehled případů užití**

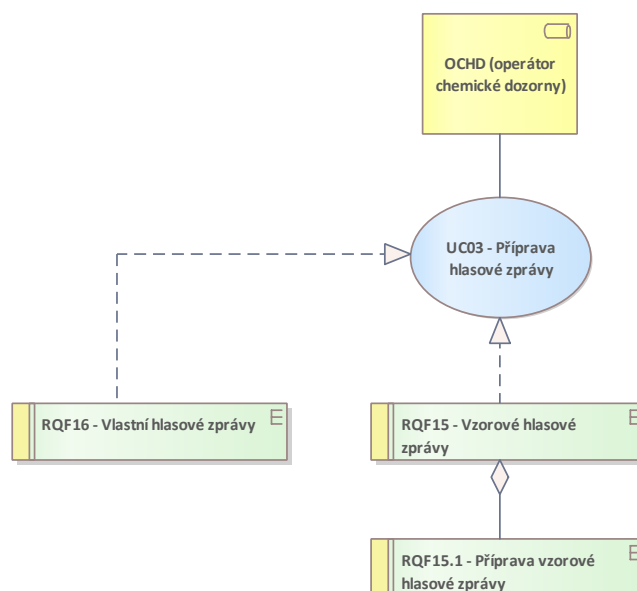
*Zdroj: vlastní zpracování*

Jejich použití se ale může rozšířit i mimo tuto bezprostřední oblast. V některých situacích může být případ užití vytvořen bez přímého naplnění specifického požadavku uživatele.

### 6.11.2 Detail konkrétního případu

Typy případů užití bez vazby na funkční požadavek

- Pre-condition – Případ užití může být předpokladem pro jiné, složitější případy užití. Představuje krok, nebo sérii úkonů, které musí být provedeny před spuštěním dalšího procesu.
- Sdružení případů užití – Jedná se o sdružení, nebo agregaci více funkcionalit do skupiny. To umožňuje lépe organizovat a strukturovat případy užití.
- Komplexní scénáře – Případ užití vytvořen k reprezentaci části systému, která je nezbytná pro jeho funkci, ale neodpovídá žádnému z uživatelských požadavků.



**Obrázek 31 IT Projekt – Vazba požadavků na případ užití**

*Zdroj: vlastní zpracování*

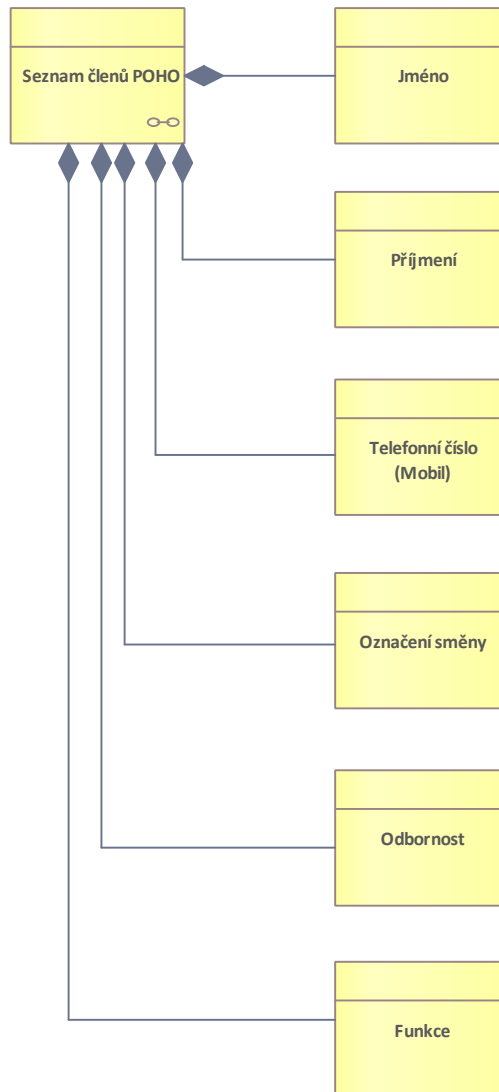
## 6.12 Datový model

Při návrhu datového modelu role analytik spolupracuje s rolí datového architekta. Architekt přebírá analytikem požadované business objekty s jejich atributy, navrhuje datové objekty pro jejich realizaci.

### 6.12.1 Konceptuální datový model

Konceptuální datový model je připraven v jazyku ArchiMate a představuje další krok vycházející ze seznamu business objektů identifikovaných v rámci analýzy business procesů. Seznam business objektů je zachycen na **obr. 27. IT Projekt – Výčet business objektů**. Dalším krokem je analýza potřebných business atributů daného objektu. K tomu opět slouží meeting se zástupci business strany, dotazování a zachycení výsledku. Příklad výstupu pro jeden konkrétní business objekt je zachycen na **obr. 33 IT Projekt – Konceptuální datový model – ukázka**.

Podobným způsobem postupujeme u dalších business objektů. Kromě zachycení požadovaných atributů je potřeba prozkoumat a zaznamenat případné vazby a závislosti mezi jednotlivými objekty.



**Obrázek 32 IT Projekt – Konceptuální datový model – ukázka**

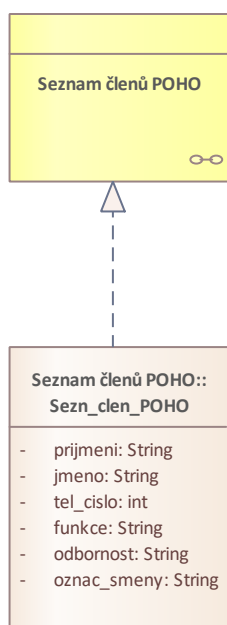
*Zdroj: vlastní zpracování*

### 6.12.2 Logický datový model

Dalším krokem v přípravě datového modelu je převod business objektů a jejich atributů do podoby srozumitelné pro návrháře datových objektů v příslušné databázi zvolené pro technické řešení. Zde bylo spolupracováno s datovým architektem, který

má potřebné znalosti o strukturách databází, možnostech zobrazení jednotlivých atributů, návrhu fyzických tabulek včetně klíčových položek a vazeb mezi nimi.

Ukázka logické reprezentace business objektu zachycené elementem jazyku Archimate v konkrétním logickém objektu tabulky reprezentované třídou jazyka UML je na následujícím *obr. 34 IT Projekt – Logický datový model – ukázka*.



**Obrázek 33 IT Projekt – Logický datový model – ukázka**

*Zdroj: vlastní zpracování*

Spolupráce analytika a datového architekta je v této části projektu velmi důležitá, a často vyžaduje aspoň částečný přesah znalostí obou osob do oblasti spolupracující osoby. Jenom tak může dojít k přesnému návrhu, který v implementační fázi vyžaduje minimální korekce a úpravy.

## 6.13 Implementace projektu

Ve fázi implementační spočívá největší úloha analytika v rozpracování případů užití na konkrétní scénáře tak, aby bylo možno co nejpřesněji specifikovat detaily použitých aplikačních komponent.



### 6.13.1 Detailní popis případů užití

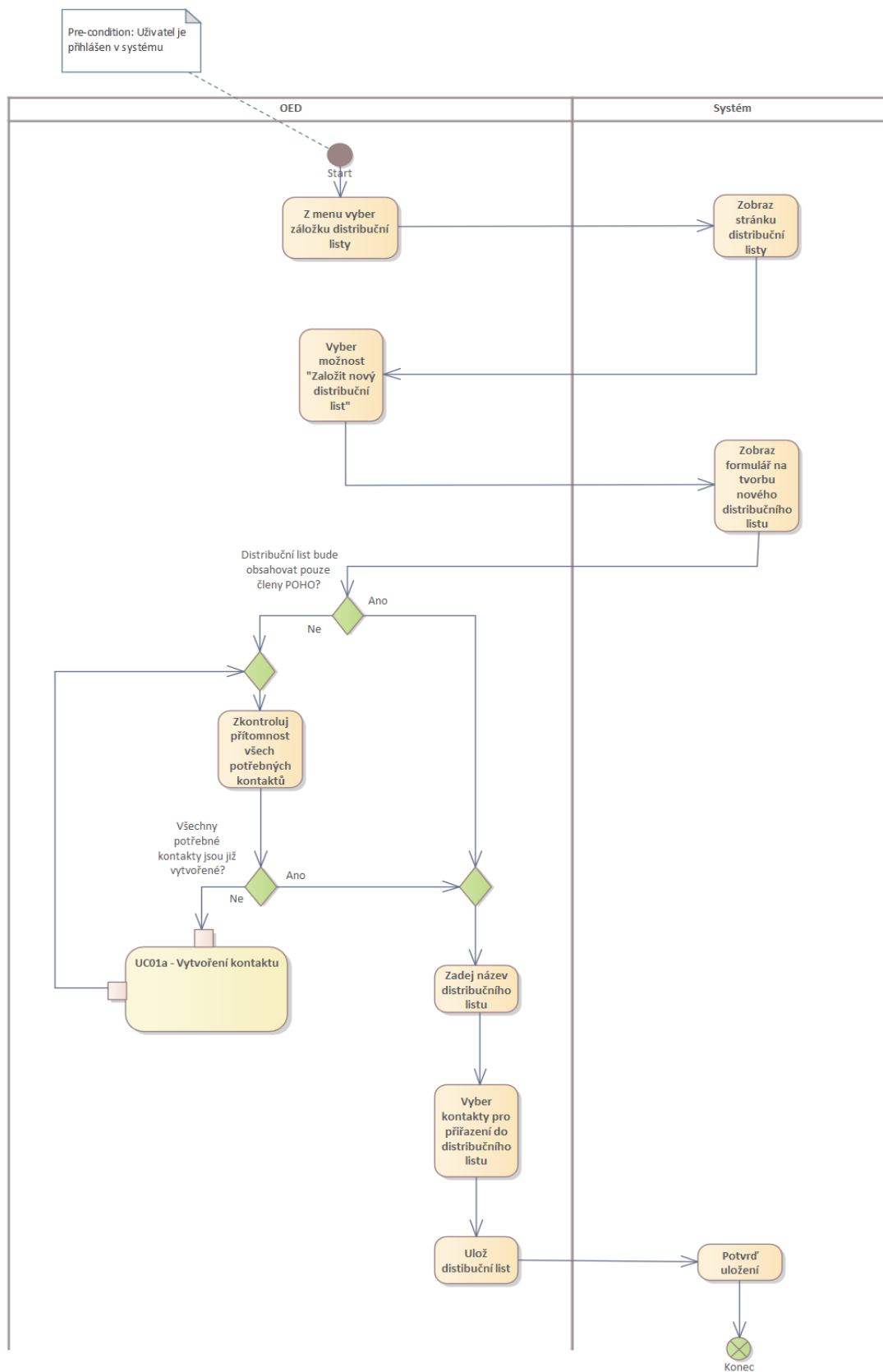
Detailní rozpracování případů užití má dva hlavní účely. Prvním z nich je podklad pro přípravu workflow a zdrojových kódů k detailní přípravě aplikačních komponent. Druhým je příprava testovacích scénářů na ověření naplnění jednotlivých požadavků.

Existují dva rozdílné přístupy k detailnímu modelování případů užití.

První přístup využívá grafické znázornění procesů a toku akcí formou diagramu aktivit, které umožňují vizuálně reprezentovat kroky a rozhodovací body zapojené do případu užití. Příklad je znázorněn na *obr. 35 IT Projekt – Založení distribučního listu*.

Druhým přístupem je textová, popisná forma. Popisuje scénář užití prostřednictvím definované struktury textu, zahrnující aktéry, předpoklady, kroky interakce a očekávané výsledky. Formát poskytuje náhled na funkce systému a způsob, jakým uživatelé s těmito funkcemi interagují, včetně alternativních toků a výjimečných situací. K doplnění detailního popisu lze využít i náhled obrazovky pro lepší představu o uživatelském rozhraní připravované aplikace.

Neexistuje jasné pravidlo pro volbu jednoho či druhého přístupu, a je na zkušenosti analytika, který přístup zvolí. Obecně však lze říci, že popisný způsob je vhodnější pro méně komplikované případy užití s minimem rozhodovacích kroků a následných větví popisu. U složitějších případů poskytne grafická forma lepší přehled a orientaci jak pro analytika, tak pro vývojáře, který případ užití realizuje či pro konečného uživatele, který výsledek bude schvalovat a testovat.



**Obrázek 34 IT Projekt – Založení distribučního listu**

*Zdroj: vlastní zpracování*

Use Case: Přihlášení do systému

ID: UC00

Název: Přihlášení do systému

Aktéři: Uživatel

Předpoklady: Nejsou

**Postup:**

1. Uživatel otevře tenkého klienta.
2. Systém se pokusí autentizovat uživatele pomocí Windows autentizace.
  - Pokud je active directory (AD) dostupná:
    1. Uživatel je automaticky přihlášen bez nutnosti zadávat jméno a heslo.
    2. Use case končí.
  - Pokud AD není dostupná:
    1. Systém zobrazí formulář pro přihlášení pomocí lokálního účtu (viz **obr. 36 IT Projekt – Návrh přihlašovací obrazovky**).
    2. Uživatel zadá svůj email a heslo.
      - Pokud uživatel zná správné heslo:
        1. Uživatel je přihlášen do systému.
        2. Use case končí.
      - Pokud uživatel nezná heslo:
        1. Uživatel klikne na odkaz pro reset hesla.
        2. Uživatel zadá svůj email a potvrdí žádost o reset hesla.
        3. Systém odešle email s jednorázovým heslem.
        4. Uživatel otevře email a použije jednorázové heslo pro přihlášení.
        5. Uživatel je přihlášen do systému.
        6. Uživatel rozklikne svůj profil a vyplní formulář na změnu hesla.
        7. Use case končí.

Tab1  
https://OKSPOHO.corp

Uživatelské jméno

Heslo

Přihlásit se

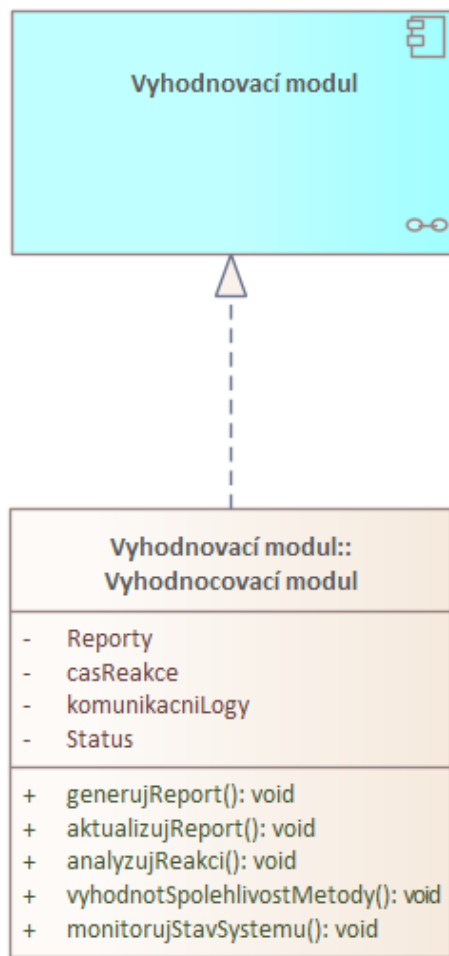
Zapoměl/a jsem heslo

***Obrázek 35 IT Projekt – Návrh přihlašovací obrazovky***

*Zdroj: vlastní zpracování*

### 6.13.2 Detail komponenty

Třída představuje softwarovou komponentu určenou pro sběr, analýzu a reportování dat souvisejících s procesy svolávání pohotovostí viz *obr. 37 IT Projekt – Příklad znázornění detailu komponenty*.



**Obrázek 36 IT Projekt – Příklad znázornění detailu komponenty**

*Zdroj: vlastní zpracování*

## 7 Závěr

Tato práce se pokusila na konkrétním případu ukázat konkrétní postupy vedoucí od úvodního pochopení zadání až ke konkrétnímu technickému řešení IT projektu. V kapitolách 2.2 a 2.3 jsou zmíněny zásady modelování, které poskytují teoretický základ pro efektivní využití analytických nástrojů a modelovacích jazyků. V části praktické jsou tyto zásady promítnuty do konkrétního postupu analytické práce na projektu.

Práce ukazuje možnosti kombinace použití dvou nejužívanějších modelovacích jazyků. Přestože tento kombinovaný postup modelování není úplně běžný, může vést k lepšímu zachycení popsaného problému, požadavků a potřeb i návrhu technického řešení. Práce ukazuje jeden z možných přístupů využití silných stránek obou zmíněných modelovacích jazyků v různých vrstvách modelu. Může posloužit případně i jako inspirace při řešení podobných zadání v konkrétní podnikové praxi.

Při přípravě konkrétního projektu se osvědčila zamýšlená koncepce použití dvou modelovacích jazyků. Jazyk ArchiMate díky své názornosti a relativní jednoduchosti dobře posloužil při přípravě zadání, diskusi nad business procesem a formulaci základních požadavků.

Diagramy v tomto jazyce byly dobře srozumitelné i těm zástupcům business strany, kteří se s ním setkali poprvé.

V dalších fázích, kdy byla potřeba nižší míra abstrakce a větší míra detailu, ukázal prospěšnost jazyk UML i díky tomu, že je poměrně dobře znám napříč rolemi architektů a vývojářů.

## 8 Seznam zkratek

AD	Active directory
ARIS	Architektura integrovaných informačních systémů
BA	Business analýza
BABOK	Business analysis body of knowledge
BPMN	Business process model and notation
CBAP	Certified business analysis professional
HŠ	Havarijní štáb
IBBA	Mezinárodní institut business analytiků
KS	Komunikační systém
MDA	Model driven architecture
MDA	Model driven architecture
MDD	Model driven development
MDD	Model driven development
OHO	Organizace havarijní odezvy
OHO	Organizace havarijní odezvy
OCHD	Operátor chemické dozorny
POHO	Podpora organizace havarijní odezvy
SI	Směnový inženýr
TOGAF	The open group architecture framework
UML	Unified modeling language

## 9 Seznam použitých zdrojů

- *ArchiMate*. (2024, January 20). Wikipedia: the free encyclopedia. Retrieved January 20, 2024, from <https://cs.wikipedia.org/wiki/ArchiMate>
- *BA Proces*. (2020).
- Fowler, M. (2009). *Destilované UML* (1st ed.). Grada.
- Hosiaislouma, E. (2019). *ArchiMate Cookbook*. ArchiMate Cookbook. Retrieved January 22, 2024, from <http://www.hosiaislouma.fi/ArchiMate-Cookbook.pdf>
- McConnell, S. (2006). *Odhadování softwarových projektů* (1st ed.). CPress.
- OpenAI. (2023). ChatGPT (Verze 3.5.) [Software]. <https://www.openai.com/chapgpt>
- Rydval, S. (2019). *UML pro analytiky* (1st ed.). Elebedial.
- The Open Group. (2023). *ArchiMate® 3.2 Specification*. ArchiMate® 3.2 Specification. Retrieved January 22, 2024, from <https://pubs.opengroup.org/architecture/archimate32-doc/>
- *Unified Modeling Language*. (2024, January 20). Wikipedia: the free encyclopedia. Retrieved January 20, 2024, from [https://cs.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://cs.wikipedia.org/wiki/Unified_Modeling_Language)
- Wiegers, K. (2008). *Požadavky na software* (1st ed.). CPress.



## 10 Seznam obrázků

Obrázek 1 Proces business analýzy .....	3
Obrázek 2 ArchiMate – Core Framework .....	12
Obrázek 3 ArchiMate – Hlavní elementy motivační vrstvy .....	13
Obrázek 4 ArchiMate – Hlavní elementy business vrstvy .....	14
Obrázek 5 ArchiMate – Hlavní elementy aplikační vrstvy .....	14
Obrázek 6 ArchiMate – Hlavní elementy technologické vrstvy .....	15
Obrázek 7 ArchiMate – Příklad prvků aktivního strukturálního aspektu .....	16
Obrázek 8 ArchiMate – Příklad prvků behaviorálního aspektu .....	16
Obrázek 10 ArchiMate – Příklad prvků pasivního strukturálního aspektu .....	17
Obrázek 11 ArchiMate – Top level metamodel .....	18
Obrázek 12 ArchiMate – Generický metamodel .....	19
Obrázek 13 UML – Typy UML diagramů .....	21
Obrázek 14 UML – Konceptuální model .....	22
Obrázek 15 UML – Logický model .....	22
Obrázek 16 UML – Fyzický model .....	23
Obrázek 17 UML – Diagram nasazení .....	24
Obrázek 18 UML – Komponentní diagram .....	24
Obrázek 19 UML – Diagram aktivit .....	25
Obrázek 20 UML – Diagram případů užití .....	26
Obrázek 21 IT Projekt – Základní organizační schéma podniku .....	30
Obrázek 22 IT Projekt – Motivační vrstva .....	31
Obrázek 23 IT Projekt – Business potřeby .....	32
Obrázek 24 IT Projekt – Základní proces .....	33
Obrázek 25 IT Projekt – Ukázka procesu svolání personálu .....	35
Obrázek 26 IT Projekt – Business role .....	35
Obrázek 27 IT Projekt – Výčet business objektů .....	36
Obrázek 28 IT Projekt – Aplikační služby .....	43
Obrázek 29 IT Projekt – Aplikační komponenty .....	44
Obrázek 30 IT Projekt – Konceptuální návrh architektury .....	45
Obrázek 31 IT Projekt – Přehled případů užití .....	46
Obrázek 32 IT Projekt – Vazba požadavků na případ užití .....	47
Obrázek 33 IT Projekt – Konceptuální datový model – ukázka .....	48
Obrázek 34 IT Projekt – Logický datový model – ukázka .....	49
Obrázek 35 IT Projekt – Založení distribučního listu .....	51
Obrázek 36 IT Projekt – Návrh přihlašovací obrazovky .....	53
Obrázek 37 IT Projekt – Příklad znázornění detailu komponenty .....	54

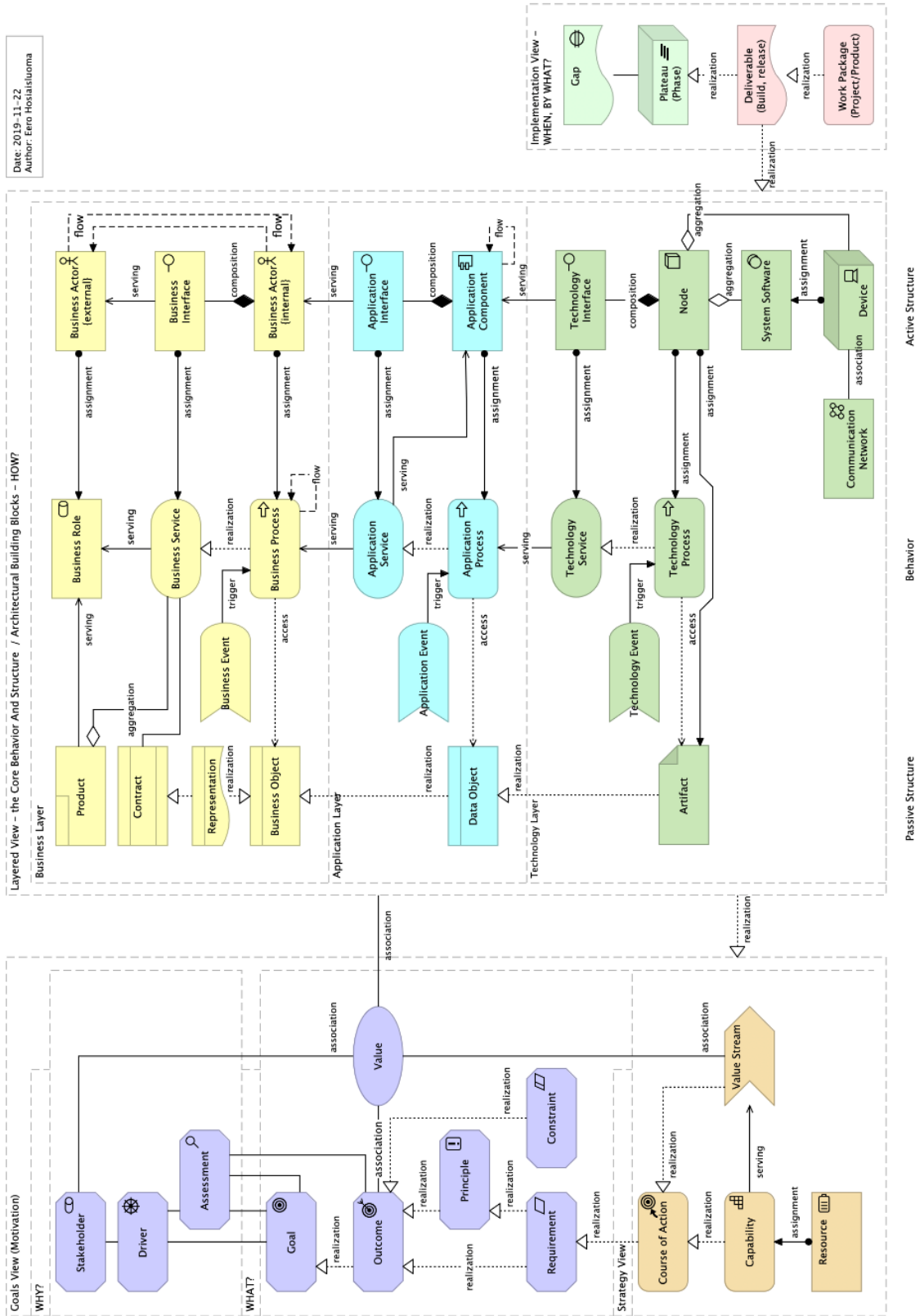
# 11 Seznam tabulek

<b>Tabulka 1 IT Projekt – Business potřeby .....</b>	<b>33</b>
<b>Tabulka 2 IT Projekt – Funkční požadavky .....</b>	<b>39</b>
<b>Tabulka 3 IT Projekt – Nefunkční požadavky .....</b>	<b>41</b>
<b>Tabulka 4 IT Projekt – Architektonické požadavky .....</b>	<b>42</b>

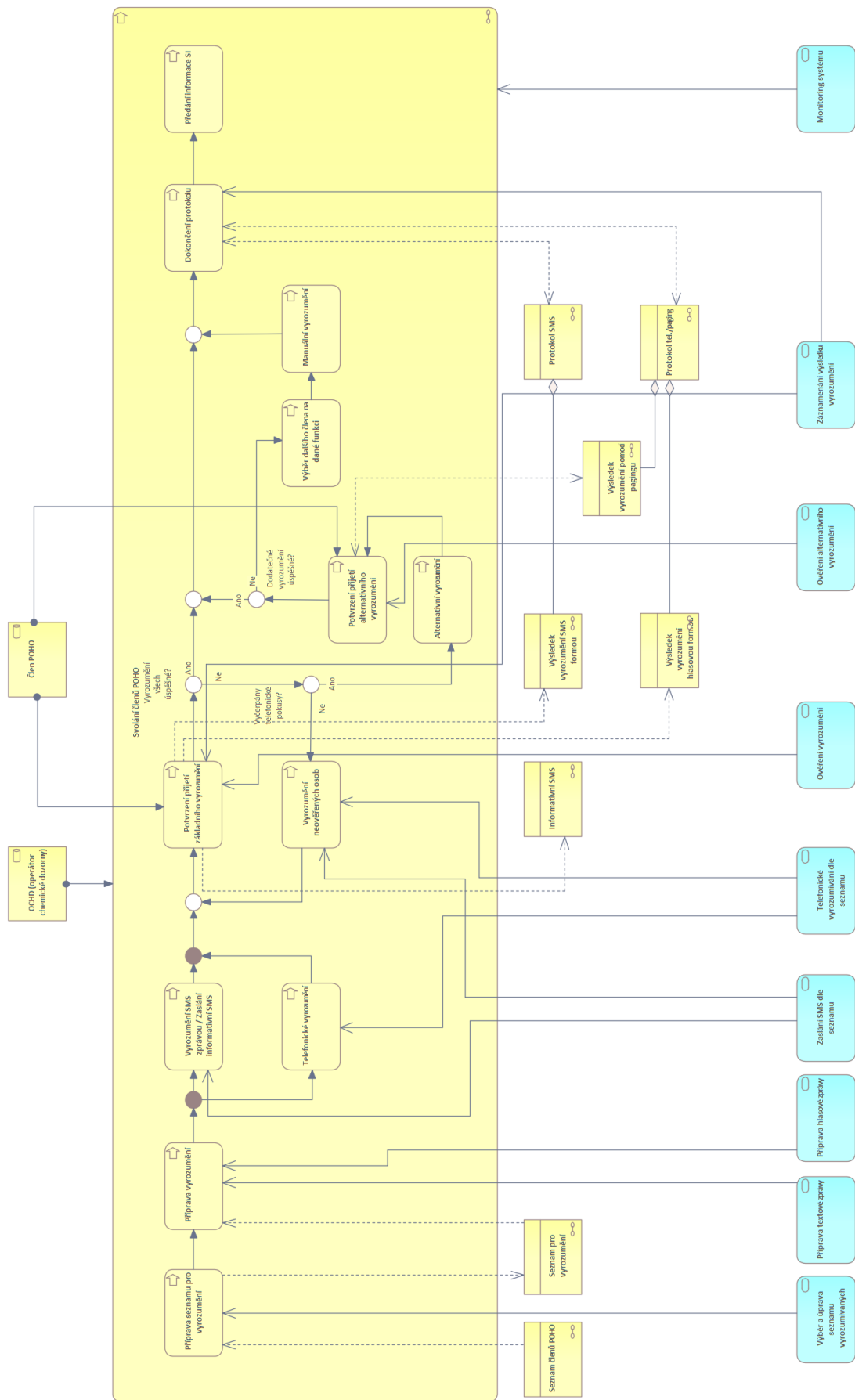
## 12 Seznam příloh

<b>Příloha 1 Kompletní metamodel.....</b>	<b>61</b>
<b>Příloha 2 Kompletní proces .....</b>	<b>62</b>

# 13 Přílohy



Příloha 1 Kompletní metamodel



**Příloha 2 Kompletní proces**