



Zadání bakalářské práce

Název:	Využití ontologické analýzy pro zajištění sémantické interoperability heterogenních dat
Student:	Ondřej Hampejs
Vedoucí:	doc. Ing. Robert Pergl, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Téma přispívá k projektu Datového inkubátoru dat. Cílem práce je ontologická analýza klíčových domén a jejich propojení s datovými sadami tak, aby byla umožněna jejich sémantická interoperabilita.

1. Seznamte se s projektem Datového inkubátoru, problematikou sémantické interoperability, Unified Foundational Ontology, jazykem OntoUML a nástrojem OpenPonk.
2. Ve spolupráci s vedoucím vyberte několik klíčových domén a souvisejících datových sad.
3. Vytvořte ontologické konceptuální modely těchto domén.
4. Propojte ontologické konceptuální modely s datovými sadami a vytvořte pravidla pro jejich mapování.
5. Zdokumentujte své řešení a přínos pro zajištění sémantické interoperability.

Bakalářská práce

**VYUŽITÍ ONTOLOGICKÉ
ANALÝZY PRO
ZAJIŠTĚNÍ SÉMANTICKÉ
INTEROPERABILITY
HETEROGENNÍCH DAT**

Ondřej Hampejs

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: doc. Ing. Robert Pergl, Ph.D.
5. května 2023

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2023 Ondřej Hampejs. Všechna práva vyhrazena.

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

Odkaz na tuto práci: Hampejs Ondřej. *Využití ontologické analýzy pro zajištění sémantické interoperability heterogenních dat.* Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Obsah

Poděkování	viii
Prohlášení	ix
Abstrakt	x
Seznam zkratk	xi
Úvod	1
1 Cíl práce a metodika	3
I TEORETICKÁ ČÁST	5
2 Konceptuální modelování	7
2.1 Kvalita konceptuálního modelování	7
2.2 Význam v oboru informačních technologií	8
3 Unified Modeling Language	9
3.1 Skupiny UML diagramů	9
3.1.1 Strukturální diagramy	9
3.1.2 Diagramy chování	11
3.2 Diagram tříd	11
3.3 UML profily	11
4 Ontologie	13
4.1 Dělení ontologie	13
4.1.1 Deskriptivní ontologie	13
4.1.2 Formální ontologie	13
4.1.3 Formalizovaná ontologie	14
4.2 Význam v oboru informačních technologií	14
4.2.1 Databáze a informační systémy	14
4.2.2 Softwarové (doménové) inženýrství	14
4.2.3 Umělá inteligence	14
5 Unified Foundational Ontology	17
5.1 Kategorizace UFO	17
5.1.1 UFO-A	17
5.1.2 UFO-B	18
5.1.3 UFO-C	18

6	OntoUML	19
6.1	Rozdělení univerzálních typů	19
6.2	Princip identity	19
6.2.1	Sortální a Non-Sortální typy	21
6.3	Rigidita	21
6.3.1	Rigidní, Anti-Rigidní a Semi-Rigidní typy	21
6.4	Jednotlivé univerzální typy	21
6.4.1	Kind	21
6.4.2	Subkind	21
6.4.3	Phase	22
6.4.4	Role	22
6.4.5	Category	22
6.4.6	Mixin	22
6.4.7	RoleMixin	23
6.4.8	PhaseMixin	23
6.5	Generalizace	23
6.6	Asociace	23
6.6.1	Formální relace	23
6.6.2	Materiální relace	24
6.6.3	Relator	24
6.6.4	Mediační relace	24
6.6.5	Charakterizační relace	24
6.7	Vztah celek-část	24
6.7.1	Povinnost	24
6.7.2	Sdílitelnost	25
6.7.3	Quantity	25
6.7.4	Collective	25
6.7.5	Funkční celek	26
6.8	Aspekty	26
6.8.1	Quality	26
6.8.2	Mode	26
7	Iniciativa FAIR	27
7.1	F – Findable	27
7.2	A – Accessible	28
7.3	I – Interoperable	28
7.4	R – Reusable	28
7.5	Nejasnosti prvotních definic	28
7.6	Úprava dat v praxi (FAIRifikační proces)	29
8	Sémantická interoperabilita	31
8.1	Sémantický web	31
9	Projekt OpeNest	33
9.1	Big Data Arena	33
9.2	Inkubátor	34
9.3	Remmark, a.s.	34
10	Platforma OpenPonk	35

II PRAKTICKÁ ČÁST	37
11 Analýza vybraných datových sad	39
12 Tvorba konceptuálních modelů	41
12.1 Šablony	41
12.2 Příklad tvorby konceptuálního modelu	42
13 Mapování dat na entity	45
13.1 Druhy mapovacích pravidel	45
13.2 MEA ontologie	46
13.3 Příklady mapování dat	46
14 „Blitzkrieg“ modelování	49
14.1 Rozdíly oproti běžnému postupu	49
14.2 Zpřesňování konceptuálních modelů	49
14.3 Příklady zpřesňování konceptuálních modelů	50
15 Zajištění sémantické interoperability	51
Závěr	53
A Příklady sémantické interoperability	55
Obsah přiloženého média	65

Seznam obrázků

2.1	Ullmannův trojúhelník podle [2]	7
2.2	Rámec z roku 1994 podle [4]	8
3.1	Hierarchické uspořádání diagramů v UML 2.5 podle [8]	10
4.1	Rozdělení ontologie dle úrovně konkrétnosti podle [19]	15
6.1	Dělení univerzálních typů dle meta-vlastností v OntoUML podle [28]	20
12.1	Propojení entit modelu <i>Seznam bank a poboček zahraničních bank na území ČR</i> reprezentující údaje týkající se adresy banky či její pobočky	43
12.2	Propojení entit modelu <i>Seznam bank a poboček zahraničních bank na území ČR</i> obsahující informace o bance	44
12.3	Vzájemné propojení skupin modelu <i>Seznam bank a poboček zahraničních bank na území ČR</i>	44
13.1	Data entita názvu banky z modelu <i>Seznam bank a poboček zahraničních bank na území ČR</i>	46
13.2	Data entita lokality finančního orgánu z modelu <i>Statistika dražeb dle krajů</i>	47
13.3	Data entita počtu komerčních letišť z modelu <i>Počet komerčních letišť</i>	48
14.1	Počet komerčních letišť z modelu <i>Počet komerčních letišť</i> zpracovávány v „Blitzkrieg“ režimu	50
14.2	Zpřesněný počet komerčních letišť z modelu <i>Počet komerčních letišť</i>	50
A.1	Entita „Adresa“ v modelu <i>Registr poskytovatelů zdravotních služeb</i>	55
A.2	Entita „Adresa“ v modelu <i>Seznam bank a poboček zahraničních bank na území ČR</i>	56
A.3	Entity „Obyvatel“ a „Osoba“ v modelu <i>Počet důchodců s exekucí srážkou podle krajů</i>	56
A.4	Entity „Obyvatel“ a „Osoba“ v modelu <i>Pohyb obyvatel za ČR, kraje, okresy, so, orp a obce – rok 2019</i>	56
A.5	Entita „Motorové vozidlo“ v modelu <i>Statistika dražeb dle krajů</i>	57
A.6	Entita „Motorové vozidlo“ v modelu <i>EU výdaje domácností na konečnou spotřebu podle účelu spotřeby</i>	57
A.7	Entita „Evropská unie“ v modelu <i>Počet komerčních letišť</i>	57
A.8	Entita „Evropská unie“ v modelu <i>EU osoby, které nahlásily nehodu vedoucí k úrazu</i>	58

Seznam tabulek

12.1	Struktura datové sady <i>Seznam bank a poboček zahraničních bank na území ČR</i>	42
------	--	----

Seznam výpisů kódu

13.1	Struktura základního mapovacího pravidla	45
13.2	Struktura pravidla podmíněného hodnotou	46
13.3	Struktura podmíněné pravidla	46
13.4	Struktura pravidel s prvky MEA ontologie	46
13.5	Pravidlo mapující hodnoty sloupce „nazev“ datové sady <i>Seznam bank a poboček zahraničních bank na území ČR</i>	47
13.6	Pravidlo mapující hodnoty sloupce „geograficka lokalita organu financni spravy“ datové sady <i>Statistika dražeb dle krajů</i>	47
13.7	Pravidlo mapující hodnoty sloupců „tra infr“ a „value“ datové sady <i>Počet komerčních letišť</i>	47

Chtěl bych poděkovat především doc. Ing. Robertu Perglovi, Ph.D. za vedení této bakalářské práce, poskytování velice užitečných rad a za pravidelné konzultace. Dále bych chtěl poděkovat Ing. Tereze Macháčové a Bc. Janě Martínkové za pravidelné konzultace k praktické části a za vedení modelovacího týmu. V poslední řadě bych chtěl poděkovat přítelkyni a rodině za psychickou podporu a pevné nervy.

Prohlášení

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

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

V Praze dne 5. května 2023

.....

Abstrakt

Tato bakalářská práce se zabývá využitím ontologické analýzy pro zajištění sémantické interoperability heterogenních dat.

Práce spočívala v ontologické analýze klíčových domén a vybraných datových sad, ve vytvoření konceptuálních modelů těchto domén v modelovacím jazyce OntoUML a v následném namapování samotných dat na takto vzniklé modely. Během tvorby byl kladen veliký důraz na zajištění sémantické interoperability nově vzniklých, ale i již existujících modelů.

U modelů bylo kromě sémantické interoperability dbáno na přesnost a rozšiřitelnost, což umožnilo jejich následné využití nejen v projektu OpeNest, ale i jako podklad analýzy datové sady, či jako dokumentace k samotným datům.

K tvorbě konceptuálních modelů a mapování dat byla využita platforma OpenPonk. Pomocí ní bylo vytvořeno šest konceptuálních modelů včetně datového mapování a jedna šablona. Zmíněné modely a šablona jsou již dnes součástí datové platformy, která je podstatnou částí projektu OpeNest.

Klíčová slova konceptuální model, UML, diagram tříd, ontologie, UFO, OntoUML, FAIR, sémantická interoperabilita, ontologická analýza, heterogenní data, OpeNest, OpenPonk

Abstract

This bachelor's thesis deals with the use of ontological analysis to ensure the semantic interoperability of heterogeneous data.

The work consisted in the ontological analysis of key domains and selected data sets, in the creation of conceptual models of these domains in the OntoUML modelling language, and in the subsequent mapping of the data itself to the models created in this way. During the creation, great emphasis was placed on ensuring the semantic interoperability of newly created, as well as existing models.

In addition to semantic interoperability, accuracy and extensibility were considered in the models, which enabled their subsequent use not only in the OpeNest project, but also as a basis for data set analysis or as documentation for the data itself.

The OpenPonk platform was used to create conceptual models and data mapping. Using it, six conceptual models including data mapping and one template were created. The mentioned models and the template are already part of the data platform, which is an essential part of the OpeNest project.

Keywords conceptual model, UML, class diagram, ontology, UFO, OntoUML, FAIR, semantic interoperability, ontological analysis, heterogeneous data, OpeNest, OpenPonk

Seznam zkratek

BORM	ORD	Business Objects Relation Modeling Object-Relation Diagrams
BPMN		Business Process Model and Notation
DOLCE		Descriptive Ontology for Linguistic and Cognitive Engineering
FAIR		Findable, Accessible, Interoperable, Reusable
FSM		Finite State Machine
GFO		General Formalized Ontology
GOL		General Ontology Language
UFO		Unified Foundational Ontology
MOF		Meta-Object Facility
SysML		Systems Modeling Language
UML		Unified Modeling Language

Úvod

S neustále rostoucím a rozvíjejícím se obsahem na internetu je na něj postupem času přenášeno čím dál tím více dat. Data pochází z různých zdrojů a jsou vedena nad nespočetným množstvím domén. Není tak divu, že jednotlivé datové sady přetékají pestrostí, co se pojmenování a značení týká. Z toho však plyne zásadní problém. Jak tato data propojit, aby nedocházelo k duplicitám a k záměně termínů, které mohou v různých odvětvích znamenat různé věci? Řešením je sémantická interoperabilita dat, o které tato práce pojednává.

Bakalářská práce je rozdělena na dvě hlavní části. První je část teoretická. Jejím úkolem je objasnit, co vše se skrývá pod pojmy konceptuální modelování a UML, čím se zabývá filozofická disciplína nesoucí název ontologie a jakou má spojitost se zkratkou UFO. Dále vysvětluje, jak lze za pomoci ontologie rozšířit základní UML v OntoUML a co obnáší problematika FAIR a samotné sémantické interoperability. Závěr této části obsahuje shrnutí projektu OpeNest a platformy OpenPonk. Druhá část je část praktická. V té jsou nejprve zanalyzovány jednotlivé datové sady. Následně je představen postup tvorby ontologických konceptuálních modelů a následné mapování dat na jejich entity. Dále je v této části poukázáno na rozdíly oproti tzv. „Blitzkrieg“ modelování, jaké jsou důsledky tohoto postupu a proč byl zaveden. V neposlední řadě praktická část obsahuje rozbor zajištění již zmiňované sémantické interoperability nad všemi přidělenými datovými sadami.

Výsledek této práce je součástí projektu OpeNest, který zastrešuje společnost Remmark, a.s. Tento projekt si mimo jiné dává za úkol sdružit jak volně dostupné datové sady od různých statistických úřadů a jiných institucí, tak i datové sady soukromé, které jsou poskytovány externími firmami, či jednotlivci. Všechny takto propojené sady budou poté přístupné v rámci datového inkubátoru přes datovou platformu.

Hlavním důvodem, proč jsem si toto téma zvolil, je možnost přispět svými znalostmi a myšlenkovými pochody do takto ambiciózního projektu, který se zaobírá, dle mého názoru, velice palčivou problematikou současné doby.

Cíl práce a metodika

Hlavním cílem této bakalářské práce je provedení ontologické analýzy klíčových domén a jejich následné propojení s datovými sadami tak, aby byla zajištěna jejich sémantická interoperabilita.

Pro dosažení tohoto cíle musí být jednotlivé sady nejprve zanalyzovány. Následně musí být co nejpřesněji zachyceny pomocí vhodného modelovacího jazyka. A v závěru musí být vytvořena mapovací pravidla, která propojují vzniklé konceptuální modely s daty vybraných datových sad.

Cílem teoretické části této práce je tak definovat využívané pojmy a představit problematiku tuto práci doprovázející. Cílem praktické části je pak již zmiňovaná ontologická analýza klíčových domén a datových sad, tvorba konceptuálních modelů těchto domén, propojení vzniklých modelů a jejich datových sad pomocí mapování a zhodnocení dosažení sémantické interoperability.

Část I

TEORETICKÁ ČÁST

Konceptuální modelování

Tato úvodní kapitola vysvětluje, co si lze představit pod pojmem konceptuální modelování a jaký má význam ve spojitosti s oborem informačních technologií.

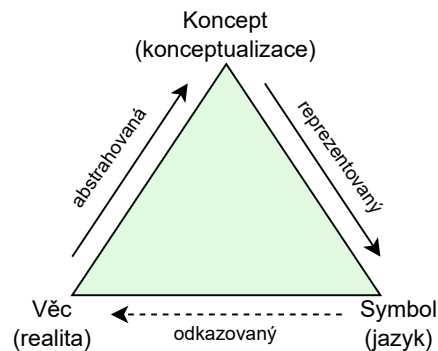
„Konceptuální modelování je činnost formálně popisující některé aspekty fyzického a sociálního světa kolem nás za účelem porozumění a komunikace.“ [1]

Tyto popisy lze označit jako konceptuální schémata, která zachycují relevantní aspekty dané domény a činností, které se v ní odehrávají. Hlavním účelem těchto schémat je sloužit jakožto společný jazyk pro skupinu, která potřebuje zajistit společné a jednoznačné porozumění všech jejích členů. Krom toho umožňuje zachytit sémantiku dané aplikace tak, aby byla dobře pochopitelná pro lidi. Srozumitelnost pro stroje je sekundární. [1]

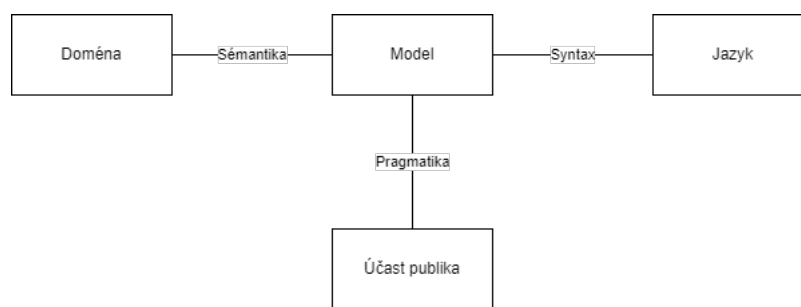
Obrázek Ullmannova trojúhelníku (viz 2.1) znázorňuje vztah mezi abstrahovanou částí reality, konceptualizací a jazykem samotným. Fakt, že je vztah mezi jazykem a realitou vždy zprostředkován určitou konceptualizací, znázorňuje přerušovaná šipka směřující od symbolu k věci, na kterou má symbol odkazovat. [3]

2.1 Kvalita konceptuálního modelování

Pánové Lindland, Sindre a Solvberg přišli v roce 1994 se zajímavým pohledem na kvalitu konceptuálního modelování, který úzce souvisí s lingvistickými koncepty. Hlavním cílem jejich rámce bylo jednoznačně rozlišit cíle a prostředky oddělením toho, čeho se snažíme dosáhnout, a způsobu, jakým se toho snažíme dosáhnout. Obrázek 2.2 ukazuje hlavní body jejich rámce. Rámec



■ **Obrázek 2.1** Ullmannův trojúhelník podle [2]



■ **Obrázek 2.2** Rámec z roku 1994 podle [4]

se inspiruje třemi důležitými lingvistickými koncepty, které následně aplikuje na čtyři aspekty modelování: jazyk, doménu, model a účast publika. [4]

Podle autorů má syntaxe za úkol propojit model a jazyk a popsat vztahy mezi jeho konstrukty bez ohledu na jejich význam. Sémantika poté vztahuje model k dané doméně a přidává tak význam k vztahům mezi konstrukty. Pragmatika nakonec propojuje model a účast publika, přičemž bere v potaz jeho interpretaci. [4]

2.2 Význam v oboru informačních technologií

Úkolem konceptuálního modelování v oblasti informačních technologií je reprezentovat potřebné jevy v dané doméně. Je velmi důležité, aby byl na tuto činnost kladen dostatečný důraz, protože dobré konceptuální schéma může pomoci s včasnou detekcí chyb a jejich následnou opravou. [5] Z tohoto důvodu v dnešní době hraje konceptuální modelování čím dál tím důležitější roli při vývoji informačních systémů a řadí se tak mezi klíčové činnosti spjaté s jejich tvorbou. [6]

Dle studie z roku 2009 byly pro účastníky nejvíce relevantní tyto účely:

1. návrh a správa databází
2. vývoj softwaru
3. zlepšení interních obchodních procesů
4. řízení pracovního toku
5. dokumentace obchodních procesů [6]

První tři účely dosáhly průměrného skóre 4,3 z 5. Řízení pracovního toku a dokumentace obchodních procesů dosáhly výsledku 4,2 z 5. Za zmínku také stojí zlepšení obchodních procesů založených na spolupráci a návrh podnikové architektury, které také přesáhly laťku 4 bodů. [6]

Unified Modeling Language

Tato kapitola pojednává o tom, co obnáší pojem Unified Modeling Language, jaké všechny typy diagramů obsahuje a popisuje základní části diagramu tříd.

Unified Modeling Language (zkráceně UML) je standardní modelovací jazyk sloužící nejen k popisu, specifikaci, návrhu a dokumentaci obchodních procesů, ale i jako podpora při analýze, návrhu a implementaci softwaru. UML je procesně nezávislý a umožňuje tak aplikaci na různé aplikační domény s různými kontexty. [7]

Návrh první verze toho jazyka (UML 1.1) byl vytvořen pány Gradyem Boochem, Ivarem Jacobsonem a Jimem Rumbaughem, načež byl následně v roce 1997 adoptován Object Management Group. Jazyk je pod záštitou tohoto konsorcia v neustálém vývoji a jeho nejnovější verze (UML 2.5) byla vydána v roce 2015. [7]

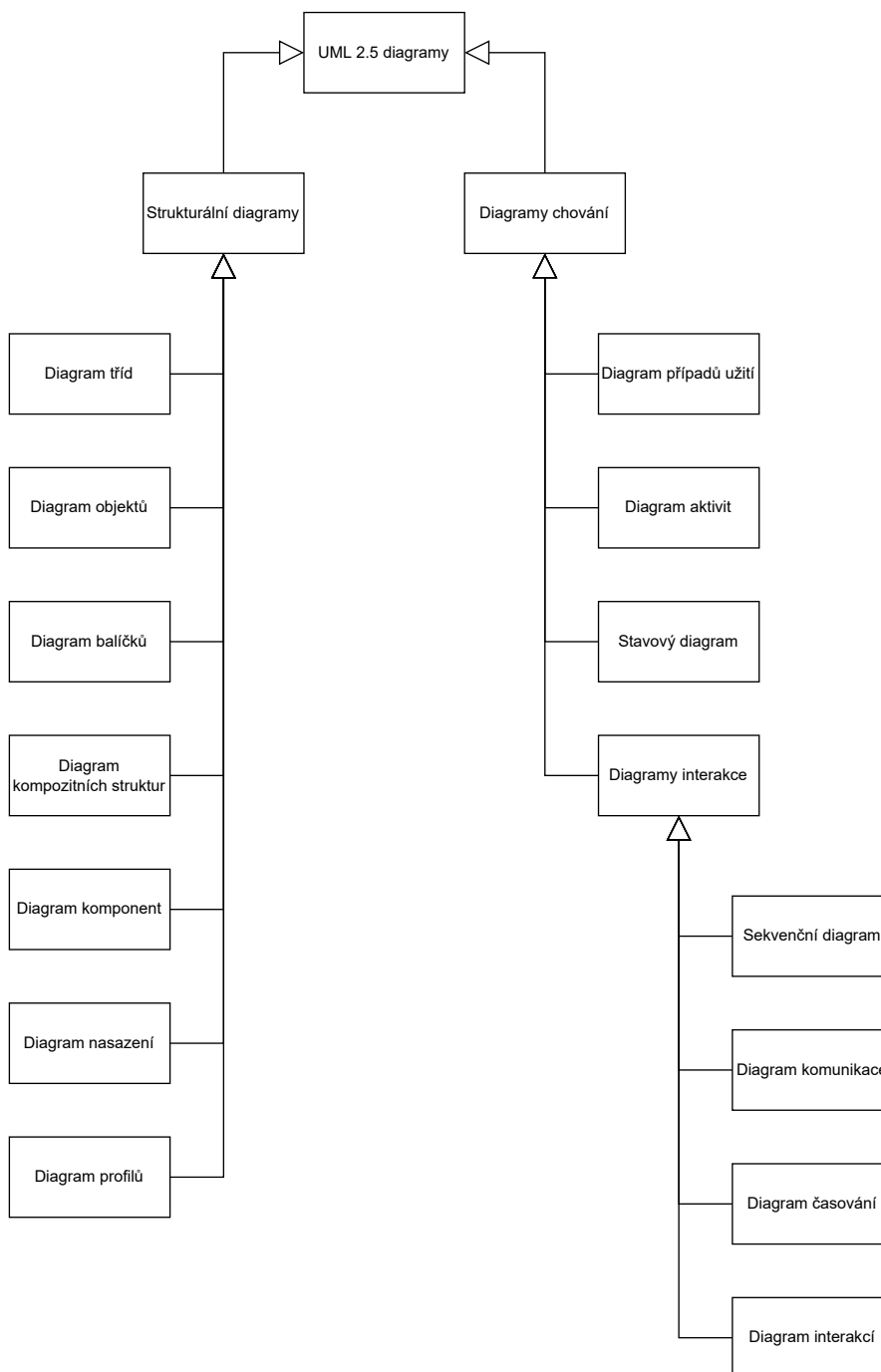
Základem všech UML digramů jsou takzvané *uzly a hrany*, které dohromady vytváří reprezentaci navrhovaného systému. Vzhled těchto prvků definuje druh používaného diagramu (například diagram obsahující aktéry a případy užití se nazývá *diagram případů užití*). Hranice mezi jednotlivými druhy UML diagramů nejsou nikterak vynucovány a je tak možné kombinovat různé druhy pro přesnější zachycení daného systému. [8]

3.1 Skupiny UML diagramů

Specifikace UML rozděluje diagramy do dvou základních skupin: strukturální diagramy a diagramy chování. Obrázek 3.1 znázorňuje hierarchické uspořádání všech druhů diagramů, které jsou součástí oficiální taxonomie UML 2.5. [8]

3.1.1 Strukturální diagramy

Strukturální diagramy znázorňují strukturu daného systému a jeho částí, přičemž pracují na různých úrovních abstrakce. Diagramy mohou také vyobrazovat chování klasifikátorů, avšak nesmí využívat koncepty související s časem. [8] Mezi zástupce skupiny strukturálních diagramů patří například diagram tříd, o kterém bude něco málo řečeno v následující podkapitole 3.2, či diagram komponent (viz levá polovina obrázku 3.1).



■ **Obrázek 3.1** Hierarchické uspořádání diagramů v UML 2.5 podle [8]

3.1.2 Diagramy chování

Diagramy chování znázorňují chování objektů daného systému jako sérii změn v průběhu času. [8] Jako zástupce této skupiny UML diagramů lze vybrat například diagram případů užití, nebo sekvenční diagram (viz pravá polovina obrázku 3.1).

3.2 Diagram tříd

„Diagram tříd je strukturní diagram UML, který zobrazuje strukturu navrženého systému na úrovni tříd a rozhraní, ukazuje jejich vlastnosti, omezení a vztahy...“ [9]

Tento druh diagramu se řadí mezi statické, protože struktura v něm popsaná platí během celého životního cyklu. [10]

Základní prvky digramu tříd jsou:

1. třída – skupina věcí, které mají stejný stav a chování
2. objekt – instance třídy
3. atribut – detail třídy (barva, počet, ...)
4. operace – funkce třídy, která vyvolává chování (součet hodnot, ...)
5. metoda – implementace operace
6. abstraktní třída – třída obsahující operace bez jejich implementací, či třída prázdná
7. vztah – spojení mezi třídami
8. rozhraní – třída obsahující vlastnosti a operace, ale neobsahuje metody
9. šablona – specifikace objektů [11]

3.3 UML profily

Profily jazyka UML představují sadu rozšiřujících mechanismů sloužící k definování nového modelovacího jazyka bez potřeby vytváření zcela nového jazyka pomocí standardu Meta-Object Facility (zkráceně MOF). Tato sada mechanismů umožňuje omezovat počet prvků standardního UML, vázat k nim omezení, nebo je obohacovat o takzvaný syntaktický cukr. [12]

UML profily byly přidány do UML verze 2.0 jakožto balíček *Profiles* a poskytují podporu pro specifikaci modelu MOF sloužící ke konkretizaci konceptů vzhledem k doméně, či implementační technologii. [12] Příkladem těchto profilů může být Systems Modeling Language (zkráceně SysML), nebo OntoUML, kterému se bude věnovat samostatná kapitola (viz 6).

Kapitola 4

Ontologie

Tato kapitola objasňuje, čím se zabývá ontologie jakožto filozofická disciplína a jak ji lze interpretovat v souvislosti s oborem informačních technologií.

Ontologie je filozofická disciplína zabývající se určením, vlastnostmi a projevy bytí. Samotný pojem ontologie je odvozen od řeckých slov *ontos* a *logos*, tedy „jsoucí výklad“ a první zmínka o tomto pojmu pochází již ze 17. století. Podobnou otázkou se zabýval také Aristoteles ve svém textu *Metafyzika* [13], který rozvíjí jeho pojetí *první filozofie*. [14]

„*Ontologie se snaží poskytnout definitivní a vyčerpávající klasifikaci entit ve všech sférách bytí.*“ Definitivnost nám umožňuje nalézt odpovědi na otázky stylu: „Jaké třídy entit jsou potřeba...“, zatímco vyčerpateľnosť říká, že by klasifikace měla obsahovat všechny entity a jejich vzájemné vztahy, aby dohromady vytvořily co největší celek. [15]

4.1 Dělení ontologie

Ontologii lze dělit do tří kategorií: deskriptivní, formální a formalizovaná. Každá kategorie se dále může dělit na doménově závislou, nebo naopak doménově nezávislou. V prvním případě se jedná o ontologii nad uzavřenou oblastí bytí, zatímco v případě druhém se jedná o obecnou ontologii. [16]

4.1.1 Deskriptivní ontologie

Deskriptivní ontologie se zabývá shromažďováním informací o daných objektech. Objekty mají vztah k jejich doméně a mohou se tak rovněž dělit na závislé (reálné), či nezávislé (ideální). [17]

4.1.2 Formální ontologie

Úkolem formální ontologie je upravovat, filtrovat a organizovat výsledky, kterých bylo dosaženo během deskriptivní ontologie. [16] Tato kategorie byla poprvé představena na začátku 20. století panem Edmundem Husserlem, který se ve svém výzkumu zabýval problémem pravdy na třech základních úrovních. [17]

Pan Nino Cocchiarella formální ontologii shrnul slovy: „*Formální ontologie je disciplína, ve které se formální metody matematické logiky kombinují s intuitivními, filozofickými analýzami a principy ontologie. Formální ontologie tak spojuje jasnost, přesnost a metodologii logické analýzy na jedné straně s filozofickým významem ontologické analýzy na straně druhé.*“ [18]

Současné teorie zabývající se touto problematikou častokrát slučují formální ontologii s ontologií deskriptivní, či formalizovanou. Toto sloučení však častokrát zapříčiní opomenutí relevance jedné z kategorií. [16]

4.1.3 Formalizovaná ontologie

Snahou formalizované ontologie je zkonstruovat formální kodifikaci pro výsledky, které nám poskytnou deskriptivní a formální ontologie. [17]

4.2 Význam v oboru informačních technologií

V souvislosti s informačními technologiemi bylo slovo ontologie poprvé zmíněno v roce 1967, avšak první větší nárůst zájmu lze pozorovat až v 90. letech. Motivace pro rozvoj tohoto odvětví byla zapříčiněná potřebou vytvořit reprezentaci doménových znalostí umělé inteligence. K velkému nárůstu výzkumu v této oblasti došlo až na začátku 21. století. [19]

Historicky lze vyzdvihnout tři oblasti, které byly zodpovědné za rozvoj ontologie v oblasti informačních technologií: databáze a informační systémy, softwarové inženýrství a umělá inteligence. Důležitý je rozdíl vnímání ontologie z pohledu těchto oblastí. Zatímco oblast informačních systémů používá pojem ontologie, jakožto systém jazykově nezávislých kategorií, v ostatních oblastech informačních technologií je tento termín konkretizován pro specifický účel a artefakt. [19] Následující tři podkapitoly se věnují vždy jedné z těchto oblastí.

4.2.1 Databáze a informační systémy

Hlavní motivací pro rozvoj v této oblasti byla snaha nalézt lepší koncepty pro reprezentaci určité části reality. Během let vzniklo několik pokusů o ontologické zachycení strukturálních aspektů pomocí tříd a vztahů mezi nimi, avšak žádný z nich neměl kategorie doopravdy založené na této filozofické disciplíně. Tyto nekonzistence vedly k několika problémům, se kterými se dodnes můžeme setkat například při integraci databází. Z tohoto důvodu začalo několik výzkumníků stavět svou práci na ontologii a pomohli tak vzbudit zájem o tuto disciplínu napříč různými komunitami. [19]

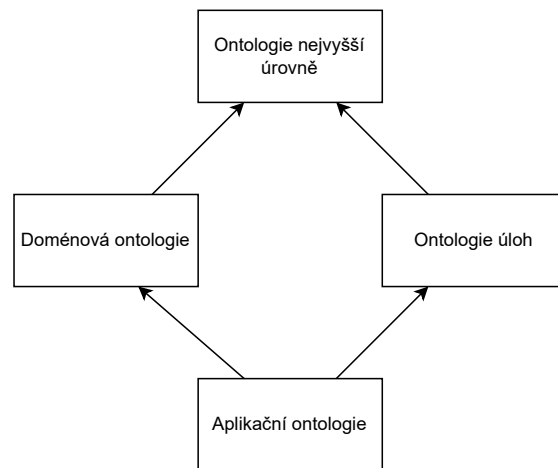
4.2.2 Softwarové (doménové) inženýrství

Oblast softwarového inženýrství byla motivována převážně potřebou snížit náklady na údržbu softwaru a umožnit jeho opětovné využití. Cílem samotného doménového inženýrství je vytvořit rámec, který lze opakovaně používat pro vývoj různých aplikací nad stejnou doménou. Z tohoto důvodu je potřeba přesná konceptualizace všech důležitých entit v dané oblasti. [19]

4.2.3 Umělá inteligence

Do nástupu ontologie v tomto odvětví bylo tradicí definovat veškeré znalosti přísně a funkčně. Cílem rozvoje v této oblasti bylo vytvoření souladu mezi znalostmi a dvěma, na sobě nezávislými, subsystemy. Těmito subsystemy jsou chování umělé inteligence a její prostředí. Tento rozvoj zapříčinil rozdělení ontologie dle úrovně konkrétnosti a úhlu pohledu (viz obrázek 4.1). [19]

Ontologie nejvyšší úrovně je nejvíce obecná a zabývá se pojmy, které nejsou doménově závislé (čas, prostor, hmota, ...). *Doménová ontologie* a *ontologie úloh* jsou již doménově závislé, ale jimi popisované domény jsou generické (bankovníctví, medicína, ...), či zachycují generickou činnost (prodej, poskytování služby, ...). Poslední úrovní je *aplikační ontologie*, která již závisí na konkrétní doméně a činnosti. Všechny tyto úrovně představují konkrétní inženýrské artefakty. [19]



■ **Obrázek 4.1** Rozdělení ontologie dle úrovně konkrétnosti podle [19]

Unified Foundational Ontology

Tato kapitola vysvětluje, co znamená zkratka UFO ve spojitosti s ontologií a rozděluje ji do tří základních kategorií.

„Unified Foundational Ontology (UFO) vzniklo jako iniciativa k rozvoji zdravých ontologických základů pro konceptuální modelovací jazyky.“ [20] První koncepty UFO, jakožto vyšší ontologie, vznikly již v roce 2004 v rámci výzkumu profesora Giancarla Guizzardiho a dalších vědců. Ti se společnými silami snažili sjednotit již existující vyšší ontologie GFO/GOL a DOLCE (vznikající ve stejné době v Německu a Itálii) a vytvořit tak jednu ucelenou ontologii.[21]

UFO je od té doby v neustálém vývoji a snaží se odpovídat na potřeby disciplíny konceptuálního modelování (viz 2). Jako příklad stálého rozšiřování si můžeme uvést připojení OntoClean metodologie. [22] Ta se zabývá analýzou ontologií založenou na formálních, doménově nezávislých vlastnostech tříd. [23]

Základem všech již zmiňovaných ontologií je Aristotelovo kategorické rozdělení do čtyř tříd. [22] Tyto základní kategorie vychází z Aristotelova vnímání a dělení jsoucen. První rozdělení se zabývá rozlišením toho, co vypovídá o nějakém podmětu (*vypovídáno o jiném*), a naopak toho, co o žádném podmětu nevypovídá. V prvním případě se jedná o univerzální typ, v druhém pak o typ individuální. Druhé rozdělení diferencuje to, co daný podmět nezbytně potřebuje ke své existenci (*přítomno v podmětu*), a naopak, co ke své existenci nutně nepotřebuje. [24]

5.1 Kategorizace UFO

Postupným rozvojem se UFO ontologie rozdělila na tři vrstvy zabývající se různými aspekty reality. [21]

5.1.1 UFO-A

UFO-A neboli ontologie endurantů se zabývá aspekty strukturálního konceptuálního modelování a je organizována jako ontologie o čtyřech kategoriích zahrnující teorie typů a taxonomických struktur. [21]

Hlavní rozdíl v této vrstvě můžeme pozorovat mezi kategoriemi individuálního a univerzálního typu. Individuální typy jsou entity, které mají jedinečnou identitu a ve skutečnosti existují. Univerzální typy jsou naopak vzorem vlastností, které lze realizovat několika různými individuálními typy. [25]

5.1.2 UFO-B

UFO-B neboli ontologie perdurantů se zabývá aspekty událostí a procesů, jejich chronologickým uspořádáním, příčinou, ale i změnou vazeb mezi nimi a enduranty. [21]

Základem této vrstvy je rozlišení mezi enduranty a perduranty. Klasicky lze toto rozdělení chápat z hlediska jejich chování v čase. Enduranti, pokud jsou přítomni, musí být vždy přítomni zcela, zatímco perduranti jsou složeni z částí, které se dějí v čase a mohou se prodlužovat, či hromadit. [25]

5.1.3 UFO-C

UFO-C neboli ontologie sociálních entit je postavena na předchozích dvou vrstvách. Kromě samotných sociálních konstruktů se zabývá například přesvědčeními, touhami, záměry, činy a závazky. [21]

Kapitola 6

OntoUML

Tato kapitola se zaměřuje na základní koncepty OntoUML a postupně prochází všechna rozšíření standardního UML, která tento jazyk poskytuje.

OntoUML je konceptuální modelovací jazyk, který vychází z jazyka UML (viz 3) rozšířeného o ontologii UFO (viz 5). [26] Oproti UML nepoužívá výraz „třída“, ale staví na individuálních a univerzálních typech vycházejících přímo z ontologie UFO-A (viz 5.1.1). [27]

První základy tohoto jazyka lze, stejně jako základy UFO ontologie, připisat profesorovi Giancarla Guizzardimu. [26] Ten ve své práci využil výše zmíněné ontologie k vytvoření nového UML profilu (viz 3.3) vycházejícího z *diagramu tříd* jazyka UML. [28] Jeho účelem bylo vytvořit ontologicky čistý a sémanticky jednoznačný modelovací jazyk pro účely konceptuálního modelování. [19] „*OntoUML byl přijat mnoha akademickými, firemními a vládními institucemi po celém světě pro vývoj koncepčních modelů v různých oblastech.*“ [26]

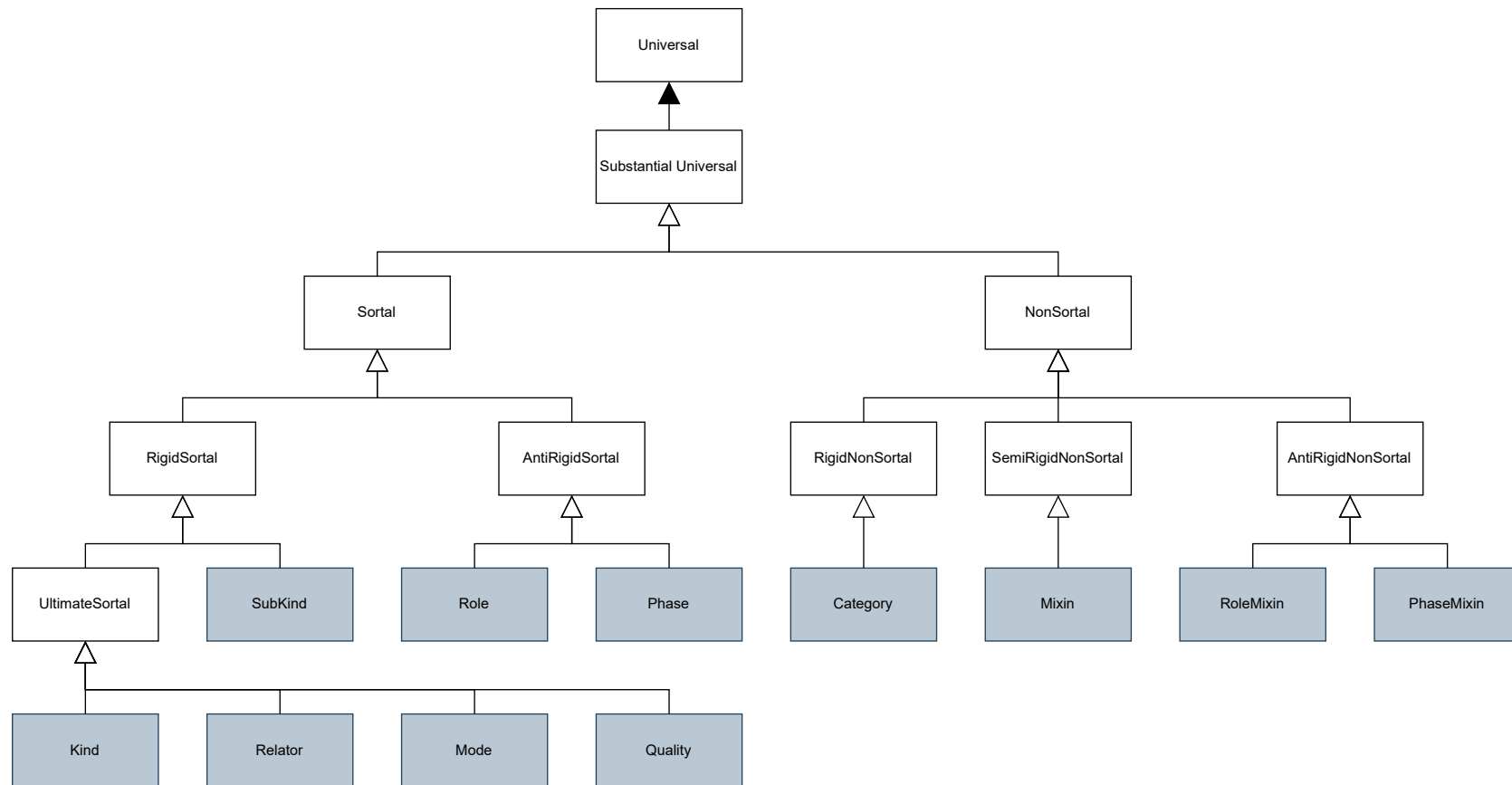
6.1 Rozdělení univerzálních typů

Na obrázku 6.1 lze pozorovat univerzální typy a jejich dělení podle principu identity a rigidity. Následující podkapitoly se budou zabývat tímto rozdělením a postupně vysvětlovat rozdíly mezi jednotlivými typy.

6.2 Princip identity

Princip identity je jeden ze základních konceptů ontologie UFO-A, jehož úkolem je umožnit rozlišení individuálních typů. Různé univerzální typy určují různé způsoby odlišování jejich identity. [27] Typy dále mohou identitu poskytovat (*Kind*, *Collective*, ...), mohou ji dědit od jiných typů (*Subkind*, *Role*, ...), či sdružovat typy řídicí se různými principy identity (*Category*, *Mixin*, ...). [29] Jako příklad si můžeme uvést rozdíl mezi osobou a budovou, které nemají stejnou identitu i přes to, že jsou obě příkladem univerzálního typu *Kind* (viz 6.4).

Různé individuální typy stejného univerzálního typu mají **vždy právě jednu** specifickou identitu. Ta je určena při jejich vzniku a po celou dobu jejich existence je neměnná. [27] Příkladem je rozlišení Karla a Vaška. Oba jsou osobou, ale z principu nemají stejnou identitu.



■ **Obrázek 6.1** Dělení univerzálních typů dle meta-vlastností v OntoUML podle [28]

6.2.1 Sortální a Non-Sortální typy

Univerzální typy lze díky principu identity obecně rozdělit do dvou kategorií. První jsou takzvané Sortální univerzální typy (viz levá část obrázku 6.1), které jsou specifické tím, že poskytují princip identity svým individuálním typům. V druhém případě se jedná o Non-Sortální univerzální typy (viz pravá část obrázku 6.1), které naopak princip identity neposkytují. [27]

6.3 Rigidita

Rigidita je vlastnost univerzálních typů definující jejich neměnnost. [27] Jinými slovy rigidita určuje, zda se individuální typy mohou měnit v průběhu svého života. [19]

6.3.1 Rigidní, Anti-Rigidní a Semi-Rigidní typy

Díky rigiditě lze univerzální typy dělit do tří kategorií. První jsou Rigidní univerzální typy, jejichž individuální typy nemohou přestat být jejich bez toho, aniž by zanikly. [27] Tyto typy totiž definují základní charakteristické vlastnosti a musí tak být vždy tohoto typu po celou dobu své existence. Patří mezi ně například *Kind*, *Subkind* a *Category*. [30]

Druhé jsou Anti-Rigidní univerzální typy, jejichž individuální typy mohou, na rozdíl od Rigidních, přestat být jejich bez toho, aniž by zanikly. Neboli pokud je individuální typ instancí tohoto typu v jednom světě, nemusí být jeho instancí ve světě jiném. [27] Jakožto příklad si můžeme uvést *Role* a *RoleMixin*. [30]

Třetí jsou Semi-Rigidní univerzální typy. Ty mohou zahrnovat jak Rigidní, tak Anti-Rigidní typy a jako jediné existují pouze v Non-Sortální podobě. [27] Pro různé individuální typy se tak mohou chovat buďto jako Rigidní typy, nebo jako Anti-Rigidní typy. Jejich jediným příkladem je *Mixin*. [31]

6.4 Jednotlivé univerzální typy

V této podkapitole budou postupně rozebrány jednotlivé univerzální typy a jejich využití. Z tohoto rozboru budou vynechány typy *Relator*, *Collective*, *Quantity*, *Quality* a *Mode*, které budou následně rozebrány ve vlastních podkapitolách 6.6.3, 6.7 a 6.8.

6.4.1 Kind

Univerzální typ *Kind* se řadí mezi Rigidní Sortální typy. Tvoří páteř modelu a definuje princip identity pro své individuální typy, kterým tak umožňuje jejich jednoznačné rozlišení. [27] *Kind* nevyžaduje žádnou relační závislost a představuje tzv. funkční celek (viz 6.7). [32]

Existují čtyři omezující podmínky pro tento typ vyplývající z jeho vlastností. První a druhá z nich říká, že *Kind* nemůže mít přímí či nepřímí nad-typ, který dědí identitu od jiného, nebo ji sám poskytuje (*Kind*, *Subkind*, ...). Třetí zakazuje přímé a nepřímé podtypy, které agregují individuální typy s různými principy identity (*Category*, *Mixin*, ...). Čtvrtá podmínka nepovoluje přímé a nepřímé Anti-Rigidní nad-typy (*Role*, *RoleMixin*, ...). [32]

6.4.2 Subkind

Subkind je univerzální typ, který se řadí mezi Rigidní Sortální typy. [27] Tento typ je používán k vyjádření Rigidní specializace poskytovatelů identit (*Kind*, *Category*, ...) [33] a na rozdíl od typu *Kind* tak nedefinuje vlastní princip identity, nýbrž ho dědí od svého předka, díky čemuž

ho může poskytovat svým individuálním typům. [27] *Subkind* opět nevyžaduje žádnou relační závislost. [33]

Generalizace (viz 6.5) zprostředkovávající dědění může mít libovolnou kombinaci meta vlastností *isDisjoint* a *isCovering* a tvořit tak generalizační sety a dědičné hierarchie. [27]

Subkind má tři omezující podmínky. První dvě jsou shodné s třetí a čtvrtou podmínkou u typu *Kind* (viz 6.4.1), protože se také jedná o Rigidní typ řídicí se stejným principem identity. Poslední omezující podmínka deklaruje, že *Subkind* musí mít vždy právě jeden nad-typ poskytující identitu. [33]

6.4.3 Phase

Phase je Anti-Rigidní Sortální univerzální typ. [34] Jeho úkolem je reprezentovat podtypy poskytovatelů identit, které vznikají v závislosti na změnách vnitřních vlastností. Individuální typy se musí řídit stejným principem identity a být součástí rozdělení, které je závislé na dané vlastnosti, nebo variaci více vlastností. [35] *Phase* je opět relačně nezávislý univerzální typ. [28]

Phase má hned pět omezujících podmínek, z čehož první zdůrazňuje nutnost příslušnosti k rozdělení. Druhá je pak totožná s třetí omezující podmínkou u *Subkind* a říká, že vždy právě jeden z přímých či nepřímých nad-typů musí být poskytovatelem identity. Třetí podmínka neumožňuje *Phase* být přímým podtypem *RoleMixin* a *Category*. Poslední dvě omezující podmínky zakazují Rigidním a Non-Sortálním typům býti pod-typy tohoto typu. [35]

6.4.4 Role

Role patří mezi Anti-Rigidní Sortální univerzální typy. [34] Podobně jako *Subkind* se tento typ využívá k vyjádření tentokrát Anti-Rigidní specializace poskytovatelů identit o jejichž konkrétnosti se stará tzv. relační kontext. Individuální typy se stejně jako u *Phase* řídí stejným principem identity. [36] *Role* je, oproti předchozím typům, relačně závislý typ, který zachycuje relační vlastnosti sdílené individuálními typy. K zachycení tohoto jevu se využívá *materiální relace* (viz 6.6.2). [28]

Univerzální typ *Role* má čtyři omezující podmínky. První podmínka opět deklaruje, že tento typ musí mít právě jeden nad-typ poskytující identitu. Druhá zdůrazňuje nutnost již zmiňované *materiální relace*. Poslední dvě jsou stejné jako u *Phase* (viz 6.4.3) a omezují možné podtypy. [36]

6.4.5 Category

Univerzální typ *Category* lze zařadit do podskupiny Rigidních Non-Sortálních typů. [34] Jedná se o typ sdružující základní vlastnosti jeho Rigidních Sortálních podtypů [28], které by měly být z podstaty disjunktní. [19] *Category* se opět řadí mezi relačně nezávislé typy. [28]

Tento typ se řídí třemi omezujícími podmínkami. První říká, že *Category* je vždy abstraktní typ, který sdružuje individuální typy s různými principy identity. Druhá a třetí pak omezují možné nad-typy pouze na *Category* a *Mixin*. [37]

6.4.6 Mixin

Mixin, jak již bylo řečeno v podkapitole 6.3.1, je jediným příkladem Semi-Rigidního Non-Sortálního typu. Jeho úkolem je sdružovat vlastnosti, které mohou být pro různé instance buďto podstatné, nebo naopak náhodné. [34] *Mixin* je opět relačně nezávislý. [31]

Mixin má pouze dvě omezující podmínky. První deklaruje, že *Mixin*, stejně jako *Category*, musí být vždy abstraktní. Druhá pak zakazuje jakýkoliv nad-typ, který není *Mixin*, nebo *Category*. [31]

6.4.7 RoleMixin

RoleMixin patří mezi Anti-Rigidní Non-Sortální univerzální typy. Představuje abstrakci pro společné vlastnosti disjunktních typů *Role* a stejně jako tento typ je relačně závislý. [19]

RoleMixin se může vyskytovat ve dvou podobách. V první z nich je definovaný samotnými typy *Role*, zatímco v druhém případě slouží jakožto zástupce *Role* pro typ *Category*. Obě tyto podoby jsou významově ekvivalentní [38] a záleží tak pouze na subjektivní preferenci.

Pro tento typ existují tři omezující podmínky. První z nich opět deklaruje, že *RoleMixin* musí být vždy abstraktní typ. Druhá pak zakazuje Sortální typy jakožto nad-typ a třetí omezuje možné podtypy pouze na *RoleMixin* a *Role* samotný. [38]

6.4.8 PhaseMixin

Univerzální typ *PhaseMixin* se řadí mezi Anti-Rigidní Non-Sortální typy. Jeho úkolem je sdružovat vlastnosti společné pro různé typy *Phase* stejně jako *RoleMixin* pro typ *Role*. [28] Na rozdíl od již zmiňovaného typu *RoleMixin* se však jedná o relačně nezávislý typ. [39] Tento univerzální typ se jako jediný nenachází v původní sadě definované ontologii UFO, avšak různí autoři ho v průběhu let shledali chybějícím. [28]

Tři omezující podmínky definované pro tento typ jsou totožné s podmínkami uvedenými pro *RoleMixin* (viz 6.4.7) s výjimkou poslední, která povoluje podtypy *PhaseMixin*, *Phase*, *RoleMixin* a *Role*. [39]

6.5 Generalizace

Generalizace je druh relace znázorňující vztah mezi nadtřídou (nad-typem) a podtřídou (podtypem). Využívá se v případě, že existuje více podtypů daného typu. Tyto podtypy díky generalizaci dědí všechny vlastnosti jejich nad-typu, přičemž je mohou rozšiřovat i o své vlastní. [27]

Jak již bylo zmíněno v podkapitole 6.4.2, více podtypů jednoho nad-typu mohou dohromady tvořit generalizační sety a dědičné hierarchie. Pro každý takovýto set musí být určeny dvě meta vlastnosti, které omezují vztahy jednotlivých individuálních typů. První z nich je *isDisjoint*, která vyjadřuje, zda jsou jednotlivé podtypy výlučné, nebo zda jich může mít jeden individuální typ více. Druhá vlastnost je *isCovering*. Ta pro změnu určuje, zda musí být individuální typ vždy instancí podtypu, nebo zda může být i instancí samotného nad-typu. [27]

6.6 Asociace

Asociace je druh relace, který má za úkol spojovat jednotlivé entity a vyjadřovat tak jejich vztah. Každá asociace musí mít definovanou násobnost na každém jejím konci. Ta vypovídá o počtu entit v dané relaci. Dále pak může mít definovaný název a případně i směr [19], který usnadňuje následné čtení tohoto vztahu.

Asociace lze obecně rozdělit do dvou kategorií: formální a materiální. [19] Následující dvě podkapitoly se budou těmito dvěma kategoriemi zabírat.

6.6.1 Formální relace

Formální relace může existovat mezi entitami na přímo bez nutnosti zprostředkovatele (viz 6.6.2). [19] Jejím účelem je vyjádření vztahů, které reprezentují porovnání vnitřních vlastností (jejich hodnot) charakterizujících dané entity. [40]

6.6.2 Materiální relace

Materiální relace, na rozdíl od té formální, vyžaduje zprostředkovatele, kterého zastupuje univerzální typ *Relator* (viz 6.6.3). Tato relace má sama o sobě materiální strukturu a považuje se za relaci odvozenou od již zmiňovaného typu *Relator* a *mediační relace* (viz 6.6.4). K vyznačení tohoto faktu se využívá tzv. *derivate*, která propojuje *Relator* a samotnou materiální relaci. [19]

6.6.3 Relator

Relator patří mezi Rigidní Sortální univerzální typy. Úkolem tohoto typu je reprezentovat zprostředkovatele *materiální relace* (viz 6.6.2) [41] a spojovat tak jednotlivé individuální typy. [19] Stejně jako typ *Kind* (viz 6.4.1) definuje vlastní princip identity a poskytuje ho svým individuálním typům. Na rozdíl od toho typu je však typ *Relator* relačně závislý na individuálních typech, které spojuje. [41]

Relator má hned šest omezujících podmínek. První z nich říká, že musí být spojen vždy alespoň s jednou mediační relací (viz 6.6.4). Druhá podmínka rozšiřuje první a stanovuje, že součet násobností na opačných koncích všech mediačních relací musí být přinejmenším roven dvěma [41], protože typ *Relator* musí spojovat vždy alespoň dva různé individuální typy. [42] Další tři podmínky omezují možné nad-typy pouze na *Category* a *Mixin*. Poslední pak zakazuje podtypy s odlišným principem identity a povoluje tak pouze *Subkind*, *Phase* a *Role* jakožto přímý, či nepřímý podtyp. [41]

6.6.4 Mediační relace

Mediační relace je druh formální relace (viz 6.6.1), která propojuje *Relator* (viz 6.6.3) a univerzální typ, který zprostředkovává. [19] Tato relace je existenčně závislá. [42]

6.6.5 Charakterizační relace

Charakterizační relace je druhem relace formální (viz 6.6.1). Tento druh se využívá k propojení univerzálních typů *Quality* a *Mode* (viz 6.8) s univerzálním typem, který tyto aspekty charakterizují. [19]

6.7 Vztah celek-část

Vztah *celek-část*, také označovaný jako vztah *část-celek*, je důležitý pro naše vnímání a řešení kognitivních úkolů. Tyto relace musí být ze své podstaty ireflexivní a antisymetrické, avšak její tranzitivitu nelze obecně určit. Kromě samotné teorie částí, která byla důležitým bodem zájmu již od presokratických filozofů, je pro tuto relaci důležitá i teorie celku, která je mnohými formálními teoriemi opomíjena. Dále je pak také podstatné rozlišení rolí samotných částí, které pomáhají diferencovat povinnost, neoddělitelnost a sdílitelnost částí a celků. [19] Následujících pět podkapitol se bude zabývat rozlišením těchto rolí a samotnými typy agregací.

6.7.1 Povinnost

Povinnost v relaci *celek-část* se obecně dělí na dva druhy podle jejich hlediska. První z nich se zabývá povinností z hlediska celku, druhý pak povinností z hlediska části. Každý z těchto pohledů je sám o sobě ontologicky důležitý a nesmí se opomenout. [43]

Povinnost z hlediska celku rozlišuje tři typy částí. První z nich je *část volitelná*, která není podstatná pro identitu celku a celek na ní není genericky, či existenčně závislý. Druhá z nich

je *část povinná*. Na té je již celek genericky závislý, což znamená, že celek potřebuje její typ, nikoli její instanci. Poslední typ části z hlediska celku je *část esenciální*. Na této části je celek existenčně závislý a vyžaduje tak již určitou instanci, bez které nemůže celek existovat. [43]

Obdobně jako u předchozího hlediska lze povinnost celku vůči části rozlišit na tři typy. První je *nepovinný celek*, na kterém není část nikterak závislá. Druhý je *povinný celek*, na kterém je část genericky závislá stejně, jako tomu bylo v případě *povinné části*. Posledním typem je *neoddělitelná část*, která představuje část existenčně závislou na celku. [43]

Anžto mohou pojmy *esenciální část* a *neoddělitelná část* působit stejně, jsou na sobě naprosto nezávislé. Esencialita, jak již bylo řečeno, představuje existenční závislost celku na části, zatímco neoddělitelnost naopak představuje existenční závislost části na celku. Z toho tedy plyne, že ani jeden z těchto pojmů neimplikuje ten druhý. [43]

Z definice *esenciální* a *neoddělitelné části* plyne, že lze tyto pojmy použít pouze pro Rigidní univerzální typy. Z tohoto důvodu se u Anti-Rigidních typů používá *neměnitelná část* místo *esenciální části* a *neměnitelný celek* namísto *neoddělitelné části*. Tyto pojmy jsou na sobě opět nezávislé. [43]

6.7.2 Sdílitelnost

Sdílitelnost je sekundární vlastnost relace *celek-část* (viz 6.7). [19] I když je tuto vlastnost možné odvodit přímo z multiplicity relace, využívá se k jejímu označení „prázdný diamant“ a „plný diamant“. „Prázdný diamant“ se používá k označení části, která je sdílitelná vícero celky, neboli k označení relace, jejíž multiplicita na straně celku je alespoň jedna. Naopak „plný diamant“ značí části nesdílitelné, jejichž relace má na straně celku multiplicitu rovnou maximálně jedné. [43]

6.7.3 Quantity

Quantity je univerzální typ patřící mezi Rigidní Sortální typy. Obecně typ *Quantity* představuje celek, který vyjadřuje maximálně topologicky spojené množství hmoty, která je nespočetná. Rozdělením této hmoty na dvě části dostaneme dvě nové instance téhož typu. Z tohoto důvodu může mít celek *Quantity* pouze další *Quantity* jako své části. [44] K vyjádření tohoto vztahu se používá relace *SubQuantityOf*, jejíž násobnost na obou koncích musí být rovna jedné a jejíž část musí být *esenciální* (viz 6.7.1). [45] Obdobně jako typ *Kind* (viz 6.4.1) definuje tento univerzální typ princip identity pro své individuální typy a je relačně nezávislý. [44]

Čtyři omezující podmínky povolují pouze *Category* a *Mixin* jakožto nad-typ a *Subkind*, *Role* a *Phase* jakožto podtyp typu *Quantity*. [44]

Kromě relace *SubQuantityOf* se k tomuto univerzálnímu typu vztahuje také relace *Containment*, která se používá k vyjádření vztahu mezi nádobou a jejím obsahem. Násobnost na obou stranách této relace musí být rovna jedné. [46]

6.7.4 Collective

Univerzální typ *Collective* je opět Rigidní Sortální typ, který je relačně nezávislý a poskytuje identitu svým individuálním typům. Tento typ reprezentuje celek s homogenní vnitřní strukturou. Části mají libovolný univerzální typ, ale musí celku přispívat stejným způsobem. Tento vztah vyjadřuje relace *MemberOf* [47], která vyžaduje násobnost na straně části rovnou alespoň dvěma. Část v této relaci je opět *esenciální* (viz 6.7.1). [48]

Collective má stejně jako typ *Quantity* (viz 6.7.3) čtyři omezující podmínky, které jsou naprosto totožné s podmínkami již zmíněného typu. [47]

Obdobně jako typ *Quantity* (viz 6.7.3) lze typ *Collective* rozdělit na více částí. K rozdělení slouží relace *SubCollectionOf* s násobností rovnou jedné na obou stranách. [49]

6.7.5 Funkční celek

Funkční celek je nejběžnějším představitelem relace *celek-část* (viz 6.7). Sám o sobě není univerzálním typem, avšak jako jeho představitele lze použít libovolný univerzální typ. Podobně jako *Collective* (viz 6.7.4) se skládá z částí s různými univerzálními typy, které však mají různé funkce. K propojení celku a části se používá relace *ComponentOf*. [43]

6.8 Aspekty

Aspekty jsou univerzální typy, které charakterizují určité individuální typy a přiřazují jim vnitřní vlastnosti. Těmto individuálním typům se říká nositelé a aspekty jsou na nich existenčně závislé. Krom samotné existenční závislosti je zároveň potřeba, aby byl aspekt a jeho nositel v relaci inherence. Tato relace je ireflexivní, asymetrická a není tranzitivní. [19] V OntoUML se tento vztah značí pomocí *charakterizační relace* (viz 6.6.5). [50] Nositeli aspektů mohou být jiné aspekty a tvořit tak řetězce inherence. Obecně lze aspekty rozdělit na *Quality* a *Mode* [19], kterými se budou zabírat následující dvě podkapitoly.

6.8.1 Quality

Univerzální typ *Quality* se řadí mezi Rigidní Sortální typy. Jedná se o vnitřní vlastnost, která má strukturovanou hodnotu, a která je existenčně závislá na svém nositeli (viz 6.8). [50]

Obecně lze typ *Quality* rozdělit na tři typy:

1. vnímatelné – hodnoty jsou měřitelné pomocí vhodného nástroje
2. nevnímatelné – hodnoty nelze změřit nástrojem
3. nominální – hodnoty odkazují přímo na nositele (například jméno) [50]

Existují čtyři omezující podmínky pro tento typ. První a druhá vynucují nutnost propojení *Quality* a jejího nositele pomocí *charakterizační relace* (viz 6.6.5) a to s násobností jedna na konci u nositele. Třetí podmínka zakazuje všechny nad-typy kromě *Category* a *Mixin*. Poslední pak omezuje možné podtypy pouze na *Subkind*, *Phase* a *Role*. [50]

6.8.2 Mode

Stejně jako *Quality* se *Mode* řadí mezi Rigidní Sortální univerzální typy. Jedná se o vnitřní vlastnost, která má však v tomto případě nestrukturovanou hodnotu, a která je opět existenčně závislá na svém nositeli (viz 6.8). [51]

Čtyři podmínky omezující univerzální typ *Mode* jsou identické s podmínkami uvedenými pro typ *Quality* (viz 6.8.1). [51]

Iniciativa FAIR

Tato kapitola popisuje, jaké všechny vlastnosti musí splňovat data, aby se dala označit jako FAIR, a průběh takzvaného FAIRifikačního procesu.

První myšlenky této iniciativy byly položeny v roce 2014 na konferenci nesoucí název „Společné navrhování datového FAIRPORTu“. Té se zúčastnilo 25 odborníků zastupujících přední výzkumné a politické instituty, specialisty, vědce a vydavatele a společně probírali první návrhy globální infrastruktury podporující objevování, sdílení, publikování a opětovné použití vědeckých dat. Výsledkem této konference byl návrh páteřní sítě, která umožňovala globální sémantickou interoperabilitu (viz 8) datových sad. [52]

Samotná iniciativa FAIR vznikla na základě potřeby již zmíněné znovupoužitelnosti vědeckých dat. Z tohoto důvodu vznikl soubor principů známý též jako *FAIR Data Principles*, který je postaven na čtyřech základních pilířích: *nalezitelnost* (**F**indable), *přístupnost* (**A**ccessible), *interoperabilita* (**I**nteroperable) a *znovupoužitelnost* (**R**eusable), které mimo jiné tvoří akronym této iniciativy. Tyto principy mají sloužit jakožto vodítko pro zlepšení opětovného použití poskytovaných dat. Největší rozdíl, kterým se FAIR odlišuje od ostatních iniciativ, je zaměření se na strojové nalezení a následné zpracování dat. [53]

V následujících čtyřech podkapitolách budou postupně rozebrány významy a kritéria jednotlivých pilířů iniciativy FAIR.

7.1 F – Findable

Prvním principem FAIR je *nalezitelnost*, která zajišťuje snadnou dohledatelnost dat a metadat pro lidi i stroje. Pro její dosažení je potřeba dodržet následující čtyři podmínky:

F1 datům a metadatům musí být přiřazen globálně unikátní identifikátor

F2 data musí být bohatě popsána metadaty (viz 7.4)

F3 metadata musí explicitně zahrnovat identifikátor popisovaných dat

F4 data i metadata musí být registrována, či indexována ve zdroji umožňujícím vyhledávání [54]

7.2 A – Accessible

Druhým principem je *přístupnost*, která nakazuje jasné definování podmínek získání přístupu k datům včetně případné autentizace a autorizace. Pro dosažení tohoto principu musí být splněny následující podmínky:

- A1 data i metadata musí být pomocí jejich identifikátoru vyvolatelná skrz standardizované komunikační protokoly
 - A1.1 komunikační protokol musí být otevřený, bezplatný a univerzálně implementovatelný
 - A1.2 komunikační protokol musí umožňovat autentizaci a autorizaci
- A2 metadata musí být přístupná i v případě zneprístupnění dat [54]

7.3 I – Interoperable

Třetí pilíř iniciativy FAIR je *interoperabilita*, která umožňuje integraci dat s jinými daty a vyžaduje jejich spolupráci s aplikacemi a postupy určenými k analýze, ukládání a zpracování dat. Interoperability lze dosáhnout za pomoci následujících tří podmínek:

- I1 data i metadata musí být reprezentována formálním, přístupným, sdíleným a široce aplikovatelným jazykem
- I2 data i metadata musí využívat slovníky řídicí se FAIR principy
- I3 data a metadata musí zahrnovat kvalifikované odkazy na jiná data či metadata [54]

Sémantickou interoperabilitou se bude zabývat samostatná kapitola (viz 8).

7.4 R – Reusable

Čtvrtým a posledním principem FAIR je *znovupoužitelnost*, která má na starosti optimalizaci opětovného použití dat a umožňuje tak jejich případnou replikaci, kombinaci, nebo obě tyto operace zároveň. Podmínky zajišťující tento princip jsou následující:

- R1 data i metadata musí být bohatě popsána relevantními atributy
 - R1.1 data a metadata musí mít dostupnou jednoznačnou licenci pro jejich používání
 - R1.2 data i metadata musí mít jasný původ
 - R1.3 data a metadata musí splňovat doménovou relevanci vzhledem ke standardům komunity [54]

7.5 Nejasnosti prvotních definic

Z výzkumu uskutečněného na přelomu roků 2016 a 2017 vyšlo najevo, že se může patnáct bodů FAIR principů zdát jednodušší, než ve skutečnosti jsou. Dle tohoto výzkumu byly některé body popsány příliš vágně (např. I3), další byly příliš otevřené (např. I2), jiné zase vyžadovaly výklad od externích stran (např. R4). Podle výzkumníků provádějících tuto analýzu je proto důležité nebrat body FAIR principů jako hůl, ale spíše jako přehled cílů, které jsou důležité, ale jejichž rozsah byl měl být otevřen k diskusi. [55]

V průběhu let vedla nekonzistentní interpretace některých bodů FAIR principů k obavám z možné nekompatibility jejich implementací. Z tohoto důvodu byl na začátku roku 2020 představen koncept úvah týkajících se implementace FAIR, který upřesňoval interpretaci jednotlivých bodů a navrhoval možnosti, jak patnáct bodů FAIR principů implementovat. [56]

7.6 Úprava dat v praxi (FAIRifikační proces)

Zatímco *nalezitelnosti* a *přístupnosti* lze dosáhnout na úrovni metadat bez vynaložení většího úsilí, *interoperabilita* a *znovupoužitelnost* již vyžadují větší práci na úrovni samotných dat. Z tohoto důvodu vznikl takzvaný *FAIRifikační proces*, který se zaměřuje nejen na úpravu dat, ale uvádí i potřebnou práci s jejich metadaty. [57]

Proces se skládá z následujících sedmi kroků:

1. načtení dat určených k FAIRifikaci – získání přístupu a dat
2. zanalyzování načtených dat – kontrola pojmů, struktury a vztahů mezi prvky dat
3. definování sémantického modelu – vytvoření sémantického modelu datové sady a jednoznačné popsání entit a vztahů mezi nimi
4. zajištění propojitelnosti dat – vytvoření propojitelných dat s pomocí sémantického modelu z předchozího kroku
 - pokud není tento krok pro daná data proveditelný, či relevantní, může být vynechán
5. přiřazení licence – definování licence za účelem opětovného použití
6. definování metadat – vytvoření relevantně a bohatě popsaných metadat s důrazem na principy FAIR
7. zveřejnění FAIRifikovaných dat – publikování dat, metadat a licence a umožnění přístupu [57]

Sémantická interoperabilita

Tato kapitola objasňuje, co znamená pojem sémantická interoperabilita, proč je tak důležitá a jak jí lze dosáhnout.

„Sémantická interoperabilita je schopnost interpretovat znalosti importované z jiných jazyků na sémantické úrovni...“ [58] Jedná se o obsahové vyjádření struktury metadat, které umožňuje následné kombinování dat z různých zdrojů. V důsledku tohoto propojení je možnost vyhledávat informace napříč heterogenními datovými sadami v rámci jednoho dotazu. [59]

Sémantická interoperabilita umožňuje mimo jiné řešit problémy související se záměnou termínů. Ta může nastat ve dvou případech. K prvnímu z nich dochází v případě použití stejného termínu pro označení dvou odlišných pojmů. Druhá možná záměna nastává v případě použití různých termínů pro označení stejného pojmu. [59]

Pro dosažení sémantické interoperability je potřeba přiřadit každé znalosti jasnou a správnou interpretaci či sadu modelů [58] a využít vhodný standard pro popis obsahu zdrojů. [59]

V roce 2020 vznikl dokument zabývající se doporučeními pro tvorbu FAIR sémantických artefaktů. Tento dokument obsahoval sedmáct doporučení týkajících se samotných FAIR principů a deset osvědčených postupů zlepšujících celkovou „FAIRovost“ těchto artefaktů. Autoři tohoto dokumentu však zdůraznili, že se jedná spíše o základy pro diskusi než nový standard. [60]

8.1 Sémantický web

Sémantická interoperabilita je jeden z nejdůležitějších požadavků pro *sémantický web*. Ten je realizovaný pomocí anotací webových stránek, které obsahují jak formální, tak neformální znalosti. Anotace samotné jsou pak častokrát vzájemně propojené a závisí na ontologiích a znalostech problematiky. Z tohoto důvodu je potřeba efektivního shromažďování, porovnávání, transformace a skládání již zmiňovaných anotací. Těchto vlastností lze dosáhnout právě pomocí sémantické interoperability. [58]

Projekt OpeNest

Tato kapitola se zaměřuje na projekt OpeNest a jeho části.

OpeNest je projekt spolufinancovaný Evropskou unií, jehož hlavním cílem je vytvořit unikátní inkubátor, který bude propojovat svět marketingu a vytěžování dat. Inkubátor je rozdělen na dvě části. První z nich je *Big Data Arena*, druhou pak samotný inkubátor. [61] O obou těchto částech budou hovořit následující podkapitoly.

9.1 Big Data Arena

Big Data Arena je jednou z částí projektu *OpeNest*. Jedná se o datovou platformu obsahující otevřená data a technologie umožňující jejich jednoduché zpracování. Projekt v současnosti obsahuje stovky datových sad zastupujících všemožné lidské obory. Sady se zaměřují na Českou republiku, ale nemalá část z nich se vztahuje na celou Evropskou unii. Jednotlivé sady jsou s využitím pokročilé umělé inteligence vzájemně propojené, což umožňuje hledání průsečíku napříč na první pohled heterogenních dat. [61]

Datová platforma obsahuje kromě kontroly kvality dat a analytických a statických nástrojů i sedm funkcí specifických pro tento projekt:

1. *datový katalog* umožňující zobrazení deskriptivní statistiky hodnot, vytvoření sémantického modelu a zobrazení mapy s hodnotami (u geografických dat)
2. *krokový uživatelský wizard* s funkcí rozpoznávání entit pro usnadnění přidávání nových datových sad
3. *nástroje pro sběr dat z webu* sloužící k integraci nových otevřených datových sad
4. *modul transformace* pro zpracování vkládaných datových sad a umožnění jejich opětovného použití
5. *dotazovací jazyk* poskytující algoritmy pro dotazování na spojitosti mezi daty
6. *ontologické konceptuální modely* graficky zobrazující propojení entit a jednotlivých datových sad
7. *mapový modul* umožňující vizualizaci bodů s možností optimalizace trasy [61]

9.2 Inkubátor

Úkolem podnikatelských inkubátorů je poskytovat začínajícím jednotlivcům či firmám zdroje, které pro ně mohou být v rané fázi jejich života těžko dostupné. [62] Inkubátorová část projektu *OpeNest* se v tomto ohledu zaměřuje na jedince a firmy, kterým k úspěchu schází propagace a marketingová strategie. V rámci těchto služeb nabízí inkubátor několik programů, které kromě samotného přístupu k *datové platformě* poskytují například pronájem prostorů, či možnost pravidelných konzultací s odborníky z různých oborů. [61]

9.3 Remmark, a.s.

Projekt *OpeNest* je jedním z mnoha projektů agentury *Remmark, a.s.* [61] Tato česká full service komunikační agentura se pohybuje na trhu již od roku 1998 a je dlouholetým členem *Asociace komunikačních agentur*. Za sebou má již stovky úspěšných projektů a kampaní a jejich tým tvoří velká škála profesionálů a odborníků přes PR a marketing. [63]

Platforma OpenPonk

Tato kapitola shrnuje základní informace o modelovací platformě OpenPonk.

OpenPonk je modelovací platforma implementovaná v jazyce *Pharo*. [64] Tento jazyk je čistě objektově orientovaný a vychází z jazyka *Smalltalk*. [65] Platforma je vyvíjena jako open-source software *Centrem pro konceptuální modelování a implementaci* a je proto zdarma ke stažení. [64] Platforma je určena pro výzkumníky, odborníky z praxe, ale i pro studenty. [66]

Kromě modulů pro digram tříd UML (viz 3.2) a profil *OntoUML* (viz 6) nabízí *OpenPonk* i moduly pro práci s *Business Objects Relation Modeling Object-Relation Diagrams* (BORM ORD), konečnými automaty (FSM), *Business Process Model and Notation* (BPMN) a Petriho sítí. [64]

Modul profilu *OntoUML* obsahuje dva podpůrné frameworky sloužící k verifikaci vytvořených modelů. První z nich kontroluje jednotlivé objekty a jejich relace dle definic popsaných v kapitole 6. [67] Druhý pak detekuje možné anti-patterny. [68] Kromě těchto modulů byla do platformy *OpenPonk* nově také přidána možnost propojení s *Gitem*, která umožňuje stahování a nahrávání modelů, možnost nahrání hotových modelů přímo do datové platformy projektu *OpeNest* (viz 9) a nástroj na vyhledávání entit a kontrolu jejich názvů.

Část II

PRAKTICKÁ ČÁST

Kapitola 11

Analýza vybraných datových sad

Tato kapitola popisuje analýzu vybraných datových sad, jejich dat a s nimi souvisejících klíčových domén.

V rámci této bakalářské práce bylo ve spolupráci s vedoucím, doc. Ing. Robertem Perglem, Ph.D., vybráno šest datových sad. Mezi těchto šest sad patří: *Počet důchodců s exekuční srážkou podle krajů*, *Počet komerčních letišť*, *Pohyb obyvatel* (za ČR, kraje, okresy, so, orp a obce – rok 2019), *Registr poskytovatelů zdravotních služeb*, *Seznam bank a poboček zahraničních bank na území ČR* a *Statistika dražeb dle krajů*.

První zmíněnou datovou sadou je *Počet důchodců s exekuční srážkou podle krajů*. Tato sada obsahuje informace o počtu osob pobírajících důchod, o průměrné výši jejich důchodu a o průměrné výši exekuční srážky z tohoto důchodu. Kromě těchto údajů je v sadě obsaženo také datum, ke kterému je údaj uveden, průměrný věk a pohlaví osob, které byly do této statistiky započítány, a hodnoty sloužící k identifikaci kraje, pod který tyto osoby spadají. Nejen z těchto informací vyplynula v závěru analýzy nutnost propojit území, osobu, důchod a exekuci.

Druhá datová sada, *Počet komerčních letišť*, byla zpracovávána v takzvaném „Blitzkrieg“ režimu (viz 14) a její analýza tak probíhala ve dvou částech. Data obsažená v této sadě poskytují informace o počtu komerčních letišť v závislosti na jejich lokalitě a referenčním období. V důsledku analýzy tak bylo potřeba propojit letiště, jejich počty a lokalitu, do které spadají.

Třetí datová sada se zkráceným názvem *Pohyb obyvatel* se zabývá lidskou migrací. Údaje v ní obsažené se týkají nejen samotné migrace (přirozený přírůstek, migrační saldo, střední stav obyvatel, ...), ale i geografického území a referenčního období, ve kterém byla data získána. Z tohoto důvodu bylo u této sady ve výsledku potřeba propojit území, osobu a samotné entity potřebné ke korektnímu zachycení lidské migrace.

Čtvrtá datová sada je *Registr poskytovatelů zdravotních služeb*. Tato sada je ze všech nejobsáhlejší a její zpracování tak vyžadovalo podrobnější ontologickou analýzu. Registr obsahuje informace o poskytovatelích zdravotních služeb. Příkladem těchto údajů může být například rozlišení mezi právnickou a fyzickou osobou, identifikační číslo osoby, či forma, obor a druh poskytované péče. Kromě informací o poskytovatelích obsahuje sada také data o jejich sídlech a zdravotnických zařízeních, které provozují. Z analýzy vyplynulo, že bude nutno propojit poskytovatele, jimi poskytovanou péči, zdravotnické zařízení a sídlo. Jak zdravotnické zařízení, tak sídlo poskytovatele jsou však identifikovány adresou (územím). Z tohoto důvodu bylo potřeba rozmyslet i způsob, jakým údaje v jednom modelu rozlišit. Tato myšlenka byla hlavním podmětem pro rozšíření šablony *Území-adresa*, která následně pomohla korektně rozlišit adresy problematických entit.

Pátá datová sada *Seznam bank a poboček zahraničních bank na území ČR* obsahuje data potřebná k identifikaci adresy pobočky banky, jako jsou ulice, obec a číslo domovní. Kromě

údajů týkajících se adresy sada obsahuje také identifikační údaje samotné banky. Zde se jedná například o název, či identifikační číslo osoby. Přesto, že sada je oproti ostatním menší, bylo i zde potřeba propojit území (adresu) a údaje týkající se banky samotné.

Šestá a poslední datová sada s názvem *Statistika dražeb dle krajů* poskytuje informace o počtu a celkových výnosech z dražeb monitorovaných skupin věcí a o časovém období, ve kterém byly dané dražby uskutečněny. Dále pak obsahuje údaje o orgánech finanční správy zaštiťující samotné dražby. Z analýzy této datové sady tak vyplynulo, že bude potřeba propojit území, dražby a jejich pořádání.

Z výše uvedené analýzy jednotlivých datových sad vyplynulo, že každá spadá pod jinou klíčovou doménu, avšak až na datovou sadu *Počet komerčních letišť* se ve všech nachází územní členění. Tohoto faktu bylo následně využito k rozšíření šablony *Území-adresa* pro zajištění sémantické interoperability.

Během analýzy bylo kromě internetu využito i popisů datových sad poskytnutých společností Remmark, a.s. v rámci projektu OpeNest (viz 9). Tyto popisy měly za úkol přiblížit datovou sadu a její data a ušetřit tak čas strávený hledáním informací potřebných k provedení korektní ontologické analýzy domén a sad. I přes tuto snahu bylo v mnoha případech potřeba informace dohledávat a v krajních případech konzultovat úpravu popisů.

Tvorba konceptuálních modelů

Tato kapitola se zaměřuje na tvorbu konceptuálních modelů vybraných datových sad a vysvětluje význam takzvaných šablon.

Analýzu datových sad následovala samotná tvorba konceptuálních modelů těchto sad. Za tímto účelem byl vybrán profil OntoUML (viz 6), který rozšiřuje jazyk UML (viz 3) o ontologii UFO (viz 5) a poskytuje tak základ pro dosažení sémanticky jednoznačného a ontologicky přesného konceptuálního modelu. Jakožto nástroj pro samotnou tvorbu byla vybrána platforma OpenPonk (viz 10), která poskytuje podporu pro práci s profilem OntoUML a na jejímž vývoji se podílela agentura Remmark, a.s. v rámci podpory projektu *OpeNest* (viz 9).

Během tvorby bylo využito rámce z roku 1994 (viz 2.2). Ten byl původně určen ke kontrole kvality konceptuálního modelování, ale jeho interpretace lze využít i během samotného modelování. Na základě interpretace toho rámce byly vždy nejprve zachyceny entity, které přímo vyplývají z analýzy datové sady a jejích dat. Následně byly entity propojeny základními vztahy, které však zatím nebyly vztaženy k dané doměně. Poté byly modely obohaceny o pomocné entity, které jsou podstatné pro danou doménu. Díky těmto pomocným entitám byly následně vytvořeny vztahy mezi těmito a původními entitami tak, aby zachycovaly význam v rámci dané domény. Poslední částí tvorby každého konceptuálního modelu byla prezentace modelu ostatním členům týmu. Jejím účelem bylo odhalit případné nedostatky týkající se interpretace vzniklého modelu.

Prezentace probíhala ve dvou etapách. První etapa bylo prezentování vytvořeného modelu na týmové schůzce, která se konala jednou týdně a jejíž odezva vystihovala možnou interpretaci širokého publika. Druhá etapa pak představovala modelové „posouzení kódu“ (code review), které probíhalo v závěru tvorby a jednalo se o detailní analýzu modelu, kterou prováděli vždy alespoň dva členové týmu. Zpracování těchto připomínek představovalo zakončení tvorby konceptuálního modelu, kterou následovalo datové mapování rozebírané v následující kapitole.

12.1 Šablony

Během konceptuálního modelování vybraných datových sad bylo ve značné míře využito takzvaných šablon. Jejich hlavním účelem bylo sloužit jakožto kostra pro modelování určitých domén či jejich částí. Tento speciální druh konceptuálního modelu nereprezentoval žádnou datovou sadu a nebyly k němu tak připojeny datové entity, nebo mapovací pravidla.

Příkladem šablony je konceptuální model *Území-adresa*, který byl v rámci této bakalářské práce rozšířen o rozdělení budovy, jejích čísel, o adresu a o poštovní adresu, která šablonu propojila s entitami osoby a právnické osoby.

■ **Tabulka 12.1** Struktura datové sady *Seznam bank a poboček zahraničních bank na území ČR*

název	popis	formát
<code>cislo_domovni</code>	číslo domovní adresy banky	Integer
<code>cislo_orientacni</code>	číslo orientační adresy banky	Integer
<code>datum_vzniku</code>	datum vzniku oprávnění banky	String
<code>ico</code>	IČO banky	Integer
<code>nazev</code>	název banky	String
<code>nazev_casti_obce</code>	název části obce adresy banky	String
<code>nazev_obce</code>	název obce adresy banky	String
<code>nazev_ulice</code>	název ulice adresy banky	String
<code>psc</code>	PSC adresy banky	String
<code>zeme_kod</code>	kód státu adresy banky	String
<code>znak_cisla_orientacniho</code>	znak orientačního čísla adresy banky	String

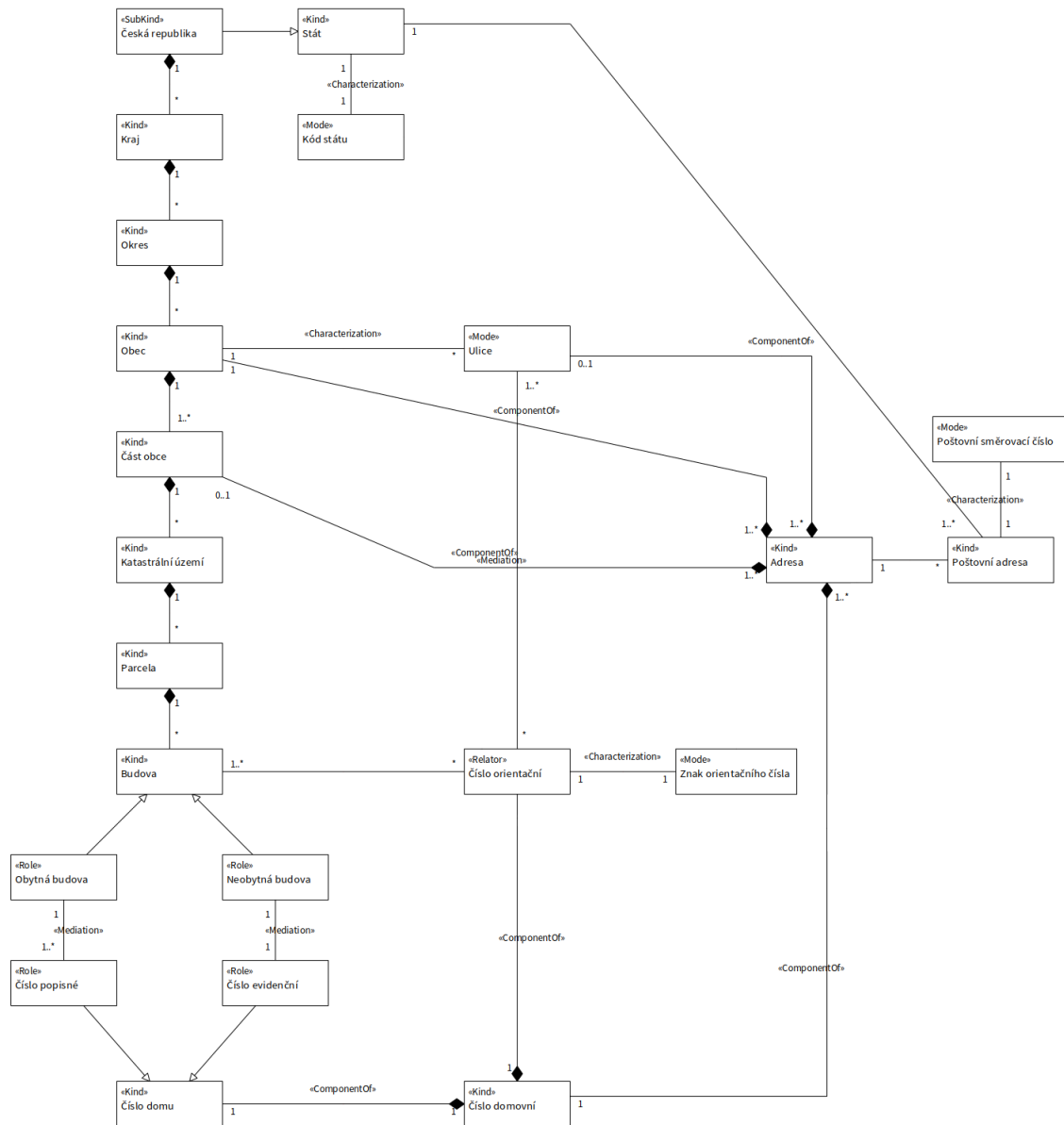
12.2 Příklad tvorby konceptuálního modelu

Pro ukázkou postupu zmíněného v této kapitole a využití šablon během modelování byla vybrána datová sada *Seznam bank a poboček zahraničních bank na území ČR*, jejíž název bude pro účely této kapitoly zkrácen na „*Seznam bank a poboček*“.

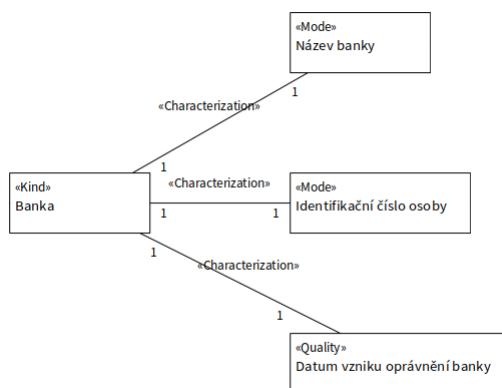
Z analýzy provedené v předchozí kapitole vyplynula potřeba propojit adresu s informacemi sloužícími k identifikaci banky. Ze struktury dat uvedených ve zjednodušené tabulce 12.1 vzešlo jedenáct entit: *číslo domu*, *číslo orientační*, *datum vzniku oprávnění banky*, *identifikační číslo osoby*, *název banky*, *část obce*, *obec*, *ulice*, *poštovní směrovací číslo*, *kód státu* a *znak orientačního čísla*. Dle interpretace rámce zmíněné na začátku této kapitoly byly následně propojeny entity, jejichž vztahy šly vyjádřit bez nutnosti pomocných entit. Jednalo se o trojici *část obce*, *obec* a *ulice* a o dvojici *číslo orientační* a *znak orientačního čísla*.

Další fází bylo přidání pomocných entit a jejich následné propojení vztahy s ohledem na význam v doméně. Entity proto byly rozděleny do dvou skupin. První z nich reprezentovala údaje týkající se adresy banky či její pobočky, druhá pak informace o samotné bance. Vzhledem k tematice první skupiny bylo k propojení entit v ní obsažených využito rozšířené šablony *Území-adresa*. Propojení entit první skupiny lze pozorovat na obrázku 12.1. Druhá skupina byla propojena přímo s entitou reprezentující banku, protože se jednalo o univerzální typ *Kind* a jeho aspekty. Propojení druhé skupiny lze pozorovat na obrázku 12.2. V závěru modelování bylo potřeba vzájemně propojit tyto dvě skupiny. Toho bylo dosaženo pomocí dvou univerzálních typů *Role*, které pomocí generalizace a mediačních relací propojily obě skupiny a vytvořily tak celistvý model. Vzájemné propojení obou entit je zachyceno na obrázku 12.3.

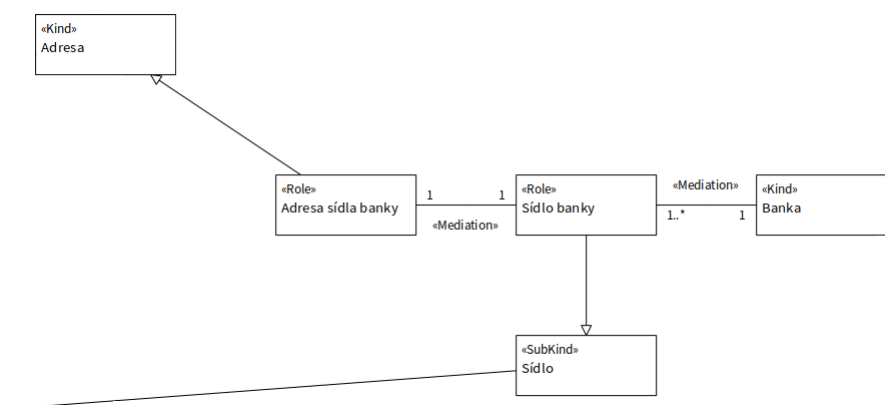
Poslední fází byla prezentace vzniklého konceptuálního modelu. Tuto fázi prošel vytvořený model bez připomínek.



■ **Obrázek 12.1** Propojení entit modelu *Seznam bank a poboček zahraničních bank na území ČR* reprezentující údaje týkající se adresy banky či její pobočky



■ **Obrázek 12.2** Propojení entit modelu *Seznam bank a poboček zahraničních bank na území ČR* obsahující informace o bance



■ **Obrázek 12.3** Vzájemné propojení skupin modelu *Seznam bank a poboček zahraničních bank na území ČR*

Mapování dat na entity

Tato kapitola se zabývá tvorbou mapovacích pravidel pro propojení dat s entitami konceptuálních modelů a využitím MEA ontologie pro zajištění jednotného formátu jednotek.

Hotové konceptuální modely bylo potřeba propojit s daty obsaženými v datové sadě. Propojení bylo dosaženo vytvořením takzvaných *data entit* a *mapovacích pravidel*, které byly následně za pomoci asociace spojeny s jednotlivými modely, respektive s entitami, které byly obsaženy v mapování. Jak *data entity*, tak *mapovací pravidla* představují nezbytnou součást konceptuálních modelů, která mimo jiné pomáhá dodržet *FAIR Data Principles* (viz 7).

Prvním z konstruktů mapování dat je *data entita*. Tento speciální druh entity je reprezentován univerzálním typem *Data*, který však není ukotven v UFO (viz 5) a slouží tak pouze k odlišení této entity od ostatních. *Data entita* jako taková musí vždy obsahovat unikátní identifikátor datové sady a svým názvem reprezentovat atributy, které jsou v ní obsaženy. Kromě unikátnosti je identifikátor vždy statický a typu *String*. Typy atributů datové sady zahrnutých v entitě závisí na formátu uvedeném v datové sadě (*Integer* či *String*) a jejich případná vnitřní struktura je zachycena pomocí MEA ontologie, která je součástí *mapovacích pravidel* a bude rozebrána v samostatné podkapitole 13.2.

Druhým konstruktem jsou *mapovací pravidla*, která se zapisují do komentáře příslušné *data entity* a jejichž úkolem je namapovat atributy obsažené v datové sadě na entity vytvořeného konceptuálního modelu. Pravidla se dělí na tři druhy: *základní pravidla*, *pravidla podmíněná hodnotou atributu* a *podmíněná pravidla*. V *mapovacích pravidlech* lze také využívat logických spojek „AND“ a „OR“ a to jak v části atributů, tak i v části entit.

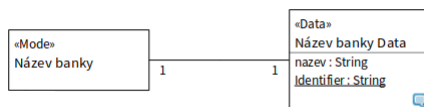
13.1 Druhy mapovacích pravidel

Základní pravidlo, jak již název napovídá, představuje nejjednodušší formu *mapovacího pravidla* a mapuje hodnoty ze sloupce přímo na entitu. Strukturu tohoto pravidla lze pozorovat ve výpisu 13.1.

■ Výpis kódu 13.1 Struktura základního mapovacího pravidla

```
"atribut datové sady" -> "entita konceptuálního modelu";
```

Pravidlo podmíněné hodnotou atributu reprezentuje formu, která mapuje hodnoty sloupce na entitu v závislosti na jejich hodnotě. Toto pravidlo se ve většině případů využívá v pozitivním tvaru, ale lze využít i takzvané „Hampejsovy negace“. „Hampejsova negace“ byla vymyšlena v rámci této bakalářské práce v reakci na data obsažená v jedné ze zpracovávaných datových sad. Strukturu klasického tvaru i tvaru v negaci lze pozorovat ve výpisu 13.2.



■ **Obrázek 13.1** Data entita názvu banky z modelu *Seznam bank a poboček zahraničních bank na území ČR*

■ **Výpis kódu 13.2** Struktura pravidla podmíněného hodnotou

```

"atribut datové sady" = "hodnota atributu" -> "entita koncept. modelu";
"atribut datové sady" != "hodnota atributu" -> "entita koncept. modelu";
  
```

Poslední formou *mapovacího pravidla* je *podmíněné pravidlo*. Tento druh umožňuje vyjádřit podmínku omezující mapování hodnot sloupce v závislosti na hodnotě jiného sloupce a lze ho kombinovat s *pravidlem podmíněným hodnotou atributu*. Základní strukturu *podmíněného pravidla* včetně její kombinace s *pravidlem podmíněným hodnotou atributu* lze pozorovat ve výpisu 13.3.

■ **Výpis kódu 13.3** Struktura podmíněné pravidla

```

("podmiňující atribut datové sady" = "podmiňující hodnota atributu") ->
  ("podmíněný atribut datové sady" -> "entita konceptuálního modelu");

("podmiňující atribut datové sady" = "podmiňující hodnota atributu") ->
  ("podmíněný atribut datové sady" = "hodnota podmíněného atributu" ->
    "entita konceptuálního modelu");
  
```

13.2 MEA ontologie

Ontologie MEA slouží k zachycení vnitřní struktury, měř a jednotek hodnot sloupců mapovaných pomocí *mapovacích pravidel*. Obsahuje základní a odvozené jednotky SI, časové jednotky, měny, statistické míry a způsob zpracování rozsahů. MEA ontologie se zapisuje do *mapovacích pravidel* obsažených v *data entitách* a to vždy za část „entita konceptuálního modelu“. Jednotlivé prvky ontologie se k této části přidávají pomocí logické spojky „AND“ (viz výpis 13.4).

■ **Výpis kódu 13.4** Struktura pravidel s prvky MEA ontologie

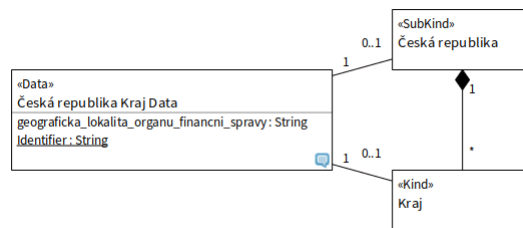
```

"atribut datové sady" -> "entita konceptuálního modelu" AND
  "prvek MEA ontologie" AND "prvek MEA ontologie";

("podmiňující atribut datové sady" = "podmiňující hodnota atributu") ->
  ("podmíněný atribut datové sady" = "hodnota podmíněného atributu" ->
    "entita konceptuálního modelu" AND "prvek MEA ontologie");
  
```

13.3 Příklady mapování dat

Prvním uvedeným příkladem je sloupec „nazev“ (viz tabulka 12.1) z datové sady *Seznam bank a poboček zahraničních bank na území ČR* rozebírané v příkladu v předchozí kapitole. Tento sloupec je typu String a obsahuje názvy bank. Pro jeho namapování byla vytvořena datová entita „Název banky Data“ (viz obrázek 13.1), která byla asociací propojena s entitou „Název



■ **Obrázek 13.2** Data entita lokality finančního orgánu z modelu *Statistika dražeb dle krajů*

banky“ obsaženou v konceptuálním modelu. Entita kromě identifikátoru a atributu obsahovala také jedno základní mapovací pravidlo (viz výpis 13.5).

■ **Výpis kódu 13.5** Pravidlo mapující hodnoty sloupce „navez“ datové sady *Seznam bank a poboček zahraničních bank na území ČR*

```
"navez" -> "Název banky";
```

Druhý příklad je z datové sady *Statistika dražeb dle krajů*. Sada obsahuje sloupec „geograficka lokalita organu financni spravy“, který je typu String a jehož hodnoty bylo potřeba namapovat na entity „Česká republika“ a „Kraj“. Za tímto účelem byla vytvořena datová entita „Česká republika a Kraj Data“ (viz obrázek 13.2), která obsahovala dvě mapovací pravidla podmíněná hodnotou a byla propojena s oběma entitami. V mapovacích pravidlech této entity bylo využito „Hampejsovy negace“ pro korektní namapování hodnot tohoto sloupce (viz výpis 13.6).

■ **Výpis kódu 13.6** Pravidlo mapující hodnoty sloupce „geograficka lokalita organu financni spravy“ datové sady *Statistika dražeb dle krajů*

```
"geograficka_lokalita_organu_financni_spravy" =
  "Česká republika" -> "Česká republika";

"geograficka_lokalita_organu_financni_spravy" !=
  "Česká republika" -> "Kraj";
```

Posledním příkladem je dvojice sloupců „tra_infr“ a „value“, které jsou obsaženy v datové sadě *Počet komerčních letišť*. Sloupce jsou typů String a Integer a společně byly mapovány na čtyři entity: „Počet komerčních letišť“, „Počet hlavních komerčních letišť“, „Počet malých komerčních letišť“ a „Počet ostatních komerčních letišť“. K mapování byla vytvořena entita „Počet komerčních letišť Data“ (viz obrázek 13.3), jejíž součástí byla čtyři podmíněná mapovací pravidla obsahující MEA ontologii (viz výpis 13.7).

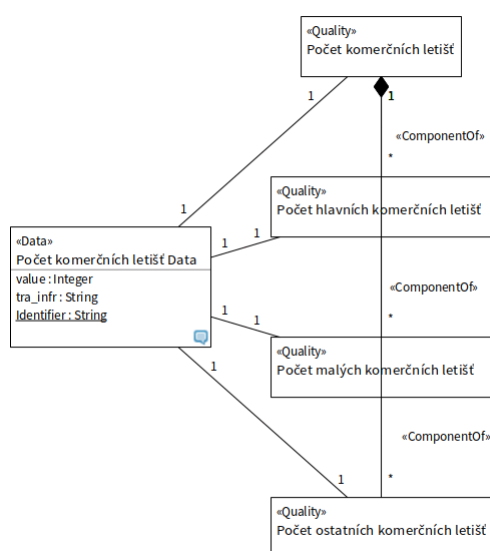
■ **Výpis kódu 13.7** Pravidlo mapující hodnoty sloupců „tra_infr“ a „value“ datové sady *Počet komerčních letišť*

```
("tra_infr"="TOTAL" OR "tra_infr"="AIRP") -> ("value" ->
  "Počet komerčních letišť" AND "MEA_Počet" AND "MEA_Rational");

("tra_infr"="AIRP_MAIN") -> ("value" ->
  "Počet hlavních komerčních letišť" AND "MEA_Počet" AND "MEA_Rational");

("tra_infr"="AIRP_SML") -> ("value" ->
  "Počet malých komerčních letišť" AND "MEA_Počet" AND "MEA_Rational");

("tra_infr"="AIRP_OTH") -> ("value" ->
  "Počet ostatních komerčních letišť" AND "MEA_Počet" AND "MEA_Rational");
```



■ **Obrázek 13.3** Data entita počtu komerčních letišť z modelu *Počet komerčních letišť*

„Blitzkrieg“ modelování

Tato kapitola vysvětluje význam „Blitzkrieg“ modelování a objasňuje, co tento postup obnášel.

Takzvané „Blitzkrieg“ modelování byl postup vymyšlen doc. Ing. Robertem Perglem, Ph.D. jakožto reakce na potřeby projektu *OpeNest* (viz 9). Hlavním úkolem tohoto postupu bylo urychlení dodání hotových modelů s přijmutím možných drobných nepřesností. Tento postup ovlivnil celkovou sémantickou interoperabilitu modelů a modely tak musely být následně zkontrolovány a případně zpřesněny. Vzhledem ke změně pracovního postupu ze sekvenčního na iterativní vznikla potřeba evidovat stav zpracování jednotlivých datových sad. K tomuto účelu bylo využito webové aplikace *Asana*, která kromě řízení toku umožňovala i přidávání komentářů a štítků. Nejenom díky těmto vlastnostem se *Asana* stala nedílnou součástí projektu a je využívána i po zrušení „Blitzkrieg“ modelování.

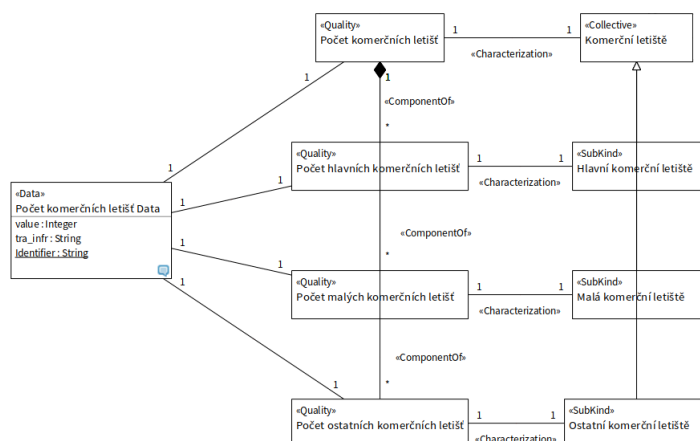
14.1 Rozdíly oproti běžnému postupu

Hlavní rozdíl postupování podle „Blitzkrieg“ modelování spočíval v samotné analýze datových sad. Postup totiž stanovoval, že by analýza měla probíhat pouze nad daty a jejich popisy poskytnutými společností Remmark, a.s. v rámci projektu. Z tohoto důvodu nebyly v první fázi nikterak analyzovány klíčové domény, což mohlo vést k již zmiňovaným nepřesnostem.

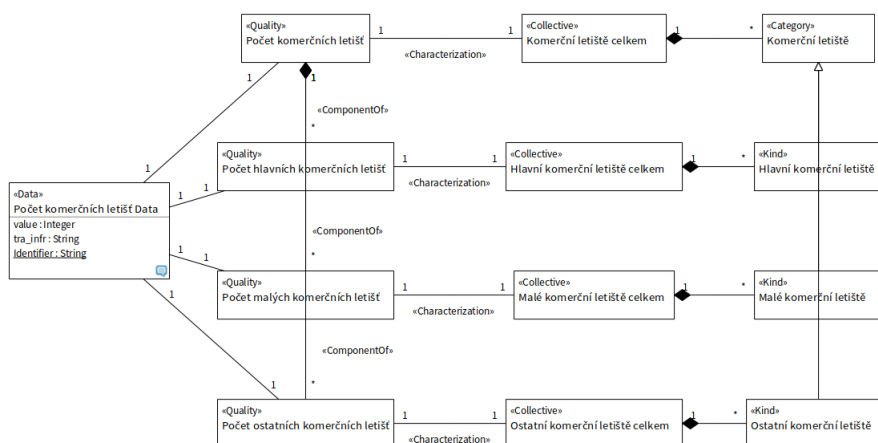
Zavedení tohoto postupu ovlivnilo také tvorbu konceptuálních modelů, která byla omezena pouze na zachycení entit, které přímo vyplývají z analýzy datové sady a jejích dat, následované jejich propojením základními vztahy. Nepropojené entity či skupiny entit byly následně spojeny pomocnými entitami. Tyto pomocné entity umožnily vzájemné propojení entit datové sady, avšak mohly zanést nepřesnosti vycházející z opomenutí analýzy domény v předchozí části. Kromě těchto změn také došlo k vynechání prezentace modelů v rámci týmových schůzek a „posouzení kódu“.

14.2 Zpřesňování konceptuálních modelů

Po skončení režimu „Blitzkrieg“ modelování bylo potřeba zkontrolovat a případně zpřesnit vytvořené konceptuální modely tak, aby byly korektně vztaheny ke klíčovým doménám a byla zajištěna jejich sémantická interoperabilita. V tomto kroku byla využita již zmiňovaná webová aplikace *Asana*, ve které byly datové sady zpracované v tomto režimu navráceny do stavu „k modelování“, přičemž k nim byl přiřazen štítek „blitzkrieg“. K sadám byla následně přiřazena osoba odpovědná za jejich kontrolu a zpřesnění. Zpřesňování modelů se již řídilo běžným postupem, jehož výstupem byl přesný a sémanticky interoperabilní model.



■ **Obrázek 14.1** Počet komerčních letišť z modelu *Počet komerčních letišť* zpracovávaný v „Blitzkrieg“ režimu



■ **Obrázek 14.2** Zpřesněný počet komerčních letišť z modelu *Počet komerčních letišť*

14.3 Příklady zpřesňování konceptuálních modelů

„Blitzkrieg“ modelování bylo zavedeno až ke konci této bakalářské práce. Z tohoto důvodu pocházejí všechny ukázky z datové sady *Počet komerčních letišť*, která byla jako jediná zpracována v tomto režimu.

První nepřesností nacházející se v tomto modelu bylo propojení jednotlivých druhů letišť a jejich počtů. Na obrázku 14.1 lze pozorovat část modelu vytvořenou v rámci „Blitzkrieg“ modelování. V této variantě byly propojeny druhy letišť přímo s jejich počty, což nebylo ontologicky korektní. Tato nepřesnost byla opravena přidáním pomocných entit univerzálního typu *Collective*, ke kterým byly následně jednotlivé druhy připojeny pomocí agregace (viz obrázek 14.2).

Druhá nepřesnost spočívala v univerzálních typech samotných druhů letišť. Z původní analýzy vzešlo „Komerční letiště“ jakožto *Collective* a jeho druhy jako *Subkind* (viz obrázek 14.1). Podrobnější analýza prováděná v rámci kontroly však ukázala, že bylo jednotlivé druhy lepší typovat jako *Kind* a samotné „Komerční letiště“ pak jako *Category* (viz obrázek 14.2).

Model *Počet komerčních letišť* vytvořený pomocí postupu „Blitzkrieg“ modelování obsahoval pár dalších nepřesností, které však nebudou v rámci této kapitoly rozebírány.

Zajištění sémantické interoperability

Tato závěrečná kapitola pojednává o zajištění sémantické interoperability jak mezi nově vzniklými modely, tak i mezi nimi a modely již existujícími.

V rámci této bakalářské práce bylo dosaženo sémantické interoperability nejen mezi ontologickými konceptuálními modely zpracovanými v této práci, ale i mezi modely, které již byly nahrány v datové platformě projektu *OpeNest* (viz 9). Tento úkol by byl nemožný bez vhodných podpůrných nástrojů, mezi které patří například automatická kontrola názvů entit prováděná během nahrávání modelu do datové platformy, či tabulka s názvy všech existujících entit, která byla průběžně doplňována a kontrolována.

Klíčovým bodem sémantické interoperability zpracovaných datových sad v této bakalářské práci byla šablona *Území-adresa* (viz 12.1), jejíž rozšíření umožnilo vzájemné propojení jak nově vytvořených konceptuálních modelů, tak i modelů již existujících. Příkladem dosažení sémantické interoperability s využitím rozšířené šablony může být entita „Adresa“ v modelech *Registr poskytovatelů zdravotních služeb* (viz obrázek A.1) a *Seznam bank a poboček zahraničních bank na území ČR* (viz obrázek A.2).

Další příklad zajištění sémantické interoperability může představovat dvojice entit „Obyvatel“ a „Osoba“, které se nachází v modelech *Počet důchodců s exekuční srážkou podle krajů* (viz obrázek A.3) a *Pohyb obyvatel za ČR, kraje, okresy, so, orp a obce – rok 2019* (viz obrázek A.4).

Ukázkou propojení konceptuálních modelů zpracovanými v této práci a modelů, které již existovaly v datové platformě, může být entita „Motorové vozidlo“ obsažené v modelech *Statistika dražeb dle krajů* (viz obrázek A.5) a *EU výdaje domácností na konečnou spotřebu podle účelu spotřeb* (viz obrázek A.6).

Posledním příkladem sémantické interoperability je entita „Evropská unie“, která se nachází v modelech *Počet komerčních letišť* (viz obrázek A.7) a *EU osoby, které nahlásily nehodu vedoucí k úrazu* (viz obrázek A.8).

Vytvoření konceptuálních modelů a zajištění sémantické interoperability mezi nově vzniklými, ale i již existujícími modely bylo mimo jiné jedním (třetím) ze sedmi kroků *FAIRifikační procesu* (viz 7.6). Tento proces je používán v rámci projektu *OpeNest* a upravuje získaná data tak, aby dodržovala všech patnáct bodů FAIR principů (viz 7).

Závěr

Cílem této bakalářské práce byla ontologická analýza klíčových domén a jejich propojení s datovými sadami tak, aby byla zajištěna jejich sémantická interoperabilita. Klíčové domény byly vybrány ve spolupráci s vedoucím práce, doc. Ing. Robertem Perglem, Ph.D., přičemž byly brány v potaz potřeby projektu OpeNest společnosti Remmark, a.s. a vhodnost pro účely bakalářské práce.

Pro dosažení cílů stanovených v této práci bylo zapotřebí provést analýzu nejen samotných klíčových domén, ale i samotných datových sad a dat v nich obsažených. Na základě této analýzy byly následně vytvořeny konceptuální modely těchto domén, pro jejichž tvorbu byl zvolen ontologický modelovací jazyk OntoUML a platforma OpenPonk.

Hotové modely byly následně propojeny s datovými sadami a byla provedena kontrola jejich sémantické interoperability s již existujícími modely. Propojení bylo uskutečněno vytvořením takzvaných data entit, jejichž atributy byly pomocí mapovacích pravidel namapovány na entity modelů. V mapovacích pravidlech bylo využito MEA ontologie, která pomáhá se zachycením měr a jednotek, a takzvané „Hampejsovy negace“, která byla vymyšlena v rámci této práce.

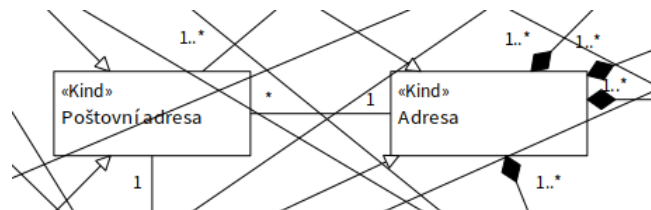
Kromě samotných konceptuálních modelů došlo v rámci této bakalářské práce také k rozšíření jedné šablony. Tento speciální druh konceptuálního modelu slouží jako podklad pro modelování domén, či jejich částí a neobsahuje tak žádné mapování entit a dat.

Využitím vhodného konceptuálního jazyka podporujícího ontologické konstrukty, konzistentního pojmenování entit a s kladením velkého důrazu na analýzu domény bylo dosaženo sémantické interoperability se všemi doposud vytvořenými modely. Modely byly následně nahrány do datové platformy, která je, díky již zmiňované sémantické interoperabilitě, dokáže vzájemně propojit a provádět nad nimi dotazy.

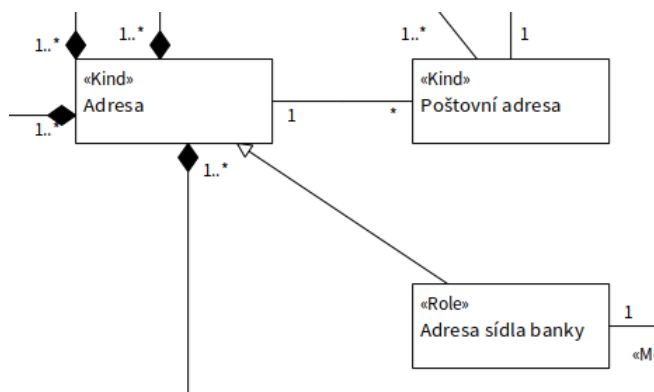
Do budoucna se v projektu OpeNest plánuje nejen přidávání nových datových sad pro zajištění co nejefektivnějšího využití datové platformy, ale i provádění pravidelných kontrol a případných oprav stávajících modelů. V rámci tohoto projektu se také rozvíjí platforma OpenPonk, na které se chystá možnost přesouvání či mazání více modelů současně.

Příklady sémantické interoperability

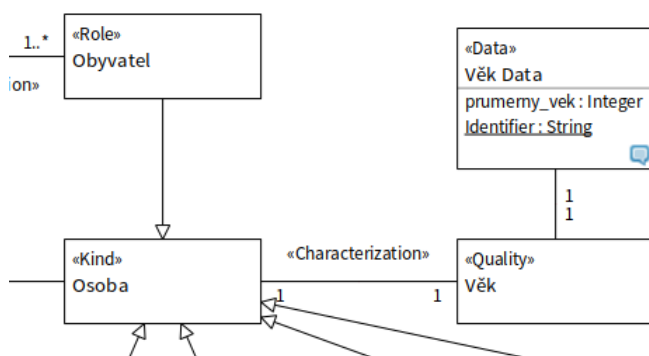
Části ontologických konceptuálních modelů zajišťující sémantickou interoperabilitu nad heterogenními datovými sadami zpracovanými v rámci této bakalářské práce.



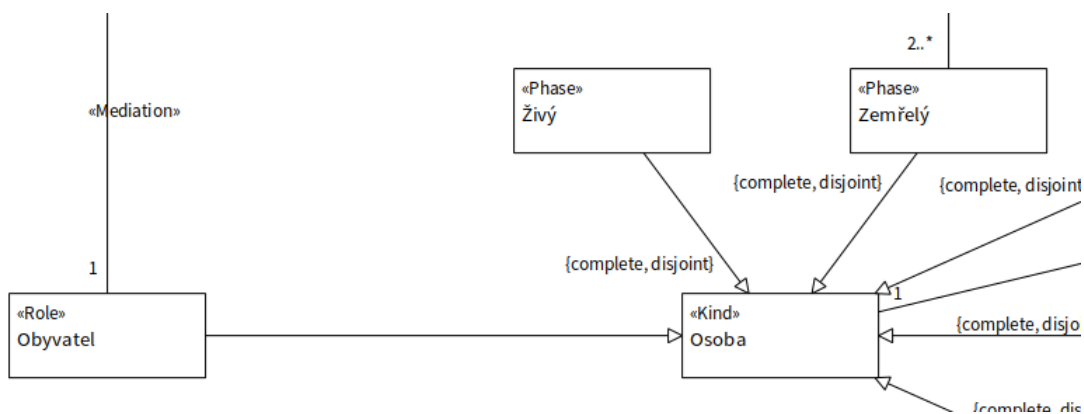
■ Obrázek A.1 Entita „Adresa“ v modelu *Registr poskytovatelů zdravotních služeb*



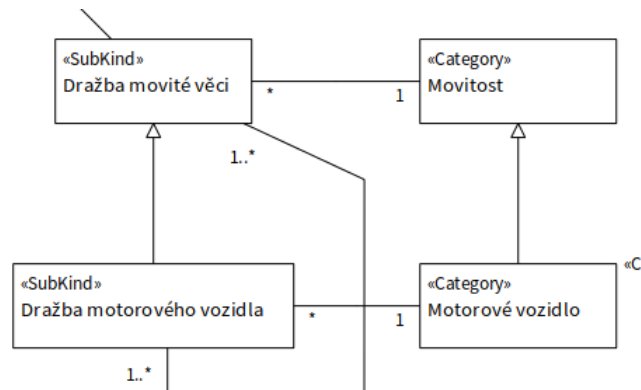
■ Obrázek A.2 Entita „Adresa“ v modelu *Seznam bank a poboček zahraničních bank na území ČR*



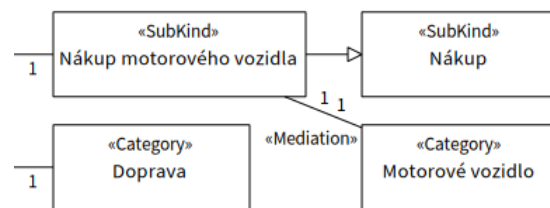
■ Obrázek A.3 Entity „Obyvatel“ a „Osoba“ v modelu *Počet důchodců s exekucí srážkou podle krajů*



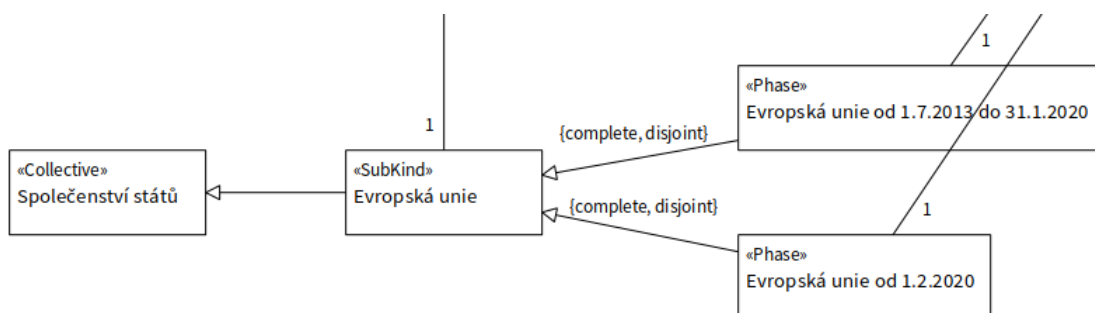
■ Obrázek A.4 Entity „Obyvatel“ a „Osoba“ v modelu *Pohyb obyvatel za ČR, kraje, okresy, so, orp a obce – rok 2019*



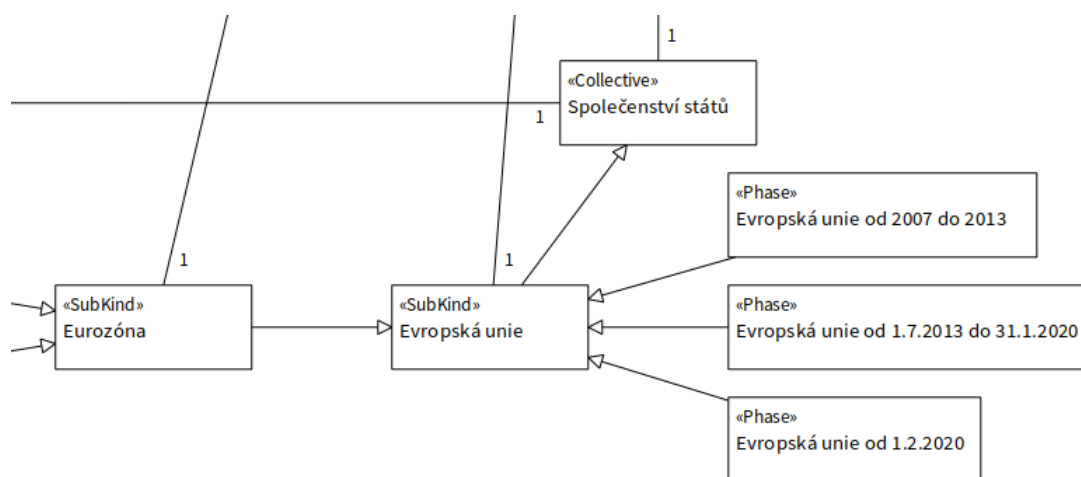
■ **Obrázek A.5** Entita „Motorové vozidlo“ v modelu *Statistika dražeb dle krajů*



■ **Obrázek A.6** Entita „Motorové vozidlo“ v modelu *EU výdaje domácností na konečnou spotřebu podle účelu spotřeby*



■ **Obrázek A.7** Entita „Evropská unie“ v modelu *Počet komerčních letišť*



■ **Obrázek A.8** Entita „Evropská unie“ v modelu *EU osoby*, které nahlásily nehodu vedoucí k úrazu

Bibliografie

1. MYLOPOULOS, John. Conceptual modelling and Telos. *Conceptual modelling, databases, and CASE: An integrated view of information system development* [online]. 1992, s. 49–68 [cit. 2022-11-10]. Dostupné z: https://www.researchgate.net/profile/John-Mylopoulos/publication/242177349_Conceptual_Modelling_and_Telos1/links/5564397508ae9963a11f0a53/Conceptual-Modelling-and-Telos1.pdf.
2. DAOUTIS, Marios; MAVRIDIS, Nikolaos. Towards a model for grounding semantic composition. In: *AISB-Artificial Intelligence and the Simulation of Behaviour 2014* [online]. 2014 [cit. 2022-11-10]. Dostupné z: <https://www.diva-portal.org/smash/get/diva2:711058/FULLTEXT01.pdf>.
3. GUIZZARDI, Giancarlo. Conceptualizations, Modeling Languages, and (Meta) Models. In: *Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference, DB&IS'2006*. IOS Press, 2007, sv. 155. ISBN 9781607502265.
4. LINDLAND, O.I.; SINDRE, G.; SOLVBERG, A. Understanding quality in conceptual modeling. *IEEE Software* [online]. 1994, roč. 11, č. 2, s. 42–49 [cit. 2022-11-10]. Dostupné z DOI: 10.1109/52.268955.
5. WAND, Yair; WEBER, Ron. Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda. *Information Systems Research* [online]. 2002, roč. 13, č. 4, s. 363–376 [cit. 2022-11-10]. ISSN 10477047. Dostupné z DOI: 10.1287/isre.13.4.363.69.
6. FETTKE, Peter. How Conceptual Modeling Is Used. *Communications of the Association for Information Systems* [online]. 2009, roč. 25 [cit. 2022-11-10]. ISSN 15293181. Dostupné z DOI: 10.17705/1CAIS.02543.
7. FAKHROUTDINOV, Kirill. *The Unified Modeling Language* [online]. [B.r.]. [cit. 2022-11-10]. Dostupné z: <https://www.uml-diagrams.org/>.
8. FAKHROUTDINOV, Kirill. *UML 2.5 diagrams overview* [online]. 2013. [cit. 2022-11-11]. Dostupné z: <https://www.uml-diagrams.org/uml-25-diagrams.html>.
9. FAKHROUTDINOV, Kirill. *UML class and Object Diagrams Overview* [online]. 2018. [cit. 2022-11-14]. Dostupné z: <https://www.uml-diagrams.org/class-diagrams-overview.html>.
10. ERIKSSON, Hans-Erik; PENKER, Magnus; LYONS, Brian; FADO, David. *UML 2 toolkit*. John Wiley & Sons, 2003. ISBN 9780764555190.
11. PILONE, Dan; PITMAN, Neil. *UML 2.0 in a nutshell*. 1st ed. Sebastopol: O'Reilly, 2005. ISBN 0596007957.

12. FUENTES-FERNÁNDEZ, Lidia; VALLECILLO-MORENO, Antonio. An introduction to UML profiles. *UML and Model Engineering* [online]. 2004, roč. 2, č. 6-13, s. 72 [cit. 2023-04-28]. Dostupné z: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a41c7070f2c8350ce670e8a322073955cadfb7b3>.
13. ŠMAJS, Josef; KROB, Josef. *Úvod do ontologie: Skriptum filoz. fak. MU*. Masarykova univerzita, 1994. Dostupné také z: https://www.phil.muni.cz/fil/eo/skripta/uvod_do_ontologie.pdf.
14. PETRŽELKA, Josef. Aristotelova metafyzika. *Čítanka antických a středověkých filosofických textů*. 2014, s. 61–82. Dostupné také z: https://digilib.phil.muni.cz/_flysystem/fedora/pdf/131070.pdf.
15. SMITH, Barry. Ontology. In: *The furniture of the world* [online]. Brill, 2012, s. 47–68 [cit. 2022-11-18]. Dostupné z DOI: 10.1163/9789401207799_005.
16. POLI, Roberto. Descriptive, Formal and Formalized Ontologies. In: *Husserl's Logical Investigations Reconsidered* [online]. Springer Netherlands, 2003, s. 183–210 [cit. 2022-11-18]. ISBN 978-90-481-6324-3. Dostupné z DOI: 10.1007/978-94-017-0207-2_12.
17. CARAZZON, Raul. *Ontology: Its Role in Modern Philosophy* [online]. 2022. [cit. 2022-11-18]. Dostupné z: <https://www.ontology.co/>.
18. COCCHIARELLA, Nino B. Formal Ontology and Conceptual Realism. In: *Formal Ontology and Conceptual Realism* [online]. Springer Netherlands, 2007, s. 3–24 [cit. 2022-11-18]. ISBN 978-1-4020-6203-2. Dostupné z DOI: 10.1007/978-1-4020-6204-9_1.
19. GUIZZARDI, Giancarlo. *Ontological Foundations for Structural Conceptual Models* [online]. 2005. [cit. 2022-11-18]. Dostupné z: https://www.researchgate.net/publication/215697579_Ontological_Foundations_for_Structural_Conceptual_Models. Dis. pr.
20. PERGL, Robert. *Conceptualisation: Chapters from Harmonising Enterprise and Software Engineering* [online]. 2018. [cit. 2022-11-10]. Dostupné z DOI: 10.13140/RG.2.2.13338.49609. Dis. pr.
21. GUIZZARDI, Giancarlo; WAGNER, Gerd; ALMEIDA, João Paulo Andrade; GUIZZARDI, Renata S.S. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Applied Ontology* [online]. 2015, roč. 10, č. 3-4, s. 259–271 [cit. 2022-11-10]. ISSN 18758533. Dostupné z DOI: 10.3233/A0-150157.
22. GUIZZARDI, Giancarlo; BENEVIDES, Alessander Botti; FONSECA, Claudenir M.; PORRELLI, Daniele; ALMEIDA, João Paulo A.; SALES, Tiago Prince; BORGIO, Stefano; GALTON, Antony; KUTZ, Oliver. UFO: Unified Foundational Ontology. *Applied Ontology* [online]. 2022, roč. 17, č. 1, s. 167–210 [cit. 2022-11-10]. ISSN 18758533. Dostupné z DOI: 10.3233/A0-210256.
23. GUARINO, Nicola; WELTY, Christopher. Evaluating ontological decisions with OntoClean. *Communications of the ACM* [online]. 2002, roč. 45, č. 2, s. 61–65 [cit. 2022-11-10]. ISSN 00010782. Dostupné z DOI: 10.1145/503124.503150.
24. VACURA, Miroslav. Systém kategorií u Aristotela. *E-LOGOS* [online]. 2009, roč. 2009, č. 1, s. 1–22 [cit. 2022-11-10]. Dostupné z: <https://e-logos.vse.cz/pdfs/elg/2009/01/22.pdf>.
25. GUIZZARDI, Giancarlo; FALBO, Ricardo; GUIZZARDI, Renata. Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In: [online]. 2008, s. 127–140 [cit. 2022-11-10]. Dostupné z: https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/grounding_software_domain_ontologies_in_the_unified_foundational_ontology__ufo___the_case_of_the_ode_software_process_ontology_2008.pdf.

26. SUCHÁNEK, Marek. *OntoUML: Introduction* [online]. 2018. [cit. 2022-11-20]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/intro/ontouml.html>.
27. RYBOLA, Zdeněk; PERGL, Robert. Towards OntoUML for software engineering: Transformation of rigid sortal types into relational databases. In: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)* [online]. 2016, s. 1581–1591 [cit. 2022-11-20]. Dostupné z: <https://ieeexplore.ieee.org/document/7733464>.
28. GUIZZARDI, Giancarlo; FONSECA, Claudenir; BENEVIDES, Alessandro; ALMEIDA, João; PORELLO, Daniele; PRINCE SALES, Tiago. Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0. In: [online]. 2018 [cit. 2022-11-20]. ISBN 9783030008468. Dostupné z DOI: 10.1007/978-3-030-00847-5_12.
29. SUCHÁNEK, Marek. *Identity - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-11]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/theory/identity.html>.
30. SUCHÁNEK, Marek. *Rigidity - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-11]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/theory/rigidity.html>.
31. SUCHÁNEK, Marek. *Mixin - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-11]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/mixin/index.html>.
32. SUCHÁNEK, Marek. *Kind - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-17]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/kind/index.html>.
33. SUCHÁNEK, Marek. *Subkind - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-17]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/subkind/index.html>.
34. GUIZZARDI, Giancarlo; WAGNER, Gerd; GUARINO, Nicola; SINDEREN, Marten van. An Ontologically Well-Founded Profile for UML Conceptual Models. In: PERSSON, Anne; STIRNA, Janis (ed.). *Advanced Information Systems Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, s. 112–126. ISBN 978-3-540-25975-6.
35. SUCHÁNEK, Marek. *Phase - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-18]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/phase/index.html>.
36. SUCHÁNEK, Marek. *Role - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-18]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/role/index.html>.
37. SUCHÁNEK, Marek. *Category - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-19]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/category/index.html>.
38. SUCHÁNEK, Marek. *RoleMixin - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-19]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/rolemixin/index.html>.
39. SUCHÁNEK, Marek. *PhaseMixin - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-19]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/phasemixin/index.html>.
40. SUCHÁNEK, Marek. *Formal - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-27]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/formal/index.html>.

41. SUCHÁNEK, Marek. *Relator - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-28]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/relator/index.html>.
42. SUCHÁNEK, Marek. *Mediation - OntoUML specification documentation* [online]. 2018. [cit. 2023-02-28]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/mediation/index.html>.
43. PERGL, Robert. *OntoUML: Vztahy Celek-Část, typy agregací; Aspekty* [online]. [B.r.]. [cit. 2023-03-05]. Dostupné z: <https://courses.fit.cvut.cz/BI-KOM/slides/pdf/lectures-czech/04-ontouml-aggr-aspects.pdf>.
44. SUCHÁNEK, Marek. *Quantity - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/quantity/index.html>.
45. SUCHÁNEK, Marek. *SubQuantityOf - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/subQuantityOf/index.html>.
46. SUCHÁNEK, Marek. *Containment - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/containment/index.html>.
47. SUCHÁNEK, Marek. *Collective - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/collective/index.html>.
48. SUCHÁNEK, Marek. *MemberOf - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/memberOf/index.html>.
49. SUCHÁNEK, Marek. *SubCollectionOf - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-10]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/subCollectionOf/index.html>.
50. SUCHÁNEK, Marek. *Quality - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-01]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/aspects/quality/index.html>.
51. SUCHÁNEK, Marek. *Mode - OntoUML specification documentation* [online]. 2018. [cit. 2023-03-01]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/aspects/mode/index.html>.
52. DATA FAIRPORT. *Data FAIRport conference: JOINTLY DESIGNING A DATA FAIRPORT* [online]. 2019. [cit. 2023-03-17]. Dostupné z: https://www.datafairport.org/component/content/article/8_news/9_item1/index.html.
53. WILKINSON, Mark D; DUMONTIER, Michel; AALBERSBERG, IJsbrand Jan; APPLETON, Gabrielle; AXTON, Myles; BAAK, Arie; BLOMBERG, Niklas; BOITEN, Jan-Willem; SILVA SANTOS, Luiz Bonino da; BOURNE, Philip E et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* [online]. 2016, roč. 3, č. 1 [cit. 2023-03-17]. Dostupné z DOI: <https://doi.org/10.1038/sdata.2016.18>.
54. GO FAIR. *FAIR Principles* [online]. 2022. [cit. 2023-03-17]. Dostupné z: <https://www.go-fair.org/fair-principles/>.
55. DUNNING, Alastair; DE SMAELE, Madeleine; BÖHMER, Jasmin. Are the FAIR Data Principles fair? *International Journal of Digital Curation* [online]. 2017 [cit. 2023-03-19]. Dostupné z DOI: <https://doi.org/10.2218/ijdc.v12i2.567>.

56. JACOBSEN, Annika; MIRANDA AZEVEDO, Ricardo de; JUTY, Nick; BATISTA, Dominique; COLES, Simon; CORNET, Ronald; COURTOT, Mélanie; CROSAS, Mercè; DUMONTIER, Michel; EVELO, Chris T.; GOBLE, Carole; GUIZZARDI, Giancarlo; HANSEN, Karsten Kryger; HASNAIN, Ali; HETTNE, Kristina; HERINGA, Jaap; HOOFT, Rob W.W.; IMMING, Melanie; JEFFERY, Keith G.; KALIYAPERUMAL, Rajaram; KERSLOOT, Martijn G.; KIRKPATRICK, Christine R.; KUHN, Tobias; LABASTIDA, Ignasi; MAGAGNA, Barbara; MCQUILTON, Peter; MEYERS, Natalie; MONTESANTI, Annalisa; REISEN, Mirjam van; ROCCA-SERRA, Philippe; PERGL, Robert; SANSONE, Susanna-Assunta; SILVA SANTOS, Luiz Olavo Bonino da; SCHNEIDER, Juliane; STRAWN, George; THOMPSON, Mark; WAAGMEESTER, Andra; WEIGEL, Tobias; WILKINSON, Mark D.; WILLIGHAGEN, Egon L.; WITTENBURG, Peter; ROOS, Marco; MONS, Barend; SCHULTES, Erik. FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence* [online]. 2020, roč. 2, č. 1-2, s. 10–29 [cit. 2023-04-28]. Dostupné z DOI: 10.1162/dint_r_00024.
57. GO FAIR. *FAIRification Process* [online]. 2019. [cit. 2023-03-17]. Dostupné z: <https://www.go-fair.org/fair-principles/fairification-process/>.
58. EUZENAT, Jérôme. Towards a principled approach to semantic interoperability. In: *Proc. IJCAI 2001 workshop on ontology and information sharing* [online]. 2001 [cit. 2023-03-24]. Dostupné z: <https://ceur-ws.org/Vol-47/euzenat.pdf>.
59. CELBOVÁ, Ludmila. *Sémantická interoperabilita* [online]. 2003. [cit. 2023-03-24]. Dostupné z: https://aleph.nkp.cz/F/?func=direct&doc_number=000000555&local_base=KTD.
60. LE FRANC, Yann; PARLAND-VON ESSEN, Jessica; BONINO, Luiz; LEHVÄSLAIHO, Heikki; COEN, Gerard; STAIGER, Christine. *D2.2 FAIR Semantics: First recommendations* [online]. FAIRsFAIR, 2020. Ver. 1.0 [cit. 2023-04-28]. Dostupné z DOI: 10.5281/zenodo.5361930.
61. OPENEST. *Inkubátor pro datové firmy* [online]. 2023. [cit. 2023-03-19]. Dostupné z: <https://www.openest.cz/>.
62. BRITISH BUSINESS BANK. *What is a business incubator?* [online]. 2022. [cit. 2023-03-19]. Dostupné z: <https://www.british-business-bank.co.uk/finance-hub/what-is-a-business-incubator/>.
63. REMMARK. *O nás* [online]. 2021. [cit. 2023-03-19]. Dostupné z: <http://www.remark.cz/>.
64. FIT, CTU. *OpenPonk modeling platform* [online]. 2022. [cit. 2023-03-26]. Dostupné z: <https://openponk.org/>.
65. PHARO. *About Pharo* [online]. [B.r.]. [cit. 2023-03-26]. Dostupné z: <https://pharo.org/about>.
66. CCMi FIT, CTU. *OpenPonk* [online]. 2020. [cit. 2023-03-26]. Dostupné z: <https://ccmi.fit.cvut.cz/nastroje/openponk/>.
67. MAREK, Bělohoubek. *Verifikace OntoUML modelů na platformě OpenPonk* [online]. 2019. [cit. 2023-03-26]. Dostupné z: <https://dspace.cvut.cz/handle/10467/83191>. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství.
68. MAREK, Bělohoubek. *Rozšířené možnosti modelování v OntoUML na platformě OpenPonk* [online]. 2021. [cit. 2023-03-26]. Dostupné z: <https://dspace.cvut.cz/handle/10467/94590>. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství.

Obsah přiloženého média

readme.txt	stručný popis obsahu média
modely	konceptuální modely a mapovací pravidla
├─ pocet-duchodcu-s-exekucni-srazkou-podle-kraju		
├─ pocet-komercnich-letist		
├─ pohyb-obyvatel-za-cr-kraje-okresy-so-orp-a-obce-rok-2019		
├─ registr-poskytovatelu-zdravotnich-sluzeb		
├─ seznam-bank-a-pobocek-zahranicnich-bank-na-uzemi-cr		
├─ statistika-drazeb-dle-kraju		
sablony	šablony
├─ uzemi_adresa		
text	text práce
├─ BP_Hampejs_Ondrej_2023.pdf	text práce ve formátu PDF
├─ BP_Hampejs_Ondrej_2023.zip	zdrojová forma práce ve formátu L ^A T _E X