



## Zadání bakalářské práce

<b>Název:</b>	Využití ontologické analýzy pro zajištění sémantické interoperability heterogenních dat
<b>Student:</b>	Jan Pecka
<b>Vedoucí:</b>	doc. Ing. Robert Pergl, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Informační systémy a management
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	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**

**Jan Pecka**

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

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2023 Jan Pecka. 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: Pecka Jan. *Využití ontologické analýzy pro zajištění sémantické interoperability heterogenních dat*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

## Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratk	ix
Úvod	1
1 Cíl práce	3
<b>I Teoretická příprava</b>	<b>5</b>
<b>2 Projekt OpeNest a sémantická interoperabilita dat</b>	<b>7</b>
2.1 Projekt OpeNest	7
2.2 Sémantická interoperabilita dat	7
2.2.1 Sémantická interoperabilita a její úskalí	7
2.2.2 Sémantická interoperabilita v OntoUML	8
2.2.3 Využití sémantické interoperability v projektu OpeNest	8
<b>3 Principy FAIR</b>	<b>9</b>
3.1 GO FAIR Foundation	9
3.1.1 FAIR principy	9
3.1.2 Proces FAIRifikace	10
<b>4 Ontologie</b>	<b>13</b>
4.1 Historie ontologie	13
4.2 Druhy ontologie	13
4.2.1 Deskriptivní, formální a formalizovaná ontologie	13
4.3 Ontologie v informačních technologiích	14
4.3.1 Výhody ontologického přístupu v konceptualizaci dat	14
4.3.2 Unified Foundational Ontology	14
<b>5 Unified Modeling Language</b>	<b>15</b>
5.1 UML Class diagram	15
5.1.1 Vztahy v Class diagramu	15
<b>6 OntoUML</b>	<b>17</b>
6.1 Stavební kámen OntoUML	17
6.2 Princip identity	18
6.3 Rigidita	18
6.4 Sortály	18
6.4.1 Rigidní sortály	18

6.4.2	Anti-rigidní sortály . . . . .	19
6.4.3	Non-sortály . . . . .	19
6.5	Vazby . . . . .	19
6.5.1	Generalizace . . . . .	20
6.5.2	Asociace . . . . .	20
6.6	Vztah celek-část . . . . .	21
6.6.1	Sdílitelnost . . . . .	21
6.6.2	Povinnost . . . . .	21
6.6.3	Funkční celek . . . . .	23
6.6.4	Stereotypy celku-části . . . . .	23
6.7	Aspekty . . . . .	23
<b>7</b>	<b>OpenPonk</b>	<b>25</b>
<b>II</b>	<b>Vlastní řešení</b>	<b>27</b>
<b>8</b>	<b>Klíčové domény</b>	<b>29</b>
<b>9</b>	<b>Modelování</b>	<b>31</b>
9.1	„Blitzkrieg“ modelování . . . . .	31
9.2	Šablony . . . . .	31
9.2.1	Tvorba šablon . . . . .	31
9.3	Tvorba modelů . . . . .	32
9.3.1	Analýza klíčové domény . . . . .	32
9.3.2	Vytváření konceptuálního modelu . . . . .	32
9.4	Praktická ukázka na modelu . . . . .	34
9.4.1	Dřevěné výrobky . . . . .	34
9.4.2	Státy Evropské unie . . . . .	35
9.4.3	Mezinárodní obchod a referenční období . . . . .	36
9.4.4	Spojení modelu . . . . .	36
<b>10</b>	<b>Mapování dat</b>	<b>37</b>
10.1	Datová entita . . . . .	37
10.1.1	Atributy datových entit . . . . .	37
10.2	Mapování . . . . .	38
10.2.1	Základní mapovací pravidla . . . . .	38
10.2.2	Základní mapovací pravidla s podmínkou . . . . .	38
10.2.3	Podmíněná mapovací pravidla . . . . .	39
10.2.4	MEA pravidla mapování . . . . .	39
10.2.5	Zajištění sémantické interoperability . . . . .	40
	<b>Závěr</b>	<b>41</b>
	<b>A Ukázky sémantické interoperability</b>	<b>43</b>
	<b>Obsah přiloženého média</b>	<b>53</b>

## Seznam obrázků

3.1	Diagram procesu FAIRifikace[7] . . . . .	10
6.1	Taxonomie UFO[24] . . . . .	17
6.2	Příklad generalizace <i>disjoint</i> a <i>complete</i> . . . . .	20
6.3	Příklad násobnosti u asociace . . . . .	20
6.4	Příklad materiální relace . . . . .	21
6.5	Příklad celku-části v UML . . . . .	22
6.6	Příklad vztahu « <i>Characterization</i> » . . . . .	23
9.1	Entity podle datové platformy . . . . .	33
9.2	Vytyčené entity podle popisu datové sady . . . . .	34
9.3	Namodelovaná struktura dřevěných výrobků . . . . .	35
9.4	Namodelovaná struktura dřevěných výrobků . . . . .	36
10.1	Namodelovaná struktura dřevěných výrobků . . . . .	38
A.1	Entita státu v modelu eu-vyroba-a-obchod-s-primarnimi-produkty-bunicina-papir-a-lepenka . . . . .	43
A.2	Entita domácnosti v modelu eu-domacnosti-typ-pripojeni-internetu . . . . .	44
A.3	Entita domácnosti v modelu eu-osoby-ktere-si-nemohou-dovolit-osobni-automobil . . . . .	45
A.4	Entita osobního automobilu v modelu eu-osoby-ktere-si-nemohou-dovolit-osobni-automobil . . . . .	46
A.5	Entita osobního automobilu v modelu eu-registrace-aut-podle-pohonu . . . . .	46
A.6	Entita osoby v modelu prehled-podnikatelu-a-zivnosti . . . . .	47

## Seznam výpisů kódu

10.1	Ukázka základního mapovacího pravidla . . . . .	38
10.2	Praktická ukázka základního pravidla . . . . .	38
10.3	Ukázka základního mapovacího pravidla s podmínkou . . . . .	39
10.4	Praktická ukázka základního pravidla s podmínkou . . . . .	39
10.5	Ukázka podmíněného pravidla . . . . .	39
10.6	Ukázka podmíněného pravidla s logickými spojkami . . . . .	39
10.7	Ukázka podmíněného pravidla s logickými spojkami . . . . .	39
10.8	Praktická ukázka základního pravidla . . . . .	40

*Chtěl bych poděkovat především vedoucímu práce doc. Ing. Robertu Perglovi Ph.D. za poskytnutí tématu práce a podporu při její tvorbě. Dále bych rád poděkoval Ing. Tereze Macháčové a Bc. Janě Martínkové za skvělé vedení modelovacího týmu a předání znalostí a zkušeností. Také bych rád poděkoval rodině za podporu při studiu.*



## 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 užit. Tyto osoby jsou oprávněny Dílo užit jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 6. května 2023

.....

## Abstrakt

Obsahem této bakalářské práce je ontologická analýza a její využití pro zajištění sémantické interoperability mezi heterogenními klíčovými doménami. Práce je součástí projektu OpeNest a jeho datového inkubátoru od společnosti Remmark ,a.s., které obohacuje o datové sady se zajištěnou sémantickou interoperabilitou. Klíčové domény jsou zachycené v ontologických konceptuálních modelech vytvořených v jazyce OntoUML. Tyto modely jsou rozšířené o datové entity a mapovací pravidla dat, aby bylo možné data mezi doménami propojovat a provádět nad nimi různé operace. Došlo k vytvoření 5 modelů se 2 šablonami. Modely obsahují 554 entit a 18 datových entit s 662 mapovacími pravidly. Šablony zachycují často používanou problematiku, a napomáhají tak dalšímu modelování. Výsledkem je analýza splnění sémantické interoperability a porovnání s modely, u nichž nebyla provedena důkladná ontologická analýza.

**Klíčová slova** sémantická interoperabilita, ontologická analýza, konceptuální model, datový inkubátor, heterogenní data, FAIR data, UFO, OpeNest, OntoUML, OpenPonk

## Abstract

The aim of this bachelor thesis is ontology analysis and its use to ensure semantic interoperability between heterogeneous key domains. The work is part of the OpeNest project and its data incubator from Remmark Inc. and helps enhancing it with datasets with assured semantic interoperability. Key domains are captured in ontological conceptual models created in OntoUML. These models are extended with data entities and data mapping rules to allow data to be linked between domains and to perform various operations on them. 5 models with 2 templates were created. The models contain 554 entities and 18 data entities with 662 mapping rules. The templates help further modeling as they capture frequently used issues. The result is an analysis of the fulfillment of semantic interoperability and a comparison with models for which no thorough ontological analysis has been performed.

**Keywords** semantic interoperability, ontological analysis, conceptual model, data incubator, heterogeneous data, FAIR data, UFO, OpeNest, OntoUML, OpenPonk

## Seznam zkratk

BORM	Business Objects Relation Modelling
BPMN	Business Process Model and Notation
ČSU	Český statistický úřad
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
EU	Evropská unie
EBHC	Evidence-based health care
FAIR	Findability, Accessibility, Interoperability, and Reuse of digital assets
GFF	Go FAIR Foundation
GFO	General Formalized Ontology
MEA	Measurement Ontology
UFO	Unified Foundational Ontology
UML	Unified Modeling Language



# Úvod

V dnešní době informačních technologií rapidně narůstá počet nasbíraných dat. S množstvím dat narůstá riziko kolizí mezi daty a tím klesá efektivita ve vyhledávání mezi daty a také kvalita výsledných informací. Proto je důležité provádět analýzu dat, aby se zvýšila kvalita informací, které z dat získáme.

Bakalářská práce je určena pro projekt OpeNest od společnosti Remmark, a.s., jehož vizí je osvěta ohledně zpracování dat a následně vytvoření datové platformy, jež sjednotí data na jedno místo. Umožní jednoduché vyhledávání v nich a tím pádem snadnou práci s nimi. Cíl bude dosažen za pomoci dodržování principů FAIR a dodržení sémantické interoperability dat. Díky dodržení těchto zásad, bude snadné projekt udržovat a nadále ho rozšiřovat dalšími datovými sadami. Součástí projektu OpeNest je datový inkubátor, jenž poskytuje již zmiňované datové sady, jak volně dostupné i placené, k využití.

Hlavním motivačním faktorem je odstranění nejednoznačnosti a nesouladu významu dat, který vzniká použitím různých modelů dat a terminologií v různých datových sadách. Použitím společné ontologie je možné sjednotit význam dat a vytvořit společné pochopení, což umožňuje hladký přenos a integraci ontologických modelů. To zlepšuje přesnost a efektivitu analýzy dat a tvorby rozhodnutí a zajišťuje, že data zůstanou užitečná a hodnotná v průběhu času, i když se zdroje dat a systémy vyvíjejí. Celkově hraje ontologická analýza klíčovou roli při umožňování interoperability dat a umožňuje organizacím plně využít výhody svých dat.

Tato práce je primárně zaměřena, na ontologickou analýzu klíčových domén a jejím sémantickým propojením s ostatními již zpracovanými datovými sadami. Tím se zajistí sémantická interoperabilita mezi sadami, tudíž se mitigují kolize mezi daty. Ontologický model se skládá z entit, které jsou následkem analýzy. Nadále se entity propojí mezi sebou tak, aby daný model co nejlépe popisoval skutečnost. Dalším krokem je popsání obsahu sad mapovacími pravidly, které jsou vázané k datovým entitám a ty jsou napojené k nim příslušícím entitám. Nakonec je ontologický model vložen na datovou platformu a propojen skrze jména entit s ostatními modely.





## Kapitola 1

# Cíl práce

Hlavním cílem bakalářské práce je ontologická analýza klíčových domén z různých oborů a jejich propojení s datovými sadami, aby byla umožněna jejich sémantická interoperabilita. Vše je součástí projektu OpeNest uskutečňovaným marketingovou společností Remmark, a.s.

Prvotním cílem je analýza již transformovaných datových sad a jejich souvisejících datových sad, které dodala společnost Remmark, a.s. a jejich rozbor z pohledu ontologie. Rozbor datových sad je dále použit jako báze ke stavbě konceptuálních modelů.

Další částí cíle je vytvoření ontologických konceptuálních modelů ke každé jednotlivé transformované datové sadě. Konceptuální model vyjadřuje výsledky ontologické analýzy.

Finálním cílem je vytvoření propojení sémantiky ontologických modelů a datových sad s využitím mapovacích pravidel spojených na jednotlivé datové entity v ontologických modelech. Samotné mapování závisí na transformaci datových sad, ale samotná tvorba transformací je mimo rozsah této bakalářské práce.

Projekt OpeNest, má za cíl vytvořit inkubátor firem. Zároveň zachování principů FAIR, které popisují efektivní publikování dat, dodržování definic sémantické interoperability a využívání ontologického konceptuálního modelování. Bakalářská práce je přínosem ke ovlivnění vývoje nástroje OpenPonk, jenž je využit ke tvorbě ontologických konceptuálních modelů a mapování.





Část I

**Teoretická příprava**



# Projekt OpeNest a sémantická interoperabilita dat

## 2.1 Projekt OpeNest

Projekt OpeNest, který pochází z dílny české full service komunikační agentury Remmark, a.s., má za úkol vybudovat výjimečný inkubátor v srdci Prahy a následně jej provozovat. Cílem tohoto projektu je vytvořit administrativní objekt, který podporuje podnikání a zavádí inovační metody v regionálním podnikatelském měřítku. Projekt OpeNest byl již roku 2020 podpořen z Evropských strukturálních fondů prostřednictvím Operačního programu Praha – pól růstu ČR. Tato finanční podpora byla poskytnuta v rámci 3. výzvy Pražského voucheru na inovační projekty. Celkově je projekt zaměřen na podporu rozvoje inovativního podnikání v Praze a jeho okolí.[1]

## 2.2 Sémantická interoperabilita dat

Interoperabilita mezi komponentami velkých měřítek distribuovaných systémů spočívá ve schopnosti vyměňovat si služby a data v rámci domluvených pravidel mezi žadatelem a poskytovatelem. K dosažení těchto cílů slouží různé nástroje a protokoly, jako jsou protokoly přenosu zpráv, chybových hlášek a specifikací typů argumentů.

Výměna dat jako takových však sama o sobě nezaručuje interoperabilitu. Je nutné zajistit, aby měla smysl i v kontextu použití. Proto je nezbytné, aby bylo dosaženo vzájemného porozumění, co se významu poskytovaných služeb a dat mezi poskytovatelem a žadatelem týče.

Konkrétním případem se zabývá článek Sandry Heiler, který popisuje aplikaci, jež si žádá informace o zákazníkovi od aplikace, která poskytuje data. Aby bylo předání těchto informací možné bez potíží, musí obě strany dojít ke společnému porozumění tomu, co vlastně znamená „zákazník“. Pouze pak bude zajištěna sémantická interoperabilita mezi aplikacemi.[2]

### 2.2.1 Sémantická interoperabilita a její úskalí

Výše zmiňované sémantické dohody často nebývají dostatečné, má-li dojít k jejich využití k novým účelům. To může být způsobeno tím, že sémantika využívaných procedur a dat není vždy jednoznačně definována. Tím vzniká riziko pro žadatele – nemohou mít důvěru v poskytovatele, neboť ti nemusí vždy plně vyhovět jejich požadavkům a předpokladům významu a výměny dat.

Pro takové situace je nutné klást větší důraz na standardy, jež vedou k tvoření jednotného sémantického modelu. To umožňuje zlepšení přesnosti dat a procedur.

Tyto faktory jsou základem pro zajištění úspěšného vytváření nových aplikací pro velké datové a distribuované systémy. Pokud by nedošlo k zajištění sémantické interoperability, mohlo by se stát, že tvorba nových aplikací bude neúspěšná. Je nezbytné vytvářet jasně definované postupy a standardy, jež podpoří sémantický model výměny dat a procesů.[2]

### 2.2.2 Sémantická interoperabilita v OntoUML

Aby bylo možné dosáhnout sémantické interoperability, je nutné zajistit ontologickou konzistenci a jasně vymezit struktury. Pro tyto účely slouží obor ontologie, který využívá jazyk OntoUML, jehož podoba je přesněji popsána níže v této práci (viz 6). OntoUML je silně podložený ontologií, vychází z Unified Modeling Language (viz 5) a jeho návrh je spjat s vizí na naplnění požadavků Unified Foundation Ontology (viz 4.3.2). Modelovací primitiva OntoUML jsou navržena tak, aby co nejlépe reflektovala ontologická rozlišení, jež prezentuje UFO. Gramatika tohoto jazyka odráží formální omezení axiomatizace UFO. Bez těchto struktur je dosažení sémantické interoperability obtížné, a tím se zvyšuje riziko chybné interpretace dat. Ontologie v tomto případě poskytuje klíčovou podporu pro sémantickou interoperabilitu.[3]

OntoUML představuje díky svým ontologicky konzistentním strukturám, jež přináší, efektivní prostředek pro zajištění sémantické interoperability. V tomto kontextu je vhodné zmínit empirické důkazy, které ukazují výhody ontologicky vedeného konceptuálního modelování nad tradičním přístupem, a sice že tato metoda umožňuje tvorbu přesnějších konceptuálních modelů, a tedy jednodušší řešení problémů s komplexními a nejednoznačnými doménami. Díky OntoUML je možno dosáhnout vyšší přesnosti a spolehlivosti v konceptuálním modelování.[4]

### 2.2.3 Využití sémantické interoperability v projektu OpeNest

V rámci projektu OpeNest dochází ke zpracování datových sad z různých zdrojů, které používají různé standardy. To má za důsledek, že jedna a ta samá skutečnost může být popsána různými termíny, což způsobuje nevyhnutelné kolize mezi popisovanými skutečnostmi. Aby bylo možné předejít těmto kolizím, je nezbytné dodržovat pravidla sémantické interoperability. Tím je zajištěna jednoznačnost popisovaných skutečností bez ohledu na zdroj, ze kterého pochází.

Projekt OpeNest společnosti Remmark, a.s., klade důraz na jednotný sémantický model, jenž se skládá z různých menších modelů, které jsou vytvořeny na základě jim příslušících datových sad. Sémantická interoperabilita mezi těmito modely je kontrolována datovou platformou, která s modely pracuje. Nedodržení pravidel sémantické interoperability by mohlo vést k výskytu kolizí mezi jednotlivými modely, čímž by jejich spojení do jednoho celku nebylo možné. Projekt OpeNest by tak nemohl být realizován také z důvodu nemožnosti efektivně vyhledávat mezi daty datových sad.

# Principy FAIR

### 3.1 GO FAIR Foundation

Samosprávná iniciativa GO FAIR Foundation (GFF) se podílí na zastřešování FAIR principů. Tento rámec byl vytvořen s cílem zajistit spravedlivé, ověřené a efektivní využívání dat. GFF byla založena v roce 2018 jako separátní entita nizozemského zákona s úkolem podporovat a vytvářet principy FAIR. Zároveň spolupracuje s klíčovými partnery na celosvětové úrovni, aby zajistila adopci FAIR principů ve všech oblastech, a to i v akademickém výzkumu, průmyslu a vládních organizacích.[5]

V dnešním světě se vlivem digitální revoluce zvyšuje objem dat, která je nutné efektivně zpracovávat strojově. Aby bylo možné usnadnit tento složitý proces a zajistit opakované využití dat, akademici, průmyslové subjekty, finanční agentury a vědecká vydavatelství společně podpořili stručný a měřitelný soubor principů známý jako FAIR Data Principles. Tyto principy slouží jako instrukce pro subjekty, které chtějí zlepšit znovupoužitelnost svých datových úložišť. Cíl těchto principů je poskytnout řád a usnadnit proces opakovaného využití pro celosvětové subjekty, jež usilují o zlepšení kvality a efektivitu datového zpracování.[6]

#### 3.1.1 FAIR principy

FAIR je akronymem pro *F*indable, *A*ccesible, *I*nteroperable, *R*eusable. Jednotlivá pravidla mají tyto významy:

##### 1. Findable:

V kontextu datového managementu se vyhledatelnost vztahuje ke schopnosti přiřazení globálního a unikátního identifikátoru k (meta)datům. Takový identifikátor je také perzistentní v čase. Data mají být bohatě popsána metadaty a metadata mají být jasně definována a obsahovat identifikátor dat, která popisují, aby se zajistilo, že jsou data srozumitelná a lehce naležitelná.[6]

##### 2. Accesible:

Pod přístupností dat se rozumí získání (meta)dat pomocí standardizovaných komunikačních protokolů a identifikátorů. Tyto protokoly by měly být volně dostupné, jednoduše implementovatelné a veškeré relevantní informace by měly být zaznamenány a indexovány v prohlédavatelném zdroji. Protokoly by zároveň měly být vybaveny autentizačními a autorizačními postupy, aby došlo k zabezpečení dat. Nakonec by měla být zajištěna dostupnost metadat, i když příslušná data nejsou nadále přístupná.[6]

### 3. Interoperable:

Interoperabilita se týká schopnosti (meta)dat využívat jazyk, který je formální, přístupný a obecně použitelný pro reprezentaci znalostí. Slovníky použité pro (meta)data by současně měly být v souladu s FAIR principy. Kromě toho by měla být zajištěna kvalifikovaná reference na další (meta)data, což pomáhá k ucelnému a komplexnímu chápání datových souborů.[6]

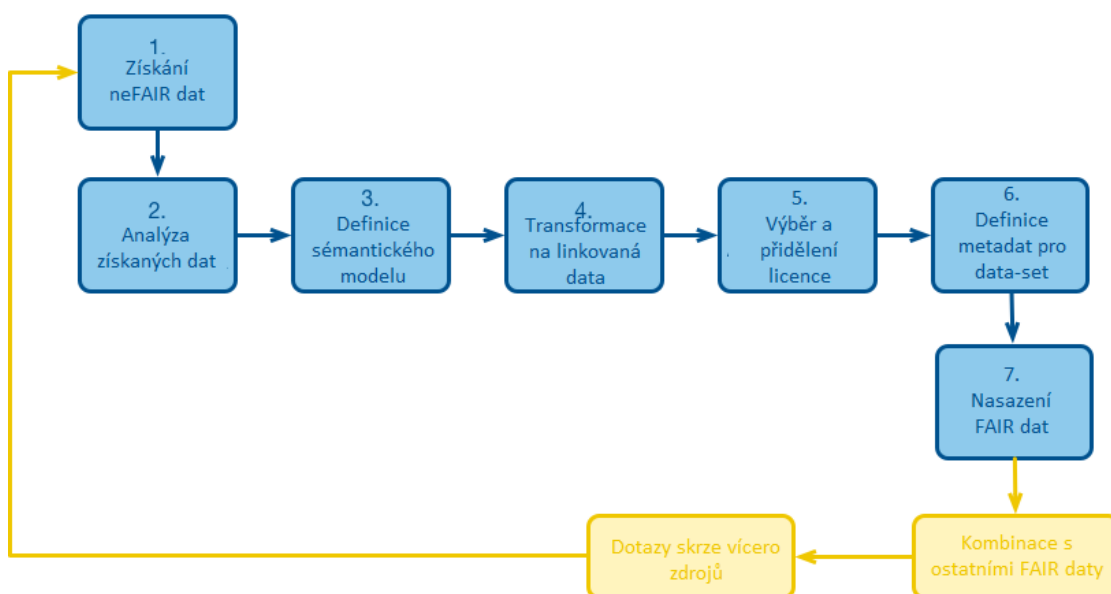
### 4. Reusable:

Znovupoužitelnost (meta)dat je kritická pro jejich širší využití. Aby bylo možné dosáhnout tohoto cíle, je nutné, aby (meta)data byla pečlivě popsána pomocí přesných a relevantních vlastností. Navíc by (meta)data měla být uvolňována s jasnými a dostupnými licencemi, které uživatelům umožní bezproblémové použití již zmiňovaných (meta)dat. Také je důležité, aby byl podrobně popsán jejich původ a aby byla souladu s normami a standardy relevantních doménových komunit.[6]

Pokud (meta)data splňují všechny výše zmíněné principy, práce s nimi a jejich případné rozšiřování by mělo být jednodušší. V další sekci bude popsán proces FAIRifikace, který vede ke splnění těchto bodů.

## 3.1.2 Proces FAIRifikace

Aby o datech bylo možné prohlásit, že naplňují principy FAIR, je nutné, aby prošla procesem takzvané FAIRifikace (viz 3.1). Tento proces se skládá z níže popsanych sedmi bodů.



■ **Obrázek 3.1** Diagram procesu FAIRifikace[7]

Popsané kroky FAIRifikace:

- 1. Získání neFAIR dat:** Aby mohl být proces FAIRifikace započat, je nezbytné získání dat vhodných pro FAIRifikaci.[7]
- 2. Analýza získaných dat:** Další na řadě je důkladná analýza získaných dat, jejímž cílem je zachycení struktury analyzovaných dat.[7] V kontextu této práce jde o ontologickou analýzu, jež zjišťuje význam a strukturu obsahu datové sady, aby mohl být vytvořen ontologický konceptuální model.

3. **Definice sémantického modelu:** Dalším krokem je definice sémantického modelu datové sady. Model je vytvořen na základě předchozí analýzy a má za úkol popsat význam a vztah popisovaných dat.[7]
4. **Transformace na linkovaná data:** Vytvořený sémantický model se aplikuje, a tím dojde k vytvoření linkovaných dat. Tento krok je možné vynechat na základě výstupů analýzy, jež ukáží, že pro tento typ dat není transformace vhodná.[7]
5. **Výběr a přidělení licence:** Výběr licence pro data určuje způsob, jakým mohou být nadále používána. Tento krok je zásadní pro data, u nichž se počítá s komerčním využitím.[7]
6. **Definice metadat pro data-set:** Metadata se k datům přiřazují z důvodu ulehčení práce s nimi. Data-set s dobře definovanými metadaty je lépe čitelný, metadata také ulehčují vyhledávání v něm a popisují jeho obsah.[7]
7. **Nasazení FAIR dat:** Po procesu FAIRifikace jsou data připravena k nasazení. Jejich dostupnost pak závisí na licenci, pod kterou byla vydána.[7]





# Ontologie

Ontologie je disciplína filozofie, která se zabývá zkoumáním bytí. Tato oblast se zabývá identifikací druhů a struktur objektů, jejich vlastností a vztahů s událostmi, procesy a jinými prvky v různých oblastech reality. Ve filozofických kruzích se pro ontologii často používá synonymum metafyzika, které odkazuje na oblast za hranicemi fyzického světa, jak již naznačuje jeho etymologie (to, co přichází po fyzice). Historicky byl termín metafyzika používán Aristotelovými žáky k označení oblasti, jíž se zabývá fyzika.[8]

## 4.1 Historie ontologie

Pojem ontologie byl zaveden nezávisle na sobě dvěma filozofy, Rudolfem Göckelem a Jacobem Lorhardem, v roce 1613, nicméně samotný koncept metafyzické disciplíny sahá až k Aristotelovi a jeho žákům. V anglicky mluvících zemích byl termín ontologie poprvé zaznamenán v Baillyho slovníku z roku 1712, jak uvádí Oxfordský slovník angličtiny.[8]

Ontologie je disciplínou, jež se snaží poskytnout konečnou a důslednou klasifikaci všech oblastí existence. Konečná klasifikace má za úkol odpovědět na otázky, jako například jaké třídy entit jsou potřebné pro úplné fungování vesmíru. Tato klasifikace by měla zahrnovat všechny možné vztahy mezi entitami a jejich kategorie. Nicméně rozdělení těchto kategorií se může lišit v závislosti na filozofických školách.[8]

## 4.2 Druhy ontologie

Roberto Poli, italský profesor, identifikoval tři typy ontologií: deskriptivní ontologii, formální ontologii a formalizovanou ontologii. Tyto ontologie se dále dělí na dvě kategorie, a to na doménově závislé a doménově nezávislé.[9]

Doménově závislá ontologie se liší od doménově nezávislé ontologie tím, že se zabývá uzavřenou kategorií entit. Tento typ ontologie klade důraz na vytvoření pevné metodologie, kde to, co je predikováno, musí být pravdivé pro třídu položek, pro kterou je to predikováno. Jinak řečeno doménově závislá ontologie se zaměřuje na specifickou oblast bytí.[9]

### 4.2.1 Deskriptivní, formální a formalizovaná ontologie

Deskriptivní ontologie může být chápána jako obecná sada informací, která se vztahuje k určité oblasti analýzy domén.[9] Formální ontologie se věnuje organizaci výsledků deskriptivní ontologie. Edmund Husserl ve své knize *Logical Investigations* charakterizoval formální ontologii jako oblast,

kteřá se zabývá kategoriemi, jako jsou věc, proces, hmota, celek, část a číslo[10]. Tyto kategorie mají zpočátku malou souvislost s použitím konkrétního formalizmu. Formalizovaná ontologie se pak snaží nalézt vhodnou formální kodifikaci pro konstrukty získané deskriptivní a formální ontologií, které jsou následně formálně očištěny[9]. Zásluhou této formalizace může být dosažena sémantická interoperabilita a dodržením jejích pravidel může být umožněno propojení více domén (viz 3).

### 4.3 Ontologie v informačních technologiích

George H. Mealy byl jedním z prvních, kdo hovořil o ontologii v informačních technologiích. Tvrdil, že při manipulaci s daty můžeme rozlišovat tři odlišné roviny. První rovinou je samotný reálný svět, zatímco druhá rovina se týká představ o něm a poslední třetí rovina se zabývá modely první sféry. Data jsou jenom malými fragmenty reálného světa, a proto zpracování dat zahrnuje přesuny mezi těmito vrstvami.[11]

#### 4.3.1 Výhody ontologického přístupu v konceptualizaci dat

V rámci tradičního přístupu modelovacích systémů se modelář snaží formalizovat uživatelův pohled na danou doménu do konceptuálního modelu. Nicméně když se konceptualizace modeláře neshoduje s konceptualizací uživatele, vznikají problémy. Tyto rozpory mohou být způsobeny mnoha faktory včetně odlišného pochopení konceptů, které chce uživatel popsat. Řešením tohoto problému může být využití ontologií k dosažení vzájemného porozumění. Zkoumání ukazují, že při návrhu a implementaci informačních systémů je vhodné použít obecně platnou formálně definovanou ontologii, která slouží jako referenční rámec pro návrh a použití systému. To přispívá ke zvýšení interoperability mezi různými systémy.[12]

#### 4.3.2 Unified Foundational Ontology

Základní ontologie poskytují uživatelům základní koncepty, na nichž se staví ontologie vázaná na určitou doménu.[13] Unified Foundational Ontology (UFO) patří mezi základní ontologie, které byly vyvinuty v posledních dvou desetiletích Giancarlem Guizzardim a dalšími vědci. UFO vzniklo sloučením dvou jiných ontologií, DOLCE a GFO, a jeho cílem bylo vytvořit ontologii pro konceptuální modelování.[14] OntoUML (viz 6) je postaveno na principech UFO.[15]

##### 4.3.2.1 Kategorie UFO

UFO lze rozdělit do třech kategorií ontologií:

1. **UFO-A:** „což je ontologie *endurantů*“[16]
2. **UFO-B:** „což je ontologie *perdurantů*“[16]
3. **UFO-C:** „což je ontologie *sociálních a intencionálních entit*“[16]

Pojem *endurant* znamená popis strukturálních aspektů ve významu entit, atributů a vztahů. *Perdurant* je potom popis chování *endurantů*, jejich událostí a aspektů.[16]

# Unified Modeling Language

Unified Modeling Language (UML) je formální jazyk souboru notací používaných při návrhu, dokumentaci a implementaci softwarových systémů. UML poskytuje abstraktní a grafickou reprezentaci konceptů a procesů, které jsou důležité pro vývoj software, včetně struktury, chování a interakcí systému. Tento jazyk se stal standardem v oboru softwarového inženýrství a je využíván při analýze a návrhu softwarových architektur, což usnadňuje komunikaci mezi vývojáři a zajišťuje konzistenci a správnost výsledného produktu.[17]

UML představuje soubor notací, které slouží k zaznamenání a dokumentaci různých aspektů softwarového systému. Tento jazyk se skládá z několika druhů diagramů, které lze obecně rozdělit do dvou základních skupin. První skupinou jsou strukturální modely, které se zaměřují na reprezentaci struktur objektů v projektu, včetně jejich atributů, omezení a vztahů s ostatními objekty. Tyto diagramy poskytují detailní pohled na vnitřní strukturu systému a umožňují lepší pochopení jeho vazeb.[18]

Druhou skupinou jsou behaviorální modely, které se zaměřují na chování softwarového systému. Tyto diagramy popisují dynamické aspekty systému, včetně jeho stavů, událostí, procesů a toků dat. Zahrnují také způsob, jakým objekty komunikují a vyměňují si informace, včetně zpráv a odpovědí. Tyto diagramy jsou zásadní pro porozumění chování systému a jeho interakce s uživatelem nebo jiným systémem. Souhrnně řečeno UML poskytuje ucelený způsob pro popis a analýzu softwarových systémů.[18]

## 5.1 UML Class diagram

UML poskytuje řadu strukturálních modelů, z nichž jeden z nejvýznamnějších je UML Class diagram, který slouží k zachycení statického modelu softwarového systému. Tyto diagramy se zaměřují na popis jednotlivých objektů (tříd), které představují předpis pro konkrétní objekt v systému a uchovávají informace o jeho názvu, atributech, metodách a funkcích. Mezi třídami existují různé druhy vazeb, které popisují jejich vztahy a interakce. UML Class diagramy jsou stěžejní pro porozumění struktuře softwarového systému.[19]

### 5.1.1 Vztahy v Class diagramu

V kontextu UML Class diagramů lze konstatovat, že vztahy mezi jednotlivými třídami zachycují komunikační toky mezi nimi. UML nabízí tři základní typy vztahů, a to *asociaci*, *agregaci* a *závislost*, které slouží k popisu různých druhů vazeb mezi třídami. Tyto vztahy mají klíčový význam pro porozumění struktuře komunikačního toku.[19]

1. Asociace je obousměrné propojení mezi třídami a ukazuje, že propojené třídy vzájemně vědí o své existenci.[19]
2. Agregace je vztahem mezi celkem a částí. Pro vývojáře značí, že mezi třídami propojenými agregací je silná vazba.[19]
3. Závislost je slabším pojídlem oproti asociaci – nezaručuje že dvě třídy vzájemně ví o své existenci.[19]

Pro vztahy mezi třídami je důležité i určení násobnosti dané vazby (pro asociace a agregace). Násobnost vyjadřuje počet instancí jednotlivých tříd, které se účastní daného vztahu. I když jsou asociace a agregace vazby obousměrné, někdy je vhodné přidat směr vazby. Ten se značí otevřenou šipkou na jednom z konců vazby. Dá se tak vyjádřit fakt, že jedna třída ví o té druhé, ale ne naopak.[19]

Za pomoci dědičnosti můžeme vyjádřit vztah mezi nadtřídou a její podtřídou. Instance podtřídy sdílí veškeré atributy, metody a funkce se svou nadřazenou třídou. Dědičnost nám dovoluje přidávat vlastnosti třídy, aniž bychom jimi zatěžovali její nadtřidu. Tím se stává implementace přehlednější a rozšiřitelnější.[19]

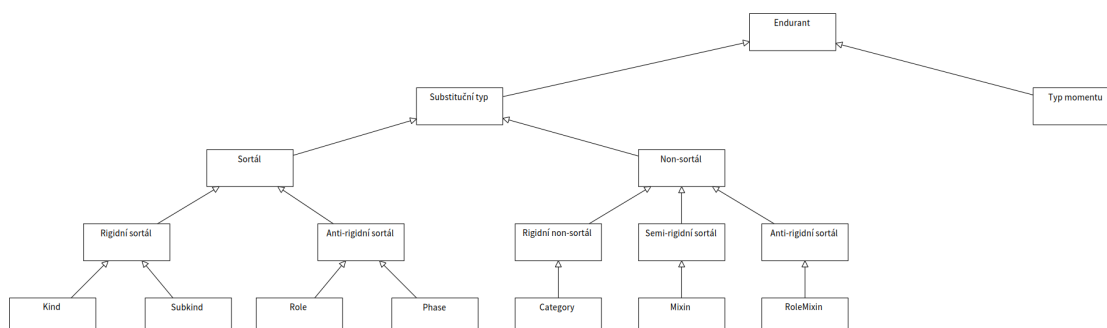
OntoUML je konceptuálním modelovacím jazykem, který byl navržen s cílem umožnit ontologické a grafické zachycení konceptuálních modelů. Je postaven na ontologii UFO (viz 4.3.2), která je, stejně jak OntoUML, vytvořena Guizzardim a Wagnerem. OntoUML platforma používá pro svou notaci diagram tříd UML.[20]

OntoUML dokáže být více přesné v popisování skutečnosti než klasické UML, a to právě díky tomu, že OntoUML vychází z ontologie UFO. Zároveň umožňuje díky své exaktnější notaci rychleji a precizněji konceptualizovat doménu.[21]

OntoUML s UFO se podařilo překonat několik problémů konceptuálního modelování, jako je například problém celku a části nebo problém rolí. Samotné OntoUML se těší praktickému užití v EBHC (medicína založená na důkazech), kde napomáhá při zachycování skutečností z tohoto odvětví.[22]

### 6.1 Stavební kámen OntoUML

Středobodem celého UFO jsou *věci*. Věc je definována jako „*cokoliv vnímatelného nebo myslitelného*“.[13] Věci se dále dělí na sety a entity. Set se dá popsat jako věc, jejímiž členy jsou další věci. Vztah *věc je členem setu* je tvořen tak, aby byl ireflexivní, asymetrický a intranzitivní. Entita je věc, která není setem. Typ je pak entita, která je množinou entit, jež jsou jejími instancemi, a musí mít nějaký smysl použitelnosti neboli záměr.[13] Tyto typy se v OntoUML používají jako třídy, protože navazují na UML Class diagram.[23]



■ Obrázek 6.1 Taxonomie UFO[24]

## 6.2 Princip identity

Před začátkem modelování v OntoUML je dobré si ujasnit, co je vlastně princip identity. Ve zkratce se jedná o druh funkce, která nám umožní od sebe odlišit dvě entity. Identita může být vyjádřena pomocí množin. „Pokud  $A = \{1, 2\}$  a  $B = \{2, 3\}$  potom  $A \neq B$ . Identita je definovaná členy množin.“[25]. Příkladem můžou být dva lidé se stejným jménem – nejsou to stejní lidé a měli bychom je rozlišit pomocí jejich rodného čísla, které je pro každého unikátní.

Možnost dědění identity zajišťuje vztah zvaný generalizace. Jedná se princip dědění podobný tomu v UML, na rozdíl od UML se však v UFO a tedy i v OntoUML dědí nejenom vlastnosti, ale i identita.[21]

## 6.3 Rigidita

Pro další dělení typu entit v hierarchii je využívána rigidita. Ta vyjadřuje vlastnost instance jedince, který je-li rigidní, zůstává svým typem po celou dobu existence instance.[21] Dobrým příkladem je osoba – v rámci své existence nabývá hodnot dítě, dospělý a senior, ale po celou dobu zůstává člověkem. Člověk je rigidní entitou.

Logickou negací pro rigidní typy jsou non-rigidní typy. Jde o takové typy, pro které existuje alespoň jeden jedinec, jehož instance může zaniknout.

Anti-rigidita je speciálním druhem non-rigidity. Pro anti-rigiditu platí, že všechny instance mohou přestat existovat bez toho, aniž by ohrozily existenci identity.

Posledním druhem jsou semi-rigidní typy, tj. takové typy, které kombinují vlastnosti rigidních a anti-rigidních typů. Instance je rigidní ve většině případů, ale v některých sférách existence může zanikat.

## 6.4 Sortály

Entity rozdělujeme podle jejich vztahu k identitě do dvou kategorií. První z nich jsou sortály – takové entity, které mají identitu. Sortály se nadále dělí na rigidní sortály, mezi které patří «*Kind*» a «*Subkind*», «*Relator*», «*Collective*», «*Quantity*», «*Mode*» a «*Quality*», na non-rigidní, kam patří «*Role*» a «*Phase*», a na anti-rigidní a semi-rigidní sortály.

### 6.4.1 Rigidní sortály

#### 1. «*Kind*»

Prvním rigidním sortálem je stereotyp «*Kind*». Mezi jeho hlavní schopnosti patří poskytování identity, což ale zabraňuje tomu, aby dědil od dalšího stereotypu s identitou. Dále nesmí dědit ani od anti-rigidního stereotypu, jelikož sám je rigidní, a tudíž by pak jeho existence porušovala vlastnosti rigidity.[26]

#### 2. «*Subkind*»

Mezi rigidní sortály se řadí také stereotyp «*Subkind*». Na rozdíl od stereotypu «*Kind*» neposkytuje identitu, což ale znamená, že ji musí zdědit od jiného stereotypu. K tomu se používá vazba generalizace.[27]

#### 3. «*Relator*»

Tento stereotyp pečetí materiální vazby a zároveň poskytuje identitu, tudíž nemůže generalizací dědit od dalších rigidních sortálů.[28]

Rigidní sortály «*Mode*» a «*Quality*» patří mezi aspekty, jsou tudíž popsány v sekci 6.7. «*Collective*» a «*Quantity*» jsou popsány u funkčního celku 6.6.3.

## 6.4.2 Anti-rigidní sortály

Do této skupiny spadají dva stereotypy, a to «*Role*» a «*Phase*», která si dále detailněji popíšeme.

### 1. «*Role*»

Stereotyp «*Role*» se používá pro vyjádření zaměření sortálu. Neposkytuje identitu a musí ji získat od jiného sortálu pomocí generalizace. Navíc je stereotypem relačně závislým, což znamená, že její existence musí být podmíněna nějakým pečetidlem. V případě OntoUML jde o «*Relator*» a materiální vazbu.[29]

### 2. «*Phase*»

«*Phase*» je druhým a posledním anti-rigidním stereotypem. Ostatním sortálům pomáhá vyjádřit fáze existence, stejně jako «*Role*» musí získat identitu od sortálu, co ji poskytuje.[30]

## 6.4.3 Non-sortály

Skupina non-sortálů je skupinou, ve které jednotlivé stereotypy mohou mít více identit a pomáhají spojit více takových sortálů podle společných vlastností. Není možné, aby non-sortál dědil od sortálu, protože není schopen přijímat identitu.[21] Do této skupiny spadají tyto stereotypy: «*Category*», «*Mixin*», «*RoleMixin*» a «*PhaseMixin*».

### 1. «*Category*»

«*Category*» je non-sortál, který je rigidní a používá se k vytyčení rigidních sortálů podle sdílených vlastností, ale s různým principem identity. Z jeho rigidity vyplývá, že není možné, aby dědil od anti-rigidního stereotypu.[31]

### 2. «*RoleMixin*»

«*RoleMixin*» sdílí vlastnosti se stereotypem «*Role*», tj. je anti-rigidním non-sortálem, ale je využíván pro stereotyp «*Role*» s rozdílným principem identity.[32]

### 3. «*PhaseMixin*»

«*PhaseMixin*» je stejně jako «*RoleMixin*» anti-rigidní non-sortál a je používán ke spojování stereotypů «*Phase*» s odlišnými principy identity.[33]

### 4. «*Mixin*»

«*Mixin*» je jedinečný v tom, že se jedná o semi-rigidní non-sortál. Pojem semi-rigidní je vysvětlen výše (viz 6.3). Spojuje rigidní a anti-rigidní stereotypy, které mají jiné principy identity, podle jejich společných vlastností. Nadtypem stereotypu «*Mixin*» může být jediné další «*Mixin*», protože jeho semi-rigidita neumožňuje dědění od rigidních ani anti-rigidních stereotypů.[34]

## 6.5 Vazby

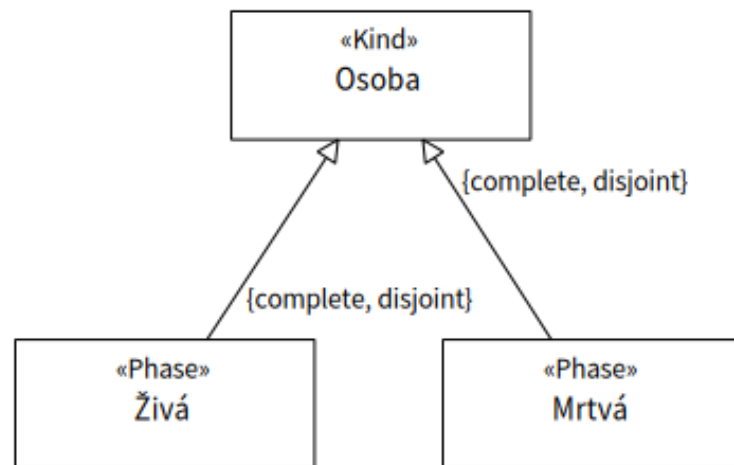
Jednotlivé entity mají mezi sebou vztahy, které je nutné nějakým způsobem popsat. K tomu se využívají vazby neboli relace. Dvě rozdílné entity mezi sebou mohou mít jen jednu vazbu, ale jedna entita může mít neomezeně mnoho entit, se kterými má vazbu.[35] V OntoUML jsou dva základní typy vazeb, a to generalizace a asociace, která se dále dělí.

### 6.5.1 Generalizace

Generalizace se využívá pro dědění vlastností identity, jak bylo zmíněno výše. Lze ji upřesnit pomocí meta-atributů, které má dva – *complete* a *disjoint*. [23]

*Complete* vyjadřuje skutečnost, že každá instance nadtypu je zároveň instancí alespoň jednoho z jejích podtypů. Ekvivalentem pro *complete* v UML je *isCovering = True*. [23]

*Disjoint* znamená, že každá instance nadtypu může být instancí maximálně jednoho z podtypů. Kombinace *complete* a *disjoint* říká, že instance nadtypu je instancí právě jednoho podtypu. [23] Na následujícím obrázku je znázorněn příklad syntaxe generalizace s meta-atributy *complete* a *disjoint*. Zachycuje skutečnost, že osoba může být buď živá, anebo mrtvá.



■ Obrázek 6.2 Příklad generalizace *disjoint* a *complete*

### 6.5.2 Asociace

Druhým typem vazeb je asociace. Ta se v OntoUML rozděluje na formální a materiální. [23] Pro asociaci oproti generalizaci přibývá název a násobnost. Násobnost vyjadřuje, kolik instancí třídy B připadá na instanci třídy A. V OntoUML je definováno několik předpisů pro násobnost. Jde o *0..1*, *1..1*, *0..\** a *1..\**. Násobnost *0..1* nám říká, že na instanci třídy A připadá 0 až 1 instance třídy B. Násobnost *0..\** je podobná předchozí násobnosti, akorát za *\** můžeme dosadit libovolné číslo z  $N$ . Tento typ násobností vyjadřuje nepovinnou vazbu, což znamená, že takové instance mohou existovat jedna bez druhé. Druhý typ je vztah povinný, kde *1..1* a *1..\** vyjadřují nutnost existence alespoň jedné instance dané třídy, a aby celý vztah mohl existovat, stejně jako u *0..\**, tak u *1..\** se *\** nahrazuje za libovolné číslo z  $N$ . Následující příklad ukazuje vztah osoby a automobilu. Násobnosti nám říkají, že automobil může být vlastněn 0 až jednou osobou, jde tedy o nepovinnou vazbu a osoba může vlastnit 0 až  $n$  automobilů.



■ Obrázek 6.3 Příklad násobnosti u asociace

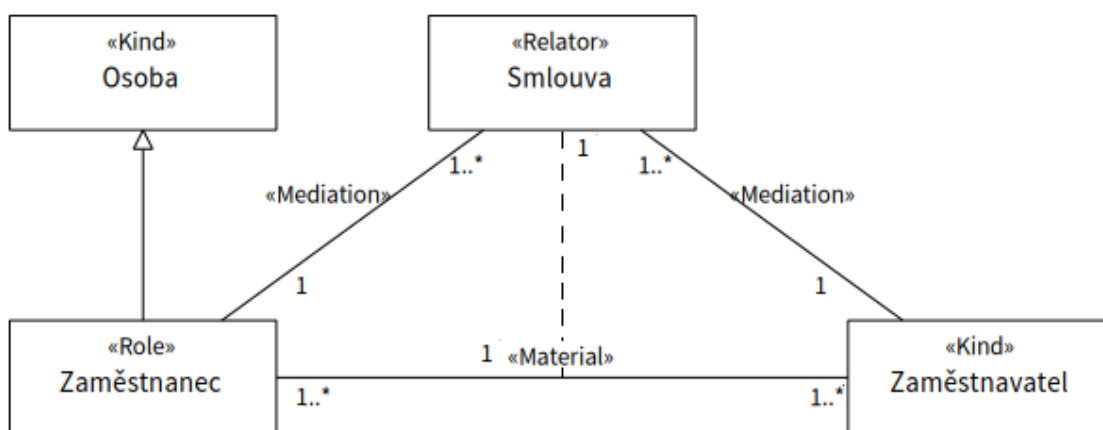


## 1. Formální relace

Formální relace nepotřebuje pro svou existenci žádné pečetidlo. Vyjadřuje charakterizační vztah jednotlivých instancí, kde dochází k porovnání jejich kvalit.[36]

## 2. Materiální relace

Tento druh relace se nazývá «*Material*» a vyžaduje pro svou funkčnost mediační individuum, kterým je v OntoUML «*Relator*» spojené vazbou «*Mediation*». Se zánikem mediačního individua zaniká celá tato vazba. Jako příklad pro materiální relaci použijeme vztah zaměstnance a zaměstnavatele, kde pečetidlem jejich vztahu je smlouva. Jejich zaměstnanecký vztah zanikne spolu se zánikem smlouvy.[37]



■ Obrázek 6.4 Příklad materiální relace

## 6.6 Vztah celek-část

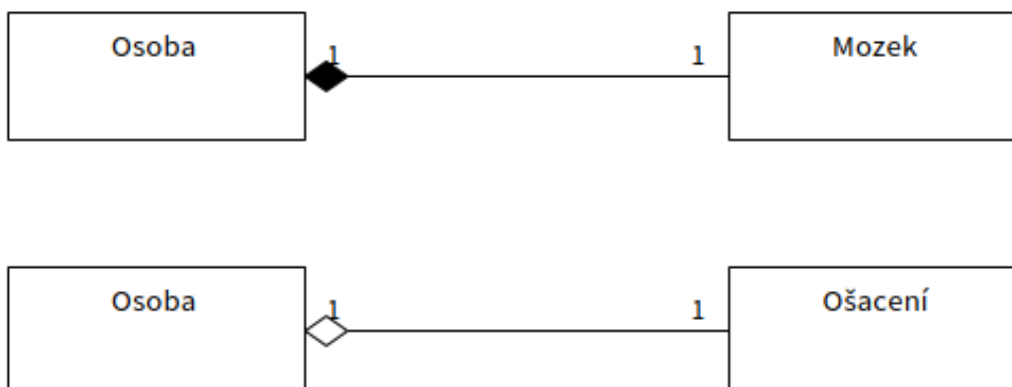
UFO se zabývá vztahem celek-část, protože podle Guizzardiho je důležitým prvkem konceptuálního modelování.[21] Vztah celek-část je speciálním druhem formální relace, který používá speciální stereotypy. Klasické UML rozděluje mezi agregací a kompozicí. Agregace určuje povinný vztah.[23] V následujícím příkladu 6.5 je použit v relaci osoby s mozkiem, kdy mozek je neoddělitelnou součástí osoby. V kompozici naopak může např. ošacení existovat mimo osobu. Toto vyjádření vztahu celek-část, jež je používáno v UML, je pro ontologické vyjádření nedostatečné, jelikož je založeno na dvou již zmíněných vztazích.

### 6.6.1 Sdílitelnost

V OntoUML vyjadřuje stejná syntax celku-části sdílitelnost, jež pojednává o tom, zda účastníci této relace mohou být součástí celku podle jeho typu. Pro entity části s plným diamantem je sdílitelnost taková, že mohou být účastníky více typů vazby celek-část, ale nemohou být účastníky více instancí jednoho stejného typu relace celek-část. Části s prázdným diamantem mohou být účastníky několika částí a vůbec u nich nezáleží na typu relace celek-část.[23]

### 6.6.2 Povinnost

Dalším rysem vztahu celek-část je povinnost. Na tu lze nahlížet jak z pohledu celku, tak z pohledu části, z nichž každý má své dělení. Prvně se podíváme na dělení z hlediska části.[23]



■ **Obrázek 6.5** Příklad celku-části v UML

### 1. Povinný celek

Povinný celek říká, že část nedokáže existovat bez celku, jelikož je na něm genericky závislá. Celek v tomto vztahu může svou instanci během své existence upravovat a měnit. Povinnost a nepovinnost se v OntoUML značí násobností na straně, na kterou „koukáme“. Zde je tedy na straně celku násobnost 1..1.[23]

### 2. Nepovinný celek

Nepovinný celek je typ vztahu, kde část je schopna existovat bez celku a dokáže po dobu existence své instance měnit celky. Díky těmto vlastnostem je na straně celku násobnost značena 0..1.[23]

### 3. Neoddělitelná část

Neoddělitelná část říká, že část v tomto vztahu je existenčně závislá na celku. Celek v tomto vztahu musí svou instanci zachovat a nijak ji neměnit. To je způsobeno faktem, že celek se podílí na identitě části. Násobnost na straně celku je 1..1, a vazbě navíc přidává meta-atribut *inseparable*. [23]

Nyní si ukážeme, jak se dělí povinnost z hlediska celku. Také se jedná o tři části.

#### 1. Povinná část

Povinná část znamená, že celek vyžaduje existenci části pro svou existenci. Zde jde také o generickou závislost, tudíž se může instance části měnit, ale nesmí zaniknout. Tento vztah značíme násobností 1..1 u části. To zaručuje nutnou přítomnost části.[23]

#### 2. Volitelná část

Celek s vlastností volitelné části není existenčně závislý na existenci části. Část nemusí vůbec existovat, což se značí násobností 0..1.[23]

**3. Esenciální část** Esenciální část je na tom podobně jako povinná část. Celek nemůže existovat bez části, a vazbu tudíž značíme násobností 1..1 na straně části. Protože je vztah esenciální, část nemůže během celé své existence s celkem měnit svou instanci, neboť se přímo podílí na jeho existenci. Vazba mezi celkem a částí je označena meta-atributem *essencial*. [23]

Značení meta-atributy *essencial* a *inseparable* se používá u rigidních stereotypů. Pro esenciální část se používá značení *essencial part* a pro neoddělitelnou část se používá *inseparable part*. Jelikož existují i anti-rigidní stereotypy, existuje také jejich speciální označení vztahu, kde je klíčovým slovo *immutable*. Pro esenciální část se používá meta-atribut *immutablePart* a pro neoddělitelnou část použijeme *immutableWhole*. Značení je rozdílné, ale význam vazby je stejný.[23]

### 6.6.3 Funkční celek

Funkční celek je označení vlastnosti entity, nejedná se tedy o stereotypy, a tudíž mohou jeho členy nabývat různých typů. Jejich vztah se nazývá «*ComponentOf*».[21]

### 6.6.4 Stereotypy celku-části

#### 1. «*Collective*»

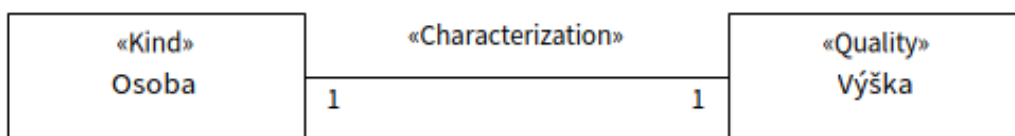
Tento stereotyp je rigidní sortál se schopností poskytovat identitu stejně jako například «*Kind*». Používá se pro označení celku, který je složený ze členů se stejnou rolí. Vztah mezi členem a celkem tohoto typu je nazván «*MemberOf*» a je nutné, aby měl minimálně dvě instance členů, proto se používá netypická násobnost 2..\*. Vztah mezi entitami typu «*Collective*» se označuje «*SubCollectionOf*».[38]

#### 2. «*Quantity*»

Stejně jako předchozí stereotyp i tento je rigidní sortál, který poskytuje identitu. Představuje celek, jehož části jsou stejného typu jako on. Používá se pro nekonečně dělitelné věci – často jde o sypké látky nebo tekutiny.[39] Vztah tohoto stereotypu s nádobou, již obývá, je označen «*Containment*».[40] Pro vztah dvou stereotypů «*Quantity*» se používá «*SubQuantityOf*».[41] Příkladem by mohlo být, že písek je «*SubQuantityOf*» hlíny.

## 6.7 Aspekty

Aspekty jsou existenčně závislé entity – nedokáží existovat bez jiného stereotypu, s nímž mají inherentní vazbu. Tato vazba se v OntoUML nazývá stereotypem «*Characterization*» a jedná se o vazbu mezi nositelem a vlastností. Vlastnost pomáhá charakterizovat nositele a je na něm existenčně závislá, jelikož je jeho inherentním momentem.[21] V OntoUML pro nositele existují dva typy stereotypů a to «*Mode*» a «*Quality*».



■ Obrázek 6.6 Příklad vztahu «*Characterization*»

#### 1. «*Mode*»

«*Mode*» je rigidní sortál a zároveň aspekt. Vyjadřuje neměřitelné a nestrukturované vlastnosti. Se svým nositelem je propojen asociací se stereotypem «*Characterization*» v násobnosti 1 : 1 nebo 1 : N. Zde je možné, aby nositel měl více nestrukturovaných vlastností stejného typu. Příkladem je program jako nositel, který by mohl vlastnit více vlastností typu podporované operační systémy.[42]

## 2. «Quality»

«Quality» je stejně jako «Mode» rigidní sortál a aspekt. Se svým nositelem je také propojen asociací typu «*Characterization*», ale může mít jen násobnost 1 : 1, neboť představuje strukturovanou vlastnost. UFO v otázce «Quality» dělí tento stereotyp na další tři podtypy. *Percievable quality*, *Non-percievable quality* a *Nominal quality*. *Percievable quality* značí měřitelnou vlastnost, dobrým příkladem jsou rozměry. Jejím opakem je *Non-percievable quality*, která reprezentuje stroji neměřitelné, jako například měnu. *Nominal quality* jsou vlastnosti nositele, jež se používají pro jeho označení.[43]

## Kapitola 7

# OpenPonk

Pro vytvoření konceptuálních modelů pro tuto bakalářskou práci byl použit nástroj OpenPonk. OpenPonk je modelovací platformou, která umožňuje vytváření konceptuálních modelů v OntoUML. Jedná se o opensource platformu, která byla vytvořena v technologii Pharo, napsána v jazyce Smalltalk a využívá principů objektově orientovaného programování.[44] Je vytvářena Centrem pro konceptuální modelování a implementaci[45] a je dostupná na operačních systémech Windows, Linux i macOS.[46]

Platforma navazuje na nástroj OpenCASE/OpenCABE, jenž byl vyvíjen na ČVUT FIT na Katedře softwarového inženýrství. OpenPonk navazuje na jeho myšlenku v poskytování modelovacích a algoritmických notací pro jazyky, které nemají dostatečné zastoupení v mainstreamovém použití.[45]

OpenPonk zastřešuje podporu pro modelovací jazyk UML, dále pro OntoUML, které je využito pro tuto bakalářskou práci, BORM, konečné automaty, Petriho sítě a BPMN. Nejvíce prostředků je dáno OntoUML – podpora pro něj je zde největší. OpenPonk dokonce umožňuje validaci OntoUML modelů a upozorňuje na anti-vzory.[46]

Pro ulehčení práce s modely v projektu OpeNest je vyvíjena speciální verze OpenPonku, která přináší spoustu nových funkcí pro OntoUML modely. Zásadním rozšířením funkcionality je možnost importování modelů, jež jsou nahrány na Git projektu. To ulehčuje práci se šablonami a zřetelně zrychluje práci na nových modelech. Tato funkce zároveň při vytváření nových entit napovídá s jejich názvem a stereotypem. Dalším vylepšením je možnost hotový model nahrát z OpenPonku přímo na Git a datovou platformu OpeNest. Další funkce jsou ve vývoji.



Část II

Vlastní řešení





# Klíčové domény

Tato bakalářská práce byla vytvořena pro účely projektu OpeNest (viz 2) v rámci ontologického týmu, který vedou Ing. Tereza Macháčová a Bc. Jana Martínková. Pod vedením doc. Ing. Roberta Pergla, Ph.D., byly vybrány klíčové domény pro ontologickou analýzu, která byla prováděna samostatně a pouze konzultována s vedením týmu a vedoucím bakalářské práce. Účelem práce bylo provést ontologickou analýzu, vytvořit konceptuální model klíčové domény a na závěr bylo provedeno namapování dat. Jelikož klíčové domény jsou heterogenního charakteru, cílem bakalářské práce bylo provedením předchozích úkonů zajistit sémantickou interoperabilitu výsledných ontologických konceptuálních modelů. V rámci práce došlo k vytvoření šablon, které napomáhají k dalšímu modelování. Zhotovené modely vycházejí z datových sad, jež spadají pod otevřenou licenci, je tedy možné je zveřejnit v bakalářské práci v jejich plném rozsahu. Zdrojem pro datové sady jsou průzkumy Českého statistického úřadu a průzkumy Statistického úřadu Evropské unie, Eurostatu. Šablona **staty-eu** byla vytvořena z důvodu častého opakování entit států Evropské unie v modelech vytvořených podle datových sad od Eurostatu. Sestavena byla ve spolupráci s Ing. Terezou Macháčovou.

Datové sady poskytl společnost Remmark, a.s., v rámci projektu OpeNest. Sady se dělí podle licence – modely z veřejných domén (Eurostat, ČSU) jsou poskytovány zdarma, ale interní datové sady (Remmark, a.s.) jsou zpoplatněny. Z důvodu existence více zdrojů je vyvíjen tlak na důkladnou ontologickou analýzu dané domény a tato rozmanitost je také důvodem provádění opakované ontologické analýzy již vyhotovených modelů, která zajišťuje zachování sémantické interoperability dat. Při kontrole vyhotovených modelů se kolize v datech hlásí a při rozsáhlejších problémech i diskutují s týmem, aby byla doména analyzována správně. Poté se kolize připíše k daným datovým sadám, aby mohlo dojít k jejich nápravě. Ke kolizím dochází z důvodu špatného pojmenování entity, kdy název může být sdílen s jiným významem, nebo může jít o nedostatečně konkrétně popsanou skutečnost. Dalším problémem jsou rozdílné stereotypy entit. Aby se předcházelo těmto chybám, je vhodné při tvorbě modelu procházet již existující modely, což značně zpomaluje práci na nové doméně.

Modelování bylo prováděno v nástroji OpenPonk (viz 7), který je vybaven podporou pro konceptuální modelování v OntoUML (viz 6) a zároveň podporuje funkce specifické pro projekt OpeNest. Před modelování je nutné provést transformaci datové sady, což má na starosti jiný tým. Transformace vyprodukuje strukturovaná data, která je nutné namapovat v modelu, ovlivňuje počet entit, které jen nutné namodelovat, a určuje počet datových entit, do kterých se zadávají mapovací pravidla pro data.

Celkem bylo vybráno 5 klíčových domén a 2 šablony, na kterých byla provedena důkladná ontologická analýza. Na jejím základě byly k doménám vytvořeny ontologické konceptuální modely v jazyce OntoUML, které byly následně převedeny na datové modely a došlo k obohacení mapovacími pravidly. Ve výsledku bylo vytvořeno 554 entit a 18 datových entit s 662 mapovacími pravidly.

## Kapitola 9

# Modelování

Stěžejním bodem bakalářské práce bylo modelování ontologických konceptuálních modelů v jazyce OntoUML. Model zachycuje vztahy mezi daty datové sady, ale již není jeho úkolem zachytit hodnoty dat. OntoUML bylo vybráno z důvodu, že je doporučeno iniciativou GO FAIR (viz 3). Výsledkem je zlepšení sémantické interoperability poskytovaných dat službou OpeNest.

### 9.1 „Blitzkrieg“ modelování

V průběhu vypracovávání práce nastal problém s tím, že bylo nutné dokončit velké množství modelů v krátkém časovém rozmezí z důvodu termínů na evropské dotace. Proto vedení týmu přišlo s návrhem takzvaného „Blitzkrieg“ modelování, kde nebyl kladen důraz na ontologickou analýzu. To přineslo spoustu úskalí s kolizemi a se sémantickou interoperabilitou celkově.

Základní pravidla „Blitzkrieg“ modelování:

1. Provádí se jen základní ontologická analýza, kde se modeluje spíše podle intuice než faktů.
2. Nehledí se na transformaci sady, tj. modely nejsou modelovány podle strukturovaných dat.
3. Mapování dat se redukuje jen na jednoduchá pravidla.
4. Modely se nekontrolují a neupravují se podle nových šablon.

### 9.2 Šablony

Šablona je klasický konceptuální model v OntoUML, na který se nevztahují žádná speciální pravidla. Rozdíl mezi šablonou a modelem podle datové sady spočívá v tom, že šablona neobsahuje žádné datové entity, a tudíž ani žádná mapovací pravidla. Důvodem pro existenci šablony jsou často opakované části modelů, které je vhodné unifikovat. V této práci došlo k vypracování dvou šablon. První z nich je šablona **pravni-osobnost**. Druhá je šablona **staty-eu** a byla vytvořena ve spolupráci s Ing. Terezou Macháčovou.

#### 9.2.1 Tvorba šablon

Tvorba šablony začíná vytyčením problematiky, na které se shodne modelovací tým na konzultaci. Daná problematika se důkladně zanalyzuje, namodeluje a obecně projde několika iteracemi, než je plně použitelná. Do ostatních modelů se vkládají jen části důležité pro danou datovou sadu.

### 9.2.1.1 Výhody šablon

Samotné šablony dělají tvorbu konceptuálních modelů pohodlnější, jelikož zabraňují opakovanému modelování téhož. Pro co největší efektivitu šablon je stěžejní funkce OpenPonku importovat již existující modely do rozpracovaných. Modelář si tak ušetří práci, protože nemusí znovu provádět ontologickou analýzu a mitiguje riziko vzniku kolizí.

## 9.3 Tvorba modelů

Před začátkem modelování se provádí ontologická analýza dané klíčové domény. Cílem je analyzovat skutečnost a význam, ze kterých datová sada vychází. Důležité je zjistit definice a význam výrazů použitých ve zpracovávané doméně. Rešerše k tématu spočívá ve čtení právních dokumentů, v dokumentacích a v konzultacích s odborníky v oboru. Například pro tvorbu šablony **prawni-osobnost** byla vedena konzultace s Ing. Miroslavou Šírovou. Zde šlo pouze o opravu faktických chyb v již existující šabloně, kterou vypracoval Bc. Václav Šír pro svou bakalářskou práci.

### 9.3.1 Analýza klíčové domény

Při ontologické analýze se definují výrazy a definice. Spojuje se skutečnost a popis transformované datové sady. Jak již bylo zmíněno, popis a transformace sady nespadá pod tuto práci a prováděl ji jiný tým. Z popisu vychází upřesnění tématu a slouží k orientaci v úplných základech.

### 9.3.2 Vytváření konceptuálního modelu

Tvorba modelu začíná vytyčením rigidních sortálů (viz 6.4.1) «*Kind*» a «*Subkind*». Z těchto stereotypů nám vzniká množina indukovaných podgrafů. V souladu s popisem sady se vytvoří entity, jež zachycují důležité prvky domény. Na tento bod v modelu se dále navazuje a jsou doplňovány další stereotypy a vazby mezi nimi. Druhým krokem je rozšíření o anti-rigidní sortály (viz 6.4.2), tj. stereotypy, které dokáží popsat specializace rigidních sortálů. Poté se definují non-sortály (viz 6.4.3). Tato skupina stereotypů pomáhá spojovat sortály podle jejich společných vlastností a při složitějších modelech jejich existence napomáhá struktuře celého modelu. Dále se doplňují aspekty (viz 6.7). Nakonec se připojují stereotypy a další a zlepšuje se přesnost modelu.

#### 9.3.2.1 Určení klíčových entit

Tvorba modelu začíná analýzou popisu datové sady, jenž je dostupný na datové platformě a vyplývají z něj entity, které budou souviset s datovými entitami. Základem je jejich definování, neboť umožňuje další rozšiřování a specifikaci modelu.

V obrázku 9.1 jsou v datové platformě popsány struktury dat pro datovou sadu **eu-vyroba-a-obchod-s-primarnimi-produkty-bunicina-papir-a-lepenka**. Zde začíná první krok ontologické analýzy vytyčením základních entit.

Vybrané entity jsou namodelované v jazyce OntoUML za pomoci OpenPonku. V tomto kroku se entitám přiřazují stereotypy, které buď vycházejí z analýzy, nebo se použije stereotyp, které byl již použit v jiném modelu. Používání stejných stereotypů entit skrze modely je klíčové pro dosažení sémantické interoperability. Příklad vytyčených entit můžeme vidět v obrázku 9.2. Z popisu vychází taky část modelu se státy Evropské unie, která byla importována ze šablony **staty-eu**.

Po provedení analýzy popisu datové sady lze snadno stanovit části, jež je nutné namodelovat podrobněji – například oblast vývozu a dovozu, měrné jednotky nebo správné rozdělení dřevěných výrobků do kategorií. Při modelování těchto skutečností nemusí být všechny entity propojené, a proto se přidávají pomocné entity, které klarifikují popisovanou skutečnost. Model nyní obsahuje

Název	Ontologická entita	Slovníkový pojem	Titulek	Datový typ	Povinný	Popis
unit			unit	String	<input checked="" type="checkbox"/>	Unit of measure/Měrná jednotka. "THS_EUR - Thousand euro / Tisíc eur, THS_NAC - Thousand units of national currency / Tisíce jednotek národní měny, THS_T - Thousand tones / Tisíc tónů"
prod_wd			prod_wd	String	<input checked="" type="checkbox"/>	Wood products/Výrobky ze dřeva. "PL - Wood Pulp / Buničina, PLO - Other pulp / Jiná dužina, PLO_NW - Pulp from fibres other than wood / Buničina z jiných vláken než ze dřeva, PLO_RC - Recovered fibre pulp / Regenerovaná vláknitá buničina, PL_CH - Chemical wood pulp / Chemická dřevní buničina, PL_CH_SAB - Sulphate bleached wood pulp / Sulfátem bělená dřevitá buničina, PL_CH_SANB - Sulphate unbleached wood pulp / Sulfátová nebělená dřevitá buničina, PL_CH_SIB - Sulphite bleached wood pulp / Sulfitem bělená dřevitá buničina, PL_CH_SINB - Sulphite unbleached wood pulp / Sulfitová nebělená dřevitá buničina, PL_DS - Dissolving grades wood pulp / Rozpouštěcí kvalitní dřevní buničiny, PL_MC - Mechanical wood pulp / Mechanická buničina, PL_MC_SCH - Mechanical and semi-chemical wood pulp / Mechanická a polochemická buničina, PL_SCH - Semi-chemical wood pulp / Polochemická dřevitá buničina, PP - Paper and paperboard / Papír a lepenka, PP_GR - Graphic papers / Grafické papíry, PP_GR_CO - Coated papers / Natírané papíry, PP_GR_MC - Uncoated mechanical papers / Mechanické papíry bez povrchové úpravy, PP_GR_NP - Newsprint / Novinový papír, PP_GR_NW - Uncoated woodfree papers / Bez dřevěné papíry bez povrchové úpravy, PP_HS - Household and sanitary papers / Domácí a hygienické papíry, PP_O - Other paper and paperboard n.e.s. / Ostatní papír a lepenka j.n., PP_PK - Packaging materials / Obalové materiály, PP_PK_CB - Cartonboard / Karton, PP_PK_CS - Case materials / Materiály pouzdra, PP_PK_O - Other papers mainly for packaging / Ostatní papíry hlavně pro balení, PP_PK_WR - Wrapping papers / Balicí papír, RCP - Recovered paper / Obnovový papír, PL_CH_DS - Chemical and dissolving grades wood pulp / Chemická a rozpustná dřevná buničina, PP_GR_CO_NCO - Graphic papers, printing and writing / Grafické papíry, tisk a psaní"
stk_flow			stk_flow	String	<input checked="" type="checkbox"/>	Stock or flow/Zásoba nebo tok. "EXP - Exports / Vývoz, EXP_XEU - Export to non-EU countries / Vývoz do zemí mimo EU, IMP - Imports / Dovozy, IMP_XEU - Import from non-EU countries / Dovozy do zemí mimo EU, PRD - Production / Výroba"
geo	<a href="#">◀ Území států Společenství států Data</a>		geo	String	<input checked="" type="checkbox"/>	Geography - Název státu nebo geopolitické oblasti. "AT - Austria / Rakousko, BE - Belgium / Belgie, BG - Bulgaria / Bulharsko, CH - Switzerland / Švýcarsko, CY - Cyprus / Kypr, CZ - Czechia / Česko, DE - Germany / Německo, DK - Denmark / Dánsko, EE - Estonia / Estonsko, EL - Greece / Řecko, ES - Spain / Španělsko, FI - Finland / Finsko, FR - France / Francie, HR - Croatia / Chorvatsko, HU - Hungary / Maďarsko, IE - Ireland / Irsko, IS - Iceland / Island, IT - Italy / Itálie, LI - Liechtenstein / Lichtenštejnsko, LT - Lithuania / Litva, LU - Luxembourg / Lucembursko, LV - Latvia / Lotyšsko, MT - Malta / Malta, NL - Netherlands / Holandsko, NO - Norway / Norsko, PL - Poland / Polsko, PT - Portugal / Portugalsko, RO - Romania / Rumunsko, SE - Sweden / Švédsko, SI - Slovenia / Slovinsko, SK - Slovakia / Slovensko, UK - United Kingdom / Spojené království, EU27_2020 - the Member States of the European Union at 1.2.2020: EU-28 minus United Kingdom (UK) / Členské státy Evropské unie k 1.2.2020: EU-28 minus Spojené království, EU28 - the Member States of the European Union from 1.7.2013 to 31.1.2020: EU-27_2007 + Croatia (HR) / Členské státy Evropské unie od 1.7.2013 do 31.1.2020: EU-27_2007 + Chorvatsko (HR)"

■ **Obrázek 9.1** Entity podle datové platformy

skupiny entit, které spolu nemusí být nutně spjaté vazbami, a je připraven na další kroky analýzy domény a modelování.

### 9.3.2.2 Specifikace detailů modelu

V této fázi je model sestaven z uskupení entit, které jsou připravené na další rozšíření o detaily. Tento krok je důležitý, protože více přibližuje význam modelu k realitě, nese ovšem i svá rizika, protože míra detailu ovlivňuje čitelnost konceptuálního modelu. Velmi detailní model byl obrovský, nečitelný a práce s ním by byla zdlouhavá, proto je nutné správně analyzovat problematiku, aby nebyl přehlcen nadbytečnými detaily a zůstal přehledným a přesným.

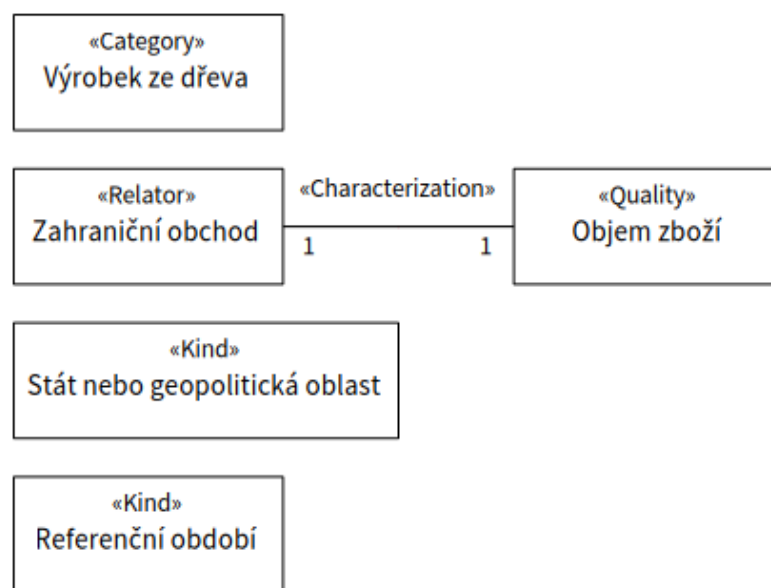
Proces specifikace detailů se odvíjí od entit, jež vychází z popisu datové sady. Tyto sady slouží jako startovní bod, podle kterého se vyhledávají dodatečné informace k vypracovávanému tématu. Informace získané rešerší slouží jako stavební kámen pro entity, které rozšiřují a specifikují konceptuální model. Při stavbě těchto nových entit je nutné se řídit pravidly OntoUML, podle nichž se musí řídit i směr rešerše.

OpenPonk značně napomáhá při vytváření entit, je totiž velmi pravděpodobné, že při vytváření entity již taková entita existuje v jiném modelu. OpenPonk má přístup ke všem modelům nahraným na datové platformě, a našeptává tedy jméno a stereotyp entity. Tato funkcionality velmi ulehčuje práci – dříve se používal takzvaný Index, tj. velká tabulka obsahující všechny entity se stereotypy a datovou sadou, pod kterou spadají. Index jako takový má stále své využití při hledání specifické entity, vyhledávání již existujících entit je totiž zásadní pro zajištění sémantické interoperability mezi modely. Snižuje se šance vzniku dvou entit, které mají stejné jméno a jiný význam nebo rozdílné jméno a stejný význam.

### 9.3.2.3 Tvorba spojitého modelu

V této fázi modelování je model rozdělen na části, z nichž každá popisuje jinou část celé problematiky, ale model jako takový není spojitý. Dalším úkolem je ze spojitých podgrafů vytvořit jeden spojitý graf.

Ontolog analyzuje jednotlivé části a hledá spojitosti mezi fragmentovaným modelem. Způsoby, jak propojit jednotlivé části, se liší podle komplexnosti problému. První možností je vztah mezi



■ **Obrázek 9.2** Vytýčené entity podle popisu datové sady

existujícími entitami. Jedná se o nejjednodušší propojení a nevyžaduje složitou analýzu. Dalším stupněm jsou propojující entity, kde se pouze přidává entita, jež je zásadní pro propojení dvou a více částí. Ontolog může dojít k závěru, že je nutné zanalyzovat celou novou problematiku, aby byl schopen model propojit. Tím se vrací k předchozímu kroku (viz 9.3.2.2). Namodelovaná nová problematika pak propojí původní části modelu a výsledkem je spojitý OntoUML konceptuální model, jenž čeká na rozšíření o datové entity a mapovací pravidla, která budou rozebrána níže.

## 9.4 Praktická ukázka na modelu

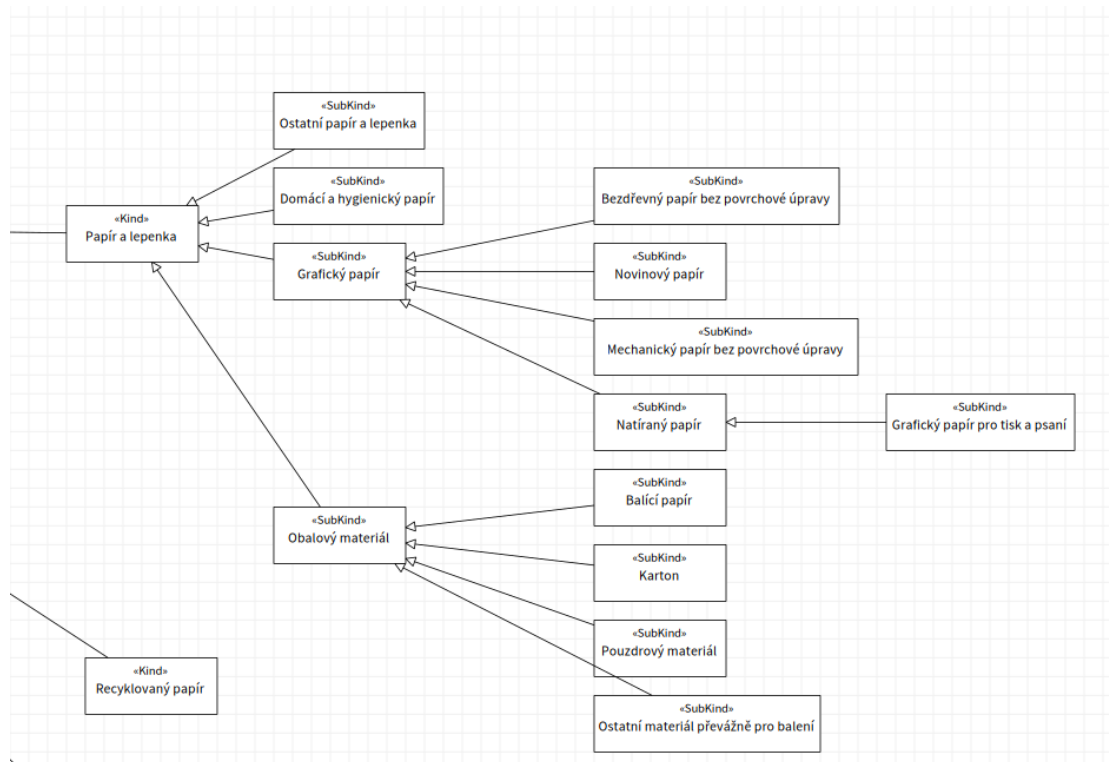
V této sekci práce si proces tvorby ontologického konceptuálního modelu popíšeme na praktickém příkladu. Pro tyto účely poslouží datová sada `eu-vyroba-a-obchod-s-primarnimi-produkty-bunicina-papir-a-lepenka`. Tato datová sada vychází z průzkumu evropského statistického úřadu Eurostat a popisuje pohyb dřevěných produktů (buničina, papír a lepenka) na mezinárodním trhu.

Analýza započala určením klíčových entit (viz 9.3.2.1). Po provedení analýzy popisu datové sady došlo k výběru entit, které slouží jako stavební kameny pro celý model. Mezi tyto entity patří: *výrobek ze dřeva*, *zahraňní obchod a jeho objem*, *stát a referenční období* (viz 9.2). Při analýze popisu datové sady byl zároveň využíván index entit přes funkcionalitu OpenPonku, aby nedošlo k vytvoření duplicitních dat, která by narušila sémantickou interoperabilitu mezi modely. Bylo také zjištěno, že bude potřeba vytvořit konceptuální modely k více druhům dřevěných výrobků, a proto se analýza rozšířila i o popisy těchto datových sad. Další průnik byl v oblasti vývozu a dovozu s modelem `eu-vyvoz-a-dovoz-clenskych-statu-eu-tretich-zemi`. Tímto byl zakončen prvotní krok analýzy a přecházíme na fázi, v rámci níž se specifikují detaily modelu (viz 9.3.2.2).

### 9.4.1 Dřevěné výrobky

První částí modelu, již bylo potřeba rozmodelovat, aby lépe zachycovala skutečnost, jsou dřevěné výrobky. V této problematice bylo nutné správně zachytit strukturu dřevěných výrobků.

Zároveň bylo po konzultaci s vedením modelovacího týmu rozhodnuto, že bude prospěšné namodelovat všechny dřevěné výrobky vyskytující se v ostatních datových sadách. V příkladu 9.3 je namodelována skutečnost vyjadřující strukturu druhů papíru a lepenky.



■ Obrázek 9.3 Namodelovaná struktura dřevěných výrobků

## 9.4.2 Státy Evropské unie

Doména *státy Evropské unie* se objevuje ve velkém množství datových sad, a proto bylo rozhodnuto, že vytvoření šablony pro tuto problematiku bude přínosem pro celý modelovací tým. Samotná šablona byla vytvořena ve spolupráci s Ing. Terezou Macháčovou. Postup pro tvorbu konceptuálního modelu je téměř identický jako pro model k datové sadě (viz 9.2.1). Prvně došlo k namodelování jednotlivých států Evropské unie, kde bylo nezbytné rozdělit Francii na zámořskou a metropolitní kvůli datovým sadám, které její rozdělení potřebují.

Dalším krokem bylo namodelování vztahu evropského hospodářského prostoru s Evropskou unií a ostatních geopolitických oblastí Evropské unie (viz obrázek 9.4). U těchto vztahů došlo k lepšímu vyjádření zkušenosti, než bylo dosud používáno v modelech.

Vytvoření této šablony vedlo ke značnému množství kolizí, jelikož modely, které byly vytvořené před touto šablonou a využívají státy Evropské unie, reprezentovaly tuto problematiku jiným způsobem, a tudíž bylo nutné zaznamenat kolize do tabulky kolizí a modely opravit. I přes tento krok navíc je šablona velkým pomocníkem při tvorbě modelů a dokáže ušetřit spoustu práce.





# Mapování dat

Pro projekt OpeNest je stěžejní, aby modely pomáhaly propojovat data datových sad. Tím vzniká pojem datový model, který specifikuje, jak se operuje s daty datové sady. V tomto případě jde o obohacení konceptuálních modelů v OntoUML o takzvané datové entity.

Datový model v kontextu projektu OpeNest je pouze nadstavba nad ontologickým konceptuálním modelem. Ten se rozšíří o již zmiňované datové entity, které se obohatí o atributy s mapovacími pravidly. Pro účel tohoto projektu byl vytvořen mapovací jazyk, který je využíván datovou platformou k práci s daty datových sad.

### 10.1 Datová entita

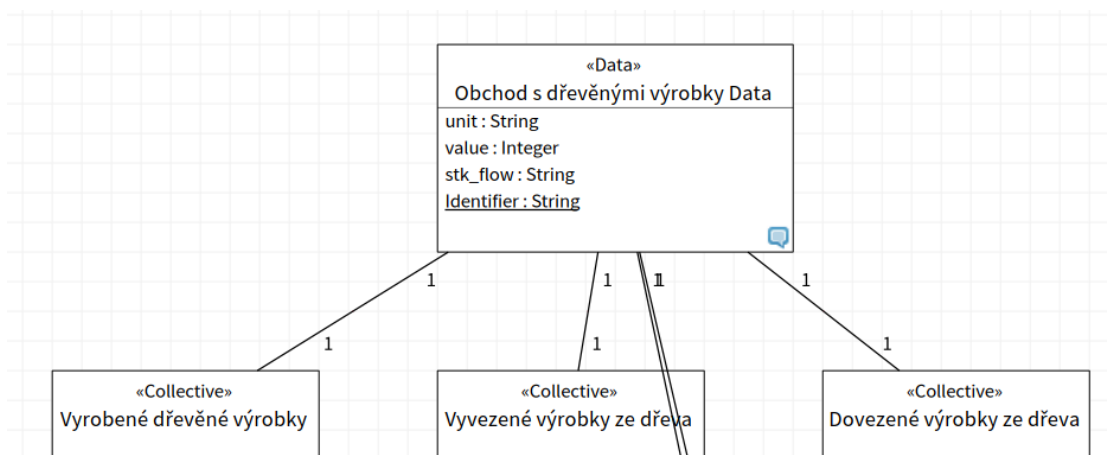
Datové entity se modelují k hotovému konceptuálnímu modelu – nad rámec OntoUML jim byl vytvořen stereotyp *Data*. Datová entita sama o sobě nic nereprezentuje, ale slouží jako pojídlo mezi modelem a daty, proto musí být doplněna o atributy a mapovací pravidla. Při jejich tvorbě je nutné navázat je na entity, jež reprezentují data datové sady. Spojují se asociací s vazbou **1:1**. Na názvu datové entity nezáleží, důležité je pouze, aby končil klíčovým slovem *Data*, modelovací tým přesto pro přehlednost zavedl konvenci jejich pojmenování. Datová entita by měla nést název entit, se kterými je propojena, a pokud je takových entit větší množství, hledá se označení, které by co nejlépe vystihovalo uskupení entit, které propojuje s daty. Vhodné pojmenování pro více entit je uvedeno v obrázku níže 10.1.

#### 10.1.1 Atributy datových entit

Atributy datových entit jsou založené na vlastnostech z UML (viz 5). Atribut má za úkol vyjadřovat data, ke kterým se píší mapovací pravidla. Název atributu je odvozen z popisu datové sady a datový typ, který se mu rovněž musí přiřadit, je převzat z UML, tj. můžeme se setkat s typy jako např. *String*.

Každá datová entita musí mít atribut jménem *Identifier*, jenž je vždy datového typu *String* a slouží jako ukazatel datové sady. Je důležitý pro mapovací pravidla, jelikož je přiřazuje k datové sadě.

Datová entita tedy nemůže existovat bez atributu ukazatele, a zároveň je podmínkou, aby měla alespoň jeden další atribut, neboť pokud není propojena s daty, postrádá význam.



■ **Obrázek 10.1** Namodelovaná struktura dřevěných výrobků

## 10.2 Mapování

Datové entity slouží k obecnému provázání konceptuálního modelu s daty. Pro specifikaci těchto vztahů slouží mapování dat, které k atributům přidává pravidla, podle nichž jsou data propojena.

Za účelem zápisu těchto pravidel byla pro projekt OpeNest vytvořena syntax, dle které jsou zapisována mapovací pravidla spojující data datové sady s ontologickým modelem.

Pro překladač pravidel v datové platformě je nutné, aby byla syntax exaktní, jinak bude pravidlo vyhodnoceno jako nevalidní. Pravidla se dělí na základní a podmíněná mapovací pravidla a jejich syntax bude popsána níže.

### 10.2.1 Základní mapovací pravidla

Jednotlivé části mapovacího pravidla musí být odděleny mezerou a celé pravidlo musí končit středníkem. Základní mapovací pravidla se dělí na dvě části (viz 10.1): *název atributu* a *název entity*. Tyto části jsou spojeny „šipkou“ `->`, která znázorňuje, že atribut je namapován na danou entitu.

■ **Výpis kódu 10.1** Ukázka základního mapovacího pravidla

```
"název atributu" -> "název entity";
```

Toto pravidlo má využití pouze tehdy, je-li atribut spojen jen s jednou entitou. Vhodnou ukázkou je datová sada s dřevěnými výrobky, kde se jedná o spojení atributu *year* a entity *referenční období*. V následujícím příkladu 10.2 je zachycena ukázka napsaného pravidla bez MEA mapování, které zde bylo pro zjednodušení vynecháno, ovšem důraz na něj bude kladen níže.

■ **Výpis kódu 10.2** Praktická ukázka základního pravidla

```
"year" -> "Referenční období";
```

### 10.2.2 Základní mapovací pravidla s podmínkou

Propojení atributu s více entitami je poměrně časté, proto se využívá základního pravidla s podmínkou. Podmínka se píše k názvu atributu přes `=` a její význam je takový, že pokud má atribut hodnotu, jaká je uvedena v podmínce, mapuje se na příslušnou entitu. Podmínka má svá pravidla

pro zápis: pokud je podmínkou číslo, neuvádí se v uvozovkách, zatímco text nebo text s číslicemi se do uvozovek píše (viz 10.3). Podmínku je také možné psát jako regulární výraz, pro který se používá tato syntax: /PR\_.\*./.

■ **Výpis kódu 10.3** Ukázka základního mapovacího pravidla s podmínkou

```
"název atributu" = "podmínka" -> "název entity";
```

Při psaní mapovacích pravidel je zásadní mít pro každou podmínku vlastní mapovací pravidlo. Příkladem je mapování na dřevěné výrobky, kdy se atribut `prod_wood` mapuje na entity dřevěných výrobků podle zkratkového kódu. V tomto případě mapování vyšlo na 29 mapovacích pravidel, z nichž vybraná část je v ukázce 10.4.

■ **Výpis kódu 10.4** Praktická ukázka základního pravidla s podmínkou

```
"prod_wood" = "PL" -> "Dřevovina";
"prod_wood" = "BLO" -> "Jiná dužina";
"prod_wood" = "PLO_RC" -> "Recyklovaná vláknitá buničina";
"prod_wood" = "PL_CH" -> "Chemická dřevovina";
```

### 10.2.3 Podmíněná mapovací pravidla

Pro určité vztahy nejsou pravidla s podmínkou dostatečná. Bývá potřeba zachytit stav, kdy pravidlo nabývá své platnosti pouze, pokud má atribut určitou hodnotu (viz 10.5).

■ **Výpis kódu 10.5** Ukázka podmíněného pravidla

```
("název atributu 1" = "podmínka") -> ("název atributu 2" -> "název entity");
```

Dále je možné atributy i entity spojovat pomocí logických spojek *AND* a *OR*.

■ **Výpis kódu 10.6** Ukázka podmíněného pravidla s logickými spojkami

```
("název atributu_1" = "podmínka" AND "název atributu 2") ->
("název atributu 3" -> "název entity 1" OR "název entity 2");
```

V práci bylo toto mapování využito pro zachycení vyprodukovaných dřevěných výrobků vyjádřených v národní měně daného státu. V příkladu 10.7 je ukázka vyprodukovaných dřevěných výrobků pro stát Rumunsko.

■ **Výpis kódu 10.7** Ukázka podmíněného pravidla s logickými spojkami

```
("unit" = "THS_NAC" AND "stk_flow" = "PRD" AND "geo" = "RO")
-> ("value" -> "Vyrobené dřevěné výrobky" AND "MEA_Finanční obnos"
AND "MEA_Tisíc" AND "MEA_RON");
```

### 10.2.4 MEA pravidla mapování

Výše bylo zmíněno MEA mapování. To je postaveno na frameworku Measurement Ontology Pattern Language, jenž je založen na OntoUML.[47]

V projektu OpeNest se MEA mapování používá pro zachycení měřitelných jednotek a je velkým pomocníkem při jejich unifikaci. Pomocí MEA se dá namapovat nominální hodnota, interval, nebo i poměr. V této práci byla MEA využita primárně pro nominální hodnoty. Příkladem může být zmiňované mapování atributu `year`. Nejprve se použije označení *MEA\_Časový okamžik* pro určení oblasti měřené jednotky a poté *MEA\_ISO*, které specifikuje formát jednotky. Pro čas je používána specifikace ISO 8601.

■ **Výpis kódu 10.8** Praktická ukázka základního pravidla

```
"year" -> "Referenční období" AND "MEA_Časový okamžik" AND "MEA_ISO";
```

## 10.2.5 Zajištění sémantické interoperability

Klíčové domény vypracované jako součást této bakalářské práce pro projekt OpeNest jsou heterogenní, protože každá z nich zachycuje statistiku jiné části reality. Pro úspěšnost projektu je zásadní, aby byla zajištěna sémantická interoperabilita mezi jednotlivými modely. Nejen z důvodu vzniku možných kolizí, ale také z důvodu rozšiřitelnosti o nové modely. Bez sémantické interoperability by tento krok byl velice časově náročný, a cílem práce tudíž nebylo jen zajištění sémantické interoperability mezi modely práce, ale i mezi modely jež jsou součástí datové platformy.

Tento krok nebyl snadný, ale i přes veškerá úskalí se povedlo tento cíl naplnit. Velkou pomocí jsou dvě šablony `staty-eu` a `pravni-osobnost`, které napomáhají další tvorbě modelů a snižují riziko vzniku kolizí. Všechny modely v této práci využívají šablony se státy, pouze model `prehled-podnikatelu-a-zivnosti` využívá šablonu s právní osobností.

Model pro datovou sadu `eu-vyroba-a-obchod-s-primarnimi-produkty-bunicina-papir-a-lepenka` má průnik s ostatními modely pouze skrze šablonu `staty-eu` (viz A.1). To je důsledkem velké odlišnosti této datové sady od ostatních, ale zároveň díky tomu dodržuje sémantickou interoperabilitu.

Datové sady `eu-domacnosti-typ-pripojeni-internetu` a `eu-osoby-ktere-si-nemohou-dovolit-osobni-automobil` spolu sdílí doménu typu domácností (viz A.2 a A.3). Pro dodržení sémantické interoperability bylo nutné doménu důkladně zanalyzovat a vyřešit kolize s ostatními sadami.

Přes entitu osobního automobilu mají průnik datové sady `eu-osoby-ktere-si-nemohou-dovolit-osobni-automobil` (viz A.4) a `eu-registrace-aut-podle-pohonu` (viz A.5). Každá z nich sice zachycuje jinou problematiku, ale bylo nutné, aby osobní automobil měl stejný stereotyp. Tento úkol byl splněn a sémantická interoperabilita byla zachována. Model pro registraci aut podle pohonu navíc rozšiřuje doménu dopravních prostředků, čímž poskytuje ontologický základ pro další modely s podobnou problematikou.

Datová sada `prehled-podnikatelu-a-zivnosti` zachovává sémantickou interoperabilitu skrze entitu osoby (viz A.6) a zároveň je založena na šabloně `pravni-osobnost`.

Sémantická interoperabilita byla zachována zásluhou důkladné ontologické analýzy klíčových domén, a i přes jejich heterogenitu je tedy možné modely mezi sebou propojit, aniž by vznikaly kolize skrze společné problematiky. Navíc byla sémantická interoperabilita zachována i mezi modely již nahranými na datové platformě. Tím modely přispívají k budoucímu rozšiřování projektu a k jemu vytyčeným cílům.

## Závěr

Cílem této závěrečné bakalářské práce bylo provedení ontologické analýzy nad klíčovými doménami a tím zajistit sémantickou interoperabilitu mezi ostatními modely z heterogenních domén. Tyto domény jsou součástí projektu OpeNest od společnosti Remmark, a.s., a jejího datového inkubátoru, kde poslouží dalšímu rozšíření báze znalostí. Vypracování práce probíhalo pod vedením doc. Ing. Roberta Pergla, Ph.D., a zároveň ve spolupráci s modelovacím týmem, který vedou Ing. Tereza Macháčová a Bc. Jana Martínková. Jednotlivými kroky pro splnění cílů bylo vyhotovení ontologické analýzy klíčových domén vybraných s vedoucím práce, poté vytvoření konceptuálních modelů a napsání mapovacích pravidel, čímž byla zaručena sémantická interoperabilita dat.

Výstupem této bakalářské práce jsou konceptuální modely v OntoUML na vybrané klíčové domény, které byly po vytvoření propojeny s ostatními modely v projektu OpeNest. Před vytvořením konceptuálního ontologického modelu byla provedena ontologická analýza a byly vytyčeny entity, jež byly dále v modelu použity tak, aby reprezentovaly co nejpřesněji skutečnost, kterou datová sada vyjadřuje. Celé vypracování modelů bylo provedeno v programu OpenPonk s verzí, která je úzce propojena s projektem OpeNest.

V modelech následně došlo k vytvoření datových entit, které vycházejí z transformovaných dat. Pro tyto entity byla vytvořena mapovací pravidla, která jsou důležitá k dotazování se na data, jelikož OpeNest a jeho datový inkubátor mají za úkol poskytovat data z různých datových sad přes speciální dotazovací jazyk. Aby tento cíl mohl být uskutečněn, je důležité, aby vyhotovené modely měly zajištěnou sémantickou interoperabilitu mezi sebou.

Pro zjednodušení práce modelovacího týmu došlo k vytvoření šablon pro modelování. Šablona vychází z často používaných pojmů, ale není vázána na specifickou datovou sadu. Díky funkci OpenPonku je snadné šablonu nainportovat do nového modelu a tím ulehčit práci na něm. Zároveň se tak nemusí opakovaně provádět analýza nad stejnými daty, čímž se snižuje riziko ohrožení sémantické interoperability.

Celkem bylo vytvořeno pět modelů s dvěma šablonami. Modely dohromady obsahují 554 entit a 18 datových entit s 662 mapovacími pravidly.

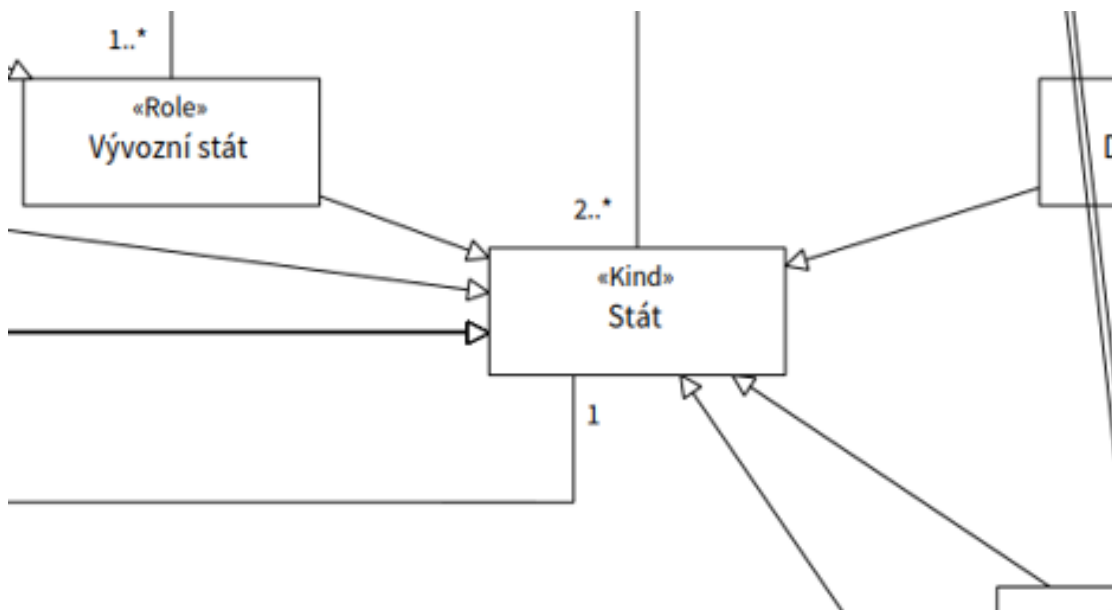
Sémantická interoperabilita byla dosažena důkladnou analýzou klíčových domén, dodržáním pravidel ve jménech entit, následnou kontrolou vazeb a vyřešením případných kolizí s ostatními modely v datové platformě. Pravidelné konzultace týmu pomohly řešit takové kolize a ve výsledku byla dosažena sémantická interoperabilita dat. Důsledkem toho je schopnost datové platformy propojovat modely a jejich data.

Práce je přínosem pro budoucnost projektu OpeNest, protože rozšiřuje počet modelů v datové platformě a svým udržením sémantické interoperability staví dobrý základ pro přidávání dalších modelů do platformy. Navíc je přínosem pro modelovací tým, jelikož rozšířila knihovnu šablon.

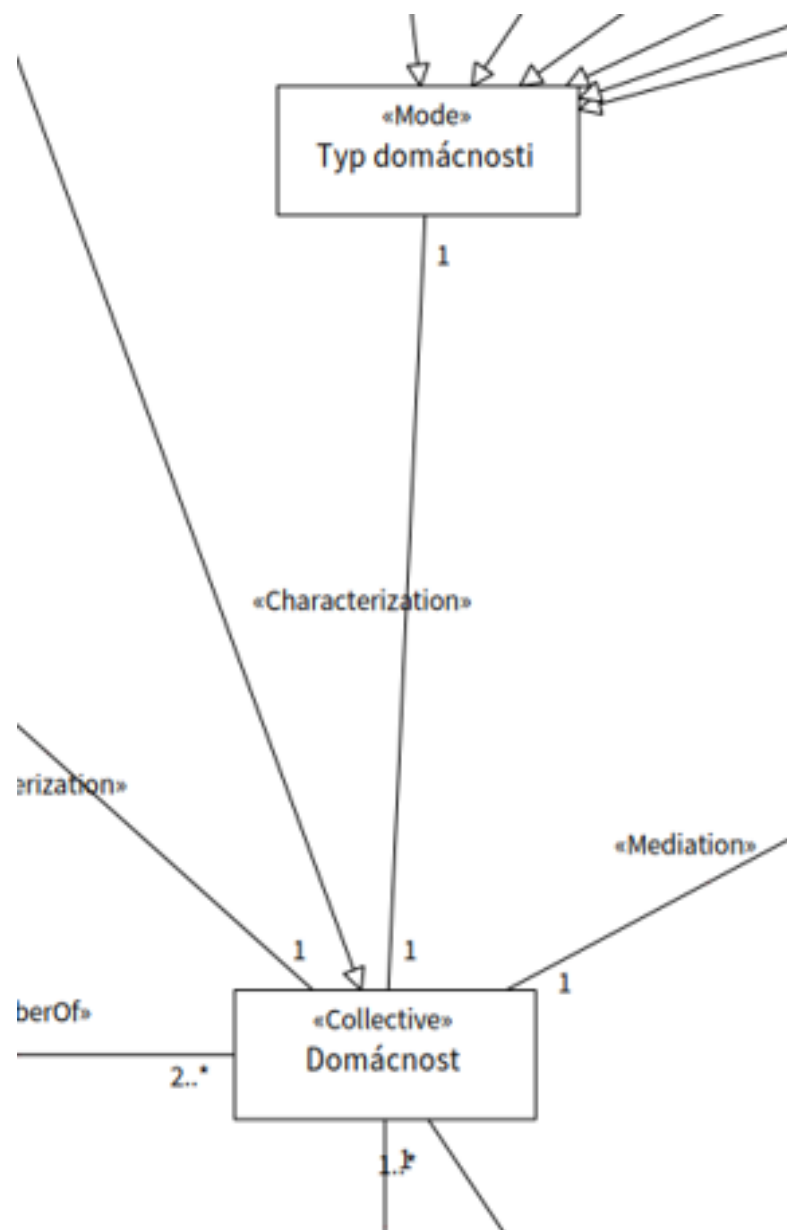


## Ukázky sémantické interoperability

Následující ukázky z ontologických konceptuálních modelů ukazují příklady ontologické analýzy, která vedla k zajištění sémantické interoperability skrze heterogenní datové sady.

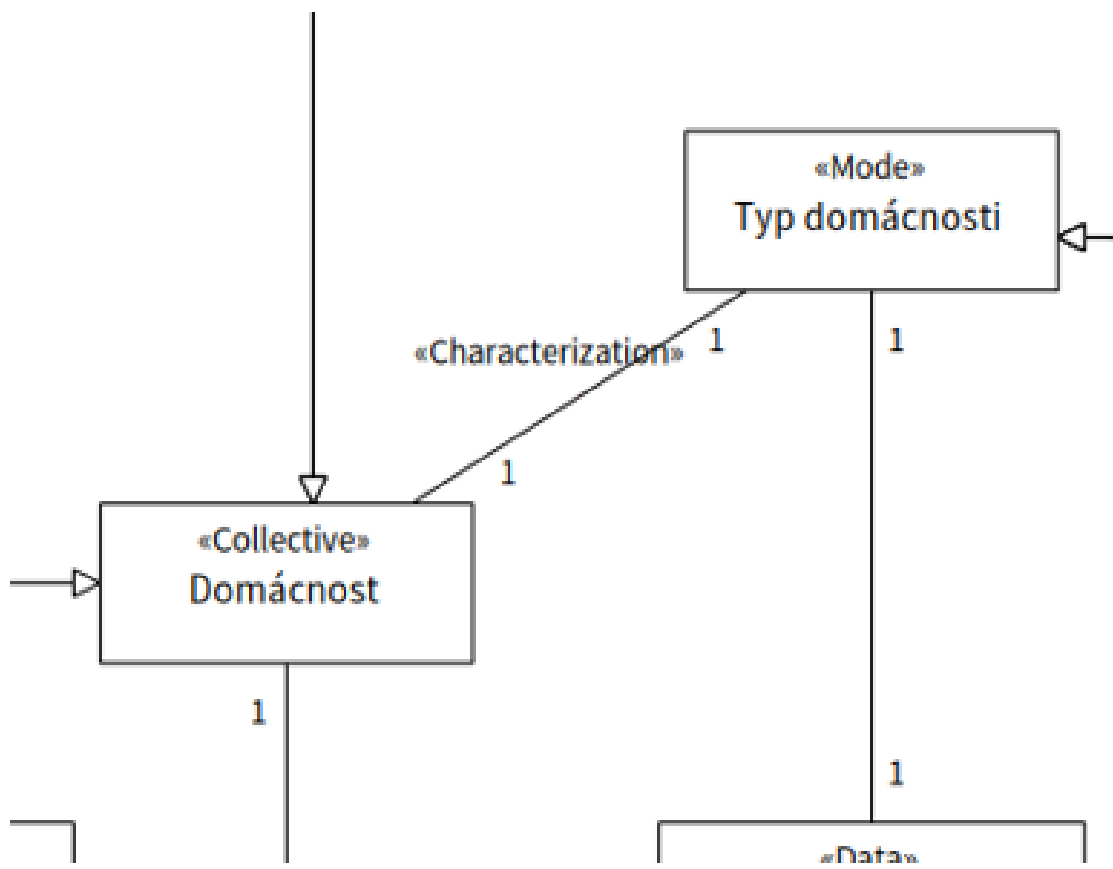


■ **Obrázek A.1** Entita státu v modelu eu-vyroba-a-obchod-s-primarnimi-produkty-bunicina-papir-a-lepenka

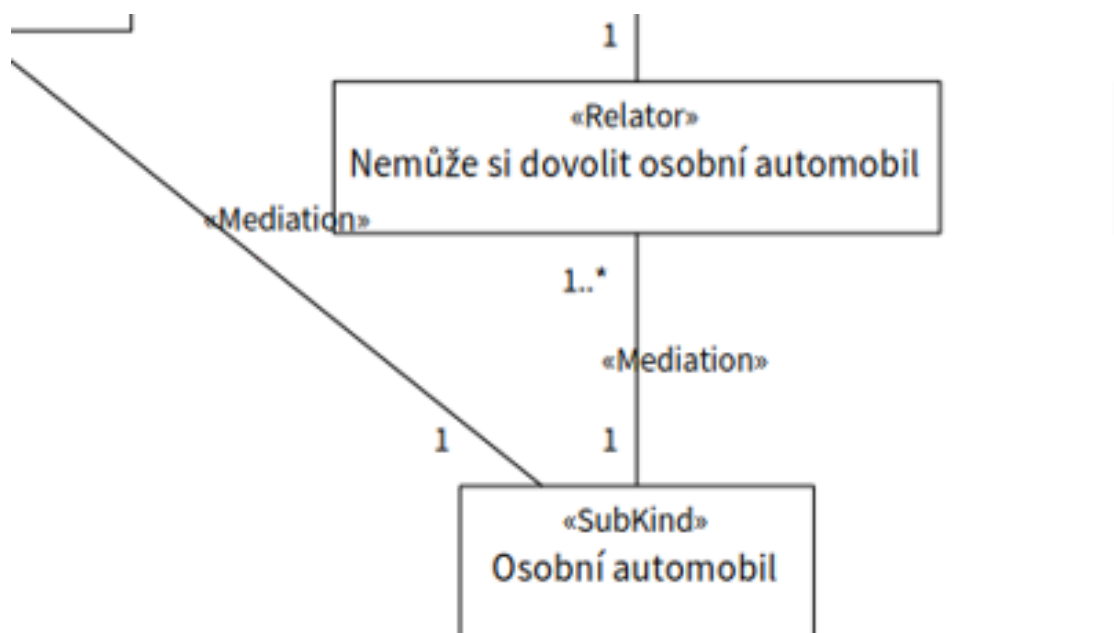


■ Obrázek A.2 Entita domácnosti v modelu eu-domácnosti-typ-pripojenni-internetu

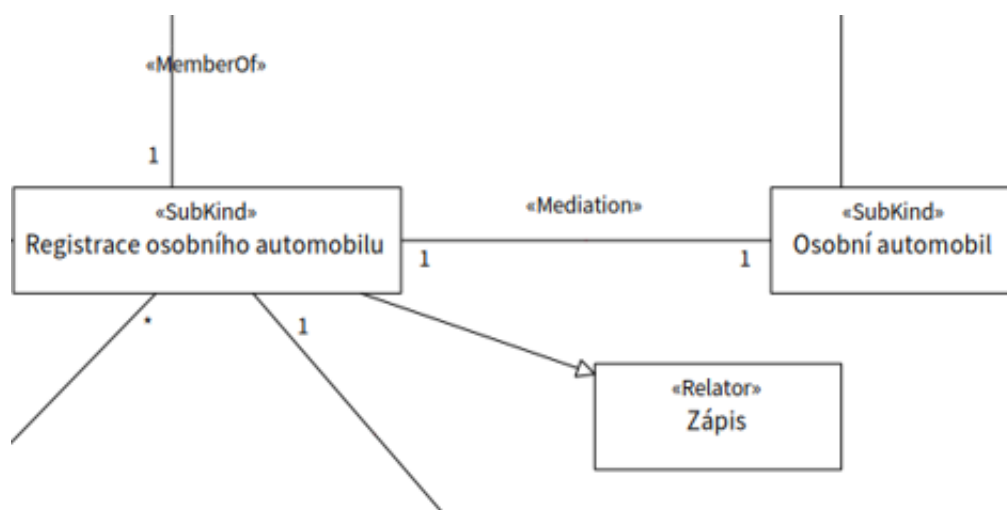




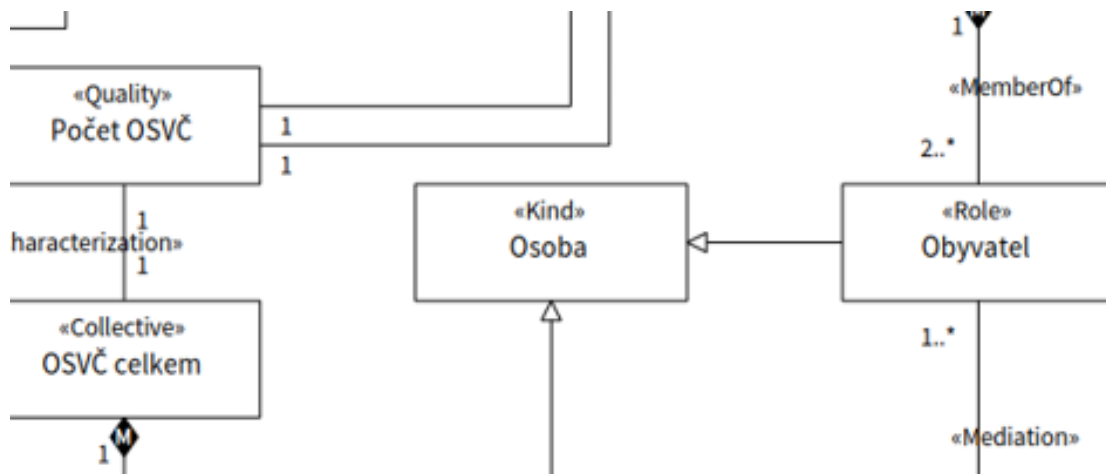
■ Obrázek A.3 Entita domácnosti v modelu eu-osoby-kttere-si-nemohou-dovolit-osobni-automobil



■ **Obrázek A.4** Entita osobního automobilu v modelu eu-osoby-ktere-si-nemohou-dovolit-osobni-automobil



■ **Obrázek A.5** Entita osobního automobilu v modelu eu-registrace-aut-podle-pohonu



■ **Obrázek A.6** Entita osoby v modelu prehled-podnikatelu-a-zivnosti



# Bibliografie

1. REMMARK, a.s. *Remmark* [online]. [B.r.]. [cit. 2022-11-08]. Dostupné z: <http://www.remark.cz/>.
2. HEILER, Sandra. Semantic interoperability. *ACM Computing Surveys*. 1995, roč. 27, č. 2, s. 271–273. Dostupné z DOI: 10.1145/210376.210392.
3. GUIZZARDI, Giancarlo. Ontology, Ontologies and the “I” of FAIR. *Data Intelligence*. 2020, roč. 2, č. 1-2, s. 181–191. ISSN 2641-435X. Dostupné z DOI: 10.1162/dint\_a\_00040.
4. VERDONCK, Michaël; GAILLY, Frederik; PERGL, Robert; GUIZZARDI, Giancarlo; MARTINS, Beatriz; PASTOR, Oscar. Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study. *Information Systems*. 2019, roč. 81, s. 92–103. ISSN 0306-4379. Dostupné z DOI: <https://doi.org/10.1016/j.is.2018.11.009>.
5. *GO FAIR* [online]. GO FAIR, 2023 [cit. 2023-04-28]. Dostupné z: <https://www.go-fair.org/go-fair-initiative/>.
6. WILKINSON, Mark D. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*. 2016, roč. 3, č. 1. ISSN 2052-4463. Dostupné z DOI: 10.1038/sdata.2016.18.
7. *FAIRification process* [online]. GO FAIR, 2023 [cit. 2023-04-28]. Dostupné z: <https://www.go-fair.org/fair-principles/fairification-process/>.
8. HURTADO, Guillermo; NUDLER, Oscar. *The furniture of the world: Essays in ontology and metaphysics* [online]. Leiden: Brill, 2012 [cit. 2023-04-12]. ISBN 978-94-012-0779-9. Dostupné z DOI: /10.1163/9789401207799.
9. POLI, Roberto. Descriptive, Formal and Formalized Ontologies. In: *FISSETTE, Denis, Husserl's Logical Investigations Reconsidered*. Dordrecht: Springer Netherlands, 2003, s. 183–210. ISBN 978-94-017-0207-2. Dostupné z DOI: 10.1007/978-94-017-0207-2\_12.
10. HUSSERL, Edmund. *Logical Investigations Volume 1*. Sv. 1. Oxford: Routledge: Routledge, 2001. ISBN 0-415-24189-8.
11. MEALY, G.H. *Another look at data*. Los Calamitos: EEE Computer Society, 1967. Dostupné z DOI: 10.1109/AFIPS.1967.112.
12. FONSECA, Frederico T.; MARTIN, James E. Learning The Differences Between Ontologies and Conceptual Schemas Through Ontology-Driven Information Systems. *Journal of the Association for Information Systems*. 2007, roč. 8, s. 4. Dostupné z DOI: 10.17705/1jais.00114.
13. GUIZZARDI, Giancarlo; WAGNER, Gerd. A Unified Foundational Ontology and some Applications of it in Business Modeling. In: 2005, s. 129–143. ISBN 9781591403418. Dostupné z DOI: 10.4018/978-1-59140-339-5.ch013.

14. GUIZZARDI, Giancarlo; WAGNER, Gerd. Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages. In: *Theory and Applications of Ontology: Computer Applications*. Ed. POLI, Roberto; HEALY, Michael; KAMEAS, Achilles. Dordrecht: Springer Netherlands, 2010, s. 175–196. ISBN 978-90-481-8847-5. Dostupné z DOI: 10.1007/978-90-481-8847-5\_8.
15. *OntoUML OntoUML Community Portal* [online]. 2017. [cit. 2023-03-26]. Dostupné z: <https://ontouml.org/ontouml/>.
16. GUIZZARDI, Giancarlo; BOTTI BENEVIDES, Alessander; FONSECA, Claudenir M.; PORELLO, Daniele; ALMEIDA, João Paulo A.; PRINCE SALES, Tiago; BORGIO, Stefano; GALTON, Antony; KUTZ, Oliver. UFO: Unified Foundational Ontology. *Applied Ontology*. 2022, roč. 17, č. 1, s. 167–210. ISSN 1570-5838. Dostupné z DOI: 10.3233/A0-210256.
17. MEDVIDOVIC, Nenad; ROSENBLUM, David S.; REDMILES, David F.; ROBBINS, Jason E. Modeling Software Architectures in the Unified Modeling Language. *ACM Transactions on Software Engineering and Methodology*. 2002, roč. 11, č. 1, s. 2–57. ISSN 1049-331X. Dostupné z DOI: 10.1145/504087.504088.
18. CLARK, Anthony; EVANS, Andy. Foundations of the Unified Modeling Language. *2nd BCS-FACS Northern Formal Methods Workshop*. 1997, s. 14–15.
19. QUATRANI, Terry. Introduction to the Unified modeling language. *A technical discussion of UML*. 2003, roč. 6, č. 11, s. 03.
20. GUIZZARDI, Giancarlo; WAGNER, Gerd; ALMEIDA, João Paulo A.; GUIZZARDI, Renata. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Applied Ontology*. 2015, roč. 10, s. 259–271.
21. GUIZZARDI, Giancarlo. *Ontological foundations for structural conceptual models*. Enschede: CTIT, Telematica Instituut, 2005. 416 s. Telematica Instituut Fundamental Research Series, no. 015 (TI/FRS/015). CTIT PhD Thesis Series, 2005. No. 05-74. ISBN 90-75176-81-3. ISSN 1381-3617. Dostupné také z: <http://doc.utwente.nl/50826/>.
22. ŘEPA, Václav. Application of Modelling Methods from Informatics in Evidence Based Health Care. In: ANDERSSON, B.; JOHANSSON, B.; CARLSSON, S.; C. BARRY, M. Lang; LINGER, H.; SCHNEIDER, & C. (ed.). *Designing Digitalization (ISD2018 Proceedings)*. 2018. ISBN 978-91-7753-876-9.
23. *Towards OntoUML for Software Engineering: Transformation of OntoUML into Relational Databases*. 2017. Dostupné také z: <https://courses.fit.cvut.cz/BI-KOM/files/Rybola-PhDThesis.pdf>. Dis. pr.
24. GUIZZARDI, Giancarlo; FONSECA, Claudenir M.; BENEVIDES, Alessander Botti; ALMEIDA, João Paulo A.; PORELLO, Daniele; SALES, Tiago Prince. Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0. In: TRUJILLO, Juan C.; DAVIS, Karen C.; DU, Xiaoyong; LI, Zhanhuai; LING, Tok Wang; LI, Guoliang; LEE, Mong Li (ed.). *Conceptual Modeling*. Cham: Springer International Publishing, 2018, s. 136–150. ISBN 978-3-030-00847-5.
25. *Identity* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/theory/identity.html?highlight=identity>.
26. *Kind* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/kind/index.html>.

27. *Subkind* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/subkind/index.html>.
28. *Relator* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/relator/index.html>.
29. *Role* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/role/index.html>.
30. *Phase* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/phase/index.html>.
31. *Category* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/category/index.html>.
32. *RoleMixin* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/rolemixin/index.html>.
33. *PhaseMixin* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/phasemixin/index.html>.
34. *Mixin* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/nonsortals/mixin/index.html>.
35. *Introduction* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/intro.html>.
36. *Formal* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/formal/index.html>.
37. *Material* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/material/index.html>.
38. *Collective* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/collective/index.html>.
39. *Quantity* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/sortals/quantity/index.html>.

40. *Containment* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/containment/index.html>.
41. *SubQuantityOf* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/relationships/aggregations/subQuantityOf/index.html>.
42. *Mode* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/aspects/mode/index.html>.
43. *Quality* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2018 [cit. 2023-04-30]. Dostupné z: <https://ontouml.readthedocs.io/en/latest/classes/aspects/quality/index.html>.
44. PERGL, Robert; UHNÁL, Peter. The OpenPonk modeling platform. In: [online]. Department of Software Engineering Faculty of Information Technology Czech Technical University in Prague, 2016 [cit. 2023-04-30]. Dostupné z: [http://esug.org/data/ESUG2016/IWST/Papers/IWST\\_2016\\_paper\\_25.pdf](http://esug.org/data/ESUG2016/IWST/Papers/IWST_2016_paper_25.pdf).
45. *OpenPonk* [online]. Centrum pro konceptuální modelování a implementace, Fakulta informačních technologií, České vysoké učení technické v Praze, 2016 [cit. 2023-04-30]. Dostupné z: <https://ccmi.fit.cvut.cz/tools/openponk/>.
46. *OpenPonk modeling platform* [online]. Fakulta informačních technologií, České vysoké učení technické v Praze, 2022 [cit. 2023-04-30]. Dostupné z: <https://openponk.org/>.
47. BARCELLOS, Monalessa; FALBO, Ricardo; FRAUCHES, V.G.V. Towards a measurement ontology pattern language. In: GUIZZARDI, Giancarlo; PASTOR, Oscar; WAND, Yair; DE CESARE, Sergio; GAILLY, Frederik; LYCETT, Mark; PARTRIDGE, Chris (ed.). *Ontologies in Conceptual Modeling and Information Systems Engineering*. 2014, sv. 1301.



# Obsah přiloženého média

readme.txt .....	stručný popis obsahu média
src	
├─ modely .....	konceptuální modely klíčových domén
├─ sablony .....	konceptuální modely šablon
├─ text .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text .....	text práce
├─ 2023-bp-peckaja2.pdf .....	text práce ve formátu PDF