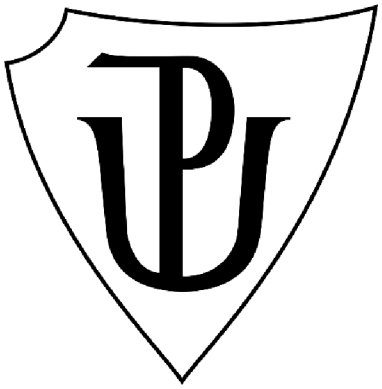
UNIVERZITA PALACKÉHO V OLOMOUCI

Přírodovědecká fakulta

Katedra biochemie



Zvýšení interoperability databáze MolMeDB za pomoci technologií sémantického webu

DIPLOMOVÁ PRÁCE

Autor:	**Bc. Dominik Martinát**
Studijní program:	N1406 Biochemie
Studijní obor:	Bioinformatika
Forma studia:	Prezenční
Vedoucí práce:	**doc. RNDr. Karel Berka, Ph.D.**
Konzultant:	**RNDr. Jakub Galgonek, Ph.D.**
Rok:	2021/2022

Prohlašuji, že jsem diplomovou práci vypracoval/a samostatně s vyznačením všech použitých pramenů a spoluautorství. Souhlasím se zveřejněním diplomové práce podle zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů. Byl/a jsem seznámen/a s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, ve znění pozdějších předpisů.

V Olomouci dne ....

Podpis studenta

Poděkování

Rád bych poděkoval svému vedoucímu doc. RNDr. Karlu Berkovi, Ph.D. za vedení práce a poskytnutí kontaktů na vědce aktivní v oboru RDF databází. Dále bych rád poděkoval Mgr. Jakubu Juračkovi za asistenci při práci s webovou aplikací MolMeDB a RNDr. Jakubu Galgonkovi, Ph.D. za cenné rady při navrhování RDF schématu a za přidání MolMeDB datasetu do IDSM. Infrastruktuře ELIXIR-CZ pak musím poděkovat za poskytnutí kapacit pro překlad datasetu do RDF a SPARQL službu napojenou na IDSM. Práce byla vypracována jako součást grantového projektu DSGC-2021-0060 *FunGIM - Effects of Functional Groups on Interactions with Membranes*.

Bibliografická identifikace

Jméno a příjmení autora	Bc. Dominik Martinát
Název práce	Zvýšení interoperability databáze MolMeDB za pomoci technologií sémantického webu
Typ práce	Diplomová
Pracoviště	Katedra biochemie
Vedoucí práce	doc. RNDr. Karel Berka, Ph.D.
Rok obhajoby práce	2022

Abstrakt

Databáze MolMeDB vznikla za účelem shromáždění informací o interakcích molekul s membránami a transportérovými proteiny. Tento typ informací je dosti specializovaný, ale v kontextu tvořeném dalšími poznatky v oblasti *Life Sciences*, mohou mít tyto informace značný přínos. Jedním ze způsobů zasazení informací v databázi do kontextu je propojení s dalšími bioinformatickými databázemi pomocí technologií sémantického webu. Tato práce se zabývá studiem využití těchto technologií pro zvýšení interoperability bioinformatických databází. Součástí této práce je vytvoření RDF verze databáze MolMeDB.

Klíčová slova	molekula, membrána, transportér, databáze, sémantický web, RDF, URI, ontologie, SPARQL, interoperabilita, MolMeDB, IDSM
Počet stran	74
Počet příloh	1
Jazyk	Český

Bibliographical identification

Autor's first name and surname	Bc. Dominik Martinát
Title	Increasing the Interoperability of MolMeDB Database Using Semantic Web Technologies
Type of thesis	Diploma
Department	Department of Biochemistry
Supervisor	doc.RNDr. Karel Berka, Ph.D.
The year of presentation	2022

Abstract

The MolMeDB database was created to gather information on the interactions of molecules with membranes and transporter proteins. This type of information is quite specialized, but in the context of other knowledge in the field of Life Sciences, this information can have a significant benefit. One way to put information in a database into context is to link it to other bioinformatics databases using semantic web technologies. This thesis deals with the study of the use of these technologies to increase the interoperability of bioinformatics databases. Part of the thesis is to create an RDF version of the MolMeDB database.

Keywords	molecule, membrane, transporter, database, semantic web, RDF, URI, ontology, SPARQL, interoperability, MolMeDB, IDSM
Number of pages	74
Number of appendices	1
Language	Czech

Obsah

1. ÚVOD 1

2. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY 2

2.1 MolMeDB 2

2.2 Sémantický web 3

2.3 RDF 6

2.4 http URI 8

2.4.1 Perzistence URI 8

2.4.2 Dereference URI 9

2.4.3 CURIE 11

2.5 Ontologie a modelování dat 12

2.6 SPARQL 1.1 16

2.7 RDF databáze 22

2.7.1 Integrace datasetů 23

3. PRAKTICKÁ ČÁST 26

3.1 Použité technologie 26

3.1.1 Ontology Lookup Service 26

3.1.2 Ontologie 26

3.1.3 R2RML 29

3.1.4 SPARQLlib 32

3.1.5 REST API 32

3.2 Schéma RDF datasetu 34

3.2.1 Sloučeniny 36

3.2.2 Membránové a transportérové interakce 38

3.2.3 Publikace 46

3.3 Slovník MolMeDB 49

3.3.1 Třídy endpointů transportérových měření 49

3.3.2 Třídy endpointů membránových měření 49

3.3.3 Ostatní třídy 50

3.3.4 Individua jednotek 50

3.4 Generování a publikace RDF datasetu 51

3.5 Dereference 52

4. VÝSLEDKY 57

4.1 Propojená data 58

4.2 Příklady federovaného dotazování 59

5. DISKUZE 64

6. ZÁVĚR 65

7. CONCLUSION 66
8. LITERATURA 67
9. SEZNAM POUŽITÝCH SKRATEK 72
10. PŘÍLOHY 74

CÍLE PRÁCE

- Vypracování literární rešerše v oblasti datového modelu sémantického webu a jeho využití v bioinformatických databázích.
- Výběr, popř. vytvoření, vhodných ontologií pro vyjádření dat z databáze MolMeDB v modelu sémantického webu pomocí metod Resource Description Framework (RDF).
- Implementace převodu dat MolMeDB z relačního modelu do modelu sémantického webu.
- Propojení databáze MolMeDB s ostatními databázemi.
- Provedení testu funkčnosti a uvedení příkladů možného využití sémantického propojení MolMeDB a ostatních databází.

1. ÚVOD

Moderní vědecké metody produkují velké množství dat, ze kterého plyne problém s jejich uchováváním, systematickou organizací a vůbec s možnostmi efektivní orientace v záplavě dostupných měření a údajů. Z těchto důvodů jsou zřizovány elektronické databáze, ve kterých jsou obvykle data uchovávána v jednotných strojově čitelných formátech umožňujících značnou míru automatizace provádění operací nad daty nutných pro efektivní práci s nimi. Toho je využíváno například při vytváření nástrojů usnadňujících práci v datovém prostoru (např. v případě biologických databází vyhledávání sekvenčních motivů), nebo externích aplikací přistupujících k uloženým údajům za pomocí programovacích rozhraní (API).

V současnosti je mnoho databází volně přístupných z internetu, čímž umožňují komunikovat výsledky v rámci širší vědecké obce a zpřístupňují je pro další výzkum. Už jen v oblasti biologických dat toto množství stále narůstá. To přináší nové příležitosti, ale i výzvy, které v mnoha ohledech nejsou nepodobné těm, jaké přineslo množství získávaných dat. Bioinformatické databáze shromažďují různé druhy informací, např. sekvence, trojrozměrné struktury nebo interakce proteinů. Kombinace informací z různých databází pak umožňují z již získaných dat odvozovat nové závěry, srovnávat dostupná data z různých zdrojů a využívat tato data v aplikacích usnadňujících orientaci odborníkům i laikům v daných problematikách.

Výzvu pak představuje roztříštěnost různých druhů informací uložených mezi různými databázemi, využívající různé datové formáty, což ztěžuje nejen orientaci lidského uživatele, ale hlavně i strojové zpracování dat napříč databázemi. V posledních desetiletích provozovatelé databází usilují o zvýšení interoperability využitím datového modelu sémantického webu. Příkladem mohou být databáze ChEMBL (Willighagen et al., 2013) a PubChem (Fu et al., 2015).

Cílem této práce je aplikovat technologie sémantického webu pro zvýšení interoperability databáze membránových interakcí Molecule on Membrane Database (MolMeDB) a následné využití propojení dat se záznamy v jiných databázích využívajících model sémantického webu.

2. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

2.1 MolMeDB

MolMeDB (Molecules on Membranes Database) je databáze interakcí malých molekul s membránami a membránovými transportéry vyvíjená na Univerzitě Palackého v Olomouci, která je veřejně dostupná na adrese https://molmedb.upol.cz/. Údaje o těchto interakcích byly před vznikem této databáze dostupné v roztříštěné formě v literatuře, což značně ztěžovalo práci s nimi. Databáze MolMeDB byla zřízena za účelem shromažďování těchto dat a jejich veřejného poskytování ve standardizovaných formátech.

Krom záznamů o výsledcích měření či výpočtů zmíněných interakcí obsahuje tato databáze také údaje o malých molekulách, membránách, transportérech, metodách měření a zdrojích dat. Záznamy jsou uloženy v relačním databázovém systému MySQL. Vyhledávání je možno provádět pomocí názvů molekul, metod a transportérů, SMILES molekul a Uniprot ID transportérů (Juračka et al., 2019). Ke květnu 2022 MolMeDB obsahuje údaje o téměř milionu interakcí pro půl milionu molekul měřených či vypočtených na 40 různých membránách za užití 52 různých metod.

Webová aplikace MolMeDB je postavena na MVC (*Model-View-Controller*) architektuře (Juračka, 2020), kdy *Controller* je hlavní část zpracovávající požadavek přicházející na server. Tato část může využít *Model* komponenty k získání požadovaných databázových dat a je poté zobrazen pomocí poslední komponenty *View* (Pitt, 2012).

Pro MolMeDB bylo vytvořeno REST API za účelem umožnění automatizovaného přístupu k datům (Juračka, 2020). V současnosti je REST API rozšiřováno o nové funkcionality, mimo jiné i v souvislosti s vytvářením RDF reprezentace dat.

2.2 Sémantický web

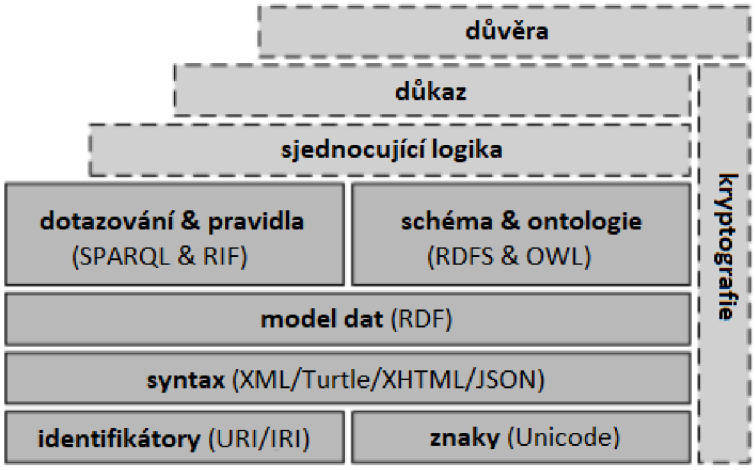
Výraz sémantický web, podobně jako umělá inteligence, může označovat jak oblast výzkumu, tak informační artefakt, který by měl být jejím cílem (Hitzler, 2021). Svou vizi pro sémantický web představili Berners-Lee et al. (2001) ve svém článku pro Scientific American, kde jej popsali jako nástavbu klasického webu, kdy klasický web je sítí dokumentů a sémantický web by měl být sítí dat. Ta by měla být přístupná ve strojově čitelných formátech a jejich sémantika by měla být popsána za pomocí ontologií, ve smyslu kolekcí strojově čitelných tvrzení, které udávají vztahy mezi koncepty a logická pravidla pro rozhodování nad nimi. Ve své finální podobě by pak mělo jít o masivní decentralizovanou síť dat procházející celým webem, umožňující zodpovídání komplexních dotazů a dokazování faktů z distribuovaných informací. Tuto síť by pak využívali softwaroví agenti, kteří by mohli bez lidského zásahu sami plnit složité úkoly vyžadující vzájemnou propojitelnost dat.

Tato vize je stále otázkou budoucnosti, nicméně již existují dílčí výsledky v této oblasti, často umožněné snahou komunit, které vzaly tuto vizi aspoň částečně za svou. Velký vliv na rozvoj technologií sémantického webu mělo vydání standardů jako RDF, OWL a SPARQL konsorciem W3C (World Wide Web Consortium), které položily syntaktické (a částečně sémantické) základy pro práci s daty v sémantickém webu (Hitzler, 2021).

V oblasti vytváření ontologií a využití technologií sémantického webu pro integraci a management dat byla komunita okolo oblasti *Life Sciences* jednou z prvních aktivních (Shadbolt et al., 2006). Zde některé projekty jako Gene Ontology a SNOMED-CT dokonce časově předcházejí sémantický web (Hitzler, 2021). Nadále aktivní zůstávají komunity okolo ontologií jako např. OBO Foundry (Smith et al., 2007) a prominentní bioinformatické databáze jako např. PubChem (Fu et al., 2015), ChEMBL (Willighagen et al., 2013) nebo ChEBI (Hastings et al., 2016) nabízejí svá data ve formátu RDF.

Mimo oblast *Life Sciences* nachází technologie sémantického webu uplatnění např. v rozvoji umělé inteligence, internetu věcí, nebo v průmyslu (Patel & Jain, 2021), avšak stále před tímto odvětvím stojí celá řada výzev (Hitzler, 2021).

Architektura sémantického webu se často znázorňuje jako diagram nazývaný **věž sémantického webu** (*semantic web tower*, někdy také *cake* nebo *stack*), kde každá část



Obr. 1: Věž sémantického webu popisující architekturu sémantického webu. Převzato a upraveno z Hogan (2016).

reprezentuje technologický díl podílející se na stavbě sémantického webu. Tento diagram se často objevuje v textech zabývajících se sémantickým webem v mnoha podobách reflektujících aktuální stav odvětví, nebo úhel pohledu autora. Na obrázku 1 je uvedena verze a popis dle Hogana (2016).

Dvě spodní vrstvy jsou společné s klasickým webem. Zahrnují kódování **znaků** pomocí Unicode a systém pro **identifikaci** zdrojů. Strojová čitelnost je umožněna formálně definovanými gramatikami s definovanou syntaxí. Často se využívají již zavedené formáty (XML, JSON, XHTML) umožňující kompatibilitu s nástroji klasického webu, ale zavádějí se i formáty specializované (Turtle).

Vyšší vrstvy jsou specifické pro sémantický web. Pro výměnu dat mezi stroji generickým způsobem musí být zaveden obecný **model dat** popisující reprezentaci informací nezávisle na jejich doméně či charakteru a na zvolené syntaxi zápisu. Základním modelem sémantického webu je RDF (viz další kapitola).

Schémata a ontologie dávají zapsaným datům význam a umožňují vytvářet o nich tvrzení. Pro jejich popis jsou definovány formální jazyky (RDFS a OWL) poskytující metaslovníky s dobře definovanou sémantikou.

Pracovat s takto uloženými a popsanými daty umožňuje **dotazování** a vytváření **pravidel**. Jejich výsledkem může být vyhledání informace v obsahu, generování nového obsahu z již existujícího, stanovení omezení na dataset, či definování operací nad daty. Současným standardem pro dotazování je jazyk SPARQL a pro využívání pravidel RIF.

Ostatní vrstvy jsou stále ve spekulativní fázi a neexistují pro ně žádné obecně přijímané standardy. **Sjednocující logika** by měla zajistit interoperabilitu dotazování, pravidel a ontologií, tak aby např. bylo možné dotazování zohledňující ontologické interpretace. Agenti samostatně pracující na sémantickém webu by pak měli být schopni vydat **důkaz** popisující, jak ke svým výsledkům došli pro validaci klientem. Z něj vychází vrstva **důvěry**, což v tomto kontextu znamená, identifikace důvěryhodných zdrojů, což umožňuje validovat výsledek pomocí důkazu. Agenti by měli být schopni sami odhadnout míru důvěryhodnosti ostatních podle dostupných informací.

Kryptografie prorůstá ostatními vrstvami. Bezpečné šifrování, ověřování identity atd. jsou pro vizi sémantického webu stejně důležité jako pro web klasický. Mnohé existující kryptografické technologie jako RSA nebo TSL/SSL jsou použitelné i pro sémantický web.

2.3 RDF

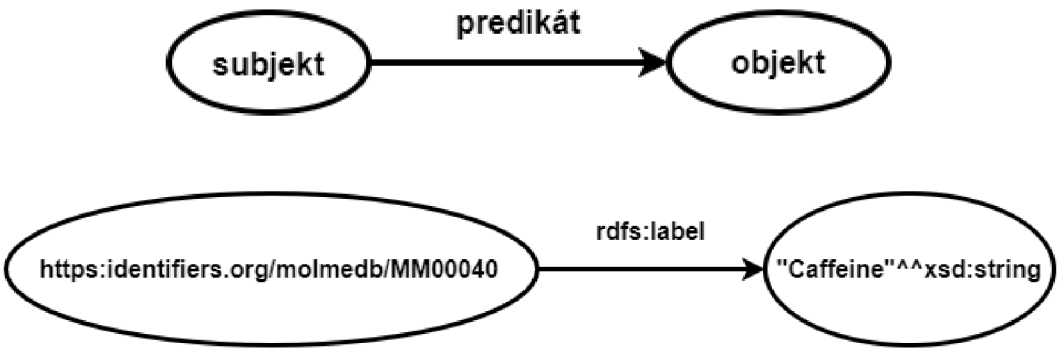
Resource Description Framework (RDF), česky „systém popisu zdrojů“, je model popisující informace v síti. Ústřední strukturou abstraktní syntaxe je množina trojic skládající se ze **subjektu**, **predikátu** a **objektu**. Kolekce trojic tvoří **RDF graf**.

RDF graf může být znázorněn jako diagram, kde je každá trojice reprezentována dvojicí uzlů spojených orientovanou hranou, kde hrana představuje predikát a vede od subjektu k objektu. (Obr. 2). Uzly je možné rozdělit do 3 skupin: **IRI** (Internationalized Resource Identifier), **literál** nebo **anonymní uzel**. Subjekt může být IRI nebo prázdný uzel, predikát musí být IRI a objekt může být IRI, prázdný uzel nebo literál.

IRI slouží jako textové označení prvků uvažovaného univerza kterým se říká zdroje (referent). Zdrojem může být cokoli o čem můžeme něco tvrdit, včetně fyzických objektů, abstraktních konceptů, dokumentů, čísel a znakových řetězců. IRI je znakový řetězec v Unicode mající globální platnost. Jedno IRI tedy vždy značí stejný referent (Cyganiak et al., 2014). IRI jsou např. i internetové odkazy (Bernes-Lee et al., 2004).

Literálem je konkrétní hodnota datového typu určujícího, jakých hodnot může nabývat, jako např znakový řetězec, číslo, nebo datum. Speciálním druhem literálu je znakový řetězec značící text v přirozeném jazyce (Cyganiak et al., 2014).

Anonymní uzel označuje zdroj bez toho, aby jej přímo jmenoval.



Obr. 2: RDF graf se dvěma uzly (subjektem a objektem) spojenými hranou (predikátem) (nahoře). Vyjádření vztahu mezi záznamem URL v databázi MolMeDB a jeho anglickým názvem v RDF. (dole)

Příkladem, jak by mohla vypadat trojice z RDF databáze MolMeDB, je vyjádření vztahu mezi IRI záznamu pro kofein a názvem látky (Obr. 2 dole), kdy subjekt je IRI, predikát má zkrácenou IRI `rdfs:label` a objekt „Caffeine“ je literál typu string.

Kolekce RDF grafů tvoří **RDF dataset**. V datasetu se nachází vždy právě jeden výchozí graf, který může být prázdný a dále libovolné množství pojmenovaných grafů. Pojmenovaný graf je tvořen svým jménem (IRI nebo anonymní uzel) a obsahem (RDF graf) (Cyganiak et al., 2014). Někdy se pro práci s pojmenovanými grafy rozšiřují trojice na čtveřice, kdy poslední prvek určuje graf, někdy též nazývaný kontext (Allemang et al., 2020).

RDF je samo o sobě abstraktní formát bez definované syntaxe. K zápisu dat slouží formáty jako např. RDF/XML nebo N-Triples. Tato data mohou být přechovávána databázovými systémy nazývanými *triplestores* (případně *quadstores*). K dotazování nad datasety se pak používají dotazovací jazyky, například SPARQL (Allemang et al., 2020).

2.4 http URI

V sémantickém webu se obvykle využívá podmnožina IRI nazývaná **http URI** (Universal Resource Identifiers) (Allemang et al., 2020). Http URI (dále jen URI) hrají významnou roli při integraci dat z různých zdrojů. Pokud se ve dvou grafech vyskytuje stejná URI, můžeme tvrdit, že jde o stejný zdroj. Toho je možné využít i při propojování datasetů. S jedním zdrojem se může pojit i více URI a v takovém případě je určování identity problematičtější. Můžou se využívat odvozovací pravidla (jako např. pokud uzel s URI1 je chemický prvek a má protonové číslo X a uzel URI2 je také chemický prvek se stejným protonovým číslem X, pak URI1 a URI2 identifikují stejný prvek), nebo může být nutný lidský zásah.

Základní forma URI je definována dokumentem RFC 3986 (Berners-Lee et al., 2005). Ve zkratce je URI řetězec v US-ASCII dodržující tento vzor:
`scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`
kde schéma značí specifikaci, které URI podléhá. Volitelná je identifikace autority s možnými přístupovými údaji a číslem portu. Povinná je pak cesta k referentu, za kterou může být položen dotaz nebo odkaz na fragment upřesňující výsledek.

Http URI podléhají HTTP schématu, což jim umožňuje využívat DNS k asociaci jmen se vzdálenými webovými servery, nebo vyjednávání formátu obsahu a přesměrovávání pomocí http protokolu. Běžně se využívají i URI podléhající HTTPS protokolu. **URL** (Universal Resource Locator) jsou podmnožinou URI. Zatímco referenty URL jsou zdroje na webu, referentem URI může být jakýkoli zdroj (Allemang et al., 2020).

IRI dodržují syntax definovaný v dokumentu RFC 3987 (Duerst & Suignard, 2005). Oproti URI, umožňují IRI používat širší množinu znaků Unicode/ISO 10646.

2.4.1 Perzistence URI

URI obecně by měly být navrhovány tak, aby byly perzistentní. Změny URI identifikující nějaký zdroj obecně přinášejí problémy např. pro uživatele a systémy používající URI původní (Berners-Lee, 1998). V případě sémantického webu vyvstávají specifické hrozby v podobě ztrát propojení mezi datasety, nebo omezení, či ztrát funkčnosti systémů používajících slovník, ve kterém došlo ke změnám (Allemang et al., 2020).

Z tohoto důvodu je nutné systém vytváření URI dobře zvážit. Existuje celá řada doporučení pro vytváření nových URI. Obecně by neměly obsahovat informace, které se mohou časem měnit a není vhodné uvádět informace poukazující na softwarovou implementaci, koncovky formátu dokumentů, autorství nebo verzi dokumentu (Berners-Lee, 1998). Naopak například pro vytváření URI zdrojů z již existujících datasetů je vhodným řešením využít již existující stálé identifikátory, jako např. primární klíče z relačních databázových systémů (Allemang et al., 2020), nebo využití webových služeb poskytujících perzistentní URL a přesměrování na adresu dokumentu (Hyland et al., 2014).

Sporné je člověkem čitelné označení referenta v URI. Význam výrazů přirozeného jazyka a pojmenování zdrojů se mohou časem měnit (Berners-Lee, 1998). Mění se i odborné názvosloví, což může být problém specializovaných slovníků. Na druhou stranu bývají čitelné IRI uživatelsky daleko přívětivější (Allemang et al., 2020).

2.4.2 Dereference URI

Jednou z nejdůležitějších vlastností URI je, že mohou být **dereferencovány**, čímž umožňují začít interakci se vzdáleným serverem. Výsledkem dereference by měl být dokument popisující referent (Cyganiak *et al.,* 2014). V případě sémantické webu je vhodné, aby byl dereferencovaný dokument ve formě RDF trojic v nějakém neproprietárním strojově čitelném formátu. Dokument by se měl odkazovat na další zdroje formou jejich URI a tam kde to dává smysl i dalších datasetů (Berners-Lee, 2009).

Správně by se mělo rozlišovat mezi URI zdroje a URL dokumentu, který jej popisuje. Referent a dokument jsou v takovém případě dva různé zdroje, o kterých je možné něco tvrdit a v případě použití společné URI by bylo problematické rozlišit, jestli se tvrzení váže k referentu nebo dokumentu. Z toho důvodu se používají jako řešení **hash URI** nebo **slash URI**. Obě řešení umožňují vyjednávání formátu výsledku pomocí HTTP protokolu.

Hash URI jsou URI obsahující část oddělenou od zbytku řetězce označenou symbolem (#) nazývanou fragment. Fragment identifikuje nějakou část dokumentu. Při dereferenci hash URI požaduje HTTP protokol odstranění fragmentu před zasláním požadavku na server. URI bez fragmentu pak již může být URL dokumentu. Hash URI se stejným prefixem před fragmentem mohou být popsány jedním souborem. Agent tak může vyhodnotit i více URI zasláním jednoho požadavku na dereferenci. Nevýhodou je

nižší přehlednost dokumentu a nutnost stahovat pro referenci celý někdy dosti objemný dokument i pro jedinou URI. Navíc přidání, ubráni nebo úprava popisu některé URI znamená změnu celého dokumentu.

Slash URI využívají funkci přesměrování pomocí HTTP statutu ***303 see other*** a někdy se kvůli tomu mluví i o **303 URI**. Při dereferenci dostane agent od serveru odpověď se statutem 303 a URL na které se dokument popisující zdroj nachází. Agent pak automaticky zašle požadavek pro dereferenci URL a vyhodnotí výsledek. Slash URI umožňují přehlednější popis referentů a požadavek na dereferenci zpracovává jen konkrétní zdroj, který zrovna agent požaduje. Nevýhodou je nutnost zasílání minimálně dvou požadavků pro dereferenci každé URI.

Obecně jsou hash URI vhodné pro menší datasety, kde se nepředpokládají časté změny v popisu referentů nebo kde ke změnám dochází najednou (často jsou takto řešeny slovníky), zatímco slash URI jsou vhodné pro práci s velkými datasety. Stažení celého datasetu by ale pak mělo být umožněno jiným mechanismem, aby nedošlo k zahlcení serveru, např. za pomoci protokolu FTP, či jiných alternativ (Sauermann & Cyganiak, 2008).

Jako ilustrace pro využití těchto možností dereference může sloužit PubChem RDF (Fu et al., 2015). PubChem udržuje svůj vlastní nepříliš rozsáhlý slovník pro který využívá hash URI. Např. http://rdf.ncbi.nlm.nih.gov/pubchem/vocabulary#encodedBy je URI pro vlastnost, která proteinové sekvenci přiřazuje genovou sekvenci, která ji kóduje. Její derefence vede k URL: https://rdf.ncbi.nlm.nih.gov/pubchem/vocabulary.owl.

Pro samotná data pak PubChem využívá slash URI. Kofeinu je například přiřazena URI http://rdf.ncbi.nlm.nih.gov/pubchem/compound/CID2519 jejíž dereference vede na dotaz REST API s výsledkem https://pubchem.ncbi.nlm.nih.gov/rest/rdf/compound/CID2519. Formát výsledného dokumentu může být upřesněn pomocí suffixu URI (např. http://rdf.ncbi.nlm.nih.gov/pubchem/compound/CID2519.ttl) nebo v příznaku *Accept* v hlavičce žádosti o dereferenci.

2.4.3 CURIE

Používání celých URI v textu i formátovaných dokumentech se často ukázalo být pro lidské čtenáře a uživatele nepraktickým a nepřehledným. Z tohoto důvodu byl zaveden syntax **CURIE** umožňující zkrácený zápis tzv. kompaktními URI.

Pro jejich použití se nejprve definují jmenné prostory společných prefixů URI. Například typ a vlastnost mají v RDF slovníku URI `http://www.w3.org/1999/02/22-rdf-syntax-ns#type` a `#Property`. Pro tento slovník se běžně používá prefix rdf a psát tyto URI pak jako `rdf:type` a `rdf:Property`. První část je název jmenného prostoru a druhá část je lokální identifikátor (Birbeck & McCarron, 2010).

Některé standardy pro zápis RDF dokumentů jako Turtle, RDF/XML nebo JSON-LD a jazyky pro páci s RDF jako SPARQL umožňují použití CURIE, vždy však s nutností definice jmenného prostoru (Allemang et al., 2020).

2.5 Ontologie a modelování dat

Při popisu ontologií se často používá Gruberova (1993) definice označující ontologie jako formální explicitní specifikace sdílené konceptualizace. V zásadě to znamená, že ontologie jsou hierarchické struktury tříd popisující uvažované pojmy. Hierarchie je tvořena relací *is-a* (**subsumpce**). Platí že A *is-a* B právě když třída A je užším vymezením třídy B. Ontologie umožňují klást přísnější požadavky pro relaci subsumpce. Třídy mohou být popsány pomocí atributů, na které mohou být kladena další omezení.

Cílem ontologií bývá spíš snaha postihnout obecnější znalosti o nějaké problematice než popsat konkrétní instance tohoto problému a poskytnout tak obecný nástroj pro práci s nimi (Zdráhal, 2013). V rámci modelu sémantického webu jsou ontologie hlavním prostředníkem zajišťujícím integraci, sdílení a nalézání dat. Jednou z hlavních idejí je znovu použitelnost různými uživateli (Hitzler, 2021).

Například ontologie CHEMINF poskytuje termíny pro vyjádření chemických informací, jako je třeba `sio:CHEMINF_000088` pro třídu molekulová hmotnost (Hastings et al., 2011). Tento termín je použitelný v libovolném RDF datasetu, kde se vyskytuje uzel, který je instancí záznamu pro molekulovou hmotnost. Použití stejné ontologie zvyšuje vzájemnou srozumitelnost datasetů a činí ontologii atraktivnější volbou při vytváření nového datasetu. Nic však nebrání tomu, aby se k jednomu zdroji vázalo více pojmů, a tedy aby jiná ontologie definovala vlastní termín pro molekulovou hmotnost (Allemang et al., 2020). S tím se váže komplexní problém vzájemného mapování ontologií. Na tomto poli stále probíhají živé diskuse a prosazují se různé přístupy (Harrow et al., 2019).

Ontologie bývají vyjádřené také v RDF. Pro vyjádření významů pojmů, jejich hierarchie a pravidel pro práci s nimi byly sestaveny jazyky **RDFS** a **OWL 2**, na které je možno nahlížet také jako na ontologie.

RDFS (RDF Schema) je sémantickým rozšířením RDF. Definuje základní pojmy pro konstrukci ontologií, jako třídy a vlastnosti (predikáty) (Brickley & Guha, 2014). Tento systém tříd a vlastností bývá často přirovnáván k objektově orientovanému programování (Allemang et al., 2020). Rozdíl však je v tom, že není definováno, jaké vlastnosti (predikáty) může třída mít, ale jakou třídu zdrojů přijímá vlastnost jako subjekt (vlastnost `rdfs:domain`) a objekt (`rdfs:range`). Také není pevně definováno, jak by měl

systém reagovat v případě nesouladu mezi typem objektu nebo subjektu a definičním oborem, či oborem hodnot predikátu. Například nástroj kontrolující konzistenci dat může upozornit na chybu, ale nástroj pro inferenci může přiřadit problémovému objektu nebo subjektu další třídu.

Základní definovanou třídou je `rdfs:Class` jejímiž instancemi jsou všechny třídy RDF objektů, včetně jí samé. Další základní třídy jsou třídy všech zdrojů (`rdfs:Resource`), predikátů (`rdf:Property`), literálů (`rdfs:Literal`) a všech jejich datových typů (`rdfs:Datatype`). Dále jsou definovány třídy pro některé typy literálů (`rdf:langString`, `rdf:HTML`, `rdf:XMLLiteral`) jako instance `rdfs:Datatype` (Brickley & Guha, 2014).

Zde by bylo vhodné zmínit, že ač například instancemi `rdfs:Class` jsou třídy, často se za instance považují individua. Záleží na použití ontologie, kdy bude pojem považován ještě za **třídu** a kdy za **individuum**. Typicky se udává příklad s biologickou taxonomií zvířat, kdy například ontologie popisující obecné vlastnosti druhů bude pracovat s druhy jako individui nějakého rodu či vyšší taxonomické jednotky. Oproti tomu ontologie pro potřeby zoologické zahrady bude pracovat s druhy jako třídami a individui budou jednotlivé kusy zvířat. Existují i nástroje pro vyjádření tříd jako individuí, obecně se to však za dobrou praxi nepovažuje (Allemang et al., 2020)

Pro popis tříd a jejich hierarchii zavádí RDFS několik vlastností. Hierarchie tříd je vyjádřena pomocí predikátu `rdfs:subClassOf`, který vyjadřuje, že všechny instance subjektové třídy jsou zároveň instancemi objektové třídy. Např. v pomyslné ontologii pro gorily v pražské ZOO by třída gorila byla podtřídou hominid. Gorilák Richard by byl instancí třídy gorila a tím pádem i třídy hominid. Příslušnost prvku třídě vyjadřuje `rdf:type`, kdy subjekt je instancí objektové třídy. V příkladové ontologii by tedy platilo `zoo:Richard rdf:type zoo:Gorila`. Samotné predikáty mají vlastní hierarchii vyjádřenou `rdfs:subPropertyOf`, který vyjadřuje, že pro všechny prvky propojené subjektovým predikátem platí i propojení objektovým. Např. ZOO musí evidovat rodokmeny zvířat. Ontologie z příkladu by třeba obsahovala vlastnost `zoo:maPredka` a podvlastnost `zoo:maOtce`. Mládětem gorily Richarda je Ajabu a v systému je tedy uvedené `zoo:Ajabu zoo:maOtce zoo:Richard`, z čehož se

pomocí rdfs:subPropertyOf dá odvodit zoo:Ajabu zoo:maPredka zoo:Richard.

Mimo výše zmíněné obsahuje RDFS také pojmový aparát pro definici kolekcí, jako jsou např. seznamy a pro anotaci, jako např. rdfs:label pro vyjádření lidsky čitelného popisku a rdfs:comment pro záznam poznámek (Brickley & Guha, 2014).

Pro konstrukci komplexnějších ontologií se používá jazyk **OWL 2** (Web Ontology Language). Ten poskytuje nástroje pro logickou konstrukci složitějších pojmů z jednodušších a pro inferenci nad daty vyjádřenými za pomocí ontologií z tohoto jazyka.

Zavádí například predikáty pro ekvivalenci tříd (owl:equivalentClass), individuí (owl:sameAs) a vlastností (owl:equivalentProperty). Dále definuje množinové operace, čímž umožňuje např. konstruovat nové třídy jako ekvivalentní výsledku nějaké množinové operace. Restrikce pak umožňují definovat třídy pomocí jejich vlastností a hodnot kterých mohou tyto vlastnosti nabývat, případně jejich kardinality.

Podobně OWL 2 definuje pojmy pro popis predikátů jako binárních relací. Například zavádí podtřídy owl:Property podle (i)reflexivity, (a)symetrie nebo tranzitivity. Také se zavádí symetrický predikát owl:inverseProperty pro inverzní vlastnosti.

Tyto a další pojmy se pak využívají v **inferenčních systémech** k odvození nových dat z explicitně zadaných (Hitzler et al 2012). Např. když se vrátíme k fiktivní ontologii pražské zoo, mohli by mít v ontologii definováno zoo:maPredka owl:inverseProperty zoo:maPotomka a v systému uvedeno že zoo:Ajabu zoo:maPredka zoo:Richard, ale ne že Richard má potomka Ajabua. Inference pomocí uvedeného vztahu inverze tento vztah snadno doplní. Podobně je možné například odvodit příslušnost individua k třídě, pokud splňuje její definici.

OWL 2 je velice silný nástroj modelování dat a inference, kvůli čemuž vyžaduje vysokou míru opatrnosti (Hitzler et al., 2012). Neopatrná definice může snadno vést k odvození chybných závěrů v podobném duchu, jako Platonova definice člověka jakožto dvojnožce bez peří vedoucí k Diogenovu odvození, že oškubaná slepice je člověk (Laërtius & Hicks, 1925).

Zavedení RDFS a OWL 2, jakožto i dalších nástrojů vedly k publikací nových verzí starších ontologií (jako Gene Ontology) kompatibilních s nimi i k publikaci ontologií nových (Hitzler, 2021). Dnes existuje velké množství nejrůznějších ontologií. Jen v oblasti biomedicíny uvádí **NCBO BioPortal** téměř tisíc ontologií.

2.6 SPARQL 1.1

SPARQL 1.1 (SPARQL Protocol and RDF Query Language) je systém specifikací poskytující jazyk a protokol pro dotazování a manipulaci s obsahem v podobě RDF grafu na webu, nebo v triplestore. Specifikace představují dotazovací a manipulační jazyky SPARQL a standardy poskytování výsledků dotazů a federované dotazování a protokol transfer dotazů pomocí http protokolu. Další specifikace upravují možnosti výpočtu výsledku pro položený dotaz a způsob jakým mohou být služby založené na SPARQLu anotovány. Nakonec SPARQL 1.1 uvádí specifikaci pro minimální způsoby nakládání s RDF grafy pomocí operací HTTP protokolu a několik testů pro implementace systémů založených na SPARQLu (Aranda et al., 2013).

Dotazovací jazyk SPARQL poskytuje aparát pro pokládání komplexních dotazů nad daty v RDF grafech. Jak může vypadat jednoduchý SPARQL dotaz ukazuje kód 1. SPARQL umožňuje pracovat s kompaktními URI. Jmenné prostory jsou uvedeny na začátku dotazu klíčovým slovem **PREFIX** a poté následuje název jmenného prostoru a prefix URI použitých termínů. URI se v nezkrácené podobě zapisují mezi špičaté závorky.

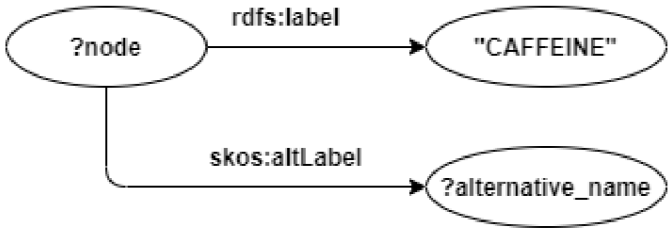
Následuje dotaz, kde je uvedeno, pro které dotazové proměnné chce dotazovatel znát odpověď zde vyjádřený klíčovým slovem **SELECT**. Za klíčovým slovem **WHERE** následuje ve složených závorkách vzor, se kterým se bude RDF dataset prohledávat. Klíčové slovo **LIMIT** udává maximální množství výsledků k návratu (Harris & Seaborne, 2013).

K zadávání dotazů webovým aplikacím slouží tzv. **SPARQL klienti**. Ti převedou dotaz na HTTP žádost a přes tzv. **SPARQL endpoint** ji pošlou službě. Endpoint je URL

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?alternative_name WHERE {
  ?node rdfs:label "CAFFEINE"
  ?node skos:altLabel ?alternative_name
} LIMIT 2
```

Kód 1: SPARQL dotaz na alternativní názvy uzlu s názvem „CAFFEINE“.



Obr. 3: Podgraf odpovídající vzoru z dotazu v kódu 1.

na které služba očekává žádost zaslanou klientem. Ta je po přijetí vyhodnocena a služba vydá HTTP odpověď (Feigenbaum et al., 2013).

Základním mechanismem, kterým SPARQL služba hledá v datasetu odpověď je subgraph matching. V uvedeném příkladu z kódu 1 se hledají všechny podgrafy datasetu odpovídající vzoru (Obr. 3). V příkladovém datasetu (Kód 2) by takto byly nalezeny tři odpovídající podgrafy. Pro všechny by na dotazovou proměnnou byla navázána URI `mol:CHEMBL113`. Proměnná *?alternative_name* by byla v každém podgrafu zvlášť navázána na hodnoty „Theine“, „Coffeine“ a „Methyltheobromine“. Trojice označující typ `mol:CHEMBL113` se ve výsledku neobjeví protože neodpovídá žádné trojici vzoru. Trojice s objektem `mol:CHEMBL25` se také neobjevuje, protože také nesplňuje vzor, tentokrát nemá nikde uvedenou jako hodnotu `rdfs:label` „CAFFEINE“. Pokud by se v datasetu vyskytoval jiný uzel s hodnotou `rdfs:label` „CAFFEINE“ ale nebyl subjektem žádné trojice s predikátem `skos:altLabel`, tak by se do výstupu nedostal rovněž. Pokročilejší implementace SPARQL služby pak mohou při hledání výsledků využít odvozování pomocí RDFS, OWL nebo pravidlových systémů (Glimm & Ogbuji, 2013). Například SKOS ontologie uvádí že `skos:altLabel` `rdfs:subPropertyOf` `rdfs:label` (Miles & Bechhofer 2009). Pokud by v dotazu z kódu 1 byl `skos:altLabel` nahrazen `rdfs:label`, tak by k podgrafům splňujícím vzor přibyl jeden s proměnou *?alternative_name* navázanou na hodnotu „CAFFEINE“. Ostatní by zůstaly také, protože by odvozovací algoritmus usoudil, že `skos:altLabel` je jen speciální případ `rdfs:label`.

```
PREFIX mol: <http://rdf.ebi.ac.uk/resource/chembl/molecule/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cco: <http://rdf.ebi.ac.uk/terms/chembl#>

mol:CHEMBL113 rdf:type cco:Substance .
mol:CHEMBL113 rdfs:label "CAFFEINE" .
mol:CHEMBL113 skos:AltLabel "Theine" .
mol:CHEMBL113 skos:AltLabel "Coffeine" .
mol:CHEMBL113 skos:AltLabel "Methyltheobromine".
mole:CHEMBL25 rdfs:label "ASPIRIN" .
mol:CHEMBL25 skos:AltLabel "o-acetylsalicylic Acid" .
```

Kód 2: Podgraf datasetu RDF CHEMBL (Willighagen et al., 2013) pro vyhodnocení příkladových dotazů (kódy 1 a 3 až 6) zapsaný v syntaxu Turtle.

Odpověď služby je klientovi poskytnuta v požádaném formátu. Server vrátí **stavový kód** podle úspěchu či neúspěchu vyhodnocení definovaný HTTP protokolem (Fielding & Reschke, 2014).

V případě neúspěchu je stavový kód 40x, pokud byl klientem odeslán nesprávný dotaz, nebo 500, pokud služba selhala při jeho vyhodnocování. Tělo odpovědi záleží na implementaci služby.

Úspěšná odpověď by měla mít 20x, a v těle odpovědi by měl být navrácen výsledek dotazu. Pro dotaz uvedený klíčovým slovem SELECT by měl být výsledek ve formátu XML, JSON nebo CSV/TSV (Feigenbaum et al., 2013). To by byl i případ příkladového dotazu (Kód 1). Zde by odpověď obsahovala dvě z hodnot navázaných na proměnnou *?alternative_name*. Běžné je, že SPARQL klienti pro lidské uživatele prezentují výstup jako tabulku (Harris & Seaborne, 2013).

SPARQL umožňuje i jiné typy dotazů, uvedené klíčovými slovy ASK, CONSTRUCT a DESCRIBE. Klíčovým slovem **ASK** se uživatel ptá na existenci podgrafu splňující vzor

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

ASK {?node rdfs:label "CAFFEINE"}
```

Kód 3: SPARQL dotaz na alternativní názvy uzlu s názvem „CAFFEINE“.

v datasetu. Výsledkem je booleovská hodnota vydaná stejným způsobem jako odpověď na SELECT (Harris & Seaborne, 2013). Kód 3 je příkladem dotazu na existenci uzlu který má hodnotu `rdfs:label` „CAFFEINE . V příkladovém datasetu by odpověď byla *true*.

CONSTRUCT a DESCRIBE slouží k vytvoření množiny trojic a tělo odpovědi obsahuje v tomto případě dokument popisující RDF graf v některé syntaxi (Feigenbaum et al., 2013).

CONSTRUCT má, jak již název napovídá, konstrukční funkci. Za klíčovým slovem následuje vzor vrácených trojic, který může obsahovat konkrétní hodnoty i dotazové proměnné. Po klíčovém slově WHERE následuje popis vzoru podgrafu datasetu, ve kterém se mají hledat hodnoty pro navázání na dotazové proměnné výsledných trojic (Harris & Seaborne, 2013). Kód 4 ukazuje dotaz, který vrátí trojice přiřazující uzlům, které jsou v datasetu subjekty trojice s predikátem `rdfs:label` typ `cco:Substance`. V případě příkladových dat se vrátí dvě trojice pro `mol:CHEMBL113` a `mol:CHEMBL25`.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cco: <http://rdf.ebi.ac.uk/terms/chembl#>

CONSTRUCT {?node rdf:type cco:Substance} WHERE {
  ?node rdfs:label ?name
}
```

Kód 4: SPARQL pro konstrukci trojic přiřazující typ `cco:Substance` všem uzlům které jsou subjekty trojic s predikátem `rdfs:label`.

```
PREFIX mol: <http://rdf.ebi.ac.uk/resource/chembl/molecule/>

DESCRIBE ?predicate WHERE {mol:CHEMBL25 ?predicate ?object}
```

Kód 5: SPARQL dotaz pro vyhledání trojic všech zdrojů, které slouží jako predikát v trojicích se subjektem `mol:CHEMBL25`.

DESCRIBE slouží k získání popisu zdrojů v datasetu. Za klíčovým slovem následují URI zdrojů nebo dotazové proměnné. Výsledkem je graf všech trojic, ve kterých se zadané zdroje vyskytují. V případě použití dotazových proměnných se uvažované zdroje omezují na ty vyskytující se v podgrafu určeném vzorem za klíčovým slovem WHERE (Harris & Seaborne, 2013). Příkladem takového dotazu může být kód 5, ptající se na popis všech zdrojů, které slouží jako predikát v trojicích se subjektem `mol:CHEMBL25`. Výsledkem jeho aplikace na příkladový dataset by byl jeho subgraf tvořený všemi trojice i s predikáty `rdfs:label` a `skos:altLabel`.

Dotazovací jazyk SPARQL dále obsahuje celou řadu klíčových slov umožňující tvořit komplexnější dotazy např. pomocí množinových operací (**UNION**, **MINUS** atd.), nebo agregací (**AVERAGE**, **COUNT** atd.) (Harris & Seaborne, 2013). Hodně těchto klíčových slov má stejný tvar a obdobnou funkci jako klíčová slova v dotazovacích jazycích řídících systémů relačních databází, jako např. SQL (Allemang et al., 2020).

Velkou výhodou dotazovacího jazyka SPARQL je možnost tzv. federovaných dotazů. S jejich pomocí se může uživatel dotazovat pomocí jednoho klienta nad více datasety obsluhovanými SPARQL endpointy zároveň. Volání vzdálené služby se do dotazu zadává jako součást vzoru. Za klíčovým slovem **SERVICE**, následuje endpoint vzdálené dotazované služby a vzor podgrafu jejího datasetu. Trojice z tohoto podgrafu se pak podílejí na vyhodnocení dotazu společně s trojicemi původního datasetu (Prud'hommeaux & Buil-Aranda, 2013).

Takové použití se dá ilustrovat příkladem, kdy uživatel s přístupem ke klientovi pracujícím s příkladovým datasetem chce znát další informace o molekule kofeinu, které v původním datasetu nejsou. Pokud ale klient ví o jiné SPARQL službě a jejím endpointu obsluhující dataset, kde se nacházejí dodatečné informace a taky to, že experiment k molekule pojí predikát `bao:BAO_0090012` s významem *has participant*, může zkusit vzdálený dataset prohledat pomocí alternativních názvů z datasetu původního a dodatečné informace tak získat (Kód 6).

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bao: <http://www.bioassayontology.org/bao#>
PREFIX mol: <http://rdf.ebi.ac.uk/resource/chembl/molecule/>

SELECT ?experiment WHERE {
  {mol:CHEMBL113 skos:altLabel ?name}
  UNION {mol:CHEMBL113 rdfs:label ?name}
  SERVICE <http://example.com/endpoint/> {
  ?experiment bao:BAO_0090012 ?participant.
  ?participant rdfs:label ?name
    }
}
```

Kód 6: federovaný dotaz pro vyhledání experimentů a jejich participujících molekul ze vzdáleného endpointu pomocí názvů pro kofein z příkladového datasetu.

2.7 RDF databáze

Možnosti aplikace sémantického webu v oblasti integrace a managementu dat byly mezi prvním které zaujaly potenciální uživatele. Z tohoto pohledu se oblast sémantického webu překrývá s oblastí databází a dalo by se říct, že jejím cílem je poskytnout efektivní metody a nástroje pro sdílení, objevování, integraci a recyklaci dat na, ale i mimo internetovou síť. Rozvoj standardů a ontologií v první dekádě 21. století vedl k publikování datasetů v RDF formátech (Hitzler, 2021).

Jedním z nejznámějších je **DBpedia**, což je komunitní projekt zaměřený na extrakci dat z Wikimedia projektů. Záznamy na wikipedii mají primárně podobu nestrukturovaného textu, ale mnoho dat je zaznamenáno pomocí strukturované syntaxe, například pomocí infoboxů, kategorických výčtů, prostorových koordinát atp. Extrakce probíhá tak, že je zdrojový kód každé stránky rozebrán a převeden do struktury abstraktního syntaktického stromu. Extrakční algoritmy pak tento strom překládají do RDF trojic, které se ukládají na triplestore. Pro organizaci dat projekt využívá vlastní ontologii (Lehmann et al., 2015). Díky širokému rozsahu pokrývaných oblastí vědění, slouží DBpedia často jako jakýsi centrální prvek při integraci dat z různých zdrojů (Allemang et al., 2020). Projekt nabízí širokou škálu možností přístupu k datům zahrnující jejich stažení či online prohlížení nebo dotazování pomocí SPARQL klienta a jiných nástrojů. DBpedia je dostupná na URL https://www.dbpedia.org/.

S Wikimedia Group je spojen také projekt **Wikidata**. Jde o otevřený veřejně přístupný projekt, podobně jako Wikipedie, zaměřený na shromažďování údajů ve strukturované formě a jejich poskytování ostatním projektům, nejen z Wiki skupiny (Vrandečić & Krötzsch, 2014). Příkladem takového projektu je **Pokusnice** (https://pokusnice.cz/), online nástroj pro učitele chemie, získávající potřebné údaje pomocí SPARQL endpointu Wikidat (Juračka, 2020). Wikipedia využívá tento projekt například pro koordinaci obsahu mezi jazykovými mutacemi. Stránky popisující stejný termín v jiném jazyce vlastně popisují stejný zdroj a tím pádem se odkazují na stejný uzel v RDF Wikidat. To usnadňuje jak vzájemné odkazování na jazykové verze stejného termínu, tak i sdílení dat, kterými se například plní infoboxy (Vrandečić & Krötzsch, 2014). Databáze Wikidata je dostupná na URL https://www.wikidata.org/.

Specializovanější databáze vznikly například v oblasti biologických dat. Některé starší projekty měly za cíl vytvořit společné repositáře se širokým rozsahem dat. Např. **Bio2RDF** byl projekt zaměřený na překlad dat z bioinformatických databází (Belleau et al., 2008) a **Chem2Bio2RDF** na vytvoření jednotného repositáře pro chemogenomická RDF data (Chen et al., 2010). Oba tyto projekty již nejsou aktivní (Galgonek & Vondrášek, 2021). Omezenou funkčnost má i RDF platforma **EBI** (Jupp et al., 2014), zde z důvodu ukončeného financování (https://www.ebi.ac.uk/rdf/).

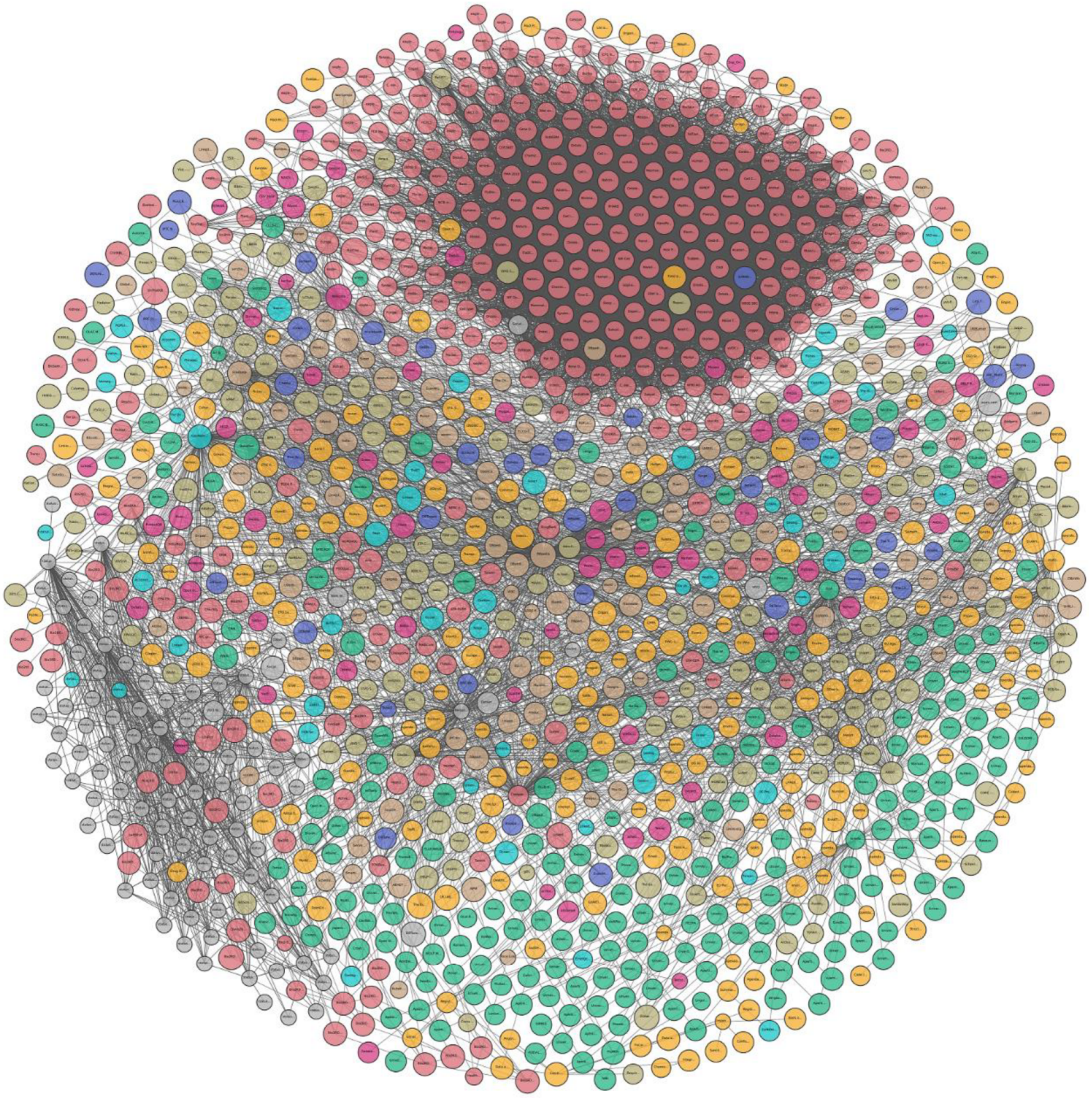
Nicméně řada RDF projektů provozovatelů bioinformatických databází je stále aktivních a mají přínos např. v oblasti objevování nových léčiv (Kanza & Frey, 2019). Mezi ně patří například **PubChem** (Fu et al., 2015), **UniProt** (Redaschi & Uniprot Consortium, 2009) a nově i **LIPID MAPS** (Bolleman et al., 2021).

2.7.1 Integrace datasetů

Ač nejde o definitivní výčet, velké množství datasetů dostupných na internetu dobře ilustruje diagram **Linked Open Data Cloud**. Ten ke květnu 2020 obsahoval 1301 datasetů a mezi nimi více než 16 tisíc propojení, což je definováno tak, že datasety mezi sebou sdílí aspoň 50 URI. Velká část tohoto cloudu je tvořena biologickými daty (Obr. 4) (McCrae, 2022).

Nicméně sdílení URI není zárukou dostatečné interoperability datasetu. Například použité ontologie nemusí být vždy kompatibilní. Mohou obsahovat termíny vztažené ke stejnému zdroji, a přitom být popsané takovým způsobem, že jsou z nich odvoditelné neslučitelné závěry. V oblasti biologických ontologií se tento problém snaží řešit **OBO Foundry**, iniciativa sdružující biomedicínské ontologie dodržující pravidla zaručující jejich vzájemnou kompatibilitu (Smith et al., 2007, Jackson et al., 2021). Repositář ontologií OBO Foundry je dostupný na URL https://obofoundry.org/. Jiným příkladem z oblasti ontologií je **BioAssay**. Hlavní snahou autorů této ontologie bylo standardizovat způsob popisu experimentů a jejich výsledků (Abeyruwan et al., 2014).

Jiný, avšak kompatibilní přístup, je integrace na úrovni datasetů. Rané příklady z biologické oblasti mohou být již zmíněné Bio2RDF a Bio2Chem2RDF, které měly celou oblast sjednotit do jednoho velkého datasetu. V minulosti se také objevily projekty cílené na integraci jednotlivých již existujících datasetů mezi sebou. **LODD** (Linking



Obr. 4: Diagram Linked Data Cloud. Červené jsou biologické datasety. Dalšími výraznými skupinami jsou lingvistické (zelené) a vládní (žluté) datasety. Převzato z McCrae (2022).

Open Drug Data) byla iniciativa skupiny zaměřené na zdravotnictví a biologické vědy (HCLS IG) pod hlavičkou W3C konsorcia, která měla za cíl převést do RDF datasety poskytnuté účastníky, propojit je mezi sebou a poskytnout dalším projektům v oblasti sémantického webu (Samwald et al., 2011). Iniciativa **Open PHACTS** (Open Pharmacological Concept Triple Store) pak měla vytvořit platformu pro zlepšení spolupráce mezi veřejným a soukromým sektorem v oblasti objevování léků a farmakologie (Williams et al., 2012).

Aktivním projektem zůstává **IDSM** (Integrated Database of Small Molecules), která sdružuje datasety PubChem RDF, ChEBI a ChEMBL. Ty byly už původně poměrně dobře propojené, nicméně žádný z těchto projektů neposkytuje vlastní SPARQL endpoint. V rámci IDSM byly datasety obohaceny o další vazby spojující společné zdroje.

Zvláštností IDSM je, že jde vlastně o relační databázi využívající na pozadí systém PostgreSQL. Ve spojení s optimalizovaným databázovým schématem umožňuje tato databáze velkou výpočetní rychlost a stabilitu při práci s velkými datasety (Galgonek & Vondrášek, 2021). Jednou z výhod IDSM je, že se nemusí pro každou vlastnost ukládat trojice a některé záznamy, jako třeba míra podobnosti mezi všemi dvojicemi molekul, mohou být řešeny uloženou procedurou. V rámci SPARQL pro tyto procedury stále neexistuje standard.

Další výhodou je možnost použít další funkcionality implementované pro relační databáze. V případě IDSM je takto použit Sachem (Kratochvíl et al., 2018), rozšíření PostgreSQL pro vyhledávání molekulových substruktur a výpočet podobnosti (Kratochvíl et al., 2019). Databáze IDSM je dostupná na URL https://idsm.elixir-czech.cz/.

3. PRAKTICKÁ ČÁST

Praktická část práce se zabývá samotným zvýšením interoperability databáze MolMeDB s dalšími databázemi pomocí technologií sémantického webu. Tohoto cíle bylo dosaženo překladem relačního datasetu MolMeDB do RDF formátu a jeho zveřejněním. Aby k tomu mohlo dojít, muselo být vytvořeno RDF schéma databáze reflektující schéma její relační části a pro toto schéma musely být nalezeny, či vytvořeny vhodné termíny z ontologií. Nakonec bylo potřeba zajistit přístup k datům a nástroje pro efektivní nakládání s nimi.

3.1 Použité technologie

3.1.1 Ontology Lookup Service

Ontology Lookup Service (OLS) je repositář biomedicínských ontologií provozovaný SPOT (Samples, Phenotypes and Ontologies) týmem **EBI**. Přístup k ontologiím je zajištěn pomocí REST API a webové vyhledávací služby. Vyhledávání je možné pomocí názvů či zkratek ontologií, zkrácených URI termínů, či jejich názvů. Hierarchie pojmů jednotlivých ontologií jsou znázorněny pomocí stromových diagramů. Služba je přístupná na webu https://www.ebi.ac.uk/ols/ (Jupp et al., 2015).

3.1.2 Ontologie

Při vytváření schématu RDF datasetu hrají zásadní roli termíny přiřazující zdrojům jejich třídy a vlastnosti. Přehled použitých ontologií uvádí tabulka 1. Použité termíny jsou uvedeny v diagramech RDF schématu v části 3.2 Schéma RDF datasetu. Následují krátké popisy ontologií.

RDF Schema (RDFS) je jazyk a ontologie sloužící ke konstrukci hierarchií ontologií (Brickley & Guha, 2014). Bližší popis je uveden v části 2.5 Ontologie a modelování dat. Ontologie byla použita pro anotační účely (`rdfs:label`, `rdfs:comment`) a pro tvorbu slovníku.

OWL 2 je vysoce expresivní jazyk a ontologie sloužící ke konstrukci ontologií (Hitzler et al., 2012). Bližší popis je uveden v části 2.5 Ontologie a modelování dat. Ontologie MolMeDB byla zkonstruovaná pomocí OWL.

Tab. 1: Ontologie použité pro vytvoření datového modelu RDF MolMeDB.

Jmenný prostor	URI prefix	název ontologie
rdf:	http://www.w3.org/2000/01/rdf-schema#	RDF Schema 1.1 (Brickley & Guha, 2014)
rdfs:	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF Schema 1.1 (Brickley & Guha, 2014)
owl:	http://www.w3.org/2002/07/owl#	OWL 2 (Hitzler et al., 2012)
xsd:	http://www.w3.org/2001/XMLSchema#	XML Schema Definition Language (Peterson et al., 2012)
skos:	http://www.w3.org/2004/02/skos/core#	Simple Knowledge Organization Systém (Miles & Bechhofer 2009)
sio:	http://semanticscience.org/resource/	Semanticscience Integrated Ontology – SIO (Dumontier et al., 2014)
		Chemical Information Ontology – CHEMINF (Hastings et al., 2011)
bao:	http://www.bioassayontology.org/bao#	BioAssay Ontology (Abeyruwan et al., 2014)
repr:	https://w3id.org/reproduceme#	REPRODUCE-ME (Samuel & König-Ries, 2019)
obo:	http://purl.obolibrary.org/obo/	The Information Artifact Ontology – IAO (Stoeckert et al., 2020)
		NCI Thesaurus OBO Edition – NCIT (Balhoff et al.)
		The Phenotype And Trait Ontology – PATO (Tan et al., 2021)
		eagle-i resource ontology – ERO (Torniai et al., 2011)
cito:	http://purl.org/spar/cito/	Citation Typing Ontology (Shotton & Peroni, 2018)
dc:	http://purl.org/dc/elements/1.1/	The Dublin Core (DC) ontology (DCMI Usage Board, 2020)
dcterms:	http://purl.org/dc/terms/	The Dublin Core (DC) ontology (DCMI Usage Board, 2020)
dcmitypes:	http://purl.org/dc/dcmitype/	The Dublin Core (DC) ontology (DCMI Usage Board, 2020)
bibo:	http://purl.org/ontology/bibo/	Bibliographic Ontology (D'Arcus & Giasson, 2016)

XML Schema Definition Language (XSD) definuje datové typy (Peterson et al., 2012) používané pro literály (Cyganiak et al., 2014).

Simple Knowledge Organization System (SKOS) je RDF slovníkem pro vyjádření částečně formálních systémů organizace znalostí jako jsou thesaury, taxonomie či klasifikační schémata. Ke zdrojům SKOS přistupuje jako ke konceptům. Mimo jiné umožňuje mapování konceptů mezi různými schématy (Isaac & Summers, 2009). Toho bylo využito při mapování molekul na jejich protějšky v jiných databázích pomocí predikátu `skos:exactMatch`.

Semanticscience Integrated Ontology (SIO) je jednoduchou ontologií tříd a predikátů pro popis objektů, procesů a jejich atributů v oblasti biomedicínského výzkumu. Původně tato ontologie vznikla jako sémantický základ projektu Bio2RDF (Dumontier et al., 2014). Použity byly predikáty pro vyjádření vazeb atributů molekul, hodnot a jednotek.

Chemical Information Ontology (CHEMINF) má za cíl vytvořit standard pro reprezentaci chemické informace, zejména ve vztahu k chemickým strukturám a vlastnostem (Hastings et al., 2011). Použity byly třídy molekulárních deskriptorů a identifikátorů pro různé databáze.

REPRODUCE-ME Ontology je ontologie zaměřená na popis vědeckých experimentů a jejích celého průběhu (Samuel & König-Reis, 2019). Použit byl predikát `repr:hasExperimentalCondition` pro vyjádření teploty a náboje molekuly měřených membránových interakcí a terminy pro pozitivní a negativní výsledek detekce inhibiční aktivity a vazby substrátu transportéru.

BioAssay Ontology (BAO) popisuje screeningové testy a jejich výsledky včetně high througput screeningu (HTS) (Abeyruwan et al., 2014). Pojmy této ontologie tvoří sémantický základ popisu výsledků měření a výpočtu hodnot membránových a transportérových interakcí. Dále tato ontologie posloužila jako základ vlastního slovníku MolMeDB, kde je valná většina termínů definována jako podtřída některé třídy BAO.

Ostatní biologické ontologie spadají do skupiny ontologií **OBO Foundry** (Open Biomedical Ontologies). Jde o otevřenou platformu vzniklou za účelem koordinace vytváření biomedicínských ontologií. Ontologie patřící do této skupiny dodržují určitá pravidla, která zajišťují mimo jiné jejich vzájemnou kompatibilitu (Smith et al., 2007).

Information Artifact Ontology (IAO) zaměřená na práci s informačními entitami (Stoeckert et al., 2020) byla použita pro mapování zastaralých identifikátorů molekul na aktuální a definice pojmů ze slovníku MolMeDB.

NCI Thesaurus je slovník zaměřený primárně na léčbu a výzkum různých druhů rakoviny (Sioutos et al., 2007). Z jeho OBO edice (Balhoff et al.) byly převzaty termíny pro jednotky.

Z **Phenotype And Trait Ontology** (PATO), slovníku vytvořeného původně za účelem popisu fenotypových projevů (Tan et al., 2021), byl převzat termín `obo:PATO_0000146` pro teplotu.

Eagle-i Resource Ontology (ERO) je ontologie vyvinutá pro potřeby eagle-i, softwarové platformy zaměřené na vytváření a sdílení sémanticky bohatých dat v biomedicínské oblasti (Torniai et al., 2011). Použit byl predikát `obo:ERO_0000547` (*has_uniPROT_ID*).

Dále byly použity ontologie obecnějšího zaměření pro anotační termíny.

Citation Typing Ontology (CiTO) je ontologie zaměřená na citování a parafrázování prací (Shotton & Peroni, 2018). Byla použita ke spojení záznamů pro membránové a transportérové interakce s jejich zdroji.

Ontologie **Dublin Core** (DC) poskytuje jednoduchý slovník pro popis metadat (DCMI Usage Board, 2020). Použity byly anotační termíny pro popis a citace metod a membrán. Mimo to byla tato ontologie použita pro anotace záznamů literárních a databázových zdrojů.

Posledním použitým slovníkem je **Bibliographic Ontology** (BIBO). BIBO je ontologie pro vyjádření citací a bibliografických referencí (D'Arcus & Giasson, 2016). V RDF schématu MolMeDB byly použity predikáty ke spojení bibliografických zdrojů s jejich DOI a PMID identifikátory.

3.1.3 R2RML

R2RML (RDB to RDF Mapping Language) je jazyk sloužící k přizpůsobenému mapování relačních datasetů do RDF datasetů. Umožňuje zvolit schéma výsledného RDF datasetu a slovníky vyjadřující vzájemné vztahy (Das et al., 2012). K vytváření RDF

datasetu z relačního může sloužit i **přímé mapování**, ale to ve výsledném schématu i slovníku vychází čistě ze struktury relační databáze a neumožňuje další úpravy během generování RDF datasetu (Arenas et al., 2012).

Mapování pro R2RML se vyjadřuje jako RDF graf zapsaný v Turtle syntaxu a je uzpůsobeno konkrétnímu relačnímu schématu, na kterém má probíhat a slovníku výsledného datasetu. Pro vyjádření toho, jakou roli bude ve výsledku mít který údaj, je zaveden vlastní slovník s prefixem `http://www.w3.org/ns/r2rml#` (jmenný prostor `rr`). Vstupem je relační databáze dodržující příslušné relační schéma a výsledkem je RDF dataset s predikáty, typy z vybraného slovníku (Das et al., 2012).

Krátký příklad je možné ukázat na mapování záznamu UniProt ID a názvů transportérů z tabulky *transporter_targets* do RDF grafu vyjadřujícího, že transportér s danou IRI v UniProt má tento název (Kód 7).

Na začátku jsou definovány prefixy a pak následují samotná mapování. Každé z nich je soubor trojic se subjektem odpovídající danému mapování, zde `<#TriplesMap1>`. K němu se váží predikáty `rr:logicalTable`, `rr:subjectMap` a `rr:predicateObjectMap`.

Objektem `rr:logicalTable` je uzel představující logickou tabulku, která se vytváří buď z SQL tabulky, SQL pohledu nebo výsledku SQL dotazu. Konkrétní možnost určuje predikát vedoucí od uzlu logické tabulky na řetězec, kterým je název tabulky nebo pohledu, či databázový dotaz. V uvedeném příkladu jde o dotaz, jehož výsledkem je tabulka 2. Za pomoci logické tabulky se doplňují údaje do příslušných trojic výsledku mapování.

Objektem `rr:subjectMap` je uzel vyjadřující tvar subjektů trojic výsledku mapování. Z uzlu vede predikát určující, jestli bude subjektem konstanta, hodnota atributu tabulky nebo jestli vznikne začleněním hodnot nějakých atributů do řetězce. V příkladu je uveden poslední jmenovaný způsob, kde je výsledkem URI vzniklá zasazení hodnoty atributu *uniprot_id* do vzorového řetězce. Možné je i připojit predikát `rr:class` jehož objekt je URI subjektů výsledku mapování. V příkladu se přiřazuje subjektům typ *transportér*.

Následuje jednou nebo vícekrát predikát rr:predicateObjectMap jehož objektem je uzel pro predikát a tvar objektů trojic výsledku mapování. Na tento uzel je navázán predikát rr:predicate, jehož objektem je URI predikátu trojic výsledku, a predikát rr:object s objektem určujícím tvar objektů trojic výsledku. Ten je určen stejným způsobem jako tvar subjektu. V uvedeném příkladě je vybraný predikát *název* a objekt je hodnota atributu *name* logické tabulky.

Generování trojic pak podle mapování probíhá zvlášť nad každým řádkem logické tabulky. Trojice se nevytváří v případě, že je hodnota některého atributu nutného pro její vytvoření *NULL*. Výsledek mapování z uvedeného příkladu nad tabulkou *transporter_targets* uvádí kód 8.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix bao: <http://www.bioassayontology.org/bao#>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.

<#TriplesMap1>
rr:logicalTable [ rr:sqlQuerry """
SELECT name, uniprot_id FROM transporter_targets LIMIT 2
"""];
rr:subjectMap [
  rr:template "http://purl.uniprot.org/uniprot/{uniprot_id}";
  rr:class bao:BAO_0000283;
];
  rr:predicateObjectMap [
  rr:predicate  rdfs:label;
  rr:objectMap [ rr:column "name" ]
].
```

Kód 7: Mapování Uniprot identifikátorů a jmen ze selekce a restrikce tabulky *transporter_targets* na trojice vyjadřující, že transportér (bao:BAO_0000283) s konkrétní IRI vytvořenou z atributu *uniprot_id* má jméno (rdfs:label) jaké je uvedné pro atribut *name*.

Tab. 2: Logická tabulka z dotazu v kódu 7

name varchar(150)	**uniprot_id** varchar(50)
SLC22A2	O15244
SLC22A1	O15245

```
<http://purl.uniprot.org/uniprot/O15244> rdf:type bao:BAO_0000283 .
<http://purl.uniprot.org/uniprot/O15244> rdfs:label "SLC22A2" .
<http://purl.uniprot.org/uniprot/O15245> rdf:type bao:BAO_0000283 .
<http://purl.uniprot.org/uniprot/O15245> rdfs:label "SLC22A1" .
```

Kód 8: RDF trojice vyjadřující výsledný graf z mapování v kódu 7 aplikovaným na tabulku *transporter_targets* zapsané za pomocí syntaxe Turtle.

3.1.4 SPARQLlib

SPARQLlib je velice jednoduchá PHP knihovna umožňující pokládání SELECT dotazů SPARQL endpointů obsluhujících RDF datasety. SPARQL lib je součástí širší PHP knihovny pro práci s RDF daty jménem Graphite PHP Linked Data Library a je veřejně dostupná na webu http://graphite.ecs.soton.ac.uk/sparqllib/ (Gutteridge, 2012). V rámci projektu MolMeDB RDF byla tato knihovna zvolena pro jednoduchost jejího zapojení do webové aplikace MolMeDB a byla použita pro implementaci HTTP dereference URI RDF datasetu na popis referentu v HTML formátu.

3.1.5 REST API

Pro přístup k datům se využívá **REST API** webové aplikace MolMeDB (Juračka 2020). Název vychází z tzv. **REST** (Representational State Transfer) architektury pro distribuované systémy, jako např. internetová síť. Tento styl architektury specifikuje, jak přistupovat ke zdrojům v systému a pracovat s nimi. Nezabývá se však detaily implementace těchto zdrojů nebo protokoly (Fielding, 2000).

Každý zdroj v systému je identifikován pomocí URI a ke zdrojům se přistupuje pomocí metod HTTP protokolu: **GET**, **POST**, **PUT** a **DELETE** (Benatallah & Motahari Nezhad, 2008). Ty jsou definovány dokumentem RFC 7231 popisujícím sémantiku a obsah HTTP protokolu (Fielding & Reschke, 2014).

Metoda **GET** vysílá žádost o transfer vybrané reprezentace cílového zdroje. Způsob výběru a zaslání vhodné reprezentace záleží na konkrétní implementaci. Příkladem může být žádost o dereferenci URI RDF datasetu. Např. dereference `mmdbint:membrane3` vydá reprezentaci membrány DOPC. Pro detailnější rozbor dereference viz. část 3.4 Dereference.

POST je metoda vysílající žádost o vytvoření nového záznamu cílovým zdrojem. Tím může být např. konstruktor zodpovědný za vytvoření záznamu dle vlastních specifikací. V těle žádosti je umístěn popis zdroje, pro který má být záznam vytvořen. Příkladem může být třeba zaslání komentáře na webové stránce.

Metoda **PUT** zasílá žádost o změnu reprezentace cílového zdroje. Může tak být vytvořen i nový záznam, avšak oproti metodě POST je výsledná reprezentace explicitně uvedena v těle žádosti a nedochází k jejímu zpracování jiným zdrojem.

DELETE je metoda zasílající žádost o smazání záznamu cílového zdroje.

Pro úplnou komunikaci klienta se serverem zasílá server výsledky požadavků klienta společně s návratovou hodnotou obsahující HTTP hlavičku. V hlavičce je uveden tzv. stavový kód který informuje klienta, zda došlo k úspěšnému či neúspěšnému vyřízení žádosti (Fielding & Reschke, 2014). Stavové kódy jsou trojciferná čísla a první číslo určuje třídu odpovědi:

- **1xx** (informativní) – Informuje klienta o přijetí požadavku. Může být vydána např. jako součást odpovědi, že požadavek je zpracováván.
- **2xx** (úspěch) – Informuje klienta o úspěšném vyřízení požadavku.
- **3xx** (přesměrování) – Pro vyřízení požadavku je nutná další akce, např. opětovné zaslání na jinou URI.
- **4xx** (chyba klienta) – Nastala chyba v syntaxi požadavku nebo požadavek nemůže být z jiného důvodu vyřízen
- **5xx** (chyba serveru) – Server nezvládl zpracovat validní požadavek.

Hlavička může obsahovat další údaje jako např. formát další informace v těle odpovědi. Tou může být třeba popis zdroje pro který byl zaslán požadavek GET.

V rámci projektu MolMeDB bylo REST API rozšířeno o implementaci dereference URI RDF datasetu.

3.2 Schéma RDF datasetu

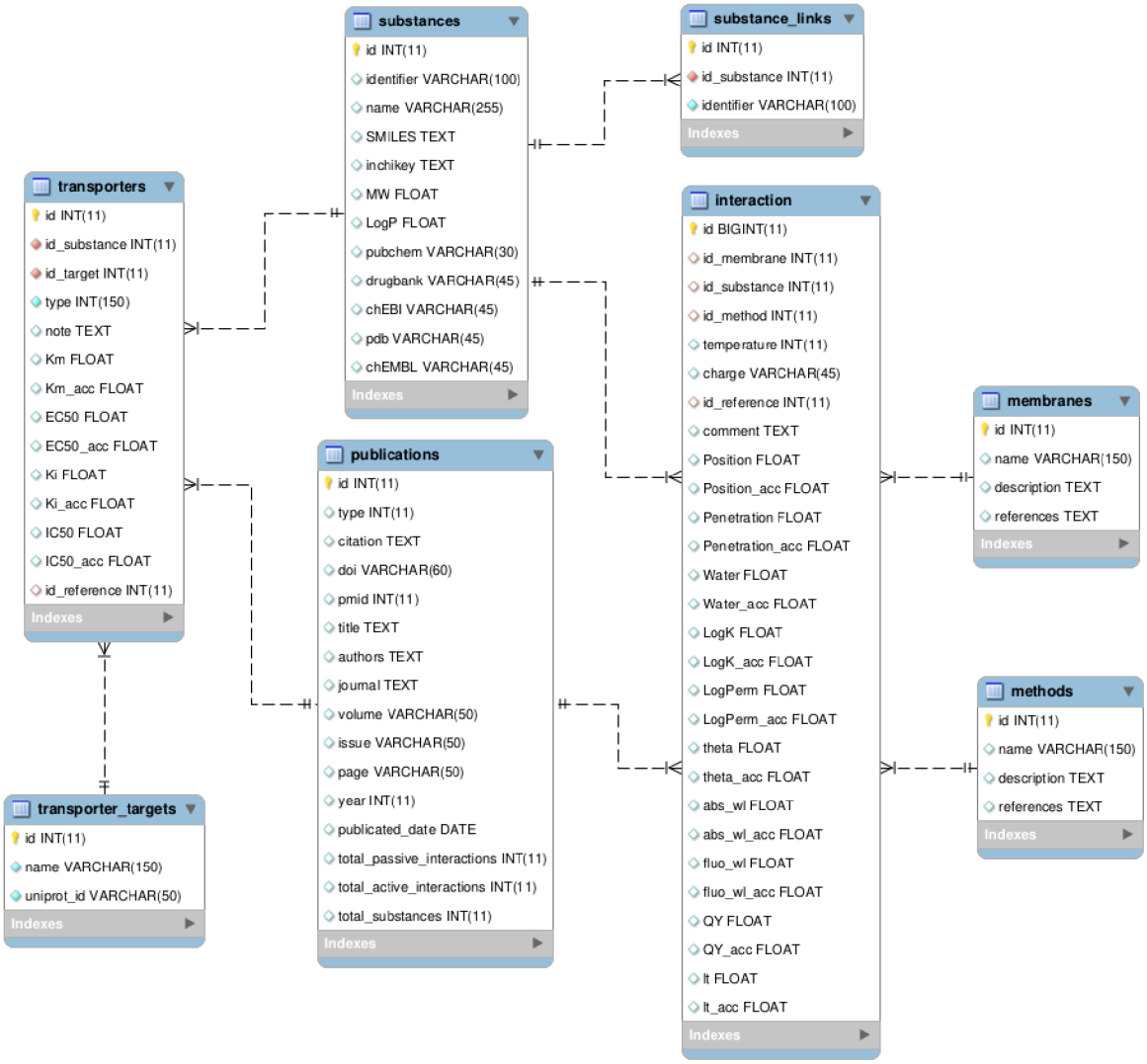
Relační schéma části databáze MolMeDB zodpovědné za poskytování dat o interakcích a souvisejících informací uvádí obrázek 5. Tabulka *substance_links* obsahuje staré identifikátory molekul a jejich současné ID z důvodu zachování starších odkazů. Obsah ostatních tabulek odpovídá jejich názvu.

Pro tuto část databáze bylo vypracováno **schéma RDF reprezentace dat**. V případě že zatím nebyla identifikována vhodná zavedená **URI**, byly zavedeny termíny do vlastní ontologie, rozšiřující již zavedené slovníky. Při vytváření schématu RDF byly preferovány zavedené ontologie využívané v RDF PubChem (Fu et al., 2015) a ChEMBLu (Willighagen et al., 2013). V případech, které nepokrývají RDF modely těchto dvou databází, byly zvoleny jiné ontologie vyhledané pomocí **Ontology Lookup Service**, nebo vlastní rozšíření.

Pro snadnější práci se schématem, byla zpracovávané tabulky rozděleny do 4 domén podle vzájemného propojení tabulek ve schématu MySQL databáze: **molekuly**, **interakce**, **transportérová měření** a **zdroje**. Pro účely vytvoření datového modelu byl pro každou z těchto oblastí zaveden jmenný prostor (Tab. 3).

Systém vytváření sufixů URI je inspirován systémem zavedeným v RDF PubChemu, kde se například u URI uzlů sloučenin k prefixu jmenného prostoru připojuje jako sufix ID dané sloučeniny. Pro uzel odpovídající vlastnosti sloučeniny pak sufix URI vznikne připojením standardizovaného názvu oné vlastnosti k ID sloučeniny. Např. uzel pro kofein má URI sufix `CID2519` a uzel pro jeho molekulární hmotnost má URI sufix `CID2519_Molecular_Weight`. V MolMeDB má kofein identifikátor MM00040, takže ekvivalentní URI ve jmenném prostoru je `mmdbsub:MM00040_molecular_weight`. Výjimkou jsou URI pro molekuly využívající již zavedený systém perzistentních identifikátorů https://registry.identifiers.org/registry/molmedb# a pro transportéry využívající perzistentní identifikátory UniProtu.

Pro každou oblast existuje ústřední uzel odpovídající záznamu molekuly, interakce, transportérovému měření nebo zdroji dat a z tohoto uzlu vedou odkazy na uzly dat uložených v mateřských tabulkách nebo na uzly záznamů v ostatních tabulkách, včetně tabulek z jiných jmenných prostorů.



Obr. 5: Schéma veřejně dostupné části relační databáze MolMeDB.

Tab. 3: Jmenné prostory pro IRI oblastí datového modelu RDF MolMeDB.

prefix	jmenný prostor	data
mmdbsub:	https://rdf.molmedb.upol.cz/substance/	tabulka substances, substance_links
mmdbint:	https://rdf.molmedb.upol.cz/interaction/	tabulky interaction, membranes, methods
mmdbtra:	https://rdf.molmedb.upol.cz/transporter/	tabulky transporters, transporter targets
mmdbref:	https://rdf.molmedb.upol.cz/reference/	tabulka publications
mmdbvoc:	https://rdf.molmedb.upol.cz/vocabulary#	slovník termínů MolMeDB

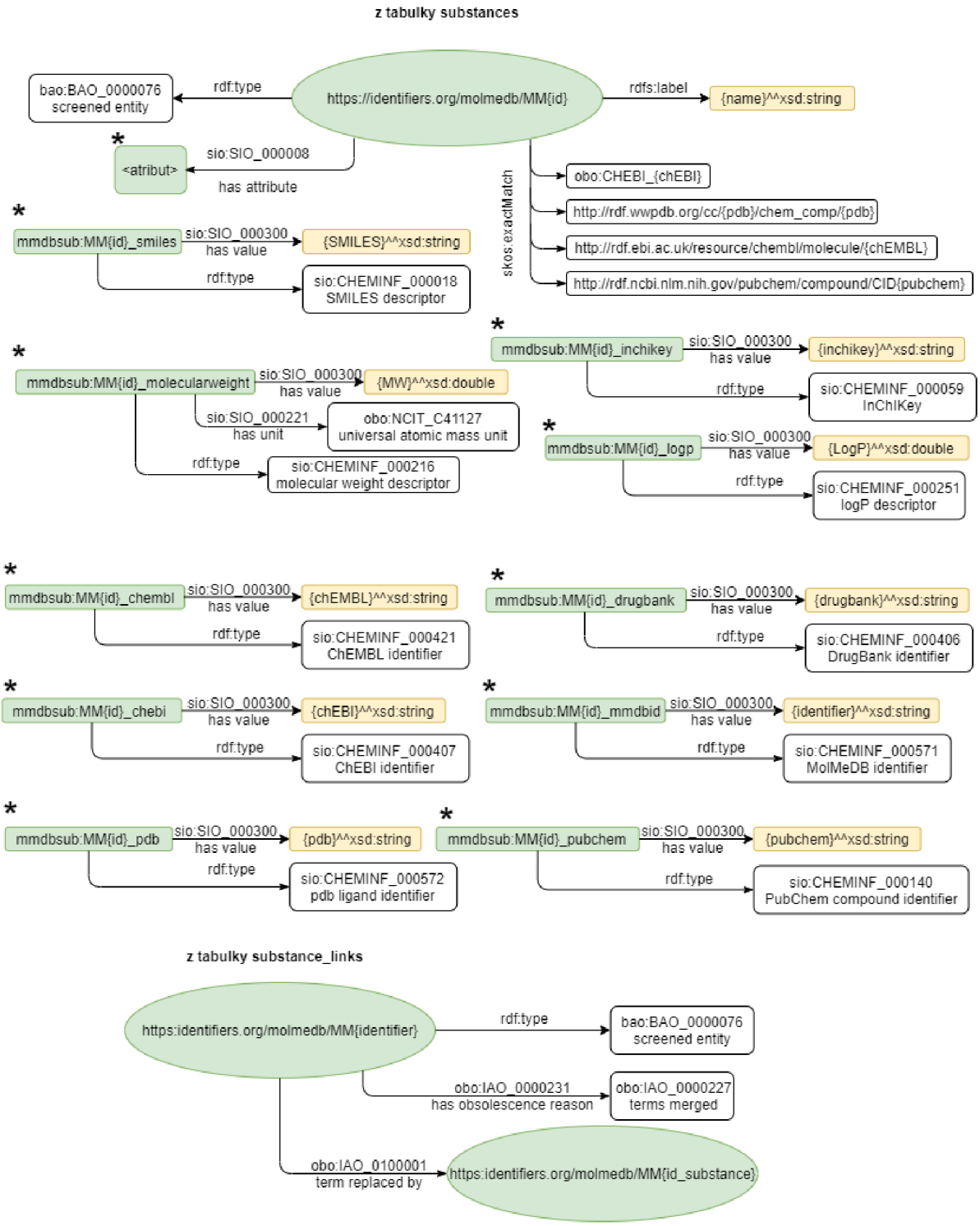
3.2.1 Sloučeniny

Většina trojic v doméně substancí vychází z tabulky *substances* (Tab. 4). Ústředním je zde IRI molekuly vytvořené kombinací vzoru perzistentního identifikátoru sloučenin MolMeDB a identifikátoru ve formě `https:identifiers.org/molmedb/MM{identifikátor}`. Typ tohoto uzlu je určen jako `bao:BAO_0000076` (*screened entity*), který je podtřídou BAO termínu pro molekulární entitu. Uzel tohoto typu je požadován restrikcí na predikát `bao:BAO_0090012` (*has participant*) u BAO termínů pro biologická měření použitích v datovém modelu membránových a transportérových měřeni.

Schéma RDF pro vyjádření záznamů z tabulky *substances* (Obr. 6) bylo z velké části převzato přímo z RDF PubChem Compound a PubChem Substance. IRI `sio:SIO_000008` s významem *has attribute* je v RDF PubChemu použita pro trojici od uzlu sloučeniny k uzlu deskriptoru a od substance k uzlu identifikátoru v jiné databázi. Stejný postup byl zvolen pro RDF MolMeDB. Popis uzlu deskriptoru v podobně trojic udávajících typ deskriptoru, hodnotu a případně jednotku a popis identifikátoru určující typ a hodnotu je také převzat z RDF PubChem. Pro typy uzlů deskriptorů byly vybrány termíny z ontologie CHEMINF.

Tab. 4: Část schématu tabulky *substances*.

Field	Type	Null	Key
id	int(11)	NO	PRI
identifier	varchar(100)	YES	UNI
name	varchar(255)	YES	MUL
SMILES	text	YES	
inchikey	text	YES	
MW	float	YES	
Area	float	YES	
Volume	float	YES	
LogP	float	YES	
pubchem	varchar(30)	YES	
drugbank	varchar(45)	YES	
chEBI	varchar(45)	YES	
pdb	varchar(45)	YES	
chEMBL	varchar(45)	YES	



Obr. 6: Diagram RDF grafu pro sloučeniny. Ve složených závorkách je uveden atribut z příslušné tabulky. Uzel <atribut> zastupuje místo uzlů pro atributy označené hvězdou. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a uzly RDF jiných databází.

Tab. 5: Část schématu tabulky *substance_links*

Field	Type	Null	Key
id	int(11)	NO	PRI
id_substance	int(11)	NO	MUL
identifier	varchar(100)	NO	

Název molekuly je textový řetězec připojený predikátem `rdfs:label` označujícím obecně lidsky čitelné názvy. Ontologie nabízejí i specializovanější označení, jako různé druhy chemických názvů, ale vzhledem k různorodosti názvů uložených v MolMeDB bylo zvoleno obecnější řešení.

Pro zvýšení provázanosti s jinými databázemi jsou identifikátory uložené v MolMeDB využity k vytvoření vazby molekuly s jejím protějškem v **ChEBI**, **PubChem**, **PDBj** a **ChEMBL** pomocí predikátu `skos:exactMatch`.

Jmenný prostor `mmdbsub` je použit také pro tabulku *substance_links* (Tab. 5) zahrnující vazby již vyřazených MolMeDB identifikátorů na aktuální záznamy pomocí predikátu `obo:IAO_0100001` s významem *term replaced by*. Uzel vyřazeného záznamu je také označen predikátem `obo:IAO_0000231` (*obsolescence reason*) na subjekt `obo:IAO_0000227` (*terms merged*).

3.2.2 Membránové a transportérové interakce

Datové modely pro interakce mezi molekulou a membránou (Obr. 7 a 8) a molekulou a transportérem (Obr. 9) jsou částečně inspirovány RDF PubChem BioAssay, MeasureGroup a Endpoint. PubChem uchovává data o měřeních jako bioassays které jsou rozdělené do measuregroups po proteinech, pro které jsou prováděny měření a ty jsou rozdělené na jednotlivé endpointy uvádějící výsledné hodnoty pro interakci substance s proteinem.

MolMeDB uchovává zvlášť záznam pro každé měření nebo výpočet. Tím pádem je každý záznam zvlášť tvořen jedním centrálním záznamem pro biologické měření a jednou příslušnou measure group propojenými predikátem `bao:BAO_0000209` (*has measure group*) a jeho inverzním protějškem. Oba tyto uzly odkazují predikátem `bao:BAO_0090012` (*has participant*) na příslušnou molekulu a membránu, nebo transportér. Z centrálního uzlu také vede odkaz na zdroj dat.

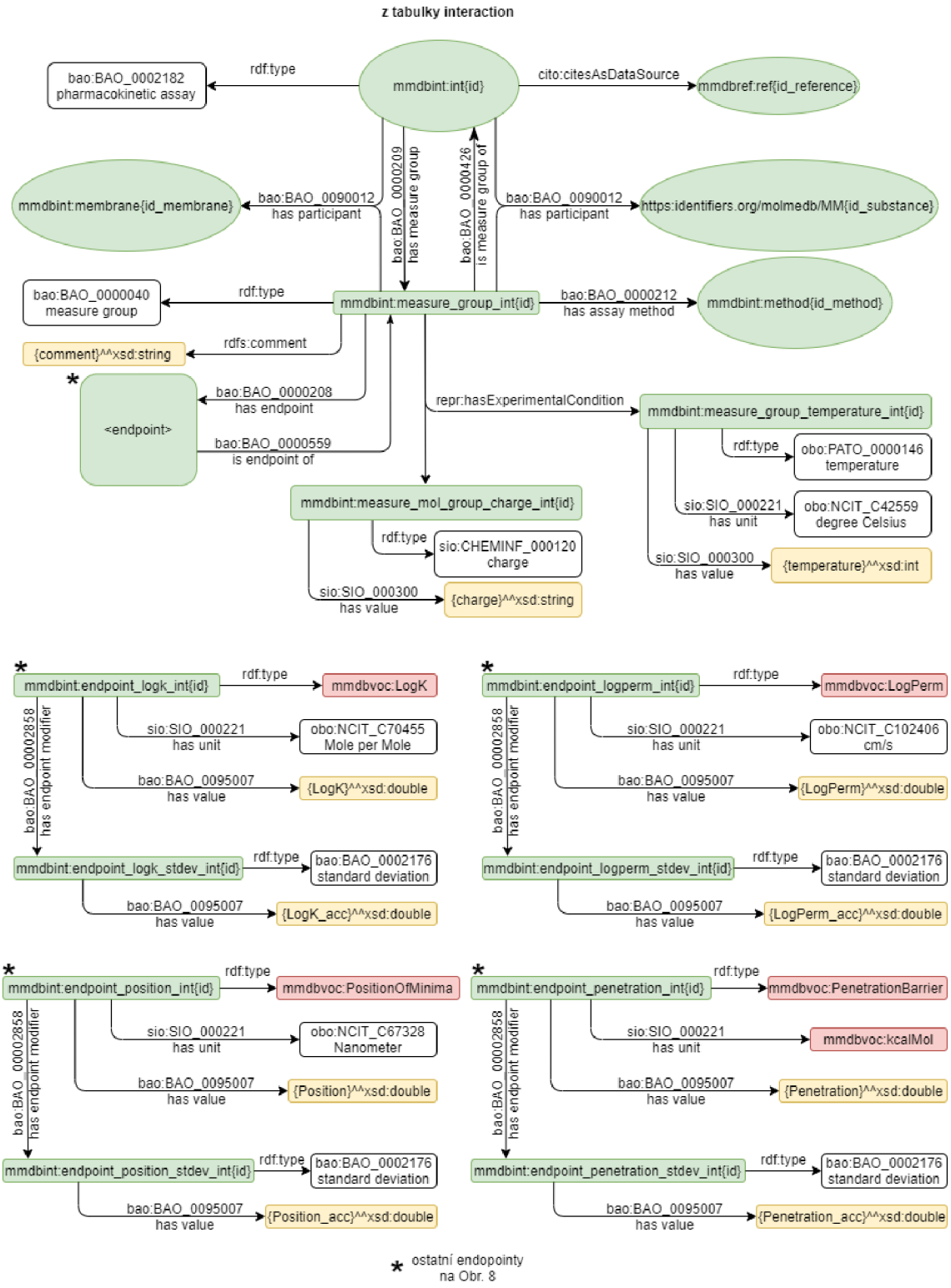
Na naměřené hodnoty se z measure group odkazuje predikát bao:BAO_0000208 (*has endpoint*) a zpětně jeho inverzní protějšek. Popis naměřený hodnot (Obr. 7, 8 a 9) je podobný jako v Pubchem Bioassays – popisuje typ, hodnotu a jednotku. Uzly hodnot mají typy pro endpointy z BAO nebo z vlastního slovníku jako podtřídy bao:BAO_0000179 (*endpoint*). V případě existence směrodatné odchylky vede z uzlu endpointu predikát bao:BAO_0002858 (*has endpoint modifier*) na uzel typu bao:BAO_0002176 (*standard deviation*) s odkazem bao:BAO_0095007 (*has value*) na hodnotu. K uzlům *measure group* je možno přiřadit komentář jako *string*, na který se odkazuje predikát rdfs:comment.

Membránové interakce jsou popsány tabulkou *interaction* (Tab. 6). Centrální uzel pro interakci s membránou (Obr. 7) má typ bao:BAO_0002182 *pharmacokinetic assay*. Navíc oproti RDF PubChemu jsou zavedeny trojice pro teplotu při experimentu či výpočtu a náboj molekuly s predikátem repr:hasExperimentalCondition s podobnou strukturou jako mají hodnoty měření a deskriptory. Pro endpointy naměřených hodnot byly vytvořeny typy jako podtřídy endpointových typů z BAO.

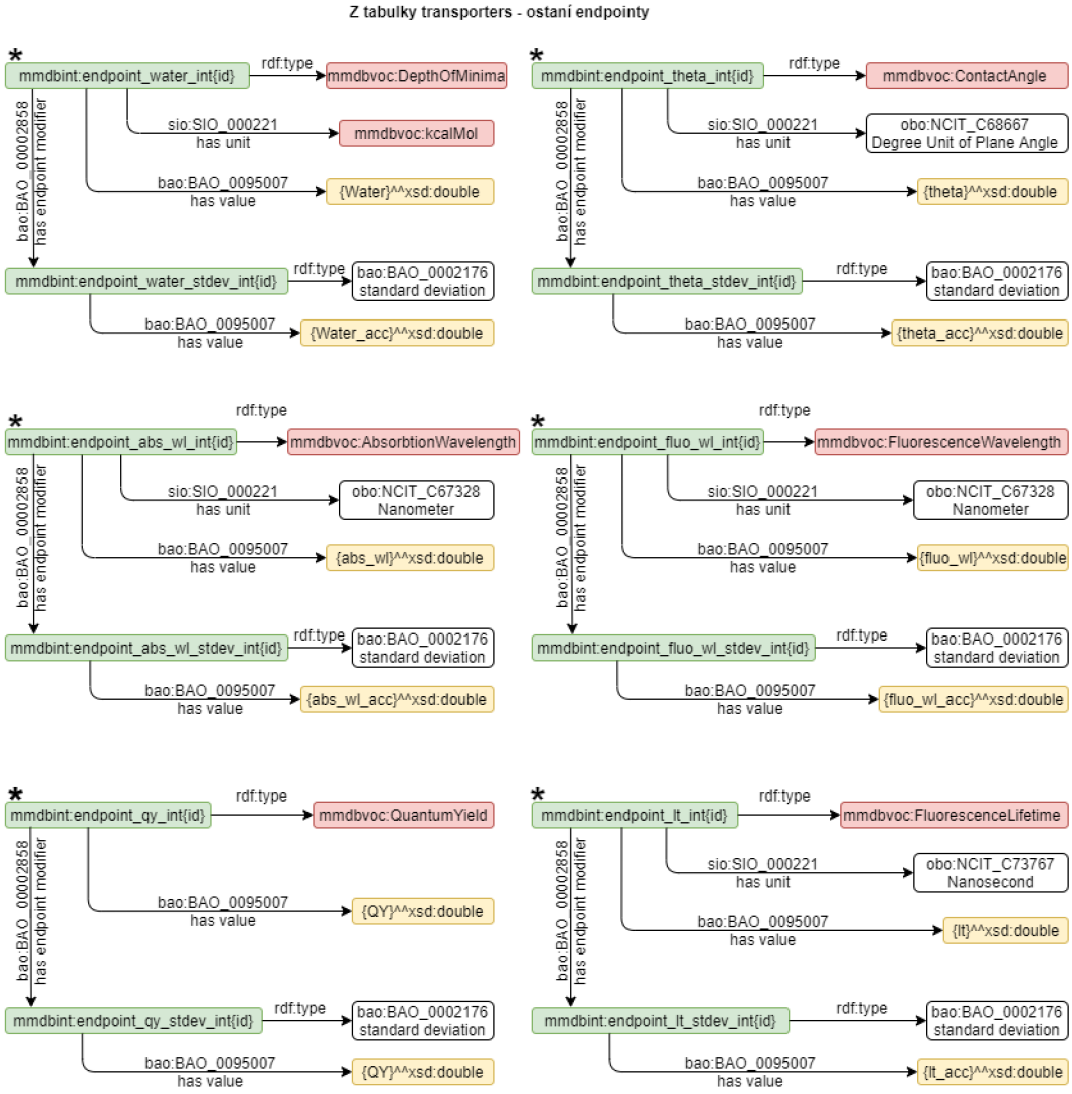
V MySQL databázi se záznamy pro membránové interakce odkazují do tabulek *substances*, *publications*, *methods* a *membranes*. V RDF modelu se tento vztah projevuje jako odkaz z uzlu záznamu interakce na centrální uzel substance z domény sloučenin, uzel zdroje dat z prostoru zdrojů. Uzly metody a membrány sdílející doménu s interakcí.

Tab. 6: Část schématu tabulky *interaction.*

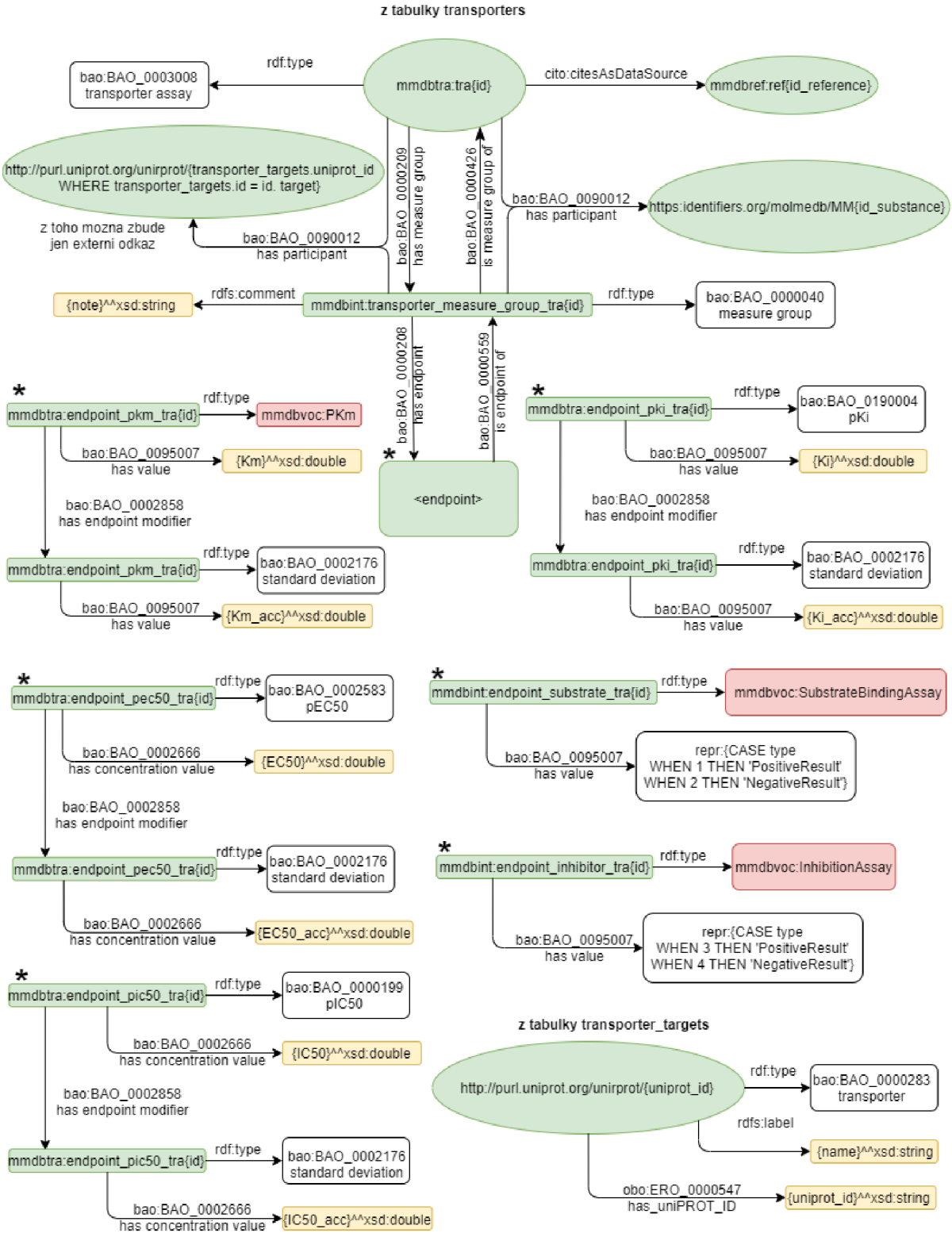
Field	Type	Null	Key
id	bigint(11)	NO	PRI
id_membrane	int(11)	YES	MUL
id_substance	int(11)	YES	MUL
id_method	int(11)	YES	MUL
temperature	int(11)	YES	
charge	varchar(45)	YES	
id_reference	int(11)	YES	MUL
comment	text	YES	
Position	float	YES	
Position_acc	float	YES	
Penetration	float	YES	
Penetration_acc	float	YES	
Water	float	YES	
Water_acc	float	YES	
LogK	float	YES	
LogK_acc	float	YES	
LogPerm	float	YES	
LogPerm_acc	float	YES	
theta	float	YES	
theta_acc	float unsigned	YES	
abs_wl	float	YES	
abs_wl_acc	float unsigned	YES	
fluo_wl	float	YES	
fluo_wl_acc	float unsigned	YES	
QY	float	YES	
QY_acc	float unsigned	YES	
lt	float	YES	
lt_acc	float unsigned	YES	



Obr. 7: Diagram RDF grafu membránových interakcí. Ve složených závorkách je uveden atribut z příslušné tabulky. Uzel <endpoint> zastupuje místo uzlů pro endpointy označené hvězdou. Zbylé endpointy jsou na obrázku 8. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, červené jsou termíny ze slovníku MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a uzly RDF jiných databází.



Obr. 8: Diagramy RDF grafů ostatních endpointů (viz. Obr 7). Ve složených závorkách je uveden atribut z příslušné tabulky. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, červené jsou termíny ze slovníku MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a uzly RDF jiných databází.



Obr. 9: Diagram RDF grafu pro jmenný prostor transportérových interakcí. Ve složených závorkách je uveden atribut z příslušné tabulky. Uzel <endpoint> zastupuje místo uzlů pro endpointy označené hvězdou. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, červené jsou termíny ze slovníku MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a uzly RDF jiných databází.

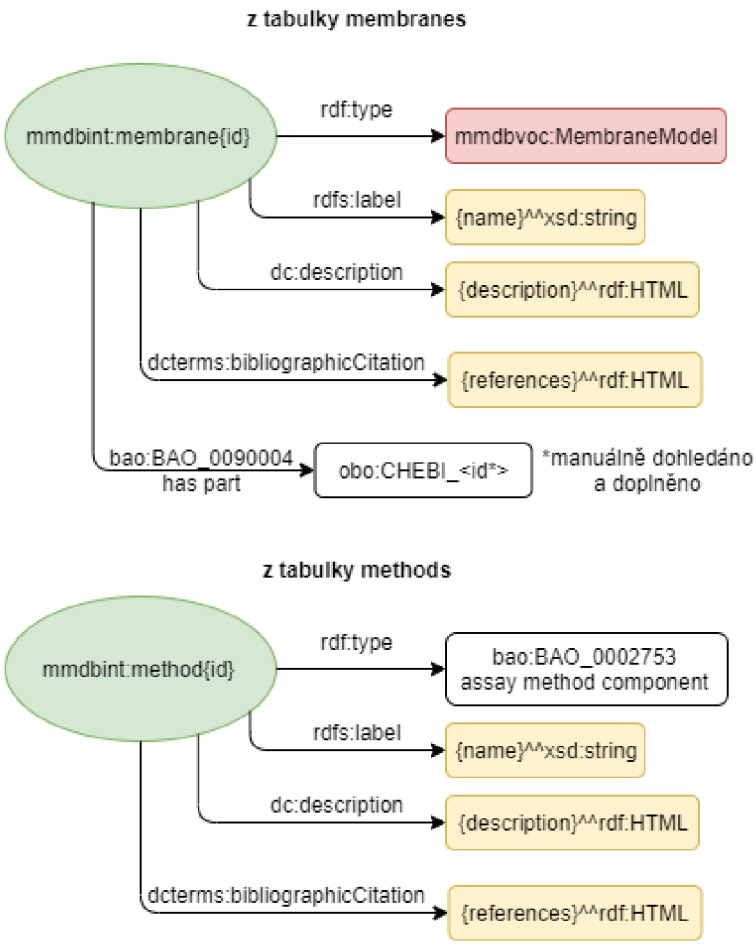
Tab. 7: Část schématu tabulek *methods* a *membranes*.

Field	Type	Null	Key
id	int(11)	NO	PRI
type	int(11)	YES	UNI
name	varchar(150)	YES	
description	text	YES	
references	text	YES	

Tabulky pro membrány a metody mají stejná schémata (Tab. 7). RDF grafy umožňují popsat metody a membrány názvem, opět uvedeným obecným predikátem `rdfs:label`, typem a případným popisem v HTML formátu a citací na jeho zdroj což odpovídá možnostem popisu tabulek *membranes* a *methods* (Obr. 10). Uzel metody je určen typem `bao:BAO_0002753` (*assay method component*), který je velice obecný, avšak jeho podtřídy už jsou příliš úzce vymezené na to aby zahrnovaly širší spektrum výpočetních a experimentálních metod. Pro uzel membrány byl vytvořen vlastní typ s IRI `mmdbvoc:MembraneModel`. Termíny pro membrány v jiných ontologiích jsou zatím příliš úzce vymezené a obvykle popisují buněčnou membránu se všemi jejími komponenty, což neodpovídá membránovým modelům použitým ve výpočtech a mnohdy ani experimentech.

Pro membrány bylo také vytvořeno mapování na jejich lipidové komponenty v databázi **ChEBI**. Jde ale jen o hrubé manuální propojení uzlů některých membrán s IRI molekul v ChEBI ontologii za pomoci predikátu `bao:BAO_0090004` (*has part*). Pro některé komponenty byly v databázi ChEBI vytvořeny nové záznamy (CHEBI:183652, CHEBI:183653, CHEBI:183654).

Transportérové interakce jsou uloženy v tabulce *transporters* (Tab. 8). Uzly pro transportérová měření (Obr. 9) jsou typu `bao:BAO_0003008` (*transporter assay*). Datový model je organizován velice podobným způsobem jako v případě membránových měření.



Obr. 10: Diagramy RDF grafů membrána metod. Ve složených závorkách je uveden atribut z příslušné tabulky. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, červené jsou termíny ze slovníku MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a uzly RDF jiných databází.

Tab. 8: Část schématu tabulky *transporters*.

Field	Type	Null	Key
id	int(11)	NO	PŘI
id_substance	int(11)	NO	MUL
id_target	int(11)	NO	MUL
type	int(150)	NO	
note	text	YES	
Km	float	YES	
Km_acc	float unsigned	YES	
EC50	float	YES	
EC50_acc	float unsigned	YES	
Ki	float	YES	
Ki_acc	float unsigned	YES	
IC50	float	YES	
IC50_acc	float unsigned	YES	

Tab. 9: Část schématu tabulky *transporter_targets*.

Field	Type	Null	Key
id	int(11)	NO	PŘI
name	varchar(150)	NO	
uniprot_id	varchar(50)	NO	UNI

Komplikovanější situace je v případě výsledku na typ interakce. Hodnoty pro tento atribut jsou v tabulce *transporters* čísla a ty jsou namapována k názvu interakce. Možných interakcí, se kterými MolMeDB počítá je celá řada a zahrnují typy jako neurčitý druh interakce nebo neznámý druh interakce, což vytvoření jasného datového modelu komplikuje. Nicméně se v současnosti (květen 2022) v databázi vyskytují jen 4 typy – *substrate*, *nonsubstrate*, *inhibitor* a *noninhibitor*. K těmto typům je přistupováno jako k výsledku měření, jestli ligand s transportérem vykazuje nebo nevykazuje substrátovou či inhibiční aktivitu a výsledek se udává jako pozitivní nebo negativní. K jejich záznamům byly vytvořeny třídy `mmdbvoc:SubstrateBindingAssay` a `mmdbvoc:InhibitionAssay` jako podtřídy `bao:BAO_0080024` (*binary endpoint*).

Transportérové proteiny jsou popsány tabulkou *transporter_targets* (Tab. 9). RDF graf pro transportéry sdílí jmenný prostor s transportérovými měřeními (Obr. 9). Uzel transportéru má typ označující transportér a je popsaný názvem na který odkazuje opět predikát `rdfs:label` a PDB ID na který odkazuje predikát `obo:ERO_0000547` (*has_UniPROT_ID*). IRI uzlu transportéru je vytvořen ze vzoru perzistentní URL pro protein v databázi **UniProt** a *uniprot_id* transportéru ve formě `http://purl.uniprot.org/unirprot/{`*uniprot_id*`}`.

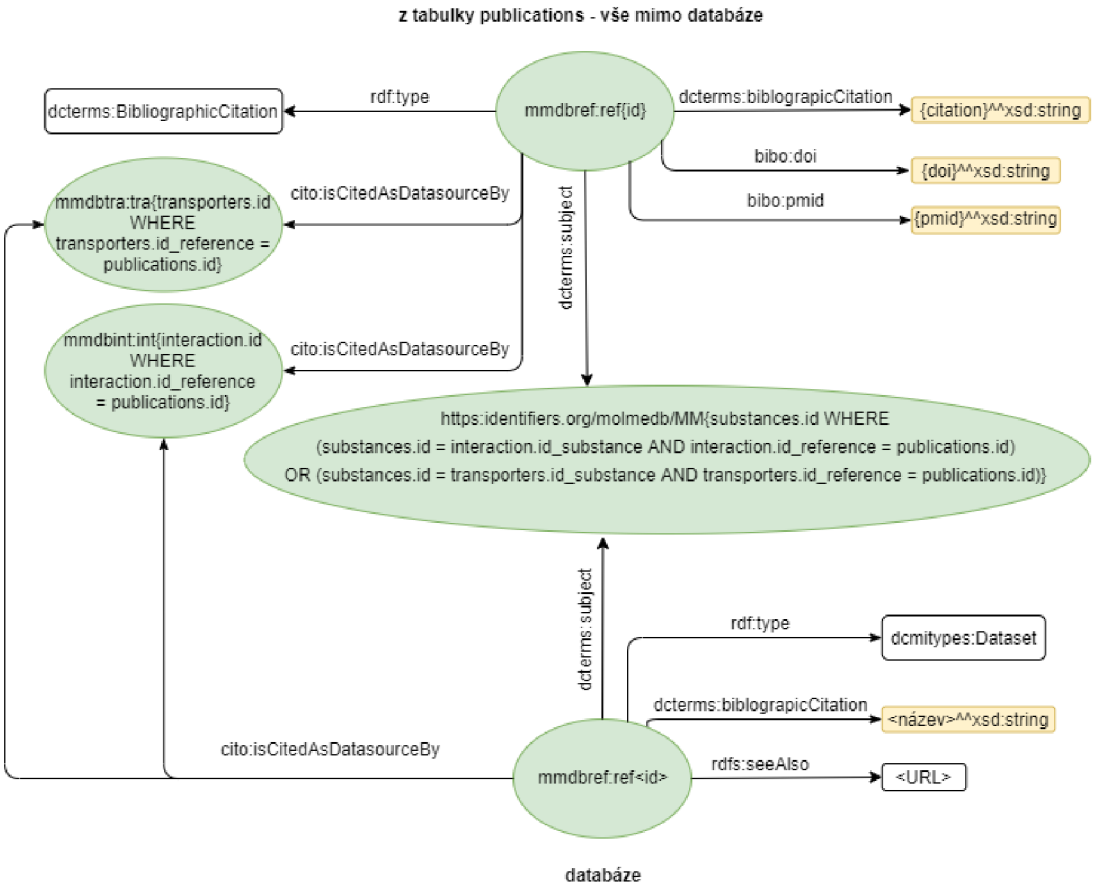
3.2.3 Publikace

Datový model pro zdroje (Obr. 11) byl částečně inspirován RDF modelem pro dokumenty ChEMBLu a PubChem Reference. Nicméně v prvé řadě bylo záměrem efektivně reprezentovat údaje v tabulce *publications* (Tab. 10).

Popis databází má jiné nároky než popis publikací. Vzhledem k tomu že se v tabulce *publications* nachází jen 5 databází, které nejsou nijak odlišené od publikací, byl pro databáze vytvořen samostatný model a jich dataset vypracován manuálně.

Tab. 10: Část schématu tabulky *publications*.

Field	Type	Null	Key
id	int(11)	NO	PRI
citation	text	YES	
doi	varchar(60)	YES	UNI
pmid	int(11)	YES	UNI
total_passive_interactions	int(11)	YES	
total_active_interactions	int(11)	YES	
total_substances	int(11)	YES	



Obr. 11: Schéma RDF grafu zdrojů. Ve složených závorkách je uveden atribut z příslušné tabulky. Ve špičatých závorkách jsou uvedené manuálně doplňované hodnoty. Elipsoidní jsou centrální uzly záznamů. Zelené uzly jsou organizační uzly záznamů MolMeDB, žluté jsou literály a bílé jsou termíny z externích slovníků a vnější odkazy.

Centrální uzel pro databáze má typ dcmitypes:Dataset. Pomocí predikátu dcterms:bibliographicCitation se odkazuje na název, tak jak je uveden ve sloupci atributu *citation* a pomocí rdfs:seeAlso na URL databáze.

Centrální uzel publikace má typ dcterms:BibliographicResource. Pomocí predikátu dcterms:bibliographicCitation se odkazuje na hodnotu atributu *citation* tabulky *publications*. Tento atribut je v mnoha případech (mimo primárního klíče) jediný bez hodnoty NULL. Mapování relačních záznamů na RDF předpokládá že v těchto případech jsou pro tento atribut uvedeny dostatečné údaje k jednoznačné identifikaci dokumentu lidským uživatelem. Z uzlů záznamů s existujícími příslušnými údaji vedou predikáty bibo:pmid a bibo:doi na hodnoty ze sloupců atributů *pmid* a *doi* s příslušnými identifikátory. Další atributy jako *journal*, *authors*, *year* atd. byly vzhledem k vysoké variabilitě způsobu zápisu a velmi časté absenci vyhodnoceny jako špatně použitelné a pro identifikaci publikace vedle citace, *doi* a *pmid* jako redundantní a nejsou tedy reprezentovány RDF modelem. V případě že by se situace změnila, může být model rozšířen.

Tabulka *publications* mimo jiné uvádí i počty interakcí a substancí pro jednotlivé zdroje. Místo toho byl zvolen přímější přístup kdy se centrální uzel pro databáze i publikace odkazuje na centrální uzly substancí pomocí dcterms:subject a na centrální uzly membránových i transportérových interakcí pomocí cito:isCitedAsDataSourceBy.

3.3 Slovník MolMeDB

Pro vyjádření zdrojů, pro které nebylo nalezeno vhodné URI byly zavedeny dočasné termíny. Vesměs se jedná o typy rozšiřující ontologii BioAssays o nové podtřídy. Jeden termín je individuum jednotky ze třídy ze slovníku NCIT, ze kterého byly použity ostatní jednotky v datovém modelu. Slovník MolMeDB je vyjádřen pomocí jazyka OWL 2 a je součástí přílohy (V aktuální verzi je rovněž dostupný na adrese https://github.com/DominikMartinat/rdf-mmdb). Všechny uvedené termíny patří do jmenného prostoru `mmdbvoc`. Slovník je součástí přílohy.

3.3.1 Třídy endpointů transportérových měření

- `:PKm` – Záporný logaritmus koncentrace substrátu při polovině maximální rychlosti enzymatické reakce. Podtřída `bao:BAO_0000477` (K_m).
- `:InhibitionAssay` – Výsledek měření zda sloučenina inhibuje enzym. Podtřída `bao:BAO_0080024` (*binary endpoint*).
- `:SubstrateBindingAssay` – Výsledek měření zda enzym váže sloučeninu jako substrát. Podtřída `bao:BAO_0080024` (*binary endpoint*).

3.3.2 Třídy endpointů membránových měření

- `:LogK` – Logaritmus partičního koeficientu mezi oktanolem a vodou. Podtřída `bao:BAO_0002128` (*physical property endpoint*).
- `:LogPerm` – Logaritmus membránové permeability. Podtřída `bao:BAO_0002808` (*permeability*).
- `:PositionOfMinima` – Vzdálenost minima křivky potenciální energie od středu membrány podél normály. Podtřída `bao:BAO_0002118` (*graphical calculation endpoint*).
- `:PenetrationBarrier` – Penetrační bariéra (rozdíl hodnoty minima a maxima na křivce potenciální energie). Podtřída `bao:BAO_0002118` (*graphical calculation endpoint*).
- `:DepthOfMinima` – Hloubka minima uvnitř membrány. Podtřída `bao:BAO_0002118` (*graphical calculation endpoint*).

- :ContactAngle – Úhel kontaktu na membráně. Podtřída bao:BAO_0080028 (*quantified endpoint*).
- :AbsorbtionWavelength – Vlnová délka světla při maximální absorbci. Podtřída bao:BAO_0002118 (*graphical calculation endpoint*).
- :FluorescenceWavelength – Vlnová délka světla při maximální fluorescenci. Podtřída bao:BAO_0002118 (*graphical calculation endpoint*).
- :QuantumYield – Počet výskytů jevu připadající na absorbci jednoho fotonu systémem. Podtřída bao:BAO_0000179 (*endpoint*).
- :FluorescenceLifetime – Doba trvání excitovaného stavu fluoroforu před emisí fotonu a návratem do základního stavu. Podtřída bao:BAO_0002114 (*time endpoint*).

3.3.3 Ostatní třídy

- :MembraneModel – Model biologické membrány použitý při experimentálním měření nebo výpočtu interakcí molekul s membránou. Zahrnuje buněčné membrány, modelové lipidové membrány, nelipidové membránové modely (např. oktanol) atd. Podtřída bao:BAO_0003114 (*assay biology component*).

3.3.4 Individua jednotek

- :kcalMol – Odvozená jednotka energie odpovídající 1000 kaloriím na jeden mol. Instance třídy obo:NCIT_C70444 (*Unit of Molar Energy*).

3.4 Generování a publikace RDF datasetu

Již v rané fázi projektu byl projeven zájem o RDF dataset MolMeDB provozovateli Integrované databáze malých molekul (IDSM), což je databáze integrující RDF datasety (Galgonek & Vondrášek, 2021). Více o IDSM v části 2.7.1 Integrace RDF datasetů. Bylo tedy rozhodnuto využít pro generování RDF datasetu a službu SPARQL kapacity IDSM.

Během akce Biohackathon Europe 2021 byla v rámci projektu FAIR Lipids vytvořena první verze schématu RDF systému MolMeDB a z něj pomocí jazyka R2RML mapování z relačního do sémantického datasetu společně se slovníkem v jazyce OWL. Mapování a slovník byly společně s relačním datasetem MolMeDB předány Jakubu Galgonkovi, který s pomocí mapování vygeneroval dataset obsluhovaný SPARQL službou s endpointem v rámci IDSM (Bolleman et al., 2021).

V IDSM bylo pro MolMeDB vytvořeno relační schéma pro PostgreSQL databázi navržené s ohledem na zamýšlené RDF schéma. Poté bylo vytvořeno mapování ze schématu PostgreSQL databáze na zamýšlené RDF schéma pro překladač mezi SPARQL službou a PostgreSQL. Nakonec byl relační dataset MolMeDB uložen do připravené databáze IDSM a byl vytvořen příslušný SPARQL endpoint.

RDF dataset generovaný IDSM byl dále upravován společně s tím, jak se vyvíjelo schéma RDF MolMeDB do současné podoby. Tento dataset je možno dotazovat pomocí SPARQL endpointu na adrese https://idsm.elixir-czech.cz/sparql/endpoint/molmedb. Celý dataset je možno stáhnout ve formátech RDF/XML, N-triples a Turtle na webu MolMeDB.

Použité mapování je ve formátu R2RML součástí přílohy. Aktualizované verze mapování jsou přístupné na adrese https://github.com/DominikMartinat/rdf-mmdb.

3.5 Dereference

Pro URI RDF MolMeDB byla vyhrazena subdoména **rdf.molmedb.upol.cz**. Veškeré dotazy zaslané na zmíněnou adresu budou pomocí reverzního proxy předány na **REST API** databáze. Samotné funkce pro dereferenci byly tedy implementovány jako nové endpointy REST API.

Generování reprezentace zdroje je implementováno pomocí dotazování **SPARQL** endpointu (viz předchozí část). Pro strojově čitelné formáty (**RDF/XML**, **Turtle**, **N-Triples, CSV, TSV**) je připravena šablona URL jednoduchého DESCRIBE dotazu, do něhož se dosadí dereferovaná URI a dojde k přesměrování na SPARQL endpoint. Formát odpovědi je určen příznakem Accept hlavičky HTTP žádosti.

HTML reprezentace se generuje pomocí jednoduchých SELECT dotazů za použití funkcí modifikované PHP knihovny SPARQLlib. Trojice, kde URI figuruje jako subjekt a kde jako objekt se generují zvlášť, v obou případech ale obdobně. Nejprve se pomocí SPARQL dotazu na predikáty v hledaných trojicích vygeneruje dvojrozměrné pole jejich popisů (každý takový popis predikátu se skládá z jeho celé URI, zkrácené URI a, v případě že existuje, názvu který je v ontologii k predikátu připojen pomocí `rdfs:label`), poté je pro každý predikát zaslán SPARQL dotaz zjišťující, které uzly jsou těmito predikáty spojeny s dereferovanou URI. Oproti popisu predikátů obsahuje popis nepovinnou položku typ literálu. Hodnota literálu se uloží na stejnou pozici, kam by se ukládala URI a pole popisů uzlů se pak připojí jako součást popisu příslušného predikátu. Příklad takového pole ukazuje obrázek 12.

Počet generovaných uzlů ke každému predikátu je omezen pomocí LIMIT ve SPARQL dotazu, což značně zvyšuje rychlost vyhodnocení dotazu a zátěž při následném generování HTML dokumentu. V současnosti je limit stanoven na 300 uzlů na predikát. Výsledkem této části je pole popisů predikátů doplněných o popisy uzlů a indikátor, zda bylo pro některý predikát dosaženo limitu.

<table>
<tr><th>predikát krátká URI</th><th>predikát celá uri</th><th>predikát název</th><th>pole popipsů objektů</th></tr>
<tr><td>rdfs:label</td><td>http://www...</td><td>-</td><td><table><tr><th>objekt krátká URI</th><th>objekt celá URI</th><th>objekt název</th><th>objekt typ</th></tr><tr><td>"Caffeine"</td><td>"Caffeine"</td><td>-</td><td>http://www...</td></tr></table></td></tr>
<tr><td>sio:SIO_000008</td><td>http://sem...</td><td>-</td><td><table><tr><th>objekt krátká URI</th><th>objekt celá URI</th><th>objekt název</th><th>objekt typ</th></tr><tr><td>mmdbsub:MM...</td><td>https://rdf...</td><td>-</td><td>-</td></tr><tr><td>mmdbsub:MM...</td><td>https://...</td><td>-</td><td>-</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr></table></td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
</table>

Obr. 12: Vícerozměrné pole pro generování HTML reprezentace kofeinu. Položkami prvního rozměru jsou jednorozměrná pole odpovídajícím řádkům. Poslední položkou pole na řádku je další vícerozměrné pole popisující objekty, které predikát k URI kofeinu připojuje.

Z takto získaných údajů se generuje HTML dokument v pohledu (Obr. 13). Pohled dostává na vstupu dvě pole a dva indikátory z předchozí části popisující vstupní a výstupní trojce dereferované URI. HTML reprezentace je generována formou dvou tabulek, nejprve pro výstupní RDF trojice a poté pro vstupní RDF trojice. URI jsou prezentovány ve své zkrácené formě a nezkrácené URI jsou použity pro vytvoření odkazů. Pokud byl nalezen název uzlu, je uveden v závorkách za URI. Literály jsou uvedeny v uvozovkách a jsou následovány URI jejich typu.

V případě, že bylo indikováno dosažení limitu, je nad příslušnou tabulkou uvedeno varování, že nejsou zobrazeny všechny trojice a je nabídnut odkaz na vyhledání všech trojic, ve kterých dereferovaná URI figuruje pomocí SPARQL dotazu DESCRIBE.

https://rdf.molmedb.upol.cz/interaction/membrane54

outgoing triples

predicate	object
rdfs:label	"1-octanol"^^http://www.w3.org/2001/XMLSchema#string
rdf:type	mmdbvoc:MembraneModel (membrane model)
dc:description	"n-Octanol is well used as a model solvent mimicking interactions of compounds with membrane. "^^http://www.w3.org/2001/XMLSchema#string
bao:BAO_0090004	obo:CHEBI_16188

incoming triples

WARNING: There are more subjects, than can be shown here.
To see complete resource description refer to SPARQL service HERE.

predicate	subject
bao:BAO_0090012	mmdbint:int101944
	mmdbint:int125502
	mmdbint:int133400
	mmdbint:int137778
	mmdbint:int142922

Obr. 13: HTML reprezentace uzlu mmdbint:membrane54. V tomto případě byl dosažen limit počtu objektů vstupních trojic a bylo vydáno varování.

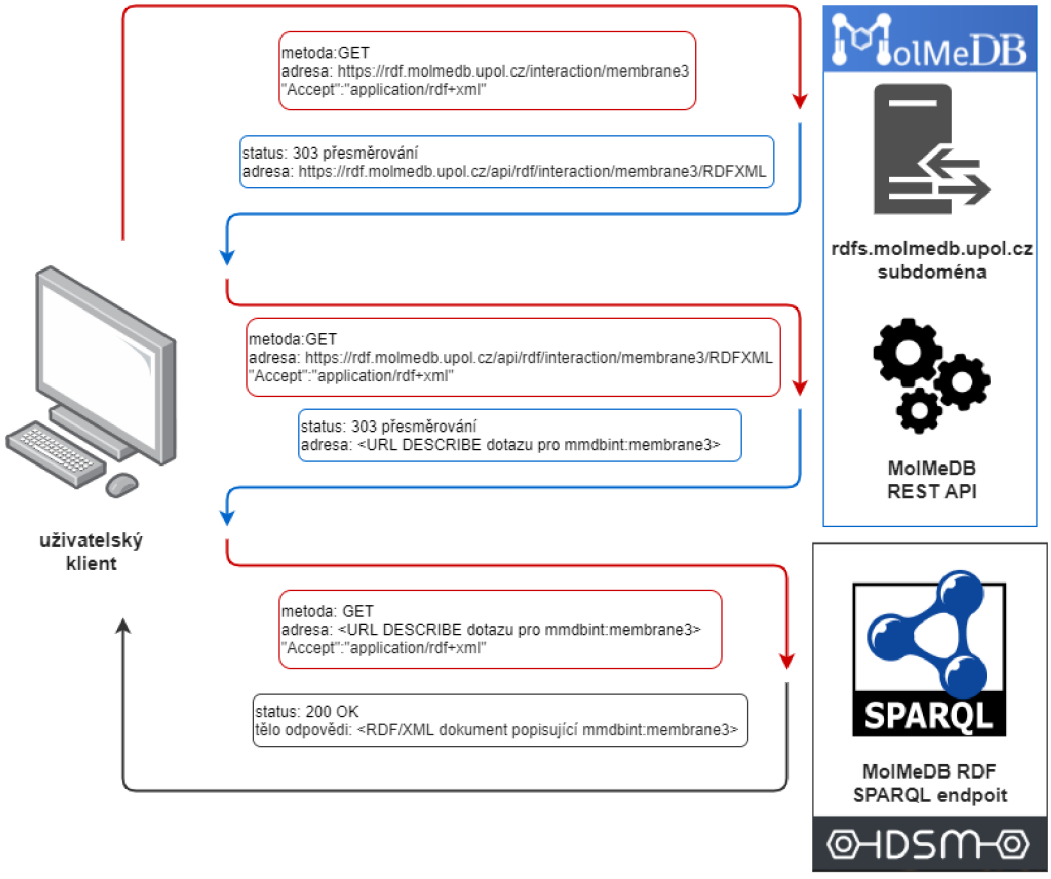
Ilustračním příkladem může být dereference URI pro membránu DOPC https://rdf.molmedb.upol.cz/interaction/membrane3. Klient pošle žádost o dereferenci URI jako HTTP požadavek. Dereferenční funkce rozhodne o dalším postupu podle hodnoty v příznaku *Accept* v hlavičce.

Pokud je požadován strojově čitelný formát dojde k přesměrování na URL https://rdf.molmedb.upol.cz/api/rdf/interaction/membrane3/RDFXML. Funkce REST API vytvoří URL pro příslušný DESCRIBE dotaz pro SPARQL endpoint a dojde k dalšímu přesměrování na tuto adresu. Výsledná odpověď má v hlavičce stavový kód 200 OK a v těle požadovaný dokument. Uvedený proces ilustruje obrázek 14.

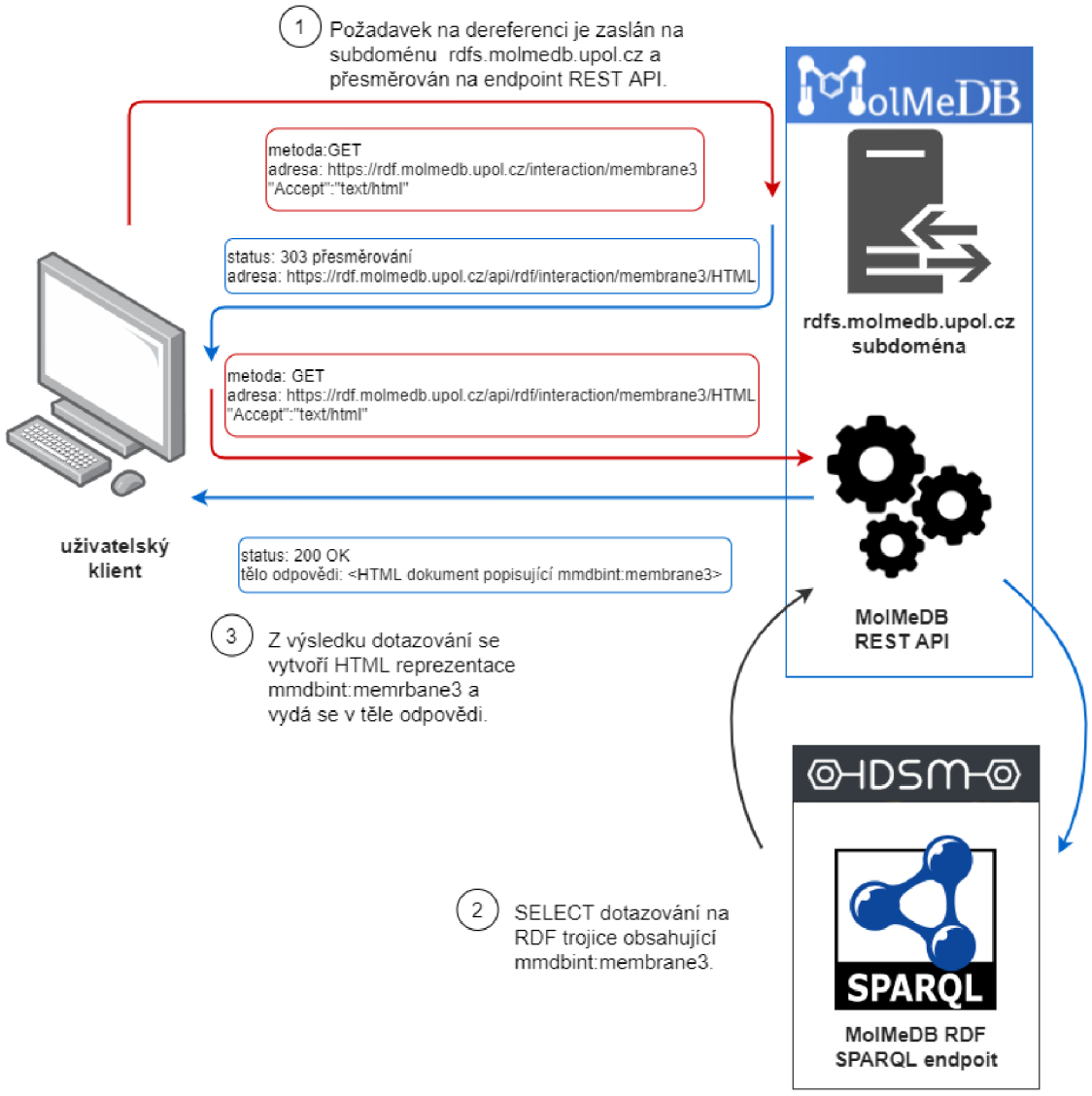
Pokud je požadován HTML dokument, proběhne přesměrování na https://rdf.molmedb.upol.cz/api/rdf/interaction/membrane3/HTML a generování HTML reprezentace podle výše zmíněného postupu (Obr. 15).

Zvláštním případem jsou URI slovníku a molekul. U slovníku probíhá dereference pomocí hash URI a podporován je zatím jen XML formát.

URI molekul se při požadavku vrácení HTML dokumentu přesměruje na stránku molekuly v databázi MolMeDB.



Obr. 14: Dereference URI mmdbint:membrane3 na RDF/XML dokument. Červeně je značena komunikace ze strany uživatelského klienta, modře ze strany webového serveru MolMeDB a šedě ze strany serveru IDSM.



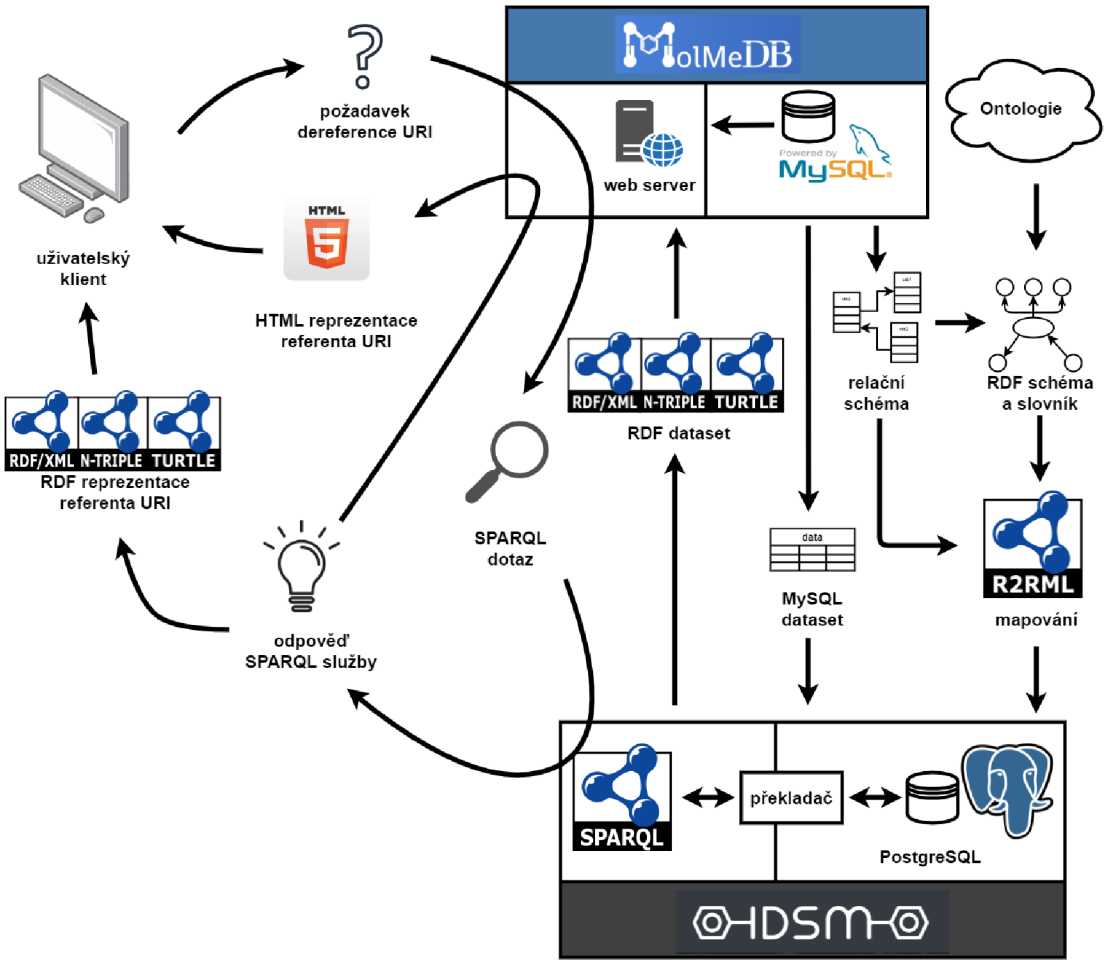
Obr. 15: Dereference URI `mmdbint:membrane3` na HTML dokument. Červeně je značena komunikace ze strany uživatelského klienta, modře ze strany webového serveru MolMeDB a šedě ze strany serveru IDSM.

4. VÝSLEDKY

Výsledným produktem této práce je **RDF verze databáze MolMeDB**. Systém obsluhující RDF dataset MolMeDB je distribuován mezi samotnou **MolMeDB** a **IDSM** (Obr. 16). Relační databáze MolMeDB slouží v tomto systému jako zdroj dat a relačního schématu podle kterého se konstruuje RDF schéma a slovník k jeho vyjádření.

Relační dataset a mapování z relačního do RDF schématu se předávají IDSM, kde jsou data uloženy do připraveného schématu PostgreSQL databáze. IDSM pak umožňuje dotazování datasetu pomocí SPARQL endpointu. Pomoci tohoto endpointu jsou také generovány **RDF datasety** předávané MolMeDB. Ty jsou ke stažení na adrese webu https://molmedb.upol.cz**.** Podporované formáty jsou **RDF/XML**, **Tutle** a **N-Triples**.

SPARQL endpoint se využívá i při vyřizování žádostí o dereferenci. Podporováno je vyjednávání o formátu vydaného dokumentu pomocí příznaku *Accept* v hlavičce žádosti.



Obr. 16: Schéma výsledného systému obsluhujícího MolMeDB RDF dataset.

4.1 Propojená data

RDF MolMeDB splňuje všechna čtyři pravidla která Tim Berners-Lee (2009) definoval pro propojená data:

1. Použití URI pro názvy entit.
2. Použití HTTP URI pro umožnění vyhledávání těchto názvů.
3. Při vyhledání URI poskytnutí informací za použití standardů (RDF, SPARQL).
4. Poskytnout odkazy na další URI pro možnost objevení dalších věcí.

První tři pravidla jsou samo popisná. První pravidlo RDF MolMeDB splňuje jakožto RDF databáze. Druhé pravidlo splňuje taktéž, všechny URI datasetu byly vygenerovány jako HTTPS URI. Termíny slovníku MolMeDB i použitých ontologií jsou taktéž identifikovány pomocí HTTP a HTTPS URI. Třetí pravidlo je splněno díky systému dereference URI RDF datasetu a slovníku MolMeDB a SPARQL endpointu.

Čtvrté pravidlo je zamýšleno primárně pro odkazy na související externí materiály. RDF dataset tento požadavek splňuje. Externí odkazy v tomto datasetu navíc hrají zásadní roli pro interoperabilitu s jinými databázemi. Záznamy molekul se odkazují pomocí predikátu `skos:exacMatch` na URI svých protějšků v RDF databází **PubChem**, **ChEMBL**, **ChEBI** a **PDBj**. Pro transportérové proteiny byly použity URI databáze **UniProt** a záznamy membrán obsahují odkaz na své komponenty v databázi **ChEBI**.

Použití těchto odkazů umožňuje snadno rozeznat, kdy záznamy reprezentují stejný zdroj a usnadňují tak práci s propojenými datasety pomocí federovaných SPARQL dotazů. Například RDF PubChem obsahuje informace o výsledcích screeningových měření a účasti v metabolických drahách pro velkou část molekul popisovaných v MolMeDB. Obdobně RDF ChEBI poskytuje informace o jejich struktuře, roli, klasifikaci a vzájemných vztazích. Tyto a další informace je pak možno kombinovat s daty MolMeDB. Příklady dotazů využívající interoperability jsou uvedeny v další části.

Nicméně sémantická síť, na kterou je RDF MolMeDB pomocí těchto odkazů napojena zahrnuje i datasety dalších databází. Ty se např. mohou napojovat do stejných databází pomocí stejných URI jako RDF MolMeDB. Databáze LIPID MAPS se odkazuje do databáze ChEBI. Díky tomu je např. možné identifikovat v této databázi některé z lipidových komponent membrán v MolMeDB.

4.2 Příklady federovaného dotazování

Možnost federovaného dotazování má zásadní přínos pro využitelnost informací v databázi MolMeDB. Umožňuje filtrovat výsledky podle dalších údajů, srovnávat napříč datasety a celkově zasazuje informace o membránových a transportérových interakcích do širšího kontextu znalostí z biomedicínské domény.

Následují příklady federovaných dotazů využívajících tato propojení. Federované dotazy zároveň demonstrují funkčnost SPARQL endpointu obsluhujícího dataset RDF MolMeDB. Všechny tyto dotazy mohou být vyhodnoceny endpointem na adrese https://idsm.elixir-czech.cz/sparql/endpoint/molmedb .

První dotaz (kód 9) demonstruje spojení s databází **ChEBI**. Při vyhodnocení dotazu jsou nejprve v databázi ChEBI vyhledány sloučeniny, které mají roli *proléčivo*. Sloučeniny jsou v RDF ChEBI vedeny jako třídy a jejich role se jim přiděluje tak, že se definují jako podtřída OWL restrikce, jejíž prvky jsou spojeny predikátem *has_role* s danou rolí. V další části dotazu se pro takto identifikovaná proléčiva vyhledají jejich protějšky v MolMeDB.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT * WHERE {
  SERVICE <https://idsm.elixir-czech.cz/sparql/endpoint/idsm>
  {
  ?prodrug rdfs:subClassOf [ rdf:type owl:Restriction;
      owl:onProperty obo:RO_0000087;
      owl:someValuesFrom obo:CHEBI_50266 ] .
  }
  ?substance skos:exactMatch ?prodrug;
             rdfs:label ?name
}
```

Kód 9: Federovaný SPARQL dotaz identifikující proléčiva v databázi MolMeDB pomocí databáze ChEBI. URL dotazu: https://tinyurl.com/29ncs3ec

Druhý dotaz (Kód 10) demonstruje propojení s **RDF PubChem** a strukturní vyhledávání pomocí funckionality **Sachem**. Nejprve jsou v RDF PubChem vyhledány léčiva schválené americkým Úřadem pro kontrolu potravina léčiv (FDA). Ty jsou poté opět identifikovány s jejich protějšky v MolMeDB. SMILES těchto molekul jsou poté zadány jako vstup pro strukturní vyhledávání funkcionality Sachem, pomocí které je zjištěno, ke kterým proléčivům vybraným z výsledku předchozího dotazu léčivo patří jako jeho substruktura. Výsledkem je seznam léčiv, jejich názvů a názvů proléčiv.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX vocab: <http://rdf.ncbi.nlm.nih.gov/pubchem/vocabulary#>
PREFIX sachem: <http://bioinfo.uochb.cas.cz/rdf/v1.0/sachem#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX substance: <https://identifiers.org/molmedb/>

SELECT ?FDA_drug ?drug ?drug_name ?prodrug_name WHERE {
  SERVICE <https://idsm.elixir-czech.cz/sparql/endpoint/idsm>
  { ?FDA_drug obo:has-role vocab:FDAApprovedDrugs }
  ?drug skos:exactMatch ?FDA_drug;
        rdfs:label ?drug_name;
        sio:SIO_000008 [ rdf:type sio:CHEMINF_000018;
                         sio:SIO_000300 ?SMILES] .
  VALUES ?prodrug {substance:MM00470 substance:MM00629
                   substance:MM01437}
  ?prodrug sachem:substructureSearch [
    sachem:query ?SMILES].
  ?prodrug rdfs:label ?prodrug_name
  FILTER (?drug != ?prodrug)
}
```

Kód 10: Federovaný SPARQL dotaz identifikující k proléčivům z výsledku předchozího dotazu léčiva schválená FDA, která jsou podstrukturami těchto proléčiv. Využity jsou služby databáze PubChem a rozšíření Sachem. URL dotazu: https://tinyurl.com/drynxxu5

Databáze **ChEMBL** mimo jiné obsahuje informace o fázi schvalování molekul jako léčiv. Toho využívá dotaz z kódu 11, který zjišťuje, které substráty transportéru Q9UNQ0 jsou v klinické fázi testování.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX cco: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX bao: <http://www.bioassayontology.org/bao#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX mmdbvoc: <https://rdf.molmedb.upol.cz/vocabulary#>
PREFIX repr: <https://w3id.org/reproduceme#>
PREFIX uniprot: <http://purl.uniprot.org/uniprot/>

SELECT DISTINCT ?drug ?drug_name WHERE {
  SERVICE <https://idsm.elixir-czech.cz/sparql/endpoint/idsm>
  {
    ?drug cco:highestDevelopmentPhase "3"^^xsd:int
  }
  ?substance skos:exactMatch ?drug ;
             rdfs:label ?drug_name .
  ?assay bao:BAO_0090012 ?substance ;
    bao:BAO_0090012 uniprot:Q9UNQ0;
    bao:BAO_0000208 [ rdf:type mmdbvoc:SubstrateBindingAssay ;
                      bao:BAO_0095007 repr:PositiveResult ].
}
```

Kód 11: Federovaný SPARQL dotaz vyhledávající substráty proteinu Q9UNQ0, které jsou ve fázi klinického testování za pomoci databáze ChEMBL. URL dotazu: https://tinyurl.com/2p8c2frp

Propojení s RDF UniProt umožňuje například filtrování podle taxonů organismů, ze kterých protein pochází. V tomto příkladu se vyhledávají lidské transportéry inhibované lékem Verapamilem a příslušné hodnoty pIC_{50} (kód 12).

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX bao: <http://www.bioassayontology.org/bao#>
PREFIX up: <http://purl.uniprot.org/core/>
PREFIX taxon: <http://purl.uniprot.org/taxonomy/>
PREFIX mmdbvoc: <https://rdf.molmedb.upol.cz/vocabulary#>
PREFIX repr: <https://w3id.org/reproduceme#>

SELECT ?transporter ?pIC50 WHERE {
  ?transporter rdf:type bao:BAO_0000283 .
  SERVICE <https://sparql.uniprot.org/sparql>
  {
    ?transporter up:organism taxon:9606
  }
  ?assay bao:BAO_0090012 ?transporter;
    bao:BAO_0090012 <https://identifiers.org/molmedb/MM00585>;
    bao:BAO_0000208 [ rdf:type mmdbvoc:InhibitionAssay;
                      bao:BAO_0095007 repr:PositiveResult ];
    bao:BAO_0000208 [ rdf:type bao:BAO_0000199;
                      bao:BAO_0002666 ?pIC50 ]
}
```

Kód 12: Federovaný SPARQL dotaz získávající hodnoty pIC_{50} pro všechny lidské transportéry inhibované lékem Verapamilem. URL dotazu: https://tinyurl.com/4uudae95

Poslední dotaz demonstrují funkčnost jiných typů propojení. Dotaz z kódu 13 vyhledává k membránám názvy jejich lipidových komponent z databáze LIPID MAPS.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bao: <http://www.bioassayontology.org/bao#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?membrane_name ?component ?lipid_name WHERE {
  ?membrane bao:BAO_0090004 ?component;
            rdfs:label ?membrane_name
  SERVICE <https://www.lipidmaps.org/sparql>
  {
    ?lipid owl:equivalentClass ?component;
           rdfs:label ?lipid_name
  }
}
```

Kód 13: SPARQL dotaz vyhledávající komponenty membrán v databázi Lipid Maps. URL dotazu: https://tinyurl.com/ywm8m97f

5. DISKUZE

Zavedení RDF výrazně zvýšuje využitelnost databáze MolMeDB zejména z pohledu strojové čitelnosti dat a umožňuje integraci s jinými projekty využívajícími RDF. Možnosti takového využití ukazují například SPARQL dotazy v předchozí kapitole. Implementace v rámci IDSM pak umožňuje využívat projekt Sachem pro strukturní dotazování (kód 4), což není obvyklou schopností RDF databází (Kratochvíl et al., 2019).

Schéma RDF MolMeDB je dále možné rozšiřovat o další uzly a odkazy, včetně odkazů do RDF jiných databází. Naopak RDF jiných databází se boudou moct odkazovat do RDF MolMeDB pomocí jeho URI, což kromě možností integrace dat bude mít přínos i ve zvýšení viditelnosti databáze a snadnějšímu přístupu k datům.

Do budoucna by bylo vhodné zavést ucelenou ontologii zabývající se metodami měření a výpočtů interakcí s membránami a modelovými membránami samotnými nebo rozšířit nějakou již zavedenou. Z těch se nabízí zejména ontologie BioAssay. Samotná MolMeDB by pak mohla být rozšířena o údaje o chemickém složení membránových modelů, což by umožnilo systematičtější mapování membrán na jejich komponenty v chemických databázích.

Zkušenosti s RDF některých databází ukázaly na potřebu vytvoření přehledné dokumentace shrnující datový model a použité slovníky. Bez ní je RDF dataset těžko přístupný lidskému uživateli, obzvlášť při hojném použití URI tvarů dostatečně nepopisujících jejich zdroj. Z tohoto důvodu je plánováno vytvoření a zveřejnění dokumentace také v anglickém jazyce.

Pro vyšší uživatelský komfort je v plánu vytvořit systém demonstračních dotazů, které budou ve spolupráci s IDSM umístěny u endpointu, podobně jako pro jiné zde integrované databáze. Tyto dotazy pak mohou tvořit základní kostru, kterou si může uživatel sám upravit tak, aby dotaz vyhovoval jeho potřebám.

Zdrojem těchto dotazů může být použití RDF MolMeDB v probíhajícím výzkumu vlivu funkčních skupin molekul na jejich interakce s membránami, kde například propojení s databází ChEBI umožňuje identifikovat, které látky v MolMeDB jsou léčivy nebo proléčivy a přiřadit léčiva jejich potenciálním proléčivům pomocí rozšíření Sachem (IDSM) pro vyhledávání molekulárních podstruktur.

6. ZÁVĚR

V této práci bylo popsáno vytvoření RDF verze databáze interakce molekul s membránami MolMeDB ke zlepšení jejich FAIR charakteristik. Pro vyjádření informací z relační databáze MolMeDB bylo vytvořeno RDF schéma inspirované schématy RDF PubChem a ChEMBL. Pro vyjádření vztahů a tříd v tomto schématu byly použity již zavedené ontologie jako BioAssasy, SIO nebo ontologie skupiny OBO. Byl také vytvořen vlastní slovník jako extenze BioAssay a NCIT.

Dále bylo vytvořeno mapování mezi relačním a RDF schématem. To pak bylo společně s relačním datasetem předáno integrované databázi malých molekul IDSM, kde byl pro tato data připraven SPARQL endpoint a překladač do RDF. Pomoci tohoto endpointu je zajištěna dereference URI a generování RDF datasetu, který je na webu MolMeDB volně ke stažení. Dataset je přístupný ve strojově čitelných formátech RDF/XML, Trutle a N-Triples. Dereference je krom těchto formátu možná i v CSV a TSV. Pro lidské uživatele je zajištěna dereference na HTML dokument.

RDF MolMeDB je pomocí odkazů na jejich URI propojeno s RDF verzemi PubChem, ChEBI, ChEMBL, PDBj a UniProt. Propojení existuje i s dalšími databázemi, které se na tyto URI odkazují, jako např. LIPID MAPS. Tyto databáze se pak napojují na další a propojují je tak s MolMeDB nepřímo. URI RDF datasetu MolMeDB pak umožňují jiným RDF databázím odkazování do tohoto datasetu.

Výsledkem je integrace databáze MolMeDB do sémantické sítě spojující biologické databáze. Díky tomuto zapojení je možné provádět komplexní dotazování využívající zároveň obsah MolMeDB i ostatních databází. Příkladem využití této schopnosti je využití RDF MolMeDB ve v současnosti probíhajícím výzkumu vlivu funkčních skupin molekul na jejich membránové interakce.

7. CONCLUSION

This thesis describes the creation of an RDF version of the molecule on membranes interaction database MolMeDB to increase its FAIRness. An RDF schema inspired by the RDF PubChem and ChEMBL schemas was created to express information from the MolMeDB relational database. Established ontologies such as BioAssasy, SIO or OBO ontologies were used to express relationships and classes in this scheme. A custom vocabulary was created as an extension of BioAssay and NCIT ontologies.

After that, a mapping between relational and RDF schema was created. This was then passed together with the relational dataset to the integrated small molecule database IDSM, where an SPARQL endpoint and an RDF translator were prepared for this data. This endpoint is used for URI dereference and RDF dataset generation. Dataset is accessible from MolMeDB website. The dataset is available in machine-readable RDF/XML, Trutle and N-triples formats. In addition to these formats, dereference is also possible into CSV or TSV document. For human users dereference into HTML document is provided.

The RDF MolMeDB is linked to the RDF versions PubChem, ChEBI, ChEMBL, PDBj and UniProt using links to their URIs. There are also links to other databases that reference these URIs, such as LIPID MAPS. These databases are then connected to others and thus link them to MolMeDB indirectly. The RDF URIs of the MolMeDB dataset then allow other RDF databases to reference this dataset.

The result is the integration of the MolMeDB database into a semantic network connecting biological databases. Thanks to this involvement, it is possible to perform complex queries using both MolMeDB content and other databases. An example of the benefits of this capability is the use of RDF MolMeDB in the current research into the influence of functional groups of molecules on their membrane interactions.

8. LITERATURA

Abeyruwan, S., Vempati, U. D., Küçük-McGinty, H., Visser, U., Koleti, A., Mir, A., Sakurai, K., Chung, C., Bittker, J. A., Clemons, P. A., et al. (2014). Evolving BioAssay Ontology (BAO): modularization, integration and applications. *Journal of Biomedical Semantics*, *5*(S1). https://doi.org/10.1186/2041-1480-5-S1-S5

Allemang, D., Hendler, J., & Gandon, F. (2020). *Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS and OWL* (Third Edition). Association for Computing Machinery, New York, NY, United States

Aranda, C. B., Corby, O., Das, S., Feigenbaum, L., Gearon, P., Glimm, B., Harris, S., Hawke, S., Herman, I., et al (2013). *SPARQL 1.1 Overview*. W3C. Retrieved May 3, 2022, from https://www.w3.org/TR/sparql11-overview/

Arenas, M., Bertails, A., & Sequeda, J. (Eds.). (2012). *A Direct Mapping of Relational Data to RDF*. W3C. Retrieved January 2, 2022, from https://www.w3.org/TR/rdb-direct-mapping/

Balhoff, J., Brush, M., & Vasilevsky, N. *NCIt OBO Edition*. GitHub. Retrieved September 9, 2021, from https://github.com/NCI-Thesaurus/thesaurus-obo-edition

Belleau, F., Nolin, M. -A., Tourigny, N., Rigault, P., & Morissette, J. (2008). Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5), 706-716. https://doi.org/10.1016/j.jbi.2008.03.004

Benatallah, B., & Motahari Nezhad, H. R. (2008). Service Oriented Architecture: Overview and Directions. In E. Börger & A. Cisternino (Eds.), *Advances in Software Engineering* (pp. 116-130). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89762-0_4

Berners-Lee, T. (1998). *Cool URIs don't change*. W3C. Retrieved April 14, 2022, from https://www.w3.org/Provider/Style/URI

Berners-Lee, T. (2009). *Linked Data*. W3C. Retrieved April 15, 2022, from https://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T., Bray, T., Connolly, D., Cotton, P., Fielding, R., Jeckle, M., Lilley, C., Meldensohn, N., Orchard, D., Walsh, N., & Williams, S., Jacobs, I., & Walsh, N. (Eds.). (2004). *Architecture of the World Wide Web: Volume One*. W3C. Retrieved January 18, 2021, from https://www.w3.org/TR/webarch/

Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *Uniform Resource Identifier (URI): Generic Syntax*. IETF. Retrieved May 5, 2022, from https://www.ietf.org/rfc/rfc3986.txt

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34 – 43. https://www.jstor.org/stable/26059207

Birbeck, M., & McCarron, S. (Eds.). (2010). *CURIE Syntax 1.0*. W3C. Retrieved May 5, 2022, from https://www.w3.org/TR/2010/NOTE-curie-20101216/

Bolleman, J., Galgonek, J., Andrews, R., Mendvelso, K., Gaud, C., Martinát, D., Hoffmann, N., Morgat, A., Bansal, P., Aimo, L., et al. (2021). *Fair Lipids paper*. GitHub. Retrieved May 5, 2022, from https://github.com/elixir-europe/biohackathon-projects-2021/blob/FAIR_Lipids_paper/projects/6/PAPER.md

Brickley, D., & Guha, R. V. (2014). *RDF Schema 1.1*. W3C. Retrieved September 9, 2021, from https://www.w3.org/TR/rdf-schema/

Cyganiak, R., Wood, D., & Lanthaler, M. (Eds.). (2014). *RDF 1.1 Concepts and Abstract Syntax*. W3C. Retrieved January 18, 2021, from https://www.w3.org/TR/rdf11-concepts/

D'Arcus, B., & Giasson, F. (2016). *BIBO (RDF)*. Dublin Core Metadata Inovation. Retrieved September 9, 2021, from https://www.dublincore.org/specifications/bibo/bibo/

Das, S., Sundara, S., & Cyganiak, R. (2012). *R2RML: RDB to RDF Mapping Language*. W3C. Retrieved January 2, 2022, from https://www.w3.org/TR/r2rml/

DCMI Usage Board. (2020). *DCMI Metadata Terms*. Dublin Core Metadata Inovation. Retrieved September 9, 2021, from https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

Duerst, M., & Suignard, M. (2005). *Internationalized Resource Identifiers (IRIs)*. IETF. Retrieved April 14, 2022, from https://www.ietf.org/rfc/rfc3987.txt

Dumontier, M., Baker, C. J. O., Baran, J., Callahan, A., Chepelev, L., Cruz-Toledo, J., Del Rio, N. R., Duck, G., Furlong, L. I., Keath, N., et al. (2014). The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery. *Journal of Biomedical Semantics*, *5*(1). https://doi.org/10.1186/2041-1480-5-14

Feigenbaum, L., Williams, G. T., & Torres, E. (Eds.). (2013). *SPARQL 1.1 Protocol*. W3C. Retrieved May 3, 2022, from https://www.w3.org/TR/sparql11-protocol/

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* [disertation]. University of California, Irvin, USA.

Fielding, R., & Reschke, J. (Eds.). (2014). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. IETF. Retrieved May 11, 2022, from https://datatracker.ietf.org/doc/html/rfc7231

Fu, G., Batchelor, C., Dumontier, M., Hastings, J., Willighagen, E., & Bolton, E. (2015). PubChemRDF: towards the semantic annotation of PubChem compound and substance databases. *Journal of Cheminformatics*, *7*(1). https://doi.org/10.1186/s13321-015-0084-4

Galgonek, J., & Vondrášek, J. (2021). IDSM ChemWebRDF: SPARQLing small-molecule datasets. *Journal of Cheminformatics*, *13*(1). https://doi.org/10.1186/s13321-021-00515-1

Glimm, B., & Ogbuji, C. (Eds.). (2013). *SPARQL 1.1 Entailment Regimes*. W3C. Retrieved May 4, 2022, from https://www.w3.org/TR/sparql11-entailment/

Gutteridge, C. (2012). *Sparqllib.php*. Graphite PHP Linked Data Library. Retrieved May 9, 2022, from http://graphite.ecs.soton.ac.uk/sparqllib/

Harris, S., & Seaborne, A. (Eds.). (2013). *SPARQL 1.1 Query Language*. W3C. Retrieved May 3, 2022, from https://www.w3.org/TR/2013/REC-sparql11-query-20130321/

Harrow, I., Balakrishnan, R., Jimenez-Ruiz, E., Jupp, S., Lomax, J., Reed, J., Romacker, M., Senger, C., Splendiani, A., Wilson, J., & Woollard, P. (2019). Ontology mapping for semantically enabled applications. *Drug Discovery Today*, *24*(10), 2068-2075. https://doi.org/10.1016/j.drudis.2019.05.020

Hastings, J., Chepelev, L., Willighagen, E., Adams, N., Steinbeck, C., Dumontier, M., & Fraternali, F. (2011). The Chemical Information Ontology: Provenance and Disambiguation for Chemical Data on the Biological Semantic Web. *PLoS ONE*, *6*(10). https://doi.org/10.1371/journal.pone.0025513

Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P., & Steinbeck, C. (2016). ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research*, *44*(D1), D1214-D1219. https://doi.org/10.1093/nar/gkv1031

Hitzler, P. (2021). A review of the semantic web field. *Communications of the ACM*, *64*(2), 76-83. https://doi.org/10.1145/3397512

Hitzler, P., Krötzsch, M., Parsija, B., Patel-Schneider, P. F., & Rudolph, S. (Eds.). (2012). *OWL 2 Web Ontology Language Primer: Second Edition*. W3C. Retrieved May 2, 2022, from https://www.w3.org/TR/owl-primer/

Hogan, A. (2016). Linked Data & the Semantic Web Standards. In A. Harth, H. Katja, & S. Ralf (Eds.), *Linked Data Management: Emerging directions in database systems and applications*. CRC Press.

Hyland, B., Atemezing, G., & Villazón-Terrazas, B. (Eds.). (2014). *Best Practices for Publishing Linked Data*. W3C. Retrieved April 15, 2022, from https://www.w3.org/TR/ld-bp/

Chen, B., Dong, X., Jiao, D., Wang, H., Zhu, Q., Ding, Y., & Wild, D. J. (2010). Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC Bioinformatics*, *11*(1). https://doi.org/10.1186/1471-2105-11-255

Isaac, A., & Summers, E. (Eds.). (2009). *SKOS Simple Knowledge Organization System Primer*. W3C. Retrieved May 7, 2022, from https://www.w3.org/TR/skos-primer/

Jackson, R., Matentzoglu, N., Overton, J. A., Vita, R., Balhoff, J. P., Buttigieg, P. L., Carbon, S., Courtot, M., Diehl, A. D., Dooley, D. M., et al. (2021). OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database*, *2021*. https://doi.org/10.1093/database/baab069

Jupp, S., Burdett, T., Malone, J., Leroy, C., Pearce, M., McMurry, J., & Parkinson, H. (2015). A New Ontology Lookup Service at EMBL-EBI. In J. Malone, R. Stevens, K. Forsberg, & A. Splendiani, *Proceedings of the 8th International Conference on Semantic Web Applications and Tools for Life Sciences* (pp. 118-119). http://ceur-ws.org/Vol-1546/paper_29.pdf

Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., Gaulton, A., Gehant, S., Laibe, C., Redaschi, N., et al. (2014). The EBI RDF platform: linked open data for the life sciences. *Bioinformatics*, *30*(9), 1338-1339. https://doi.org/10.1093/bioinformatics/btt765

Juračka, J. (2020). *Rozvoj a propojování chemoinformatických databází* [diplomová práce]. Univerzita Palackého v Olomouci, Česká Republika

Juračka, J., Šrejber, M., Melíková, M., Bazgier, V., & Berka, K. (2019). MolMeDB: Molecules on Membranes Database. *Database*, *2019*. https://doi.org/10.1093/database/baz078

Kanza, S., & Frey, J. G. (2019). A new wave of innovation in Semantic web tools for drug discovery. *Expert Opinion on Drug Discovery*, *14*(5), 433-444. https://doi.org/10.1080/17460441.2019.1586880

Kratochvíl, M., Vondrášek, J., & Galgonek, J. (2018). Sachem: a chemical cartridge for high-performance substructure search. *Journal of Cheminformatics*, *10*(1). https://doi.org/10.1186/s13321-018-0282-y

Kratochvíl, M., Vondrášek, J., & Galgonek, J. (2019). Interoperable chemical structure search service. *Journal of Cheminformatics*, *11*(1). https://doi.org/10.1186/s13321-019-0367-2

Laërtius, D., & Hicks, R. D. (1925). Book VI. In D. Laërtius & R. D. Hicks, *Lives of the Eminent Philosophers*. https://en.wikisource.org/wiki/Lives_of_the_Eminent_Philosophers/Book_VI

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2015). DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, *6*(2), 167-195. https://doi.org/10.3233/SW-140134

McCrae, J. P. (2022). *The Linked Open Data Cloud*. Retrieved May 6, 2022, from https://lod-cloud.net/

Miles, A., & Bechofer, S. (Eds.). (2009). *SKOS Simple Knowledge Organization System Reference*. W3C. Retrieved December 27, 2021, from https://www.w3.org/TR/skos-reference/

NCBO BioPortal. Retrieved May 3, 2022, from https://bioportal.bioontology.org/

Patel, A., & Jain, S. (2021). Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*, *43*(5), 413-422. https://doi.org/10.1080/1206212X.2019.1570666

Peterson, D., Gao, S., Malhotra, A., Sperberg-McQueen, C. M., & Thompson, H. S. (Eds.). (2012). *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. W3C. Retrieved May 7, 2022, from https://www.w3.org/TR/xmlschema11-2/

Pitt, C. (2012). *Pro PHP MVC*. Springer Science Business Media.

Prud'hommeaux, E., & Buil-Aranda, C. (Eds.). (2013). *SPARQL 1.1 Federated Query*. W3C. Retrieved May 4, 2022, from https://www.w3.org/TR/sparql11-federated-query/

Redaschi, N., & Consortium, U. P. (2009). UniProt in RDF: Tackling Data Integration and Distributed Annotation with the Semantic Web. *Nature Precedings*. https://doi.org/10.1038/npre.2009.3193.1

Samuel, S., & König-Ries, B. (2019). *REPRODUCE-ME Ontology*. Sheeba-samuel. Retrieved September 9, 2021, from https://sheeba-samuel.github.io/REPRODUCE-ME/doc/index-en.html

Samwald, M., Jentzsch, A., Bouton, C., Kallesøe, C. S., Willighagen, E., Hajagos, J., Marshall, M. S., Prud'hommeaux, E., Hassanzadeh, O., Pichler, E., & Stephens, S. (2011). Linked open drug data for pharmaceutical research and development. *Journal of Cheminformatics*, *3*(1). https://doi.org/10.1186/1758-2946-3-19

Sauermann, L., & Cyganiak, R. (Eds.). (2008). *Cool URIs for the Semantic Web*. W3C. Retrieved April 15, 2022, from https://www.w3.org/TR/cooluris/

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, *21*(3), 96-101. https://doi.org/10.1109/MIS.2006.62

Shotton, D., & Peroni, S. (2018). *CiTO, the Citation Typing Ontology*. SPAR Ontologies. Retrieved September 9, 2021, from https://sparontologies.github.io/cito/current/cito.html

Sioutos, N., Coronado, S. de, Haber, M. W., Hartel, F. W., Shaiu, W. -L., & Wright, L. W. (2007). NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, *40*(1), 30-43. https://doi.org/10.1016/j.jbi.2006.02.013

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., et al. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, *25*(11), 1251-1255. https://doi.org/10.1038/nbt1346

Stoeckert, C., Arabandi, S., Boelling, C., Brochhausen, M., Brush, M., Ceusters, W., Ciccarese, P., Courtot, M., Dipert, R., Dumontier, M., et al. (2020). *Information artifact ontology (IAO)*. GitHub. Retrieved September 9, 2021, from https://github.com/information-artifact-ontology/IAO/

Tan, S., Mungall, C., Vasilevsky, N., Matentzoglu, N., Osumi-Sutherland, D., Caron, A., reality, Dahdul, W., Blumberg, K., Balhoff, J., et al. (2021). *PATO - the Phenotype And Trait Ontology*. GitHub. Retrieved September 9, 2021, from https://github.com/pato-ontology/pato/

Torniai, C., Brush, M., Vasilevsky, N., Segerdell, E., Wilson, M., Johnson, T., Corday, K., Shaffer, C., & Haendel, M. (2011). Developing an Application Ontology for Biomedical Resource Annotation and Retrieval: Challenges and Lessons Learned. In *ICBO-2011 International Conference on Biomedical Ontology* (pp. 101-108). http://ceur-ws.org/Vol-833/paper14.pdf

Vocabularies. W3C. Retrieved September 9, 2021, from https://www.w3.org/standards/semanticweb/ontology

Vrandečić, D., & Krötzsch, M. (2014). Wikidata. *Communications of the ACM*, *57*(10), 78-85. https://doi.org/10.1145/2629489

Williams, A. J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E. L., Evelo, C. T., Blomberg, N., Ecker, G., Goble, C., & Mons, B. (2012). Open PHACTS: semantic interoperability for drug discovery. *Drug Discovery Today*, *17*(21-22), 1188-1198. https://doi.org/10.1016/j.drudis.2012.05.016

Willighagen, E. L., Waagmeester, A., Spjuth, O., Ansell, P., Williams, A. J., Tkachenko, V., Hastings, J., Chen, B., & Wild, D. J. (2013). The ChEMBL database as linked open data. *Journal of Cheminformatics*, *5*(1). https://doi.org/10.1186/1758-2946-5-23

Zdráhal, Z. (2013). Ontologie: od filosofie k umělé inteligenci. In V. Mařík, O. Štěpánková, & J. Lažanský (Eds.), *Umělá inteligence (6)*. Academia.

9. SEZNAM POUŽITÝCH SKRATEK

- **API** – Application Programming Interface
- **ASCII –** American Standard Code for Information Interchange
- **BAO –** BioAssay Ontology
- **BIBO –** Bibliographic Ontology
- **CiTO –** Citation Typing Ontology
- **CSV/TSV –** comma-separated values/ tab-separated values
- **CURIE –** Compact URI
- **DC –** Dublin Core
- **DNS –** Domain Name System
- **doi –** Digital Object Identifier
- **ERO –** Eagle-i Resource Ontology
- **FAIR –** Findability, Accessibility, Interoperability, and Reuse
- **FDA –** Food and Drug Administration
- **HTML –** HyperText Markup Language
- **HTS –** High throughput screening
- **HTTP –** Hypertext Transfer Protocol
- **HTTPS – Hypertext** Transfer Protocol Secure
- **ChEBI –** Chemical Entities of Biological Interest
- **CHEMINF –** Chemical Information Ontology
- **IAO** – Information Artifact Ontology
- **ID –** identifier
- **IDSM –** Integrated Database of Small Molecules
- **IRI –** Internationalized Resource Identifier
- **JSON** – JavaScript Object Notation
- **JSON-LD –** JavaScript Object Notation for Linked Data
- **LODD –** Linked Open Drug Data
- **mmdbint –** MolMeDB interactions
- **mmdbref –** MolMeDB references
- **mmdbsub** – MolMeDB substances
- **mmdbtra** – MolMeDB transporters

- **mmdbvoc** – MolMeDB vocabulary
- **MolMeDB** – Molecules on Membranes Database
- **MVC** – Model-View-Controller
- **NCBO –** National Center for Biomedical Ontology
- **NCIT –** National Cancer Institute Thesaurus
- **OBO** – Open Biological and Biomedical Ontology
- **OLS –** Ontology Lookup Service
- **Open PHACTS –** Open Pharmacological Concept Triple Store
- **OWL –** Web Ontology Language
- **PATO –** Phenotype And Trait Ontology
- **PDBj** – Protein Data Bank Japan
- **pmid –** PubMed identifier
- **R2RML** – RDB to RDF Mapping Language
- **RDF –** Resource Description Framework
- **RDFS** – RDF Schema
- **REPR** – REPRODUCE-ME Ontology
- **REST –** Representational State Transfer
- **RFC –** Request For COmments
- **RIF –** Rule Interchange Format
- **RSA –** Rivest–Shamir–Adleman šifra
- **SIO** – Semanticscience Integrated Ontology
- **SKOS –** Simple Knowledke Organization System
- **SPARQL –** SPARQL Protocol and RDF Query Language
- **SPOT –** Samples, Phenotypes and Ontologies Team
- **SQL –** Structured Query Language
- **TSL/SSL –** Transport Layer Security/Secure Socket Layer
- **URI** – Uniform Resource Identifier
- **URL** – Uniform Resource Locator
- **W3C –** World Wide Web Consortium
- **XHTML –** Extensible HyperText Markup Language
- **XML –** Extensible Markup Language

10. PŘÍLOHY

V příloze je zip soubor obsahující R2RML mapování z relačního do RDF schématu, 2 Turtle soubory (záznam databází do literatury a připojení komponent z ChEBI k membránám) a slovník MolMeDB. V aktualizovaných verzích jsou tyto soubory přístupné i na URL https://github.com/DominikMartinat/rdf-mmdb.