Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Методологија за развој на апликации базирани на поврзани податоци

Докторска дисертација

Милош Јовановиќ

Ментор:
Проф. д-р Димитар Трајанов

Скопје, 2016

Комисија

Академик проф. д-р Љупчо Коцарев, претседател
Факултет за информатички науки и компјутерско инженерство
Универзитет „Св. Кирил и Методиј“ во Скопје

Проф. д-р Димитар Трајанов, ментор
Факултет за информатички науки и компјутерско инженерство
Универзитет „Св. Кирил и Методиј“ во Скопје

Вонр. проф. д-р Борис Делибашиќ, член
Факултет организационих наука
Универзитет у Београду

Вонр. проф. д-р Андреа Кулаков, член
Факултет за информатички науки и компјутерско инженерство
Универзитет „Св. Кирил и Методиј“ во Скопје

Вонр. проф. д-р Љупчо Антовски, член
Факултет за информатички науки и компјутерско инженерство
Универзитет „Св. Кирил и Методиј“ во Скопје

Методологија за развој на апликации базирани на поврзани податоци

Докторска дисертација

Милош Јовановиќ

Апстракт

Огромниот број податоци достапни преку дистрибуираната инфраструктура на Вебот иницираше развој на техники за нивна репрезентација, складирање и искористување. Една од овие техники е и парадигмата на поврзани податоци, која има за цел да обезбеди унифицирани практики за објавување и контекстно поврзување на податоци на Вебот, со користење на стандардите на Веб Конзорциумот и технологиите на Семантичкиот Веб. Ваквиот пристап овозможува трансформација на Вебот од мрежа на документи, во мрежа на податоци. Со тоа Вебот станува дистрибуирана мрежа за пристап до податоци која може да се користи од страна на софтверски агенти и машини. Поврзаната природа на дистрибуираните податочни множества обезбедува основа за креирање напредни кориснички сценарија за крајните корисници и нивните апликации, сценарија претходно недостапни над изолирани податочни множества. Ова креира можност за генерирање нова бизнис вредност во индустријата.

Прифаќањето на принципите на поврзаните податоци од страна на научната заедница и објавувачите на податоци од индустријата доведе до креирање на т.н. облак на поврзани податоци - широка колекција од меѓусебно поврзани податочни множества, објавени и достапни преку постоечката инфраструктура на Вебот. Искуството при креирањето на овие поврзани податочни множества доведе до развој на неколку методологии за трансформација и објавување на поврзани податоци. Сепак, иако овие методологии го покриваат процесот на моделирање, трансформација / генерирање и објавување на поврзани податоци, тие пропуштаат да го земат предвид повторното искористување на чекорите од животниот циклус на податочното множество, во рамките на даден домен. Ова резултира со одвоени и независни потфати за генерирање поврзани податочни множества во даден домен, кои секогаш поминуваат низ сите чекори од животниот циклус.

Во оваа дисертација, врз база на нашето искуство во генерирање, објавување и искористување поврзани податочни множества во повеќе домени и врз база на постоечките методологии за поврзани податоци, дефинираме нова методологија за поврзани податоци, фокусирана на концептот на повторно искористување. Таа се состои од пет чекори кои ги опфаќаат задачите за проучување на доменот, моделирање на податоците, трансформација на податоците, објавување на податочното множество и негово искористување. Во секој од чекорите, методологијата содржи насоки за објавувачите

на податоци за дефинирање на компоненти кои можат повторно да се искористат, во форма на алатки, шеми и сервиси, за дадениот домен. Со ова, идните објавувачи на поврзани податоци во доменот ќе бидат во можност да ги искористат повторно овие компоненти при изминувањето на чекорите од животниот циклус на податочното множество, што директно влијае на нивната ефикасност и продуктивност. Дополнително, повторното искористување на податочните шеми во даден домен резултира со поврзани податочни множества кои се компатибилни со останатите податочни множества генерирани со истите компоненти, што дополнително ја зголемува вредност на податочните множества.

Ваквиот пристап има за цел да ги охрабри објавувачите на податоци да генерираат висококвалитетни поврзани податочни множества од разни домени, што би водело кон понатамошен раст на бројот на податочни множества во LOD облакот, нивниот квалитет и нивното искористување. Со актуелизирањето на интердисциплинарни научни полиња како што е науката базирана на податоци, креирањето и објавувањето на висококвалитетни поврзани податочни множества на Веб станува уште поважно, поради тоа што на тој начин се формира отворен *податочен простор*, изграден над постоечките Веб стандарди. Таков податочен простор им дава можност на научниците кои работат со податоци да извршуваат податочни анализи над прочистените, структурираните и порамнетите податоци достапни во него, со цел генерирање нови знаења и вредности во рамките на даден домен. Имајќи предвид дека принципите на поврзани податоци можат да се применат и во рамките на затворени околини над податоци кои не се отворени по природа, истите методи и пристапи можат да се употребат и во доменот на компании.

Клучни зборови: Поврзани податоци, наука базирана на податоци, методологија, повторна употреба, методи, алатки, отворени податоци, Семантички Веб.

Ментор: д-р Димитар Трајанов, Редовен професор

Linked Data Application Development Methodology

PhD Thesis

Milos Jovanovik

Abstract

The vast amount of data available over the distributed infrastructure of the Web has initiated the development of techniques for their representation, storage and usage. One of these techniques is the Linked Data paradigm, which aims to provide unified practices for publishing and contextually interlinking data on the Web, by using the World Wide Web Consortium (W3C) standards and the Semantic Web technologies. This approach enables the transformation of the Web from a web of documents, to a web of data. With it, the Web transforms into a distributed network of data which can be used by software agents and machines. The interlinked nature of the distributed datasets enables the creation of advanced use-case scenarios for the end users and their applications, scenarios previously unavailable over isolated data silos. This creates opportunities for generating new business values in the industry.

The adoption of the Linked Data principles by data publishers from the research community and the industry has led to the creation of the Linked Open Data (LOD) Cloud, a vast collection of interlinked data published on and accessible via the existing infrastructure of the Web. The experience in creating these Linked Data datasets has led to the development of a few methodologies for transforming and publishing Linked Data. However, even though these methodologies cover the process of modeling, transforming / generating and publishing Linked Data, they do not consider reuse of the steps from the life-cycle. This results in separate and independent efforts to generate Linked Data within a given domain, which always go through the entire set of life-cycle steps.

In this PhD thesis, based on our experience with generating Linked Data in various domains and based on the existing Linked Data methodologies, we define a new Linked Data methodology with a focus on reuse. It consists of five steps which encompass the tasks of studying the domain, modeling the data, transforming the data, publishing it and exploiting it. In each of the steps, the methodology provides guidance to data publishers on defining reusable components in the form of tools, schemas and services, for the given domain. With this, future Linked Data publishers in the domain would be able to reuse these components to go through the life-cycle steps in a more efficient and productive manner. With the reuse of schemas from the domain, the resulting Linked Data dataset will be compatible and aligned with other datasets generated by reusing the same components, which additionally leverages the value of the datasets.

This approach aims to encourage data publishers to generate high-quality, aligned Linked Data datasets from various domains, leading to further growth of the number of datasets on

the LOD Cloud, their quality and the exploitation scenarios. With the emergence of data-driven scientific fields, such as Data Science, creating and publishing high-quality Linked Data datasets on the Web is becoming even more important, as it provides an open dataspace built on existing Web standards. Such a dataspace enables data scientists to make data analytics over the cleaned, structured and aligned data in it, in order to produce new knowledge and introduce new value in a given domain. As the Linked Data principles are also applicable within closed environments over proprietary data, the same methods and approaches are applicable in the enterprise domain as well.

Keywords: Linked Data, Data Science, Methodology, Reuse, Methods, Tools, Open Data, Semantic Web.

Supervisor: Prof. Dimitar Trajanov, PhD

Благодарност

Патот до оваа дисертација беше исполнет со поддршка, разбирање, инспирација и верба од повеќе луѓе, на кои сум им искрено благодарен.

Сакам да изразам особена благодарност до мојот ментор, проф. д-р Димитар Трајанов, кој со својот несебичен пристап ме водеше изминативе години низ мојот високообразовен процес - од додипломските студии, преку магистерските студии, до докторските студии. Охрабрувајќи ме да прифаќам големи предизвици на секој чекор и насочувајќи ме низ нив, ми овозможи да прераснам во добар истражувач и професионалец во областа.

Сакам да им се заблагодарам и на моите колеги за прекрасната соработка во рамките на секојдневната работа, во текот на докторските студии и во рамките на научноистражувачките проекти на Факултетот: Ристе Стојанов, Владимир Здравески, Александра Богојеска, Игор Мишковски, Сашо Граматиков, Соња Филипоска, Анастас Мишев, Бојан Најденов, Костадин Мишев, Матеј Петров и Ѓорѓи Стрезоски. Дополнително, сакам да им се заблагодарам и на професорите и колегите со кои работевме на истражувања и публикации кои се дел од оваа дисертација: академик проф. д-р Љупчо Коцарев, проф. д-р Коста Митрески, колегите Петар Ристоски, Марјан Георгиев, Симона Мицевска, Анѓела Давиткова, Дамјан Ѓуровски, Горан Петковски, Дамјан Темелковски, Анѓел Ќосевски, Никола Калемџиевски, Христијан Пејчиноски, Кристина Циева, Елена Мишевска, Мартина Јаневска, Александар Карески, Мартин Костовски и Александар Андреевски.

Им се заблагодарувам и на моите родители и мојот брат, кои ми пружаа неизмерна поддршка во текот на студиите и гордо ги поддржуваа моите академски и научноистражувачки амбиции.

Посебна благодарност изразувам кон мојата сопруга Елена, која безрезервно ме поддржуваше низ сите успеси и падови кои работата на една дисертација ги носи. Без нејзината поддршка, оваа дисертација немаше да биде завршена.

Содржина

Листа на слики

Листа на табели

Глава 1

Вовед

Едно особено активно научно поле во последната деценија е полето на управување со податоци: претставување, складирање и пристап. Огромната количина податоци достапни преку Вебот иницираше развој на техники за управување на податоците дистрибуирани преку неговата постоечка инфраструктура. Една од овие техники е и парадигмата на поврзани податоци (Linked Data, анг.), која има за цел да обезбеди унифицирани практики за објавување на меѓусебно поврзани податоци на Вебот, користејќи ги стандардите на Веб Конзорциумот (World Wide Web Consortium - W3C, анг.) и Семантичкиот Веб (Semantic Web, анг.) [93, 92, 122]. Ова овозможува трансформација на класичниот Веб од мрежа на документи во мрежа на податоци, што е во линија со оригиналната идеја на Семантичкиот Веб [86]. Со неговата трансформација во дистрибуирана мрежа за пристап до податоци, Вебот може да се користи од страна на софтверски агенти и машини [184, 143, 142, 192].

Поврзаната природа на овие дистрибуирани податочни множества претставува основа за развој на напредни кориснички сценарија за крајните корисници и нивните апликации, сценарија претходно недостапни над изолираните податочни множества. Со скорешната актуелизација на науката базирана на податоци (Data Science, анг.), постоењето на глобален, отворен податочен простор достапен преку постоечката инфраструктура и стандарди на Вебот, овозможува извршување податочни анализи над поврзаните, структурирани податоци достапни во него, со цел генерирање нови знаења и вредности во рамките на даден домен. Дополнително, ваквата мрежа од податоци на Вебот претставува солидна основа за развој на нови бизнис ориентирани решенија за ИКТ индустријата и за независните развивачи на софтвер. Тие можат да развиваат нови типови на апликации и сервиси за крајните корисници, со што ќе се креира нова бизнис вредност во индустријата и општеството [87, 150, 192, 184, 145].

Прифаќањето на принципите на поврзаните податоци [85] од страна на голем број компании, истражувачки центри и институции од целиот свет, доведе до креирање на глобален податочен простор со меѓусебно поврзани податоци од различни домени: луѓе, компании, книги, публикации, филмови, музика, ТВ и радио програми, информации за гени, протеини, генерички лекови, клинички испитувања, онлајн комуникација и соци-

јални медиуми, статистички и научни податоци, итн. [122]. Оваа мрежа од податоци е наречена облак од поврзани податоци (Linked Open Data (LOD) Cloud, анг.) - широка мрежа од податочни множества објавени и меѓусебно поврзани согласно принципите на поврзани податоци. Податоците од оваа мрежа се достапни преку постоечката инфраструктура на Вебот.

Податоците од LOD облакот можат да се пристапат и пребаруваат со користење на технологиите на Семантичкиот Веб, т.е. со користење на SPARQL прашалниот јазик и концептот на SPARQL здружување на прашања [169]. Ова овозможува пристап до податоци од дистрибуирани податочни извори на Вебот на начин сличен на пристапот до локална база на податоци. LOD облакот овозможува развој на нови апликации и сервиси во конкретни домени или во комбинација од повеќе домени. За разлика од Web 2.0 апликациите кои се развиваат да функционираат со предефинирано множество податочни извори, апликациите кои ги користат податочните множества од LOD облакот можат да пристапуваат до неограничен, глобален податочен простор на униформен начин, со единствено множество стандарди. Со секое ново податочно множество кое е додадено во LOD облакот, генерализираните апликации кои ги користат податоците од облакот можат автоматски да ги искористат и податоците од новото множество.

Принципите на поврзани податоци обезбедуваат генерални насоки за начинот на кој едно податочно множество треба да се форматира и објави на Вебот во формат на поврзани податоци. Но, постојат различни алатки, методи и техники за генерирање и објавување на такви податочни множества, а нивната примена зависи од типот на изворните податоци, нивната природа, како и ред други фактори. Овие методи и техники се обединети во неколку методологии за животниот циклус на поврзаните податоци, кои нудат различен пристап за справувањето со поврзаното податочно множество во конкретни домени и за конкретни намени. Дел од овие методологии се насочени кон поврзани податоци од владини извори, како на пример методологиите [127] и [195]. Методологијата опишана во [81] е специфична за множеството алатки развиени како дел од LOD2 проектот [36]. Една генерална методологија е методологијата од [120]. Во [171] и [198], авторите претставуваат методологии насочени кон поврзани податоци од доменот на телевизија и библиотекарско работење. Во [174] и [147], пак, авторите претставуваат методологија фокусирана на обезбедување повисок квалитет на поврзани податочни множества. Иако најголемиот дел од овие методологии го покриваат процесот на моделирање, трансформација / генерирање и објавување на поврзани податоци, тие пропуштаат да го земат предвид повторното искористување на чекорите од животниот циклус на податочното множество, во рамките на даден домен. Ова резултира со одвоени и независни потфати за генерирање поврзани податочни множества во даден домен, кои секогаш поминуваат низ сите чекори од животниот циклус.

Во оваа дисертација, врз база на нашето искуство во генерирање, објавување и искористување поврзани податочни множества во повеќе домени и врз база на постоечките методологии за поврзани податоци, дефинираме нова методологија за поврзани податоци, фокусирана на концептот на повторно искористување. Таа се состои од пет

чекори кои ги опфаќаат задачите за проучување на доменот, моделирање на податоците, трансформација на податоците, објавување на податочното множество и негово искористување. Во секој од чекорите, методологијата содржи насоки за објавувачите на податоци за дефинирање на компоненти кои можат повторно да се искористат, во форма на алатки, шеми и сервиси, за дадениот домен. Со ова, идните објавувачи на поврзани податоци во доменот ќе бидат во можност да ги искористат повторно овие компоненти при изминувањето на чекорите од животниот циклус на податочното множество, што директно влијае на нивната ефикасност и продуктивност. Дополнително, повторното искористување на податочните шеми во даден домен резултира со поврзани податочни множества кои се компатибилни со останатите податочни множества генерирани со истите компоненти, што дополнително ја зголемува вредност на податочните множества.

Ваквиот пристап има за цел да ги охрабри објавувачите на податоци да генерираат висококвалитетни поврзани податочни множества од разни домени, што би водело кон понатамошен раст на бројот на податочни множества во LOD облакот, нивниот квалитет и нивното искористување. Со актуелизирањето на интердисциплинарни научни полиња како што е науката базирана на податоци, креирањето и објавувањето на висококвалитетни поврзани податочни множества на Веб станува уште поважно, поради тоа што на тој начин се формира отворен *податочен простор*, изграден над постоечките Веб стандарди. Таков податочен простор им дава можност на научниците кои работат со податоци да извршуваат податочни анализи над прочистените, структурираните и порамнетите податоци достапни во него, со цел генерирање нови знаења и вредности во рамките на даден домен. Имајќи предвид дека принципите на поврзани податоци можат да се применат и во рамките на затворени околини над податоци кои не се отворени по природа, истите методи и пристапи можат да се употребат и во доменот на компании.

Дисертацијата е организирана на следниот начин: во Глава 2 е направен преглед на концептот на поврзани податоци и техниките за генерирање и објавување на поврзани податочни множества. Во Глава 3 е претставена темелна анализа на постоечките методологии за поврзани податоци, нивните заеднички карактеристики, нивните специфични карактеристики, предности и недостатоци во однос на фазите од животниот циклус на поврзаните податочни множества. Во Глава 4 е претставена имплементацијата на концептот на поврзани податоци во различни домени: доменот на криминални податоци и безбедност, доменот на јавен транспорт и аерозагадување, финансискиот домен, доменот на мултимедија, доменот на здравство и фармација, како и доменот на гастрономија. Врз основа на анализите и практичните истражувања од Глава 3 и Глава 4, во Глава 5 е претставена новата методологија за поврзани податоци, фокусирана на принципот за повторно искористување на процесот кој го претставува животниот циклус на едно поврзано податочно множество. На крајот, во Глава 6 се резимираат заклучокот и придонесите од докторската дисертација.

Глава 2

Поврзани податоци

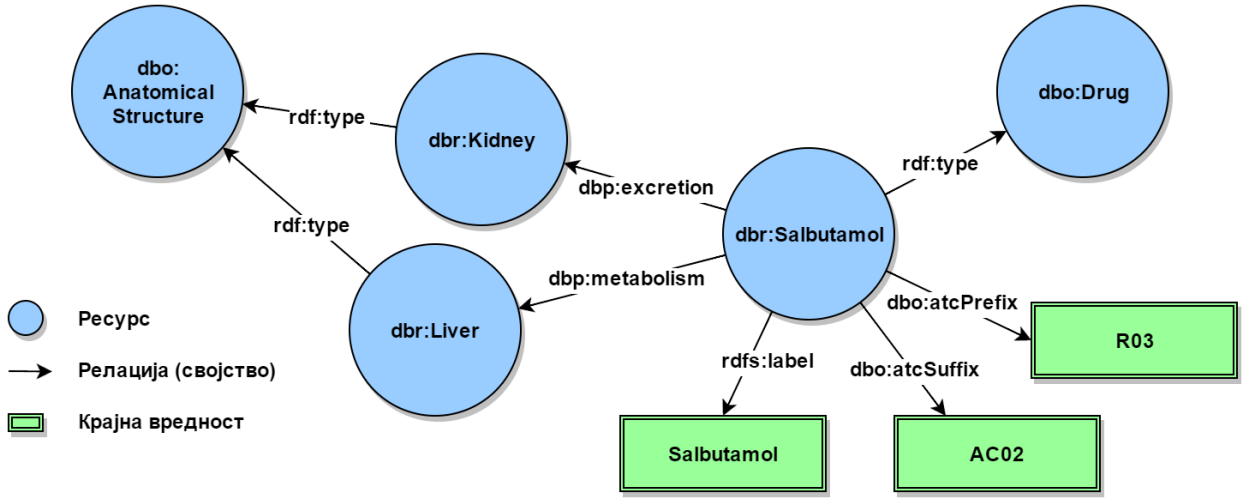
Во 2001 година, во статија објавена во Scientific American, Тим Бернерс-Ли, Џејмс Хендлер и Ора Ласила ја објавуваат својата популарна визија за иднината на Вебот, т.н. Семантички Веб (Semantic Web, анг.) [86]. Според нив, Семантичкиот Веб треба да ги структурира содржините на Вебот во форма во која тие имаат недвосмислено значење, притоа создавајќи околина во која софтверските агенти, кои одат од веб страна на веб страна, ќе можат да извршуваат софистицирани задачи зададени од корисниците. Оттогаш, Веб Конзорциумот заедно со голем број истражувачи од светот работеше на развој на технологии, во вид на стандарди, спецификации, препораки и алатки, кои имаат за цел да го искористат целосниот потенцијал на Вебот и да ја материјализираат неговата трансформација од мрежа на документи во мрежа на податоци.

Како резултат од овие активности, меѓу другото, дефинирани се спецификациите за Resource Description Framework (RDF), RDF Schema (RDFS), Web Ontology Language (OWL) и SPARQL прашалниот јазик. RDF претставува рамка која се користи за претставување на информациите на Вебот [104, 121]. Тој се состои од искази напишани во вид на RDF тројки, составени од подмет (subject, анг.), предикат (predicate, анг.) и предмет (object, анг.), односно:

$$Subject \rightarrow Predicate \rightarrow Object$$

Подметот секогаш претставува ресурс, односно ентитет за кој се однесува исказот. Предикатот е релација односно својство во исказот, додека предметот може да биде или друг ресурс кој е предмет на релацијата во исказот, или пак крајна вредност. Колекцијата од вакви RDF тројки формира т.н. RDF граф (Слика 2.1, Пример 2.1). Притоа, секој од трите елементи од една RDF тројка мора да има уникатен идентификатор (освен крајните вредности), за да се обезбеди недвосмисленост на исказот. Користењето на HTTP URI идентификатори за оваа цел овозможува недвосмисленост на ресурсите и својствата во рамките на Вебот. Според тоа, RDF претставува стандардизиран модел за размена на податоци на Вебот, кој овозможува обединување на податочни множества од различни извори, преку флексибилноста и можноста за децентрализирано проширување на нивната податочна шема. Освен тоа, RDF овозможува и автоматска обработка

на информациите на Вебот од страна на софтверски агенти.



Слика 2.1: Пример RDF граф, составен од 8 RDF тројки.

Пример 2.1

```
prefix dbr: <http://dbpedia.org/resource/>
prefix dbo: <http://dbpedia.org/ontology/>
prefix dbp: <http://dbpedia.org/property/>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

dbr:Salbutamol rdf:type dbo:Drug .
dbr:Salbutamol dbo:atcPrefix ‘‘R03’’ .
dbr:Salbutamol dbo:atcSuffix  ‘‘AC02’’ .
dbr:Salbutamol rdfs:label ‘‘Salbutamol’’ .
dbr:Salbutamol dbp:excertion dbr:Kidney .
dbr:Salbutamol dbp:metabolism dbr:Liver .
dbr:Kidney rdf:type dbo:AnatomicalStructure .
dbr:Liver rdf:type dbo:AnatomicalStructure .
```

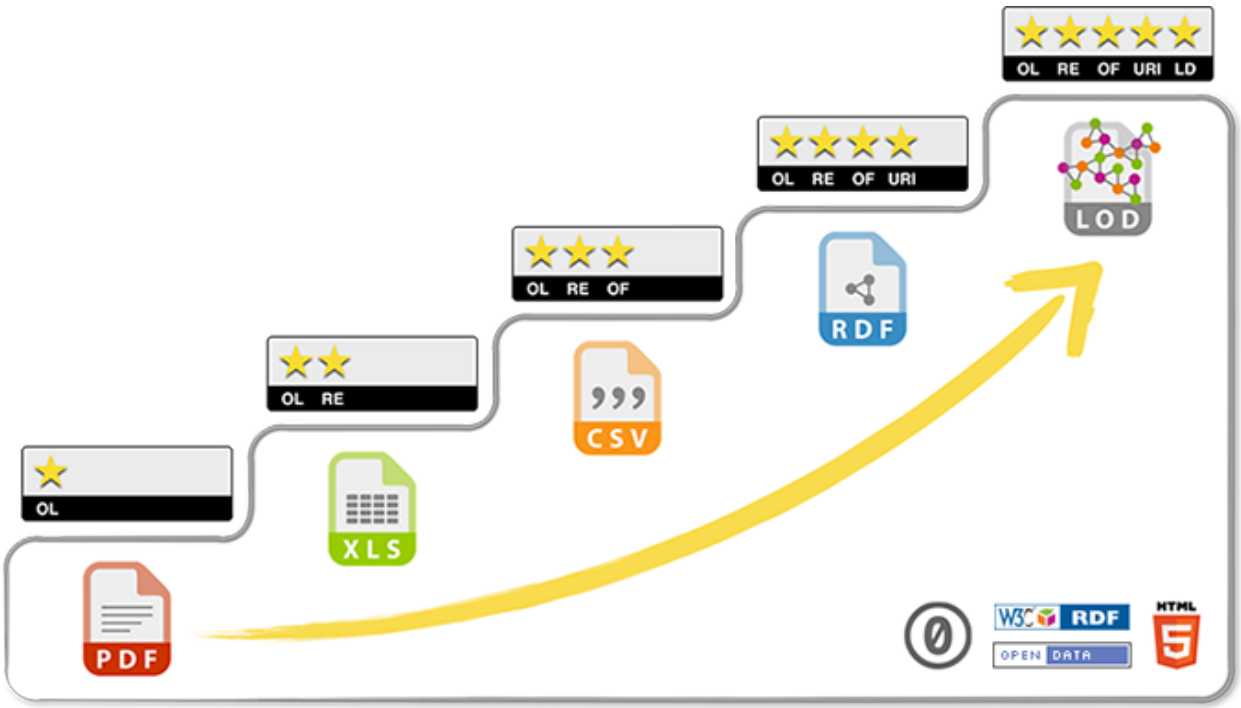
RDFS и OWL се јазици за дефинирање онтологии, односно податочни шеми за податоците опишани во RDF формат [97, 116]. Тие овозможуваат дефинирање и децентрализирано ажурирање на класи, својства и нивни хиерархии, со што се добива клучната флексибилност на RDF како модел за претставување на податоци во рамки на децентрализираната и дистрибуирана околина на Вебот. Како и кај другите јазици кои се базираат на податочни шеми, RDF податоците од дадено податочно множество мора да одговараат на одредена RDFS или OWL онтологија, или на комбинација од нив.

SPARQL (SPARQL Protocol and RDF Query Language) претставува јазик за поставување прашања врз податоци опишани во RDF формат [117]. Јазикот дозволува пра-

шањата да се состојат од шеми за препознавање во RDF тројки, спојувања, разлики или опционални шеми.

Откако воведувањето на овие стандарди доведе до прифаќање на идејата и објавување на податоци во RDF формат како дел од Семантичкиот Веб, од страна на истражувачите и индустријата [122], се појави нов концепт чија идеја се состои во меѓусебно контекстуално поврзување на RDF податочните множества, во т.н. поврзани податочно множества. Со тоа, се иницираше идејата за поврзани податоци (Linked Data, анг.). Овој концепт на поврзани податоци има за цел да ја материјализира основната идеја на Семантичкиот Веб - поврзување на податочни множества достапни на Вебот, преку самата инфраструктура на Вебот, на начини кои овозможуваат непречен пристап помеѓу поврзаните податочни множества и податоците кои се наоѓаат во нив. Со користење на постоечките техники и технологии, како HTTP, RDF, RDFS, OWL, SPARQL, Turtle, JSON, XML, итн., софтверски агенти, апликации и сервиси би можеле да пристапуваат до податоците достапни на Веб како поврзани податоци, овозможувајќи голем број нови и иновативни кориснички сценарија, претходно недостапни над изолирани податочни множества.

За да ја дефинираме попрецизно поврзаноста помеѓу стандардите дефинирани во рамките на Семантичкиот Веб и она што претставува концептот на поврзани податоци, ќе ја погледнеме категоризацијата на податоците на Веб според нивниот квалитет, направена од страна на креаторот на Вебот и на Семантичкиот Веб, Тим Бернерс-Ли [84] (Слика 2.2). Важно е да се напомене дека нивоата на квалитет се кумулативни.



Слика 2.2: Петте нивоа на квалитет на отворени податоци, според Тим Бернерс-Ли.

Основното ниво (1-star) се однесува на јавно достапни податоци на Вебот, без ра-

злика на нивниот формат, како на пример скениран документ, слика, PDF документ, итн. Второто ниво (2-star) се однесува на јавно достапни податоци на Вебот, објавени во структуриран формат, како на пример Microsoft Excel датотека со податоци. Третото ниво (3-star) се однесува на јавно достапни податоци на Вебот, објавени во структуриран формат, но формат кој работи со бесплатен софтвер, како на пример CSV датотека наместо Excel документ. Четвртото ниво (4-star) се *семантички податоци* - структурирани податоци, објавени во RDF формат, кои користат URI идентификатори за ентитетите и релациите помеѓу нив. Петтото ниво, нивото со највисок (5-star) квалитет, се однесува на *поврзани податоци* - податоци од 4-star ниво кое дополнително содржат линкови кон други податочни множества објавени на Вебот.

Според квалитативната категоризација, поврзаните (5-star) податоци претставуваат семантички (4-star) податоци (податоци во RDF формат) кои се експлицитно поврзани со други податочни множества. Сите останати карактеристики им се заеднички, што овозможува имплементација на истите методологии за генерирање, одржување и искористување на податоците. Едно множество од семантички податоци - податоци од четврта категорија - може да се користи и во рамките на мрежата од поврзани податоци: други податочни множества можат да се поврзуваат со него, поради тоа што неговите ентитети користат HTTP URI идентификатори; податочното множество може да се комбинира со други податочни множества, поради тоа што користи онтологии и вокабулари како и поврзаните податочни множества; може да се пребарува со SPARQL прашалниот јазик, бидејќи го користи RDF податочниот модел); итн.

Концептот на поврзани податоци не воведува нови стандарди, туку се потпира на постоечките Веб стандарди и Веб инфраструктура. Што поточно подразбираме кога зборуваме за поврзани податоци, ќе видиме во продолжение.

2.1 Принципи на поврзани податоци

Поимот поврзани податоци се однесува на множество најдобри практики за објавување и поврзување на структурирани податоци на Веб [93][92][122]. Овие најдобри практики се познати и под името Принципи на поврзани податоци [85]. Принципите на поврзани податоци се следните:

1. Користење на URI идентификатори како имиња на нешта;
2. Користење на HTTP URI идентификатори, за да можат луѓето да им пристапат на тие нешта;
3. Кога некој ќе побара одреден URI идентификатор преку HTTP, треба да му се обезбедат корисни информации, со користење на стандарди (RDF, SPARQL);
4. Користење линкови до други URI идентификатори, за корисниците да можат да откријат повеќе нешта;

Првиот принцип препорачува користење URI идентификатори не само за Веб документи, туку и објекти од реалниот свет и апстрактни концепти: луѓе, градови, локации,

возила и слично, но и за настани, релации од типот "е пријател со", итн. Со ова, принципите на Веб идентификација се прошируваат од онлајн светот во реалниот свет и објектите и концептите во него.

Вториот принцип се однесува на користење HTTP URI идентификатори, со цел искористување на добро познатиот механизам за идентификација, пристап и побарување на конкретен документ на Вебот. Користењето на HTTP URI идентификатори значи можност за еднозначно идентификување на објектот или концептот во глобалниот податочен простор на Вебот, побарувањето за негова дефиниција и пристап до самиот објект или концепт.

Третиот принцип се однесува на користењето на стандарди за објавување, поврзување и пребарување на структурирани податоци на Вебот. Принципот налага користење на еден податочниот модел за објавување и поврзување на податоците – Resource Description Framework (RDF) моделот [104, 121]. RDF моделот претставува едноставен граф-базиран податочен модел, кој е дел од технологиите на Семантичкиот Веб. За пребарување, пак, на овие податоци, се налага користењето на уште една од технологиите на Семантичкиот Веб – прашалниот јазик SPARQL [117].

Четвртиот принцип се однесува на користење линкови во податочните множества, со цел поврзување на објекти и концепти помеѓу себе, по принцип сличен на хиперлинковите присутни на документите (веб страните) на стандардниот Веб. Со ова можеме да поврземе два ентитети со одредена релација, а тие не мора да бидат дел од истото податочно множество, туку можат да се објавени на која било веб локација.

Доколку, на пример, податочното множество од Пример 2.1 сакаме да го претвориме во поврзано податочно множество, потребно би било да додадеме релации помеѓу ентитетите од него со ентитети од друго податочно множество, достапно на Веб во RDF формат и формат на поврзани податоци. Тоа би можеле да го направиме со додавање на нови RDF тројки, како на пример:

Пример 2.2

```
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugs/>

dbr:Salbutamol owl:sameAs drugbank:DB01001 .
```

Оваа RDF тројка го поврзува генеричкиот лек од податочното множество со соодветен ентитет од друго податочно множество, достапен преку неговиот HTTP URI идентификатор `http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugs/DB01001`. Достапноста на овој ресурс согласно принципите на поврзани податоци, овозможува пристап до податоците за него, т.е. неговите RDF тројки од неговото податочно множество, преку постоечките W3C стандарди и преку постоечката инфраструктура на Вебот. На тој начин, се обезбедува непречен пристап до податоци управувани на децентрализиран начин, од други органи и тела.

Придржувањето кон овие принципи овозможува пребарување, лоцирање и извлекување податоци и информации од целиот податочен простор на оваа мрежа на податоци,

со користењето на постоечките стандарди и постоечката инфраструктура на Вебот. Така, на пример, една апликација која работи со поврзани податоци може да побара одреден URI идентификатор и да добие RDF податоци кои опишуваа некоја личност, како на пример, омилениот актер, а потоа преку линковите во добиените податоци може да продолжи кон други податочни множества и податоци кои се наоѓаат на други сервери на Вебот и да стигне до податоци, на пример, кои го опишуваат најновиот филм во кој игра конкретниот актер. Ваквиот пристап кон податоците на Вебот овозможува креирање како на специфични, така и на генерални апликации и сервиси кои работат над вакви поврзани податоци, кои можат на корисникот да му понудат повеќе податоци, од разни извори и домени, а со тоа да му понудат и поголема вредност на услугата.

2.2 Мрежа на поврзани податоци

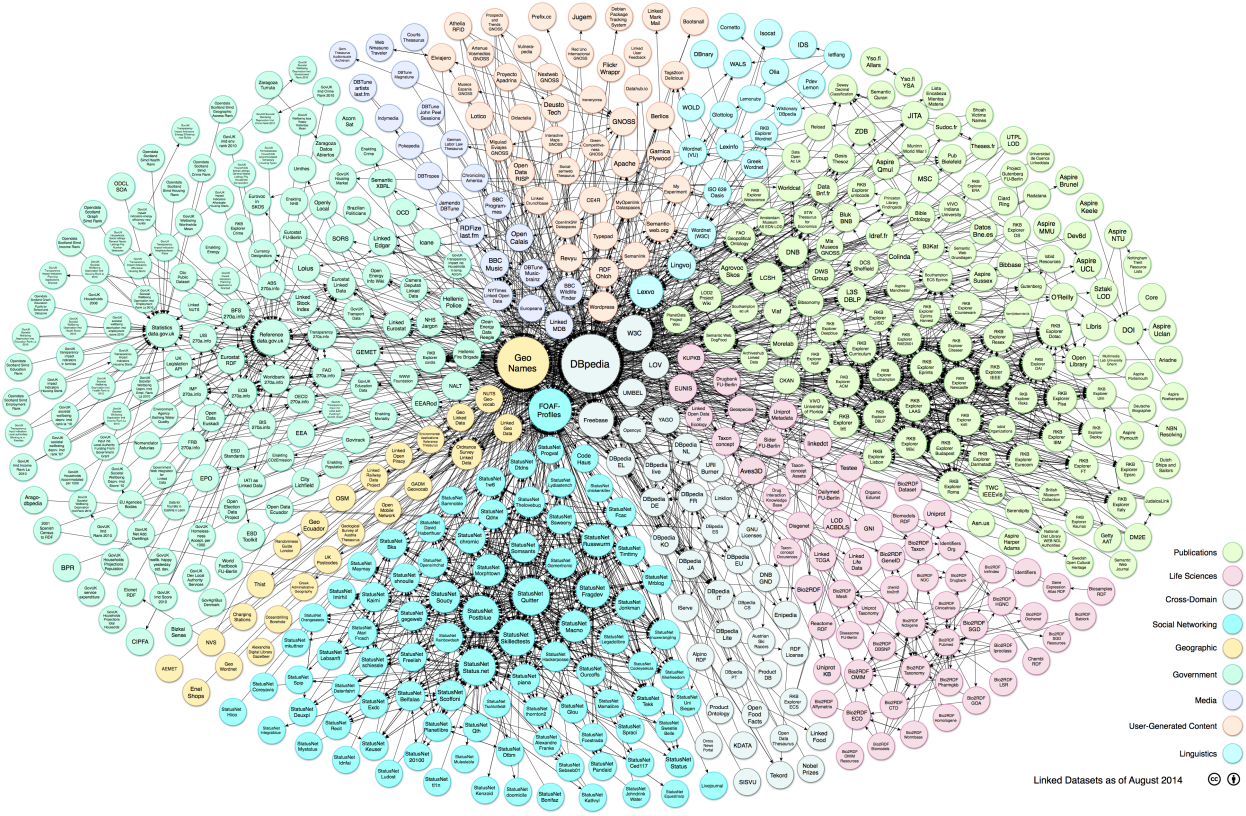
Прифаќањето на принципите на поврзаните податоци од страна на огромен број компании, истражувачки центри и институции од целиот свет, доведе до креирање на реален глобален простор на податоци кој поврзува податоци од различни домени, како што се луѓе, компании, книги, публикации, филмови, музика, ТВ и радио програми, генетски информации, протеини, лекови и клинички тестирања, онлајн комуникации и социјални мрежи, статистички и научни податоци, итн [122].

Ваквата мрежа од податоци е наречена Linked Open Data (LOD) Cloud [30] – *облак* од поврзани и отворени податоци, кој претставува широка мрежа на податочни множества објавени и поврзани согласно принципите на поврзани податоци. Податоците од оваа мрежа се достапни за пристап преку постоечката инфраструктура на Вебот (Слика 2.3).

Овие податоци од LOD облакот можат да се пребаруваат и пронаоѓаат со користење на технологиите на Семантичкиот Веб, т.е. со користење на SPARQL прашалниот јазик и концептот на SPARQL здружување на прашања [169]. Ова обезбедува пристап и пронаоѓање на податоци низ дистрибуирани податочни извори на Вебот, на начин аналоген на пристапувањето до локална база на податоци. LOD облакот обезбедува и нови можности за апликации од еден специфичен домен или пак од комбинација од домени. За разлика од Веб 2.0 mesh-up апликации, кои функционираат со фиксно множество податочни извори, апликациите кои користат поврзани податоци можат да оперираат над скоро неограничен, глобален податочен простор. Ова овозможува ваквите апликации да обезбедат повеќе одговори со секое проширување на LOD облакот.

2.3 Техники за генерирање на поврзани податоци

Принципите на поврзани податоци даваат генерални насоки како одредено структурирано податочно множество да се направи достапно преку Вебот, како дел од мрежата на поврзани податоци. Но, постојат повеќе различни методи и технички пристапи кон генерирањето и објавувањето на ваквите податоци, кои зависат од типот на структурираните податоци, нивната природа, но и низа други фактори. Во продолжение ќе



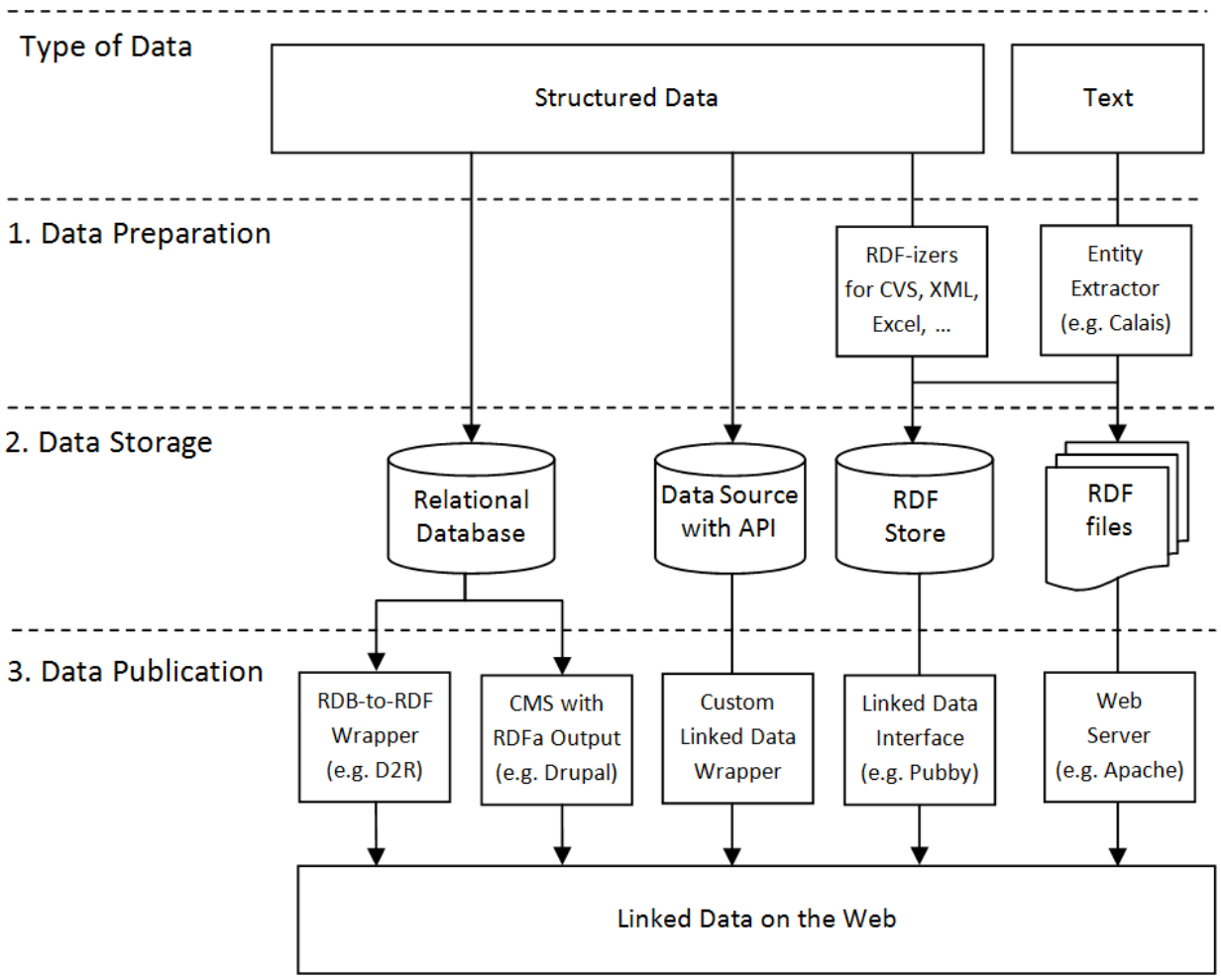
Слика 2.3: Мрежа на поврзани податоци. Состојба од август 2014 година.

направиме преглед и ќе ги споредиме овие различни методи (Слика 2.4), како и нашето досегашно искуство со нив.

2.3.1 Генерирање од релациони бази на податоци

Доколку изворните податоци кои треба да се објават во формат на поврзани податоци се наоѓаат во класична релациона база на податоци, процедурата за генерирање е релативно едноставна. Причината за ова е веќе постоечката шема на податоците, која може да се искористи за директна трансформација на податоците во RDF формат. Во овие случаи, потребно е да се креира мапирање на структурата на релационата база на податоци во RDF граф. За оваа цел се користат разни алатки, меѓу кои најзначајни се D2R Server и Virtuoso Universal Server.

D2R Server [10] овозможува директно мапирање на податоците од релациона база на податоци во RDF граф, преку користење на D2RQ [103] јазикот за мапирање. При мапирањето, како идентификатори на секој од ентитетите од табелите на базата се користат HTTP URI идентификатори. Серверот од една страна нуди веб базиран пристап преку HTML, RDF и SPARQL, а од друга страна комуницира со D2RQ инстанца која со помош на мапирачкиот документ знае како да ги преведе SPARQL барањата добиени од D2R серверот во стандарден SQL прашален јазик и да ги препрати до релационата база на податоци. Резултатот кој се враќа назад до веб корисникот повторно се ма-

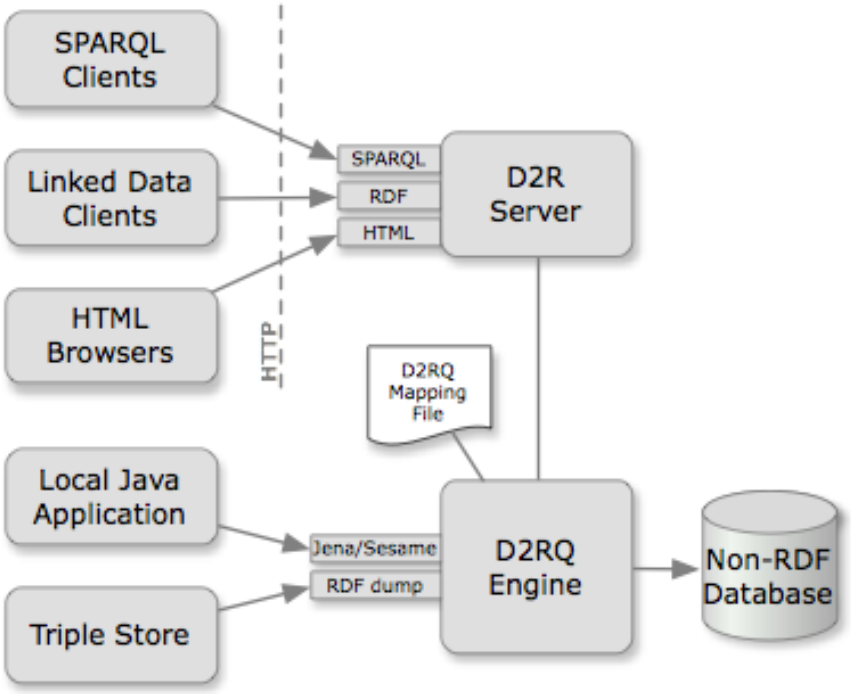


Слика 2.4: Генерален преглед на механизми за генерирање и објавување на поврзани податоци [122].

пира во RDF формат со истите механизми (Слика 2.5). На овој начин, се овозможува пристап во живо до податоците согласно принципите на поврзани податоци, иако тие реално се наоѓаат во класична релациона база на податоци. Ова го прави решението транспарентно за крајните корисници и овозможува работа над реални податоци кои редовно се ажурираат и модифицираат од страна на класичните апликации кои работат со релационата база на податоци.

Ваквиот пристап за објава на податоци ние го искористивме при објавата на податоците за факултетите во рамките на Универзитетот „Св. Кирил и Методиј“ во Скопје [159]. Како извор на податоци беше искористена анонимизирана верзија на релационата база на податоци од системот за е-учење на Факултетот за електротехника и информациски технологии и Факултетот за информатички науки и компјутерско инженерство.

Virtuoso Universal Server [68] нуди слична функционалност како D2R Server од аспект на трансформација на релациона база на податоци во податочно множество согласно принципите на поврзани податоци. Virtuoso користи свој сопствен формат за дефинира-



Слика 2.5: Архитектура на D2R Server.

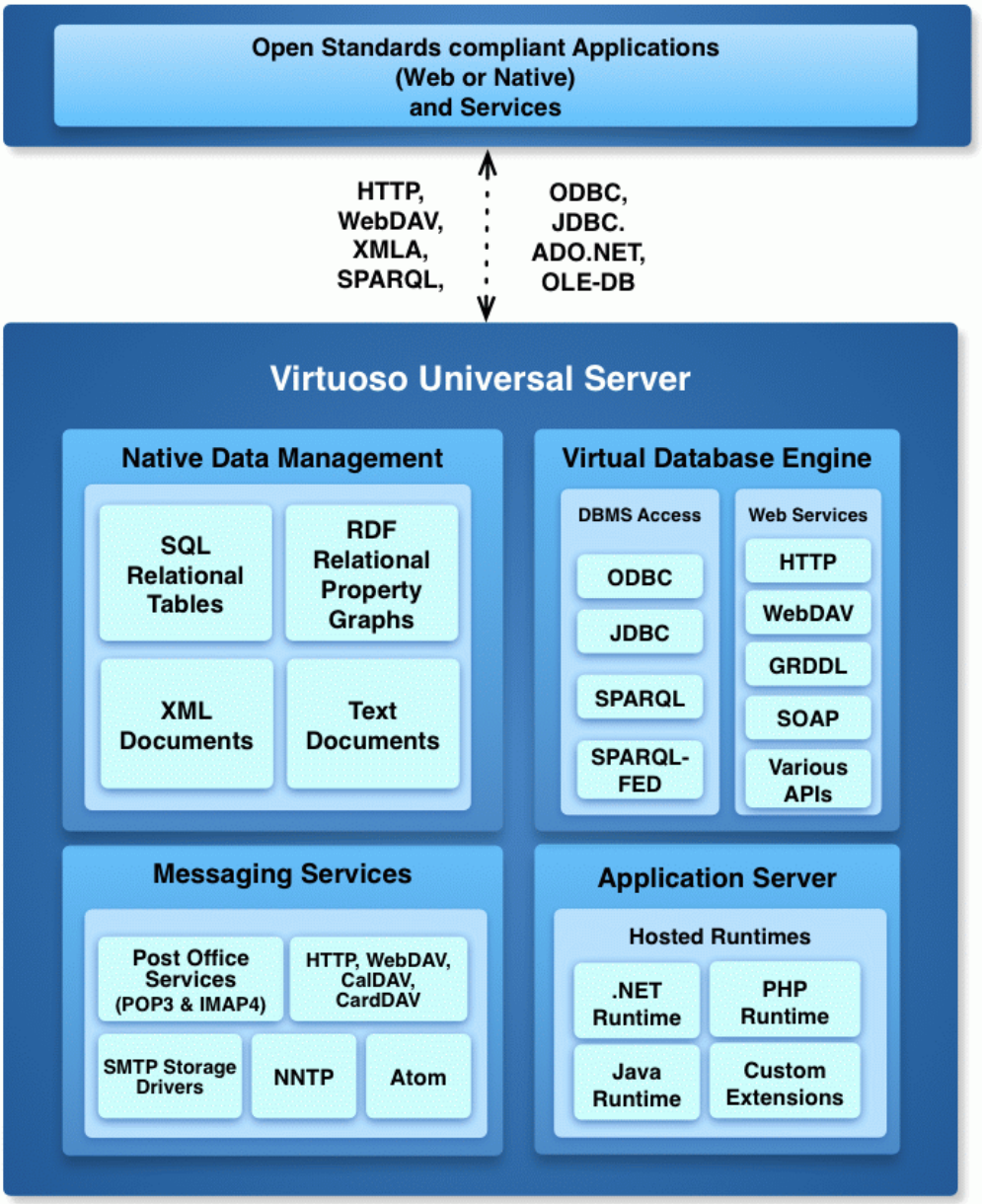
ње на мапирањата, кои потоа се користат за преведување на SPARQL барањата во SQL и обратно (Слика 2.6). Со помош на мапирањето всушност се креираат т.н. RDF View погледи, кои се користат како основа за превод на прашањата. Предноста на Virtuoso е што истиот овој пристап за мапирање на структурирани податоци тој го користи и за податоци кои не се наоѓаат во релациони бази на податоци, туку може да бидат дадени во CSV, TSV или Excel формат, во XML, итн.

2.3.2 Генерирање од статички структурирани податоци

Покрај активни релациони бази на податоци, често се среќаваат сценарија во кои податоците кои треба да се објават во формат на поврзани податоци, се статички. Под овој тип податоци спаѓаат структурирани податоци кои се во CSV формат, во Excel формат, во XML формат или во форма на статички копии од релациони бази на податоци. На сличен принцип како и активните релациони бази на податоци, овие податоци треба да се измапираат и трансформираат во RDF графови кои ќе бидат генерирани во формат на датотеки или пак ќе се вчитаат во RDF складови за објавување.

За оваа цел најкомплетна алатка е Virtuoso Universal Server, кој на сличен начин како и кај активните релациони бази на податоци овозможува креирање на RDF View погледи кои се користат за превод на SPARQL прашања во SQL прашања и кои овозможуваат RDF и SPARQL пристап до поврзаните податоци кои оригинално потекнуваат од CSV, Excel, XML или MDB датотеки.

Покрај Virtuoso, постојат и бројни помали апликации и сервиси кои можат да нап-



Слика 2.6: Архитектура на Virtuoso Universal Server.

раваат статичка трансформација на податоци во RDF. Ваквите сервиси генерално се нарекуваат RDFizers. Екстензивна листа на вакви алатки може да се најде на веб страната [7] на World Wide Web Consortium (W3C).

Овој пристап се покажа како најдобар метод во голем број сценарија во кои имавме работа со статички структурирани податоци. Така, мапирањето на податоците за лекови од Фондот за здравствено осигурување на Македонија (ФЗОМ) ги трансформиравме и објавивме како поврзани податоци со користење на CSV датотеки како извор на податоци и користење на Virtuoso инстанца [66] за нивна трансформација и објавување [140]. За трансформација и објавување на податоците за здравствените институции во Македонија и нивните гео-локации, како и лагер листите на дел од аптеките од ЗАМ

здружението на аптеки, ја искористивме истата методологија [139].

Овој метод за генерирање поврзани податоци го искористивме и при работа со податоци од областа на музиката [141], јавниот транспорт [158], финансиски податоци [163] и податоци за емисии на штетни гасови од превозни средства [162]. Го искористивме и за креирање на поврзаните податочни множества за лекови и рецепти, потоа употребени во нашата анализа за влијанието на светските кујни врз различните категории лекови[137]. Во секој од овие проекти се користеа специфични мапирања и онтологии, кои овозможуваа генерирање на RDF графови кои обезбедуваат уникатни HTTP URI идентификатори за објектите и ентитетите и кои овозможуваа поврзување на податочните множества со податоци од остатокот од глобалната мрежа на податоци, односно LOD облакот.

2.3.3 Генерирање од Веб извори

Како извор на податоци за трансформација во поврзани податоци можат да се користат и разни типови веб базирани извори, како на пример статички или динамички веб страници, социјални медиуми, веб сервиси и API интерфејси, итн. Нивната трансформација во поврзани податоци е малку посложена, генерално поради нивната неструктурираност, односно користењето различна структура и шема во оние случаи кога податоците имаат некаква структура.

За трансформација на слободен текст од веб страници (но, и какви било други текстуални извори) се користат сервиси меѓу кои најпознатите се OpenCalais [42], Alchemy API [1], Dandelion API [11], DBpedia Spotlight [15] и многу други. Резултатот од овие трансформации се RDF графови кои можат да се објават како RDF датотеки или да се вчитаат во RDF слад и да се објават на Веб преку SPARQL пристапна точка и/или Facet прелистувач. Методологијата на користење на овие сервиси за генерирање RDF податоци од слободен текст ја користиме како дел од Semantic Sky проектот, кој претставува платформа за интегрирање на различни типови сервиси и се базира на технологиите на Семантичкиот Веб [192].

Интегриран дел од Virtuoso Universal Server е т.н. Sponger Middleware [67], софтверска компонента која овозможува трансформација на различни типови податоци од Веб извори во RDF графови кои директно се зачувуваат во RDF складот на самата Virtuoso инстанца. Virtuoso Sponger компонентата дозволува развој на т.н. Sponger Cartridge софтверски компоненти, кои се задолжени за трансформација на податоци од специфични извори во соодветни RDF граф репрезентации. Како дел од нашата досегашна работа во полето на поврзани податоци, имаме работено на развој на повеќе Sponger Cartridge компоненти за трансформација на: податоци од каталозите за отворени податоци базирани на CKAN платформата [6], CSV податоци (локали, ски центри, хотели и сл.) од општините Тренто и Тоскана во Италија, GTFS [22] податоци за градски транспорт во Тренто, Италија, XML податоци за временска прогноза, итн. Користењето на ваквите трансформатори овозможува автоматско генерирање на поврзани податоци до RDF формат, секогаш кога низ системот ќе се пуштат податоци во некој од наведените

формати. Ова дозволува автоматско проширување на податочното множество објавено преку соодветната Virtuoso инстанца.

2.4 Техники за објавување на поврзани податоци

Множествата од поврзани податоци генерирани со некој од методите презентирани во претходниот дел, потребно е да бидат објавени на Веб и достапни преку неговата постоечка инфраструктура, преку HTTP URI идентификатори и HTTP URL локатори. Во продолжение ќе направиме преглед на механизмите за објава на поврзани податоци.

2.4.1 Објавување на статички RDF датотеки

Наједноставниот метод за објава на поврзани податоци е публикација на RDF датотека која ги содржи, на одредена Веб локација. Со оглед на тоа што Вебот претставува мрежа од документи, оваа RDF датотека ќе биде достапна преку стандардниот HTTP URL пристап, па со тоа и самите поврзани податоци ќе можат да се добијат кога ќе бидат побарани од краен корисник или негова апликација која работи со ваков тип на податоци. За искористување на овој метод за објава, доволно е објавувачот да располага со приватен или јавен веб сервер, преку кој ќе може да додели уникатно HTTP URL на својата RDF датотека и таа да стане дел од глобалниот податочен простор на поврзани податоци.

2.4.2 Објавување на вгнездена RDF содржина во HTML датотеки

Претходниот пристап го има недостатокот на редовно ажурирање на објавените поврзани податоци во RDF датотеката. Еден начин да се избегне ова е поврзаните податоци во RDF формат да се додадат динамички, како вгнездена содржина на HTML веб страници. За таа цел се користат стандардизираните RDFa [74] и Microdata [123] нотации, кои овозможуваат репрезентација на RDF графови во рамки на HTML страни. Генерирањето на ваквата RDF содржина во овој пристап е на страната на веб платформата која ја користи самиот веб сајт. Поддршка за ваквиот метод на објава имаат голем број платформи, од кои најзначајни се Drupal [17] (со директна поддршка) и WordPress [70] (со поддршка преку додатоци).

2.4.3 Објавување директно од релациони бази на податоци

Методот на објавување поврзани податоци директно од релациони бази на податоци е пристапот кој го користат D2R Server и Virtuoso платформите. Како што веќе објаснивме, покрај трансформацијата на податоците во поврзани податоци репрезентирани како RDF граф, двете платформи обезбедуваат веб интерфејси за преглед на податоците, како и SPARQL пристапни точки за директно или сервис-базирано пребарување низ

поврзаните податоци. Овој пристап го искористивме за објава на поврзаните податоци од Универзитетот [159].

2.4.4 Објавување директно од RDF складови

Virtuoso платформата, како што веќе објаснивме, покрај работа со релациони бази на податоци, овозможува работа и со CSV, TSV, XML, Excel, MDB и други извори на податоци. При трансформација на податоците во поврзани податоци, изворните податоци се зачувуваат во Virtuoso инстанцата или во класични релациони табели над кои се креираат RDF View погледи, или пак се трансформираат во RDF графови. Оваа платформа овозможува автоматско креирање на веб интерфејс за преглед на овие податоци, како и SPARQL пристапна точка за манипулација со податоците, аналогно како кај податоците од претходната точка. Овој метод на објава го искористивме за објава на податоците од неколкуте различни домени во кои работевме [140, 139, 141, 158, 163, 162, 138]. Овие податочни множества со поврзани податоци се достапни преку јавна Virtuoso инстанца [66].

Глава 3

Методологии за поврзани податоци

Специфичноста на доменот на поврзани податоци во однос на останатите типови на структурирани и полуструктурирани податоци, ја потенцираше потребата од дефинирање на нов пристап во управувањето со податоци од овој тип. Како резултат на тоа, во доменот на поврзани податоци се дефинирани неколку методологии за животниот циклус на поврзаните податочни множества, дел фокусирани на специфичен тип податоци, а дел општи. Со цел да ги идентификуваме специфичностите, предностите и недостатоците на постоечките методологии за поврзани податоци, во оваа Глава ќе направиме нивна детална анализа.

Актуелизирањето на прашањето за отворено општество и отворена влада кај голем број влади на глобално ниво, со цел зголемена одговорност и транспарентност [46], Веб Конзорциумот (WWW Consortium - W3C, анг.) и неговата работна група 'W3C Government Linked Data Working Group' креираа официјални насоки за објава и пристап до отворени владини податоци со користење на принципите на поврзани податоци [126]. Работната група потенцира дека користењето на принципите на поврзани податоци во доменот на отворени владини податоци одговара на голем број од потребите, од аспект на објавување, дисеминација, поврзување и повторно искористување.

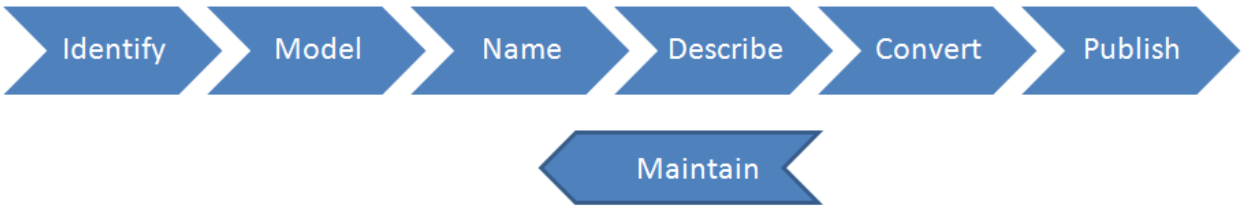
Со цел да ги подготват владините институции, но и останатите заинтересирани страни, работната група посочува три методологии, односно модели на животниот циклус на поврзани владини податоци, кои ги смета за најзначајни. Овие методологии имаат голем дел заеднички активности, но користат различно разграничување помеѓу нив. Тие тврдат дека ниту една не е супериорна над останатите, туку едноставно имаат различен пристап во идентификувањето на чекорите во процесот на работа со поврзани податоци.

Овие три методологии за поврзани податоци се: (а) методологијата на Hyland et al., (б) методологијата на Hausenblas et al. и (в) методологијата на Villazón-Terrazas et al. Во продолжение ќе направиме подетален преглед на секоја од нив, со цел да ги идентификуваме заедничките елементи, но и разликите и специфичностите кои постојат. Во анализата ќе ја вклучиме и методологијата развиена како дел од LOD2 проектот, во кој учествуваа и дел од авторите на трите наведени методологии.

3.1 Методологија на Hyland et al.

Hyland et al. [127] дефинираат методологија која се состои од шест чекори за моделирање, креирање, објавување и дисеминација на поврзани владини податоци. И покрај тоа што оригинално нивната намена е за владини податоци, доменот на податоците не влијае значително во примената на генералните чекори од нивната методологија. Таа се базира на спецификациите и најдобрите пракси дефинирани од Веб Конзорциумот (W3C) и се состои од следниве чекори (Слика 3.1):

1. Идентификација
2. Моделирање
3. Именување
4. Опишување
5. Трансформација
6. Објавување
7. Одржување



Слика 3.1: Методологијата на Hyland et al.

3.1.1 Чекор 1: Моделирање на податоците

Во [127], авторите нивната методологија ја нарекуваат "рецепт" во кој првата поголема фаза е моделирањето на податоците. Таа вклучува идентификација, моделирање, именување и проверка. Експертите за моделирање на податоци во доменот на поврзани податоци генерално креираат модели кои не зависат од контекстот на нивната подоцнежна употреба во апликации и сервиси, за разлика од традиционалното моделирање на (бази на) податоци. Ваквиот пристап овозможува повторно искористување и полесно спојување на податоците. Според [127], овој процес на моделирање на поврзани податоци одзема од неколку часа за поедноставни податочни множества, до неколку недели за податочни множества за кои е потребна соработка помеѓу експерти од доменот и експерти за поврзани податоци. Ова е значително помало време во споредба со бројот на месеци, а понекогаш и години, потребни за организација и одржување на податоците во поголеми институции.

Овој процес на моделирање има за цел да ги отвори податоците кон полесна интеграција со други податоци внатре во организацијата или надвор, преку Вебот.

При моделирање на податоци кои иницијално се сместени во релациони бази на податоци, чекорите кои треба да се преземат се: (а) идентификација, (б) моделирање, (в) именување и (г) проверка.

(а) При идентификација, потребно е да се обезбеди копија од логичкиот и физичкиот модел на базата на податоци, да се екстрахираат или реплицираат податоците и да се идентификуваат објекти од светот кои се од интерес во доменот - луѓе, места, предмети, настани, итн.

(б) При моделирање, пак, потребно е да се скицираат објектите и да се поврзат со линии објектите кои имаат некаква логичка поврзаност во доменот и да се направи истражување како претходно биле опишувани податоци од ист или сличен домен, со цел повторно искористување на заеднички вокабулар што понатаму овозможува поедноставно поврзување и повторно искористување. Понатаму, потребно е да се отстранат евентуални дупликати и да се (де)нормализираат податоците, да се користи "здрав разум" во одлуката да се постави или не врска помеѓу два објекти, како и да не се размислува веднаш за конкретни сценарија на употреба на податоците.

(в) При именување, потребно е да се користат URI идентификатори за именување на објектите од претходниот чекор, во согласност со принципите на поврзани податоци, но и да се размисли за евентуални промени кои би можеле да се случат во податоците со тек на време.

(г) Последен дел од овој чекор е проверка на одлуките од претходните точки со експерт од доменот кој е запознаен со податоците, со цел наоѓање грешки и корекција на моделирањето.

Hyland et al. потенцираат две особено важни работи во овој процес: итеративен и колаборативен пристап во идентификацијата на објектите и нивната меѓусебна поврзаност и поврзаност со објекти од светот, како и целосно отфрлање на прашањата поврзани со потенцијалната употреба на податоците во иднина. Првото овозможува креирање на покомплетен модел, додека второто овозможува креирање на вистински модел на поврзани податоци - модел подготвен за поврзување со други податочни множества и домени, над кој корисниците ќе можат да дефинираат свои кориснички сценарија.

3.1.2 Чекор 2: Именување на нештата со URI идентификатори

Следниот чекор кај Hyland et al. е именување на нештата идентификувани во претходниот чекор. За таа цел, следејќи ги принципите на поврзани податоци, потребно е да се користат URI идентификатори. Со оглед на тоа што URI идентификаторите ја имаат централната техничка улога во доменот на поврзани податоци, нивниот формат е исклучително важен. Според нив, овој чекор не е воопшто едноставен. Поради тоа, во [127] авторите наведуваат неколку дополнителни извори каде може да се добијат совети околу Веб доменот на URI идентификаторите, структурата на патеката, начините за справување со промени во податоците, како и опслужување на формати наменети за луѓето и машините. Како сублимат, нивните препораки за дефинирање URI идентификатори се дадени во продолжение.

Препорака 1: Користење на HTTP URI идентификатори, со што се овозможува употреба на постоечките Веб стандарди за идентификација.

Препорака 2: Користење на "чисти" и стабилни URI идентификатори, односно идентификатори кои не се однесуваат на имплементацијата, туку се погенерални, поопшти.

Препорака 3: Користење на веб домен кој е под наша контрола, со што се обезбедува доверба дека идентификаторите ќе се одржуваат.

Препорака 4: Користење природни клучеви; наместо шифри како идентификатори, во URI идентификаторите да се користат природни имиња на нештата, кои можат да се комбинираат во директориуми од URI патеката. На пример, користење `http://.../drugs/Palifermin`, наместо `http://.../3f4sa45`.

Препорака 5: Користење неутрални URI идентификатори со кои ќе се избегне наведувањето на конкретната верзија на податокот или технологијата во самиот идентификатор. На пример, идентификатор `http://.../drugs/Palifermin` може да редиректира кон конкретна (најнова) верзија на информациите за лекот, наместо да се користат идентификатори како `http://.../drugs/Palifermin-v1-03`, `http://.../drugs/Palifermin-v1-04`, `http://.../drugs-v2/Palifermin`, и слично.

Препорака 6: Внимателно користење на фрагмент идентификатори; идентификувањето на нешта со користење на знакот # на крајот од URI идентификаторот овозможува идентификација на клиентска страна, поради тоа што делот од URI идентификаторот по знакот # не се испраќа до серверот. Како последица, најчесто целиот документ во кој се наоѓа идентификуваниот ентитет се презема на клиентска страна. Ваквиот пристап е добар за идентификација на концепти од онтологии и вокабулари, бидејќи се релативно помали (на пример: `http://purl.org/net/po#Song`). При идентификација, пак, на ентитети од бази на знаење, кои се значително поголеми од онтологиите, подобро е користењето на знакот / за недвосмислена идентификација на ентитетот (пример: `http://purl.org/net/lmd/data/ghost-ella-henderson`).

Препорака 7: Користење датум во URI идентификатори; одредени типови податоци, како што се статистички податоци, регулаторни податоци и податоци со спецификации, бараат верзионирање. Согласно практиките на W3C, одредени URI идентификатори можат да користат и датум за да ја означат верзијата на податоците кои ги идентификуваат.

3.1.3 Чекор 3: Искористување на постоечки вокабулари

Во овој чекор, авторите даваат предлог за искористување на постоечки онтологии и вокабулари за опис на ентитетите од податочното множество. Со користење на класи и релации од постоечки онтологии, податоците стануваат разбирливи во доменот на поврзани податоци на Вебот. Тековно постојат бројни онтологии кои се специјализирани за различни домени, а дел се и од поопшт карактер. Авторите препорачуваат искористување на FOAF, DOAP, Dublin Core, VoID, vCard, WGS84, BIBO, CC, GeoNames, GoodRelations, итн. секаде и секогаш кога тоа е возможно.

Доколку доменот на податоците е специфичен и не постои онтологија која може да

ги опише сите ентитети, потребно да се креира сопствена онтологија или да се прошири постоечка онтологија. За таа цел, авторите препорачуваат користење на RDFS и OWL јазиците. Доколку е потребно да се дефинира еквиваленција помеѓу класи од новата онтологија и класи од постоечки онтологии, се препорачува користење на SKOS вокабуларот.

3.1.4 Чекор 4: Објавување опис наменет за луѓе и за машини

Отворениот пристап до поврзани податоци објавени на Вебот бара податоците да бидат јасни самите по себе (self-describing, анг.), односно да носат информација за типот на секој податок. Ова се постигнува со користење на онтологии и вокабулари за дефинирање на типот на ентитетите и нивните вредности. Ваквото моделирање надвор од конкретен апликациски контекст бара валидацијата да биде изведена при конкретната примена, во нејзиниот контекст.

Препораките во овој чекор се описот на податочното множество да биде достапен истовремено и во форма разбирлива за човек (текстуален опис) и во форма разбирлива за машините (RDF, RDFS, OWL, VoID). Ова овозможува полесна детекција на податочното множество кога е јавно достапно преку Веб, со цел искористување во конкретно корисничко сценарио, преку апликација, сервис, итн.

3.1.5 Чекор 5: Трансформација на податоците во RDF

По извршеното моделирање, следува трансформација на податоците во соодветна репрезентација за поврзани податоци. Постоењето на повеќе RDF синтакси бара избор на соодветен формат за конкретниот контекст - едни RDF синтакси овозможуваат подобра компатибилност со други технологии, додека други се побрзи за трансфер преку Веб и полесно читливи за луѓето. Препорачана практика во овој чекор е валидација на подмножество од податоците по нивната трансформација, за што постојат повеќе алатки. Тоа овозможува рана детекција на грешки во трансформираните податоци, пред тие да бидат вчитани во некој RDF репозиториум.

Трансформацијата на податоци може да биде во една од трите категории: (а) автоматска конверзија или триплификација (triplification, анг.), (б) конверзија со парцијално скриптирање, или (в) моделирање од експерти во областа и скриптирана конверзија. Првата од категориите е применлива во случаи кога количеството податоци кои треба да се трансформира е многу големо, како на пример при работа со податоци од сензори. Но, недостатокот на моделирање на податоците во овој случај може да резултира во RDF податоци со низок квалитет според нивоата на квалитет на отворени и поврзани податоци [84] (Слика 2.2), што влијае директно врз можноста за нивно повторно искористување. Според Hyland et al., доколку поврзаните податоци не можат да се искористат од други луѓе, тогаш воопшто нема потреба да се трансформираат.

Подобра опција претставува вклучувањето експерти од доменот во процесот, за да направат моделирање на податоците. Овој процес е ист како стандардниот процес на

моделирање кој постои веќе со децении, со единствената разлика што идентификаторите за нештата се URI идентификатори и шемата на податоците е јавно достапна во формат разбирлив за луѓето и формат разбирлив за машините. Според авторите, процесот на моделирање трае од две до четири недели, при што се моделираат субјектите, предикатите и објектите, се идентификуваат вокабуларите и онтологиите кои можат да се искористат за анотација и по потреба се дефинираат сопствени вокабулари и онтологии. Потребно е процесот на моделирање да се документира, со цел да се објави заедно со податоците и нивната шема - ова овозможува подобри услови за повторно искористување. Ваквиот документ не мора да биде комплексен и целосно технички.

Трансформираните податоци треба да се објават и да станат достапни до публиката. За таа цел, Hyland et al. препорачуваат користење на услуги од компании кои обезбедуваат одржување на веб страни. Изборот на соодветна компанија за соработка може да биде комплексен. Одлуката за тоа која компанија ќе се користи треба да биде водена од серија прашања [127], која се фокусира на подготвеноста на понудената платформа да работи со поврзани податоци и во согласност со W3C стандардите, отвореноста и генералноста на кодот и алатките, претходните проекти на компанијата од ист домен, итн. Покрај работа со податоци од највисок квалитет - податоци со 5 ѕвезди - важна е и работата со правилен MIME податочен тип. Ова овозможува HTTP content negotiation и дозволува правилен пристап до податоци наменети и за машини и за луѓе.

3.1.6 Чекор 6: Јавно известување за новокреираното поврзано податочно множество

Објавувањето и јавното известување за новокреираното поврзано податочно множество треба да биде во линија со полисите за податоци кои веќе постојат во институцијата. Овие полиси треба да бидат дел во објавата и да вклучуваат информации за приватност, квалитет, информации за податоци од други извори, цитирања, референци, итн. Доколку поврзаното податочно множество е достапно преку SPARQL endpoint, потребно е да се имплементираат мерки за контрола на пристапот. Тие можат да вклучуваат: пристап само за авторизирани корисници, пристап до ограничено подмножество од податоците, ограничување на количеството податоци кои можат да се добијат при едно пребарување и забрана за модификации на податоците преку SPARQL барања. Од аспект на перформансите на системот кој ги чува и служи податоците, потребно е да се внимава на потребите на податоците и корисничките сценарија и можностите на платформата. За таа цел, Hyland et al. препорачуваат консултација со експерти од доменот на поврзани податоци и SPARQL, со цел подобро моделирање на решението од аспект на користен софтвер и хардвер.

Покрај тоа што поврзаното податочно множество треба да ги задоволува принципите на поврзани податоци, доколку сакаме да стане дел од облакот на поврзани податоци тогаш тоа ќе треба да ги задоволи и неговите критериуми [32]. Овие критериуми вклучуваат ограничувања според кои податочното множество треба да брои минимум 1.000

RDF тројки и мора да постојат најмалку 50 линкови кон или од други податочни множества кои се веќе дел од LOD облакот.

Објавата на ново поврзано податочно множество може да биде придружена со соопштение за јавноста. Дополнително, работната група на W3C за поврзани владини податоци нуди серија добри практики [24] кои вклучуваат објавување на податочното множество на Datahub податочниот портал [12], на Swoogle [59], Sindice [56] и на веб страната на заедницата која работи со поврзани податоци [29].

3.1.7 Дополнителни совети

Социјална одговорност на објавувачот

Луѓето, организациите и компаниите кои објавуваат поврзани податочни множества треба да бидат свесни за социјалната одговорност која доаѓа со објавувањето на податоците. Тие треба одговорно да ги одржуваат податоците, да ги ажурираат редовно, да се грижат за нивната точност и да ги отстрануваат пријавените проблеми. Доколку повторното искористување на податоците е приоритет, тогаш следењето на најдобрите практики при моделирање и генерирање, внимателното креирање на URI стратегија и објавувањето VoID опис стануваат уште позначајни.

Објавувачот мора да се погрижи и за достапноста на податоците после извесен период. Ненамерното или намерното отстранување на податоци или податочни множества од Вебот може да предизвика прекин или нарушување на работата на апликации од трета страна, за кои можеби објавувачот воопшто не знае дека постојат. Поради тоа, грижата за достапноста на податочните множества претставува основа на непишаниот социјален договор помеѓу објавувачот и корисниците.

Лиценцирање

Hyland et al. препорачуваат експлицитно користење на лиценци на секое објавено податочно множество. Со оглед на тоа што изворот на податоците може да биде различен, тие може да подлежат на различни законски регулативи кои повлекуваат користење на соодветни лиценци. Доменот и природата на податоците исто така влијае на изборот на соодветна лиценца за податочното множество. Користењето на лиценци е особено важно при објавувањето на владини поврзани податоци.

3.2 Методологија на Hausenblas et al.

Според Hausenblas et al. [120], пак, класичните пристапи за управување со податоци - кои претпоставуваат контрола над податоците, нивната шема и нивното креирање - не можат да се применат во доменот на Вебот, поради неговата отворена и децентрализирана природа. Тие сметаат дека екосистемот на поврзани податоци претставува отворена платформа за поддршка на податочни простори (dataspaces, анг.), базирана на стандар-

ди. Нивната методологија се состои од шест чекори (Слика 3.2), добиени преку нивното долгогодишно искуство со објавување и искористување на поврзани податоци:

1. Познавање на податоците
2. Моделирање
3. Објавување
4. Лоцирање
5. Интеграција
6. Кориснички сценарија

Првиот и последниот чекор претставуваат помошни чекори, додека чекорите 2 - 5 го сочинуваат јадрото на нивната методологија.

Слика 3.2: Методологијата на Hausenblas et al.

3.2.1 Чекор 1: Познавање на податоците

Првиот чекор од методологијата се однесува на познавањето на тековната состојба на отворени и поврзани податоци од страна на лицата и институциите заинтересирани за објавување поврзани податоци. Како значајни извори за тековните случувања, авторите ги наведуваат: облакот на поврзани податоци [30], рејтинг системот на Тим Бернерс-Ли за отворени податоци [84] (Слика 2.2) и разни веб локации за размена на искуства при работа со отворени и поврзани податоци (на пример: порталот за отворени податоци на Open Knowledge Ireland фондацијата [47]).

3.2.2 Чекор 2: Моделирање

Следниот чекор, прв од јадрото на методологијата, се однесува на моделирањето на податоците. Тоа подразбира дефинирање на репрезентацијата на ентитетите, класите, својствата и релациите помеѓу ентитетите во доменот. За таа цел, тие го препорачуваат Neologism едиторот за креирање онтологии [83] и неговата инстанца во DERI истражувачкиот центар [16], која овозможува пребарување низ објавените онтологии и наоѓање на најсоодветната за доменот од интерес.

Како важен вокабулар од високо ниво, авторите го издвојуваат Schema.org [52], вокабулар развиен од страна на Google, Yahoo и Microsoft, кој овозможува подетално означување на содржината на веб страните. Целта на развивачите е да се подобри контекстното пребарување кое тие го нудат, но вокабуларот може да се користи и за анотација при креирање поврзани податочни множества. Високото хиерархиско ниво на вокабуларот,

односно неговиот општ карактер, овозможува широка употреба низ различни домени, како и зголемување на интероперабилноста помеѓу поврзаните податочни множества.

Hausenblas et al. го издвојуваат и проектот SchemaRDFS.org [51], кој се обидува да го доближи Schema.org вокабуларот до креаторите на поврзани податочни множества. Проектот се стреми да ја постигне својата цел преку примери и упатства за објавување и користење поврзани податоци анотирани со Schema.org вокабуларот, креирање и одржување на мапирања од други широко употребувани вокабулари и онтологии кон Schema.org класи, одржување листа на развојни алатки и библиотеки за работа со податоци анотирани со Schema.org вокабуларот, како и одржување на Schema.org вокабуларот во Turtle, RDF/XML и NTriples формат.

3.2.3 Чекор 3: Објавување

Третиот чекор се однесува на RDF трансформација и објавување на моделираните податоци од претходниот чекор. Авторите имаат развиено посебен работен тек на процесот на објавување на податоци, комплетно имплементиран во Google Refine алатката (денеска се нарекува Open Refine [48]), кој се состои од следните активности: (а) вчитување на податоците кои се во формат разбирлив за машините, (б) прочистување на податоците, (в) автоматска корекција на податоците, (г) трансформација во RDF и поврзување со други податочни множества и (д) споделување на новокреираното податочно множество. За да се овозможи овој работен тек во рамките на Google Refine алатката, тимот на авторите развил неколку додатоци за алатката, кои му помагаат на корисникот во секој од овие чекори.

Покрај Google Refine, авторите ги наведуваат и RDB2RDF [106, 80] техниките, кои овозможуваат "превод" на моделот на податоците од релациони бази на податоци во RDF модел. Овие техники, заедно со алатките развиени подоцна, овозможуваат креирање на RDF податоци директно од широко распространетите релациони бази на податоци.

3.2.4 Чекор 4: Лоцирање

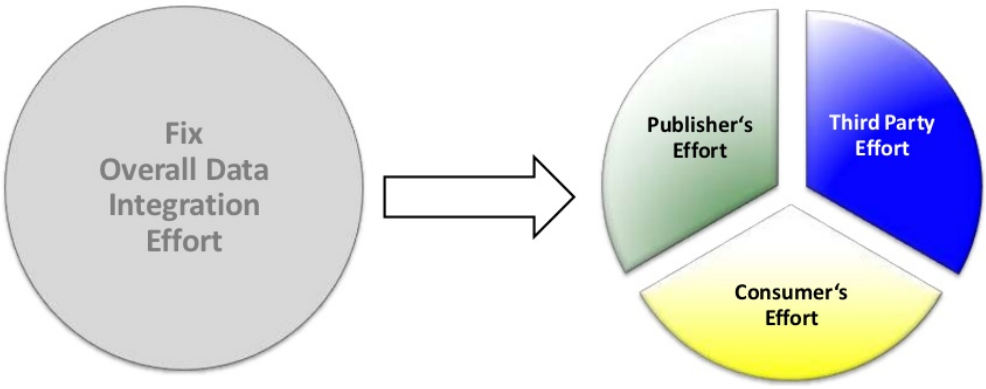
Во четвртиот чекор, Hausenblas et al. се фокусираат на овозможување на корисниците полесно лоцирање и идентификување на податочните множества и ентитетите во нив. Нивниот истражувачки тим е автор на VoID (Vocabulary of Interlinked Datasets) вокабуларот [77, 76], кој е наменет специјално за овој чекор од методологијата. VoID вокабуларот овозможува креирање метаподатоци за податочно множество од поврзани податоци, нудејќи својства за опишување на генерални податоци за множеството (на пример: креатор, објавувач, лиценца), податоци за пристап до множеството (на пример: SPARQL endpoint), структурни податоци за логичката организација на множеството, како и својства за опис на т.н. множества од линкови (linksets, анг.) кои носат информација за поврзаноста на податочното множество со други податочни множества.

3.2.5 Чекор 5: Интеграција

Следниот чекор од методологијата се однесува прашањето "зошто податочното множество треба да биде множество со 5-star квалитет?". Множество со 5-star квалитет е, всушност, множество кое е поврзано со други податочни множества, односно множество кое го имплементира четвртиот принцип на поврзани податоци. Според тоа, оваа фаза се однесува на увидување на вистинските потреби и предности кои поврзувањето со други податочни множества би го донело. Додавањето линкови во едно податочно множество кон друго, обезбедува контекст за податоците.

Поврзувањето може да биде од еден ентитет кон одредени податоци за него, достапни во друго податочно множество (на пример: поврзување на податоци за одредена компанија и нејзината локација, со конкретни вредности за географска ширина и должина од друго податочно множество), или, пак, поврзување на ентитетот со податочен ентитет од друго множество кој го опишува истиот објект од реалниот свет (на пример: поврзување на одредена компанија од податочното множество со истата компанија од друго податочно множество). Во двата случаи агент (корисник или софтвер) кој работи со првото податочно множество, може да дојде до повеќе податоци следејќи ги линковите.

Овој чекор е директно во линија со генералната идеја на поврзаните податоци да се децентрализира трудот за интеграција на податоците (Слика 3.3). Имајќи ја предвид идната интеграција, оригиналните објавувачи на податочни множества можат да понудат квалитетни линкови до други податочни множества, кои понатаму ќе послужат како основа за проширување на поврзувањата, но и на корисничките сценарија на апликациите кои ги користат.



Слика 3.3: Децентрализација на трудот за интеграција на податоци.

3.2.6 Чекор 6: Кориснички сценарија

Последниот чекор од методологијата се однесува на корисничките сценарија. Иако станува збор за помошен чекор, Hausenblas et al. сметаат дека тој е од огромна

важност за демонстрирање и валидирање на придобивките од примената на принципите на поврзани податоци во дадениот домен. Креирањето кориснички сценарија - преку кориснички апликации, сервиси, проекти - ги вади на виделина придобивките од технологиите на поврзани податоци: пред сè економските предности и зголемената транспарентност во доменот на е-Влада.

3.3 Методологија на Villazón-Terrazas et al.

На сличен начин како и Hyland et al., врз база на нивното искуство во продукција на владини поврзани податоци во неколку различни контексти, Villazón-Terrazas et al. дефинираат методологија за генерирање, објавување и искористување на поврзани владини податоци [195]. Нивната методологија користи модел на инкрементален животен циклус, кој се базира на континуирано подобрување и проширување на поврзаните податоци, преку изведување на неколку итерации. Иако методологијата е иницијално наменета за владини податоци, таа може да се примени и на податоци од други домени.

Методологијата се состои од следните пет чекори (Слика 3.4):

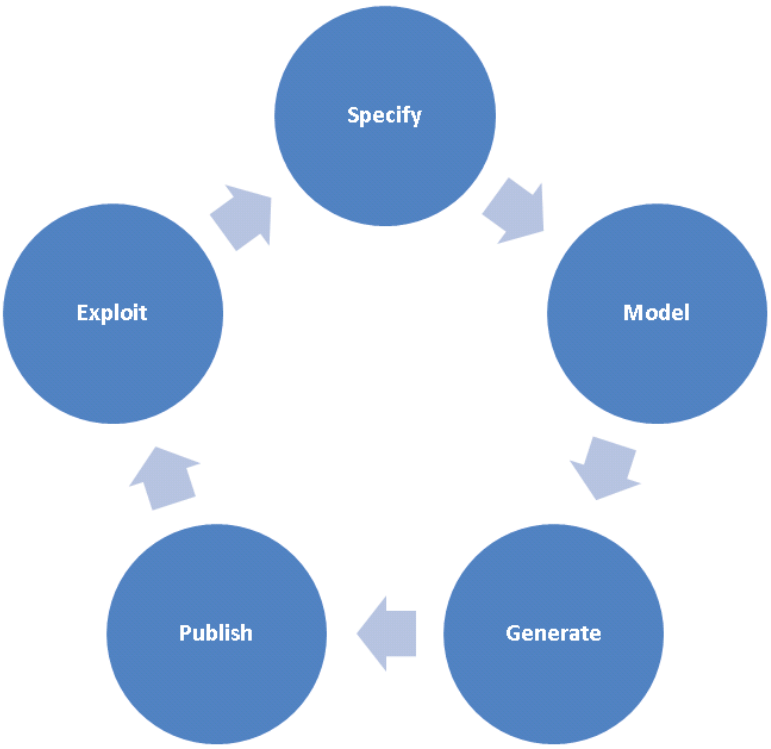
1. Спецификација
2. Моделирање
3. Генерирање
4. Објавување
5. Искористување

Секој од овие чекори се состои од една или повеќе задачи, кои користат разни техники и алатки. Редоследот на чекорите може да се промени во зависност од специфичните потреби на одредена институција.

3.3.1 Чекор 1: Спецификација

Како и кај повеќето проекти кои имаат за цел имплементација и понатамошен развој на едно ИТ решение, првата активност се однесува на креирање детална спецификација за потребите. Во овој чекор од креирањето поврзани податочни множества, за владините институции се важни и потребите на институцијата, покрај софтверските и хардверските побарувања. Задачите кои треба да се исполнат во оваа фаза се (а) идентификација и анализа на податочните извори, (б) URI дизајн и (в) дефинирање лиценца.

(а) Првата задача се состои од идентификација и селекција на податоците кои би се објавиле. Притоа, се прави дистинкција помеѓу податоци кои треба прв пат да се отворат и објават и податоци кои се веќе отворени и објавени, но кои можат да се прошират и употребат повторно. За селектираните податочни извори треба да се пронајдат сите достапни податоци и документација за податоците и податочниот модел, да се идентификува шемата на концептуалните компоненти и нивните релации, како и ентитетите кои се опишани во податочното множество.



Слика 3.4: Методологијата на Villazón-Terrazas et al.

(б) Во "глобалната база на податоци" која се добива со користење на поврзани податоци, основен идентификатор на еден ресурс е неговото URI. Поради тоа, Villazón-Terrazas et al. ги нагласуваат генералните препораки од W3C [177, 176] според кои URI идентификаторите треба да бидат дизајнирани имајќи ја предвид нивната едноставност, стабилност и управувањето. Дополнително, нивната методологија ги користи и препораките специфични за доменот на владини податоци [107], како и препораките за повеќејазични домени [160].

Сумарно, нивните препораки содржат:

- URI идентификаторите треба да бидат доволно дескриптивни. Тие ќе се користат и од луѓе, поради што се препорачува да бидат разбирливи и недвосмислени уште на прв поглед.
- Се препорачува користење на коса црта (/), наместо тараба (#), за дефинирање на ентитетот во URI идентификаторот. Недостатокот на идентификаторите кои користат коса црта е потребата од две HTTP барања за да се добие ресурсот и неговиот опис при HTTP content negotiation процесот [113, 112], но тие се значително поефикасни кога станува збор за големи податочни множества и можат да го идентификуваат ресурсот директно [122]. Користењето тараба резултира со преземање на целата содржина од веб локацијата означена со идентификаторот до знакот тараба, што најчесто може да значи преземање на целото податочно множество кога бараме информации само за еден ентитет од него.
- Онтологијата и инстанците треба да имаат различни URI патеки. Поради тоа,

потребна е основна URI структура (на пример: `http://linkeddata.finki.ukim.mk/lod/dbm/`, `http://healthcare.linkeddata.finki.ukim.mk`), структура за онтологиите која би го содржела зборот ontology (на пример: `http://linkeddata.finki.ukim.mk/lod/dbm/ontology/`) и структура за ентитетите од податочното множество (на пример: `http://linkeddata.finki.ukim.mk/lod/dbm/data/`, `http://linkeddata.finki.ukim.mk/lod/dbm/resource/`).

- Кога станува збор за владини институции и податоци, потребно е да се користи официјалниот јазик на институцијата.

(в) Villazón-Terrazas et al. препорачуваат дефинирање на лиценца за владините податоци кои се објавуваат. Дел од лиценците кои веќе постојат и можат да се искористат во оваа фаза, ги вклучуваат Open Government лиценцата од Велика Британија [45], Creative Commons лиценцата [8] како и низа лиценци објавени од страна на Open Knowledge фондацијата [44].

3.3.2 Чекор 2: Моделирање

Следниот чекор во методологијата на Villazón-Terrazas et al. опфаќа моделирање на доменот на податоците со соодветна онтологија, односно вокабулар. Нивната централна препорака се однесува на повторно искористување на постоечки вокабулари [195, 91]. Покрај јасните придобивки од користење на постоечки и широко користени вокабулари за шематски опис на податоците во контекст на нивна глобална достапност и користење, авторите ја потенцираат и заштедата на време, труд и ресурси од страна на владините институции.

Дефинирањето на онтологија за моделирање на доменот на податоците треба да се состои од следните задачи: (а) барање соодветна онтологија за повторно искористување, (б) проширување на постоечка онтологија во случај кога не може да се лоцира онтологија која комплетно го опишува доменот и (в) креирање комплетно нова онтологија во случај кога не постојат ресурси кои би можеле да се искористат повторно.

(а) За првата задача, авторите препорачуваат пребарување низ постоечките каталози на онтологии и вокабулари и следење на препораките од [186] за селекција на најсоодветната онтологија согласно нивото на грануларност, доменот, но и општите карактеристики на онтологијата. Дел од каталозите кои тие ги препорачуваат се Swoogle [59] и Linked Open Vocabularies (LOV) [33].

(б) Во случај кога не може да се пронајде онтологија која целосно би го опишала доменот, Villazón-Terrazas et al. препорачуваат пребарување низ други ресурси од доменот кои не се стандардни онтологии или вокабулари, но кои можат да искористат повторно и да помогнат во дефинирањето на нова онтологија за доменот [194].

(в) Последната опција подразбира креирање на сопствена онтологија, од почеток. За таа цел, авторите препорачуваат следење на првото сценарио предложено од NeOn методологијата [185].

3.3.3 Чекор 3: Генерирање

Во фазата на генерирање на поврзани податоци, селектираните податоци од Чекор 1 треба да се трансформираат во RDF согласно онтологиите и вокабуларите селектирани или развиени во Чекор 2. Генерирањето на поврзани податоци се состои од следните задачи: (а) трансформација, (б) чистење и (в) поврзување.

(а) Трансформацијата подразбира "превод" на податоците од оригиналниот податочен формат во RDF. Целите на трансформацијата се двократни: (1) да се обезбеди целосна трансформација, односно прашањата поддржани од оригиналното податочно множество да бидат поддржани и од страна на трансформираното податочно множество и (2) генерираните RDF инстанци треба да бидат во линија со структурата на онтологијата која се користи. Техниките кои авторите ги препорачуваат се истите техники претставени во поглавје 2.3. Препорачаните методи, пак, за трансформација, се методите елаборирани во [194].

(б) Искуството на експертите во доменот на поврзани податоци покажува дека во податочните множества, скоро без исклучок, постои т.н. "шум", кој ги попречува апликациите во ефективно искористување на податоците [125]. Задачата за прочистување на податоците се состои од два дела: (1) идентификување и наоѓање на грешките и (2) корекција на пронајдените грешки. Авторите од [125] издвојуваат и неколку чести (типови) грешки кои се среќаваат:

- проблеми со достапност на HTTP URI идентификаторите,
- појавување на именски простори кој не се вокабулари (на пример: `rss:item`),
- појавување на релации креирани за специфичен домен (на пример: `foaf:tagLine` кај LiveJournal),
- појавување на погрешно именувани релации (на пример: `foaf:image` и `foaf:img`),
- погрешно трансформирани податочни типови (на пример: `true` вредност интерпретирана како `xsd:int` податок со вредност `1`).

Корекцијата на податоците може да се одвива на апликациска страна, кога развивачот на апликација која го користи податочното множество ќе пресретне грешки во податоците, или пак на страна на објавувачот / одржувачот на податочното множество. За вториот пристап во [125] препорачуваат користење на сервиси како што е RDF Alerts [50].

(в) Задачата на поврзување на податочното множество со други, надворешни податочни множества е во линија со четвртиот принцип на поврзани податоци, кој обезбедува основа за откривање нови податоци преку тековните. Креирањето на линковите помеѓу ентитетите кои се во некаква меѓусебна релација може да биде рачно, асистирано или целосно автоматско. Чекорите од кои се состои креирањето линкови се:

- Идентификација на целни податочни множества кон кои ќе се креираат линкови. За таа цел потребно е рачно да се истражат објавените податочни множества преку Datahub [12] и да се провери нивната содржина.

- Идентификација на релациите помеѓу ентитетите од податочното множество и ентитетите од целните податочни множества. Поврзувањето на ентитетите кои се во одредена меѓусебна релација може да се реализира со користење на алатки како што се Silk [196] и LIMES [165].
- Валидација на креираните линкови. Овој чекор најчесто се изведува од страна на експерти од доменот, кои можат да ја потврдат валидноста на релациите.

3.3.4 Чекор 4: Објавување

Објавувањето на RDF податоците се состои од: (а) објавување на податочното множество, (б) објавување на метаподатоци и (в) овозможување на ефективно откривање на податоците.

(а) Податочното множество, трансформирано од оригиналниот формат во RDF, треба да се објави на Веб. За таа цел, авторите препорачуваат користење на некој од механизмите детализирани во поглавје 2.4. Изборот на вистинската алатка, или множество од алатки, зависи од потребите: дали е неопходен пристап преку SPARQL endpoint, преку HTML интерфејс и / или преку Linked Data интерфејс. За повеќе детали околу постоечките "рецепти" за објава на RDF податоци, авторите посочуваат кон [122].

(б) Откако податочното множество ќе биде објавено, потребно е да се објават метаподатоци за него. За таа цел се користат два вокабулари: (1) VoID (Vocabulary of Interlinked Datasets) [76], кој овозможува опис за структурата на податочното множество, начините за пристап до него, како и опис на линковите на податочното множество со други податочни множества; и (2) OPM (Open Provenance Model) [161], кој овозможува опис на потеклото на податоците во податочното множество, елемент кој е од голема важност кога станува збор за владини податоци.

(в) Последната задача при објавувањето е преземањето одредени чекори кои ќе овозможат ефективно откривање на податочното множество. За таа цел, потребно е да се преземат следните акции:

- Да се генерира sitemap опис за податочното множество. Ваквиот опис содржи информации како што се време на последни измени на податоците, фреквенција на промени и приоритет, што им овозможува на семантичките веб пребарувачи да знаат кои се најнови податоци и измени во множеството, без правење посебна анализа. Во оваа фаза треба да се генерира sitemap.xml датотека и да се поднесе кај веб пребарувачите како Google и Sindice.
- Податочното множество треба да се вклучи во дијаграмот на облакот на поврзани податоци [30]. За таа цел, потребно е податочното множество да се објави во Datahub [12] каталогот, согласно дефинираните критериуми [32].
- Податочното множество треба да се објави и на каталози за отворени владини податоци. Каталозите кои Villazón-Terrazas et al. ги препорачуваат за оваа намена веќе не постојат.

3.3.5 Чекор 5: Искористување

Според Villazón-Terrazas et al., главната цел на објавувањето на владини податоци е овозможувањето транспарентност, придонесот кон повеќе јавни апликации и охрабрување за јавна и комерцијална употреба на владините информации [195]. За да се постигне ова, тие сметаат дека треба да се развиваат апликации над отворените владини поврзани податоци, апликации кои нудат богат графички кориснички интерфејс за граѓаните.

Генерално, постојат два типа на апликации изградени над поврзани податоци: генерички апликации и апликации за конкретен домен [122]. Како генерички апликации можат да се сретнат прелистувачи за поврзани податоци и пребарувачи за поврзани податоци. Како апликации специфични за одреден домен, авторите ги издвојуваат апликациите на US Global Foreing Aid [65], која комбинира и визуелизира податоци од различни гранки од владата на САД, Talis Aspire [60], која има помага на наставниците во креирање и одржување на листи со едукативни ресурси и DBpedia Mobile [14], која им помага на туристите при посета на нов град.

Без разлика на типот на апликацијата, авторите на методологијата потенцираат дека најважна е интеграцијата на поврзани податоци од различни (владини и невладини) извори.

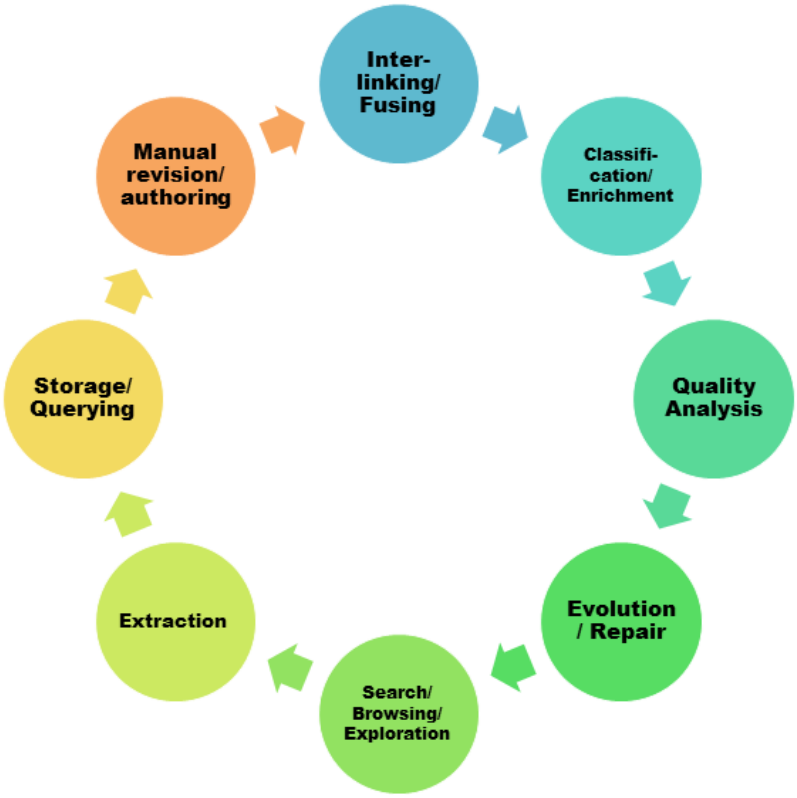
3.4 Методологија на LOD2 проектот

LOD2 е проект во чии рамки е развиена интегрирана околина од нови и претходно постоечки алатки, која го поддржува комплетниот животен циклус на екстракција, креирање, проширување, поврзување и одржување на поврзани податоци [81]. Животниот циклус на поврзаните податоци поддржан од LOD2 интегрираната околина, се состои од (Слика 3.5): (а) екстракција, (б) складирање, (в) креирање, (г) поврзување, (д) класификација и проширување, (ѓ) квалитет, (е) еволуција и корекција, (ж) пребарување и истражување.

(а) Првата фаза, екстракција, се однесува на издвојување на податоците кои треба да се трансформираат во поврзано податочно множество, од оригиналниот извор. Екстракцијата може да се изведува со помош на техники за процесирање на природен јазик, текстуално рударење и анотација, кога изворот е во неструктуриран формат. Доколку станува збор за полуструктурирани или структурирани податоци, тогаш можат да се користат други техники и алатки опишани во поглавје 2.3.

(б) Фазата на складирање се однесува на изборот на технологија за складирање на поврзаното податочно множество. Поради релативно младата проблематика на работа со RDF податоци, складирањето и управувањето со RDF податоци е попредизвикувачки проблем отколку работата со релациони бази на податоци.

(в) Фазата на креирање, пак, се однесува на општо на генерирањето RDF податоци, преку техниките опишани во поглавјето 2.3.



Слика 3.5: Методологијата на LOD2 проектот.

(г) Креирањето и одржувањето на линкови помеѓу податочните множества, ја сочинуваат третата фаза. Авторите од [81] потенцираат дека (полу)автоматизирањето на овој процес сè уште претставува предизвик, кој е од суштинска важност за воспоставување кохерентност и обезбедувањето податочна интеграција.

(д) Класификацијата се однесува на користење онтологии од повисока хиерархиска поставеност, кои би помогнале во попрецизна и поконкретна поделба на податочните множества во категории. Ваквата класификација ги олеснува процесите на интеграција, спојување и пребарување на податоците од податочните множества достапни на Вебот. Покрај класификација, податочните множества треба да бидат и проширени со дополнително знаење, добиено преку RDF расудување.

(ѓ) Слично како што квалитетот на содржината на веб страните на Вебот варира, варира и квалитетот на (поврзаните) податоци на Вебот. Поради тоа, значаен чекор во работата со поврзани податоци е користењето на техники за контрола на квалитетот, контекстот и структурата на податоците од податочните множества.

(е) Податоците на Вебот, како и другите податоци, се динамични. Животниот циклус на едно поврзано податочно множество треба да вклучува и техники за справување со промени во податоците, онтологиите и вокабуларите. Самиот дизајн на RDF моделот и RDFS и OWL јазиците за онтологии дозволува нивно проширување, но евентуалните промени во значењето на постоечките класи и релации искористени за анотација на податочното множество бараат надзор и ажурирање на самите податоци.

(ж) Последната фаза од LOD2 методологијата се однесува на зголемување на видливоста на поврзаните податочни множества. Тие препорачуваат имплементација на различни механизми за пребарување, прелистување, истражување и визуелизација на податоците (просторни, временски, статистички), со цел доближување на "мрежата од податоци" до вистинските корисници.

Според авторите, наведените фази не треба да се третираат изолирано; при нивното решавање треба да се имаат предвид и останатите фази, затоа што одлуките во една фаза можат да имаат големо влијание во друга фаза. Пример за тоа се дефинирањето мапирања на ниво на онтологии или вокабулари, детектирањето на погрешни мапирања помеѓу онтологии, користењето RDF складишта кои овозможуваат подобри перформанси при SPARQL прашања, итн. Авторите зборуваат и за користење на техники од машинско учење за одредување на нови мапирања помеѓу користените онтологии, како и за откривање на погрешно дефинирани мапирања. Прецизно дефинирани онтологии и мапирања помеѓу нив понатаму овозможуваат користење на RDF расудување, со цел проширување на податочните множества со ново знаење.

Глава 4

Трансформација и употреба на поврзани податоци во различни домени

Во рамките на нашите истражувања работевме на трансформација на податоци од различни домени во висококвалитетни отворени (семантички, 4-star) и поврзани (5-star) податоци (Слика 2.2). Целта на истражувањата беше двократна: да се анализираат методите, техниките и алатките за генерирање и работа со отворени и поврзани податоци во секој од домените; да се истражат придобивките од имплементацијата на принципите на поврзани податоци во различните домени.

Во оваа Глава ќе направиме преглед на истражувањата со поврзани податоци и добиените резултати во домените на полицијата, јавниот транспорт, финансиите, мултимедијата, здравството, фармацијата и гастрономијата.

4.1 Отворени податоци за криминалот во Македонија

Доменот на полицијата е од голем интерес за истражувачите од областа на податоци, отворени податоци и поврзани податоци [108][99]. Поради тоа, еден од нашите истражувачки потфати се однесуваше токму на податоците од Министерството за внатрешни работи (МВР) на Република Македонија [190]. Истражувањето резултираше со развој на отворена платформа [9] за мониторинг и анализа на кривичните дела регистрирани на територијата на Република Македонија и објавени од страна на МВР на нивната официјална веб страна.

Како дел од програмата за отворена влада, Министерството за внатрешни работи на Република Македонија од јуни 2011 година започна со редовна објава на дневни билтени на нивната официјална веб страна [40] за селектирани криминални активности кои се случиле претходните денови. Поради тоа што билтените се напишани во описна текстуална форма (Слика 4.1), станува збор за податоци од прва категорија согласно скалата за квалитет на отворени податоци (Слика 2.2). Со цел да ја зголемиме употребливоста на овие податоци, креиравме систем за трансформација на податоците во податоци од трета категорија и имплементиравме веб апликација која ги користи овие

податоци.

Слика 4.1: Дневен билтен од МВР за 09.01.2016.

4.1.1 Трансформација на податоците

Прв чекор беше добивањето на податоците. За таа цел имплементиравме ‘пајак’ - софтверски агент кој посетува одредена веб страна и ја собира целата нејзина содржина. Нашиот софтверски агент беше конфигуриран автоматски да се стартува секој ден во точно определено време, да ја посети официјалната веб страна на МВР каде што се објавуваат најновите дневни билтени и да ја собере содржината.

Иако содржината на веб страната со дневните билтени на МВР е во текстуален формат, поради формалниот пристап на известувањето во официјалните билтени реченици-те и параграфите следат одредена структура. Анализата на структурата нѝ овозможи да ја предвидиме позицијата на податоците од интерес во секој од параграфите. Со примена на техники за процесирање на природен јазик успеавме да ги добиеме податоците за (а) типот на кривичното дело, (б) локацијата (конкретен град или село во

Македонија), (в) адресата, (г) описот и (д) датумот.

За полесна детекција на типот на криминалното дело користевме листа на криминални дела наведени во Кривичниот законик на Република Македонија. Формалниот карактер на билтените наложува користење на точните називи на секое дело, што овозможува недвосмислена детекција на типот на делото. За полесна детекција, пак, на локацијата, односно градот или селото во кое се случило кривичното дело, користевме база од сите населени места во Република Македонија. Датумот и адресата ги детектиравме со користење на регуларни изрази, додека описот се состои од целиот параграф кој се однесува на кривичното дело.

Детектираните податоци за секое од кривичните дела од детектираниот дневен билтен ги складиравме во класична релациона база на податоци (MySQL).

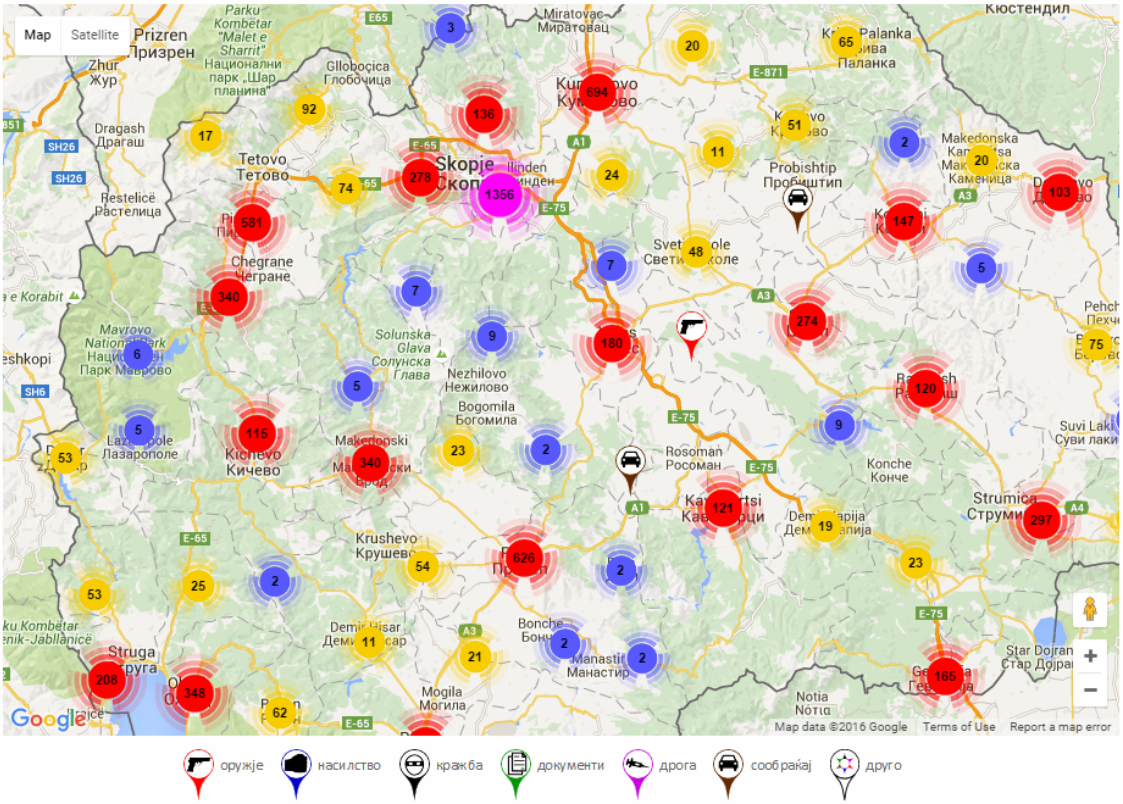
4.1.2 Веб апликација за мониторинг на криминалот во Македонија

За да ја демонстрираме предноста од користење отворени податоци со повисок квалитет - податоци од трета категорија, наместо прва категорија - развивме веб апликација [9] која овозможува географски преглед и анализа на кривичните дела во Македонија. Веб апликацијата е развиена со PHP програмскиот јазик, ја користи MySQL релационата база на податоци и го користи JavaScript јазикот на презентациска страна. Апликацијата го користи сервисот за мапи од Google Maps [187] за да ги исцрта како точки од интерес настаните детектирани во дневните билтени на МВР.

Со оглед на тоа што во дневните билтени како локација стои информацијата за адреса и населено место, пред да ги запишеме во база на податоци, податоците за одредено кривично дело ги прошируваме со податок за географска ширина и географска должина на која се случил настанот. За оваа цел го користиме сервисот за геокодирање на Google Maps [187], кој како влезни параметри ги користи адресата и населеното место за да ја одреди точната позиција на даденото место на мапа.

Користејќи ги податоците од креираната база на податоци, ги исцртуваме кривичните дела врз мапата на Република Македонија и ги означуваме со соодветна икона, според типот на делото (Слика 4.2 и 4.3). Мапата овозможува преглед според локација, според тип на кривично дело, според конкретен датум или временски опсег. Со селекција на одредена точка од интерес апликацијата го презентира текстуалниот опис на кривичното дело, според содржината објавена во билтените на МВР.

Веб апликацијата нуди можност и за директно преземање на трансформираните податоци од трета категорија во XML формат или како MySQL копија. Во духот на отворените податоци, ова овозможува понатамошни истражувања од областа на податоци од МВР и полицијата.



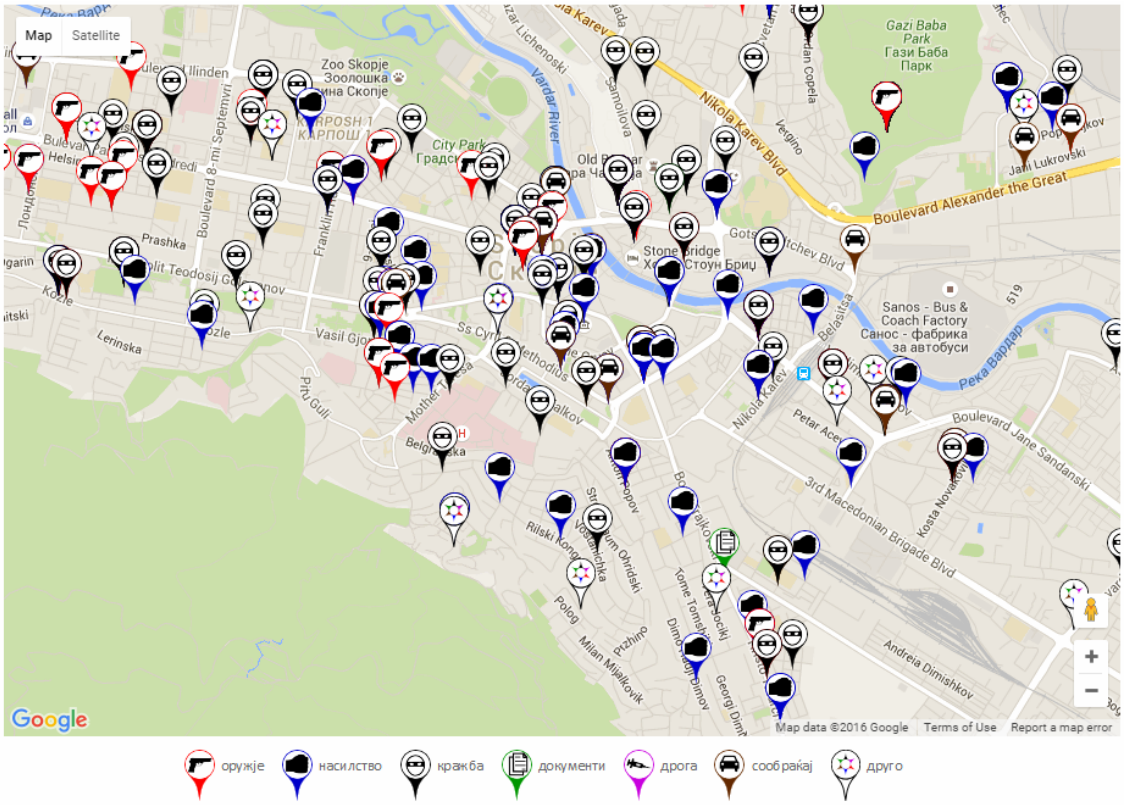
Слика 4.2: Изглед на мапата на криминал над целата територија на Република Македонија.

4.1.3 Дискусија

Автоматизираниот систем за секојдневно читање на новите дневни билтени од веб страната на МВР, трансформација во податоци од трета категорија, геокодирање и зачувување во базата, овозможува веб апликацијата и нејзината мапа постојано да бидат ажурирани и да прикажуваат најнови податоци. Во досегашната работа на апликацијата, од почетокот на 2012 година до почетокот на 2016 година, детектирани се и обработени над 8.000 кривични дела објавени во дневните билтени на МВР.

Мапата на криминалот во Македонија овозможува анализа на криминалот според тип и локација за самото МВР и за поширокаta јавност, но и компаративна анализа на стапките на одредени криминални активности со други социо-економски параметри. Првиот вид на анализа е значаен за секое министерство за внатрешни работи, па и за македонското МВР. Таа овозможува детекција на шеми на однесување на одредени криминални структури и одредување на криминални жаришта, што може да помогне во поефективна и поефикасна организација на ресурсите во борбата со одредени типови криминал.

Јавната достапност на ваков тип на мапа на криминал, пак, допринесува и до пошироka информираност на граѓаните за криминалните дејствија кои се случуваат на локации од нивен интерес. Ваквата мапа и информациите на неа можат да им помогнат



Слика 4.3: Изгледа на мапата на криминал на дел од територијата на градот Скопје.

на граѓаните при избор на ново место на живеење, избор на градинка или училиште за нивните деца, итн.

Компаративната анализа, пак, на шемите на криминалните настани со други социоекономски параметри може да се направи доколку над податоците од мапата за криминал се додадат податоци од други множества: податоци за степен на образование по општина, податоци за економската состојба на семејствата по општина, податоци за старосната структура по општина, итн. Отворениот пристап до податоците од мапата на криминал нуди можност за вакви и слични истражувања во иднина.

4.2 Отворени податоци за јавен транспорт и аерозагадување

Јавниот транспорт, управувањето со сопственото време и дневните патеки на движење низ урбаната средина имаат големо влијание врз квалитетот на животот на граѓаните. Пристапот до вистинските податоци во вистинското време може значително да влијае врз времето кое го поминуваме во патување од една до друга локација во градот при секојдневни активности. Поради тоа, го истражувавме ефектот кој отворените и поврзани податоци можат да го дадат во доменот на јавниот транспорт во Македонија и во Шведска, CO_2 емисиите од возилата во Европската Унија и во доменот на

квалитетот на воздухот во градот Скопје.

Во продолжение ќе направиме преглед на текот на истражувањата и нивните резултати.

4.2.1 Отворени податоци за јавен транспорт во Македонија

Првото истражување од доменот на јавниот транспорт беше фокусирано на податоци од Република Македонија. За целите на истражувањето работевме со податоците од јавното сообраќајно претпријатие (ЈСП) Скопје. Податоците ги трансформиравме во стандардизираниот General Transit Feed Specification (GTFS) формат, а потоа ги трансформиравме во семантички, отворени податоци од четврта категорија. За последната трансформација ги користиме Transit онтологијата и вокабуларот W3C Geospatial онтологијата, како и наша сопствена онтологија, GTFS-ext. Генерираното податочно множество со податоци од четврта категорија е објавено согласно принципите на поврзани податоци и може да ги поддржи корисничките сценарија предложени во нашето истражување [158].

Трансформација на податоците

GTFS форматот [22] е наменет за моделирање на податоците од доменот на јавниот транспорт. Форматот е развиен од страна на Google и е наменет како влезен формат за дел функционалностите на нивниот Google Maps сервис. Овие функционалности вклучуваат информации за крајните корисници за кој јавен превоз можат да го искористат за да во одреден термин стигнат од точка А до точка Б во одредено населено место кое има јавен транспорт. Податоците во GTFS формат можат да се искористат и прикажат на мапа и од страна на кориснички апликации [79].

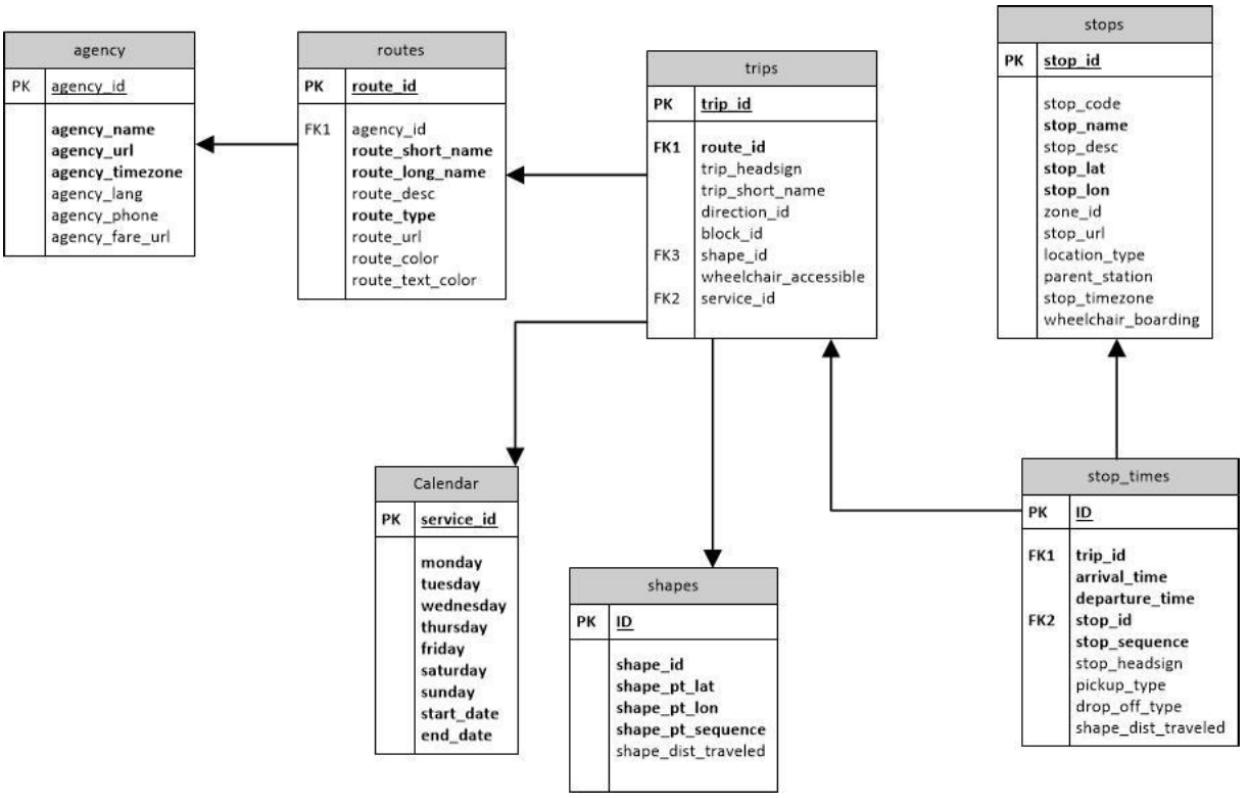
Според дефиницијата на GTFS форматот, податоците се состојат од .txt датотеки во CSV формат, при што секоја од датотеките претставува различен дел од податоците, напишан во стриктно дефиниран формат [144].

Јавното сообраќајно претпријатие (ЈСП) Скопје е јавниот превозник во градот Скопје, кој обезбедува услуги за јавен транспорт со своите автобуси кои оперираат по дефинирани патеки на движење и дефиниран временски распоред. Овој временски распоред е достапен на официјалната веб страна на ЈСП Скопје [28], во стандарден HTML формат. Како дел од друг проект на Факултетот за информатички науки и компјутерско инженерство во Скопје, податоците од веб страната на ЈСП Скопје беа собрани, прочистени и трансформирани во GTFS формат. Базата на GTFS податоци од ЈСП Скопје се состои од седум GTFS табели (Слика 4.4):

- `agency` - содржи информации за една или повеќе агенции за јавен транспорт
- `stops` - содржи информации за сите транспортни станици во населеното место
- `routes` - содржи информации за сите патеки на движење на агенцијата
- `trips` - содржи информации за конкретни возења, односно секвенци од две или повеќе транспортни станици до кои транзитира агенцијата во одреден временски

период

- `stop_times` - содржи информации за времето на доаѓање и тргнување на јавното превозно средство на конкретна транспортна станица за конкретно возење
- `calendar` – содржи информации за возниот ред
- `shapes` - содржи информации за просторниот приказ на патеката на движење, за да може да биде исцртана на мапа

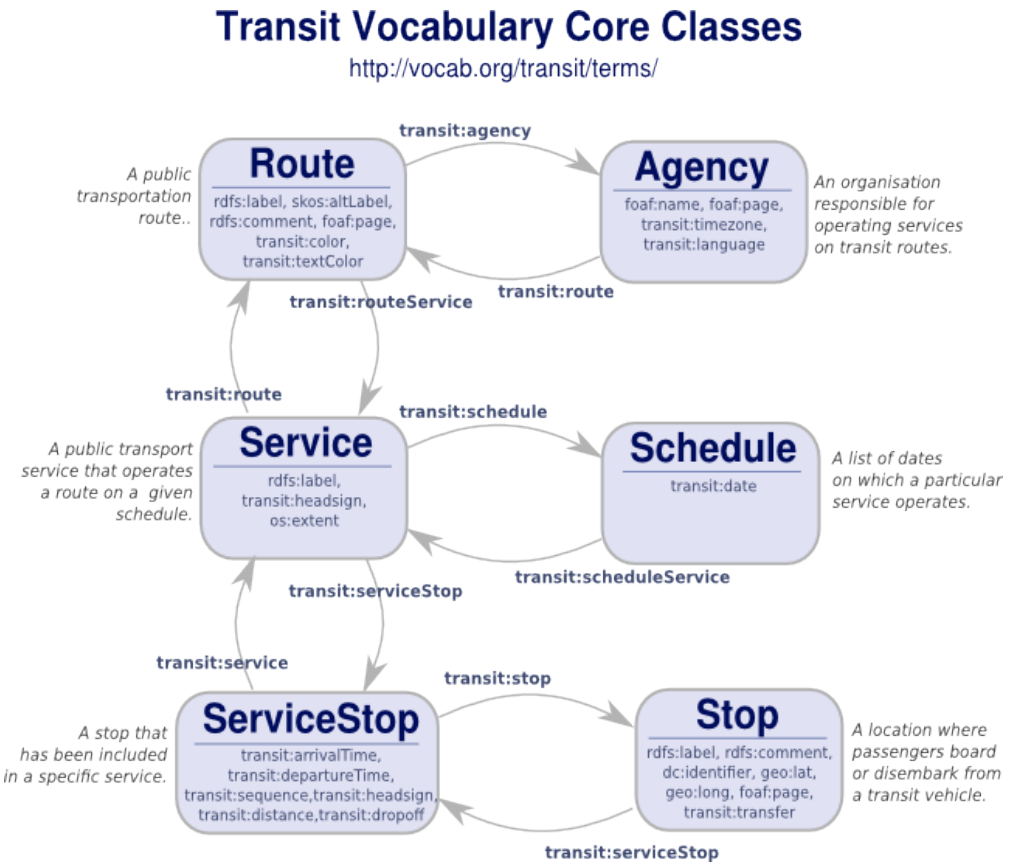


Слика 4.4: GTFS шемата на податоците од ЈСП Скопје.

Податоците од ЈСП Скопје во нивниот GTFS формат претставуваат податоци од трета категорија. Со цел да ја зголемиме нивната употребливост во рамките на мрежата од податоци, ги трансформиравме во податоци од четврта категорија. За таа цел, потребно е да се користи соодветна онтологија или вокабулар.

Истражувањето на доменот покажа дека постои Transit онтологијата [109] (Слика 4.5), која содржи дел од класите и релациите кои нѝ се потребни. И покрај тоа што главната намена на онтологијата е анотација на GTFS податоци, фактот дека дел од податоците според форматот се опционални значи дека онтологијата не е целосно компатибилна со нашите GTFS податоци од ЈСП Скопје. И покрај тоа, согласно принципите на поврзани податоци одлучивме да ја искористиме оваа онтологија за сите класи и релации за кои таа одговара, а за останатите елементи од шемата ги искористивме својствата од W3C Geospatial онтологијата [96] и креиравме наша онтологија, GTFS-ext.

Од Transit онтологијата ги искористивме класите: `transit:Route`, `transit:Stop` и



Слика 4.5: Transit онтологијата.

`transit:Agency`. Како објектни својства ги искористивме својствата `transit:route`, `transit:agency` и `transit:stop`, додека како податочни својства ги искористивме својствата `transit:timezone`, `transit:language`, `transit:sequence`, `transit:distance`, како и `transit:headsign`, `transit:color` и `transit:textColor`. Од W3C Geospatial онтологијата ги искористивме `geo:lat` и `geo:long` за означување на географската ширина и географската должина на точките од интерес.

Останатите класи и својства ги дефиниравме во нова онтологија, GTFS-ext, која ја проширува Transit онтологијата и ги обезбедува класите и релациите потребни за целосно мапирање на GTFS податоците од ЈСП Скопје во RDF податоци. Класите од GTFS-ext онтологијата се прикажани во Табела 4.1, додека објектните својства во Табела 4.2.

Согласно препорачаните практики од W3C, GTFS-ext онтологијата е објавена на Веб преку систем кој поддржува HTTP content negotiation: `http://linkeddata.finki.ukim.mk/lod/ontology/gtfs-ext#`.

По дефинирањето на онтологиите, следниот чекор беше трансформација на податоците во RDF. За оваа цел ги трансформиравме податоците од релационата база на податоци во CSV формат, а потоа искористивме инстанца од Virtuoso Universal Server [68] која обезбедува механизми за трансформација и менаџирање со разни типови на податоци. Инстанцата служи како сервер за поврзани податоци кој овозможува ло-

Табела 4.1: Класи од GTFS-ext онтологијата

Класа	Опис
`gtfs-ext:Shape`	Класа која соодвествува на `shapes` табелата од GTFS базата на ЈСП Скопје
`gtfs-ext:Trip`	Класа која соодвествува на `trips` табелата од GTFS базата на ЈСП Скопје
`gtfs-ext:StopTimes`	Класа која соодвествува на `stop_times` табелата од GTFS базата на ЈСП Скопје
`gtfs-ext:Calendar`	Класа која соодвествува на `Calendar` табелата од GTFS базата на ЈСП Скопје

Табела 4.2: Објектни својства од GTFS-ext онтологијата

Својство	Опис
`gtfs-ext:service`	Поврзува инстанца од `gtfs-ext:Trip` со инстанца од `gtfs-ext:Calendar`
`gtfs-ext:st_times`	Поврзува инстанца од `gtfs-ext:Trip` со инстанца од `gtfs-ext:StopTimes`
`gtfs-ext:shape`	Поврзува инстанца од `gtfs-ext:Trip` со инстанца од `gtfs-ext:Shape`

кално и дистрибуирано пребарување (преку мрежа или преку Вебот) на податоците со користење на SPARQL прашалниот јазик. Оваа функционалност е овозможена од страна на SPARQL endpoint кој се активира автоматски и кој може да се користи како REST-базиран веб сервис.

Процесот на мапирање се одвиваше во неколку фази. Најпрвин, податоците беа вчитани во релациона база на податоци во Virtuoso. Потоа, со користење на R2RML [106] - јазикот за мапирање на релациони податоци во RDF - креиравме т.н. RDF Views (RDF погледи) над релационата база на податоци. Со оглед на тоа што мапирањето се одвива табела по табела, оваа фаза резултираше со седум индивидуални RDF погледи во базата, по еден за секоја од GTFS табелите од ЈСП Скопје, кои потоа беа трансформирани во RDF графови во Virtuoso. Разликата помеѓу RDF поглед и RDF граф е во тоа што првото претставува виртуелна (логичка) организација на RDF тројките во базата, додека второто претставува нивна физичка организација. Виртуелната, односно логичка организација овозможува читање и пребарување на RDF тројките, но не и нивно менување или дополнување. За таа цел се користат RDF графовите.

Следниот чекор се однесуваше на поврзување на податоците од седумте индивидуални RDF графови. Поврзувањето, како и кај сите поврзани податочни множества, се реализира со додавање нови RDF тројки во кои субјектот е URI идентификатор од едно податочно множество, а објектот е URI идентификатор од друго. Ваквите RDF линкови ги додадовме со користење на SPARQL endpoint-от на Virtuoso инстанцата. Притоа, беа

креирани следните линкови:

- Ги поврзавме `transit:Route` инстанците со инстанци од `transit:Agency` со релацијата `transit:agency`.
- Ги поврзавме `gtfs-ext:Trip` инстанците со:
  - инстанци од `transit:Route` со `transit:route` релацијата.
  - инстанци од `gtfs-ext:Calendar` со `gtfs-ext:service` релацијата.
  - инстанци од `gtfs-ext:Shape` со `transit:shape` релацијата.
  - инстанци од `gtfs-ext:StopTimes` со `transit:st_times` релацијата.
- Ги поврзавме `gtfs-ext:StopTimes` инстанците со инстанци од `transit:Stop` со `transit:stop` релацијата.

Со тоа, го завршивме процесот на семантичка анотација на GTFS податоците од ЈСП Скопје, со што добивме податоци од четврта категорија.

Кориснички сценарија

Основната идеја зад креирањето на податоци со повисок квалитет, како што се транспортните податоци од четврта категорија од ЈСП Скопје, е креирањето јавно достапно податочно множество кое би можело едноставно да се користи од страна на други корисници како самостојно множество или во комбинација со други множества од сличен домен, со што би се креирале разновидни кориснички сценарија кои вклучуваат јавен транспорт.

Со нашето трансформирано податочно множество можеме да обезбедиме информации за автобуската станица и термините во кои корисникот треба да фати одреден автобус за да пристигне на саканата локација во градот Скопје, автобуската станица на која треба да се спушти и времето на пристигнување.

Прво корисничко сценарио. Доколку сакаме да го најдеме времето на тргнување на автобусите од одредена автобуска станица, за одредена автобуска патека на движење, во даден временски опсег во денот и во конкретен период од годината (зима, лето), можеме да го генерираме следното корисничко сценарио: ги бараме времињата на тргање на автобусите од линијата 'R15' од автобуската станица 'Ново Лисиче', во периодот од 09:00 - 10:00 часот, за време на работните денови во текот на зимскиот период. За оваа цел, можеме да го искористиме следното SPARQL прашање:

SPARQL прашање 4.1

```
PREFIX ont: <http://linkeddata.finki.ukim.mk/lod/ontology/transit-ont#>
PREFIX transit: <http://vocab.org/transit/terms/>
SELECT DISTINCT ?departure
WHERE {
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/routes#> {
    ?r ont:route_id "R15" .
  }
```

```
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/calendar#> {
    ?s ont:service_id "DELNIK_ZIMEN" .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/trips#> {
    ?x transit:route ?r ;
       ont:service ?s ;
       transit:headsign "Карпош 4" ;
       ont:st_times ?st .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/stop_times#> {
    ?st ont:departure_time ?dep ;
        transit:sequence 1 .
  }
  FILTER regex(?dep,"(^09:)|(^9:)")
}
```

Резултатите од SPARQL прашањето над податочното множество се дадени во Табела 4.3.

Табела 4.3: Резултати од SPARQL прашањето

Време на тргнување
"09:55:00"

SPARQL прашањето го добива URI идентификаторот на патеката на движење идентификувана со `ont:route_id` = "R15" од RDF графот со патеки и го наоѓа возниот ред кој има `ont:service_id` со вредност "DELNIK_ZIMEN" од RDF графот со календари за возниот ред. Потоа, од RDF графот со возења ги наоѓа конкретните движења на автобусите кои се однесуваат на пронајдената патека на движење и пронајдениот возен ред и кои како крајна дестинација ја имаат станицата "Карпош 4". За пронајдените возења ги селектираме времињата на застанување на секоја попатна автобуска станица. Од нив, го селектираме времето на поаѓање од првата страница и ги филтрираме резултатите за да останат времињата на поаѓање кои се помеѓу 09:00 - 10:00 часот.

Второ корисничко сценарио. Како второ корисничко сценарио, може да претпоставиме дека е потребно да ги најдеме сите патеки на движење кои поминуваат низ една конкретна автобуска станица. За оваа цел би нѝ биле потребни четири од RDF графовите: графот со автобуски станици, од каде ќе го најдеме URI идентификаторот на станицата; графот со времиња на застанување, од каде ќе ги најдеме времињата на поаѓање и застанување за конкретната станица и конкретните возења кои поминуваат низ неа; графот со возења, од каде ќе ги најдеме генералните патеки на движење кои одговараат на пронајдените возења; и графот со патеки на движење, од каде ќе ги најдеме имињата на патеките.

Нека автобуската станица за која бараме информации биде некоја пофреквентна, како на пример станицата “Мал Одмор”. За таа цел, можеме да го искористиме следното SPARQL прашање:

SPARQL прашање 4.2

```
PREFIX transit: <http://vocab.org/transit/terms/>
PREFIX ont: <http://linkeddata.finki.ukim.mk/lod/ontology/transit-ont#>
SELECT DISTINCT ?route ?name
WHERE {
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/stops#> {
    ?stop ont:name "МАЛ ОДМОР" .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/stop_times#> {
    ?stopTimes transit:stop ?stop .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/trips#> {
    ?trip ont:st_times ?stopTimes ;
          transit:route ?route .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/routes#> {
    ?route ont:name ?name .
  }
}
```

Резултатите од SPARQL прашањето над податочното множество се дадени во Табела 4.4.

Табела 4.4: Резултати од SPARQL прашањето

Патека	**Име**
http://vocab.org/transit/terms/Route/R2	“Сарај - Автокоманда”
http://vocab.org/transit/terms/Route/R4	“11 Октомври - Нас. Хром”
http://vocab.org/transit/terms/Route/R7	“Нас. Лисиче - Карпош 3”
http://vocab.org/transit/terms/Route/R15	“Ново Лисиче - Карпош 4”
http://vocab.org/transit/terms/Route/R19	“Шуто Оризари - Карпош 4”
http://vocab.org/transit/terms/Route/R22	“Транспортен центар - Волково”
http://vocab.org/transit/terms/Route/R24	“Кисела Вода - Тафталиџе”
http://vocab.org/transit/terms/Route/R59	“Карпош 3 - Гробишта Бутел”

За да се добие листата од резултати, со SPARQL прашањето го правиме следново: започнувајќи од графот со автобуски станици, ја селектираме станицата која го носи

името "МАЛ ОДМОР", користејќи ја `ont:name` релацијата. Со ова, во променливата `?stop` го добиваме URI идентификаторот на станицата. Следно, селектираниот URI идентификатор го користиме за да од графот со времиња на застанување ги добиеме времињата на застанување и тргнување за конкретната автобуска станица, преку `transit:stop` релацијата. Овие времиња, пак, ги користиме во графот за возења од кој ги селектираме сите поединечни возења кои се поврзани со времињата на застанување и тргнување, преку релацијата `ont:st_times`. За овие возења ги селектираме и соодветните генерални патеки на движење, кои се поврзани со нив преку `transit:route` релацијата. На крај, од графот со патеките на движење ги добиваме името на секоја од пронајдените патеки, преку `ont:name` релацијата.

Трето корисничко сценарио. Над дефинираното податочно множество можеме да имаме и корисничко сценарио во кое нѝ е потребна информација за бројот на возења кои се случуваат на одредена автобуска станица во текот на еден ден, при одреден возен ред. На пример, може да го земеме "NEDELA_ZIMEN" возниот ред, кој важи за денот недела во зима и автобуската станица "Палма". Со овие параметри би ги избројале бројот на возења, односно нивни застанувања, кои ќе се случат на станицата во текот на денот недела во зима. За оваа цел можеме да го искористиме следното SPARQL прашање:

SPARQL прашање 4.3

```
PREFIX ont: <http://linkeddata.finki.ukim.mk/lod/ontology/transit-ont#>
PREFIX transit: <http://vocab.org/transit/terms/>
SELECT COUNT(DISTINCT ?trip) as ?COUNT
WHERE {
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/calendar#> {
    ?service ont:service_id 'NEDELA_ZIMEN' .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/trips#> {
    ?trip ont:service ?service ;
          ont:st_times ?stopTimes .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/stop_times#> {
    ?stopTimes transit:stop ?stop .
  }
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/stops#> {
    ?stop ont:name "ПАЛМА" .
  }
}
```

Резултатите од SPARQL прашањето над податочното множество се дадени во Табела 4.5.

Слично SPARQL прашање можеме да упатиме и за саботниот зимски возен ред и да ги споредиме резултатите со оние од Табела 4.5. Единствената промена потребна

Табела 4.5: Резултати од SPARQL прашањето

Вкупно
193

во SPARQL прашањето би била менување на распоредот (`ont:service_id` релацијата) во "SABOTA_ZIMEN". Резултатот од ова прашање е `COUNT = 275`, од што можеме да изведеме заклучок дека јавните автобуски линии во градот Скопје се пофреквентни во сабота отколку во недела, барем за конкретната автобуска станица.

Наведените SPARQL прашања за овие кориснички сценарија можат да се испратат директно до SPARQL endpoint-от [57] на нашата Virtuoso инстанца, која работи и како REST сервис. Ова значи дека апликациите кои сакаат да го користат ова податочно множество, можат да ги пребаруваат податоците со едноставни HTTP GET и HTTP POST барања и да ги добијат бараните податоци во широк спектар на RDF и не-RDF формати, како што се RDF/XML, Turtle, JSON-LD, RDF/JSON, N3, JSON, CSV, HTML, итн. Општиот формат на HTTP GET барањето е:

```
http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

каде што `SPARQLQUERY` е URL-кодираното SPARQL прашање кое сакаме да го поставиме, а `FORMAT` е форматот во кој сакаме да ги добиеме резултатите од пребарувањето.

4.2.2 Поврзани податоци за јавен транспорт во Шведска

Второто истражување од доменот на јавниот транспорт се однесуваше на развој на автоматизиран систем за генерирање на поврзани податоци во транспортниот домен, кој беше тестиран и валидиран врз база на отворените податоци од Шведската транспортна администрација (Swedish Transport Administration, STA). Во рамките на истражувањето ја развивме Transport Administration Ontology (TAO) онтологијата. Резултантните висококвалитетни поврзани податоци ги искористивме за напредни кориснички сценарија базирани на податоците од STA [164].

Трансформација на податоците

Шведската транспортна администрација (STA) [64] претставува владина агенција која е одговорна за долгорочно планирање и развој на националната мрежа на патишта на Шведска. Агенцијата собира голем број податоци за сообраќајот, во текот на сите 24 часови во текот на еден ден, преку цела година. Овие податоци и информациите добиени од нив се достапни до јавноста преку нивната веб страна и преку веб сервиси достапни на барање на развивачи на софтвер или истражувачи. Овие информации вклучуваат состојби на патиштата и сообраќајот во поголемите градови, на регионалните патишта и на автопатите низ Шведска.

Податоците на STA се достапни преку SOAP веб сервиси и се структурирани во Datex II формат [13]. Овој податочен формат е изведен од XML форматот и се користи како стандард за голем број владини институции во рамките на Европската Унија. Поради овој формат, податоците од STA можеме да ги класифицираме како 3-star податоци.

Со цел нивна трансформација во висококвалитетни, 5-star поврзани податоци, дизајниравме и изработивме автоматизиран систем кој во две фази ги собира и трансформира податоците:

- Автоматско собирање на податоци
  - Скрипта која се извршува на временски интервал пристапува до веб сервисите на STA и ги добива најновите XML податоци од податочното множество од интерес.
  - XML податоците се парсираат и трансформираат во RDF/XML формат со користење на онтологии за семантичка анотација.
  - RDF/XML податоците се зачувуваат во RDF граф во инстанца на Apache Jena Fuseki Server или се додаваат на веќе постоечкиот RDF граф.
- Трансформација во 5-star поврзани податоци
  - Извршуваме SPARQL-базирани процедури кои креираат линкови помеѓу ентитети од локалниот RDF град и соодветни ентитети од LOD облакот.

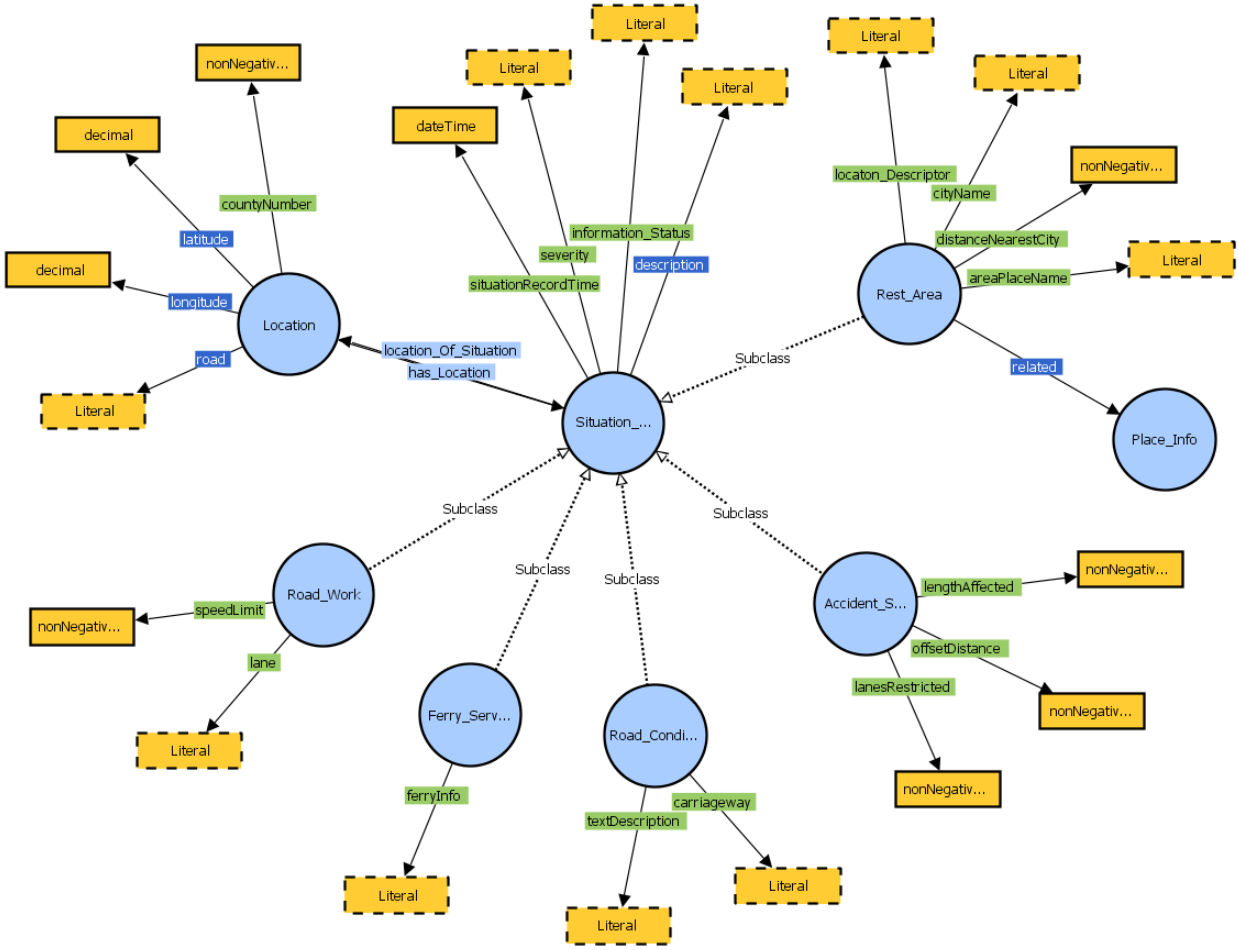
Автоматско собирање на податоци. Процесот на автоматско собирање на податоци се извршува со помош на Windows Batch скрипта во која е реализирано побарувањето, собирањето и локалното складирање на податоците. Скриптата се извршува на одредени предефинирани временски интервали. Собраните XML датотеки се парсираат и трансформираат во RDF/XML датотеки, кои понатаму се вчитуваат во инстанца на Apache Jena Fuseki Server. При секое извршување на скриптата, новогенерираните RDF/XML датотеки се додаваат на постоечкиот RDF град во Fuseki инстанцата. Овој главен RDF граф, кој постојано содржи нови податоци, е достапен преку перзистентно URI и е достапен преку SPARQL endpoint.

TAO онтологија. При трансформацијата на XML датотеките и податоците од нив во RDF/XML, главната компонента е онтологијата потребна за семантичка анотација на податоците. Нашето истражување во областа покажа дека таква готова онтологија не постои за доменот на транспорт, па за таа цел ја креиравме Transport Administration Ontology (TAO) онтологијата. Во рамките на TAO онтологијата употребивме и постоечки својства од други онтологии кои имаат широка употреба.

TAO онтологијата се состои од класи и својства кои ги опишуваат ентитетите и нивните атрибути од транспортниот домен (Слика 4.6). Според тоа, во онтологијата се дефинирани следните класи: `tao:Road_Condition`, `tao:Road_Work`, `tao:Rest_Area`, `tao:Ferry_Service` и `tao:Accident_Service` - сите изведени од заедничката суперкласа `tao:Situation_Record` - и класата `tao:Location`. Објектните својства дефинирани

во онтологијата се `tao:has_Location` и `tao:location_of_Situation`. Овие две својства се инверзни едно на друго и се користат за поврзување на инстанца од класата `tao:Situation_Record` (или нејзините под-класи) со инстанца од `tao:Location`.

Како податочни својства во онтологијата се дефинирани: `tao:situationRecordTime`, `tao:informationStatus`, `tao:speedLimit`, `tao:lengthAffected` и `tao:lanesRestricted`.



Слика 4.6: TAO онтологијата.

Согласно препорачаните практики од W3C, TAO онтологијата е објавена на Веб преку систем кој поддржува HTTP content negotiation: `http://linkeddata.finki.ukim.mk/lod/ontology/tao#`.

Трансформација во 5-star поврзани податоци. По генерирањето на RDF графот, следниот чекор во автоматизираниот процес е трансформација на податочното множество во 5-star поврзани податоци. За ваквата трансформација потребно е креирање линкови помеѓу ентитетите од локалното податочно множество со ентитети достапни преку LOD облакот. Со цел да реализираме поврзувања со ентитети од DBpedia, го користиме својсвото `skos:related` од SKOS онтологијата [156] и со него креираме RDF тројки кои ги поврзуваат инстанците на градови кои се среќаваат во нашето податочно множество, со соодветните инстанци на градови од DBpedia.

Кориснички сценарија

Со цел да ја демонстрираме предноста од достапност на овие податоци како поврзани податоци, анализиравме две сценарија во кои користиме податоци од нашето податочно множество, но и податоци од DBpedia.

Прво корисничко сценарио: податоци за сообраќајни незгоди. Во рамките на податоците кои се добиваат од Шведската транспортна агенција се наоѓаат и податоци кои се однесуваат на сообраќајни незгоди кои се случиле на одреден пат, во одреден временски период. За добивање на овие податоци можеме да искористиме ваков тип на SPARQL прашање:

SPARQL прашање 4.4

```
SELECT ?Time (fn:concat(?long, ", "+?lat) AS ?Point) ?Road ?Length
WHERE {
   ?Accident tao:situationRecordTime ?Time ;
             tao:lengthAffected ?Length ;
             tao:has_Location ?Location .
   ?Location place:Road ?Road ;
             geo:longitude ?long ;
             geo:latitude ?lat .
}
ORDER BY DESC(?Time)
LIMIT 2
```

Како одговор на ваквото прашање добиваме листа на последните 2 сообраќајни незгоди, времето на нивно случување, географската локација на незгодата означена преку географска ширина и должина, информација за патот на кој се случила незгодата како и должината на патот во метри која е зафатена поради незгодата. Пример резултати од извршувањето на прашањето се прикажани во Табела 4.6.

Табела 4.6: Резултати од SPARQL прашањето

Време	Локација	Пат	Должина
2015-06-10T19:35:28	12.0203247, 57.48617	Road 158	1516
2015-06-10T18:59:24	13.6646309, 55.75424	Road 1106	2060

Второ корисничко сценарио: користење податоци од DBpedia. RDF тројките кои со релацијата `skos:related` ги поврзуваат градовите од нашето податочно множество со градови од DBpedia, обезбедуваат основа за извлекување дополнителни податоци од DBpedia податочното множество, тргнувајќи од нашите податоци. Едно такво сценарио е добивање на дополнителни информации за попатни одморалишта на автопатите во Шведска кои се наоѓаат најблиску до одреден град:

SPARQL прашање 4.5

```
SELECT DISTINCT ?cityName ?RestAreaName ?abstract ?thumbnail
WHERE {
   ?city tao:cityName ?cityName;
         skos:related ?dbCity.
   SERVICE <http://dbpedia.org/sparql> {
      ?NearestCity dbpedia-owl:nearestCity ?dbCity;
                   dbpedia-owl:abstract ?abstract;
                   dbpedia-owl:thumbnail ?thumbnail;
                   dbpprop:name ?RestAreaName.
      FILTER langMatches(lang(?abstract), "en")
   }
}
```

Ова SPARQL прашање првин се извршува врз нашето податочно множество и го бара градот со името `?cityName`, кое може да се користи како параметар. За пронајдениот град се лоцира и соодветниот DBpedia ресурс со кој тој е поврзан, означен преку `?dbCity`. Потоа, преку SPARQL federation автоматски се испраќа потпрашање до SPARQL endpoint-от на DBpedia, од каде се наоѓаат сите попатни одморалишта за кои најблискиот град е `?dbCity` и се селектираат нивните резултати. Овие резултати се враќаат назад и му се прикажуваат на корисникот.

Наведените сценарија можат да се извршат директно на јавниот SPARQL endpoint од проектот, кој може да се искористи како REST-базиран сервис преку апликација, со праќање повици со следниот формат:

```
http://sta.linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

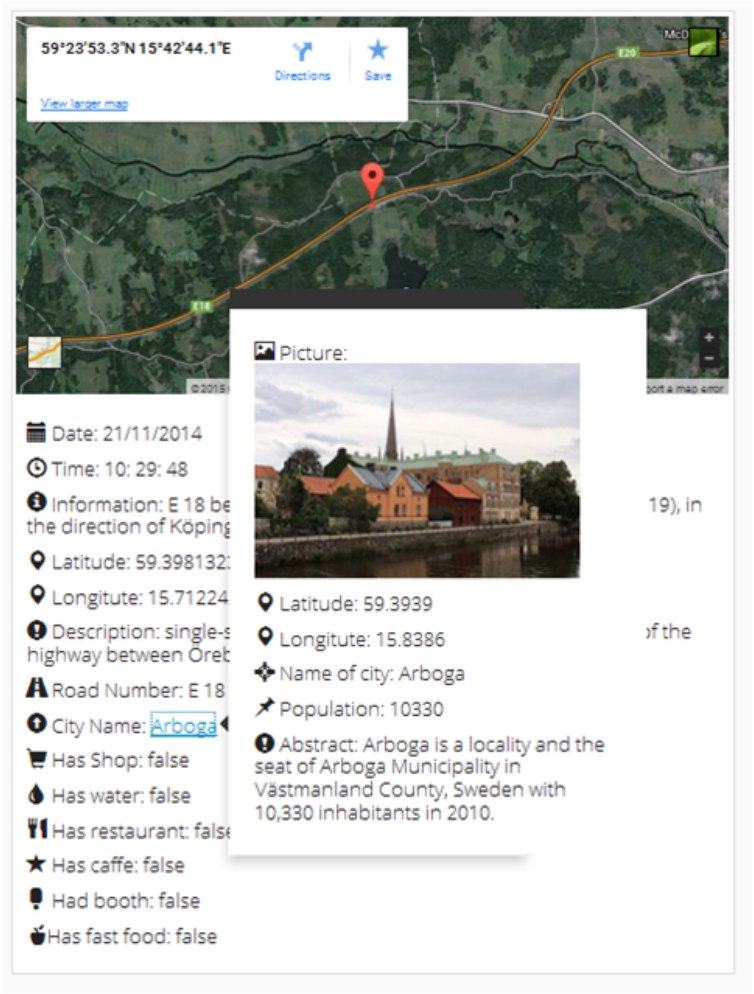
Тука, `SPARQLQUERY` се однесува на кодираното SPARQL прашање кое сакаме да се изврши, додека `FORMAT` се однесува на посакуваниот формат на одговорот и може да биде HTML, XML, JSON, CSV, RDF/XML, N3, Turtle, JSON-LD, итн.

Исто како и во претходните проекти и истражувања, ова може да се искористи од развивачи на софтвер за пристап до поврзани податоци од податочното множество.

Веб апликација

За подобра демонстрација на предностите од користењето на овие податоци во формат на поврзани податоци, развивме и веб апликација [58] (Слика 4.7). Веб апликацијата користи податоци од Apache Fuseki инстанцата и обезбедува информации врз база на најновите транспортни податоци од Шведската транспортна администрација. Веб апликацијата го користи SPARQL endpoint-от за пристап до податоците од локалното поврзано податочно множество и од LOD облакот.

Едно пример корисничко сценарио е прикажувањето на информации за одреден град (опис, популација, геолокација, итн.) добиени од надворешен извор - во случајов



Слика 4.7: Изглед на веб апликацијата.

од DBpedia. Овие податоци можат да се добијат со добивање на информации за конкретната инстанца од `tao:Rest_Area` класата и селекција на името на градот. Потоа, информациите му се прикажуваат на корисникот во еден едноставен прозорец, прикажан на Слика 4.7.

4.2.3 Поврзани податоци за CO_2 емисии од возила во ЕУ

Третото истражување од доменот на јавниот транспорт се однесуваше на трансформација на податоци за CO_2 емисии од возила, објавени од страна на Европската агенција за животна средина (European Environment Agency, EEA) и неколку други извори, во висококвалитетни 5-star поврзани податоци. Како дел од истражувањето ја развивме и Vehicle Emissions Ontology (VEO) онтологијата. Врз генерираното поврзано податочно множество дефиниравме кориснички сценарија со кои ја демонстриравме предноста од постоење на вакви податоци во формат на поврзани податоци [162].

Трансформација на податоците

Податоци за CO_2 емисии од возила. Европската агенција за животна средина (EEA) [18] е агенција во рамките на Европската Унија чија главна цел е обезбедување точни информации во врска со состојбите на животната средина. Агенцијата брои 33 земји членки кои претставуваат вреден извор на информации за ентитетите кои ги развиваат и евалуираат полисите поврзани со животната средина. Агенцијата на својата веб страна објавува отворени податоци во CSV и MDB формат. Овие податоци опфаќаат информации за температура на површина на мориња, емисии на сулфур диоксид, PM2.5 и PM10 вредности за загаденоста на воздухот, проекции, интерполирани мапи, итн. Податоците за кои ние бевме заинтересирано во рамките на истражувањето беа податоците за CO_2 емисии од патнички возила.

Со цел да го прошириме множеството на изворни податоци, ги вклучивме и податоците од Американската асоцијација на превозници со автобуси (American Bus Association, ABA) [2]. Асоцијацијата како дел од своите активности реализирала истражување во кое е направена компаративна анализа на CO_2 емисиите кај различни типови на транспорт [95], па дел од нивните податоци ги вклучивме како извор во нашето истражување.

За да ги опфатиме сите типови на транспорт, како извор ги вклучивме и податоците за емисии од авиони. Анализирајќи ги публикациите од доменот, како извор за ваквите податоци ги искористивме [168] и [166], во кои покрај другото се наоѓаат и податоците за CO_2 емисии кај различни типови на авиони.

Онтологии за анотација. Следејќи ги најдобрите практики за користење на онтологии, го анализиравме доменот за да лоцираме потенцијални онтологии кои би можеле да се искористат за анотација на податоците при трансформацијата од податоци со 1-star, 2-star и 3-star квалитет, во 5-star поврзани податоци. Анализата покажа дека неколку постоечки онтологии можат делумно да се употребат, но беше неопходно да се дизајнира и нова онтологија која комплетно ќе одговара на доменот.

Податочното множество од EEA содржи информации за специфични типови на патнички возила, како на пример производител, комерцијално име, капацитет на мотор, ширина на осовина, како и CO_2 емисиите на конкретниот тип возило. Дел од овие информации можат да се анотираат со релации од постоечки онтологии. Во Табела 4.7 се прикажани онтологиите, а во Табела 4.8 нивните својства кои ги користиме во нашиот модел за анотација.

Како класи за анотација на ентитетите кои се среќаваат во изворните податочни множества: патнички автомобили, автобуси, возови и авиони, ги искористивме класите `vvo:Automobile`, `vso:BusOrCoach`, `veo:Train` и `peo:AirplaneModel`, соодветно.

VEO онтологија. Со цел да обезбедиме можност за целосно мапирање на сите изворни податоци, ја креиравме Vehicle Emissions Ontology (VEO) онтологијата. Таа обезбедува дефиниции за класи и својства кои недостасуваат во постоечките онтологии. VEO онтологијата ја дефинира класата `veo:Train` како потткласа на `vso:Vehicle`, која ја користиме за означување на ентитетите кои претставуваат возови во податочното множество. Во онтологијата се дефинирани 16 податочни својства, кои одговараат на

Табела 4.7: Постоечки онтологии искористени при анотација

Онтологија	**Префикс**	**URI**
Geo Names	gn	`http://www.geonames.org/ontology#`
Good Relations	gr	`http://purl.org/goodrelations/v1#`
Volkswagen Vehicle Ontology	vvo	`http://purl.org/vvo/ns#`
Vehicle Sales Ontology	vso	`http://purl.org/vso/ns#`
Proton Extended Ontology	peo	`http://www.ontotext.com/proton-ontology`

Табела 4.8: Постоечки својства искористени при анотација

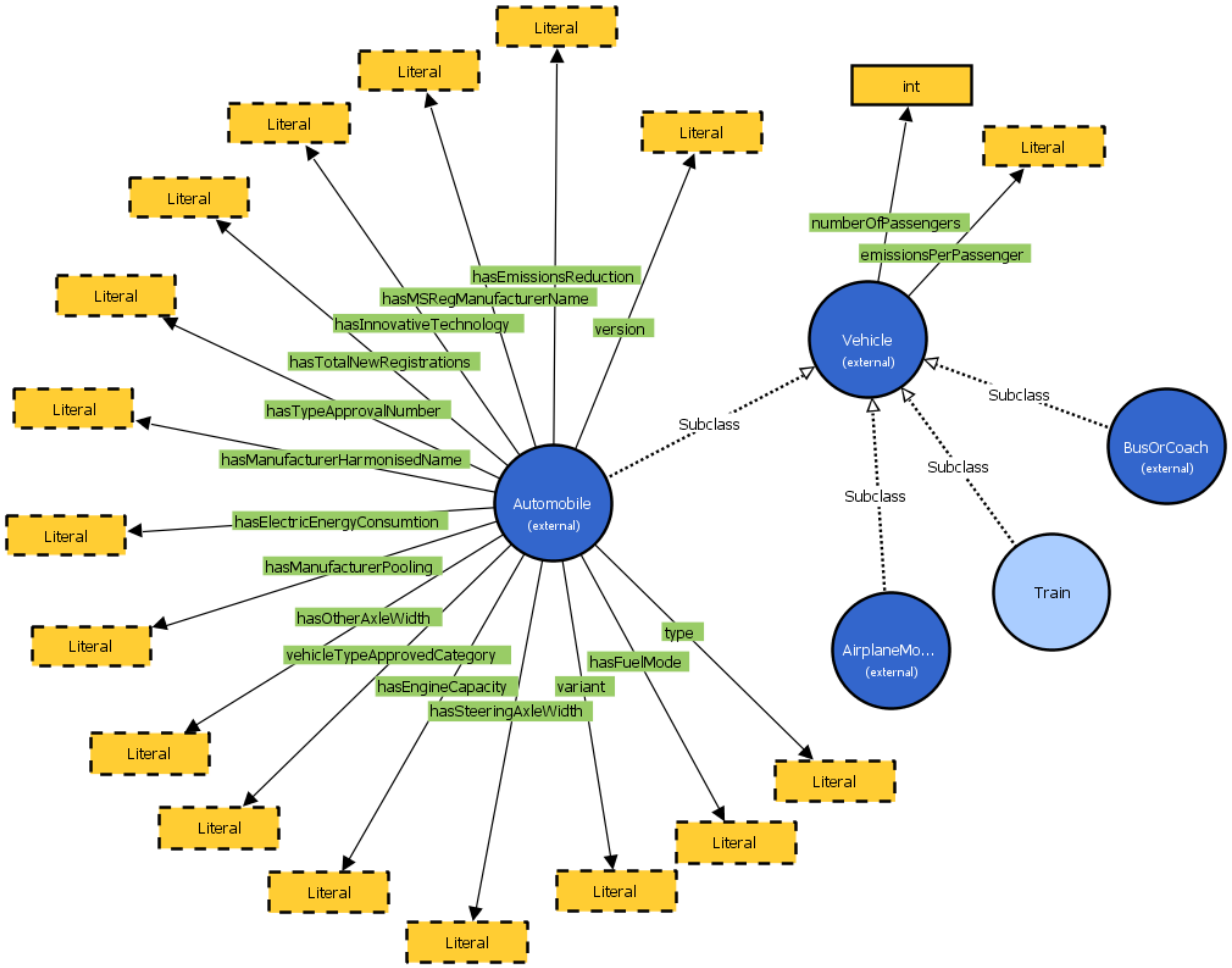
Својство	**Опис**
`gn:countryCode`	Код на држава согласно ISO 3166
`gr:hasManufacturer`	Производител на возилото
`gr:hasMakeAndModel`	Името на моделот на возилото
`vvo:marketingName`	Комерцијално име на возилото
`vvo:emissions`	CO_2 емисии, во g/km
`vso:weight`	Тежина на возилото без товар
`vso:wheelbase`	Должина на осовината
`vso:fuelType`	Типот на гориво
`vso:enginePower`	Моќност на моторот, во KWT

податоците од изворните податочни множества за кои не постојат својства во другите онтологии (Слика 4.8).

За да обезбедиме поддршка за подетални кориснички сценарија, дефинираме две својства кои не се мапираат директно на изворните податоци: `veo:numberOfPassengers` и `veo:emissionsPerPassenger`. Овие својства ги користиме за означување на информациите за просечни CO_2 емисии по патник, пресметани од наша страна врз база на изворните податоци.

VEO онтологијата е објавена и јавно достапна согласно принципите на поврзани податоци, на локација која поддржува HTTP content negotiation: `http://linkeddata.finki.ukim.mk/lod/ontology/veo#`.

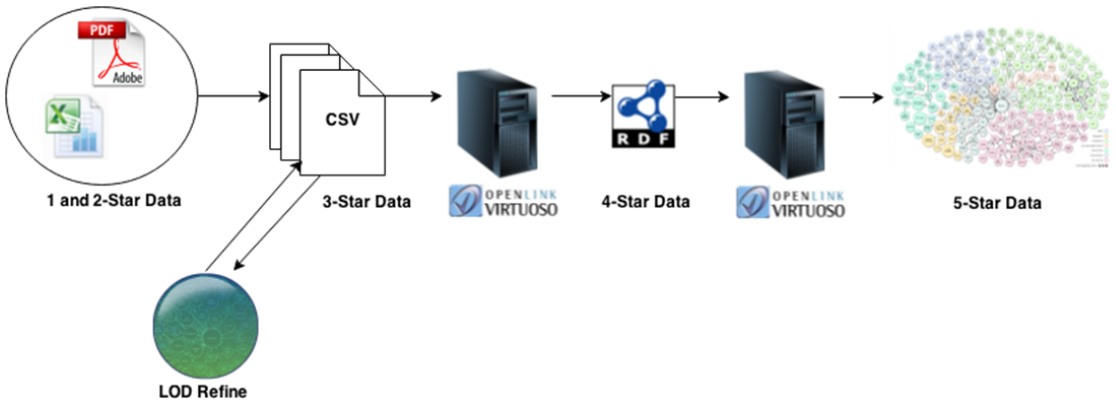
Креирање поврзани податоци. По собирањето на податоците од наведените извори, го започнавме процесот на нивно трансформирање во поврзани податоци. Прв чекор во работниот тек (Слика 4.9) беше користење на LODRefine [37] алатката со цел усогласување на податоците за возила од нашето податочно множество со податоци



Слика 4.8: VEO онтологијата и помошните класи од постоечките онтологии.

од DBpedia. Ваквото усогласување се однесува на наоѓање на ентитети на DBpedia кои се споменуваат во изворното податочно множество, со цел креирање линкови помеѓу нив и креирање на поврзано податочно множество. За поврзувањето користиме го `rdfs:seeAlso` својството за време на мапирањето во RDF формат, обезбедувајќи поврзаност на нашето податочно множество со LOD облакот.

Со цел да ги трансформираме и објавиме податоците како поврзано податочно множество, користиме инстанца од Virtuoso Universal Server. CSV датотеките кои се излез од LODRefine алатката се вчитуваат во Virtuoso, од каде понатаму со користење на R2RML мапирачкиот јазик се трансформираат во RDF формат, согласно онтологијата и шемата од Слика 4.8. Со оглед на тоа што работиме со четири независни CSV датотеки: за патнички автомобили, автобуси, возови и авиони; како резултат добиваме четири независни RDF графови, кои понатаму ги комбинираме во еден единствен RDF граф кој ги содржи сите потребни податоци.



Слика 4.9: Работен тек на трансформацијата на податоците.

Кориснички сценарија

Целта на трансформација на податоците во висококвалитетни, 5-star поврзани податоци беше овозможување на дополнителни сценарија во кои преку податоците од податочното множество пристапуваме до податоци од LOD облакот. Во продолжение ќе погледнеме две такви кориснички сценарија, кои работат со креираното поврзано податочно множество.

Прво корисничко сценарио. Во првото корисничко сценарија пристапуваме до податоци поврзани со CO_2 емисии на патнички автомобили произведени од производителот 'Mazda', при што ги бараме петте автомобили со најголема количина CO_2 емисии, заедно со нивните детали:

SPARQL прашање 4.6

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX veo: <http://purl.org/net/veo#>
PREFIX vvo: <http://purl.org/vvo/ns#>

SELECT DISTINCT ?cName ?em ?pass ?emPerPass
FROM <http://purl.org/net/veo/data#>
WHERE {
   ?veoCar rdf:type vso:Automobile ;
           veo:hasMSRegManufacturerName ‘‘MAZDA’’ ;
           vvo:marketingName ?cName ;
           vvo:emissions ?em ;
           veo:numberOfPassengers ?pass ;
           veo:emissionsPerPassenger ?emPerPass .
}
ORDER BY DESC (?emissions)
LIMIT 5
```

Резултатите од извршувањето на прашањето се прикажани во Табела 4.9. Од нив мо-

жеме да заклучиме дека двата модели на патничкиот автомобил “Mazda” имаат вкупни CO_2 емисии од 243 g/m, односно по 48,6 g/m по патник. Трите останати модели имаат CO_2 емисии од 224 g/m, односно 44,8 g/m по патник.

Табела 4.9: Резултати од SPARQL прашањето

Модел	**CO_2 емисии (g/m)**	**Патници**	**Ем. по патник**
MAZDA CX-7	243	5	48,6
CX-7	243	5	48,6
MAZDA3/SP/2.3I/MPS	224	5	44,8
MAZDA3/SP/2.3I/MPS SPEZIAL	224	5	44,8
3 MPS	224	5	44,8

Второ корисничко сценарио. Во второто сценарио демонстрираме користење на податоци од DBpedia за патничкото возило “Audi Q7” од нашето податочно множество. Ваквиот тип прашања се овозможени благодарение на релациите додадени во нашето податочно множество кои ги поврзуваат ентитетите од него со ентитети од LOD облакот. Со користење на концептот на SPARQL здружени прашања, започнувајќи од нашиот SPARQL endpoint можеме имплицитно да упатиме прашање до SPARQL endpoint-от на DBpedia и да ги добиеме бараните податоци. Ваквото SPARQL прашање е дадено во продолжение:

SPARQL прашање 4.7

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX vvo: <http://purl.org/vvo/ns#>

SELECT ?abstract ?links ?photos where {
   GRAPH <http://purl.org/net/veo/data#> {
      ?veoCar vvo:marketingName ‘‘AUDI Q7’’ ;
              rdfs:seeAlso ?DBcar .
   }
   SERVICE <http://dbpedia.org/sparql> {
      ?DBcar dbpedia-owl:abstract ?abstract ;
             dbpedia-owl:wikiPageExternalLink ?links ;
             dbpprop:hasPhotoCollection ?photos .
      FILTER langMatches( lang(?abstract), ’en’)
   }
}
LIMIT 1
```

Резултатите од извршувањето на прашањето се прикажани во Табела 4.10. Во овие резултати апстрактот, дополнителните линкови и фотогалеријата се податоци кои не се дел од нашето поврзано податочно множество - овие податоци се добиваат пребарувајќи го нашето податочно множество, продолжувајќи преку постоечките линкови до DBpedia и добивање на податоците од податочното множество на DBpedia.

Табела 4.10: Резултати од SPARQL прашањето

Апстракт	**Линкови**	**Фотографии**
"The Audi Q7 is a full-size luxury crossover SUV unveiled in September 2005 at the Frankfurt Motor Show. Production of the Q7 began in autumn of 2005 in Bratislava, Slovakia..."	http://www.audi.co.uk/audi/uk/en2/new_cars/q7.html	http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/photos/Audi_Q7

4.2.4 Поврзани податоци за аерозагадувањето во Скопје

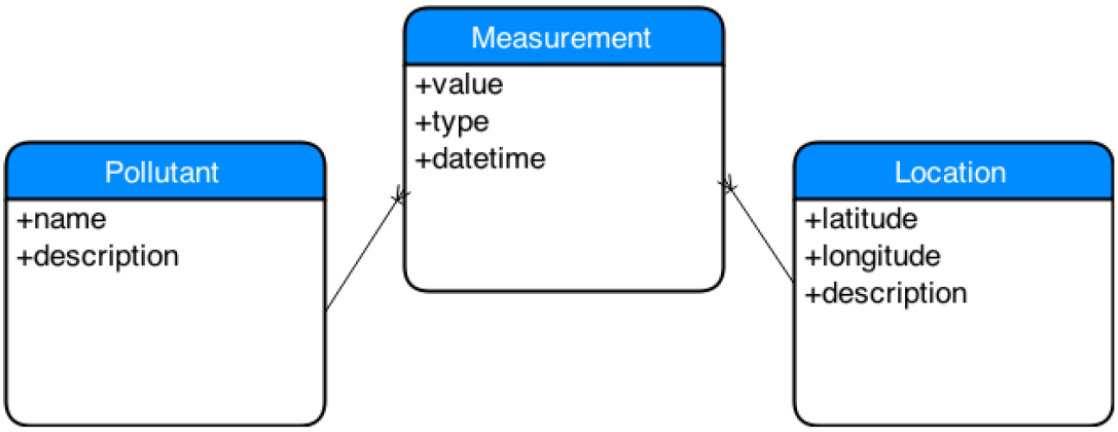
Аерозагадувањето е голем глобален проблем, но проблем присутен и во Македонија, особено во главниот град Скопје. Скопје има проблем со висока концентрација на PM2.5 и PM10 честички во одредени периоди во годината. Ова се должи на високата населеност на градот и неговите географски карактеристики, кои придонесуваат на зголемување на количеството гасови во воздухот, но и прашина и честички од различни извори [82]. Како четврто истражување во оваа подобласт, дизајниравме систем за централизирано собирање на податоци за аерозагадувањето во Скопје од повеќе мерни станици, интерполација на податоците со постоечки еко модели со цел одредување на нивото на квалитет на воздух во просторот помеѓу мерните станици, како и трансформација на податоците во 4-star и 5-star поврзани податоци со користење на постоечки онтологии од доменот. Како корисничко сценарио, развивме веб апликација која го користи поврзаното податочно множество и генерира топлотна мапа за нивоата на загаденост во регионот на градот Скопје [157].

Трансформација на податоците

Податоци за аерозагадување. Постојат повеќе мерни станици за квалитетот на воздухот дистрибуирани во регионот на градот Скопје, кои обезбедуваат множество индикатори. Најголемиот број од овие станици обезбедуваат REST-базиран сервис за пристап до мерењата во реално време. Во рамките на нашето истражување, ги користиме сервисите од Министерството за животна средина и просторно планирање, мерните станици на Лабораторијата за еко-информатика во рамките на Факултетот за информатички науки и компјутерско инженерство (ФИНКИ) во Скопје и мерните CO_2 станици од 'Skopje Green Route' проектот.

Мерните станици од Министерството и лабораторијата за еко-информатика обезбедуваат мерења за CO, NO_2, PM2.5, PM10, SO_2 и O_3. Овие мерења се достапни преку REST-базиран сервис кој обезбедува ажурирани податоци на секој час во текот на денот. Овие мерни станици се распределени во општина Центар, општина Карпош, населбата Лисиче, општина Гази Баба и во реонот на Ректоратот на Универзитетот „Св. Кирил и Методиј“. Мерните станици од 'Skopje Green Route' проектот се поставени на најфреквентните крстосници во градот Скопје: кај Судска палата, кај Црвен Крст и кај Земјоделскиот факултет. Тие обезбедуваат информации за CO_2 нивото на секои 5 минути во текот на денот.

Со користење на REST-базираните сервиси на наведените мерни станици, податоците ги собираме на одредени временски интервали и ги зачувуваме во локална база на податоци, притоа додавајќи им временски жиг. За таа цел користиме MySQL база (Слика 4.10).



Слика 4.10: EA дијаграм на базата на податоци.

Проширување на податоците со интерполација. Процесот на интерполација се базира на најновите податоци од мерните станици, кои го претставуваат просекот од концентрацијата на соодветниот параметар во последниот час. За интерполација користевме модел кој зависи од временските услови и структурата на областа, имплементиран во ArcGIS Simulation Server инстанца. Централно собраните податоци од мерните станици се испраќаат до серверот, кој со помош на моделот генерира интерполирани вредности за секој од параметрите во регионот на градот Скопје, помеѓу самите мерни станици. Излезот од процесот е слика, поради што имплементиравме алгоритам за трансформација на податоците од сликата во нумерички формат. Интерполираните и трансформирани нумерички вредности за количеството на секој од параметрите на сите локации низ градот Скопје се зачувуваат назад во базата на податоци со соодветен временски жиг и придружени со информации за моменталната временска состојба.

Онтологии за анотација. Следејќи ги најдобрите практики и препораки за користење на онтологии, направивме истражување на онтологиите од доменот и увидовме дека PESCaDO онтологијата [173] најдобро одговара за нашето податочно множество.

Онтологијата е развиена со цел да обезбеди мапирање на податоци од мерења на PM10, PM2.5, CO и други индикатори, како и мапирање и на временски услови [110]. Од онтологијата ги искористивме податочните својства прикажани на Табела 4.11 и Табела 4.12.

Табела 4.11: Податочни својства за временски услови од PESCaDO онтологијата

Својство	**Опис**
`pescado:HumidityValue`	Влажност на воздухот
`pescado:TemperatureValue`	Температура на воздухот
`pescado:WindSpeedValue`	Брзина на ветер

Табела 4.12: Податочни својства за аерозагадување од PESCaDO онтологијата

Својство	**Опис**
`pescado:PM10IndexValue`	Количина на PM10 честички
`pescado:PM2.5IndexValue`	Количина на PM2.5 честички
`pescado:COIndexValue`	Концентрација на CO гас
`pescado:NO2IndexValue`	Концентрација на NO_2 гас
`pescado:SO2IndexValue`	Концентрација на SO_2 гас
`pescado:O3IndexValue`	Концентрација на O_3 гас

За мапирање на географските локации на измерените и интерполираните вредности, го искористивме Basic Geo вокабуларот од W3C [96] со неговата класа `geo:Point` и својствата `geo:lat`, `geo:long` и `geo:location`. За секоја од измерените вредности имаме прецизна гео-локација, преку локацијата на самата мерна станица, па секоја од овие локации се анотира со `geo:lat` и `geo:long` својствата. За анотација на интерполираните вредности, ја поделивме територијата на град Скопје на зони, секоја со свои гео-координати и интерполирани вредности. Овие координати на секоја од зоните ги анотираме со истите својства од Geospatial вокабуларот.

Креирање поврзани податоци. За трансформација на централно собраните и интерполираните податоци за квалитетот на воздухот во Скопје во 5-star поврзани податоци, искористивме D2RQ сервер инстанца. D2RQ серверот се поврзува со базата на податоци во која се наоѓаат нашите податоци и преку дефинирање на соодветна мапирачка датотека, обезбедува RDF-базиран пристап до податоците. Мапирачката датотека ја дефинираме на начин кој обезбеди анотација на податоците од базата на податоци во RDF податоци кои ги користат PESCaDO онтологијата и Basic Geo вокабуларот. При мапирањето го користиме својството `geo:location` за да ги поврземе локациите од податочното множество со инстанцата на градот Скопје на DBpedia, односно во LOD облакот. Со тоа, обезбедуваме поврзување на податочното множество со LOD облакот и

добиваме податоци со 5-star квалитет. Дополнително, со ова демонстрираме и начин на кој мерења од други региони можат да се поврзат со соодветните локации од LOD облакот, со цел проширување на глобалното множество на поврзани податоци со податоци за квалитет на воздухот.

Кориснички сценарија

Со цел демонстрација на предноста од собирањето на податоците за квалитетот на воздухот во Скопје од повеќе извори, нивна интерполација и трансформација во поврзани податоци, креиравме веб апликација која овозможува преглед на регионот на градот Скопје преку т.н. топлотни мапи. Овие мапи овозможуваат брз преглед на концентрацијата на одредени честички или гасови на територијата на Скопје, благодарение на измерените и интерполираните вредности.

Имајќи ги податоците во формат на поврзани податоци, можеме да пристапуваме до нив со користење на SPARQL прашалниот јазик. На пример, доколку сакаме да ги добиеме измерените и интерполираните вредности за концентрацијата на CO гасот во воздухот, го одреден временски момент, можеме да го искористиме следново прашање:

SPARQL прашање 4.8

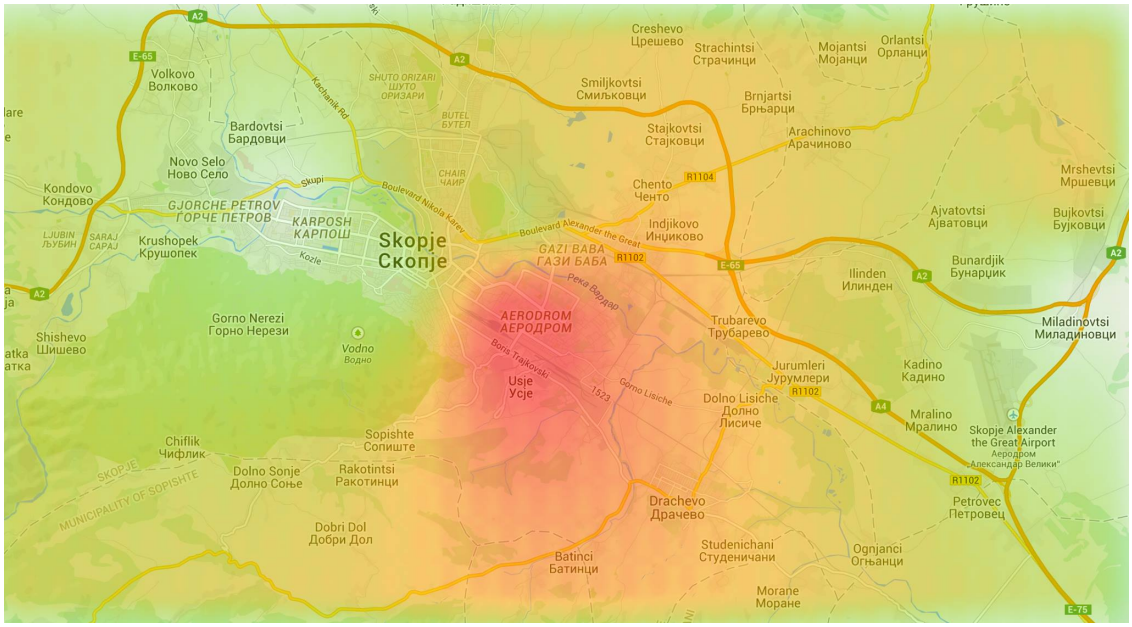
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pescado: <https://ontohub.org/fois-ontology-competition/
                              PESCaDO_Ontology/pescadoData.owl#>
PREFIX prov: <http://www.w3.org/ns/prov#>
SELECT DISTINCT ?lat ?lng ?value
WHERE {
   ?s rdf:type pescadoData:COIndexValue ;
      rdf:value ?value ;
      prov:atLocation ?location ;
      prov:generatedAtTime "2015-03-03T19:15:46"^^xsd:dateTime .
   ?location geo:lat ?lat ;
             geo:lng ?lng .
}
```

Резултатот од SPARQL прашањето ќе бидат вредностите на CO концентрација на локации во градот Скопје. Дел од добиените резултати се прикажани во Табела 4.13. Овие резултати можат да се искористат и како влез за генерирање топлотна мапа која би ги прикажала нивоата на CO во воздухот на територијата на Скопје. Еден пример приказ на овие резултати е даден на Слика 4.11. Пример од топлотна мапа со нивото на PM10 честички на територијата на Скопје е даден на Слика 4.12.

Поврзаното податочно множество со измерени и интерполирани вредности на различните мерки за квалитетот на воздухот во градот Скопје се достапни преку SPARQL endpoint, кој може да се користи и како REST-базиран сервис, на `http://airpollution.b1.finki.ukim.mk/`.

Табела 4.13: Резултати од SPARQL прашањето

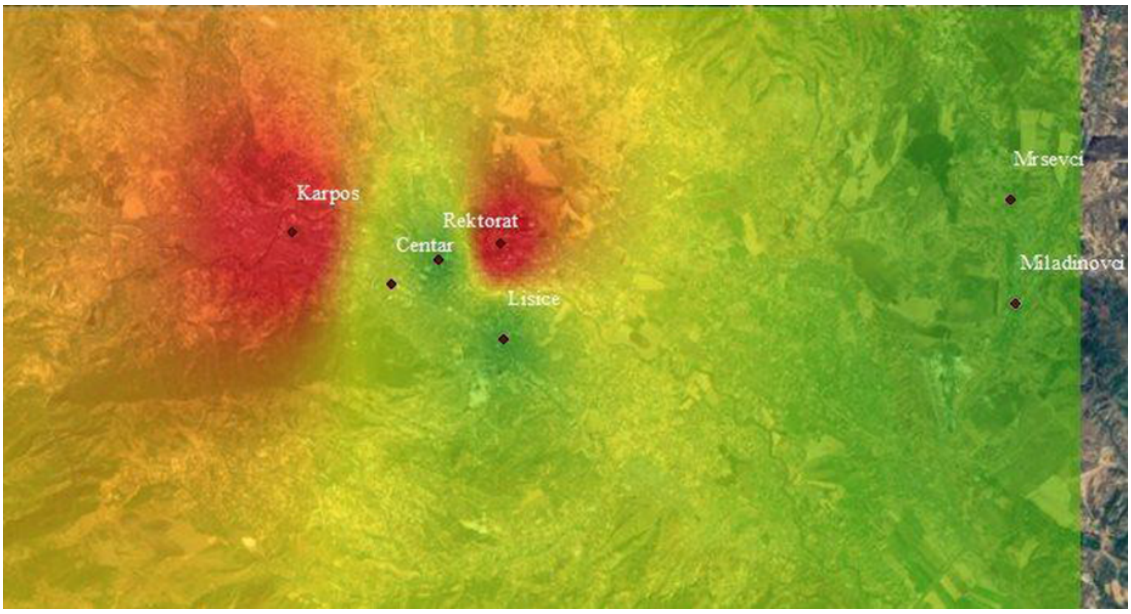
Геогр. ширина	Геогр. должина	Вредност
“42.05”^^xsd:float	“21.32”^^xsd:float	0.30
“41.96”^^xsd:float	“21.31”^^xsd:float	0.33
“41.94”^^xsd:float	“21.29”^^xsd:float	0.37
“42.03”^^xsd:float	“21.30”^^xsd:float	0.31



Слика 4.11: Топлотна мапа за количеството на CO гасот на територија на градот Скопје, на ден 03.03.2015, околу 19:00 часот.

4.2.5 Дискусија

Во рамките на доменот на јавен транспорт и аерозагадување реализиравме четири истражувања. Првото истражување [158] се однесуваше на податоци од јавниот транспорт во Република Македонија, поточно на податоците од јавното сообраќајно претпријатие ЈСП Скопје. Во рамките на истражувањето, изворните податоци од веб страната на претпријатието ги трансформиравме во отворени податоци во RDF формат. Резултантното отворено податочно множество има 6.005.619 RDF тројки: 5.227.956 тројки се наоѓаат во графот `stop_times`, 725.758 во графот `trips`, 46.914 во графот `shapes`, 4.649 во `stops`, 170 во `routes`, 165 во `calendar` и само 7 во `agency`. Врз база на ова податочно множество дефинираме пример кориснички сценарија, со цел да ги мотивираме заинтересираните чинители од доменот да започнат со објава и користење на отворени податоци од доменот на јавен транспорт. Резултантното отворено податочно множество е достапно преку јавен SPARQL endpoint, кој може да се користи како REST-базиран сервис.



Слика 4.12: Топлотна мапа за количеството на PM10 честички на територија на градот Скопје, на ден 03.03.2015, околу 19:00 часот.

Во второто истражување [164] развивме автоматизиран систем за собирање, трансформација и објавување на висококвалитетни поврзани податоци од транспортниот домен, овој пат со користење на податоците од Шведската транспортна администрација како извор. Изворните податоци ги анотиравме со нашата нова, TAO онтологија. Со цел да ги демонстрираме предностите од постоење на податоци од овој домен како поврзани податоци, креиравме веб апликација [58] која на корисниците им нуди серија кориснички сценарија, притоа користејќи го единствено нашето поврзано податочно множество како податочно ниво. Со помош на линковите кои се дел од самото поврзано податочно множество, кон ентитети од LOD облакот, понудените кориснички сценарија се пошироки и понапредни, отколку оние над изолираното изворно податочно множество. Резултантното податочно множество исто така е достапно преку јавен SPARQL endpoint.

Во третото истражување [162] од доменот, работевме со податоци за CO_2 емисии од возила, објавени од страна на Европската агенција за животна средина и неколку други извори. Овие изворни податоци ги трансформиравме во висококвалитетни поврзани податоци, со користење на нашата нова, VEO онтологија. Како и во другите истражувања, врз генерираното поврзано податочно множество дефиниравме и демонстриравме неколку кориснички сценарија со кои може да се увиди предноста од постоење на вакви податоци во формат на поврзани податоци. Резултантното податочно множество од ова истражување исто така е достапно преку јавен SPARQL endpoint.

Во четвртото истражување [157] реализиравме систем за мерење и објава на податоците за квалитетот на воздухот во Скопје, во вид на отворени и поврзани податоци. Со помош на 7 мерни станици за квалитет на воздух и 3 мерни станици за CO_2, систе-

мот собира информации за количината на CO_2, CO, NO_2, PM2.5, PM10, SO_2 и O_3 на соодветните локации, во одредени временски интервали. Над овие податоци, системот потоа прави интерполација, со цел добивање на средни вредности за секоја од мерките на просторот помеѓу мерните станици. Сумарно, овие резултати се зачувуваат во рамки на релациона база на податоци, над која поставивме D2R Server инстанца. Оваа инстанца обезбедува пристап над податоците во RDF формат, во вид на поврзани податоци. На тој начин, добиваме пристап до податоците преку SPARQL endpoint кој може да се користи како REST-базиран сервис. Над отвореното податочно множество и достапниот SPARQL endpoint, демонстрираме низа кориснички сценарија кои имаат за цел да покажат како овие податоци можат да се искористат во кориснички апликации кои би работеле со податоци за аерозагадување во нашиот главен град.

4.3 Поврзани финансиски податоци од Македонската берза и Светска Банка

Во областа на финансиски податоци, работевме на истражување во кое ги собравме, трансформиравме и консолидиравме податоците од Македонската берза, веб страните на големите македонски компании и од Светската банка. Податоците ги трансформиравме во 4-star и 5-star поврзани податоци, кои се објавени и достапни согласно принципите на поврзаните податоци и кои овозможуваат низа нови кориснички сценарија претходно недостапни над изолираните изворни податочни множества [163].

4.3.1 Трансформација на податоците

Финансиски податоци. Македонската берза [39] е единствената финансиска институција во Република Македонија која е авторизирана а организира, извршува и регулира тргување со хартии од вредност. Формирана е во 1995 година како акционерско друштво кое работи на непрофитна основа со цел да обезбеди ефикасно, транспарентно и сигурно функционирање на организираниот секундарен пазар на хартии од вредност во Македонија. Податоците со кои располага Македонската берза се објавуваат на нејзината веб страна во вид на PDF документи и HTML табели. Од објавените податоци, кои вклучуваат цени на акции, различни индекси, информации за трендови на раст на хартии од вредност, итн., наш интерес за истражувањето беа финансиските извештаи за компаниите членки на берзата. Податоците ги земавме од веб страната, ги трансформиравме во соодветен CSV формат, со цел да ги доведеме до ист формат.

Дополнително, направивме анализа за податоците достапни на официјалните веб страни на поголемите компании на територијата на Република Македонија, од каде заклучивме дека повеќето од нив имаат значајна количина податоци од финансиски извештаи. Поради тоа, одлучивме да ги собереме и овие податоци и да ги трансформираме во униформен CSV формат.

Светската Банка, исто така, објавува податоци од јавен карактер на својата веб

страна. Овие податоци вклучуваат и финансиски податоци кои беа од интерес во нашето истражување [71]. Овие податоци беа во различни формати: CSV, JSON, PDF, RDF, RSS, XLS, XLSX и XML. Дел од податоците се директно достапни преку нивниот јавен SPARQL endpoint [73]. Податочните множества кои беа од интерес за истражувањето беа податоците за финансиските заеми од Интернационалната банка за обнова и развој (International Bank for Reconstruction and Development, IBRD) и кредитите од Меѓународната асоцијација за развој (International Development Association, IDA) [72]. Овие податочни множества се достапни директно во RDF формат.

Онтологии за анотација. Податоците од Светската Банка, кои беа достапни во RDF формат и не бараа дополнителни трансформации, ги вчитавме директно во Virtuoso инстанца. Податочното множество се состои од ентитети кои репрезентираат заеми од IBRD и кредити од IDA, со детали за компанијата која ги добила, државата во која таа се наоѓа, висината на заемот или кредитот, итн.

За анотација на податоците собрани од веб страните на македонските компании и од Македонската берза, потребна беше соодветна онтологија. За таа цел, ги анализиравме онтологиите кои се користат од страна на Open Corporates иницијативата [43], која содржи податоци за компаниите регистрирани во голем број држави во светот - во кои, за жал, Република Македонија не е вбројана - во XML, RDF и JSON формат. Онтологиите кои ги селектиравме за опис на податоците за македонските компании и правни лица се дадени во Табела 4.14. Ја користиме класата `rov:RegisteredOrganization` за анотација на регистрирани правни лица и компании кои се среќаваат во собраното податочно множество. За секоја инстанца од оваа класа поврзуваме RDF тројки користејќи ги податочните својства прикажани во Табела 4.15.

Табела 4.14: Постоечки онтологии искористени при анотација

Онтологија	**Префикс**	**URI**
Registered Organization Vocabulary	rov	`http://www.w3.org/ns/regorg#`
Asset Description Metadata Schema	adms	`http://www.w3.org/ns/adms#`
Friend of a Friend	foaf	`http://xmlns.com/foaf/0.1/`
vCard Ontology	vCard	`http://www.w3.org/2006/vcard/ns#`
Simple Knowledge Organization System	skos	`http://www.w3.org/2004/02/skos/core#`

За опис на годишните финансиски извештаи од Македонската берза, особено на билансите на компаниите кои се дел од берзата, беше потребна дополнителна онтологија. Анализата на постоечките онтологии од доменот покажа дека Financial Report Ontology [19] најдобро одговара на нашите потреби, со оглед на тоа што ги опишува основните поими од областа на финансиски извештаи. Во неа е дефинирана класата

Табела 4.15: Постоечки својства искористени при анотација

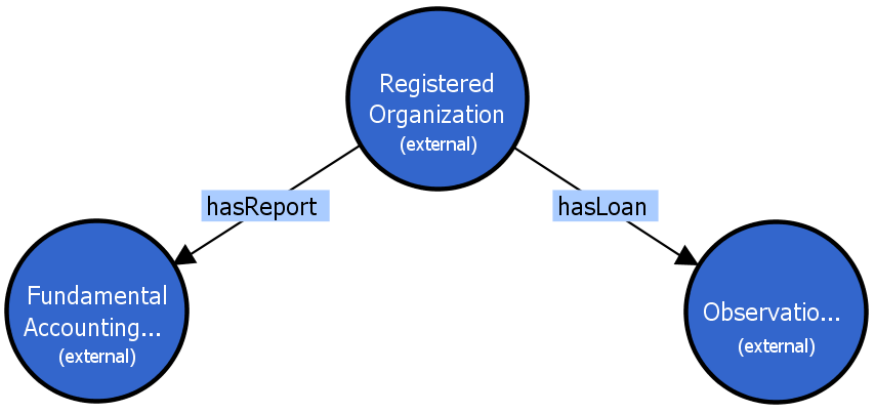
Својство	Опис
rov:legalName	Регистрираното име на компанијата
rov:registration	Се користи за релации помеѓу правни лица и телото каде тие се регистрирани.
vCard:extended-address	Адресата на ентитетот
vCard:hasTelephone	Телефонскиот број на ентитетот
skos:notation	Име на компанијата, кое може да се пишува различно од регистрираното име
foaf:homepage	Веб страна на ентитетот
rdfs:label	Информации за основните активности на компанијата

fac:FundamentalAccountingConcept која служи за опис на финансиски извештај, како и својства поделени во пет групи: генерални информации, својства за биланс на состојба, за биланс на приливи, за сеопфатна добивка и за парични текови. За нашето податочно множество ги користиме својствата кои се однесуваат на генерални информации, биланс на состојба и биланс на приливи.

Одвоените податочни множества беше потребно да се поврзат. За таа цел имавме потреба од дополнителни својства кои ќе ги поврзат компаниите како ентитети со нивните заеми или кредити од програмите на Светска банка, како и со нивните финансиски извештаи. Анализата на постоечките онтологии и вокабулари покажа дека такви својства не се дефинирани. Поради тоа, креиравме нова онтологија, Corporate Financial Reports and Loans Ontology (CFRL), во која дефинираме две својства: cfrl:hasReport и cfrl:hasLoan. Првото својство го искористивме за креирање RDF тројки во кои ентитет кој репрезентира компанија или правно лице беше поврзан со ентитет кој репрезентира негов финансиски извештај. Второто својство, пак, го искористивме за креирање RDF тројки во кои компаниите и правните лица беа поврзани со нивните заеми од IBRD или кредити од IDA. CFRL онтологијата е објавена согласно најдобрите практики и препораки [88], односно преку перзистентно URI кое поддржува HTTP content negotiation: http://linkeddata.finki.ukim.mk/lod/ontology/cfrl#.

Поврзување на податоците. Целта на проектот беше поврзување на компаниите и нивните податоци, собрани од официјалните веб страни на компаниите, поединечно, со заемите и кредитите од програмите на Светска Банка и со финансиските извештаи од Македонската берза (Слика 4.13).

Следниот чекор во постапката беше мапирање и трансформација на CSV податочните множества во RDF. За таа цел искористивме инстанца од Virtuoso Universal Server во која слично како и во претходните проекти употребивме мапирање базирано на R2RML мапирачкиот јазик. Со оваа постапка, користејќи ги наведените онтологиите, ги тран-



Слика 4.13: Поврзување на ентитетите од трите податочни множества.

сформиравме податочните множества со податоци за компаниите регистрирани во Република Македонија, како и нивните финансиски извештаи од Македонска берза. Податочното множество за заеми и кредити од програмите на Светска Банка веќе беа достапни во RDF формат, па нив директно ги вчитавме во Virtuoso инстанцата.

По трансформацијата, резултантните RDF податочни множества беа вчитани во еден заеднички RDF граф. Со цел креирање на 5-star поврзани податоци, ги искористивме дизајнираните својства од CFRL онтологијата за да ги повземе ентитетите од различните податочни множества. Притоа, својството `cfrl:hasReport` го искористивме за поврзување на инстанците од класата `rov:RegisteredOgranization` со инстанци од класата `fac:FundamentalAccountingConcept`, користејќи ги имињата на компаниите и правните лица за нивно поврзување со соодветните финансиски извештаи. Имињата на компаниите кај `rov:RegisteredOgranization` ентитетите се поврзани со релацијата `skos:notation`, додека имињата на компаниите кај ентитетите од класата `fac:FundamentalAccountingConcept` со релацијата `fac:EntityRegistrantName`. Својството `cfrl:hasLoan` го искористивме за креирање RDF тројки помеѓу инстанци од `rov:RegisteredOgranization` со соодветните заеми и кредити на компанијата. Слично како и со претходното својство, за креирање на RDF тројките ги искористивме имињата на компаниите, означени со `rov:legalName` својствата на `rov:RegisteredOrganization` ентитетите и вредностите на `worldbank:supplier` својството на заемите и кредитите.

Поврзувањата ги реализиравме преку SPARQL INSERT прашања над RDF графот во кој беа вчитани трите податочни множества. Резултантното поврзано податочно множество е достапно на Веб преку јавен SPARQL endpoint [57].

4.3.2 Кориснички сценарија

Целта на трансформацијата на трите податочни множества во RDF и креирање на поврзано податочно множество беше зголемување на вредноста на изворните податоци преку овозможување нови, понапредни кориснички сценарија, кои инаку не се достапни над изолираните изворни податочни множества. Во продолжение ќе погледнеме две

такви сценарија.

Прво корисничко сценарио. Генерираните RDF тројки со `cfrl:hasLoan` својство можеме да го искористиме за пристап до информации за компанија која добила заем или кредит од програмите на Светската Банка. Овие информации ја вклучуваат вкупната сума во на заемот или кредитот, датумот кога е потпишан договор за негово земање од страна на македонската компанија и Светска Банка, како и секторот во кој е регистрирана за дејност компанијата на територија на Република Македонија. Едно пример SPARQL прашање во кое ги селектираме петте заеми и кредити со највисока вредност, е дадено во продолжение.

SPARQL прашање 4.9

```
PREFIX cfrl: <http://linkeddata.finki.ukim.mk/lod/ontology/cfrl#>
PREFIX worldbank: <http://finances.worldbank.org/resource/>
PREFIX rov: <http://www.w3.org/ns/regorg#>

SELECT ?company ?date ?sum ?sector
WHERE {
   ?comEntity rov:legalName ?company .
   ?company cfrl:hasLoan ?loan .
   ?loan worldbank:contract_signing_date ?date ;
         worldbank:supplier_contract_amount_usd ?sum ;
         worldbank:major_sector ?sector .
}
ORDER BY DESC (?sum)
LIMIT 5
```

Резултатите од извршувањето на прашањето над нашето податочно множество се прикажани во Табела 4.16.

Табела 4.16: Резултати од SPARQL прашањето

Компанија	**Датум**	**Сума**	**Сектор**
Granit	Mar 26, 2009	$9,802,524.00	Transportation
Granit	Dec 04, 2009	$6,197,108.00	Transportation
Granit	Dec 04, 2009	$5,323,028.00	Transportation
Granit	Mar 26, 2009	$4,519,095.00	Transportation
Granit	Dec 04, 2009	$3,785,761.00	Transportation

Второ корисничко сценарио. Во дополнително креираните RDF тројки, со кои ги поврзавме трите податочно множества, го искористивме и својството `cfrl:hasReport` за поврзување на компаниите со нивните финансиски извештаи од Македонската берза. Користејќи ги овие RDF тројки, можеме да добиеме информации за петте компании со

најголем профит за одредена година. Едно такво пример SPARQL прашање е дадено во продолжение.

SPARQL прашање 4.10

```
PREFIX cfrl: <http://linkeddata.finki.ukim.mk/lod/ontology/cfrl#>
PREFIX fac: <http://www.xbrlsite.com/2013/FinancialReportOntology/
                    Prototype04/FundamentalAccountingConcepts.xml#>
PREFIX rov: <http://www.w3.org/ns/regorg#/>

SELECT ?name ?profit ?period
WHERE {
   ?company cfrl:hasReport ?report ;
            rov:legalName ?name .
   ?report fac:GrossProfit ?profit ;
           fac:FiscalPeriod ?period .
   FILTER (?period = 2012)
}
ORDER BY ?profit
LIMIT 5
```

Резултатот од извршување на прашањето над нашето поврзано податочно множество, во кое се прикажани петте компании со највисок просек за 2012 година, е прикажан на Табела 4.17.

Табела 4.17: Резултати од SPARQL прашањето

Компанија	**Профит (МКД)**	**Период**
ALKALOID AD SKOPJE	3,291,423	2012
Stopanska Banka AD Skopje	2,376,477	2012
Tikvesh AD Skopje	339,049	2012
GD GRANIT AD - Skopje	291,238	2012
Vitaminka AD Prilep	102,378	2012

4.3.3 Дискусија

Целта на ова истражување беше примена на концептите на поврзани податоци во доменот на финансии во Република Македонија. Со оглед на тоа што организирани податоци за компаниите од Македонија не постојат, нашиот тим мораше да собира основни податоци за компаниите поединечно, од нивните официјални веб страни. Со тоа, успеавме да креираме податочно множество со значајни информации за големите македонски компании. Дополнително, користејќи ги податоците достапни од официјалната

веб страна на Македонската берза, креиравме уште едно податочно множество со финансиски извештаи на македонските компании кои учествуваат на берзата. Како трето податочно множество во истражувањето го искористивме постоечкото RDF податочно множество од Светска Банка за заемите и кредитите на компаниите од Република Македонија, добиени од програмите IBRD и IDA.

Трите податочни множества ги анотиравме и трансформиравме во RDF податочни множества, а потоа ги поврзавме креирајќи RDF тројки помеѓу нивните ентитети. Со тоа, креиравме поврзано податочно множество во областа на финансиите на македонските компании. Како поддршка за креираното поврзано множество презентиравме и две кориснички сценарија во кои се искористуваат податоците од различните податочни множества за добивање дополнителни информации, претходно недостапни над изолираните податочни множества.

4.4 Поврзани музички податоци од глобалните радио топ-листи

Имајќи предвид дека музиката претставува значаен дел од секојдневието на голем број луѓе низ целиот свет, разбирливо е зошто Вебот содржи голем број податоци од доменот на музиката. Како и останатите податоци достапни преку Вебот, музичките податоци се наменети за користење од страна на луѓето, па поради тоа се среќаваат во различни формати. Примената на принципите на поврзани податоци во доменот на музиката би можела да обезбеди нови, напредни кориснички сценарија. Па така, постоењето на поврзано податочно множество од музичкиот домен би можело да обезбеди преглед во динамиката на глобалните музички топ-листи, од аспект на застапените музички артисти, нивната географска распределеност по држави и континенти, застапеноста на жанрови по региони, итн.

Поради тоа, реализиравме истражување во кое работевме со 1-star и 2-star податоци за музички топ-листи од различни глобални радио станици како извор и преку специјализиран систем ги трансформиравме во висококвалитетни, 5-star поврзани податоци. Во рамките на истражувањето ја дизајниравме и Playlist онтологијата, како и веб апликација која ги прикажува предностите од вакво консолидирано и поврзано податочно множество [141]. Со оглед на тоа што генератор на самите музички топ-листи се слушателите, односно публиката, сметавме дека пристап до вакво податочно множество кое обезбедува дополнителни податоци за самите песни, артисти и изданија од повеќе различни извори, може да претставува добра основа за развој на апликации за истата таа публика.

4.4.1 Трансформација на податоците

За да го генерираме поврзаното податочно множество од областа на музиката, креиравме систем кој ги собира, трансформира, поврзува, објавува и ажурира податоците

од радио топ-листите, на глобално ниво. Системот се состои од неколку делови, кои го сочинуваат целиот работен тек. Работниот тек потоа може да се извршува на одредени временски интервали, со цел ажурирање на податоците.

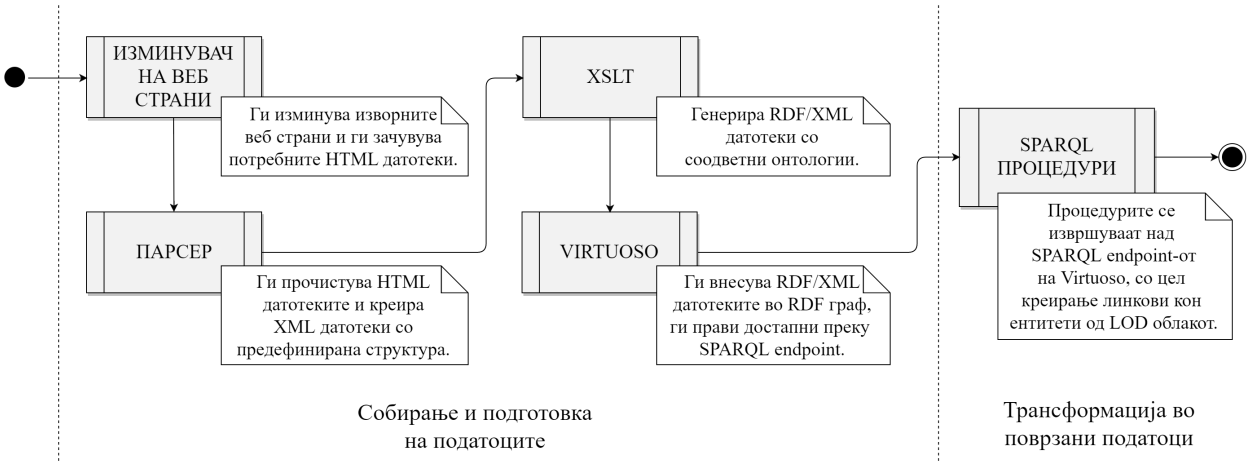
Како податочен извор ги користевме радио станиците од BBC: Radio 1, Radio 1Xtra, Radio 2, Radio 6 Music, Asian Network, Radio Scotland [5]. Од BBC Radio 1 ги користевме и поединечните официјални топ-листи. Иако содржината на официјалните веб страни на овие радио станици е од ист домен, структурата на податоците не е униформна. Поради тоа, беше потребно да употребиме сопствено софтверско решение кое ќе ги измине наведените веб страни и врз база на нивната структура ќе ги собере, прочисти и зачува потребните информации во униформен, XML формат. Помеѓу другите, овие податоци го содржат името на песната, изведувачот и тековната позиција на соодветната топ-листа. Податоците собрани во XML формат потоа ги трансформираме во RDF формат, го вчитуваме во RDF граф и ги поврзуваме со податочни множества од LOD облакот.

Автоматизираниот работен тек за генерирање на поврзано податочно множество од музичките радио топ-листи се состои од следниве чекори (Слика 4.14):

- Собирање и припрема на податоците
  - Користиме сопствено софтверско решение кое ги изминува изворните веб страни и ги собира соодветните податоци.
  - Со помош на парсер, собраните податоци се филтрираат и прочистуваат, пред да се снимат во XML формат.
  - Со XSL трансформација се преведуваат прочистените XML податоци во RDF (RDF/XML) формат.
  - RDF/XML датотеките се вчитуваат во RDF граф во Virtuoso инстанца.
- Трансформација во поврзани податоци
  - Извршуваме SPARQL-базирани процедури кои креираат линкови помеѓу ентитети од локалниот RDF град и соодветни ентитети од LOD облакот.

Собирање и припрема на податоците. Процесот на собирање на податоците го изведовме со развој на сопствено софтверско решение кое ги изминува изворните веб страни и ја снима нивната HTML содржина локално. Потоа, со посебни парсери направивме екстракција на потребните податоци - името на топ-листата, името на радио станицата, листа на песните на топ-листата заедно со нивните изведувачи и позиции на листата, итн. - и прочистените HTML документи ги зачувавме локално во XML формат.

Складираните XML податоци ги трансформираме со помош на XSL трансформација, која како резултат дава RDF податоци, во RDF/XML формат. Иако употребата на RDF/XML форматот е значително намалена во последните години поради големината на RDF/XML датотеките во споредба со другите RDF формати, сепак форматот најмногу одговара за употреба при XSL трансформации на изворни XML датотеки. XSL трансформацијата ја користи RDF шемата опишана подолу, во која XML елементите и атрибутите се трансформираат директно во RDF тројки репрезентирани со RDF/XML синтакса.



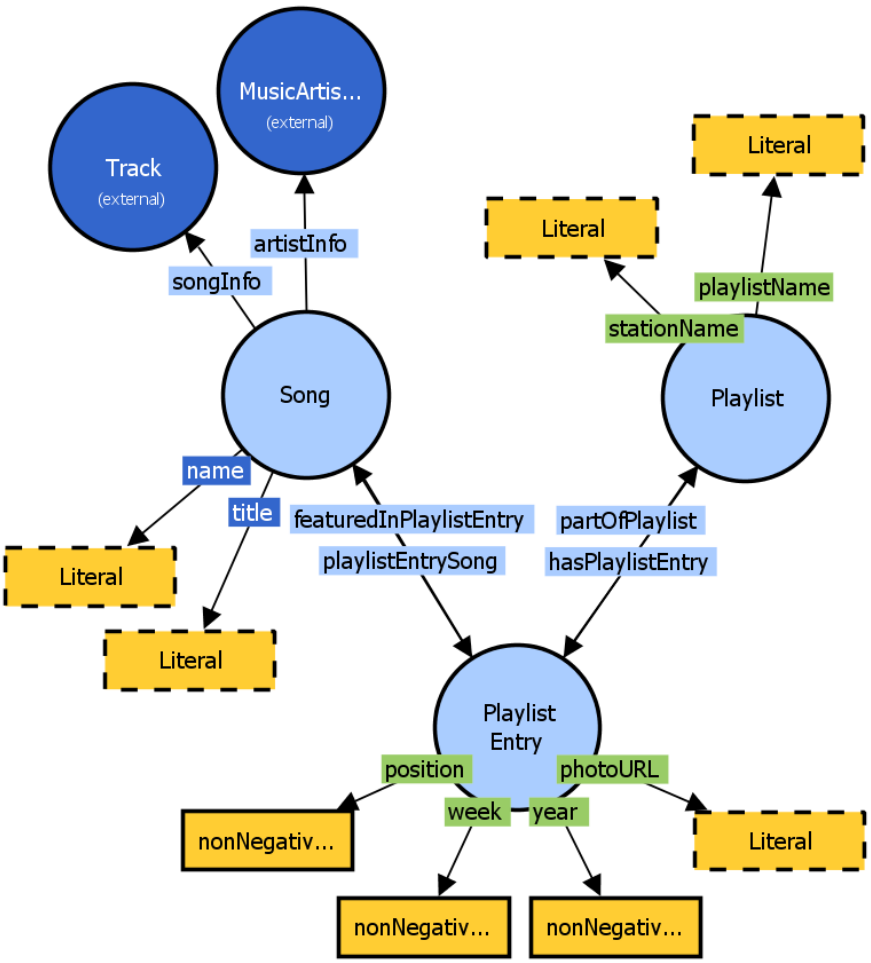
Слика 4.14: Работен тек на трансформацијата на податоците.

Резултантните RDF/XML датотеки се вчитуваат во Virtuoso инстанца, во еден заеднички RDF граф. При секое наредно извршување на автоматизираниот работен тек од Слика 4.14, податоците се додаваат во истиот RDF граф, односно се ажурира самото податочно множество. Овој RDF граф е објавен и јавно достапен преку перзистентно URI, кое поддржува HTTP content negotiation: `http://purl.org/net/lmd/data#`.

Playlist онтологија. Со цел анотација на податоците при трансформација од XML во RDF формат, беше потребна онтологија. Постоечките проекти од областа на поврзани податоци кои се однесуваат на музичкиот домен, како што се LinkedBrainz [34] и MusicBrainz [41], ја користат Music онтологијата [63]. Music онтологијата се користи за опишување на широк спектар информации поврзани со музика. Онтологијата обезбедува класи и концепти како што се артисти, изведувачи, албуми, песни, итн., како и својства за биографија, инструмент, времетраење, итн. [170].

Сепак, со оглед на тоа што ентитетите опишани во нашето податочно множество се записи од топ-листа, кои по својата природа се разликуваат од ентитетите кои се дел од LinkedBrainz и MusicBrainz, не бевме во можност директно да ги искористиме класите и својствата од Music онтологијата при анотација. Поради тоа, креиравме нова онтологија - Playlist Ontology (Слика 4.15). Онтологијата се состои од класи и својства кои се неопходни за опис и анотација на податоците од нашето податочно множество. Со цел да овозможиме поврзување на податоците од нашето податочно множество со податочни множества од LOD облакот, дизајниравме дополнителни објектни својства кои обезбедуваат механизми за креирање поврзувачки RDF тројки.

Playlist онтологијата дефинира три класи (Слика 4.15). Класата `po:PlaylistEntry` се користи за репрезентација на еден запис од музичка топ-листа. Еден ваков запис не е само песна, туку песна која се наоѓа на конкретна позиција во конкретна топ-листа, во конкретен временски момент. Класата `po:Playlist` се користи за претставување на една топ-листа од конкретна радио станица, додека класата `po:Song` се користи за репрезентација на една песна.



Слика 4.15: Playlist онтологијата и помошните класи од постоечките онтологии.

Овие класи се меѓусебно поврзани со четири објектни својства (Табела 4.18, Слика 4.15); `po:hasPlaylistEntry` и `po:playlistEntrySong` се главните својства во моделот, додека `po:partOfPlaylist` и `po:featuredInPlaylistEntry` се нивните инверзни својства, соодветно. Иако додавањето инверзни својства генерално внесува редундантност, т.е. се пишуваат RDF тројки кои ја носат истата информација, одлучивме да ги вклучиме во дизајнот на онтологијата со цел постигнување подобри перформанси при одредени SPARQL прашања, наменети за одредени кориснички сценарија. Класата`po:Song` користи две објектни својства за поврзување со инстанци од LOD облакот кои се анотирани со Music онтологијата: `po:artistInfo` и `po:songInfo` (Табела 4.18, Слика 4.15). Онтологијата користи и шест податочни својства (Табела 4.19, Слика 4.15). Дополнително, при анотација на податочното множество го користиме и `foaf:name` својството за дефинирање на името на артистот кој ја изведува `po:Song` песната, како и `dc:title` својството за дефинирање на насловот на `po:Song` песната (Табела 4.20, Слика 4.15).

Playlist онтологијата е објавена согласно најдобрите практики и препораки [88], односно преку перзистентно URI кое поддржува HTTP content negotiation: `http://purl.org/net/po#`.

Табела 4.18: Објектни својства од Playlist онтологијата

Својство	Опис
po:hasPlaylistEntry	Се користи за поврзување на po:Playlist инстанца со po:PlaylistEntry инстанци, за означување на елементите кои се дел од топ-листата. Инверзно својство на po:partOfPlaylist.
po:partOfPlaylist	Се користи за поврзување на po:PlaylistEntry инстанца со po:Playlist инстанца. Инверзно својство на po:hasPlaylistEntry.
po:playlistEntrySong	Се користи за поврзување на po:PlaylistEntry инстанца со po:Song инстанца. Инверзно својство на po:featuredInPlaylistEntry.
po:featuredInPlaylistEntry	Се користи за поврзување на po:Song инстанца со po:PlaylistEntry инстанца. Инверзно својство на po:playlistEntrySong.
po:artistInfo	Се користи за поврзување на po:Song инстанца со mo:MusicArtist инстанца од LOD облакот.
po:songInfo	Се користи за поврзување на po:Song инстанца со mo:Track инстанца од LOD облакот.

Табела 4.19: Податочни својства од Playlist онтологијата

Својство	Опис
po:position	Позиција на елементот во топ-листата.
po:week	Недела од годината во која се појавува елементот во топ-листата.
po:year	Година во која се појавува елементот во топ-листата.
po:photoURL	URL со слика за елементот од топ-листата.
po:playlistName	Името на топ-листата.
po:stationName	Името на радио станицата.

Табела 4.20: Надворешни податочни својства кои се користат при анотација

Својство	Опис
foaf:name	Се користи за означување на името на артистот на една po:Song инстанца.
dc:title	Се користи за означување на името песната, која е po:Song инстанца.

Трансформација во поврзани податоци. Откако во RDF графот се вчитани трансформираните податоци, потребно е нивно претворање во поврзано податочно множество. За таа цел, потребно е да се поврзан ентитетите од нашето податочно множество со ентитети од LOD облакот. За вакво поврзување, нашата анализа покажа дека најмногу одговараат податоците од MusicBrainz, поточно нивната верзија во формат на поврзани податоци - LinkedBrainz. Дополнителна предност беше фактот што поврзаното податочно множество од LinkedBrainz е достапно преку јавен SPARQL endpoint.

За реализација на поврзувањето, ги искористивме двете објектни својства дефинирани во нашата Playlist онтологија токму за оваа намена: `po:songInfo` и `po:artistInfo`. Својството `po:songInfo` го користиме за поврзување на `po:Song` инстанца со `mo:Track` инстанца опишана на LinkedBrainz. За оваа цел, ја лоцираме `mo:Track` инстанцата од LinkedBrainz која го има истиот наслов со нашата `po:Song` инстанца, која е изведена од страна на истиот артист и за нив креираме RDF тројка која ја поврзува `po:Song` инстанцата со `mo:Track` инстанцата, преку `po:songInfo` својството (Слика 4.15).

На сличен начин, го искористуваме `po:artistInfo` својството за поврзување на `po:Song` инстанца од нашето множество со `mo:MusicArtist` инстанца од LinkedBrainz, каде поврзувањето се прави врз основа на името на артистот која ја изведува `po:Song` инстанцата и името на `mo:MusicArtist` инстанцата.

Поврзувањето беше имплементирано со помош на процедури дефинирани преку SPARQL прашања, кои се активираат со помош на скрипта откако RDF графот е креиран или ажуриран (Слика 4.14).

Овие `po:songInfo` и `po:artistInfo` релации претставуваат *порта* преку која можеме да добиеме дополнителни информации и детали за песната и нејзиниот изведувач, односно да овозможиме голем број на нови кориснички сценарија. По креирање на овие RDF тројки помеѓу нашето податочно множество и податочното множество на LinkedBrainz добиваме пристап до нови информации од музичкиот домен поврзани до ентитетите од нашето податочно множество, но не само директно од LinkedBrainz податочното множество, туку и од останатите поврзани податочни множества од LOD облакот кои се поврзани со него (Слика 2.3). Ова обезбедува можност за потенцијално изминување на целиот LOD облак, тргнувајќи од нашето податочно множество со песни на топ-листи, со што бројот на потенцијални сценарија за употреба на множеството значително се зголемува.

4.4.2 Кориснички сценарија и веб апликација за анализа на музичките топ-листи

Прво корисничко сценарио. Својството `po:songInfo` овозможува *движење* надвор од нашето поврзано податочно множество, преку кое можеме да добиеме дополнителни податоци од податочното множество на LinkedBrainz за песната од интерес. На пример, доколку сакаме да дознаеме од која албум или издание е песната која ја гледаме на одредена топ-листа, заедно со датумот на објава, можеме да искористиме

SPARQL прашање слично на следното:

SPARQL прашање 4.11

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX po: <http://purl.org/net/po#>
PREFIX pd: <http://purl.org/net/lmd/data#>

SELECT distinct ?artist str(?songTitle) str(?releaseTitle)
                ?releaseDate ?releasePlace
WHERE {
   GRAPH <http://purl.org/net/lmd/data#> {
      pd:JYChLBn-1-3 po:playlistEntrySong ?song .
      ?song po:songInfo ?mbs ;
            foaf:name ?artist .
   }
   SERVICE <http://linkedbrainz.org/sparql> {
      ?mbs dc:title ?songTitle .
      ?record mo:track ?mbs .
      ?release mo:record ?record ;
               dc:title ?releaseTitle .
      ?releaseEvent mo:release ?release ;
                    dc:date ?releaseDate ;
                    event:place ?place .
      ?place rdfs:label ?releasePlace .
  }
}
ORDER BY ?releaseDate
```

Прашањето започнува со извршување над локалното податочно множество, барајќи ја инстанцата од `po:Song` која репрезентира песна која е дел од записот `pd:JYChLBn-1-3` од одредена топ-листа. Во овој случај, записот се однесува на песната "Give Life Back to Music" од групата "Daft Punk", која е дел од една од топ-листите во нашето податочно множество. Лоцираната `po:Song` инстанца е веќе поврзана со песна од LinkedBrainz преку соодветна RDF тројка, па идентификаторот на LinkedBrainz песната се испраќа како променлива во соодветно под-прашање наменето за SPARQL endpoint-от на LinkedBrainz, преку концептот на SPARQL здружување на прашања [169]. Како резултат од овој дополнителен повик, ги добиваме потребните податоци за песната од интерес. Поради обемноста на резултатите, мал дел од нив се прикажани во Табела 4.21. Како што можеме да видиме од резултатите, ваквото прашање може да се искористи од страна на корисничка апликација која на корисниците ќе им обезбеди дополнителни информации за песната и албумот или изданието од кое таа е дел, податоци кои не се достапни во нашето податочно множество, ниту кај нашите податочни извори.

Табела 4.21: Резултати од SPARQL прашањето

Песна	Албум	Датум	Место
Give Life Back to Music	Random Access Memories	2013-05-17	United States
Give Life Back to Music	Random Access Memories	2013-05-17	Germany
Give Life Back to Music	Random Access Memories	2013-05-17	Netherlands
Give Life Back to Music	Random Access Memories	2013-05-20	United Kingdom

Второ корисничко сценарио. Друго можно сценарио е добивањето дополнителни информации за самиот изведувач на песната која е дел од одредена топ-листа. На пример, во таков случај можеме да побараме слика, опис и линк до официјална веб страна на изведувачот, за што би можеле да искористиме SPARQL од следниот тип:

SPARQL прашање 4.12

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX po: <http://purl.org/net/po#>
PREFIX pd: <http://purl.org/net/lmd/data#>

SELECT distinct ?thumbnail ?abstract ?website
WHERE {
   GRAPH <http://purl.org/net/lmd/data#> {
      pd:ghQTOqj-1-4 po:playlistEntrySong ?song .
      ?song po:artistInfo ?artist .
   }
   SERVICE <http://linkedbrainz.org/sparql> {
      ?artist owl:sameAs ?dbArtist .
   }
   SERVICE <http://dbpedia.org/sparql> {
      ?dbArtist dbpedia:thumbnail ?thumbnail ;
                foaf:homepage ?website ;
                dbpedia:abstract ?abstract .
      FILTER langMatches(lang(?abstract), "EN")
   }
}
```

Прашањето повторно започнува од локалното податочно множество, но потоа продолжува да бара дополнителни податоци од податочните множества на LinkedBrainz и DBpedia, со SPARQL здружување на прашања, со цел да се обезбедат повеќе информации за корисничкото сценарио. Резултати од прашањето се дадени во Табела 4.22.

Табела 4.22: Резултати од SPARQL прашањето

Слика	Апстракт	Веб страна
http://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Katy_Perry_UNICEF_2012.jpg/200px-Katy_Perry_UNICEF_2012.jpg	"Katheryn Elizabeth Hudson (born October 25, 1984), known by her stage name Katy Perry, is an American recording artist, songwriter, and actress..."	http://www.katyperry.com/

Податоците добиени како дел од одговорите на претходните две прашања не се дел од нашето поврзано податочно множество, ниту пак можат да се најдат на изворните веб страни од каде го генериравме податочното множество. Овие податоци ги добиваме од други, дистрибуирани податочни множества, кои следејќи ги принципите на поврзани податоци се достапни преку користење на W3C стандарди во рамките на постоечката инфраструктура на Вебот. Овие податоци можат да се искористат во кориснички апликации кои на корисниците ќе им понудат повеќе информации за песните кои се дел од топ-листите, како и повеќе информации за нивните изведувачи. Наведените пример SPARQL прашања можат да бидат извршени и како дел од REST-базиран повик преку апликација, односно со употреба на SPARQL endpoint-от како REST-базиран сервис, со повици во форматот:

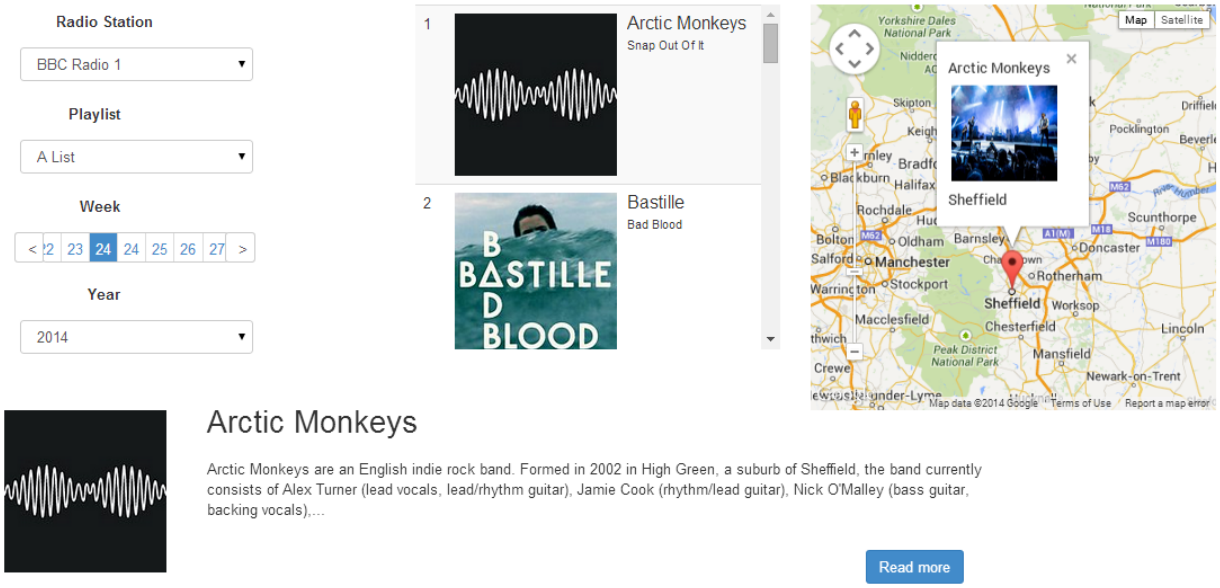
```
http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

Тука, `SPARQLQUERY` се однесува на URL-кодирана верзија од SPARQL прашањето, додека `FORMAT` го претставува форматот на одговорот кој може да биде HTML, XML, JSON, CSV, RDF/XML, N3, Turtle, JSON-LD, итн. SPARQL endpoint-от прифаќа и користење на `Accept` заглавието од HTTP стандардот, преку кое може да се дефинира преферираниот излезен формат.

Покрај овие, можат да се постигнат и други корисни кориснички сценарија преку нашето поврзано податочно множество. На пример, би можеле да ги добиеме адресите на официјалните профили на изведувачите на социјални медиуми, да ги најдеме издавачките куќи кои ги издале нивните албуми, или пак да направиме аналитички прашања кои би ги детектирале изведувачите или издавачките куќи кои имаат најголем број песни присутни на одредени радио топ-листи, во различни региони во светот, итн. Ваквите кориснички сценарија би требало да се користат од страна на развивачи на софтвер во нивни апликации од доменот на музика и забава, со цел да им се овозможи на нивните корисници искуство збогатено со музичките податоци од LOD облакот.

Веб апликација. Со цел да ја демонстрираме употребливата вредност на корисничките сценарија, развивме специјализирана веб апликација. Веб апликацијата го користи нашето поврзано податочно множество од јавната Virtuoso инстанца и има за цел да им обезбеди на крајните корисници основни информации за изведувачите и песните од

изворните топ-листи (Слика 4.16), како и да им овозможи аналитички поглед на податоците - глобален преглед на државите од кои потекнуваат изведувачите на песните кои се дел од дадена топ-листа, како и следење на нивната динамика од недела во недела (Слика 4.17).

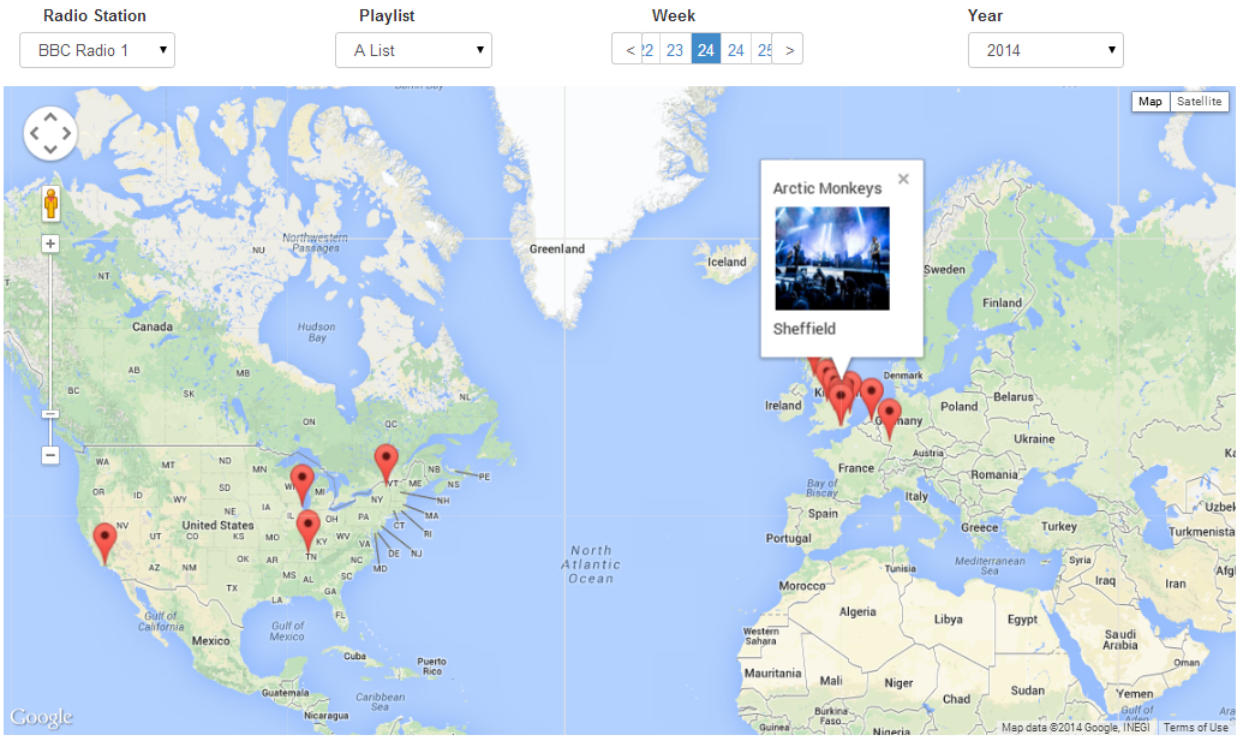


Слика 4.16: Детали за топ-листата и изведувачот во веб апликацијата.

Веб апликацијата го користи нашиот SPARQL endpoint [57] за пристап до податоците од нашето поврзано податочно множество, но и податоците од LOD облакот, преку SPARQL здружување на прашања. Едно основно корисничко сценарио во веб апликацијата е приказ на дополнителни информации за изведувачот од интерес. Ова корисничко сценарио може да се реализира преку користење на топ-листата на соодветна радио станица, записите кои се дел од топ-листата од тековната недела од локалното податочно множество, заедно со дополнителни детали за изведувачот - слика, кратка биографија, гео-локација на местото на раѓање или потекло на изведувачот - добиени од LOD облакот (Слика 4.16).

За корисници заинтересирани за аналитички преглед на податоците, веб апликацијата обезбедува корисничко сценарио во кое се дава глобален приказ на местата од каде потекнуваат изведувачите од дадена топ-листа, во одредена недела од годината (Слика 4.17). Со селекција на различна топ-листа, корисникот може да добие увид во разликите помеѓу радио станиците и различната застапеност на државите и изведувачите кај нив. Дополнително, со менување на неделата за дадена топ-листа, корисникот може визуелно да ја анализира истата динамика, од недела во недела. Ваквото корисничко сценарио користи податоци од LOD облакот, со цел да се добијат информации за изведувачот, местото на раѓање или потекло и гео-локациските податоци за самото место.

Овие кориснички сценарија од веб апликацијата директно ја поддржуваат идејата која ја поставивме иницијално: да демонстрираме дека употребата на принципите



Слика 4.17: Преглед на географската застапеност на глобално ниво на изведувачите чии песни се дел од селектираната топ-листа, во одредена недела од годината.

на поврзани податоци во доменот на музиката може да донесе бенефит за крајните корисници од доменот, преку обезбедување нови, понапредни и пошироки кориснички сценарија.

4.4.3 Дискусија

Во рамките на ова истражување дизајниравме систем за автоматизирано преземање и трансформирање на податоци од глобалните радио станици и нивните официјални топ-листи, во висококвалитетни поврзани податоци. Ја дизајниравме и објавивме и Playlist онтологијата. На крајот, покрај демонстрираните кориснички сценарија овозможени од поврзаната природа на податочното множество, развивме и веб апликација која ги директно ги користи и прикажува бенефитите од можноста за непречен и отворен пристап до дополнителни податоци од LOD облакот.

Како што веќе знаеме од [87] и [150], ваквиот тип на податоци може од една страна да им помогне на бизнис секторот и на независните развивачи на софтвер преку креирање на нова бизнис вредност во економијата со уникатни кориснички сценарија за апликации и сервиси, но од друга страна и на јавноста која е краен корисник на овие апликации и сервиси. Целта на нашето истражување беше да се демонстрираат предностите од користење на принципите на поврзани податоци во областа на музиката и забавата, преку кои се креираат нови кориснички сценарија, претходно недостапни над изолираните изворни податоци. Ваквите нови кориснички сценарија, заедно со јавните

поврзани податочни множества, обезбедуваат солидна основа за понатамошен развој на апликации и сервиси од страна на заедницата на развивачи на софтвер и од компании, со што потенцијално може да се создаде нова бизнис вредност во индустријата.

4.5 Поврзани здравствени податоци

Пристапот до консолидирани податоци од областа на лековите и здравството, преку Веб, е предизвикувачка задача. Главната причина за тоа се наоѓа во фактот што податоците се достапни во различни формати, на дистрибуирани локации на Вебот. Дополнително, најголемиот дел од податочните множества за лекови или други здравствени поддомени се креирани и објавени со специфична намена, па согласно на тоа, имаат лимитиран спектар на употреба. Поради тоа, еден од главните предизвици во доменот на здравствени податоци е интегрирање и консолидирање на големи податоци хетерогени податоци, во еден комплетен податочен простор. Тековната структура на податоците од доменот достапни на Веб го забавува развојот на податочно-базирани апликации. Како дел од нашите истражувања, се фокусиравме на овој домен и преку имплементација на принципите на поврзани податоци успеавме да креираме консолидирани поврзани податочни множества за лековите и медицинските установи од Македонија. Податоците за генерички лекови на глобално ниво ги корелиравме со податоци за рецепти од сите глобални кујни, со што направивме компаративна анализа на негативното влијание на различните национални кујни врз лековите од различни категории. Оваа анализа овозможи креирање на глобални прегледи на распределбата на ваквото влијание, како и идентификација на најпроблематичните состојки по региони - состојки кои најчесто се одговорни за негативните интеракции помеѓу храната и лековите.

Во продолжение ќе направиме преглед на наведените истражувањата и нивните резултати.

4.5.1 Поврзани податоци за лекови во Македонија: Фонд за здравствено осигурување

Првото истражување во областа го реализиравме со податоци за лекови од Фондот за здравствено осигурување на Република Македонија (ФЗОМ). Со цел да обезбедиме напредни кориснички сценарија за искористување на податоците од доменот на здравството во Македонија, генериравме висококвалитетно, 5-star поврзано податочно множество од податоците на ФЗОМ како извор. Ова податочно множество се состои од сите лекови регистрирани во Република Македонија, заедно со нивните цени контролирани од Фондот и Министерството за здравство на Република Македонија. Секој од овие лекови е поврзан со соодветен генерички лек од DrugBank податочното множество, со што нашето податочно множество се поврзува со LOD облакот [140].

Трансформација на податоците

Фондот за здравствено осигурување на Република Македонија (ФЗОМ) е единствената осигурителна организација во која се спроведува задолжителното здравствено осигурување врз начелата на заемност солидарност. Во Фондот се обезбедуваат средствата за основната здравствена заштита, за мрежата на здравствените организации и финансирање на дејноста на здравствените организации, врз основа на цени на здравствени услуги односно програми и договори за спроведување на здравствената заштита на осигурениците. Дополнително, Фондот заедно со други државни институции како што се Бирото за лекови на Република Македонија и Министерството за здравство, ги регулира листите на лекови покриени од здравственото осигурување на осигурениците и ги дефинира референтните цени на дел од лековите. Имајќи ја предвид ваквата улога на Фондот во сферата на здравството во Македонија, сметавме дека пристапот до нивните податоци во формат на поврзани податоци може да донесе големи предности за крајните корисници.

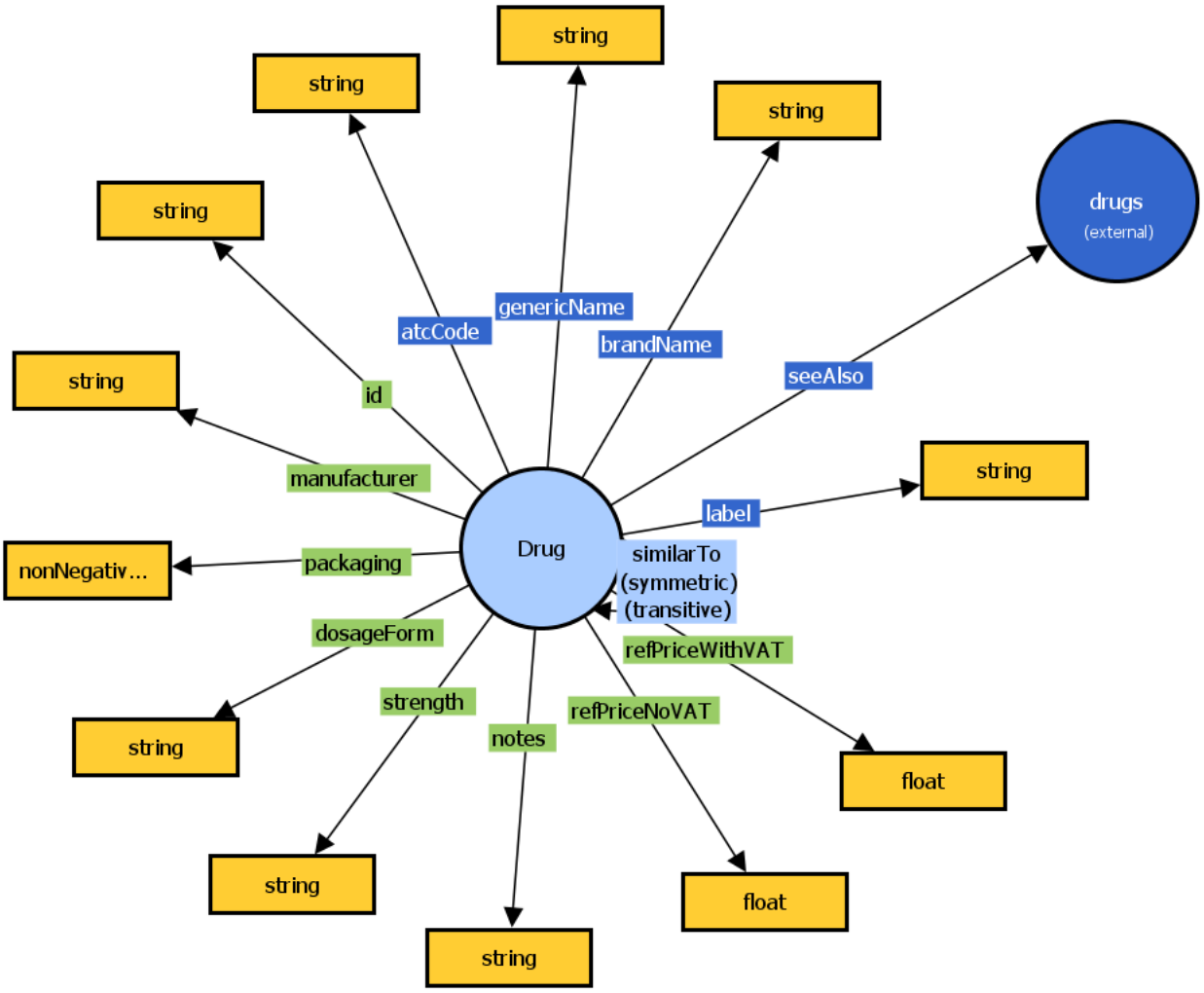
Податоци од ФЗОМ. ФЗОМ своите податоци од јавен карактер редовно ги објавува на својата официјална веб страна [25]. Овие податоци содржат информации за здравствените услуги и нивните цени, статистика за искористеноста на болничките легла, извештаи од инспекцискиот надзор во јавните и приватните здравствени установи, финансиски податоци од Фондот, информации за осигурувањето, референтни цени на лекови, информации за јавните и приватните здравствени установи кои соработуваат со Фондот, итн. Иако овие податоци можат технички да се сметаат за отворени податоци, тие главно се достапни во PDF или Excel формат, со што нивниот квалитет е 1-star или 2-star. Со цел да ја зголемиме употребливоста на јавните податоци од Фондот, одлучивме да ги трансформираме во 5-star поврзани податоци: најпрвин да ги трансформираме во RDF, а потоа да ги поврземе со други податочни множества од LOD облакот.

Како почетна точка го одбравме податочното множество со лекови, кое содржи фармаколошки и фармацевтски податоци за лековите регистрирани во Република Македонија, како и референтна листа на цени за дел од нив.

HIFM онтологија. Податоците за лекови објавени од Фондот се распределени во неколку различни податочни множества, кои содржат различни типови информации. Сумарно, во нив се среќаваат информации за името на лекот, генеричкото име, производителот, референтната цена, пакувањето, јачината и дозирањето. Дополнително, секој лек е идентификуван со ID идентификатор генериран од Фондот, како и глобално дефиниран ATC код кој се користи за класификација на лекот и е контролиран од страна на Светската здравствена организација (СЗО).

Со цел да ги трансформираме и репрезентираме податоците во RDF формат, потребна беше онтологија. Следејќи ги најдобрите практики за развој на онтологии, одлучивме да искористиме некои од постоечките онтологии за лекови. Во овој процес беше важно да се одбере онтологија која ќе користи и во поврзувањето на податоците со LOD облакот, односно беше потребна онтологија која веќе се користи од страна на податочни множества кои се дел од LOD облакот. Поради ова, се одлучивме да ја користиме онто-

логијата на DrugBank податочното множество [62]. Онтологијата на DrugBank проектот содржи класа `drugbank:drugs`, која ги претставува лековите. Таа дефинира и релации за ATC кодот, генеричкото име и продажното име на лекот. За анотација на нашето податочно множество ја искористивме `drugbank:drugs` класата, како и својствата `drugbank:atcCode`, `drugbank:genericName` и `drugbank:brandName` (Слика 4.18, Табела 4.23).



Слика 4.18: HIFM онтологија и помошните класи и својства од постоечките онтологии.

Табела 4.23: Својства од DrugBank онтологијата искористени при анотација

Својство	Опис
`drugbank:atcCode`	Глобалниот ATC код на лекот.
`drugbank:genericName`	Генеричкото име на лекот.
`drugbank:brandName`	Продажното име на лекот.

Табела 4.24: Својствата од HIFM онтологијата

Својство	Опис
`hifm:id`	Идентификатор на лекот, дефиниран од ФЗОМ.
`hifm:manufacturer`	Производител на лекот.
`hifm:refPriceNoVAT`	Референтна цена на лекот во МКД, без ДДВ.
`hifm:refPriceWithVAT`	Референтна цена на лекот во МКД, со ДДВ.
`hifm:packaging`	Тип на пакување на лекот.
`hifm:dosageForm`	Начин на дозирање на лекот.
`hifm:strength`	Јачина активната супстанца на лекот.
`hifm:similarTo`	Својството поврзува два лека кои имаат иста активна супстанца и иста функција, но може да се најдат под различно продажно име, од различен производител, со различна јачина, пакување, итн.

Сепак, онтологијата на DrugBank не беше доволна за да се анотираат сите податоци достапни од Фондот. Анализата на другите онтологии од доменот покажа дека ниту една друга постоечка онтологија не може да се искористи за нив. За таа цел, дизајниравме и развивме нова онтологија: HIFM онтологијата (Слика 4.18). HIFM онтологијата дефинира сопствена класа за лекови, `hifm:Drug`, седум податочни својства и едно објектно својство, `hifm:similarTo` (Слика 4.18, Табела 4.24).

Покрај својствата од DrugBank и својствата дефинирани во нашата HIFM онтологија, ги искористивме и `rdfs:label` и `rdfs:seeAlso` својствата. Првото од нив го користиме за анотација на генеричкото име на лекот, додека второто го користиме за поврзување на лек од нашето податочно множество со лек од податочното множество на DrugBank. Повеќе детали за ваквото поврзување ќе бидат наведени подолу во текстот.

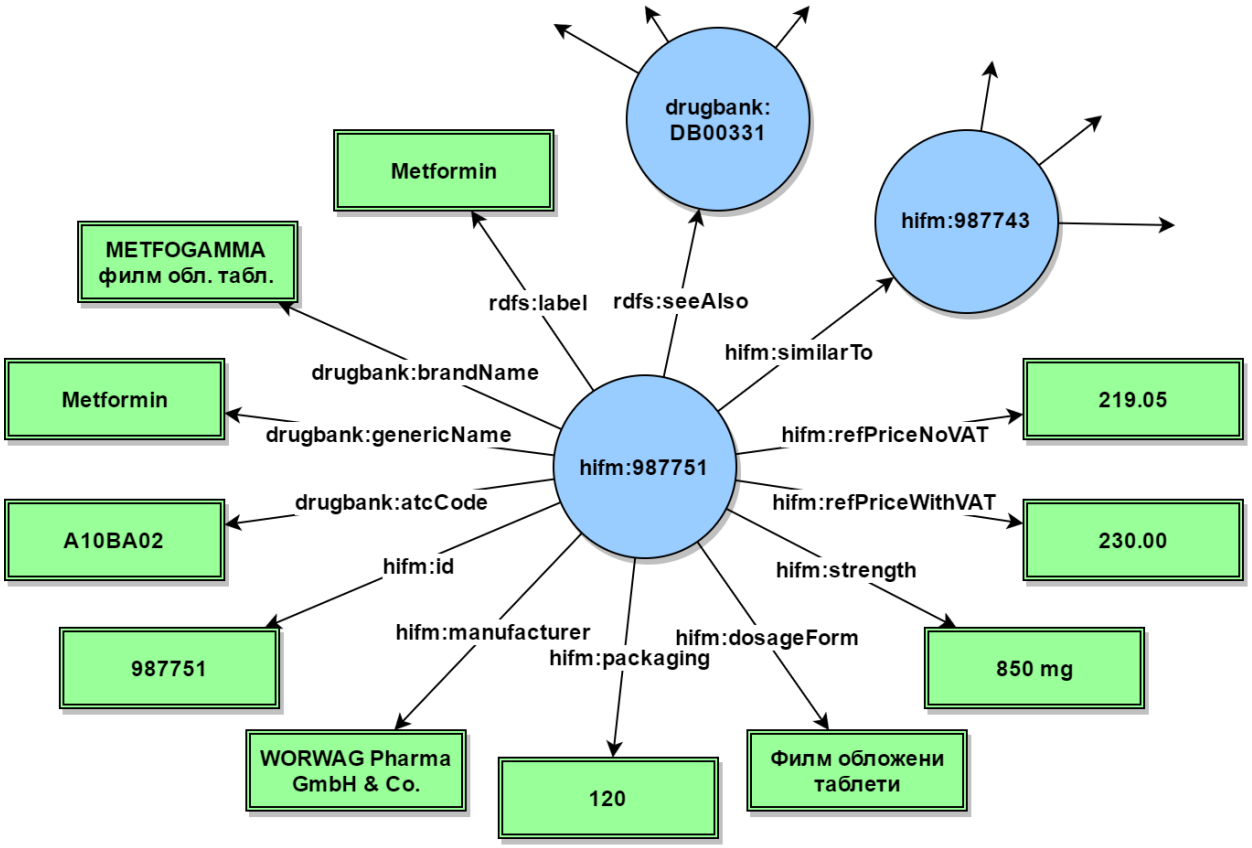
Трансформација во поврзани податоци. По дефинирањето на шемата на податочното множество (Слика 4.18), следен чекор беше трансформација на податоците во RDF, а потоа и во поврзани податоци. За трансформација искористивме инстанца од Virtuoso Universal Server, а самото мапирање се одвиваше во два чекори. Првин, во Virtuoso инстанцата ги импортиравме CSV датотеките генерирани од Excel датотеките достапни на веб страната на Фондот. Потоа, со користење на R2RML, јазикот за мапирање на податоци од релациони бази на податоци во RDF формат, креиравме RDF погледи над релационите табели од внесените CSV датотеки. Мапирањата ја користеа шемата дефинирана со HIFM онтологијата, DrugBank онтологијата и неколкуте помошни својства. Како идентификатор на лек-ентитетите ја одбравме ID вредноста доделена од Фондот. Секоја инстанца од лек беше инстанца и од `drugbank:drugs` и од `hifm:Drug` класата, додека останатите податоци беа анотирани со соодветните својства од двете онтологии (Слика 4.18, Табела 4.23, Табела 4.24).

Со оглед на тоа што изворните податоци од веб страната на Фондот беа поделени во повеќе датотеки, процесот резултираше со неколку посебни RDF погледи, односно RDF графови, со различни информации за лековите. Со цел да ги консолидираме податоците за лековите, со користење на SPARQL ги искомбиниравме податоците од поединечните графови и ги внесовме во еден заеднички RDF граф. Користењето на ист формат за URI идентификаторите на лековите овозможи непречено поврзување на податоците во еден граф, поради тоа што еден лек имаше ист URI идентификатор и во различни RDF графови.

Откако ги консолидиравме податоците за лековите во еден единствен RDF граф, следен чекор беше креирање линкови помеѓу самите лекови од податочното множество. Како што веќе наведовме во описот на HIFM онтологијата, во неа го дефиниравме објектното својство `hifm:similarTo` (Слика 4.18, Табела 4.24), како транзитивно и симетрично својство. Целта на својството е поврзување на Лек А со Лек Б (и обратно), доколку го имаат истиот ATC код. Бидејќи дел од лековите објавени од страна на Фондот содржат повеќе од стандардните седум карактери во нивниот ATC код, при што дополнителните карактери служат за разликување на локално ниво на истите лекови кои доаѓаат во различно пакување или јачина, ги користиме само оригиналните седум карактери од кодот. Поврзувањето на два лека со ист ATC код со две RDF тројки - по една во секоја насока - овозможува контекстно поврзување на лекови кои ја имаат истата функција, односно ги поседуваат истите терапевтски, фармаколошки и хемиски својства. Овие лекови може да се разликуваат единствено во името под кое се продаваат, производителот, јачината и слично, но нивната намена е иста. Како што ќе видиме подолу во текстот, овие поврзувања на лековите овозможуваат нови кориснички сценарија за крајните корисници.

За да го трансформираме податочно множество во висококвалитетно, 5-star поврзано податочно множество, потребно беше да креираме линкови помеѓу ентитетите од податочното множество со ентитети кои се појавуваат во LOD облакот. За таа цел, одлучивме да се поврземе со DrugBank податочното множество, кое е најголемото во областа и кое содржи најмногу детали за генерички лекови на Вебот. Слично како и во процесот на меѓусебно поврзување на лековите од нашето податочно множество, ги искористивме ATC кодовите за да ги идентификуваме генеричките лекови од DrugBank кои по функција одговараат на лековите кои се наоѓаат во продажба во Република Македонија. За овие генерички лекови од DrugBank креиравме RDF тројки во нашето податочно множество кои ги поврзуваат соодветните локални лекови со нив. За таа цел го искористивме својството `rdfs:seeAlso` и како што ќе видиме подолу во текстот, новите RDF тројки овозможија нови, напредни кориснички сценарија над податоците за лекови од ФЗОМ.

Пример лек од генерираното поврзано податочно множество со лекови регистрирани за продажба во Република Македонија е даден на Слика 4.19. На сликата може да се видат својствата на лекот, како и неговата поврзаност со друг лек од истото множество и еден генерички лек од DrugBank.



Слика 4.19: Пример лек од RDF графот со податоци од ФЗОМ.

Го одбравме својството `rdfs:seeAlso` наместо својството `owl:sameAs`, поради тоа што ентитетите од нашето податочно множество и ентитетите во DrugBank не се идентични. Во нашето податочно множество еден *лек* претставува лек кој се продава на територија на Република Македонија и кој има свое продажно име, производител, пакување, јачина, дозирање, референтна цена, итн. Во DrugBank податочното множество, пак, еден *лек* претставува генерички лек, односно активна супстанца која може да се сретне во еден или повеќе лекови во продажба, под различно продажно име, од различен производител, со различна јачина, итн. За овие генерички лекови во податочното множество на DrugBank постојат информации за хемиската формула, молекуларната маса, организмите кај кои делува, интеракциите со други генерички лекови, интеракциите со храна, итн.

Последниот чекор во постапката, откако го трансформиравме податочното множество во поврзано податочно множество, беше негово објавување на Веб. Поврзаното податочно множество го објавивме на јавна Virtuoso инстанца на Факултетот за информатички науки и компјутерско инженерство [66], согласно најдобрите практики и препораки на W3C [126]. Преку SPARQL endpoint-от на оваа Virtuoso инстанца, RDF графот со поврзаното податочно множество од ФЗОМ е достапен преку Веб. Функционирањето на SPARQL endpoint-от како REST-базиран сервис овозможува пристапување

до податочното множество и директно од кориснички апликации.

Дополнително, поврзаното податочно множество го објавивме и во вид на RDF датотеки на глобалниот Datahub податочен портал [130]. Корисниците можат слободно да пристапат до податоците од овој податочен каталог и да ги искористат во своите апликации и анализи.

Кориснички сценарија

Целта на трансформацијата на изворните податоци во поврзани податоци е зголемување на нивната употребливост, преку овозможување нови, напредни сценарија кои не се достапни над изолираното податочно множество. Дополнително, форматот на поврзани податоци овозможува директен пристап до податоците преку постоечката инфраструктура на Вебот и постоечките W3C стандарди.

Прво корисничко сценарио. Едно ново корисничко сценарио над креираното поврзано податочно множество би било искористувањето на `hifm:similarTo` релацијата со цел пристап до слични лекови на тековно разгледуваниот. Под слични лекови мислиме на лекови кои ја имаат истата активна супстанца и намена како и разгледуваниот лек, но може да имаат различно продажно име, различна цена, пакување, јачина, може да се произведени од друг производител, итн. На пример, доколку го разгледуваме лекот "NIFADIL, film coated tablets, 50 x 10mg" од нашето поврзано податочно множество, можеме да ги идентификуваме лековите слични на него со користење на следново SPARQL прашање:

SPARQL прашање 4.13

```
PREFIX hifm: <http://www.fzo.org.mk/ontology/hifm#>
PREFIX drugbank: <http://wifo5-04.informatik.uni-mannheim.de/
                                    drugbank/resource/drugbank/>

SELECT ?brandName ?price ?manufacturer
WHERE {
   hifm:79588 hifm:similarTo ?drug .
   ?drug drugbank:brandName ?brandName ;
         hifm:refPriceWithVAT ?price ;
         hifm:manufacturer ?manufacturer .
}
ORDER BY ASC (?brandName)
```

Во прашањето, `hifm:79588` е лекот кој го разгледуваме, па преку `hifm:similarTo` релациите можеме од него да стигнеме до повеќе детали за лековите од истото податочно множество кои ја имаат истата функција. Во конкретното прашање ги селектираме продажното име, цената и производителот за лековите слични со лекот `hifm:79588`, а резултатите се прикажани во Табела 4.25.

Табела 4.25: Резултати од SPARQL прашањето

Продажно име	Цена	Производител
CORDIPIN R, 30 x 20mg	14,00	KRKA
CORDIPIN XL, 20 x 40mg	19,00	KRKA
KORINCARE NEO, 20 x 40mg	19,00	TCHAIKAPHARMA
KORINCARE, 20 x 20mg	9,00	TCHAIKAPHARMA
NIFADIL RETARD, 30 x 20mg	14,00	ALKALOID
NIFEDIPIN RETARD, 30 x 20mg	14,00	REPLEKFARM
NIFEDIPIN, 50 x 10mg	35,00	JAKA 80
NIFELAT RETARD, 30 x 20mg	14,00	ZDRAVLJE

Второ корисничко сценарио. Бидејќи нашето поврзано податочно множество со лекови од ФЗОМ содржи линкови кон други податочни множества на Вебот, можеме да ги искористиме за да ги изминуваме овие оддалечени податочни графови. Со други зборови, искористувајќи ги `rdfs:seeAlso` релациите, можеме да добиеме информации кои се наоѓаат во DrugBank податочното множество и кои не се дел од нашите локални податоци. На пример, доколку сакаме да добиеме информации за интеракциите со храна на лекот "DILACOR, tablets, 20 x 0,25mg", можеме да го искористиме следново SPARQL прашање:

SPARQL прашање 4.14

```
PREFIX hifm: <http://www.fzo.org.mk/ontology/hifm#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX drugbank: <http://wifo5-04.informatik.uni-mannheim.de/
                                    drugbank/resource/drugbank/>

SELECT ?foodInteraction
WHERE {
   hifm:32964 owl:seeAlso ?dbdrug .
   SERVICE <http://wifo5-04.informatik.uni-mannheim.de/drugbank/sparql> {
      ?dbdrug drugbank:foodInteraction ?foodInteraction .
   }
}
ORDER BY ASC (?foodInteraction)
```

Ова SPARQL прашање започнува од локалниот RDF граф, барајќи ги сите RDF тројки кои означуваат дека лекот `hifm:32964` има `rdfs:seeAlso` релација со лек од DrugBank податочното множество. Пронајдените лекови од ова пребарување се всушност ентитети кои претставуваат генерички лекови и кои се опишани во DrugBank.

Користејќи го принципот на SPARQL здружување на прашања, праќаме потпрашање до SPARQL endpoint-от на DrugBank во кое ги бараме деталите за интеракции со храна за соодветните генерички лекови, поврзани со нашиот локален лек `hifm:32964`. Како резултат од целосното SPARQL прашање, добиваме листа од интеракции со храна на сите генерички лекови кои се поврзани со нашиот `hifm:32964` лек. Во овој случај, лекот `hifm:32964` има само со една активна супстанца, т.е. е поврзан само со еден генерички лек од DrugBank, а неговите интеракции со храна се прикажани како резултат од прашањето во Табела 4.26 .

Табела 4.26: Резултати од SPARQL прашањето

Интеракции со храна
Avoid avocado.
Avoid bran and high fiber foods within 2 hours of taking this medication.
Avoid excess salt/sodium unless otherwise instructed by your physician.
Avoid milk, calcium containing dairy products, iron, antacids, or aluminium salts 2 hours before or 6 hours after using antacids while on this medication.
Avoid salt substitutes containing potassium.
Limit garlic, ginger, gingko, and horse chestnut.

Податоците прикажани во Табела 4.26 не се наоѓаат во нашето поврзано податочно множество, ниту во изворните податоци од ФЗОМ. Овие податоци се достапни благодарение на поврзаната природа на генерираното податочно множество, во кое имаме RDF тројки кои поврзуваат ентитети од локалното податочно множество, со ентитети од податочното множество на DrugBank. Користејќи го ова контекстно поврзување, можеме да пристапиме до сите податоци кои се наоѓаат во DrugBank податочното множество.

За овие кориснички сценарија важно е да напоменеме дека можат да се извршуваат директно од кориснички апликации, благодарение на фактот што SPARQL endpoint-от на Virtuoso може да се користи и како REST-базиран сервис. Притоа, слично како и кај останатите елаборирани истражувања и проекти, повиците до веб сервисот се во следниот формат:

```
http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

Тука, `SPARQLQUERY` го претставува SPARQL прашањето во URL-кодиран формат, додека `FORMAT` се однесува на посакуваниот формат на резултатите од прашањето. Форматот може да биде HTML, XML, JSON, Javascript, CSV, Spreadsheet, RDF/XML, N3, Turtle, итн.

4.5.2 Поврзани податоци за медицински установи во Македонија

Фондот за здравство на Република Македонија, покрај тоа што објавува податоци од јавен карактер за лековите и нивните цени, објавува и други податоци од областа на неговата дејност: листа на регистрирани здравствени установи, нивната официјална дејност, соодветно категоризирана, нивното работно време и нивната официјална адреса. Овие податоци се исто така од големо значење за крајните корисници, но нивниот формат на веб страната на Фондот е со низок квалитет - најчесто е 1-star или 2-star. Тоа значи дека за да се овозможи директен пристап до податоците за кориснички апликации, потребна е одредена обработка и трансформација на податоците. Слично како и во претходното истражување, нашата цел беше генерирање на поврзано податочно множество од овие податоци. За да ги овозможиме потребните кориснички сценарија над ваквиот тип податоци, односно за да ја демонстрираме предноста од постоењето на ваквите податоци во формат на поврзани податоци, ги вклучивме и податоците за расположливоста на лековите во аптеките, од Здружението на аптеки на Македонија (ЗАМ). Поврзувајќи ги овие податочни множества и податочното множество од претходното истражување [140], успеавме да добиеме нови кориснички сценарија кои ја искористуваат поврзаната природа на податоците [139]. Овие кориснички сценарија можат понатаму да се користат во апликации и сервиси кои обезбедуваат релевантни податоци за своите корисници.

Трансформација на податоците

За да генерираме комплетно податочно множество за здравствените установи од Република Македонија, потребно беше да се соберат, трансформираат и поврзат податоци од неколку различни извори. Потребни беа податоци за имињата, адресите и дејноста на здравствените установи, нивната точна географска локација, листата на услуги достапни во нив, како и листата на лекови достапни во аптеките во тековниот месец. Дел од овие податоци се директно достапни на Вебот, во 1-star и 2-star квалитет, дел од податоците беше потребно да се преработат и генерираат, а дел од податоците мораше и лично да ги побараме и добиеме, со цел да ги демонстрираме идеите и предностите од вакво комплетирано поврзано податочно множество.

Извори на податоци. Фондот за здравство на Република Македонија на својата официјална веб страна објавува и одржува листа на регистрирани здравствени установи. Во оваа листа се наоѓаат информации за името, за типот на дејност на установата и нејзината адреса (Табела 4.27). Покрај тоа, во посебна листа се објавуваат информациите за дежурствата на установите и нивните работни времиња (Табела 4.28). Двете податочни множества се достапни за преземање во XML, односно Excel формат, соодветно. Податоците ги преземавме, но пред да ги трансформиравме во RDF формат, беше потребно да се прочистат и прошират.

Со оглед на тоа што податоците за медицинските установи достапни од веб страната

Табела 4.27: Својства дефинирани во податочното множество на медицински установи од Република Македонија и нивни географски локации

Својство	Опис
`hifm:id`	Идентификацискиот број на мед. установа.
`hifm:archiveNumber`	Уникатниот идентификатор на мед. установа во Фондот.
`hifm:medicalFacilityName`	Името на мед. установа.
`hifm:activityType`	Полето на дејност на мед. установа.
`geo:city`	Градот во кој е лоцирана мед. установа.
`geo:address`	Адресата на мед. установа.
`geo:latitude`	Географска ширина на локацијата на мед. установа.
`geo:longitude`	Географска должина на локацијата на мед. установа.

Табела 4.28: Својства дефинирани во податочното множество со дежурства на медицинските установи од Република Македонија

Својство	Опис
`hifm:id`	Идентификатор на записот за дежурството.
`geo:city`	Градот во кој е лоцирана медицинската установа.
`hifm:medicalFacilityName`	Името на мед. установа.
`hifm:dateOnDuties`	Датата за која се однесува дежурството.
`vcard:hasTelephone`	Телефонскиот број на медицинската установа.
`vcard:hasNote`	Дополнителни информации / Работно време.

на Фондот содржат информација за адресата на секоја од институциите, за да се зголеми употребливоста на податочното множество беше потребно да се изведе гео-лоцирање на секоја од нив. Со помош на колегите од истражувачкиот тим од Еко-информатика од Факултетот за информатички науки и компјутерско инженерство и нивниот софтвер, иницијалното податочно множество со медицински установи беше проширено со две дополнителни својства: географска ширина и географска должина. Овие податоци беа генерирани врз база на адресните податоци на секоја од институциите и беа вклучени во самото податочно множество (Табела 4.27).

Овие податоци од Фондот можат да обезбедат поддршка за бројни нови кориснички сценарија во рамки на кориснички апликации и сервиси, како што веќе видовме во останатите истражувања, особено во [140]. Сепак, со цел да ја демонстрираме зголемената употребливост на крајното поврзано податочно множество, одлучивме кон овие

податоци да додадеме и податоци за достапноста на конкретни лекови во конкретни аптеки во Република Македонија. За жал, овие листи на расположливи лекови не се достапни на Вебот. Но, за целите на истражувањето, добивме директен пристап до месечните листи на расположливи лекови за дел од аптеките од ЗАМ, кои сметавме дека ќе бидат доволни за демонстрирање на предностите кои ваков тип на податоци би можеле да ги донесат во доменот. Овие податоци ги добивме во Excel формат, од каде ги трансформиравме најпрвин во CSV, а потоа и во RDF формат (Табела 4.29).

Табела 4.29: Својства дефинирани во податочното множество со расположливи лекови во аптеките од Република Македонија

Својство	**Опис**
`hifm:id`	Идентификатор на лекот.
`rdfs:label`	Името на лекот.
`hifm:dosageForm`	Начин на земање на лекот.
`hifm:strength`	Јачина на лекот.
`hifm:manufacturer`	Производител на лекот.
`hifm:available`	Достапна количина.

Анотација на податоците. Пред да ја реализираме трансформацијата на податоците, потребно беше да се поврзат податоците од податочното множество со медицински установи и податочното множество со дежурства. Поради тоа што заедничка информација во двете податочни множества е името на здравствената установа, го искористивме овој податок и со помош на OpenRefine алатката [48] ги поврзавме податочните множества. Резултатот од спојувањето беше едно податочно множество, во кое ентитетите се медицински установи кои како својства ги имаат податоците од двете иницијални податочни множества.

Анотацијата на податоците при нивната трансформација во RDF бараше користење на соодветни онтологии и вокабулари. За таа цел, ги искористивме vCard онтологијата [128], W3C Geospatial онтологијата [96] и Schema.org вокабуларот [52]. Првата онтологија ја употребивме за анотација на контакт информациите за секоја од медицинските установи, додека втората ја употребивме за нивната локација (Табела 4.27, Табела 4.28). Schema.org вокабуларот го искористивме за дефинирање на типот на записите за медицинските установи, `schema:MedicalOrganization`. Овие онтологии и вокабуларот се помеѓу најкористените во рамките на LOD облакот, што оди во прилог на употребливоста на податочното множество во глобални рамки [33, 193].

Сепак, дел од податоците за медицински установи и нивни дежурства објавени од страна на Фондот не можеа да се опфатат со наведените или со други постоечки онтологии и вокабулари. За таа цел, наместо развивање на нова онтологија од почеток, ја проширивме HIFM онтологијата од претходното истражување [140]. Во HIFM онтологијата додадовме нови податочни својства кои се однесуваат на името на медицин-

ската установа, типот на дејност, архивскиот број на установата во Фондот, датумот на дежурство и достапната количина на лекот (Табела 4.27, Табела 4.28, Табела 4.29). Третото податочно множество, кое содржи информации за тековно достапните лекови во неколку аптеки членки на ЗАМ, го анотиравме со својствата прикажани во Табела 4.29.

Трансформацијата во RDF, односно во 4-star податоци, ја реализиравме со помош на Virtuoso Universal Server инстанца [111, 68]. CSV податоците ги вчитавме во Virtuoso, а потоа со R2RML јазикот за мапирање креиравме RDF погледи над вчитаните податоци. Податоците изложени преку овие RDF погледи ги ископиравме во еден единствен RDF граф, кој ги содржеше податоците од сите три изворни множества. Со цел да го поврземе податочното множество со листи на достапни лекови по аптеки со останатите две податочни множества, претходно веќе комбинирани во едно, ја искористивме информацијата за тоа на која аптека (медицинска установа) се однесува листата со лекови, па креиравме RDF тројки со кои ги споивме аптеките (медицинските установи) со соодветните лекови кои се достапни кај нив. За таа цел ја искористивме новокреираната релација `hifm:hasAvaliableMedicine`.

Трансформација во поврзани податоци. За да ги трансформираме податоците во 5-star поврзано податочно множество, потребно беше да се креираат RDF тројки кои ќе репрезентираат линкови помеѓу ентитетите од нашето податочно множество и ентитети од други податочни множества, кои се дел од LOD облакот. Со оглед на тоа што претходното истражување на оваа тема резултираше со поврзано податочно множество со лекови објавени од Фондот за здравствено осигурување на Република Македонија [140], кое е поврзано со податочни множества од LOD облакот, одлучивме да ги поврземе ентитетите од новото податочно множество со ентитетите од него. Поврзувањето го изведовме со креирање RDF тројки со `owl:sameAs` релацијата помеѓу лековите од нашето податочно множество и лековите од податочното множество од [140]. Станува збор за исти лекови, објавени од Фондот, но во едното податочно множество имаме детали за самите лекови како продукти, додека во другото имаме информација за тоа во која аптека се достапни истите тие лекови. Ваквото поврзување на податочните множества овозможува креирање на кориснички сценарија кои ќе обезбедат детални информации за конкретен лек - кој се продава во конкретна аптека, за која ја имаме локацијата и работно време - како на пример интеракциите со други лекови, интеракциите со храна, хемиската формула, други производители, алтернативни лекови, итн.

Кориснички сценарија

Поврзувањето на податочните множества со други податочни множества објавени на Вебот, кои се дел и од LOD облакот, овозможува креирање на нови, покомплексни кориснички сценарија во доменот. Тргнувајќи од податоците за медицински установи, објавени од страна на Фондот, ние можеме да добиеме податоци за географската локација, работното време, листата на достапни лекови, како и прецизни и опширни детали за секој од лековите. Овие податоци не се достапни во оригиналните изворни податочни

множества, односно нивната трансформација во поврзани податоци со 5-star податочен квалитет значително ја зголемува нивната употребливост и вредност. Во продолжение, ќе претставиме неколку кориснички сценарија кои имаат за цел да демонстрираат оваа зголемена вредност на податоците од ФЗОМ.

Прво корисничко сценарио. Едно можно корисничко сценарио би било лоцирање на аптека од одреден град, која е дежурна 24 часа во текот на конкретен ден. Вакво сценарио би можело да се искористи во мобилна апликација, на пример, која би ја лоцирала најблиската дежурна аптека, врз база на тековната локација на корисникот. Во следново SPARQL прашање ја наоѓаме аптеката која работи 24 часа во текот на 2 февруари 2014 година, во Битола:

SPARQL прашање 4.15

```
PREFIX hifm: <http://www.fzo.org.mk/ontology/hifm#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>

SELECT ?pharmacyName ?lat ?long ?phone ?notes
FROM <http://linkeddata.finki.ukim.mk/lod/data/hifmpharm#>
WHERE {
   ?pharmacy rdf:type schema:MedicalOrganization ;
             hifm:medicalFacilityName ?pharmacyName ;
             geo:city ?city ;
             geo:latitude ?lat ;
             geo:longitude ?long ;
             hifm:dateOnDuties ?date ;
             vcard:hasTelephone ?phone ;
             vcard:hasNote ?notes .
   FILTER (contains(lcase(?city),'bitola') && str(?date)='2/22/2014')
}
```

Табела 4.30: Резултати од SPARQL прашањето

Својство	**Вредност**
Име на мед. установа	PZU Apteka Medika Karta
Географска ширина	41.02503967285156
Географска должина	21.31836891174316
Телефон	047/225-285
Забелешка	from 23:00h to 07:00h

Резултатот од извршувањето на прашањето над нашето податочно множество е даден во Табела 4.30.

Второ корисничко сценарио. Друго корисничко сценарио би можело да ги опфати сите три изворни податочни множества од ова истражување. Со него, на пример, би можеле да ги добиеме информациите за лековите достапни за купување во конкретна аптека. Ваквото сценарио би можело да се искористи во апликации за лекови, со цел да го извести корисникот дали бараниот лек е достапен во некоја конкретна аптека или, пак, да му ја лоцира најблиската аптека во која е достапен лекот. Во продолжение е дадено пример SPARQL прашање со кое можеме да добиеме скратена листа на лекови достапни во една конкретна аптека (ID=25046508):

SPARQL прашање 4.16

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

SELECT ?label ?dosage ?strength ?quantity
FROM <http://linkeddata.finki.ukim.mk/lod/data/hifmpharm#>
WHERE {
   ?pharmacy hifm:id '25046508' ;
             hifm:hasAvaliableMedicine ?drug .
   ?drug hifm:available ?quantity ;
         hifm:dosageForm ?dosage ;
         hifm:strength ?strength ;
         rdfs:label ?label .
}
LIMIT 10
```

Резултатите од прашањето се прикажани во Табела 4.31.

Табела 4.31: Резултати од SPARQL прашањето

Име на лек	**Форма**	**Јачина**	**Количина**
FURAL	Capsules	30X100MG	2
FURAL	Suspension	90ML	2
GENTAMICIN	AMP.	10X40MG	2
MENDILEX	Tablets	2MG.X50	2
NIFLAM	Tablets	20X200MG	2
PROCULIN	TEARS SOL.	10ML	2
REGLAN	Syrup	120ML.	2
SUMETRIN	Tablets	50MGX3	2
TIMOLOL	Eye drops	0.5% 5ML	2
VASOFLEX	Tablets	30X1MG	2

Трето корисничко сценарио. Бидејќи во текот на трансформацијата на податоците во поврзани податоци додадовме RDF тројки кои репрезентираат линкови помеѓу нашето податочно множество и податочното множество од претходното истражување, кое понатаму е поврзано со ентитети од DrugBank податочното множество, можеме овие линкови да ги искористиме во ново корисничко сценарио. Во него би можеле, на пример, да добиеме дополнителен опис за конкретен лек кој е од интерес за корисникот. Бидејќи ваквите детали не се достапни во оригиналните изворни податочни множества од Фондот, ќе треба да изминеме неколку податочни множества од LOD облакот за да ги добиеме. Тоа можеме да го изведеме благодарение на поврзаната природа на податочните множества и концептот на SPARQL здружување на прашања.

Во продолжение е дадено пример SPARQL прашање преку кое за конкретен лек (ID=25046508) од нашето податочно множество ги добиваме описите кои постојат за него и неговата генерика во DrugBank и DBpedia податочните множества:

SPARQL прашање 4.17

```
PREFIX hifm: <http://www.fzo.org.mk/ontology/hifm#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dbpedia-owl: <http://dbpedia.org/owl/>
PREFIX drugbank: <http://wifo5-04.informatik.uni-mannheim.de/
                                    drugbank/resource/drugbank/>

SELECT DISTINCT ?name str(?dbdesc) str(?dpdesc)
WHERE {
   graph <http://linkeddata.finki.ukim.mk/lod/data/hifmpharm#> {
      ?pharmacy hifm:pharmacyID '25046508' ;
                hifm:hasAvaliableMedicine ?drug .
      ?drug owl:sameAs ?hifmDrug.
   }
   graph <http://linkeddata.finki.ukim.mk/lod/data/hifm#> {
      ?hifmDrug rdfs:seeAlso ?dbdrug ;
                drugbank:genericName ?name .
   }
   SERVICE <http://wifo5-04.informatik.uni-mannheim.de/drugbank/sparql> {
      ?dbdrug drugbank:description ?dbdesc ;
              owl:sameAs ?dpdrug .
   }
   SERVICE <http://dbpedia.org/sparql> {
      ?dpdrug dbpedia-owl:abstract ?dpdesc .
      FILTER langMatches( lang(?dpdesc), "EN" )
   }
}
```

Резултатите од прашањето се дадени во Табела 4.32.

Табела 4.32: Резултати од SPARQL прашањето

Име	Опис од DrugBank	Опис од DBpedia
Amoxicillin	A broad-spectrum semisynthetic antibiotic similar to ampicillin except that its resistance to gastric acid permits higher serum levels with oral administration. [PubChem]	Amoxicillin, formerly amoxycillin, and abbreviated amox, is a moderate-spectrum, bacteriolytic, β-lactam antibiotic used to treat bacterial infections caused by susceptible microorganisms...

Наведените кориснички сценарија, како и сите други над податочното множество, можат да се искористат од страна на мобилни и веб апликации користејќи го SPARQL endpoint-от [57] како REST-базиран сервис:

```
http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

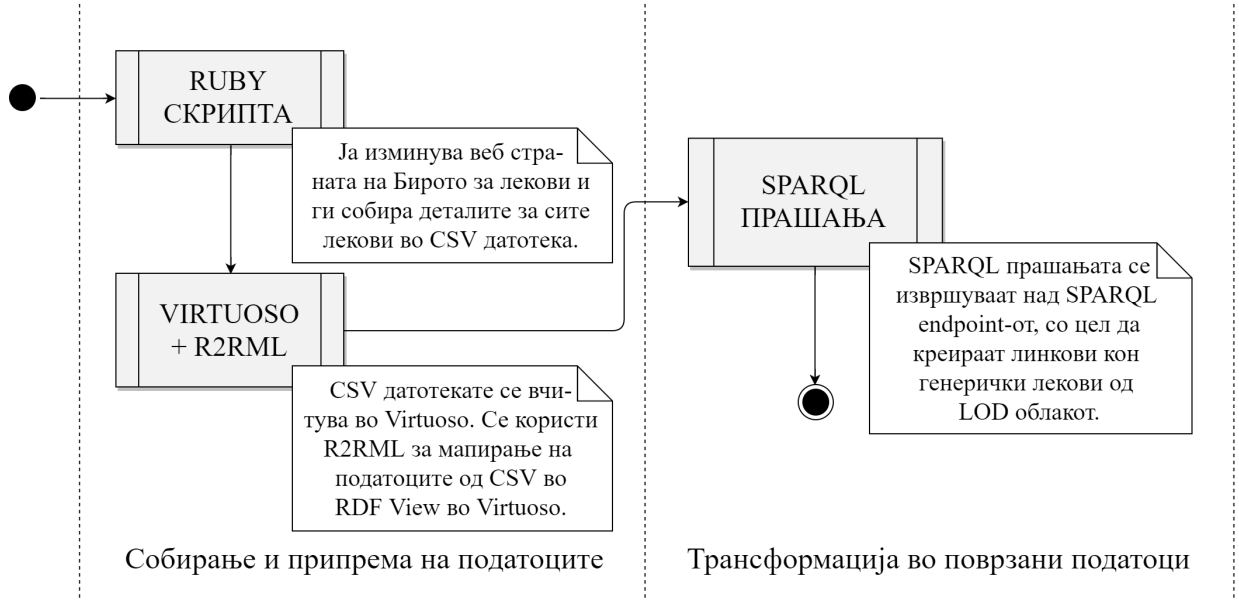
Тука, `SPARQLQUERY` го претставува SPARQL прашањето во URL-кодиран формат, додека `FORMAT` се однесува на посакуваниот формат на резултатите од прашањето. Форматот може да биде HTML, XML, JSON, Javascript, CSV, Spreadsheet, RDF/XML, N3, Turtle, итн.

4.5.3 Поврзани податоци за лекови во Македонија: Биро за лекови

Истражувањето во кое работевме на трансформација на податоците за лекови од ФЗОМ демонстрираше напредни кориснички сценарија овозможени од поврзаноста на генерираното податочно множество со лекови со ентитети од податочните множества на LOD облакот. Градејќи се на тоа искуство и откако Бирото за лекови на Република Македонија го објави комплетниот регистар на лекови регистрирани за продажба и достапни во Македонија, започнавме ново истражување кое имаше за цел да ги имплементира истите принципи над ова ново податочно множество. Регистарот на лекови објавен од Бирото за лекови содржи комплетна листа на регистрирани лекови, со многу поголем број податоци за секој од нив: продажно име на кирилица и латиница, бар код, информација за тоа дали лекот е на позитивната листа, детали за лиценцата на лекот, како и линкови кон PDF документи кои ги содржат корисничките упатства за лекот и извештаите за него од Бирото. Големината на податочното множество, споредена со податоците од ФЗОМ, беше мотив да развиеме автоматизиран и самоодржлив систем кој ги трансформира податоците за лекови од Бирото во висококвалитетни, 5-star поврзани податоци, како и да развиеме корисничка мобилна апликација која ќе ги демонстрира предностите на ваквото поврзано податочно множество [138].

Трансформација на податоците

За да дизајнираме автоматизиран систем за консолидирање на податоците за лекови, го одбравме податочното множество за лекови регистрирани во Македонија, одржувано и објавено од страна на Бирото за лекови на Република Македонија. Овие податоци се достапни преку веб страната на Бирото [38], во структуриран формат наменет за употреба од страна на луѓе. Веб страната обезбедува кориснички интерфејс за пребарување низ податоците за лековите регистрирани во Македонија, како и можност за преглед на деталите за секој од овие лекови. Автоматизираниот процес на трансформација на овие податоци во 5-star поврзани податоци беше сличен како и кај претходните проекти (Слика 4.20).



Слика 4.20: Работен тек на автоматизираната трансформација на податоците.

Собирање и припрема на податоците. Првиот чекор секогаш се состои од собирање на податоците од изворното податочно множество. За таа цел, креиравме сопствено софтверско решение во форма на Ruby скрипта, која на однапред одредени интервали ги изминува податоците од веб страната на Бирото, ги чита потребните податоци, ги прочистува според однапред изготвени трансформации и ги зачувува локално во CSV формат (Слика 4.20). Резултантните CSV датотеки од изминувањето на веб страната имаат предефинирана и точно одредена структура, која одговара на податоците достапни на веб страната на Бирото за лекови.

Како и поголемиот број податочни множества достапни преку Вебот, податоците за лекови од Бирото за лекови не се целосно прочистени. Поради тоа, Ruby скриптата содржи програмска логика која покрај собирањето на податоци реализира и мали трансформации за нивно прочистување. За генерирање на URI идентификаторот на секој од лековите ја идентификуваме ID вредноста на лекот според неговата URL адреса на веб страната на Бирото. Бројот на вредности во полињата за ‘генеричко име’ и ‘произ-

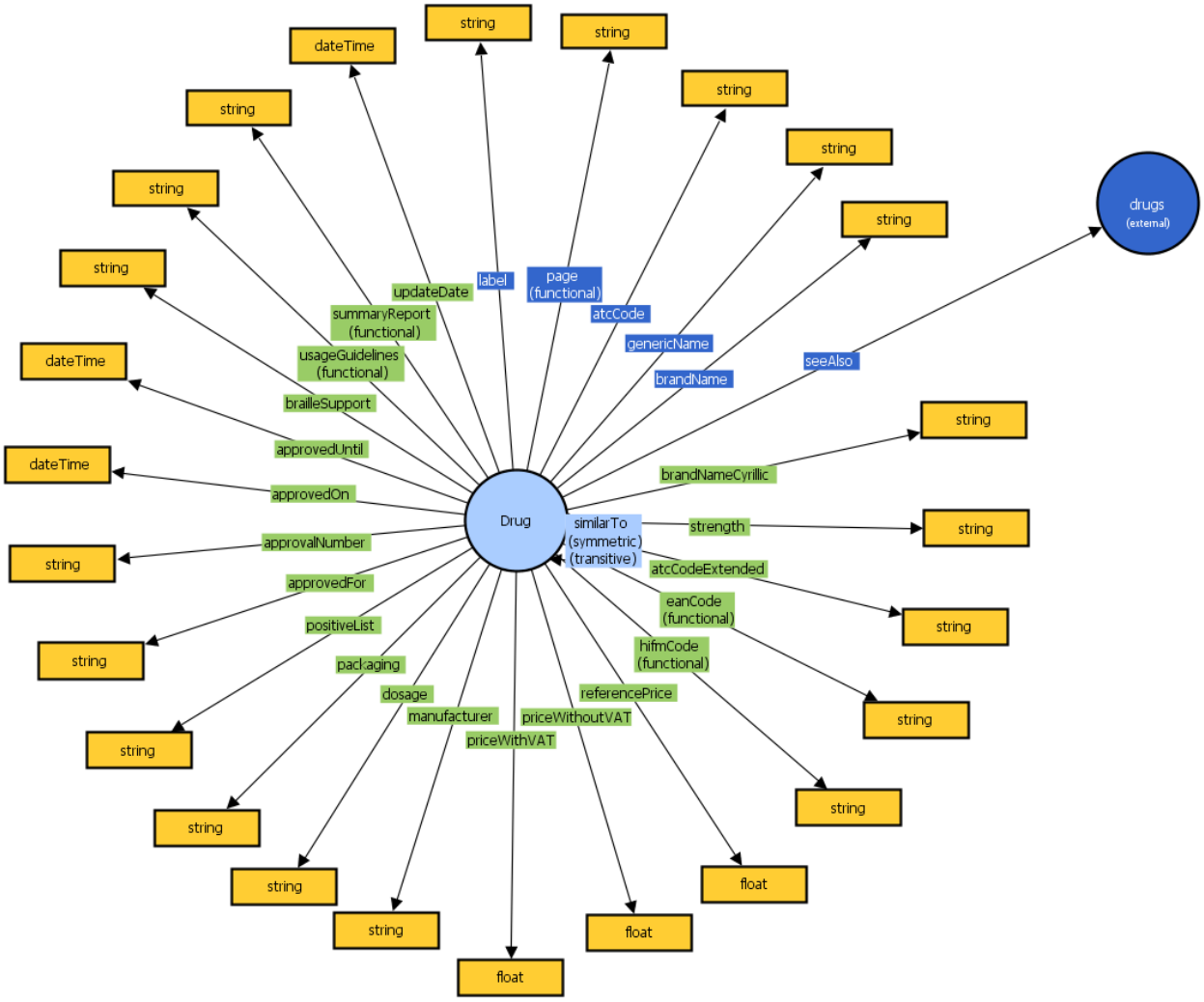
водител(и)' варира од еден лек до друг, поради што посебните вредности ги одвојуваме во посебни колони во CSV датотеката. Ова на крајот резултира во повеќе од една RDF тројка која го означува генеричкото име, односно производителот на лекот, со што се обезбедува потребната грануларност за пребарување низ податочното множество. Од друга страна, дел од полињата, како на пример бар кодот, референтна цена и состојките често имаат непотребни празни карактери пред или после нивните вредности, поради што е потребна корекција од страна на скриптата. Вредноста на датумите исто така се корегира во `YYYY-MM-DD` формат, а се прави и замена на децималните ознаки од запирки (,) во точки (.). По собирањето и чистењето на податоците, скриптата ги зачувува податоците во локална CSV датотека. Оваа CSV датотека потоа се вчитува во локална Virtuoso инстанца, со помош на BASH скрипта, при што содржината на датотеката се зачувува во класична релациона база на податоци.

DBM онтологија. За да ги моделираме податоците од податочното множество од Бирото за лекови како RDF податоци, потребен беше соодветен вокабулар, односно онтологија. Следејќи ги најдобрите практики за дизајн на онтологии, започнавме со анализа на онтологиите од доменот, селекција на постоечките класи и својства кои одговараа за податочното множество и продолживме со развој на сопствена онтологија со својствата кои недостасуваа.

Анализата на податоците од веб страната на Бирото за лекови покажа дека можеме да ги искористиме `drugbank:atcCode`, `drugbank:brandName` и `drugbank:genericName`, својства дефинирани во DrugBank онтологијата која се користи за анотација на податочното множество DrugBank [62]. Дополнително, ги искористивме и својствата `rdfs:label`, `foaf:page` и `rdfs:seeAlso` за означување на името на лекот, за уникатната веб страна за лекот од Бирото за лекови и за поврзување на лекот со соодветен генерички лек од LOD облакот. Овие својства се едни од најкористените својства во рамките на податочните множества од LOD облакот [33, 193].

Но, и покрај тоа, во постоечките онтологии и вокабулари од доменот недостасуваа својства за опишување на останатите дваесетина типови вредности за лековите од Бирото. Поради тоа, ја развивме DBM онтологијата (Слика 4.21). Онтологијата се состои од `dbm:Drug` класата, наменета за опис на лековите од податочното множество. Во онтологијата се дефинирани 20 податочни својства, односно својства кои ги опишуваат производителот на лекот, цената, дозирањето, јачината, итн. Во DBM онтологијата е дефинирано и објектното својство `dbm:similarTo`, чија намена е поврзување на две `dbm:Drug` инстанци кои ја имаат истата функција и активна супстанца, односно се наменети за истата состојба на пациентот. Ова својство е дефинирано како симетрично и транзитивно. Подолу во текстот ќе објасниме подетално за процесот на генерирање RDF тројки со ова својство, како и негово искористување во конкретни кориснички сценарија.

DBM онтологијата е објавена согласно најдобрите практики на W3C [88]. Достапна е преку перзистентно URI кое овозможува Веб пристап и HTTP content negotiation: `http://purl.org/net/dbm/ontology#`.



Слика 4.21: DBM онтологијата и помошните класи од постоечките онтологии.

Трансформација во поврзани податоци. Вчитаната CSV датотека од претходниот чекор се наоѓа во релациона база на податоци во Virtuoso. Како следен чекор, со помош на BASH скрипта се активира процедура во Virtuoso која го користи R2RML мапирачкиот јазик за да ги трансформира податоците од релационата база на податоци во RDF поглед, т.е. за да генерира податоци во RDF формат. Со цел да се изврши трансформацијата, релационите податоци треба да се трансформираат во соодветен RDF модел, за што се користи мапирачка датотека. Оваа мапирачка датотека ја користи нашата DBM онтологија и соодветниот RDF вокабулар од Слика 4.21. Како резултат, се добива RDF граф во рамките на Virtuoso инстанцата, кој се состои од податоците собрани од веб страната на Бирото за лекови, овој пат во RDF формат. По ова, може да започне трансформацијата на податоците во поврзани податоци.

Трансформацијата на податочното множество во поврзано податочно множество ја имплементираме преку креирање врски помеѓу лековите од самото податочно множество, како и врски помеѓу лековите од множеството со лекови од други податочни мно-

жества од LOD облакот. За првото, креираме RDF тројки помеѓу лекови од податочното множество кои имаат ист ATC код, а со тоа имаат и иста активна супстанца, односно функција. За второто, пак, поврзуваме лек од податочното множество со генерички лек од DrugBank податочното множество, кое е дел од LOD облакот. Притоа, повторно ја искористуваме вредноста на ATC кодот за да ги најдеме соодветните поврзувања кои треба да се реализираат преку RDF тројки.

Двете поврзувања ги реализираме со SPARQL INSERT прашања, кои за секој пронајден пар лекови од локалното податочно множество со ист ATC код креира две RDF тројки, како на пример:

Пример 4.1

```
@PREFIX dbm:  <http://purl.org/net/dbm/ontology#> .
@PREFIX dbm-drug:  <http://purl.org/net/dbm/data#> .

dbm-drug:54969  dbm:similarTo  dbm-drug:55476 .
dbm-drug:55476  dbm:similarTo  dbm-drug:54969 .
```

Овие две RDF тројки претставуваат *двонасочна врска* помеѓу лековите, означувајќи ја нивната сличност. Самото SPARQL INSERT прашања ги креира овие тројки во истиот RDF граф во Virtuoso во кој се наоѓа трансформираното податочно множество. Во едно од последните извршувања на работниот тек (Слика 4.20), 33.872 `dbm:similarTo` тројки беа додадени во овој чекор, т.е. 16.936 парови на лекови од Бирото за лекови беа идентификувани како лекови кои ја имаат истата функција. Овие дополнителни врски помеѓу лековите ги користиме во корисничките сценарија во кои на корисниците им се понудуваат алтернативни лекови за нивната потреба.

За креирање врски помеѓу лековите од Бирото и генеричките лекови од LOD облакот, одлучивме да се поврземе со инстанците од DrugBank податочното множество [62], како и во претходното истражување. Во DrugBank податочното множество генеричките лекови имаат свој ATC код, што овозможи користење на SPARQL INSERT прашање слично на она од претходниот чекор, со цел да се идентификуваат генеричките лекови кои одговараат на локалните лекови од податочното множество и да се креираат соодветни RDF тројки како врска помеѓу нив:

Пример 4.2

```
@PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@PREFIX dbm-drug:  <http://purl.org/net/dbm/data#> .
@PREFIX drugbank-drug:
  <http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/> .

dbm-drug:56093  rdfs:seeAlso  drugbank-drug:DB00016 .
```

Оваа пример RDF тројка означува дека локалниот лек `56093` од податочното множество на Бирото за лекови ја има истата функција како и генеричкиот лек `DB00016` од DrugBank податочното множество. Овие дополнителни RDF тројки се зачувуваат во истиот RDF граф со трансформираното податочно множество. Едно од последните извршувања на автоматизираниот работен тек резултираше со 2.791 вакви тројки, кои поврзуваат лек од Бирото за лекови со генерички лек од DrugBank. Овие дополнителни врски го трансформираат податочното множество во поврзано податочно множество. Тие овозможуваат проширување на SPARQL прашањата кои можат да се поставуваат над податочното множество, бидејќи со користење на концептот на SPARQL здружување на прашања може да се добие пристап до податоците од DrugBank кои одговараат на тековно разгледуваниот лек од Бирото за лекови и со тоа да се прошири спектарот на достапните информации во контекстот во кој работи и пребарува корисникот.

SPARQL INSERT прашањата кои ги изведуваат овие две поврзувања се извршуваат автоматски и нивните резултати се запишуваат назад во оригиналниот RDF граф. Со ова завршува автоматизираниот работен тек (Слика 4.20) за креирање на поврзаното податочно множество на лекови од Бирото за лекови на Република Македонија. Овој работен тек се активира на претходно дефиниран интервал, иницијално поставен на еднаш неделно, со што сме сигурни дека сите промени во податоците од страна на Бирото би биле навремено ажурирани во поврзаното податочно множество.

Генерираното поврзано податочно множество и неговите последно ажурирани податоци се објавуваат и се достапни на Веб согласно најдобрите практики за објавување на поврзани податоци [126], преку перманентно URI достапно преку Веб, кое поддржува HTTP content negotiation: `http://purl.org/net/dbm/data#`.

Поврзаното податочно множество е хостирано од страна на јавна Virtuoso инстанца, во јавно достапен RDF граф со идентификатор `<http://linkeddata.finki.ukim.mk/lod/data/dbm#>`, преку SPARQL endpoint-от [57] на Virtuoso инстанцата.

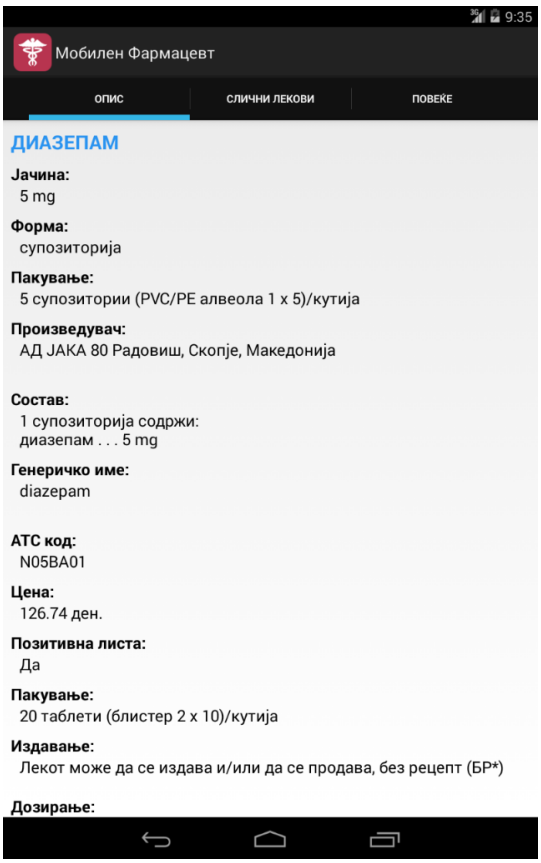
Мобилна апликација "Мобилен Фармацевт"

Откако го добивме поврзаното податочно множество со податоци за лекови од Бирото за лекови на Република Македонија, следниот чекор беше демонстрирање на предностите од ваквиот формат на податоците. За таа цел, ја дизајниравме и развивме мобилната апликација "Мобилен Фармацевт". Станува збор за Android мобилна апликација која како податочен слој го користи поврзаното податочно множество. Апликацијата пристапува до податоците објавени на јавната Virtuoso инстанца [66], користејќи стандардни HTTP GET и POST повици до SPARQL endpoint-от, преку кој се селектираат потребните податоци. Ваквиот дизајн гарантира дека корисниците на апликацијата секогаш ќе имаат пристап до најновите податоци. Секако, податоците се кешираат и локално на мобилниот телефон, за подобрување на перформансите. Со оглед на фактот дека податочното множество преку автоматизираниот работен тек се обновува еднаш неделно, кеширањето не влијае на пристапот до најнови податоци.

Основниот екран на мобилната апликација прикажува влезно поле во кое корисни-

ците го внесуваат името на лекот од интерес (Слика 4.22).

Слика 4.22: "Мобилен Фармацевт": Екран за пребарување.



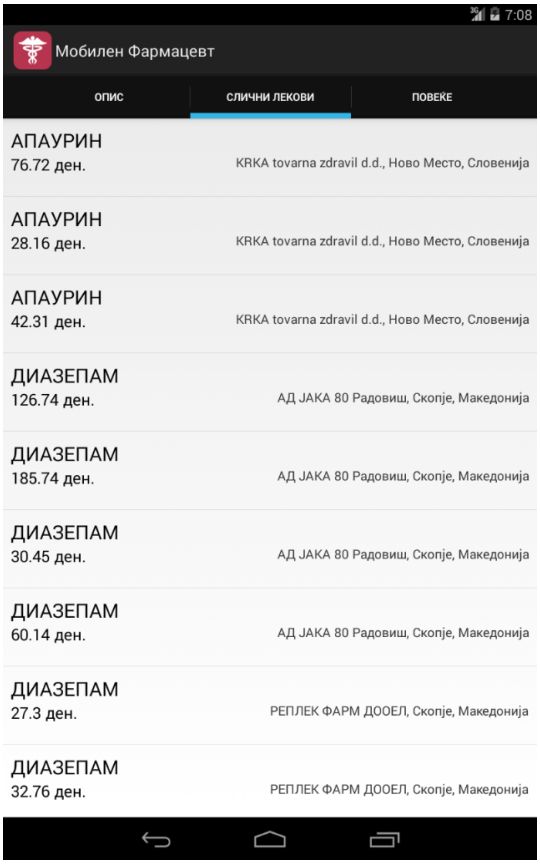
Слика 4.23: "Мобилен Фармацевт": Екран со опис на лек.

Откако корисникот ќе селектира лек од регистарот на лекови на Бирото, апликацијата го прикажува екранот со опис. На тој екран се презентираат деталите и карактеристиките на лекот достапни во оригиналното податочно множество од Бирото (Слика 4.23). Корисникот тука може да го види името на лекот на кирилица и латиница, генеричкото име, јачината, пакувањето, состојките, формата на издавање, дозирањето, дополнителните предупредувања, листата производители, информации за лиценцата и одобрението од Бирото, цената, како и линкови до PDF документи кои ги содржат упатствата за користење и извештајот од Бирото. Апликацијата го прикажува и датумот кога податоците за лекот последен пат биле ажуриран во нашиот систем, за оценување на релевантноста на податоците.

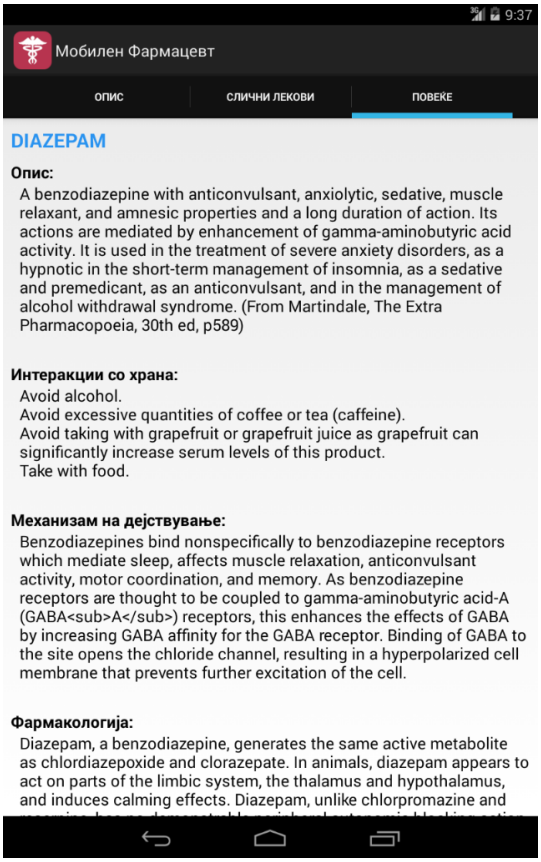
Преку овој екран, корисниците практично ги добиваат податоците од веб страната на Бирото за лекови во мобилен формат.

Следниот екран во апликацијата ги прикажува лековите слични на селектираниот (Слика 4.24). На овој екран се прикажуваат информациите добиени преку искористување на `dbm:similarTo` релациите кои селектираниот лек ги има со останатите лекови во податочното множество. Како што веќе објаснивме, овие релации ги означуваат лековите кои го имаат истиот АТС код, односно ја истата активна супстанца, а со тоа и

функција. Според тоа, ова екран овозможува пристап до информации за лековите кои во одредена мера можат да го заменат лекот од интерес на корисникот, што значи може да им помогне на фармацевтите и лекарите во одредување на алтернативен лек кој може да се употреби како дел од тековната терапија на пациентот. Покрај имињата на алтернативните лекови, на овој екран се прикажуваат и нивните соодветни цени, според Бирото за лекови. Оваа информација може дополнително да игра улога во селекцијата на соодветниот лек за терапијата на пациентот.



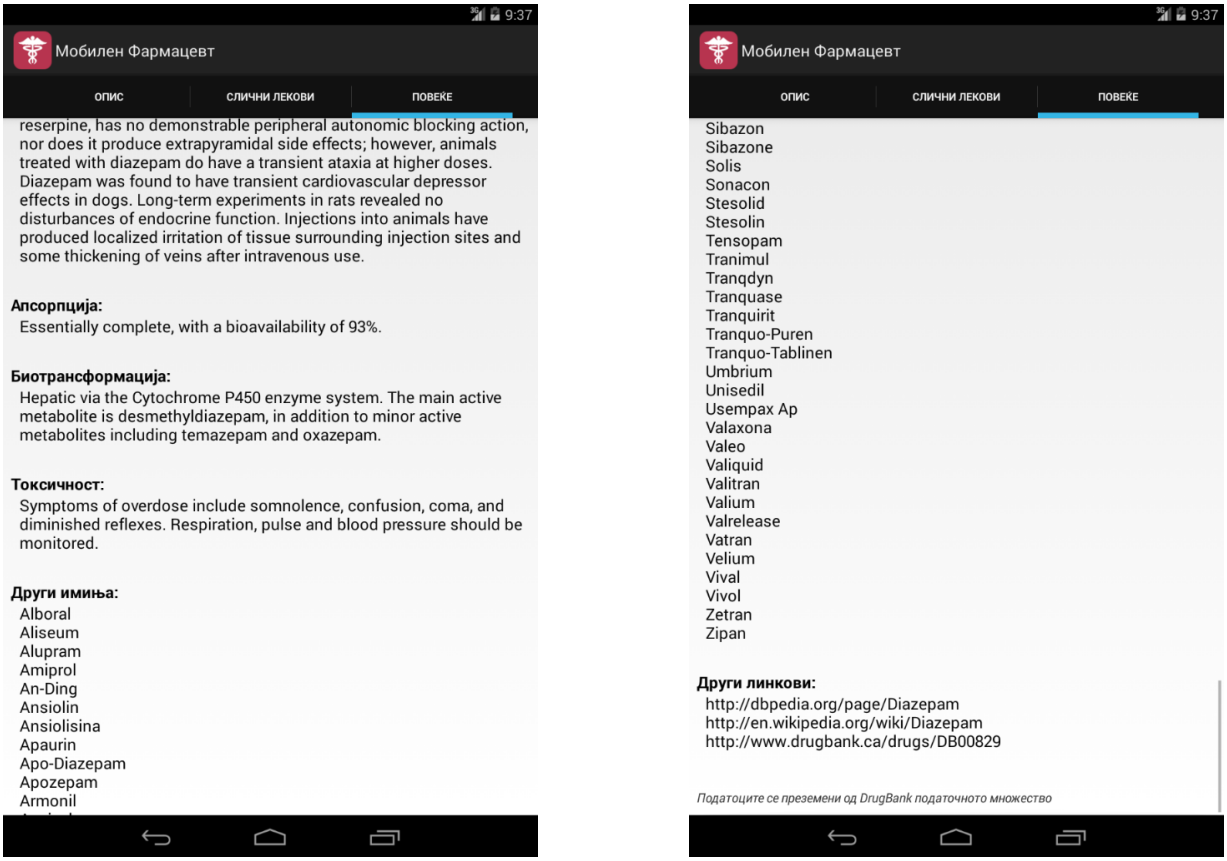
Слика 4.24: "Мобилен Фармацевт": Екран со слични лекови.



Слика 4.25: "Мобилен Фармацевт": Екран со дополнителен опис (дел 1).

Третиот екран за селектираниот лек е екранот со дополнителни информации и опис (Слика 4.25 и 4.26). Тука, апликацијата ги користи `rdfs:seeAlso` релациите на лекот присутни во самото податочно множество, кои го поврзуваат лекот со инстанци од генерички лекови од DrugBank податочното множество, врз база на ATC кодот. Апликацијата користи SPARQL прашање во кое со помош на концептот на SPARQL здружување на прашања, дел од прашањето се префрла од SPARQL endpoint-от на нашата Virtuoso инстанца до SPARQL endpoint-от на DrugBank од каде што се селектираат потребните податоци за контекстно поврзаниот генерички лек. Податоците кои се селектираат и прикажуваат на овој екран се однесуваат на дополнителен опис на лекот, интеракциите на лекот со други лекови, интеракциите на лекот со храна, механизмот на дејство на лекот, фармакологијата на лекот, апсорпцијата, биотрансформацијата, токсичноста,

листата на алтернативни продажни имиња на лекот, како и листа на веб страни каде можат да се најдат повеќе информации за него. Овие податоци не се достапни на изворната веб страна на Бирото за лекови на Република Македонија, односно не се дел од јавниот онлајн регистар на лекови. Сепак, контекстната поврзаност на лековите од нашето податочно множество со генерички лекови од DrugBank, овозможуваат пристап до ваков тип дополнителни информации за лековите кои се дел од регистарот на лекови на Бирото.



Слика 4.26: “Мобилен Фармацевт”: Екран со дополнителен опис (дел 2).

Преку ова корисничко сценарио директно се демонстрираат предностите од форматот на поврзаните податоци како и целта на истражувањето: пребарувајќи низ локални податоци од Бирото, да стигнеме до дополнителни податоци и информации кои се достапни на Вебот во формат на поврзани податоци, а се објавени и одржувани од страна на други институции. Со оглед на тоа што DrugBank податочното множество е поврзано податочно множество, т.е. содржи линкови кои ги поврзуваат неговите ентитети со други ентитети од LOD облакот, можеме да употребуваме кориснички сценарија во кои тргнувајќи од лековите од Бирото за лекови на Република Македонија, преминуваме во податочното множество на DrugBank, а од таму продолжуваме понатаму во други податочно множества од LOD облакот, согласно потребите на сценариото.

4.5.4 Глобалното влијание на националните кујни врз лековите

Контекстното поврзување на податочни множества, покрај овозможувањето нови, понапредни кориснички сценарија, овозможува и нова анализа на податочните множества чии ентитети се поврзани согласно принципите на поврзани податоци. Придонесот на поврзаните податоци во аналитичките сценарија решивме да го тестираме во рамките на истражување во кое направивме преглед на негативното влијание кое различни национални кујни го имаат врз различни групи на лекови [137].

Интеракциите помеѓу храна и лекови се добро проучени, но малку е познато за интеракциите помеѓу националните кујни и лековите. Кујните кои не се локални за одредена земја стануваат достапни и популарни во неа. Во истражувањето направивме анализа на распределеноста на негативните влијанија на храната и лековите во различни национални кујни, на глобално ниво. Дополнително, го анализиравме и ефектот на конкретни состојки врз различните категории на лекови, во различни делови од светот. Целта на анализата беше потенцирање на важноста од интеракциите кои одредени национални кујни ги имаат со одредени категории лекови, за да се овозможи советување на пациентите кои примаат терапија од одредена категорија да избегнуваат одредени кујни [137].

Потребата за наоѓање и идентификување храна која придонесува за целокупното човеково здравје, ги задоволува нутритивните и енергетските потреби и истовремено не предизвикува труење, постои од почетокот на човештвото. Човековата исхрана, која на организмот му обезбедува есенцијални нутритивни елементи за неговото здравје, е под влијание на голем број различни фактори, како што се културолошките навики, социо-економскиот статус и климатските услови. На пример, употребата на зачини во топлите региони во светот е тесно поврзана со потребата од одржување на храната отпорна на бактерии подолг временски период [90].

Добро познат факт е дека одредена храна може да влијае на ефектот на даден лек во еден организам [180, 105, 115, 98]. Промените во биодостапноста (степенот и ратата со која даден лек се апсорбира во нечиј систем) на даден лек предизвикани од храна, го менуваат клиничкиот ефект на лекот. Генерално, интеракциите помеѓу храна и лекови можат да резултираат со значително намалување на биодостапноста на лекот, или преку директна интеракција помеѓу супстанца од храната и хемиска компонента од лекот, или преку физиолошкиот одговор на внесот на храна (на пример: лачење на желудочна киселина). Ова често може да доведе до неуспешно лекување. Дополнително, интеракциите помеѓу храна и лекови можат да резултираат и со зголемување на биодостапноста, или преку зголемување на способноста за растворање на лекот како директна последица од некоја супстанца во храната, или индиректно, преку лачење на желудочна киселина иницирано од храната. Иако ова води кон зголемување на ефектот на лекот, често пати може да резултира со сериозна токсичност [180].

Најпродаваните лекови во светот вклучуваат антинеопластици и имуномодулатори, лекови за респираторниот систем, лекови за системот за варење и метаболизам, лекови за кардиоваскуларниот систем и лекови за нервниот систем [146]. Статистиката пока-

жува дека скоро 70% од популацијата во САД консумира најмалку еден лек добиен на рецепта, бројка која била само 48% во 2010 година. Дваесет проценти од нив се на терапија со пет или повеќе лекови [101]. Според [102], алармантна бројка од 1,5 милиони луѓе имаат последици од лекови, вклучувајќи ги и грешките настанати поради недоволно информации добиени од фармацевтите или несовесноста на пациентите да ги прочитаат и следат упатствата за употреба на лековите.

Различни региони во светот користат различни состојки и храна како дел од нивните кујни, па поради тоа негативните интеракции кај лековите предизвикани од храна варираат од едно до друго место во светот. Дополнително, со растот на популарноста на странски кујни (поради патување или поради нивното ширење во сите делови од светот), ефектот кој различните кујни го имаат над одредени лекови и категории на лекови станува многу важен.

Поврзани податочни множества за лекови и храна

Со цел да обезбедиме релевантна анализа, потребно беше користење на реални податоци за лекови и рецепти. Различни фактори влијаеа на одлуката за податочните множества кои ќе ги користиме: валидноста на податоците, обемот и нивната временска релевантност со тековните состојби. Во продолжение ќе го претставиме процесот на селекција, собирање и трансформација на двете податочни множества користени во анализата. Со цел да ги доведеме на исто репрезентациско ниво, податочните множества моравме да ги трансформираме и контекстно да ги поврземе, согласно принципите на поврзани податоци [122].

Поврзано податочно множество за лекови. Во рамките на LOD облакот, како резултат од повеќе независни истражувања и проекти, генерирани се неколку поврзани податочни множества со податоци од областа на здравството. Истражувачката група Semantic Web Health Care and Life Sciences (HCLS) Interest Group [53], која е дел од WWW конзорциумот, е фокусирана на истражувања кои ги користат технологиите на Семантичкиот Веб и поврзаните податоци во рамките на здравството, здравствените науки, клиничките истражувања и медицината. Како резултат од нивната работа, групата има генерирано податочни множества во рамки на т.н. Linked Open Drug Data (LODD) облак, кој воедно е дел и од LOD облакот [175]. Во рамките на LODD облакот постојат над 380 милиони RDF тројки [35].

Дел од LODD облакот е и веќе познатото DrugBank податочно множество со поврзани податоци за генерички лекови [62]. Од неговата иницијална објава во 2006 година, DrugBank податочното множество е често употребувано за истражувања од страна на фармацевти, хемичари, фармацевтски истражувачи, податочни научници, итн. [151]. Поради ова, но и нашето претходно искуство во работа со ова податочно множество, го селектиравме за нашата анализа.

Покрај податоците за генерички лекови кои ги видовме во претходните истражувања, DrugBank податочното множество содржи и информации за интеракциите на секој лек со одредена храна. Во него има вкупно 968 храна - лек интеракции, кои поврзу-

ваат 525 различни лекови со различни индикации со храна. Овие индикации содржат референца кон една или повеќе состојки и најчесто се негативни, како на пример "Избегнувајте алкохол.", или "Да не се зема со млеко.". Сепак, постојат и случаи кога интеракцијата на лекот со храна е неутрална ("Да се зема без обзир кон оброците."), како и случаи кога интеракцијата е всушност позитивна ("Зголемете го внесот на магнезиум, фолијати, витамин B6 и B12 преку храна и/или земете мултивитамини.").

Поради тоа, потребно беше прецизно да го означиме сентиментот на секоја храна - лек интеракција. За таа цел, обезбедивме локална копија на дел од DrugBank податочното множество, дел кој беше доволен за анализата и во него креиравме нови RDF својства со кои ги означуваме различните типови интеракции: негативни, неутрални и позитивни. Анализата на сентиментот на секоја од интеракциите ја изведовме полу-автоматски. Потоа, за секој лек од податочното множество ги анализиравме негативните интеракции со храна, од нив ги детектиравме споменатите состојки, кои понатаму ги лоциравме во конкретни рецепти од податочното множество со рецепти. За секој таков лек - рецепт пар додаваме нова релација во податочното множество со лекови, означувајќи дека лекот има негативна интеракција со конкретниот рецепт. Со тоа, добиваме поврзани податочни множества, кои ги следат принципите на поврзани податоци. Податочните множества ги сместивме во јавно достапна Virtuoso инстанца [66]. Со користење на SPARQL прашања поставени над SPARQL endpoint-от на наведената Virtuoso инстанца, можевме да пристапуваме до податоците од двете податочни множества и да ги извлечеме податоците потребни за анализата.

Нашата проширена верзија на DrugBank податочното множество е објавена следејќи ги принципите на поврзани податоци и е достапна преку нашиот јавен SPARQL endpoint [57]. Дополнително, податочното множество е достапно и преку Datahub податочниот портал [131].

Поврзано податочно множество за рецепти. Со растот на Вебот и присуството на мобилните и веб апликации во секојдневието, бројот на онлајн рецепти и податочни множества со рецепти е значително пораснат, што овозможува лесен и брз пристап до милиони рецепти од различни глобални кујни. Дел од рецептите на Вебот се достапни како комерцијални податочни множества и се наменети за користење во рамките на мобилни апликации: Yammly[1], Food2Fork[2], BigOven[3], додека други се достапни на веб страници и се слободни за користење: AllRecipes.com[4], Epicurious[5], Taste.com.au[6], FoodNetwork.com[7], итн.

За потребите на нашата анализа го искористивме податочното множество со рецепти обезбедено во [75]. Податочното множество е креирано користејќи рецепти од три различни извори: *allrecipes.com*, *epicurious.com* и *menupan.com*. Во него има податоци

[1] `https://developer.yummly.com/`
[2] `http://food2fork.com/about/api`
[3] `http://api.bigoven.com/`
[4] `http://allrecipes.com`
[5] `http://www.epicurious.com/`
[6] `http://www.taste.com.au/`
[7] `http://www.foodnetwork.com/`

за вкупно 56.458 рецепти, нивните состојки и кујните на кои припаѓаат. Рецептите се поделени во 11 кујни: Северна Америка (41.525), Западна Европа (2.659), Источна Европа (381), Јужна Европа (4,180), Северна Европа (250), Блиски Исток (645), Јужна Азија (621), Југоисточна Азија (457), Источна Азија (2.512), Латинска Америка (2.917) и Африка (352).

Со цел да овозможиме интероперабилност помеѓу податочните множества, го трансформираме прочистеното множество од CSV формат во RDF формат. За RDF анотацијата ја искористивме Food онтологијата [20], која овозможи да ги означиме кујната и состојките за секој од рецептите во податочното множество. Податочното множество не го проширувавме или поврзувавме дополнително, поради тоа што податочното множество со лекови ги содржи RDF тројките кои претставуваат врски помеѓу двете податочни множества. Податочното множество со рецепти го објавивме на ист начин како и проширеното податочно множество со лекови, односно тоа е јавно достапно преку истиот SPARQL endpoint. Дополнително, податочното множество со рецепти е достапно и преку Datahub податочниот портал [132].

Анализа

По селекцијата, прочистувањето и трансформацијата во RDF и поврзани податоци, ги вчитавме податочните множества во јавната Virtuoso инстанца [66], која обезбедува REST-базиран пристап до податоците од двете податочни множества, во RDF формат. Анализата ја изведовме со користење на SPARQL endpoint-от за пребарување низ поврзаните податочни множества со лекови и рецепти. Користевме SPARQL прашања кои ги искористуваат RDF релациите од податочното множество со лекови кои ги поврзуваат лековите со конкретни рецепти кои имаат негативни интеракции со нив и ги анализиравме различните аспекти на доменот и резултатите кои произлегоа.

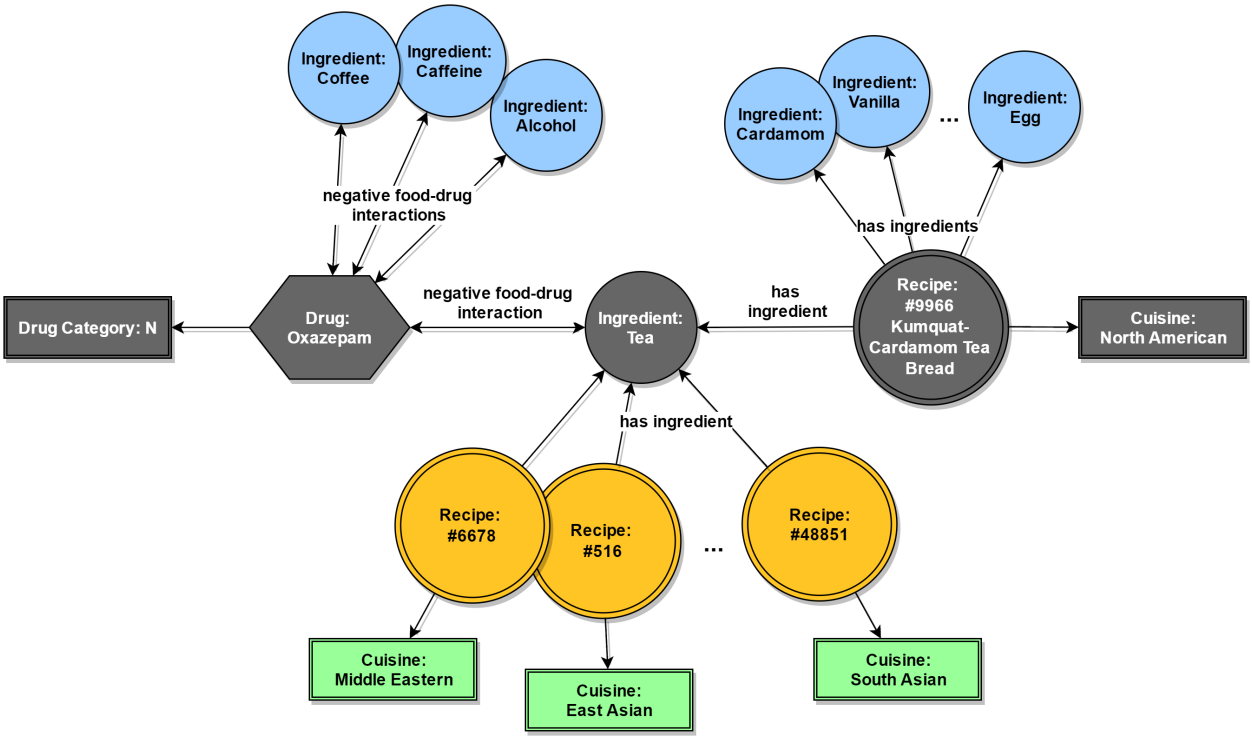
Односот на интеракции помеѓу категорија на лекови и кујна го мериме во промили, според:

$$(E_I/P_I) * 1000 \tag{4.1}$$

каде E_I е бројот на постоечки интеракции идентификувани во податочните множества, а P_I е бројот на можни интеракции помеѓу одредена категорија на лекови и одредена кујна, пресметана како бројот на лекови во категоријата помножен со бројот на рецепти во кујната. Ја користиме мерката на промили за овој однос со цел да го прикажеме бројот на пациенти, од 1.000 пациенти третирани со лек од дадена категорија, кои можат да имаат негативна интеракција со храна при консумирање на оброк од конкретната кујна. За пресметка на E_I, бројот на постоечки интеракции помеѓу конкретна категорија на лекови и конкретна кујна, го броиме бројот на постоечки негативни интеракции со храна кои лек од конкретната категорија ги има со рецепти од конкретната кујна.

За илустрација на оваа анализа, можеме да искористиме конкретен пример: интеракцијата помеѓу лекот 'Oxazepam' и чај (Слика 4.27). 'Oxazepam' е бензодиазепин кој

се користи за третман на анксиозни нарушувања, апстиненцијални симптоми при одвикнување од алкохол и несоница. Според DrugBank, тој има три клинички докажани храна - лек интеракции: (а) избегнувајте алкохол, (б) избегнувајте прекумерни количини на кафе или чај (кофеин) и (в) да се зема со храна. Во нашата анализа, заклучивме дека (а) и (б) се негативни интеракции на 'Oxazepam' со алкохол, кафе, чај и кофеин. Интеракцијата (в) ја сметаме за позитивна интеракција со храна и не ја земам предвид во анализата. 'Oxazepam' има ATC код `N05BA04`, со што спаѓа во ATC категоријата N.



Слика 4.27: Заклучената кујна - лек интеракција врз база на клинички познатата негативна интеракција помеѓу 'Oxazepam' и чај.

Од друга страна, чај се среќава како состојка во 102 различни рецепти од 8 различни кујни во нашето податочно множество. Еден од тие 102 рецепти е рецептот #9966, "Kumquat-Cardamom Tea Bread", кој припаѓа во кујната на Северна Америка и кој ги има следните состојки: кардамон, јајце, растително масло, путер, пченица, сок од лимон, ванила, орев, пченка, цуцест портокал, ананас и чај.

Во нашата анализа во овој случај заклучуваме дека состојката чај е одговорна за негативна кујна - лек интеракција помеѓу кујната на Северна Америка и N категоријата на лекови (Слика 4.27). Оттука ја броиме оваа кујна - категорија интеракција како една постоечка негативна интеракција.

Доколку во овој случај рецептот #9966, "Kumquat-Cardamom Tea Bread" содржеше и друга состојка која има негативна интеракција со лекот 'Oxazepam', како на пример алкохол или кафе, интеракцијата помеѓу рецептот (и кујната на Северна Америка) со 'Oxazepam' (и категоријата на лекови N) повторно ќе ја броевме како единствена

негативна интеракција. Ова го правиме поради тоа што нашата анализа се базира на соодносот помеѓу постоечките (E_I) и можните (P_I) негативни интеракции, при што можните негативни интеракции ги пресметуваме на лек - кујна ниво. Поради тоа, мораме да го пресметуваме бројот на постоечки негативни интеракции на истото тоа ниво.

Резултати од анализата

Во анализата разгледуваме два аспекти на храна - лек интеракциите: (1) негативните интеракции помеѓу лекови од дадена категорија и рецепти од дадена кујна, и (2) влијанието на состојките во негативните храна - лек интеракции во различни делови од светот.

Табела 4.33: Листа на ATC кодови

Код	**Опис**
A	Гастроинтестинален тракт и метаболизам
B	Крв и крвотворни органи
C	Кардиоваскуларен систем
D	Дерматолошки препарати
G	Генито-уринарен систем и полови хормони
H	Системски хормонски препарати
J	Антиинфективни лекови за системска употреба
L	Антинеопластични и имуномодулаторни агенси
M	Мускуло-скелетен систем
N	Нервен систем
P	Антипаразитски препарати, инсектициди и репеленти
R	Респираторен систем
S	Сензорни органи
V	Разновидно

Интеракции помеѓу кујни и категории на лекови. За анализа на негативните интеракции помеѓу одредена категорија на лекови, т.е. категорија на лекови групирани според ATC класификацијата (Табела 4.33) и одредена кујна, ги пресметуваме промилите на постоечки интеракции помеѓу нив (Табела 4.36) со користење на Формулата 4.1. Овој сооднос ја претставува веројатноста за појава на негативна интеракција со храна кога пациент кој прима терапија со лек од дадена категорија конзумира оброк кој спаѓа во дадена кујна. Користиме промили за соодносот со цел да го покажеме бројот на пациенти, од 1.000, кои можат да добијат негативна интеракција со храна при комбинирање на лек од соодветната категорија и храна од кујната. Целта на овој аспект

на анализата е да се идентификува негативното влијание од консумирањето храна од специфични кујни во период кога пациентот е под терапија.

Резултатите (Табела 4.36) покажуваат дека едни од најинтензивните храна - лек интеракции се појавуваат помеѓу лекови од B категоријата и рецепти од кујните на Азија, Африка и Латинска Америка, лековите од категоријата J и рецепти од Северна Америка и Европа, како и лекови од категоријата V и рецепти од скоро сите делови на светот. Исто така, можеме да забележиме дека лековите од категориите H, P и R многу ретко појавуваат негативни интеракции со храна, додека лековите од категорија M воопшто дури немаат негативни интеракции. Генерално, резултатите од Табела 4.36 овозможуваат распознавање на три различни шеми на храна - лек интеракции од гледна точка на кујните и категориите на лекови (Слика 4.28).

Шема 1. Првата шема се состои од лекови од категориите B, C, N и V. Како што може да се забележи од Табела 4.36, лековите од овие четири категории имаат значително повеќе негативни храна - лек интеракции со рецепти од Јужна Европа, Блиски Исток, Јужна Азија, Југоисточна Азија, Источна Азија, Латинска Америка и Африка, споредено со другите кујни. Оваа појава јасно е видлива на Слика 4.28.а.

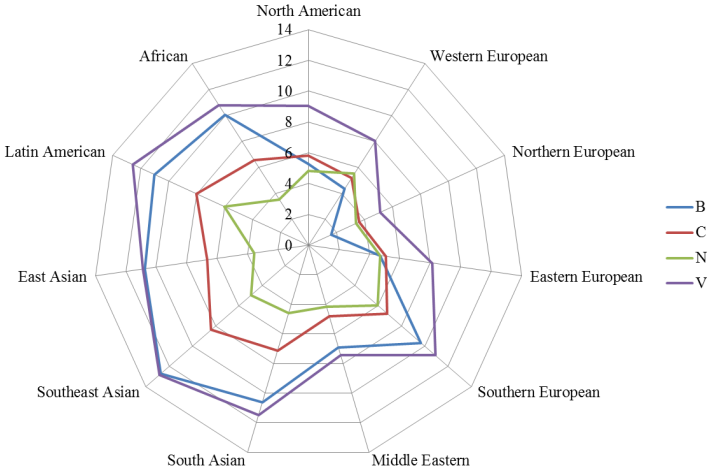
Причината зад оваа шема на влијание е фактот што лековите од овие четири категории имаат негативни интеракции со состојките лук и ѓумбир. Овие две состојки се значајно присутни и директно одговорни за негативните храна - лек интеракции во овие географски региони (Табела 4.35, Табела 4.37). Дополнително, овие лекови имаат негативни интеракции и со авокадо, сладунец и цитрон, што ја зголемува разликата помеѓу овие категории на лекови и останатите. Интеракциите со кафе се исто така присутни во овие категории, но тие се присутни и во останатите категории, па ефектот на овие интеракции не влијае значително во формирањето на шемата на влијание.

На Слика 4.28 е прикажан интензитетот на негативни храна - лек интеракции на лековите од категоријата B, прикажан преку скала од бои. Сликата го покажува бројот на пациенти од 1.000, кои се под терапија со лек од категоријата B, кои би можеле да имаат негативна интеракција со рецепт од дадена кујна. Белите делови на мапата претставуваат кујни за кои немаме податоци.

Бидејќи лековите од категорија B припаѓаат во првата шема, на Слика 4.29а го прикажува истиот интензитет на негативни храна - лек интеракции кои ги гледаме и на Слика 4.28а, т.е. лековите од категорија B имаат значително повеќе интеракции со рецепти од Јужна Европа, Блиски Исток, Јужна Азија, Југоисточна Азија, Источна Азија, Латинска Америка и Африка.

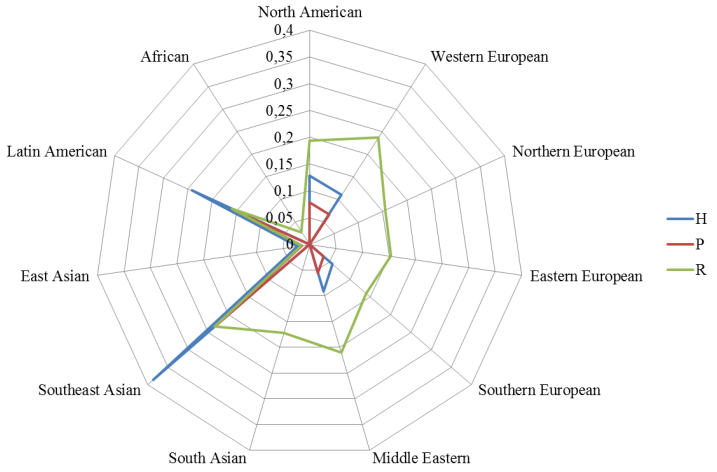
Шема 2. Втората шема се состои од лекови од категориите A, D, G, J, L и S. Овие лекови имаат значително повеќе негативни храна - лек интеракции со рецепти од Северна Америка, Западна Европа, Северна Европа и Источна Европа, споредено со останатите кујни. Оваа шема е видлива на Слика 4.28б.

Појавувањето на оваа шема на влијание е главно поради негативните интеракции кои лековите од оваа категорија ги имаат со млеко. Како што може да се види од Табела 4.34, млекото е број еден причинител за негативните храна - лек интеракции на



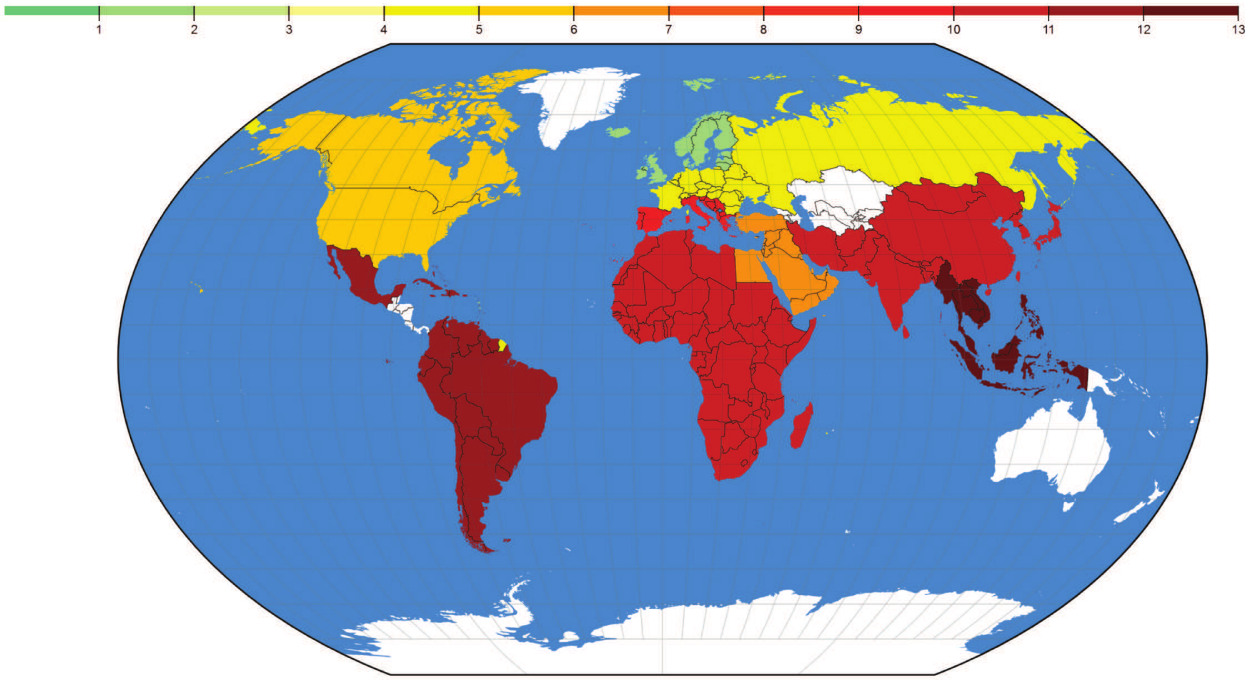
(а) Негативни интеракции на лековите од категорија B, C, N и V, во промили.

(б) Негативни интеракции на лековите од категорија A, D, G, J, L и S, во промили.

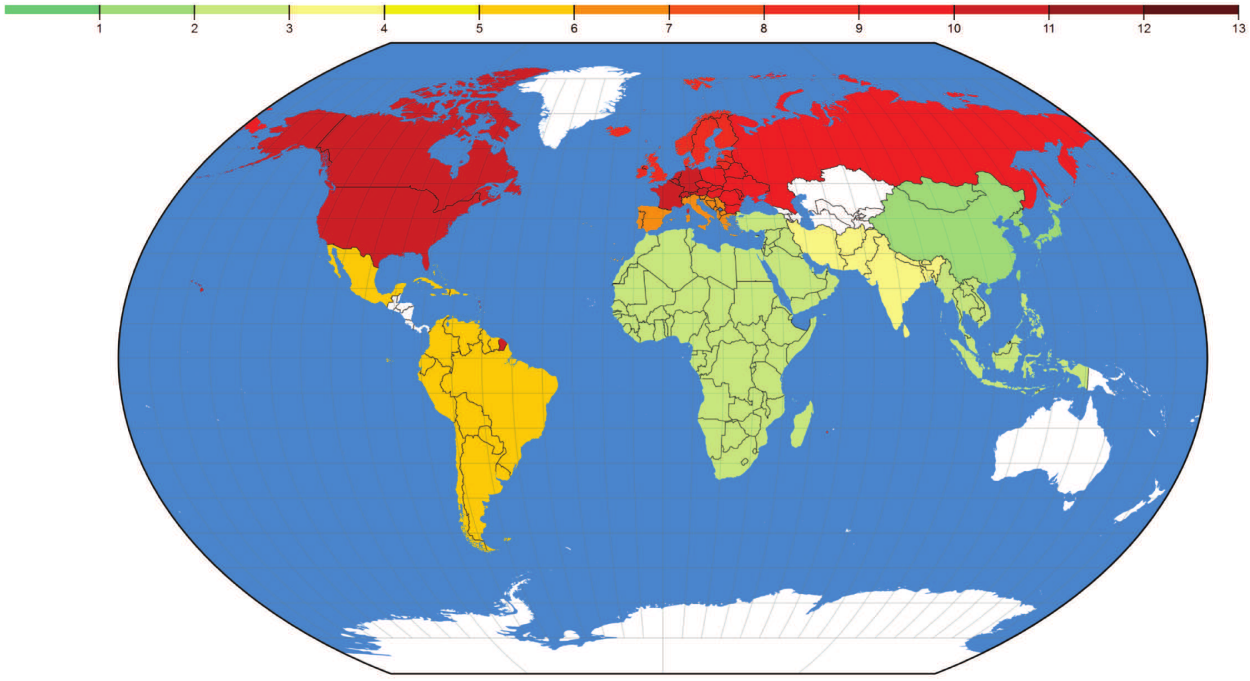


(в) Негативни интеракции на лековите од категорија H, P и R, во промили.

Слика 4.28: Трите шеми на негативни храна - лек интеракции помеѓу лекови од категории и рецепти од кујни, изразени во промили.



(а) Глобална дистрибуција на негативните интеракции на лекови од категоријата B.



(б) Глобална дистрибуција на негативните интеракции на лекови од категоријата J.

Слика 4.29: Број на пациенти (на 1.000) со можни негативни храна - лек интеракции додека се под терапија со лек од категорија B или категорија J, во различни кујни на глобално ниво. Мапите се генерирани со користење на d3.js библиотеката.

глобално ниво, а Табела 4.35 и Табела 4.37 покажуваат дека тоа е примарниот извор на негативни интеракции токму во овие делови од светот. Бидејќи овие кујни користат млеко во голем дел од нивните рецепти, за разлика од останатите кујни, таквата шема на влијание е очекувана.

На Слика 4.29б можеме да го видиме интензитетот на негативни храна - лек интеракции за лековите од категорија J. Бидејќи лековите од ова категорија припаѓаат на втората шема, оваа слика го прикажува истиот интензитет на негативни храна - лек интеракции како оние од Слика 4.28б, т.е. лековите од категорија J имаат значително повеќе негативни интеракции со рецепти од Северна Америка, Западна Европа, Северна Европа и Источна Европа.

Шема 3. Останатите категории на лекови, H, P и R, имаат значително помал интензитет на негативни храна - лек интеракции во споредба со останатите категории. Тие формираат различна шема, прикажана на Слика 4.28в, но поради малиот сооднос на интеракциите, оваа шема не е компактна како претходните. Лековите од овие категории имаат негативни ефекти при интеракција со кафе, чај и цитрон (Табела 4.38) и немаат негативни интеракции со останатите состојки од податочното множество. Како што може да се види од Табела 4.35 и Табела 4.37, кафето влегува во топ три состојки кои се одговорни за интеракциите во Северна Америка, Европа, Блиски Исток и во Латинска Америка, што кореспондира со шемата од Слика 4.28в. Големата употреба на цитрон и чај во рецептите од Југоисточна Азија (Табела 4.37) е одговорна за високото ниво на интеракции на лековите од овие категории со оваа кујна.

Анализа на состојки. Вториот аспект од анализата имаше за цел да ги идентификува главните состојки вклучени во негативните храна - лек интеракции. Во Табела 4.34 се прикажани процентите на негативни храна - лек интеракции за кои одредена состојка е одговорна. Процентот е пресметан од вкупниот број на негативни храна - лек интеракции идентификувани за време на анализата, кој изнесува 298.762 интеракции.

Од Табела 4.34 јасно се гледа дека две состојки се најчести во негативните интеракции: млекото, одговорно за над 56% од негативните интеракции и лукот со над 22%.

Овие две состојки имаат различни ефекти на храна - лек интеракциите во различни делови од светот и кај различни категории на лекови. Во Табелата 4.35 се прикажани по три состојки кои во рамките на дадена кујна се одговорни за најголемиот број негативни интеракции со лекови. Како што може да се види, млекото е состојка која предизвикува најголем дел од негативните интеракции во Северна Америка, Западна, Северна и Источна Европа. Од друга страна, најпроблематичната состојка кај рецептите од Јужна Европа, Блиски Исток, Азија, Латинска Америка и Африка е лукот.

На Слика 4.30 е прикажан овој тренд на мапа. Сликата 4.30а го илустрира појавувањето на млекото во вкупниот број на негативни храна - лек интеракции во рамките на одредена кујна, додека Сликата 4.30б го илустрира истото влијание на лукот.

Целосен преглед на влијанието кое дел од состојките го имаат во различни делови на светот е даден во Табела 4.37. Од тука може да се види дека состојките кои најчесто се одговорни за негативните храна - лек интеракции се млекото, лукот, кафето и ѓумбирот,

Табела 4.34: Процент на негативни храна - лек интеракции за кои е одговорна состојката, на глобално ниво

Состојка	**Учество во интеракции (%)**
млеко	56.110%
лук	22.617%
кафе	8.388%
ѓумбир	5.109%
сирење	2.197%
сланина	2.165%
црвено вино	1.865%
цитрон	1.684%
шунка	1.296%
вино	1.174%
чај	1.149%
авокадо	0.869%
пиво	0.304%
сладунец	0.120%

Табела 4.35: Топ 3 состојки кои се среќаваат во негативните храна - лек интеракции од одредена кујна

Кујна	**Топ 3 состојки**
Северна Америка	млеко, лук, кафе
Западна Европа	млеко, лук, кафе
Северна Европа	млеко, кафе, ѓумбир
Источна Европа	млеко, лук, кафе
Јужна Европа	лук, млеко, кафе
Блиски Исток	лук, млеко, кафе
Јужна Азија	лук, ѓумбир, млеко
Југоисточна Азија	лук, ѓумбир, млеко
Источна Азија	лук, ѓумбир, млеко
Латинска Америка	лук, млеко, авокадо
Африка	лук, ѓумбир, млеко

додека останатите состојки имаат значително послабо влијание. Сепак, од овие четири состојки, млекото и лукот отскокнуваат и имаат најзначајно влијание (Табела 4.34, Табела 4.35, Табела 4.37), поради што дискусијата е фокусирана на нив.

Влијанието на млекото. За млекото е познато дека има негативен ефект над антибиотици [98, 180], а антибиотиците припаѓаат на категориите A, C, D, G, J, L и S [167]. Млекото ја намалува биодостапноста, па дури и ја спречува апсорпцијата на некои од овие лекови. Ова кореспондира со нашите резултати од Табела 4.38.

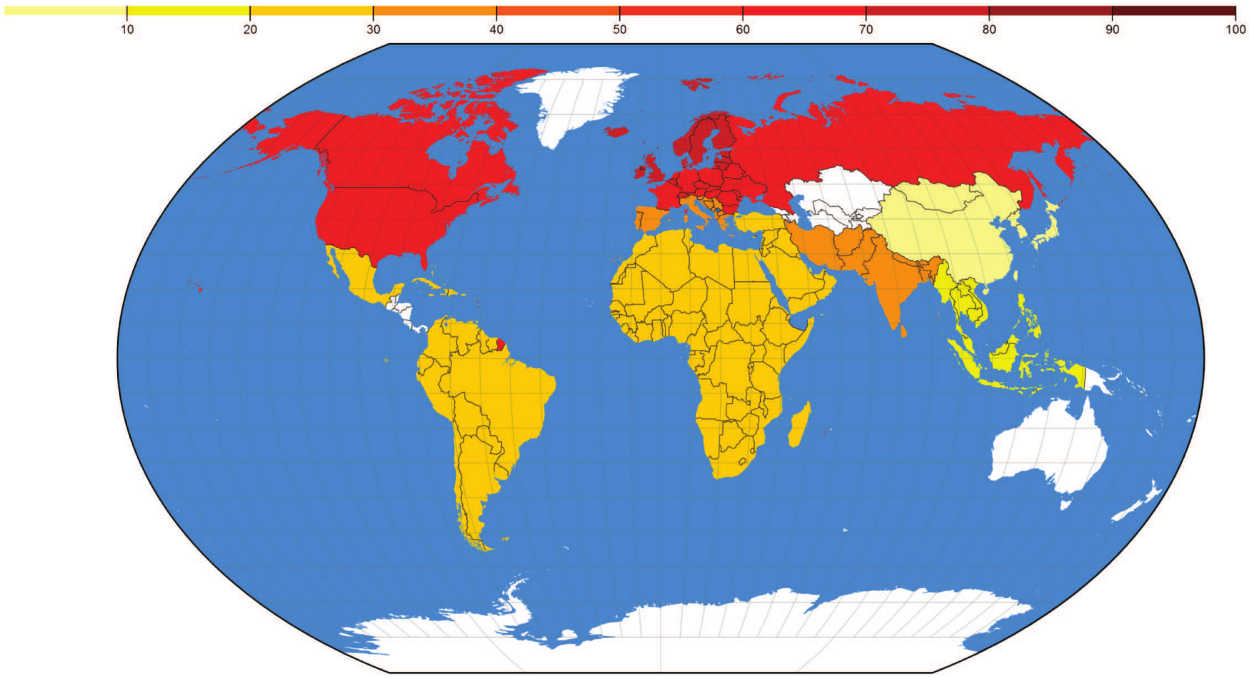
Причините за високата застапеност на млекото во негативните храна - лек интеракции во западната култура (Слика 4.30a, Табела 4.35, Табела 4.37) би можеле да се лоцираат во генерално широката употреба на млеко и млечни производи во овој дел од светот. Консумирањето на млеко во западната култура, особено во Северна Европа каде процентот на учество на млекото во негативните интеракции со лекови е најголем, најверојатно е директна последица на високата толеранција на лактоза која ја има популацијата од овие региони. Државите од Европа, особено од Северна Европа, имаат највисок процент на популација со толеранција на лактоза, на глобално ниво [188, 129, 197]. Од друга страна, регионите во Југоисточна Азија, Источна Азија и Јужна Африка се познати како региони со висок процент на популација која е нетолерантна на лактоза [129]. Ова најверојатно директно влијае на нивото на консумација на млеко во овие региони, што води до намалување на појавата на негативни ефекти од употребата на млеко со лекови, во овие делови од светот.

Влијанието на лукот. Причината поради која лукот е одговорен за над 22% од сите негативни храна - лек интеракции кои ги идентификувавме (Табела 4.34), е неговата негативна интеракција со антикоагуланти [189, 94], кои припаѓаат на категориите B, C и S. Од Табела 4.35 и Табела 4.37 јасно се гледа дека лукот е главно одговорен за негативните интеракции во Јужна Европа, Блиски Исток, Азија, Латинска Америка и Африка. Неговото влијание е евидентно и во остатокот од светот, но со многу помал интензитет. Оваа шема на употреба на лукот во кујните низ светот најверојатно има културна и историска позадина, имајќи предвид дека лукот се користел во Египет, Грција, Рим, Кина и Индија уште од антички времиња: се користел за превенција и третман на болести, за сила и зголемување на работната способност на физичките работници, па дури и како супстанца за зголемување на перформансите кај Олимписките натпреварувачи [172].

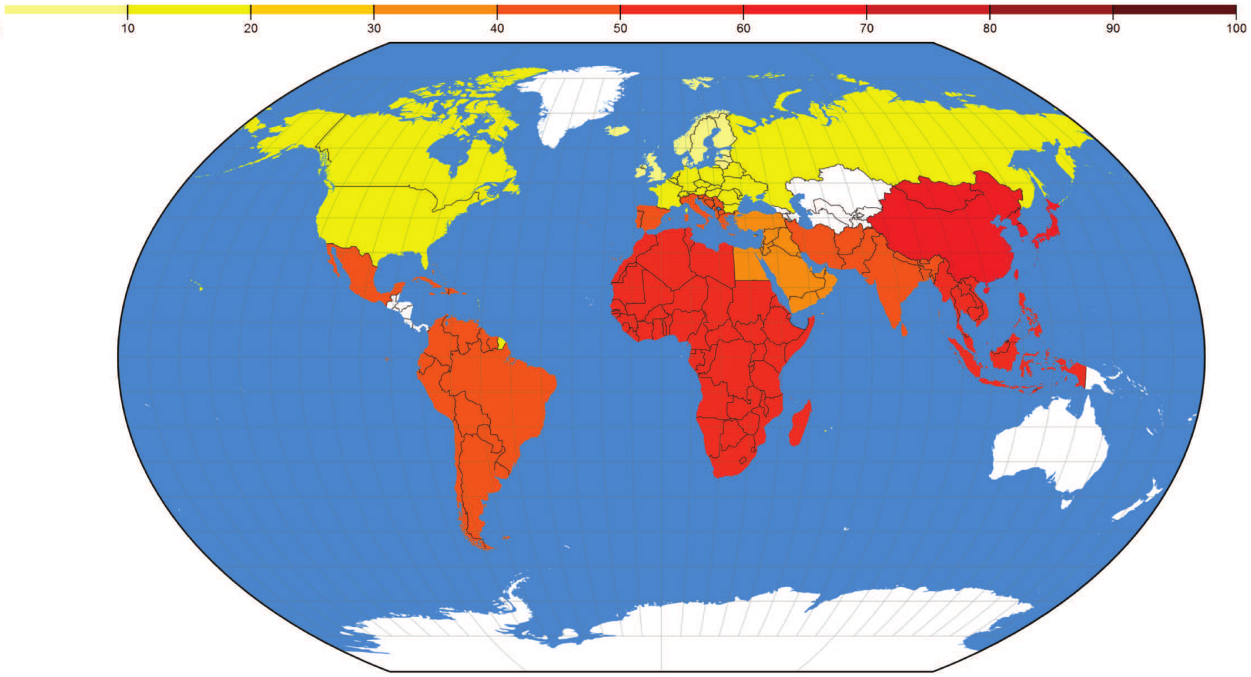
Мапи. Мапи слични на оние од Слика 4.29 и Слика 4.30 за останатите категории на лекови и состојки можат да се погледнат во рамките на нашата веб апликација за визуелизација на негативните кујна - лек интеракции [133]. Овие мапи ги користат податоците и резултатите презентирани во ова поглавје.

4.5.5 Дискусија

Во доменот на здравствени податоци реализиравме четири истражувања. Во првото истражување [140] генериравме поврзано податочно множество со лекови објавени од страна на Фондот за здравствено осигурување на Република Македонија. Податочното



(а) Глобалното влијание на млекото како предизвикувач на негативни интеракции.



(б) Глобалното влијание на лукот како предизвикувач на негативни интеракции.

Слика 4.30: Процент на појавувања на состојките млеко и лук во негативни храна - лек интеракции во различни кујни, на глобално ниво. Мапите се генерирани со користење на d3.js библиотеката.

Табела 4.36: Промили на постоечки интеракции помеѓу лекови од АТС категории и рецепти од кујни

	С.Ам.	З.Ев.	С.Ев.	И.Ев.	Ј.Ев.	Б.Ис.	Ј.Аз.	ЈИ.Аз.	И.Аз.	Л.Ам.	Аф.
A	5.288‰	5.263‰	4.462‰	4.930‰	2.547‰	1.531‰	2.085‰	1.080‰	0.408‰	2.356‰	1.238‰
B	5.271‰	4.345‰	1.643‰	4.687‰	9.663‰	6.894‰	10.640‰	12.699‰	10.756‰	11.013‰	10.045‰
C	5.821‰	5.207‰	3.640‰	5.083‰	6.790‰	4.815‰	7.123‰	8.371‰	6.645‰	8.003‰	6.554‰
D	7.081‰	7.011‰	5.983‰	6.573‰	3.401‰	1.928‰	2.717‰	1.351‰	0.543‰	3.130‰	1.655‰
G	7.240‰	7.504‰	6.326‰	6.836‰	3.789‰	2.848‰	3.520‰	1.781‰	0.528‰	3.237‰	1.652‰
H	0.129‰	0.111‰	0.000‰	0.000‰	0.056‰	0.091‰	0.000‰	0.386‰	0.023‰	0.242‰	0.000‰
J	10.242‰	10.532‰	8.468‰	9.793‰	6.276‰	2.965‰	3.682‰	2.035‰	1.234‰	5.641‰	2.642‰
L	4.430‰	4.673‰	3.818‰	4.156‰	2.744‰	1.644‰	1.720‰	1.276‰	0.844‰	2.137‰	1.442‰
M	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰	0.000‰
N	4.829‰	5.510‰	3.397‰	4.706‰	5.935‰	4.157‰	4.588‰	4.923‰	3.555‰	6.032‰	3.514‰
P	0.078‰	0.067‰	0.000‰	0.000‰	0.034‰	0.055‰	0.000‰	0.234‰	0.014‰	0.147‰	0.000‰
R	0.193‰	0.237‰	0.155‰	0.153‰	0.139‰	0.211‰	0.172‰	0.234‰	0.015‰	0.160‰	0.028‰
S	5.117‰	5.074‰	4.302‰	4.779‰	2.437‰	1.433‰	1.975‰	1.012‰	0.398‰	2.270‰	1.206‰
V	9.022‰	8.023‰	5.123‰	8.150‰	10.912‰	7.453‰	11.498‰	12.822‰	10.888‰	12.582‰	10.815‰

Табела 4.37: Процент на учество на состојките во негативни храна - лек интеракции, по кујна

	С.Ам.	З.Ев.	С.Ев.	И.Ев.	Ј.Ев.	Б.Ис.	Ј.Аз.	ЈИ.Аз.	И.Аз.	Л.Ам.	Аф.
млеко	62.288%	60.132%	70.955%	62.030%	32.665%	26.331%	30.302%	13.634%	8.637%	29.196%	23.833%
лук	17.606%	13.616%	3.119%	17.868%	41.754%	39.943%	45.090%	59.570%	67.848%	47.122%	53.333%
кафе	8.662%	11.213%	10.234%	8.883%	8.711%	14.837%	4.012%	3.671%	0.456%	5.648%	2.917%
ѓумбир	3.929%	2.670%	4.678%	1.218%	0.689%	6.971%	40.352%	32.092%	42.626%	1.399%	28.000%
сирење	1.631%	2.203%	0.780%	2.132%	6.241%	0.754%	0.382%	0.315%	0.182%	7.302%	0.500%
сланина	2.244%	3.344%	0.877%	4.416%	1.795%	0.000%	0.115%	0.000%	0.899%	1.815%	0.250%
цитрон	1.705%	1.435%	0.000%	0.000%	0.823%	2.025%	0.000%	6.765%	0.560%	3.470%	0.000%
цр. вино	1.434%	3.337%	3.119%	2.843%	5.303%	5.652%	0.611%	0.210%	1.303%	1.829%	6.000%
шунка	1.234%	1.602%	1.462%	0.457%	2.757%	0.000%	0.000%	0.157%	1.133%	0.746%	0.000%
вино	0.847%	1.575%	1.170%	0.203%	2.431%	1.507%	0.611%	4.195%	7.817%	0.403%	1.333%
чај	1.141%	1.589%	3.314%	1.726%	0.000%	4.805%	10.394%	5.349%	0.443%	0.000%	0.000%
авокадо	0.554%	0.133%	0.780%	0.000%	0.211%	0.565%	0.000%	0.210%	1.042%	7.746%	0.333%
пиво	0.299%	0.601%	0.292%	0.152%	0.101%	0.141%	0.000%	0.000%	0.156%	0.605%	0.000%
сладунец	0.141%	0.000%	1.754%	0.000%	0.000%	0.000%	0.000%	0.000%	0.234%	0.000%	0.000%

Табела 4.38: Процент на лекови, по ATC категорија, кои имаат негативни интеракции со состојката

	A	B	C	D	G	H	J	L	M	N	P	R	S	V
млеко	1.923%	0.000%	1.058%	2.609%	2.326%	0.000%	3.509%	1.515%	0.000%	0.431%	0.000%	0.000%	1.887%	1.754%
лук	0.000%	1.786%	1.058%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.431%	0.000%	0.000%	0.000%	1.754%
кафе	1.282%	1.786%	1.058%	0.870%	6.977%	0.000%	1.170%	0.758%	0.000%	8.621%	0.000%	0.971%	0.943%	0.000%
ѓумбир	0.000%	1.786%	1.058%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.431%	0.000%	0.000%	0.000%	1.754%
сирење	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.000%	0.000%	0.431%	0.000%	0.000%	0.000%	0.000%
сланина	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.000%	0.000%	0.862%	0.000%	0.000%	0.000%	0.000%
цр. вино	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.758%	0.000%	0.862%	0.000%	0.000%	0.000%	0.000%
цитрон	1.923%	0.000%	9.524%	1.739%	0.000%	5.882%	1.754%	3.030%	0.000%	5.172%	3.571%	1.942%	1.887%	0.000%
шунка	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.000%	0.000%	0.862%	0.000%	0.000%	0.000%	0.000%
вино[a]	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.758%	0.000%	0.862%	0.000%	0.000%	0.000%	0.000%
чај	1.282%	1.786%	1.058%	0.870%	6.977%	0.000%	0.585%	0.758%	0.000%	8.621%	0.000%	0.971%	0.943%	0.000%
авокадо	0.000%	0.000%	1.058%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.862%	0.000%	0.000%	0.000%	1.754%
пиво	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.585%	0.000%	0.000%	0.862%	0.000%	0.000%	0.000%	0.000%
сладунец	0.000%	0.000%	9.524%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%	0.000%

[a]Постои значителен број рецепти кои содржат ‘вино’ како состојка. Во негативните интеракции на лековите се среќава само ‘црвено вино’, што значи дека ‘бело вино’ се смета за безбедно. Покрај двосмисленоста кога во даден рецепт состојката е наведена единствено како ‘вино’, одлучивме да го користиме тоа во анализата, бидејќи таа состојка сепак може да предизвика негативна интеракција со лек во случај кога би се искористило ‘црвено вино’ во подготовката на рецептот. Како што може да се види од табелата, состојката ‘вино’ има негативни интеракции со оние лекови со кои ‘црвено вино’ е во негативна интеракција.

множество на крајот броеше 1.020 уникатни лекови, за кои имавме податоци за името, јачината, пакувањето, АТС кодот, цената, производителот. Дополнително, секој од лековите од податочното множество беше поврзан со низа лекови од истото податочно множество со кои ја дели истата функција, но и со генерички лек(ови) од DrugBank податочното множество. Двете поврзувања се базираа на АТС кодовите на лековите и генеричките лекови. Овие две последни информации се всушност клучни за овозможувањето на дополнителните, нови, претходно недостапни кориснички сценарија во доменот, кои користат податоци кои не се објавени од страна на Фондот, туку од други институции и се наоѓаат на други Веб локации. Преку линковите кон DrugBank податочното множество, корисниците можат да започнат со пребарување низ податоците од генерираното поврзано податочно множество, за потоа да се префрлат кон други податочни множества од LOD облакот и да пристапат до податоци со поширок контекст.

Резултантното поврзано податочно множество содржеше над 21.000 RDF тројки за 1.020 лекови од Фондот, 9.946 `hifh:similarTo` линкови помеѓу нив и 1.015 `rdfs:seeAlso` линкови кон генерички лекови од DrugBank податочното множество. Податочното множество е достапно преку активна Virtuoso инстанца, како и преку Datahub податочниот портал.

Во рамките на второто истражување [139] го проширивме поврзаното податочно множество од првото истражување [140], со тоа што во него го вклучивме податоците од Фондот кои се однесуваат на медицинските установи и нивните 24-часовни дежурства, како и податоците за расположливите лекови по аптеки на месечно ниво, добиени од Здружението на аптеки на Македонија (ЗАМ). Дополнителните податоци овозможија употреба на нови кориснички сценарија, преку кои корисникот може да добие информации за, на пример, најблиската аптека која е отворена 24 часа во тековниот ден и која го има на располагање лекот од интерес. Ваквото сценарио беше овозможено благодарение на поврзувањата направени помеѓу трите нови и податочното множество од претходното истражување, што е особено важно во доменот во кој одвоените податочни складишта предизвикуваат сериозна пречка при добивањето на релевантни информации и податоци [100]. Резултантното поврзано податочно множество е достапно преку активна Virtuoso инстанца.

Иако изворните податоци од ЗАМ не се јавно достапни, туку беа добиени на барање од нашиот истражувачки тим, целта на истражувањето беше демонстрирање на предностите од постоење на такво контекстно поврзување на податоците од доменот, со надеж дека одговорните ќе се потрудат да ги направат ваквите податоци достапни за поширокаа јавност. Тоа би овозможило проширување на податочното множество на редовна база, со што неговата употребливост како податочен слој за различни апликации и сервиси значително би се зголемила [150].

Во третото истражување [138] ги применивме искуствата од првото истражување [140] и генериравме поврзано податочно множество со лекови од Бирото за лекови на Република Македонија, кое обезбедува слободен Веб пристап до голем број детални информации за лековите регистрирани за продажба во Македонија. Ова истражување

стана возможно откако Бирото ги објави овие податоци на Веб, кои беа недостапни во периодот кога го реализиравме првото истражување од доменот. Лековите оф генерираното податочно множество го поврзавме со генерички лекови од DrugBank податочното множество. Со тоа, овозможивме кориснички сценарија со кои фармацевт, лекар или самиот пациент можат да добијат детални информации за можните лек - лек и храна - лек интеракции на лекот од интерес, за можните несакани дејствија од лекот, идентификување на слични лекови на лекот од интерес кои можеби се подостапни или поевтини во одредени случаи, итн. Сите овие податоци и информации се достапни на Вебот, но бараат мануелен пристап и анализа. Користењето на концептот на поврзани податоци овозможува контекстно поврзување на ентитетите од овие независни податочни множества, додека технологиите на Семантичкиот Веб овозможуваат непречен пристап до нив преку постоечката инфраструктура на Вебот што води кон искористување на нивниот потенцијал [87].

Поврзаното податочно множество резултираше со 134.714 RDF тројки кои опишуваат 3.407 лекови од Бирото. Овие лекови имаат 33.872 `dbm:similarTo` линкови помеѓу себе и 2.791 `rdfs:seeAlso` линкови кон соодветни генерички лекови од DrugBank. Податочното множество е достапно преку нашата јавна Virtuoso инстанца.

Над ова податочно множество ја развивме мобилната апликација "Мобилен Фармацевт", која директно ги користи податоците од поврзаното податочно множество преку SPARQL endpoint-от на Virtuoso инстанцата. Преку мобилната апликација ги демонстриравме главните придонеси за крајните корисници - фармацевти, лекари и пациенти - од пристапот до консолидирани податоци за лекови од Бирото и нивни генерики објавени од други институции, на други локации на Вебот. Ваквите напредни и нови кориснички сценарија го материјализираат потенцијалот на отворените и поврзаните податоци за кои зборува авторот на [150].

Последното наше истражување во доменот на здравствените податоци беше фокусирано на аналитичките сценарија овозможени од контекстно поврзување на хетерогени податочни множества [137]: во него ги анализиравме интеракциите кујна - категорија на лек, преку што добивме глобален преглед на различното влијание помеѓу категориите лекови и рецептите од различни кујни. Во истражувањето увидовме две значајни шеми на негативните интеракции: лековите од категориите B, C, N и V имаат негативни интеракции со храна од Јужна Европа, Азија, Латинска Америка и Африка, додека лековите од категориите A, D, G, J, L и S имаат негативни интеракции со храна од Северна Америка и Европа (Западна, Северна и Источна Европа). Овие шеми потекнуваат од различните состојки кои се користат во различните кујни во светот, при што лукот и ѓумбирот се најмногу одговорни за првата шема, додека млекото е одговорно за втората. Влијанието на млекото и лукот варира во различните делови од светот, главно од културни, историски и биолошки причини за нивното присуство (или отсуство) во рецептите од дадена кујна.

Целта на ова истражување беше потенцирање на важноста за дополнително професионално советување на пациенти кои се под терапија со лекови кои имаат познати

негативни интеракции со храна. Пациент кој е под терапија со таков лек би требало да биде советуван од страна на фармацевтот или лекарот за храната, состојките и, генерално, кујните кои треба да ги избегнува или целосно исклучи од својата исхрана. Нашата анализа на глобалната дистрибуција на негативните интеракции помеѓу различните лекови и кујните може да даде генерален преглед и генерални насоки за опасностите од правење погрешни кујна - лек комбинации.

4.6 Резултати

Во овој дел од дисертацијата направивме преглед на истражувачките проекти кои ги реализиравме во рамките на истражувањата поврзани до поврзани податоци. Поточно, истражувачките проекти се однесуваа на примената на принципите на поврзани податоци во неколку домени: податоци за криминал, јавен транспорт и аерозагадување, финансискиот домен, доменот на мултимедија и музика, како и доменот на здравство и лекови. Во оваа Глава дадовме детален опис на нашата работа на 11 различни проекти. Како дел од овие проекти, дизајниравме и објавивме седум онтологии; трансформиравме, консолидиравме, поврзавме и објавивме 12 поврзани податочни множества; развивме голем број кориснички ориентирани и аналитички сценарија над развиените податочни множества; развивме и објавивме 6 веб и мобилни апликации кои како податочен слој ги користат единствено нашите поврзани податочни множества.

Во рамките на истражувањата се придржувавме до идејата сите дефинирани онтологии и генерирани податочни множества да ги објавуваме на единечно место на Вебот, како точка преку која ќе бидат постојано достапни. Како што елаборираме за секој од проектите поединечно, нашите онтологии и поврзани податочни множества се дел од наша јавна Virtuoso инстанца [66], односно се достапни преку нејзиниот јавно достапен SPARQL endpoint-от [57]. Како и секој останат Virtuoso SPARQL endpoint, тој може да се користи како REST-базиран сервис, на следниот начин:

```
http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT
```

Притоа, `SPARQLQUERY` го претставува SPARQL прашањето во URL-кодиран формат, додека `FORMAT` се однесува на посакуваниот формат на резултатите од прашањето и може да биде HTML, XML, JSON, Javascript, CSV, Spreadsheet, RDF/XML, N3, Turtle, итн. Со ова, податоците од нашите поврзани податочни множества и онтологии можат да се пристапуваат директно од мобилни, веб и десктоп апликации. Сѐ што е потребно од технички аспект е користење на стандардни HTTP повици преку постоечката инфраструктура на Вебот.

4.7 Заклучок

Работата презентирана во оваа Глава имаше за цел да нѝ овозможи практична анализа на методите и техниките кои можат да се искористат во секој од чекорите на

животниот тек за поврзано податочно множество во даден домен. Целта беше да добиеме искуство за различните начини на кои можат да се соберат изворните податоци, за различните пристапи во моделирање на податоците, за најдобрите практики за искористување и дефинирање на онтологии и вокабулари, за различните методи за трансформација на податоци кои можат да варираат од мануелни до целосно автоматизирани, за различните начини на објавување на податочното множество на Вебот, како и за големиот број начини на кои податочните множества можат да се искористат во кориснички апликации и сервиси. Преку имплементацијата на принципите на поврзани податоци во неколку различни домени, се обидовме да ги идентификуваме сите специфичности и најдобри практики за секој од чекорите на животниот циклус на едно поврзано податочно множество.

Глава 5

Методологија за поврзани податоци со фокус на повторно искористување

5.1 Мотивација

Постоечките методологии за поврзани податоци се фокусираат на чекорите потребни за лоцирање, генерирање / трансформирање и објавување на поврзани податоци. Притоа, пропуштаат да го опфатат делот за повторно искористување на животниот тек на поврзаното податочно множество, т.е. делот кој се однесува на искористување на веќе креираните податочни шеми, алатки, сервиси, итн., од страна на други објавувачи на податоци во доменот. Ниту една од постоечките методологии не нуди совети, механизми или методи кои би ги мотивирале објавувачите на податоци, или би им помогнале, да ги споделат механизмите развиени во текот на генерирањето и објавувањето на поврзаното податочно множество од даден домен. Поради тоа, знаењето содржано во процесот на креирање на податочното множество останува одвоено од заедницата која работи со поврзани податоци и најчесто е сместено во научни трудови или извештаи. Со тоа, секогаш кога нови објавувачи на податоци ќе одлучат да генерираат поврзани податочни множества во истиот домен, принудени се да ги имплементираат чекорите од животниот тек на поврзани податоци од почеток.

Во домени во кои повеќе објавувачи на податоци би биле заинтересирани за генерирање на поврзани податочни множества кои покриваат исто или слично множество ентитети, од голема корист би била методологија која обезбедува насоки и совети за креирање и употреба на компоненти кои овозможуваат повторно искористување, а кои ги претставуваат чекорите од животниот тек на едно поврзано податочно множество. На тој начин, идните објавувачи на податоци би биле во можност да го употребат не само знаењето, туку и конкретните алатки, шеми и сервиси развиени од иницијалните објавувачи. Повторното искористување на шемите недвосмислено резултира со споредливи и порамнети податочни множества, кои можат да се поврзат меѓусебно на едноставен начин. Бенефитот од една таква методологија е спуштањето на бариерата за објавувачите на податоци да генерираат и објавуваат поврзани податочни множества, што

води кон поголемо количество поврзани податоци од дадениот доменот, поврзани со претходно развиените податочни множества од доменот и од LOD облакот.

Мотивирани од ваквата состојба, врз база на нашето широко искуство во доменот на поврзани податоци, развивме нова методологија за поврзани податоци која се фокусира на принципот на повторно искористување на чекорите од животниот тек на поврзани податоци, за даден домен. Методологијата содржи чекори за моделирање и порамнување на податоците, трансформација во 5-star поврзани податоци, објавување на креираните податочни множества на Вебот и дефинирање на кориснички сценарија или развој на апликации и сервиси над податочните множества. Методолошките насоки имаат за цел да им асистираат на сопствениците и објавувачите на податоци од даден домен во објавување на нивните податоци во ист, порамнет формат, кој подлежи на принципите на поврзани податоци. Нивните податоци, трансформирани во поврзани податоци и поврзани со друго податоци објавени со истите компоненти од животниот тек, можат потоа да се користат во нови кориснички и аналитички апликации и сервиси.

Како валидација на предложените методолошки насоки, ги применивме во доменот на здравство и лекови. Методологијата ја применивме во рамките на автоматизиран систем кој собира податоци за лекови од официјалните национални регистри на дваесет и три различни држави, ги прочистува податоците, ги порамнува и трансформира во 5-star поврзани податоци и ги објавува на Вебот во единствено, порамнето и консолидирано податочно множество со податоци за лекови. Врз база на методологијата, развивме компоненти за животниот тек на поврзаните податоци, компоненти кои поддржуваат повторно искористување: заедничка RDF шема, податочен предефиниран формат, податочен трансформатор, SPARQL-базирана алатка за проширување и меѓусебно поврзување на податочното множество и веб-базирана алатка за трансформација, меѓусебно поврзување и објавување на податочното множество. Потоа демонстрираме низа кориснички и аналитички сценарија над генерираното податочно множество, кои се недостапни во ситуација кога корисникот работи со податоците достапни на Веб во HTML веб страни.

5.2 Дефинирање на методологијата

Врз база на искуството од имплементација на принципите на поврзани податоци во доменот на здравство, сообраќај, криминал и финансии, презентирани во Глава 4 и постоечките методологии за работа со поврзани податоци презентирани во Глава 3, развивме методологија за поврзани податоци фокусирана на повторна употребливост на нејзините чекори. Методологијата содржи насоки за запознавање со доменот од интерес, моделирање и порамнување на податоците, трансформација на податоците во висококвалитетни поврзани податоци, објавување на креираното податочно множество на Веб и дефинирање на кориснички сценарија или развој на апликации и сервиси на податочното множество. Овие методолошки насоки се изградени над постоечките методологии и се состојат од чекори чија цел е да им послужат на објавувачите на податоци

како водич низ процесот на генерирање висококвалитетни поврзани податоци, со цел контекстуално поврзување и консолидирање со други податочни множества.

Врз база на нашето досегашно искуство во областа и анализата на постоечките методологии, нашите методолошки насоки можеме да ги групираме во пет генерални чекори (Табела 5.1, Слика 5.1):

I. Запознавање со податоците од доменот и од Вебот
II. Моделирање на податоците
III. Трансформација во 5-star поврзани податоци
IV. Објавување на податочното множество на Веб
V. Кориснички сценарија и апликации

5.2.1 Чекор 1: Запознавање со податоците од доменот

Првиот чекор соодветствува со првите чекори од постоечките методологии: важно е објавувачот на податоци да биде добро запознаен со доменот од интерес и податоците во него. Ваквото разбирање на шемата на изворните податоци и нивната семантика е суштинско за следните чекори кои вклучуваат моделирање, порамнување на шеми и трансформација на податоци.

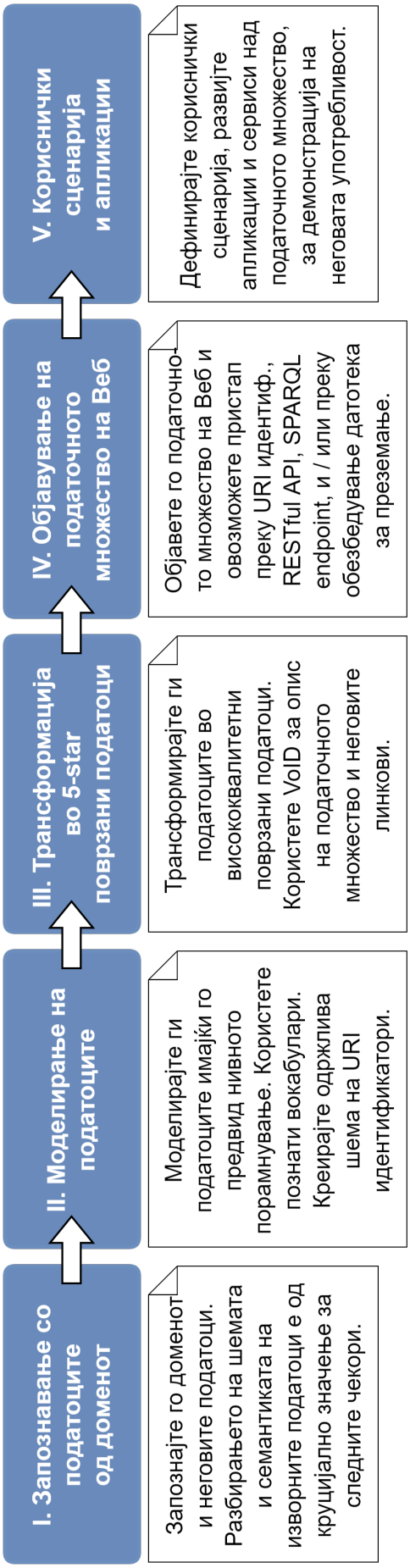
Доколку ова е прв пат во кој објавувачот се среќава со концептот на поврзани податоци, нашиот совет е тој најпрвин да се запознае со нивоата на квалитет на податоци од Тим Бернерс-Ли [84], четирите принципи на поврзани податоци [85] и LOD облакот [30]. Потоа, значајно е тој да се запознае со спецификите на доменот за кој станува збор и со значењето на податочното множество селектирано за трансформација. За оваа цел, препорачуваме и консултации со експерти од доменот. Дополнително, тој треба да се запознае и со постоечките поврзани податочни множества кои се од истиот или од сродни домени. За оваа цел, може да се искористи Datahub порталот [12] и кешираната верзија од LOD облакот [31]. Овие активности ќе му помогнат на објавувачот на податоци да добие подобар увид во типовите на податоци тековно достапни како поврзани податоци, нивната шема, нивните сличности и разлики, како и нивните постоечки и потенцијални поврзувања со други податочни множества. Покрај тоа, ваквите активности ќе му овозможат да ги одреди онтологиите и вокабуларите кои најчесто се користат во доменот, што е особено значајно за следниот чекор.

5.2.2 Чекор 2: Моделирање на податоците

Во следниот чекор, објавувачот на податоци треба да се фокусира на моделирањето на податоците и нивно *порамнување* (alignment, анг.) со други постоечки или идни податочни множества. Во овој контекст, порамнувањето на податоците се однесува на користење иста или еквивалентна шема за опис на податоците, која би овозможила поедноставено и недвосмислено комбинирање на податоците од различните податочни множества. Во овој чекор, објавувачот на податоци треба да направи избор на шемата

Табела 5.1: Компаративна анализа на постоечките методологии за поврзани податоци и нашите методолошки насоки

	Наша методологија	Hyland et al.	Hausenblas et al.	Villazón-Terrazas et al.	LOD2
Модуларност за повторна употреба	**I. Запознавање со податоците**	1. Идентификација	1. Познавање на податоците	1. Спецификација	1. Екстракција 2. Складирање
	II. Моделирање на податоците	2. Моделирање 3. Именување 4. Постоечки вокабулари	2. Моделирање	2. Моделирање	
	III. Трансформација во поврзани податоци	5. Опис 6. Трансформација	3. Објавување 4. Лоцирање 5. Интеграција	3. Генерирање	3. Креирање 4. Поврзување 5. Класификација 6. Квалитет 7. Еволуција и корекција
	IV. Објавување на податочното множество	7. Објавување и известување		4. Објавување	
	V. Кориснички сценарија и апликации		6. Кориснички сценарија	5. Искористување	8. Пребарување и истражување



Слика 5.1: Чекорите од нашата методологија

за податоците од податочното множество, со цел да се постигне правилна анотација, т.е. да се искористат податочните полиња кои се потребни за финалните кориснички сценарија, да се анотираат со недвосмислена и прецизна семантика според контекстот и да се направи правилниот избор за шемата кој ќе дозволи комбинирање со други податочни множества. Дополнително, тој треба да ја дефинира шемата за креирање на URI идентификаторите за ентитетите од податочното множество, а по можност и за класите и својствата од онтологиите.

Шема на податоци. Шемата на податоците се дефинира преку изборот на вокабулари или онтологии. Принципите за креирање и користење на онтологии се развиени токму за оваа цел: да ги максимизираат шансите за повторно искористување, со што се овозможува полесно порамнување на податочните множества [78]. Ова значи дека објавувачот на податоци треба секогаш да се обиде да искористи веќе постоечки вокабулари и онтологии, давајќи им притоа предност на оние кои се најраспространети и најмногу користени. Постојат неколку алатки за пребарување и наоѓање онтологии и вокабулари, кои објавувачот треба да ги искористи во овој чекор. Две од нив се Linked Data Vocabularies (LOV) [33] и DERI Vocabularies [16], кои обезбедуваат статистика за користење на онтологиите за кои имаат информации, што овозможува проценка на значењето на дадена онтологија или вокабулар во одреден домен.

Сепак, податочните множества многу често имаат специфични податоци кои не се покриени од страна на постоечките онтологии и вокабулари. Во тој случај, постоечките онтологии или вокабулари треба да се прошират, или да се креира нова онтологија или вокабулар од почеток. При дефинирањето на нова онтологија или вокабулар, важно е да се дефинираат и мапирањата помеѓу новите класи и својства со класите и својствата на други широко користени онтологии, со цел да се обезбеди основа за порамнување на податоците и RDF базирано расудување.

Уште еден важен аспект во овој чекор е користењето на општи онтологии, односно онтологии од високо ниво (upper-level ontologies, анг.); тие обезбедуваат податочна шема за голем број различни и специфични домени, поради нивната генералност. Кога две или повеќе податочни множества се анотирани со истата општа онтологија или општ вокабулар, се овозможува нивното поврзување и расудување, односно се олеснува порамнувањето кое е неопходно за консолидирање на податоците.

Формат на URI идентификаторите. При дефинирањето на шемата за URI идентификаторите кои ќе се користат за идентификација на ентитетите од податочното множество, важно е да се одредат типовите на ентитети кои постојат во податочното множество. Согласно принципите на поврзани податоци, секој ентитет во податочното множество, како и секоја класа и својство од онтологијата, треба да има уникатен идентификатор во форма на HTTP URI. Со цел обезбедување подобри перформанси во иднина, нашето искуство сугерира користење посебни URI патеки за различните типови ентитети, на пример: `http://example.com/drug/`, `http://example.com/interaction/`, `http://example.com/disease/`, итн. Дополнителна препорака е користењето на URI идентификатори базирани на знакот '/', наместо URI идентификатори базирани на '#'.

Ова може да резултира во употреба на дополнително HTTP барање од страна на машината која пристапува до ваков тип на URI идентификатор, но овој недостаток станува помалку важен кога ќе се спореди со бенефитот - подобри перформанси при пристапот до големи податочни множества [69].

5.2.3 Чекор 3: Трансформација во 5-star поврзани податоци

Во рамките на третиот чекор, изворното податочно множество треба да се трансформира во RDF формат и да се креираат линкови до други податочни множества, со што ќе се креира поврзано податочно множество од највисок, 5-star квалитет. По трансформацијата, треба да креираат дополнителни метаподатоци за податочното множество.

Трансформација на податоците. Процесот на трансформација може да се изврши на повеќе различни начини - објаснети во детали во Поглавје 2.3 - и со користење на различни софтверски алатки, како на пример OpenRefine [48], LODRefine [37], D2R Server [10], Virtuoso [68], Silk Framework [55], итн. За да донесат вистинските одлуки во овој чекор, важно е да се одредат неколку карактеристики на податочното множество. Природата на податочното множество ќе одреди (а) дали трансформацијата е еднократна операција, операција која треба да се извршува повторно на одредени временски интервали (на пример, еднаш месечно), или е континуирана операција; (б) дали старите верзии на трансформираното податочно множество е потребно да се чуваат, дали при следните трансформации се трансформираат само новите и променетите делови од податочното множество (правење на т.н. 'делта' ажурирања), или можеби старите верзии на податочните множества повеќе не се потребни во конкретните кориснички сценарија во кои се користат; (в) дали е потребна мануелно или автоматизирано прочистување на податоците пред првата или следните трансформации; (г) дали изворното множество е секогаш достапно на истата локација и пристапно преку истите интерфејси. Овие карактеристики на податочното множество треба да му овозможат на објавувачот на податоци да утврди дали процесот на трансформација може да биде целосно или делумно автоматизиран и да ги идентификува деловите од работниот тек на трансформацијата во кои е потребно човечко внимание и интеракција.

Метаподатоци. Додавањето метаподатоци за новокреираното поврзано податочно множество е од големо значење од аспект на негово повторно искористување - користењето вокабулари како VoID [77] помага во недвосмислено одредување на карактеристиките на податочното множество и линковите кои податочното множество ги има кон други поврзани податочни множества, преку софтверски агенти. VoID метаподатоците содржат информации за името, описот и категоријата на податочното множество, информации за тековната верзија на податоците во него и фреквенцијата на ажурирање, контакт информации, лиценца под која податоците се достапни, линкови кон SPARQL endpoint-ите, користените вокабулари со нивните својства и класи. Вокабуларот експлицитно ги опишува и линковите помеѓу податочното множество и други поврзани податочни множества, дефинирани во самото податочно множество. Користењето на VoID вокабуларот е експлицитно наведено во соодветните чекори во методологиите не

Hyland et al., Hausenblas et al. и Villazón-Terrazas et al.

5.2.4 Чекор 4: Објавување на податочното множество на Веб

Во четвртиот чекор, генерираното поврзано податочно множество, заедно со неговите VoID метаподатоци, треба да се објават на Вебот. Објавувањето треба да се направи согласно препораките на W3C за објавување на поврзани податоци [126], кои посочуваат кон овозможување директна достапност преку HTTP URI идентификатори до ентитетите, обезбедување RESTful API пристап, обезбедување пристап преку SPARQL endpoint и / или обезбедување опција за директно преземање на податочното множество како датотека.

Постои широка палета на алатки и софтверски платформи кои обезбедуваат едноставно објавување на поврзани податочни множества. Помеѓу нив се D2R Server [10] и Virtuoso [68], кои овозможуваат објавување на податочни множества од поврзани податоци кои изворно се наоѓаат во RDF датотеки (Turtle, N3, RDF/XML, JSON-LD, итн.), во CSV датотеки, или во релациони бази на податоци. Овие платформи потоа обезбедуваат пристап по поврзаните податоци од податочното множество преку HTML страни, преку директно преземање на RDF датотеки, како и преку SPARQL endpoint-и кои можат да се користат и како RESTful API сервиси.

Значаен дел од овој чекор е и соопштувањето пред заинтересираната јавност и публика дека е креирано ново податочно множество со поврзани податоци во конкретниот домен. За таа цел, потребно е податочното множество, заедно со неговите VoID метаподатоци и генералниот опис да се објави на некој од популарните податочни портали, како што е Datahub [12]. Овој податочен портал може да послужи како почетен чекор за вклучување на податочното множество во самиот график на LOD облакот [32]. Овие две акции би требало да овозможат поголема видливост на податочното множество. Соопштувањето треба да се изведе и преку постоечките комуникациски канали на објавувачот и неговата организација. Со цел да се осигура понатамошното користење и повторно искористување на податочното множество, важно е да се обезбеди и форма за комуникација или контакт електронска адреса, за да заинтересираните чинители во доменот можат да пријават проблеми со податоците или пристапот до нив, како и да можат да дадат повратен одговор и мислење. Добиените пријави за проблеми со податоците и податочното множество е важно да бидат навремено одговорени и навремено да се преземат потребните чекори; во спротивно, нивото на корисност на самото податочно множество значително опаѓа.

5.2.5 Чекор 5: Кориснички сценарија и апликации

Петтиот чекор се однесува на дефинирањето кориснички сценарија и / или развој на конкретни апликации и сервиси кои ги искористуваат податоците од поврзаното податочно множество, со цел да се прикажат можностите за (повторно) искористување на податочното множество и неговите линкови кон други податочни множества со

поврзани податоци. Таквиот чекор ќе им послужи како пример на заинтересираните чинители во иднина, кои ќе можат да го увидат потенцијалот од искористувањето на контекстуално поврзаните податочни множества.

Корисничките сценарија можат да бидат описни сценарија, конкретни SPARQL прашања или прототип апликации, кои ќе ги опишат начините на кои податоците од новото податочно множество можат да се пребаруваат, добиваат и користат. Во овој чекор, значаен фокус треба да им се даде на начините на кои линковите кон другите податочни множества од поврзани податоци можат да се искористат за добивање пристап до други податоци, недостапни во оригиналното податочно множество, со цел да се прошири контекстот. Со ова, објавувачот на податоци ќе им демонстрира на заинтересираните чинители дека оригиналното податочно множество има поголема вредност кога ќе се комбинира со податочни множества од ист или сличен контекст, наместо да се користи во изолирана околина. Покрај ваквите кориснички сценарија, истите ефекти на поврзаните податоци од податочното множество можат да се прикажат и преку развој на апликации и сервиси. Тие носат поголема видливост на генералната употребливост на поврзаното податочно множество, но вообичаено бараат повеќе време и труд.

Креираните кориснички сценарија, апликации и / или сервиси, треба да бидат споделени и соопштени пред јавноста, заедно со самото податочно множество и неговите VoID метаподатоци. За таа цел се препорачува користење на истите канали како и во претходниот чекор.

5.2.6 Модуларност

Со цел да им се помогне на идните објавувања на поврзани податоци во доменот, од страна на истите или други објавувачи на податоци, препорачуваме развој на компоненти за чекорите од методологијата, компоненти кои можат повторно да се искористат. Објавувачите на податоци треба да го изведат животниот циклус на нивното поврзано податочно множество како модуларен, т.е. да конструираат слабо-поврзани компоненти кои можат да се искористат повторно во рамките на истиот домен. Тука, под слабо-поврзани мислиме на компоненти кои можат да се користат независно едни од други доколку тоа е потребно, но кои исто така можат да формираат и заеднички работен тек за генерирање на висококвалитетно, 5-star поврзано податочно множество. Повторното искористување на таквите компоненти, како и во други случаи на развој на софтвер, го редуцира времето потребно за развој и ја зголемува продуктивноста [149, 154].

Чекор 1, кој се фокусира на проучувањето на доменот од интерес, не може да се енкапсулира во софтверска алатка или компонента. Сепак, здобиеното знаење од овој чекор помага во оформувањето на задачите во следните чекори, што значи дека тоа ќе биде интегрирано во рамки на алатките и компонентите од следните чекори. Доколку е таков случајот, идните објавувачи на податоци од доменот ќе можат побргу да поминат низ овој чекор, па дури и целосно да го прескокнат.

Задачите во Чекор 2, каде објавувачите на податоци треба да ја развие податочната шема и да го земе предвид порамнувањето на податоците со постоечки и идни податоч-

ни множества од доменот, можат да се развијат во посебна компонента, која може да се искористува повторно. Компонентата може да биде во форма на податочна шема дефинирана во RDFS или OWL, која го репрезентира вокабуларот / онтологијата која ќе се користи за анотација на изворното податочно множество, со цел да се трансформира во 4-star RDF податочно множество. Вокабуларот / онтологијата треба да се состои од постоечки или новокреирани класи и својства, кои обезбедуваат добро порамнување со постоечки податочни множества од доменот, но и кои овозможуваат идно порамнување со поврзани податочни множества кои ќе бидат објавени во рамките на истиот домен.

Трансформацијата на изворното податочно множество во поврзано податочно множество во Чекор 3 е процес кој може да се развие како компонента за повторна употреба. Сите софтверски алатки кои можат да се искористат во овој чекор бараат структурирано податочно множество како влез, користат мапирање од изворната податочна шема во RDF шема и генерираат RDF како излез. Според тоа, доколку објавувачот на податоци во Чекор 2 дефинира RDF шема која може повторно да се употреби, тој може да дефинира (а) предефиниран формат за изворното податочно множество, (б) мапирање на изворното податочно множество базирано на предефинираниот формат на влезни податоци и податочната шема од Чекор 2, (в) процес, софтверска компонента, алатка или веб базиран сервис кој ќе го искористи изворното податочно множество и мапирањето да го трансформира податочното множество во RDF формат. На пример, предефинираниот формат може да претставува XML, CSV или TSV датотека, или датотека во некој друг формат, мапирањето може да биде дефинирано во јазик поддржан од алатката за трансформација, како на пример [106], мапирање базирано на RDF додатокот на OpenRefine [37], итн. Самиот процес на трансформација може да биде енкапсулиран во автоматизирана скрипта кој го зема изворното податочно множество форматирано согласно предефинираниот формат, го испраќа до алатката за трансформација и го зема излезниот RDF. Ова може да се изведе на различни начини, во зависност од алатките кои се користат. Во случајот на D2R Server, податоците треба да се вчитаат во релациона база на податоци и серверот да биде стартуван со соодветниот мапирачки фајл. Во случајот на Virtuoso, податоците траба да бидат вчитани во релациона база на податоци или како CSV податоци, по што може да се активира трансформацијата базирана на соодветното мапирање. Во случај, пак, на OpenRefine / LODRefine, процесот треба да биде мануелен, со цел податоците да се вчитаат и да се аплицираат рачно чекорите за трансформација. Сепак, во овој случај може да се искористи и инстанца на BatchRefine [21], која обезбедува HTTP REST-базиран интерфејс над OpenRefine / LODRefine инстанца. Ова значи дека трансформацијата на изворното податочно множество може да се активира со HTTP POST повик до BatchRefine инстанца, во кој покрај изворното податочно множество се испраќа и мапирањето базирано на RDF додатокот на OpenRefine.

Во Чекор 4, објавувачот на податоци треба да го објави генерираното поврзано податочно множество на Веб, согласно препораките на W3C. Доколку во Чекор 3 се користи јавно достапна инстанца на Virtuoso или D2R Server, податочното множество е

веќе јавно достапно. SPARQL endpoint-ите на двете платформи дозволуваат пристап до податоците од податочното множество за пребарување и во вид на RESTful API. За овозможување пристап директно преку URI идентификаторите, важно е форматот на дефинираните URI идентификатори од Чекор 3 да одговара со доменот користен за генерирање и објавување на податоците. Во случај на користење на D2R Server, ентитетите од податочното множество по автоматизам се достапни преку нивните URI идентификаторите како HTTP идентификатори. Во Virtuoso, пак, потребно е да се дефинираат конкретни правила за презапишување на URL вредностите (URL rewrite rules, анг.): скриптата која го изведува овој процес може исто така да се развие како компонента за повторна употреба. Со цел да им се помогне на идните објавувачи на податоци во доменот, објавувачот треба да ја разгледа можноста за обезбедување сервис за објавување на податоците: сервисот би го добивал новото генерирано поврзано податочно множество, би го складирал во платформата и би го направил достапен на Веб преку соодветни HTTP URI идентификатори и SPARQL endpoint.

Корисничките сценарија, апликациите и сервисите од Чекор 5 зависат од објавувачот на податоци и неговата идеја за искористување на поврзаната природа на генерираното податочно множество. Сепак, во зависност од доменот, можно е да постои апликација или сервис која користи податоци од одреден RDF репозиториум во кој можат да се додадат новите генерирани поврзани податочни множества. Доколку објавувач на податоци ги искористи постоечките компоненти од животниот циклус, развиени од претходни објавувачи во дадениот домен, резултантното податочно множество ќе содржи ентитети од истиот тип, анотирани со истиот вокабулар или онтологија, па според тоа ќе биде и целосно порамнет со постоечкото податочно множество. Во тој случај, сервис кој им овозможува на новите објавувачи на податоци да ги додадат своите генерирани поврзани податочни множества од доменот, развиени со истите компоненти за повторна употреба, во ист, постоечки RDF репозиториум, би овозможил апликациите и сервисите изградени над него да можат директно и без пречки да ги користат и новите податоци.

Формализација

Практичниот дел од нашиот животен циклус на поврзани податоци, за даден домен (d), го дефинираме како n-торка (tuple, анг.) од слабо-поврзани компоненти:

$$T_d = (C_{2d}, C_{3d}, C_{4d}, C_{5d}) \tag{5.1}$$

каде C_{2d}, C_{3d}, C_{4d} и C_{5d} се компонентите за повторна употреба од Чекор 2, Чекор 3, Чекор 4 и Чекор 5, во доменот d, соодветно. Компонентите можат да бидат композитни, т.е. можат да се состојат од други помали компоненти и алатки кои имаат специфична задача во конкретниот чекор, како на пример $C_{3d} = (C_{3_a d}, C_{3_b d})$. Како што веќе елаборираме, Чекор 1 нема практична компонента; тој се фокусира на проучување на доменот со цел имплементација на специфичностите на доменот во следните чекори.

Кога објавувач на податоци работи на генерирање поврзано податочно множество во домен d и развие множество повторно употребливи компоненти: $C_{2d}, C_{3d}, C_{4d}, C_{5d}$, кои

можат да се искомбинираат во n-торка $T_d = (C_{2d}, C_{3d}, C_{4d}, C_{5d})$, тогаш идни објавувачи на податоци во истиот домен d би биле во можност повторно да ја искористат истата n-торка во целост, т.е. да го искористат целото множество развиени компоненти во дадениот редослед, со цел да го трансформираат изворното податочно множество со кое работат во поврзано податочно множество. Во овој случај, поврзаните податочни множества генерирани со истата n-торка T_d, би биле комплетно порамнети: ја користат истата податочна шема (C_{2d}), истиот предефиниран формат за изворното податочно множество и истиот процес за трансформација (C_{3d}). Ова би овозможило едноставна и комплетна консолидација на податочните множества, без потреба од дополнителни мапирања помеѓу онтологии и податочно порамнување. Ваквите дополнителни задачи сè уште преставуваат голем предизвик во доменот на управување на податоци [178], што го прави овој придонес од повторна употреба на компонентите уште позначаен.

Во случај кога објавувач на податоци има изворно податочно множество кое делумно се разликува од податочното множество за кое е креирана n-торка за повторна употреба $T_d = (C_{d2}, C_{3d}, C_{4d}, C_{5d})$ тој може да ја модифицира податочната шема C_{2d} и да добие C'_{2d}. Модификацијата на податочната шема би повлекла модификација на процесот на трансформација, што значи дека објавувачот на податоци би морал да дефинира нова компонента за повторна употреба, C'_{3d}. Зависно од имплементацијата на C_{4d} и C_{5d}, тие би можеле да бидат модифицирани или директно да бидат употребени. Во случај да останат непроменети, новата n-торка би ја добила следната форма:

$$T'_d = (C'_{2d}, C'_{3d}, C_{4d}, C_{5d}) \tag{5.2}$$

каде C'_{2d} и C'_{3d} се новите, модифицирани компоненти за Чекор 2 и Чекор 3. Овие компоненти можат исто така да бидат објавени и достапни за идните објавувачи на податоци во домен d, заедно со оригиналните компоненти C_{2d} и C_{3d}. Во овој случај, новото поврзано податочно множество нема да биде комплетно порамнето со податочните множества генерирани со n-торката T_d, но податочните множества ќе бидат значително поблиски од аспект на податочната шема, во споредба со случај во кој новата податочна шема е развиена од почеток.

Генерално, објавувач на податоци од домен d не мора да дефинира повторно употребливи компоненти за сите чекори, како што е случајот со n-торката 5.1. Постојат уште неколку валидни комбинации, како што е $T_d = (C_{2d})$, каде само податочната шема е дефинирана како повторно употреблива компонента; $T_d = (C_{2d}, C_{3d})$, каде податочната шема, предефинираниот формат за изворното податочно множество и процесот за трансформација се дефинирани како повторно употребливи компоненти; $T_d = (C_{2d}, C_{3d}, C_{4d})$, каде е дефиниран и сервис за објавување на податочните множества; $T_d = (C_{4d})$, каде е дефиниран само таков сервис за објавување на податочните множества, без шемата и процесот за трансформација; $T_d = (C_{4d}, C_{5d})$, каде е дефиниран и сервис за додавање на податочните множества во податочното ниво на постоечки апликации и сервиси; $T_d = (C_{5d})$ каде таквиот сервис е единствената повторно употреблива компонента која е дефинирана. Во сите случаи, компонентите наменети за повторна употреба дефинира-

ни како дел од n-торката ќе овозможат поефикасно извршување на животниот циклус од страна на идни објавувачи на податоци во истиот домен.

Потенцијални недостатоци

Како што посочуваат авторите на [153], ваквиот пристап на однапред планирана повторна употреба има неколку потенцијални недостатоци: (а) развојот на компоненти кои можат повторно да се употребат може да биде поскапо отколку развој на стандардни компоненти [114], (б) одлуката кои компоненти да се развијат за повторна употреба не е едноставна задача [191] и (в) компонентите за повторна употреба често се развиваат со одредени претпоставки, кои можат да го ограничат опсегот на употребливоста во иднина [89]. Сепак, постојат одредени разлики помеѓу дизајнирање на компоненти од животниот циклус на поврзано податочно множество како повторно употребливи компоненти и дизајнирање на општи софтверски компоненти како такви.

Првиот недостаток, дополнителната цена, не важи за алатките и компонентите од Чекор 2 и Чекор 3. Податочната шема и процесот за трансформација мора да бидат развиени во секој случај, дури и повторната употребливост да не е од интерес. Објавувањето на податочната шема (C_{2d}), предефинираниот формат за изворното множество и мапирачките датотеки (C_{3d}) на локација каде идни објавувачи на податоци во доменот би можеле да им пристапат, не внесува дополнителни трошоци во процесот. Обезбедувањето на јавна инстанца на BatchRefine, Virtuoso или D2R Server за процесот на трансформација, на пример, може да додаде дополнителни трошоци, но имајќи предвид дека овие алатки се алатки со отворен код, објавувањето на податочната шема и мапирачките датотеки би обезбедило доволно ниво на повторна употребливост на процесот за идните објавувачи. Компонентите од Чекор 4 и Чекор 5 би барале дополнителни ресурси од првите објавувачи во доменот, бидејќи развојот и поставувањето на сервис за објава на податочни множества од други објавувачи не е дел од основниот животен циклус. Сепак, мотив за објавувачите на податоци да обезбедат такви сервиси би можела да биде идната достапност на нови податочни множества од доменот - генерирани со истите компоненти за повторна употреба - во рамките на апликациите и сервисите изградени над нив. Овозможувањето на идни објавувачи на податоци да ги објават своите генерирани податочни множества од доменот во заедничка платформа, би обезбедило дополнителни податоци во RDF складот на платформата, кој од друга страна се користи од страна на апликации и сервиси изградени над податоците. Доколку таквите апликации и сервиси му припаѓаат на објавувачот кој ја креирал самата платформа, тој ќе добие проширено податочно множество како податочно ниво. Доколку апликациите и сервисите припаѓаат на некои други чинители, тогаш би можеле да се применат различни економски модели кои би ја направиле платформата исплатлива опција.

Вториот недостаток, проблемот со одлучување кои компоненти да се развијат со повторна употреба на ум, е надминат со нашата дефиниција на можната структура на n-торката T_d. Објавувачот на податоци од доменот може да одбере која од формите на

n-торката ќе ја развие како множество од повторно употребливи компоненти за идните објавувачи.

Третиот недостаток, ограниченоста на повторната употреба поради претпоставки направени во текот на имплементацијата, е избегната поради опсегот на компонентите и n-торката: доменот. Во равенката 5.1, компонентите и n-торката се однесуваат на доменот d во рамките на кој можат да се користат. Оттука, дадена компонента наменета за повторно искористување, како што е C_{3d}, како опсег го има доменот d и не би можело погрешно да се претпостави дека истата може да се искористи и во друг домен. Истата логика важи и на нивото на под-компоненти, $C_{3d} = (C_{3_a d}, C_{3_b d})$, како и на нивото на n-торката, $T_d = (C_{2d}, C_{3d})$, каде и под-компонентите и n-торката како опсег го имаат доменот d.

5.3 Евалуација на методологијата

Со цел да ја евалуираме методологијата и предложените препораки, го одбравме доменот на лекови. Како изворни податочни множества ги користиме официјалните национални регистри на лекови кои ги содржат лековите регистрирани за продажба во земјата. Со цел да го изведеме процесот на генерирање поврзани податочни множества со лекови од сите изворни држави, развивме n-торка наменета за повторно искористување во доменот на регистрирани лекови:

$$T_{drugs} = (C_{2drugs}, C_{3drugs}, C_{4drugs}) \tag{5.3}$$

каде $C_{3drugs} = (C_{3_a drugs}, C_{3_b drugs}, C_{3_c drugs})$ е композитна компонента за Чекор 3. Со тоа, n-торката ја добива формата:

$$T_{drugs} = (C_{2drugs}, (C_{3_a drugs}, C_{3_b drugs}, C_{3_c drugs}), C_{4drugs}) \tag{5.4}$$

Ќе направиме преглед на компоненти за повторна употреба во доменот на лекови, а потоа ќе презентираме проект чија цел е евалуација на методологијата, проект во кој ги употребуваме предложените методолошки насоки и конкретните компоненти. Системот собира податоци за лекови кои се наоѓаат во продажба во дваесет и три различни земји, преку нивните официјални регистри на лекови, извршува чистење на податоците, ги порамнува, ги трансформира во висококвалитетни поврзани податоци и ги објавува на Веб во заедничко, порамнето и консолидирано поврзано податочно множество за лекови.

5.3.1 Компоненти за повторна употреба во доменот на лекови

Како дел од насоките во методологијата и со цел да им се помогне на објавувачите на податоци кои работат во доменот на лекови, дизајниравме и развивме множество алатки во вид на повторно употребливи компоненти. Алатките вклучуваат RDF податочна шема (C_{2drugs}), предефиниран CSV формат за влезните податоци ($C_{3_a drugs}$),

OpenRefine скрипта за трансформација ($C_{3_b drugs}$), SPARQL-базирана алатка за проширување и меѓусебно поврзување на податочното множество ($C_{3_c drugs}$) и веб базирана алатка за автоматска трансформација, поврзување и објавување (C_{4drugs}) на генерираното поврзано податочно множество со лекови.

RDF податочна шема (C_{2drugs})

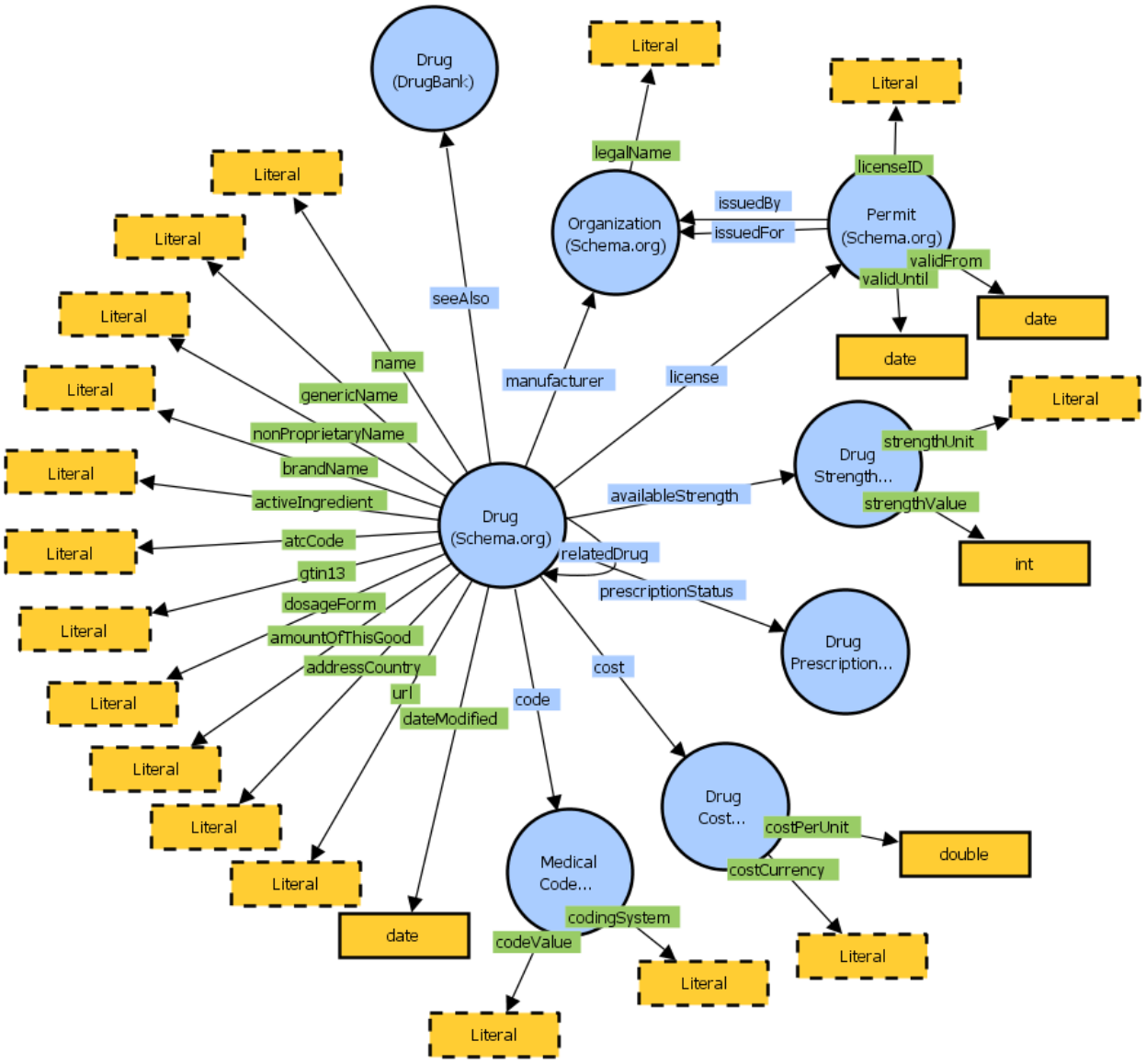
Со цел да го моделираме доменот на лекови регистрирани за продажба на глобално ниво, согласно Чекор 2 од методологијата, беше потребно да креираме една заедничка шема која може повторно да се употребува за сите национални регистри со лекови, која потоа ќе се искористи за анотација на изворните податоци за лекови. Со ова, целта беше да се обезбеди порамнување на податоците за лекови достапни од различни извори, во различни формати и со различни нивоа на грануларност, со цел да се овозможи поедноставно искористување на конечното податочно множество.

Прво, ги анализиравме националните регистри со лекови на 31 држава[1], што нѝ помогна да дефинираме заедничко множество својства кои постојат и кои би можеле да се од корист во резултантното податочно множество. Множеството се состоеше од 24 својства, вклучувајќи го продажното име на лекот, генеричкото име, ATC кодот, EAN кодот (бар-код), листата од активни супстанци, јачината на лекот, дозирањето, цената, производителот, државата во која е регистриран за употреба, детали за лиценцата, итн. Важно е да се напомене дека не сите изворни регистри на лекови ги обезбедуваат сите податоци и својства селектирани за нашата податочна шема. И покрај тоа, одлучивме тие својства да останат во податочната шема, поради тоа што се корисни кога се достапни.

Следејќи ги најдобрите практики за користење на онтологии и вокабулари [78], започнавме со анализа на постоечките вокабулари и онтологии со цел да ги идентификуваме класите и својствата кои би можеле да ги искористиме. Како појдовна точка го користевме множеството својства кои ги селектиравме од изворните регистри на лекови. Анализата покажа дека Schema.org вокабуларот [52] комплетно одговара на нашите својства од доменот. Schema.org вокабуларот содржи дефиниција за класата `schema:Drug` и содржи голем број својства за неа [61]. Поради тоа, RDF податочната шема која ја дефиниравме за изворните податоци (Слика 5.2) се базира на Schema.org вокабуларот. Дополнително, таа користи класи и својства и од DrugBank и RDFS онтологиите, но тоа ќе го објасниме подолу во текстот.

Schema.org е заедничка иницијатива на Google, Bing, Yahoo и Yandex, како унифициран вокабулар наменет за структурирано означување на содржината на веб страни [118, 181, 152]. Тој се користи од овие пребарувачи со цел да можат да ги прикажат резултатите од пребарувањето во структурирана форма која нуди специфични акции за интеракција, во зависност од типот на ентитетите за кои се однесуваат податоците: луѓе,

[1] Австрија, Азербејџан, Белгија, Египет, Естонија, Агенцијата за лекови на Европската Унија, Ирска, Италија, Јужноафриканска Република, Канада, Кипар, Костарика, Македонија, Малта, Нигерија, Нов Зеланд, Норвешка, Романија, Русија, Словачка, Словенија, Србија, Уганда, Украина и САД, Унгарија, Франција, Холандија, Хрватска, Чешка и Шпанија.



Слика 5.2: RDF податочната шема искористена за анотација на поврзаното податочно множество. Составена е главно од Schema.org вокабуларот, со неколку помошни елементи од DrugBank онтологијата и RDFS. Лековите во податочното множество се инстанци од класата `schema:Drug`, прикажана централно на сликата.

настани, продукти, филмови, ТВ серии, ресторани, книги, музички албуми, итн. Покрај тоа, Google го користи вокабуларот во рамките на својот Knowledge Graph, Gmail и Microsoft Cortana го користат во електронски пораки кои се однесуваат на резервации и сметки (од ресторани, хотели, авио-компании, итн.), додека Apple го користи во рамките на Siri [119]. Употребата на Schema.org вокабуларот на Вебот е во постојан раст во последните неколку години, раст кој е неспоредливо поголем со оној на поригорозните, генерални вокабулари и онтологии развиени пред него [155]. Неговиот успех главно се базира на неговата едноставност: тој користи генерално рамна хиерархија на класи, за да пречките за негова имплементација од страна на објавувачи на податоци, развивачи на веб страни и веб администратори се држат доволно ниско.

Растот на употребата на Schema.org вокабуларот, како и неговиот широк спектар на покриени домени, го става во позиција во која може да се користи за порамнување на постоечки онтологии и податочни множества. Овој тренд е присутен и во рамките на доменот на здравство [182]. Поради тоа, одлучивме да го користиме Schema.org вокабуларот наместо онтологии специфични за доменот [124], со цел да обезбедиме заедничка податочна шема за лековите на глобално ниво.

За да обезбедиме порамнување помеѓу поврзаните податочни множества кои ќе се генерираат со оваа RDF податочна шема и постоечките LODD и DrugBank податочни множества, ги одбравме двете класи, `schema:Drug` и `drugbank:drugs`, како класи за нашите ентитети кои претставуваат лекови. За поврзување на лековите со соодветните генерички лекови од LOD облакот, ја одбравме `rdfs:seeAlso` релацијата, како најмногу користена релација за поврзување на слични ентитети во рамките на самиот LOD облак [179].

Како и секоја друга RDF податочна шема, вокабулар и онтологија, нашата податочна шема (Слика 5.2) може да еволуира со текот на времето; може да биде проширувана и модифицирана во иднина од наша страна или од страна на други чинители во доменот, согласно со промените во доменот на лекови.

Предефиниран CSV формат за влезните податоци ($C_{3_a drugs}$)

Со цел да им овозможиме објавувачите на податоци да ги анотираат своите податоци за лекови со RDF податочната шема од Слика 5.2, како што е дефинирано во Чекор 3, имаше потреба да дефинираме конкретен формат за влезните податоци, со кој би биле подготвени за процесот на трансформација. Оттука, креиравме CSV предефиниран формат, кој е јавно достапен [135]. Предефинираниот формат содржи 39 колони кои се однесуваат на потребните податочни полиња од влезните податоци, за процесот на трансформација кој следува. Помеѓу нив се наоѓаат URI идентификаторот на лекот, продажното име, едно или повеќе генерички имиња, еден или повеќе производители, ATC кодот, една или повеќе активни супстанци, јачината, цената, итн. Колоните се моделирани да одговараат со RDF податочната шема и ги опфаќаат сите податоци неопходни за висококвалитетно моделирање на доменот.

Податочниот тип на колоните најчесто е текст вредност, со неколку исклучоци. Некои од позначајните забелешки за податочните типови се: јачината на лекот е одвоена во колона со целобројна вредност во која е означена јачината, додека единицата мерка за јачината е дел од посебна текстуална колона; на сличен начин, цената на лекот е одвоена во бројчана вредност со децимална запирка и текстуална вредност за валутата, каде валутата е означена согласно соодветниот ISO стандард [27]; во неколкуте колони кои содржат датум, вредностите мора да бидат во форматот "YYYY-MM-DD"; типот на лекот од аспект на начинот на препишување треба да биде "OTC" за лекови кои се издаваат без рецепта, или "PrescriptionOnly" за лековите за кои рецепта е задолжителна; земјата во која е регистриран лекот мора да биде означена согласно кодот на земјата според соодветниот ISO стандард [26]; доколку лекот има повеќе генерички имиња, про-

изводители или активни супстанци, нивните вредност треба да се одвојат и да стојат една по една во соодветните `genericNameN`, `manufacturerN` и `activeSubstanceN` колони, соодветно, etc. Деталите за останатите податочни типови на колоните се достапни на веб страната на проектот [135].

Предефинираниот CSV формат користи вертикална линија (`|`) како карактер за одвојување на вредностите, поради тоа што стандардните карактери за одвојување во CSV датотеки, како што се запирка (`,`) и точка-запирка (`;`) често се среќаваат во вредностите на колоните кога се работи со податоци за лекови, што претставува проблем при интерпретација на вредностите од CSV датотеката. Важно е да се напомене дека редоследот на колоните во CSV датотеката не е важен, доколку се користи нашата OpenRefine скрипта за трансформација.

Како што е случајот и со RDF податочната шема, предефинираниот CSV формат е отворен и јавно достапен, па според тоа може да биде прошируван или модифициран во иднина од наша страна или од страна на други чинители во доменот, како што еволуира истражувачкото поле на податоци за лекови.

OpenRefine скрипта за трансформација ($C_{3_b drugs}$)

Чекор 3 од методологијата ја содржи задачата за трансформација на изворните податоци во семантички анотирани податоци согласно RDF податочната шема од Чекор 2. Со оглед на тоа што дефинираме RDF податочна шема која може да се користи во доменот на податоци за лекови регистрирани во различни држави, развиваме и обезбедиме алатка за автоматизација на процесот на трансформација, како повторно употреблива компонента. Оваа алатка обезбедува анотација на изворните податоци со дефинираната RDF податочна шема, со цел генерирање на поврзано податочно множество со порамнети и висококвалитетни податоци од доменот на лекови. Целта на алатката е да ги намали пречките во процесот на трансформација на изворните податоци во RDF и во поврзани податоци за оние објавувачи кои немаат доволно искуство во работата со семантички и поврзани податоци, како и за искусните објавувачи на поврзани податоци кои не се детално запознаени со доменот на лекови и здравство.

Алатката за генерирање на поврзано податочно множество ја развиваме во вид на OpenRefine скрипта за трансформација. OpenRefine [48] е софтверска алатка со отворен код која работи со структурирани податоци, како CSV, TSV, XML, итн. Таа им овозможува на корисниците работа со големи податочни множества: корисниците можат да ги изведат саканите модификации на мало подмножество редици од изворните податоци, а потоа да ги аплицираат над целиот изворен документ. Тука, модификациите кои можат да се направат вклучуваат податочни трансформации, спојувања, прочистувања, итн. Алатката располага и со RDF додаток, кој овозможува усогласување на вредностите на ќелиите со RDF податоци достапни преку даден SPARQL endpoint. Ова овозможува поврзување на вредностите од ќелиите со ентитети од RDF податочни множества, најчесто за недвосмислена идентификација на ентитетите од изворното податочно множество. RDF додатокот овозможува и мапирање на изворните податоци во RDF, преку

дефинирање на т.н. 'RDF скелет'. Излезот од оваа акција е RDF датотека генерирана од изворното податочно множество, согласно дефинициите во RDF скелетот.

Можноста на OpenRefine да се снимат акциите на корисникот и да се експортираат во JSON формат, овозможува повторно искористување на акциите над различни изворни податочни множества. Ова нѝ овозможува да ја дефинираме потребната трансформација на податоците, која потоа може повторно да се употреби над различните изворни податочни множества за лекови, кои ги имаат истите колони во CSV форматот. Со оглед на тоа што дефинираме наш предефиниран CSV формат за влезните податоци за лекови, ваквата трансформација можеме да ја додадеме кон нашето множество алатки. Листата со предефинирани акции за трансформација на влезното податочно множество со лекови, која ја формира нашата OpenRefine скрипта за трансформација, е отворена и јавно достапна како компонента за повторно искористување [135].

Нашата OpenRefine скрипта за трансформација е дизајнирана за да работи со предефинираниот CSV формат, а на излез генерира поврзано податочно множество за лекови согласно нашата RDF податочна шема. Скриптата за трансформација содржи три акции:

А. усогласување на колоните `genericName1`, `genericName2`, ..., `genericName5` со податоци од DBpedia,
Б. усогласување на колоната `atcCode` со податоци од DrugBank, и
В. креирање на RDF шема за податоците

Акцијата А. ја користи функционалноста на RDF додатокот за OpenRefine алатката, кој ја користи вредноста од одредена ќелија од одредена колона во изворната датотека за да детектира потенцијални ентитети од одреден SPARQL endpoint кои можат да соодветствуваат на ентитетот претставен преку самата редица. Во нашиот случај, ги користиме петте `genericName` колони - каде секоја од нив содржи по едно од генеричките имиња на активната супстанца на лекот - и се обидуваме да најдеме соодветни ентитети за генерички лекови од DBpedia SPARQL endpoint-от, со помош на нивната `rdfs:label` вредност. Доколку сервисот за усогласување на OpenRefine детектира потенцијален кандидат лек, инстанца од класата `dbo:Drug`, кој го претставува генеричкиот лек означен во соодветната ќелија од нашето податочно множество, го користиме во чекор В. за да креираме RDF тројка која ќе го поврзе лекот од нашето податочно множество со пронајдениот генерички лек од DBpedia, со користење на `rdfs:seeAlso` релацијата, на пример:

Пример 5.1

```
@prefix dbp: <http://dbpedia.org/resource/>
@prefix mkd: <https://lekovi.zdravstvo.gov.mk/drugsregister/detailview/>

mkd:55446 rdfs:seeAlso dbp:Clopidogrel .
```

Акцијата Б. изведува слично усогласување, но овој пат на колоната `atcCode` од нашето податочно множество во CSV и вредноста на ATC кодовите на лековите достапни преку DrugBank SPARQL endpoint-от. OpenRefine се обидува да пронајде совпаѓања помеѓу вредностите во колоната `atcCode` на наша страна и вредностите на `drugbank:atcCode` релацијата во инстанците од `drugbank:drugs` класата кои постојат во RDF податочното множество на DrugBank. За разлика од состојбата кај А., тука имаме можност еден наш лек да се совпадне со повеќе од еден генерички лек од DrugBank. Причината е поради тоа што повеќе `drugbank:drugs` инстанци од DrugBank можат да имаат ист ATC код, т.е. ги поседуваат истите терапевтски, фармаколошки и хемиски својства. Слично како и во чекор А., ги користиме сите совпаднати кандидати од процесот на усогласување во чекор В., преку креирање RDF тројки кои го поврзуваат нашиот лек со соодветните генерички лекови од DrugBank, како на пример:

Пример 5.2

```
@prefix mkd: <https://lekovi.zdravstvo.gov.mk/drugsregister/detailview/>
@prefix dbd: <http://wifo5-04...uni-mannheim.de/drugbank/resource/drugs/>

mkd:841690570 rdfs:seeAlso dbd:DB00201 ;
              rdfs:seeAlso dbd:DB00316 .
```

Акцијата В. го креира RDF скелетот согласно шемата на податоците, која всушност претставува мапирање на консолидираната CSV датотека во RDF. Согласно RDF шемата на податоците (Слика 5.2), дефинираме мапирање помеѓу колоните од CSV датотеката и конкретни RDF шеми од тројки. Дел од мапирањата се едноставни, како на пример мапирањата на продажното име на лекот, генеричкото име, дозирањето, државата, URL вредноста, описот, итн. За нив, го користиме URI идентификаторот на лекот како субјект во RDF тројката, го назначуваме конкретното својство како релација во тројката и ја користиме вредноста на соодветната колона како вредност или објект во тројката. На пример, продажните имиња на лековите се мапираат во RDF тројки во следниот формат:

Пример за мапирање 5.1

```
<drug-URI> schema:name <value-of-brandName-column> ;
           drugbank:brandName <value-of-brandName-column> .
```

Сепак, дел од мапирањата се покомплексни. Мапирањата на вредностите како што се ATC кодот, цената, јачината, производителот, деталите за лиценците, итн., бараат креирање на нови ентитети од различни типови. На пример, со цел додавање на информациите за ATC кодот кон креираната инстанца на лекот, потребно е да креираме нов празен јазол од типот `schema:MedicalCode`, кој има две дополнителни тројки: една со `schema:codeValue` релацијата и друга со `schema:codingSystem` релацијата. Според тоа, мапирањето на ATC кодот може да се претстави како:

Пример за мапирање 5.2

```
<drug-URI> schema:code <blank-node-ID> .
<blank-node-ID> rdf:type schema:MedicalCode ;
                schema:codeValue <value-of-atcCode-column> ;
                schema:codingSystem ‘‘ATC’’ .
```

Мапирањето на информациите за лиценцата на лекот е најкомплексно, како што може да се види на Слика 5.2 и во продолжение:

Пример за мапирање 5.3

```
<drug-URI> schema:license <blank-node-1-ID> .
<blank-node-1-ID> rdf:type schema:Permit ;
      schema:licenseID <value-of-licenseNumber-column> ;
      schema:validFrom <value-of-licenseValidFrom-column>^^xsd:date ;
      schema:validUntil <value-of-licenseValidUntil-column>^^xsd:date ;
      schema:issuedBy <blank-node-2-ID> ;
      schema:issuedFor <blank-node-3-ID> .
<blank-node-2-ID> rdf:type schema:Organization ;
      schema:legalName <value-of-licenseIssuedBy-column> .
<blank-node-3-ID> rdf:type schema:Organization .
      schema:legalName <value-of-licenseIssuedFor-column> .
```

За среќа, корисничкиот интерфејс на OpenRefine и BatchRefine нуди можност за едноставно креирање на овие мапирања со помош на визуелна репрезентација и минимално пишување дополнителен код. Дополнителниот код, во нашиот случај, беше потребен за користење на сите пронајдени кандидат лекови при процесот на усогласување во чекор Б., каде со користење на GREL јазикот ги селектиравме сите пронајдени генерички лекови за поврзување и ги искористивме во соодветните `rdfs:seeAlso` тројки.

Како резултат од скриптата за трансформација, се добива поврзано податочно множество за лекови, кое содржи линкови кон LOD облакот. Слично како и останатите компоненти, скриптата за трансформација е достапна како отворена JSON датотека, која може да се проширува и модифицира во иднина.

SPARQL-базирана алатка за проширување и поврзување на податочното множество ($C_{3_c drugs}$)

Откако податочното множество е трансформирано во поврзано податочно множество, со помош на наведените алатки, следна неопходна акција според Чекор 3 е додавањето линкови помеѓу лековите од податочното множество кои ја имаат истата функција, односно ги имаат истите терапевтски, фармаколошки и хемиски својства. Ова би резултирало со подобра основа за идни кориснички сценарија, како што ќе видиме подоцна.

Притоа, целта е да се креираат линкови помеѓу два лека од нашето податочно множество кои ја имаат истата функција, односно кои се наменети за истата состојба на пациентот, па поради тоа ги користиме ATC кодовите на лековите. Според Светската здравствена организација (СЗО) и нејзината шема за кодирање на лековите [3], два лека кои го имаат истиот ATC код, имаат и иста функција. За таа цел, дефинираме SPARQL прашање [135] кое може повторно да се употребува, кое ги детектира сите парови на лекови од самото податочно множество кои го имаат истиот ATC код и ја користиме `schema:relatedDrug` релацијата за да креираме пар RDF тројки за нив, како на пример:

Пример 5.3

```
@prefix rus: <http://www.vidal.ru/drugs/>
@prefix mkd: <https://lekovi.zdravstvo.gov.mk/drugsregister/detailview/>

rus:trombopol__22439 schema:relatedDrug mkd:51201 .
mkd:51201 schema:relatedDrug rus:trombopol__22439 .
```

Овие две тројки креираат двонасочна врска помеѓу лековите во податочното множество, означувајќи ја нивната функционална поврзаност. SPARQL прашањето ги зачувува новите RDF тројки во истиот RDF граф каде што веќе е сместено податочното множество. Овие поврзувања можеме да ги искористиме во кориснички сценарија во кои на крајните корисници им се обезбедува листа на алтернативни лекови кои одговараат на нивната состојба, а кои се наоѓаат во истата или можеби во друга држава.

Со оглед на фактот дека не сите регистри на лекови содржат ATC код како информација за регистрираните лекови и со цел да го зголемиме бројот на меѓусебно поврзани лекови во податочното множество како поддршка за кориснички и аналитички сценарија, дефинираме дополнително SPARQL прашање [135]. Прашањето е генерализирано и може да се употребува повторно, при што се обидува да им додели ATC кодови на сите лекови од податочното множество кои ја немаат оваа информација. SPARQL прашањето ги детектира лековите од податочното множество кои немаат ATC код, потоа за секој од нив го наоѓа генеричкиот лек од DBpedia со кој соодветниот лек е поврзан преку `rdfs:seeAlso` релација, го зема ATC кодот на генеричкиот лек од DBpedia и го доделува на лекот од нашето податочно множество. Бидејќи претходното SPARQL прашање за меѓусебно поврзување на лековите зависи токму од ATC кодовите, ова SPARQL прашање за проширување на податочното множество со ATC кодови кои недостасуваат треба да се изврши прво.

Двете SPARQL прашања се параметризирани. Тие можат да се извршат над складиштето за поврзани податоци во кое се наоѓа поврзаното податочно множество генерирано со претходните алатки и компоненти.

Веб базирана алатка за автоматска трансформација, поврзување и објавување (C_{4drugs})

Според препораките од Чекор 4, генерираното поврзано податочно множество со лекови треба да се објави на Веб согласно принципите на поврзани податоци и најдобрите практики. Со цел да им помогнеме на објавувачите на податоци од овој домен, Чекор 4 може автоматски да се изведе со помош на нашата веб базирана алатка. Објавувачите на податоци можат да ги постават своите генерирани поврзани податочни множества на веб страната на проектот, LinkedDrugs [136], каде после соодветна проверка на квалитет од наша страна, податочното множество ќе биде автоматски објавено. За ова користиме наша јавно достапна инстанца на Virtuoso [66], преку која новото податочно множество е достапно на Веб во вид на поврзано податочно множество, преку нејзиниот SPARQL endpoint [57]. Идентификаторот на RDF графот во кој се складира поврзаното податочно множество му се соопштува на објавувачот по успешниот процес на објава.

Покрај објавување на комплетирано поврзано податочно множество за лекови, веб базираната алатка и нејзиниот автоматизиран процес можат да ги извршат и претходните чекори од методологијата за објавувачот: (а) да генерираат поврзано податочно множество од влезна CSV датотека и (б) да го прошират податочното множество со ATC кодови кои недостасуваат и да ги поврзе лековите меѓусебно со `schema:relatedDrug` релации, од влезна RDF датотека. За првото, поставената CSV датотека треба да е форматирана согласно нашиот предефиниран CSV формат, за да може нашата веб базирана алатка и нејзиниот серверски процес врз база на неа, предефинираната RDF податочна шема и OpenRefine скриптата за трансформација, да генерира поврзано податочно множество. Со користење на SPARQL-базираната алатка, веб алатката ќе го прошири податочното множество со ATC кодовите кои недостасуваат и ќе генерира линкови помеѓу самите лекови од податочното множество, врз база на нивните ATC кодови. За второто, пак, веб базираната алатка веднаш преминува на додавање на `drugbank:atcCode` релациите кои недостасуваат и `schema:relatedDrug` релациите помеѓу сличните лекови од поставеното податочно множество во RDF формат. Со ова, обезбедуваме можност најголемиот дел од задачите за процесирање на изворните податоци да се оддалечи од објавувачите на податоците, со цел да им се поедностави нивниот работен тек.

Кога објавувач на податоци ја користи нашата веб базирана алатка на [136] за да го објави своето поврзано податочно множество со лекови, нашиот систем го додава и во рамките на глобалното поврзано податочно множество на лекови - LinkedDrugs податочното множество - така што го складира во друг, дополнителен RDF граф. Во дополнителниот граф, алатката потоа генерира нови `schema:relatedDrug` тројки, со цел да ги поврзе лековите од новото податочно множество со соодветните слични лекови од постоечките податочни множества во LinkedDrugs, но и обратно. Со тоа, LinkedDrugs податочното множество содржи податоци за лекови кои се во продажба во повеќе различни земји, податоци обезбедени од различни објавувачи, вклучувајќи го и нашиот тим. Ова податочно множество е достапно преку перзистентно URI достапно преку Вебот, кое поддржува HTTP content negotiation [49].

5.3.2 Примена на методологијата во доменот на податоци за лекови

Откако ги развивме компонентите наменети за повторна употреба во доменот на лекови, ја применивме n-торката $T_{drugs} = (C_{2drugs}, (C_{3_adrugs}, C_{3_bdrugs}, C_{3_cdrugs}), C_{4drugs})$, т.е. чекорите од методологијата, во рамките на конкретен проект. Проектот се состоеше од дизајн и развој на автоматизиран систем за трансформација и генерирање на поврзано податочно множество со лекови од дваесет и три држави: Австрија, Азербејџан, Египет, Естонија, Ирска, Јужноафриканска Република, Кипар, Костарика, Македонија, Малта, Нигерија, Нов Зеланд, Норвешка, Романија, Русија, САД, Словачка, Словенија, Србија, Уганда, Украина, Холандија и Шпанија. Државите беа одбрани на начин кој ја репрезентира глобалната разноликост и кој би демонстрирал дека е возможно сеопфатно решение за генерирање поврзани податоци за лекови на глобално ниво.

Автоматизираниот систем и неговиот работен тек преставуваат конкретен пример за примената на методологијата и повторно употребливите компоненти развиени како дел од оваа дисертација, според што служат како нивно евалуациско сценарио.

Генерирање на поврзаното податочно множество за лекови: LinkedDrugs

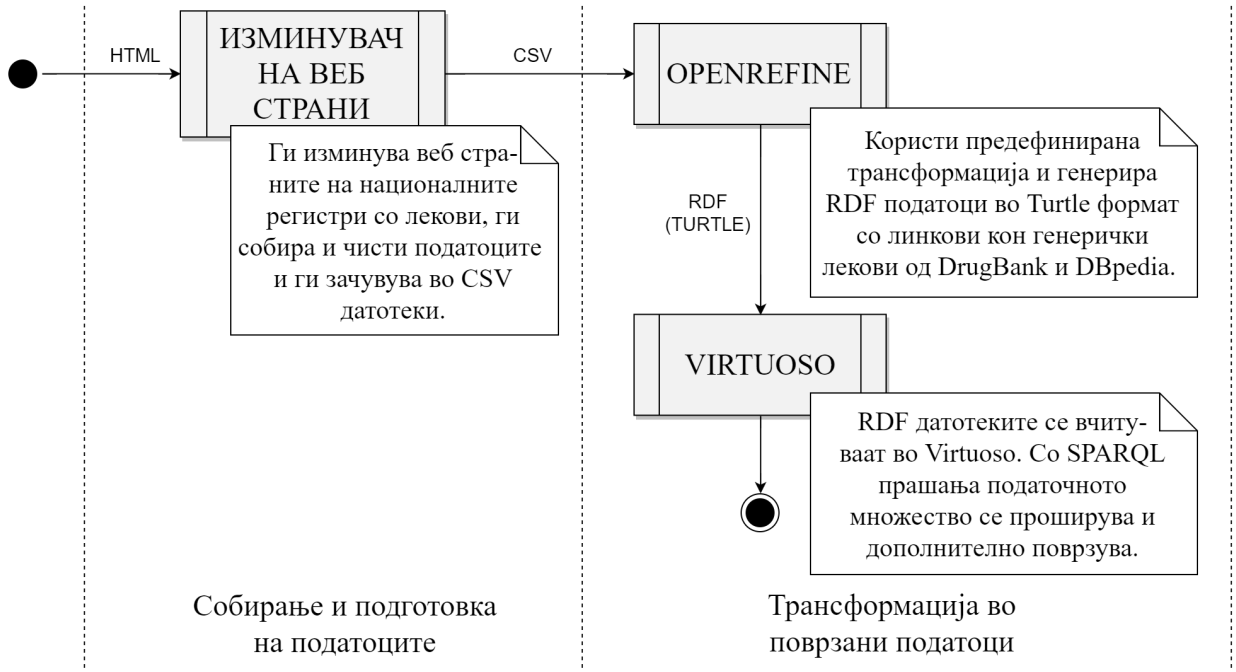
Националните регистри на лекови на голем број држави се достапни на Вебот. Како што веќе наведовме, направивме опсежна анализа на веб страните на националните регистри на лекови на голем број држави, со цел да дефинираме заедничко множество својства, т.е. податочна шема, која ќе ја искористиме за опис на консолидираните податоци. Овие активности за анализа на доменот и дефинирање на RDF податочната шема кореспондираат со активностите наведени во Чекор 1 и Чекор 2 од методологијата, кои се веќе завршени и се дел од C_{2drugs} компонентата. Според тоа, нив директно ги искористуваме во нашиот конкретен систем.

Со цел да го дизајнираме, тестираме и валидираме нашиот автоматизиран систем за собирање на податоци за лекови со квалитет од 2-ѕвезди од националните регистри на лекови и генерирање високо-квалитетни поврзани податоци за лекови на глобално ниво, селектиравме група од дваесет и три држави. Нашата цел се состоеше во селекција на широка група држави, од различни делови на светот.

Регистрите со лекови на овие држави се достапни онлајн. Веб страните каде можат да се најдат конкретните регистри на лекови за државите со кои работиме се наведени на страната на проектот на GitHub [135]. На веб страните на регистрите со лекови, податоците најчесто се достапни во структуриран формат наменет за користење од страна на луѓето, преку пребарување и прелистување на самите веб страни. Кај помал број држави, пак, податоците се достапни преку структурирани датотеки достапни за преземање, најчесто во Microsoft Excel или PDF формат.

Во ова поглавје ќе го опишеме автоматизираниот систем кој ги собира податоците, ги прочистува, ги подготвува согласно предефинираниот CSV формат за влезни податоци (C_{3_adrugs}), ја користи скриптата за трансформација (C_{3_bdrugs}) и SPARQL прашањата

($C_{3_c drugs}$) за да ги трансформира податоците во RDF формат, да ги прошири со ATC кодови кои недостасуваат и да додаде линкови помеѓу самите лекови во податочното множество, но и од нив кон лекови од податочните множества на DrugBank и DBpedia, со што податочното множество ефективно се претвора во поврзано податочно множество (Слика 5.3). Овие чекори ги претставуваат активностите од Чекор 3 од методологијата.



Слика 5.3: Работен тек: Трансформација на отворени податоци со 2-ѕвезди од различни национални регистри на лекови, во високо-квалитетни поврзани податоци за лекови.

Собирање и подготовки на податоците. Со цел да креираме одржлив систем за поврзани податоци за лекови, беше неопходно да се дизајнира механизам за собирање на податоците за лекови од државните регистри на лекови, на одреден период. Поради тоа, развивме апликација која ги изминува наведените веб страни со лекови од националните регистри, ги собира потребните информации од нив, ги прочистува и ги складира во предефиниран CSV формат (Слика 5.3). Апликацијата се состои од повеќе различни модули кои работат со различните формати на податоци достапни на изворните веб страни. Излезните CSV датотеки ја користат предефинираната CSV структура ($C_{3_a drugs}$).

Како и поголемиот дел од податоците достапни на Веб, податоците за лековите од националните регистри не се подеднакво структурирани, ниту се комплетно 'чисти'. Поради тоа, апликацијата која ги изминува наведените веб страни ја проширивме со дополнителни функционалности кои ги изведуваат задачите за чистење на податоците и кои имаат задача за детекција на сите варијации на одредени податочни вредности кај изворните веб страници.

Со цел да ги дефинираме URI идентификаторите за лековите во податочното множество, ги користевме постоечките URL вредности на самите веб страни каде што се опишани поединечните лекови, како на пример `https://lekovi.zdravstvo.gov.mk/`

`drugsregister/detailview/53457`. Согласно принципите на поврзани податоци, URI идентификаторите на ентитетите во множеството треба да бидат Веб локации каде корисниците и софтверските агенти можат да добијат повеќе информации за ентитетот, па нашиот пристап го задоволува овој услов.

Значителен дел од лековите имаат информација за повеќе од едно генерично име, повеќе од еден производител, активна супстанца и јачина, поради што апликацијата која ги собира податоците има задача и да ги одвои овие повеќекратни вредности во посебни и соодветни колони во CSV датотеките. Дополнително, информациите за цената на лекот и неговата јачина треба да се одвојат во посебни колони кои ја означуваат вредноста и валутата односно единицата мерка, како на пример *јачина: 500, единица мерка: mg*; *цена: 80, валута: MKD*. Апликацијата која ги собира податоците е задолжена и да се грижи за специфичните формати на податоци потребни за дел од колоните, како што се датумите, кодовите на држави, кодовите за валути и начинот на издавање на лекот.

Податоците за лековите од неколку држави асе исклучок, поради тоа што тие се достапни за преземање како Microsoft Excel или PDF датотеки. За овие држави, апликацијата за собирање на податоците има задача да ги реструктурира колоните од изворните податоци и да генерира CSV датотека која го следи нашиот предефиниран формат. За овие лекови, генерираме посебни URI идентификатори, кои го следат форматот `http://linkeddata.finki.ukim.mk/lod/data/loddw/drugs/{countryCode}#{drugID}`. Тука, `drugID` е идентификаторот за лекот генериран од самата апликација за собирање на податоците, `countryCode` е кодот на држава (согласно ISO стандардот [26]) на државата во која лекот е регистриран, додека останатите делови од URI идентификаторот се однесуваат на самиот проект, типот на податок и изворот на лекот во рамките на нашата веб локација за поврзани податоци: `/lod/data/loddw` е проектот и `/drugs` е категоризацијата на податоците.

Резултатот во оваа фаза од работниот тек (Слика 5.3), во нашиот конкретен случај со дваесет и три држави, е множество од дваесет и три посебни CSV датотеки кои ја имаат истата податочна шема, т.е. ги содржат истите колони, во истиот редослед, согласно компонентата $C_{3adrugs}$. Единствената разлика е во тоа што некои од CSV датотеките може да се случи да немаат податоци во некоја од колоните, во случаи кога тие податоци не се достапни на самите национални регистри објавени на Вебот. Откако ги имаме сите дваесет и три CSV датотеки, исчистени и спремни во соодветниот формат, првиот дел од работниот тек е завршен и можеме да продолжиме со вториот дел.

Трансформација во поврзани податоци. CSV датотеките можат да бидат искомбинирани во една CSV датотека, или пак да останат одвоени. Единствената разлика ќе бидат перформансите во следниот чекор кој може да се изведе како еден подолг процес, или како дваесет и три одвоени и побрзи процеси. Со цел да добиеме пократко процесирачко време по трансформација, користиме 23 посебни CSV датотеки, при што секоја од нив ги содржи лековите од националниот регистар на посебна држава.

CSV датотеките ги испраќаме до BatchRefine [4] инстанца поставена на наш сервер.

BatchRefine е верзија на OpenRefine со додаток за работа со RDF податоци, која дополнително може да се користи и како REST-базиран сервис. Со помош на BASH скрипта испраќаме HTTP POST барања до BatchRefine REST-базираниот сервис, кои содржат (а) CSV датотека која треба да се трансформира и (б) нашата OpenRefine скрипта за трансформација ($C_{3_b drugs}$). Резултатот од секое од овие POST барања е трансформиран RDF излез, кој содржи дел од нашето целно поврзано податочно множество за лекови.

Излезот од нашата BatchRefine трансформација се 23 RDF датотеки во Turtle формат. Овие датотеки го преставуваат нашето поврзано податочно множество: ги содржат високо-квалитетните поврзани податоци за лековите од дваесет и три држави, заедно со линковите кои тие ги имаат кон генерички лекови од LOD облакот. Како што ќе покажеме подолу во текстот, овие линкови можеме да ги искористиме за добивање дополнителни податоци за лековите од нашето множество, податоци кои ги нема во нашето податочно множество ниту во изворните национални регистри на лекови, но кои постојат во други податочно множества на Вебот и кои можат да им бидат од корист на корисниците на податочното множество.

Откако ќе завршат сите трансформации со BatchRefine, ги вчитуваме RDF датотеките во Virtuoso инстанца [66] со помош на BASH скрипта. Сите RDF датотеки се вчитуваат во еден ист RDF граф. Последното извршување на автоматизираниот работен тек (Слика 5.3) резултираше со над 248.000 различни лекови во овој чекор, претставени преку над 7.450.000 RDF тројки и со над 278.000 линкови кон лекови од LOD облакот.

Откако RDF податоците се поставени во RDF графот во Virtuoso, ги користиме SPARQL прашањата за проширување и поврзување на податочното множество ($C_{3_c drugs}$). Прашањата ги извршуваме над нашето податочно множество складирано во Virtuoso инстанцата, со помош на BASH скрипта. Во последното извршување на работниот тек (Слика 5.3), над 38.000 нови ATC кодови беа додадени за лекови за кои овој податок недостасува во изворните податоци. Потоа, над 91.780.000 `schema:relatedDrug` тројки беа додадени во овој чекор, т.е. над 45.890.000 парови од лекови од нашето поврзано податочно множество беа идентификувани како лекови со иста функција, но кои постојат под различно продажно име, или се регистрирани во различни држави, произведени од различни производители, или можеби имаат различна количина во пакувањето, различна јачина, цена, итн. Како што ќе видиме во продолжение, овие поврзувања можеме да ги искористиме во кориснички сценарија во кои на крајните корисници им се обезбедува листа на алтернативни лекови кои одговараат на нивната состојба, а кои се наоѓаат во истата или можеби во друга држава.

Работниот тек прикажан на Слика 5.3 се активира на одреден временски интервал, кој обезбедува гаранција дека секоја промена во националните регистри на лекови ќе биде видлива и во нашето поврзано податочно множество за лекови, во разумно време. Временскиот интервал на кој се повторува работниот тек тековно е поставен на еден месец. Со цел да се справиме со промените на податоците за време на ажурирањето, првин го преместуваме RDF графот кој го содржи нашето податочно множество преку промена на неговиот идентификатор во Virtuoso (го додаваме тековниот датум како

суфикс) и потоа го доделуваме предефинираниот идентификатор на новокреираниот граф. Со ова обезбедуваме верзионирање на податоците, со цел да ги зачуваме претходните верзии на графот и податоците, а истовремено да обезбедиме гаранција дека предефинираниот идентификатор на графот секогаш ќе покажува кон најновите податоци за лековите. Овој тип на верзионирање одговара во нашиот случај, поради тоа што податочното множество не е многу големо споредено со можностите и капацитетот на серверот. Последното извршување на работниот тек резултираше со над 99.000.000 RDF тројки вкупно во LinkedDrugs податочното множество.

Согласно препораките во Чекор 4 од нашата методологија, податочното множество треба да биде објавено и јавно достапно на Вебот. Поради тоа, нашето консолидирано, поврзано податочно множество за лекови од дваесет и три различни држави е објавено согласно најдобрите практики и препораки за објава на поврзани податоци [126], преку перманентно URI кое поддржува HTTP content negotiation, на `http://linkeddata.finki.ukim.mk/lod/data/drugs#` [49]. Податочното множество е поставено на нашата активна Virtuoso инстанца [66], во RDF граф со идентификатор `<http://linkeddata.finki.ukim.mk/lod/data/drugs#>`, кој е јавно достапен и преку SPARQL endpoint [57] кој може да се користи како REST-базиран сервис. Дополнително, податочното множество е објавено и на Datahub [134].

Нови кориснички сценарија врз поврзаното податочно множество

Со автоматизираниот систем и неговиот работен тек започнуваме со дваесет и три различни, дистрибуирани податочни множества, објавени на Веб во формат за прелистување и прегледување од страна на луѓето и со помош на насоките од методологијата и компонентите достапни за повторна употреба успеваме да креираме консолидирано и порамнето податочно множество од меѓусебно поврзани лекови од различни држави, дополнително поврзани и со лекови од LOD облакот. Со цел да ги демонстрираме предностите на користење на податоци за лекови од ваков висок квалитет, а со тоа и да ги имплементираме препораките од Чекор 5 од методологијата, ќе претставиме неколку кориснички и аналитички сценарија преку SPARQL прашања. Со оглед на тоа што нашиот Virtuoso SPARQL endpoint [57] може да се користи како REST-базиран сервис, SPARQL прашањата претставени во овој дел од текстот можат да се искористат од различни типови на апликации кои преку него можат да пристапат до најновите податоци од податочното множество.

Генерални податоци за лекови. Првото, наједноставно корисничко сценарио, би било селекција на сите детали за конкретен лек од податочното множество. Лекот може да биде селектиран преку страна во некоја корисничка апликација која овозможува пребарување и прелистување користејќи го истиот SPARQL endpoint, а потоа преку користење на SPARQL прашање да пристапи до сите детали за селектираниот лек. Ова би им овозможило на крајните корисници да го дознаат продажното име на лекот, генеричкото име, јачината, пакувањето, активните супстанци, дозирањето, производителот, информации за лиценцата, цената, како и информација за тоа кога последен пат биле

ажурирани податоците за лекот во нашиот систем, за релевантност. Ова им нуди на корисниците можност да ги имаат деталите за лекот од интерес постојано при рака, преку нивната апликација. Пример SPARQL прашање кое ги селектира деталите за лекот "Buto-Asma" е дадено во продолжение:

SPARQL прашање 5.1

```
prefix schema: <http://schema.org/>
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>

SELECT ?name ?genericName ?activeIngredient ?atcCode ?gtin ?strengthValue
?strengthUnit ?dosageForm ?amount ?costPerUnit ?costCurrency ?manufacturerName
?licenseID ?licenseValidFrom ?licenseValidUntil ?licenseBy ?licenseFor
?prescriptionStatus ?country ?dateModified
FROM <http://linkeddata.finki.ukim.mk/lod/data/drugs#>
WHERE {
  <https://lekovi.zdravstvo.gov.mk/drugsregister/detailview/56589>
      schema:name ?name ;
      drugbank:genericName ?genericName ;
      schema:activeIngredient ?activeIngredient ;
      drugbank:atcCode ?atcCode ;
      schema:gtin13 ?gtin ;
      schema:dosageForm ?dosageForm ;
      schema:amountOfThisGood ?amount ;
      schema:availableStrength ?strengthEntity ;
      schema:cost ?costEntity ;
      schema:manufacturer ?manufacturerEntity ;
      schema:license ?licenseEntity ;
      schema:prescriptionStatus ?prescriptionStatus ;
      schema:addressCountry ?country ;
      schema:dateModified ?dateModified .
  ?strengthEntity schema:strengthValue ?strengthValue ;
      schema:strengthUnit ?strengthUnit .
  ?costEntity schema:costPerUnit ?costPerUnit ;
      schema:costCurrency ?costCurrency .
  ?manufacturerEntity schema:legalName ?manufacturerName .
  ?licenseEntity schema:licenseID ?licenseID ;
      schema:validFrom ?licenseValidFrom ;
      schema:validUntil ?licenseValidUntil ;
      schema:issuedBy ?byOrganization ;
      schema:issuedFor ?forOrganization .
  ?byOrganization schema:legalName ?licenseBy .
  ?forOrganization schema:legalName ?licenseFor .
}
```

Прашањето може да се генерализира со менување на URI идентификаторот на лекот во линија 10 од прашањето, во URI на лекот за кој бараме детали. Комплетните резултати од извршеното прашање над нашите податоци можат да се погледнат на Веб, во рамки на апликацијата Seminant [54], на следната локација: `http://seminant.com/queries/5803df8d73656d19eb5f5d00`. Парцијален резултат од прашањето е даден во Табела 5.2.

Табела 5.2: Парцијален резултат од Прашање 5.1

Име	BUTO-ASMA
Генеричко име	Salbutamol
Активна супстанца	Salbutamol
ATC код	R03AC02
Баркод	5310201000576
Јачина (вредност)	100
Јачина (единица)	mcg
Цена (вредност)	140.8
Цена (валута)	MKD
Производител	Laboratorio Aldo - Union SA, Барселона, Шпанија
Лиценца	15-229/12
Држава	MK

Податоци базирани на меѓусебната поврзаност на лековите. Меѓусебното поврзување на лековите од податочното множество, кое го реализиравме со генерирањето на `schema:relatedDrug` тројките, може да се искористи во кориснички сценарија кои на крајните корисници им овозможуваат откривање на лекови од истата или од друга држава, лекови кои ги имаат истите терапевтски, фармаколошки и хемиски својства како лекот од негов интерес. Ова е корисно сценарио во ситуации кога лекот од интерес не е достапен или кога корисникот патува во друга држава. Добивањето на информации за лекови кои ги имаат истите својства и нивните цени може да биде корисно за детекција на лек од одредена категорија или со одредена намена кој е најмногу достапен за пациентот, во одредена ситуација. Овие информации можат да се користат од фармацевтите и лекарите, но и од пациентите, за собирање дополнителни информации и одредување на соодветниот третман. Пример SPARQL прашање кое може да се користи за идентификација на лековите од сите држави кои имаат исти терапевтски, фармаколошки и хемиски својства како лекот од интерес, е дадено во продолжение.

SPARQL прашање 5.2

```
prefix schema: <http://schema.org/>
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
```

```
SELECT ?drug ?name ?gtin ?strengthValue ?strengthUnit ?costPerUnit
?costCurrency ?manufacturerName ?prescriptionStatus ?country
FROM <http://linkeddata.finki.ukim.mk/lod/data/drugs#>
WHERE {
  <http://www.legemiddelverket.no//Legemiddelsoek/Sider/Legemiddelvisning.
                          aspx?pakningId=c5a02a30-d471-4ba0-8197-38955f384dd8>
      schema:relatedDrug ?drug .
  OPTIONAL { ?drug schema:name ?name }
  OPTIONAL { ?drug schema:gtin13 ?gtin }
  OPTIONAL { ?drug schema:prescriptionStatus ?prescriptionStatus }
  OPTIONAL { ?drug schema:addressCountry ?country }
  OPTIONAL {
    ?drug schema:cost ?costEntity .
    ?costEntity schema:costPerUnit ?costPerUnit ;
                schema:costCurrency ?costCurrency .
  }
  OPTIONAL {
    ?drug schema:availableStrength ?strengthEntity .
    ?strengthEntity schema:strengthValue ?strengthValue ;
                    schema:strengthUnit ?strengthUnit .
  }
  OPTIONAL {
    ?drug schema:manufacturer ?manufacturerEntity .
    ?manufacturerEntity schema:legalName ?manufacturerName .
  }
}
```

Табела 5.3: Парцијален резултат од Прашање 5.2

Лек	**Производител**	**Држава**
Activent Sr	Medical Union Pharmaceuticals - Egypt	EG
Aerolin 100mcg/dose Inhaler		EG
Aeroline 400 Inhaler		EG
Aerotropa	Pharco B International-egpyt	EG
Agolin	Agog Pharma Ltd	UG
Airomir	iNova Pharmaceuticals (New Zealand)	NZ
Airomir Autohaler	Teva Sweden AB	NO
Airomir Autohaler	iNova Pharmaceuticals (New Zealand)	NZ
Airomir Autohaler 100 microgramos	Teva Pharma S.L.U.	ES

Во прашањето, само мал дел од податоците за поврзаните лекови се селектираат, но во зависност од конкретното корисничко сценарио, можат да се прошират, слично како кај SPARQL прашање 5.1. Прашањето 5.2 може да се модифицира за да го вклучи конкретниот лек од интерес, со менување на URI идентификаторот во линија 8. Во нашиот пример во SPARQL прашање 5.2, го користиме лекот "Airomir" од Норвешка како пример, при што добиваме резултати за над 300 различни лекови од пет држави кои ја имаат истата функција како него. Прашањето и комплетните резултати можат да се погледнат во рамките на Seminant сервисот `http://seminant.com/queries/5803e77573656d19eb6c5d00`. Парцијален резултат од прашањето е даден во Табела 5.3.

Податоци базирани на поврзаноста со LOD облакот. Главната предност од линкови кои поврзуваат различни и дистрибуирани податочни множества е можноста да се поставуваат SPARQL прашања на унија од податоците од една точка, преку постоечката инфраструктура на Вебот, со користење на W3C стандарди, како HTTP, SPARQL и RDF. Со оглед на тоа што имаме `rdfs:seeAlso` линкови помеѓу лековите од нашето податочно множество со соодветни генерички лекови од DrugBank и DBpedia, можеме да ги искористиме за да добиеме дополнителни информации за лековите од нашето податочно множество секогаш кога ги разгледуваме. Овие дополнителни информации би доаѓале директно од податочните множества на DrugBank и DBpedia, а можат да вклучуваат дополнителен опис на лекот, интеракциите на лекот со други лекови, интеракциите на лекот со храна, механизмот на акција на лекот, неговата фармакологија, апсорпција, биотрансформација и токсичност, листата на алтернативни продажни имиња, како и листа на дополнителни веб страни каде што е опишан лекот. Овие податоци не се достапни на изворните веб страни на националните регистри на лекови; овие податоци би биле директно добиени од дистрибуираните податочни множества на DrugBank и DBpedia dataset, со користење на концептот на SPARQL здружување на прашања, односно *сојување* на податоци дистрибуирани преку Вебот [169]. Пример за ваков тип на SPARQL прашање, кое селектира податоци за лекот од интерес од DrugBank и DBpedia, е даден во продолжение.

SPARQL прашање 5.3

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix schema: <http://schema.org/>
prefix drugbank: <http://wifo5-04.informatik.uni-mannheim.de/drugbank/
                                                    resource/drugbank/>
prefix dbo: <http://dbpedia.org/ontology/>
prefix dbp: <http://dbpedia.org/property/>

SELECT ?loddrug ?genericName
(group_concat(distinct ?brandName; separator = ", ") AS ?brandNames)
?comment ?description ?biotransformation ?affectedOrganism ?absorption
?chemicalFormula ?toxicity
(group_concat(distinct ?foodInteraction; separator = " ") AS ?foodInteractions)
(group_concat(distinct concat(?interactingDrugLabel, ': ', ?interactionStatus);
```

```
                                          separator = ". ") AS ?drugInteractions)
(group_concat(distinct ?url; separator = ", ") AS ?urls)
WHERE {
  GRAPH <http://linkeddata.finki.ukim.mk/lod/data/drugs#> {
      <http://www.aemps.gob.es/cima/especialidad.do?metodo=
                                            verPresentaciones&codigo=79539>
              rdfs:seeAlso ?loddrug .
  }
  SERVICE <http://wifo5-04.informatik.uni-mannheim.de/drugbank/sparql> {
      OPTIONAL { ?loddrug drugbank:description ?desc . }
      OPTIONAL { ?loddrug drugbank:genericName ?gname . }
      OPTIONAL { ?loddrug drugbank:brandName ?bname . }
      OPTIONAL { ?loddrug drugbank:biotransformation ?biotransformation . }
      OPTIONAL { ?loddrug drugbank:affectedOrganism ?affectedOrganism . }
      OPTIONAL { ?loddrug drugbank:absorption ?absorption . }
      OPTIONAL { ?loddrug drugbank:chemicalFormula ?chemicalFormula . }
      OPTIONAL { ?loddrug drugbank:foodInteraction ?foodInteraction . }
      OPTIONAL { ?loddrug foaf:page ?page . }
      OPTIONAL { ?loddrug drugbank:toxicity ?toxicity . }
      OPTIONAL {
          ?drugInteractionEntity drugbank:interactionDrug1 ?loddrug ;
                                 drugbank:interactionDrug2 ?interactingDrug ;
                                 drugbank:text ?interactionStatus .
          ?interactingDrug rdfs:label ?interactingDrugLabel .
      }
  }
  SERVICE <http://dbpedia.org/sparql> {
      OPTIONAL { ?loddrug dbo:abstract ?abstract .
                 FILTER (langMatches(lang(?abstract), "en")) }
      OPTIONAL { ?loddrug rdfs:label ?label .
                 FILTER (langMatches(lang(?label), "en")) }
      OPTIONAL { ?loddrug rdfs:comment ?comment .
                 FILTER (langMatches(lang(?comment), "en")) }
      OPTIONAL { ?loddrug dbp:tradename ?tradename . }
      OPTIONAL { ?loddrug dbo:wikiPageExternalLink ?externalLink . }
  }
  BIND(IF(bound(?abstract), ?abstract, ?desc) as ?description)
  BIND(IF(bound(?bname), ?bname, ?tradename) as ?brandName)
  BIND(IF(bound(?gname), ?gname, ?label) as ?genericName)
  BIND(IF(bound(?page), ?page, ?externalLink) as ?url)
}
```

Прашањето селектира дел од важните информации за лекот од интерес и неговите активни супстанци од DrugBank и DBpedia. Најзначајни се хемиските, биолошките

Табела 5.4: Парцијален резултат од Прашање 5.3

Опис (од DBpedia)
Duloxetine (Cymbalta, and generics) is a serotonin-norepinephrine reuptake inhibitor (SNRI) created by Eli Lilly. It is mostly prescribed for major depressive disorder, generalized anxiety disorder, fibromyalgia and neuropathic pain. Duloxetine failed to receive US approval for stress urinary incontinence amid concerns over liver toxicity and suicidal events; however, it was approved for this indication in the UK, where it is recommended as an add-on medication in stress urinary incontinence instead of surgery.
Интеракции со храна
Food does not affect maximum levels reached, but delays it (from 6 to 10 hours) and total product exposure appears to be reduced by only 10 percent. People taking this product who drink large amounts of alcohol are exposed to a higher risk of liver toxicity. Take without regard to meals.
Интеракции со лекови
Amitriptyline: Possible increase in the levels of this agent when used with duloxetine. Ciprofloxacin: Ciprofloxacin increases the effect/toxicity of duloxetine. Desipramine: Possible increase in the levels of this agent when used with duloxetine. Flecainide: Possible increase in the levels of this agent when used with duloxetine. Fluvoxamine: Fluvoxamine increases the effect and toxicity of duloxetine. Imipramine: Possible increase in the levels of this agent when used with duloxetine. Isocarboxazid: Possible severe adverse reaction with this combination. Nortriptyline: Possible increase in the levels of this agent when used with duloxetine. Phenelzine: Possible severe adverse reaction with this combination. Propafenone: Possible increase in the levels of this agent when used with duloxetine. Rasagiline: Possible severe adverse reaction with this combination. Thioridazine: Increased risk of cardiotoxicity and arrhythmias. Tranylcypromine: Possible severe adverse reaction with this combination

и фармаколошките својства на лекот, заедно со интеракциите кои тој ги има со други лекови и со храна. Овие податоци најчесто воопшто не постојат на националните регистри на лекови, но се од големо значење за крајните корисници, особено за фармацевтите и лекарите кои може да ги искористат овие податоци за да одредат третман за нова, акутна состојба на пациент кој е веќе под третман за друга, хронична здравствена состојба.

Едно пример извршување на SPARQL прашањето 5.3, за лекот "Duloxetina" од Шпанија, резултира во детали за генеричкиот лек "Duloxetine" од DrugBank и DBpedia: `http://seminant.com/queries/5803e9b973656d19eba65e00`. Меѓу другото, резултатите прикажуваат и три храна - лек интеракции на лекот, како и 13 лек - лек интеракции кои тој ги има. Парцијален резултат од прашањето е даден во Табела 5.4.

Аналитички сценарија. Покрај овие кориснички сценарија, нашето поврзано податочно множество за лекови може да се искористи и за аналитички сценарија. Анали-

тичките прашања овозможуваат заинтересираните чинители да добијат подобар увид во состојбата на пазарот на лекови во различни држави, овозможувајќи им да ги анализираат постоечките консолидирани податоци преку користење на единствена влезна точка за прашања и еден прашален јазик. Ваквата аналитика може да биде вградена во специфични аналитички апликации, или пак може да се изведува со одвоени и независни SPARQL прашања.

Со цел да добиеме подобар увид во аналитичките можности над консолидираните податоци за лекови од повеќе држави, ќе погледнеме едно аналитичко прашање со кое ќе ги идентификуваме најчестите категории на лекови по држава. Ваквото прашања му овозможува на корисникот, на пример една фармацевтска компанија, да добие подобар увид во националните пазари за лекови и да направи информирана одлука при пласирањето на својот нов лек на нов пазар во одредена држава. Генерално SPARQL прашање за вакво аналитичко сценарио е дадено во продолжение.

SPARQL прашање 5.4

```
prefix schema: <http://schema.org/>
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>

SELECT count (distinct ?drug) as ?count ?atc ?country
FROM <http://linkeddata.finki.ukim.mk/lod/data/drugs#>
WHERE {
      ?drug a schema:Drug ;
          schema:addressCountry ?country ;
          drugbank:atcCode ?atcCode .
      FILTER (strlen(?atcCode) > 3)
      BIND(SUBSTR(xsd:string(?atcCode), 1, 3) AS ?atc)
}
GROUP BY ?country ?atc
ORDER BY DESC (?count)
```

Извршувањето на SPARQL прашање 5.4 покажува дека Романија, Шпанија, Холандија, Ирска и Словачка имаат најмногу лекови во категоријата агенси кои делуваат на ренин-ангиотензин системот (лекови со ATC код кој започнува на C09), Русија и Јужноафриканска Република имаат најмногу лекови во категоријата антибактериски лекови за системска употреба (лекови со ATC код кој започнува на "J01"), додека САД имаат најмногу лекови во категоријата психолептици (лекови со ATC код кој започнува на "N05"). Овие парцијални резултати се прикажани во Табела 5.5. Комплетните резултати од прашањето се достапни на `http://seminant.com/queries/5803ebc473656d19ebac5e00`.

Второ аналитичко сценарио би можело да ја одреди просечната цена на лекови по категорија, по држава. Резултатите би можеле да им бидат од интерес на медицинските власти во одредена држава за да ја одредат состојбата со цените во одредена категорија

Табела 5.5: Парцијален резултат од Прашање 5.4

Лекови	**ATC префикс**	**Држава**
5362	C09	RO
2152	C09	ES
1536	J01	RU
1488	C09	NL
1270	N05	US
976	J01	ZA
758	C09	IE
709	C09	SK
707	N02	NZ

на лекови и да ја споредат состојбата во различни држави, што би можело да им помогне при носење на локална регулатива. Ваквата аналитика би можела да се искористи и од страна на фармацевтска компанија за да го одреди рангот на цената за нов лек од одредена категорија, пред да биде пласиран на пазарот. Пример SPARQL прашање кое може да се искористи за ваква анализа е дадено во продолжение.

SPARQL прашање 5.5

```
prefix schema: <http://schema.org/>
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>

SELECT (?totalCost / ?drugCount as ?averageCost) ?costCurrency ?atc
        ?drugCount ?country
WHERE {
  SELECT count (distinct ?drug) as ?drugCount
         sum (xsd:float(?cost)) as ?totalCost
         ?costCurrency ?atc ?country
  FROM <http://linkeddata.finki.ukim.mk/lod/data/drugs#>
  WHERE {
        ?drug a schema:Drug ;
            schema:addressCountry ?country ;
            drugbank:atcCode ?atcCode ;
            schema:cost ?costEntity .
        ?costEntity schema:costPerUnit ?costPerUnit ;
                schema:costCurrency ?costCurrency .
        FILTER (strlen(?atcCode) > 3)
        BIND(SUBSTR(xsd:string(?atcCode), 1, 3) AS ?atc)
        FILTER (?costPerUnit != "0"^^xsd:double)
        BIND(REPLACE(?costPerUnit, ",", ".") AS ?cost)
```

```
  }
}
GROUP BY ?country ?atc ?costCurrency
ORDER BY DESC (?averageCost)
```

Табела 5.6: Парцијален резултат од Прашање 5.5

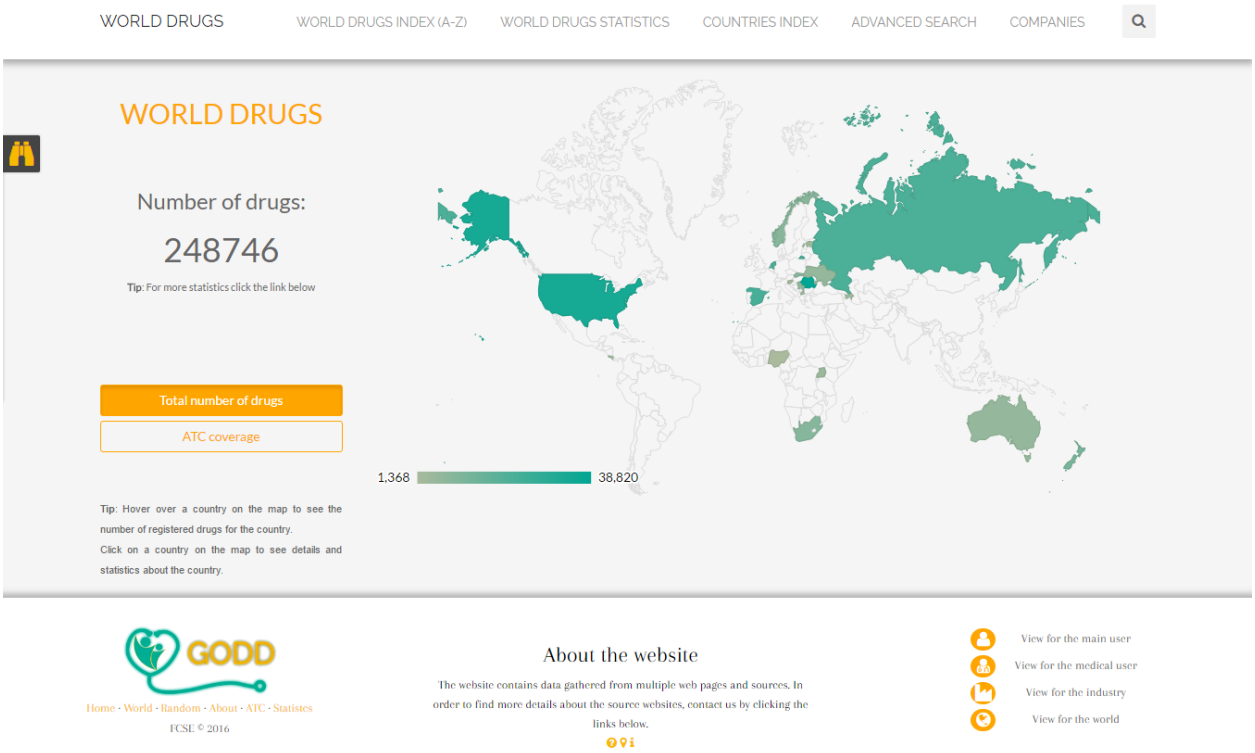
Просечна цена	**Валута**	**ATC префикс**	**Држава**
93480.60	NOK	M09	NO
47221.40	NOK	R07	NO
39557.30	NOK	A16	NO
32021.40	MKD	A16	MK
28837.20	MKD	H01	MK
27478.20	MKD	B02	MK
22500.00	AUD	R07	AU
17822.00	SVN	R07	SI
13360.10	EUR	V10	CY
10679.50	ZAR	LO4	ZA
10127.10	ZAR	B06	ZA
9880.81	ZAR	A16	ZA

Резултатите од извршувањето на SPARQL прашање 5.5 покажуваат дека ATC категориите кои имаат највисока просечна цена во Норвешка се лекови за нарушувања на мускулно-скелетниот систем (лекови со ATC код кој започнува на “M09”), лекови за респираторниот систем (лекови со ATC код кој започнува на “R07”) и лекови за дигестивниот систем и метаболизмот (лекови со ATC код кој започнува на “A16”). Во Македонија тоа се лекови за дигестивниот систем и метаболизмот (лекови со ATC код кој започнува на “A16”), питуитарни и хипоталамусни хормони и аналози (лекови со ATC код кој започнува на “H01”) и антихеморагици (лекови со ATC код кој започнува на “B02”), додека во Австралија и Словенија тоа се лекови за респираторниот систем (лекови со ATC код кој започнува на “R07”) и во Јужноафриканска Република тоа се имуносупресанти (лекови со ATC код кој започнува на “L04”), хематолошки агенси (лекови со ATC код кој започнува на “B06”) и лекови за дигестивниот систем и метаболизмот (лекови со ATC код кој започнува на “A16”). Овие парцијални резултати се дадени во Табела 5.6, додека комплетните резултати кои вклучуваат и други држави се достапни на `http://seminant.com/queries/5803ed6e73656d19eb537e00`.

Во случај кога е потребна споредба на цените помеѓу различни држави, апликацијата која го користи ова податочно множество би можела да искористи конвертор на валути за да ги трансформира вредностите во заедничка валута, со цел да ја изведе споредбата.

5.4 Дискусија и заклучок

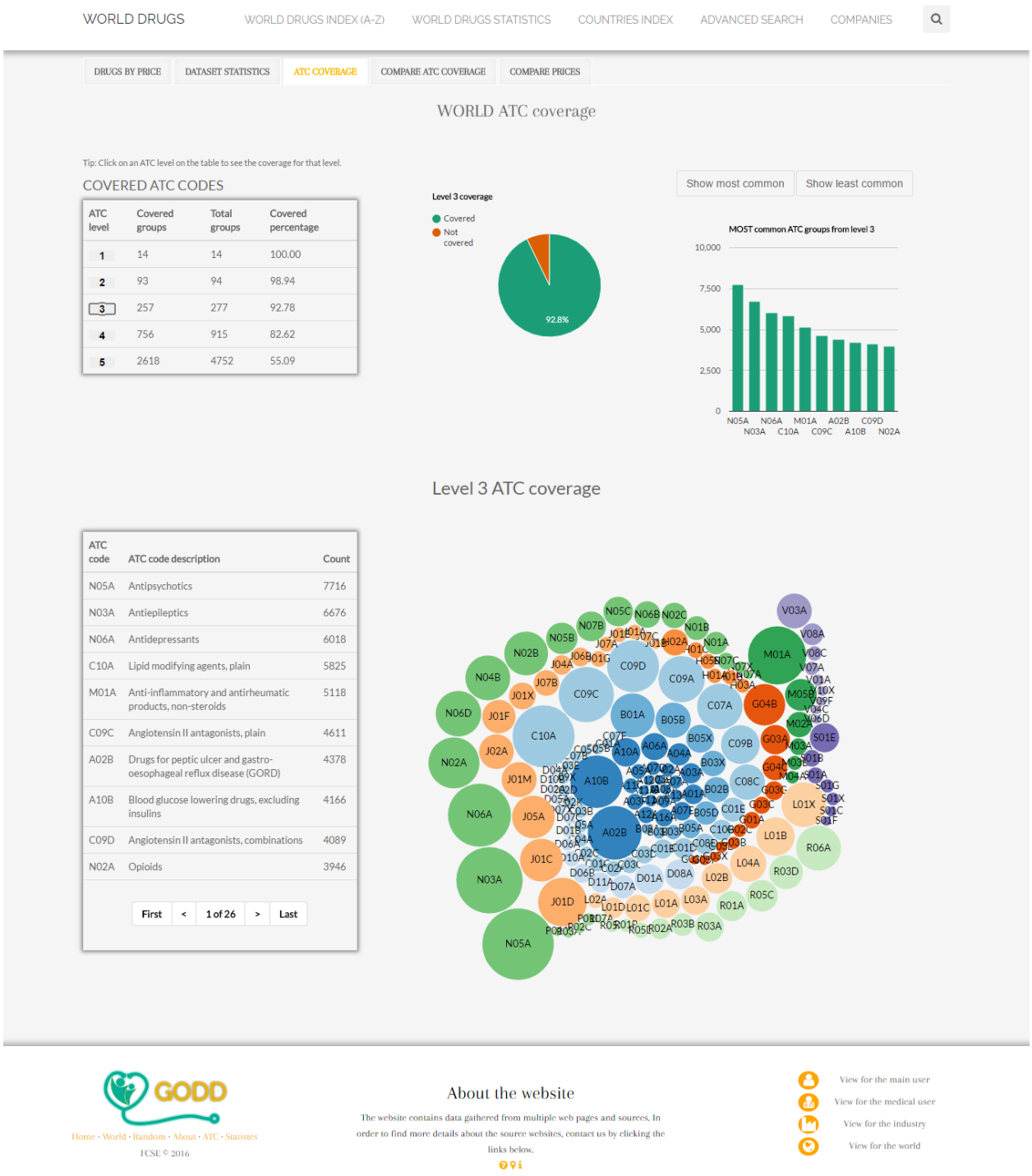
Во ова поглавје ја претставивме нашата нова методологија за поврзани податоци, фокусирана на принципот за повторно искористување на процесот кој го претставува животниот циклус на едно поврзано податочно множество. Попрецизно кажано, нашата методологија се фокусира на повторно искористување на чекорите од животниот циклус за даден домен. Методологијата е резултат на нашата анализа на постоечките методологии за поврзани податоци (Глава 3) и нашето искуство во примена на принципите на поврзани податоци го широк спектар домени (Глава 4).



Слика 5.4: Почетната страна од “Global Open Drug Data (GODD)” веб апликацијата.

Методологијата содржи чекори кои се насочени кон подобро запознавање со доменот од интерес, моделирање и порамнување на податоците, трансформација на податоците во висококвалитетни поврзани податоци, објавување на креираното податочно множество на Веб и дефинирање на кориснички сценарија или развој на апликации и сервиси на податочното множество. Насоките од методологијата имаат за цел да им асистираат на сопствениците и објавувачите на податоци од даден домен во објавување на нивните податоци во истиот порамнет формат, како поврзани податоци. Нивните податоци, откако ќе бидат трансформирани во формат на поврзани податоци и откако ќе бидат поврзани со други податоци објавени со користење на истите компоненти кои можат да се употребуваат повторно, можат да станат дел од нови кориснички ориентирани и аналитички апликации и сервиси.

За да ја валидираме предложената методологија, ја применивме во доменот на ле-



Слика 5.5: Приказ на глобалната покриеност со лекови на трето ниво од ATC класификацијата, во рамките на "Global Open Drug Data (GODD)" веб апликацијата.

кови. Методологијата ја употребивме во рамките на автоматизиран систем за собирање податоци за лекови од официјалните национални регистри со лекови на дваесет и три различни држави, кој ги прочистува податоците, ги порамнува и трансформира во висококвалитетни поврзани податоци и ги објавува на Веб во заедничко, консолидирано податочно множество на поврзани податоци за лекови. Врз база на насоките од методо-

логијата, развивме компоненти од животниот циклус кои можат повторно да се искористат: заедничка шема на податоците, предефиниран формат за податоците, скрипта за трансформација на податоците, SPARQL-базирана алатка за проширување и поврзување на податочното множество и веб базирана алатка за трансформација, поврзување и објавување на податоците. На крајот, демонстриравме множество кориснички-ориентирани и аналитички сценарија над генерираното поврзано податочно множество, кои не се достапни или би одземале неспоредливо повеќе време во ситуација кога корисникот би требало да ги користи само податоците достапни во изворните регистри на лекови на Веб.

За време на пишувањето на оваа дисертација, податочното множество се користи како дел од веб апликацијата Global Open Drug Data (GODD) [23] (Слика 5.4), развиена од група студенти од Факултетот за информатички науки и компјутерско инженерство во Скопје (ФИНКИ). Оваа апликација како податочно ниво го користи единствено нашето поврзано податочно множество со лекови од дваесет и три држави. Врз база на овие податоци, апликацијата нуди широка палета на сервиси за корисниците: преглед на деталите за поединечните лекови, добиени од изворните регистри на лекови, но и од податочни множества од LOD облакот; компаративна анализа на поединечни лекови и на групи лекови од различни држави; анализа на глобалната покриеност со лекови и категории лекови, како и покриеноста по држави; компаративна анализа на апсолутните и просечните цени на лекови и категории лекови во различни држави, итн. (Слика 5.5). Оваа апликација ги демонстрира токму оние предности поради кои го генерираме поврзаното податочно множество со лекови, односно ја демонстрира палетата од бројни кориснички и аналитички сценарија кои може да бидат од голем интерес за пациентите, фармацевтите, лекарите, фармацевтските компании и здравствените власти, на глобално ниво.

Глава 6

Заклучок

Главната цел на оваа дисертација беше развој на генерална методологија за поврзани податоци која целосно го опфаќа животниот циклус на едно поврзано податочно множество во даден домен: идентификација, моделирање, анотација, трансформација, објавување и ефективно искористување, со посебен фокус на повторно искористување на компонентите од самиот животен циклус. Со цел да го постигнеме ова, требаше: (а) да го имплементираме животниот циклус на поврзани податоци во широк спектар домени и (б) да ги анализираме постоечките методологии на поврзани податоци.

Првата истражувачка задача нѝ овозможи практична анализа на методите и техниките кои можат да се искористат во секој од чекорите на животниот тек за поврзано податочно множество во даден домен. Извлековме искуство за различните начини на кои можат да се соберат изворните податоци, различните пристапи во моделирање на податоците, најдобрите практики за искористување и дефинирање на онтологии и вокабулари, различните методи за трансформација на податоци кои можат да вариираат од мануелни до целосно автоматизирани, различните начини на објавување на податочното множество на Вебот, како и големиот број начини на кои податочните множества можат да се искористат во кориснички апликации и сервиси. Голем број од чекорите и одлуките во голема мера зависат од типот или доменот на изворните податоци, но постојат и пристапи кои се под влијание на целта за која се генерира поврзаното податочно множество. Се обидовме да ги идентификуваме сите специфичности и најдобри практики, преку бројни истражувачки проекти во изминатите неколку години, презентирани во Глава 4.

Втората истражувачка задача нѝ овозможи да ги идентификуваме специфичностите, предностите и недостатоците на постоечките методологии за поврзани податоци. Преку анализата ги идентификувавме генералните чекори за животниот циклус на секое поврзано податочно множество, независно од доменот. Дополнително, за секој од чекорите извлековме заклучоци за специфичните и генерални практики кои можат да станат дел од нова, генерална методологија за поврзани податоци. Деталите од нашата анализа се презентирани во Глава 3.

Врз база на овие два дела од истражувањето, предложивме и развивме нова мето-

дологија за поврзани податоци, фокусирана на принципот за повторно искористување на процесот кој го претставува животниот циклус на едно поврзано податочно множество. Поконкретно, нашата методологија се фокусира на повторно искористување на чекорите од животниот циклус за даден домен. Таа содржи чекори кои се насочени кон подобро запознавање со доменот од интерес, моделирање и порамнување на податоците, трансформација на податоците во висококвалитетни поврзани податоци, објавување на креираното податочно множество на Веб и дефинирање на кориснички сценарија или развој на апликации и сервиси на податочното множество. Насоките од методологијата имаат за цел да им асистираат на сопствениците и објавувачите на податоци од даден домен во објавување на нивните податоци во истиот порамнет формат, како поврзани податоци. Нивните податоци, откако ќе бидат трансформирани во формат на поврзани податоци и откако ќе бидат поврзани со други податоци објавени со користење на истите компоненти кои можат да се употребуваат повторно, можат да станат дел од нови кориснички ориентирани и аналитички апликации и сервиси.

Како валидација на предложените методолошки насоки, ги употребивме во доменот на здравство и лекови. Методологијата ја применивме во рамките на автоматизиран систем за собирање податоци за лекови од официјалните национални регистри со лекови на дваесет и три различни држави, кој ги прочистува податоците, ги порамнува и трансформира во висококвалитетни поврзани податоци и ги објавува на Веб во заедничко, порамнето и консолидирано податочно множество на поврзани податоци за лекови. Врз база на насоките од методологијата, развивме компоненти од животниот циклус кои можат повторно да се искористат: заедничка шема на податоците, предефиниран формат за податоците, трансформер за податоците, SPARQL базирана алатка за проширување и поврзување на податочното множество и веб базирана алатка за трансформација, поврзување и објавување на податоците. Потоа демонстрираме множество кориснички-ориентирани и аналитички сценарија над генерираното поврзано податочно множество, кои не се достапни или би одземале неспоредливо повеќе време во ситуација кога корисникот би требало да ги користи само податоците достапни во изворните регистри на лекови на Веб.

Со ова, покажуваме дека методологија која обезбедува насоки за развој на компоненти од животниот циклус кои можат да се искористат повторно, им овозможува на објавувачите на податоци да ја споделат својата експертиза од даден домен, притоа спуштајќи ја границата за идните објавувачи на поврзани податоци да ги развијат и објават своите податочни множества од истиот домен. Методологијата им овозможува пристап до алатки, сервиси и други компоненти на сопствениците и објавувачите на податоци кои не се детално запознаени со концептите на поврзани податоци, кои би можеле да се искористат повторно во рамките на доменот од нивен интерес, со цел да ги генерираат и објават нивните податоци во истиот, порамнет формат на поврзани податоци. Од друга страна, методологијата им пружа механизми и на искусните објавувачи на поврзани податоци да можат на брз и поедноставен начин да се вклучат во домен во кој немаат големо искуство, преку истите компоненти кои можат повторно да

се употребуваат.

Веруваме дека овој комбиниран бенефит на различните чинители во доменот на поврзани податоци ќе води кон зголемување на бројот на висококвалитетни, порамнети поврзани податочни множества објавени како дел од LOD облакот, водејќи со тоа кон поквалитетни апликации и сервиси кои зависат од структурираните податоци достапни на Вебот. Скорешното актуелизирање на интердисциплинарни научни полиња како што е науката базирана на податоци, креирањето и објавувањето на висококвалитетни структурирани податоци станува уште поважно. Тие им овозможуваат на научниците кои работат со податоци да прават анализи над прочистени, структурирани и порамнети податоци, со цел да генерираат ново знаење и нова вредност во даден домен. Оттука, високото ниво на квалитет на објавените поврзани податочни множества ќе води кон добивање подобри аналитички резултати.

Библиографија

[1] AlchemyAPI. http://www.alchemyapi.com/. Accessed: 2016-01-22.

[2] American Bus Association. http://www.buses.org/. Accessed: 2016-01-22.

[3] ATC Codes: Structure and Principles. http://www.whocc.no/atc/structure_and_principles. Accessed: 2016-01-22.

[4] BatchRefine. https://github.com/fusepoolP3/p3-batchrefine. Accessed: 2016-03-23.

[5] BBC Radio. http://www.bbc.co.uk/radio/. Accessed: 2016-01-22.

[6] CKAN: Open-Source Data Portal Platform. http://ckan.org/. Accessed: 2016-01-22.

[7] ConverterToRdf. https://www.w3.org/wiki/ConverterToRdf. Accessed: 2016-01-22.

[8] Creative Commons. https://creativecommons.org/. Accessed: 2016-01-22.

[9] Crime Map of the Republic of Macedonia. http://crimemap.finki.ukim.mk/. Accessed: 2016-01-22.

[10] D2R Server: Accessing databases with SPARQL and as Linked Data. http://d2rq.org/d2r-server. Accessed: 2016-01-22.

[11] Dandelion API. http://dandelion.eu/. Accessed: 2016-01-22.

[12] Datahub Portal. http://datahub.io/. Accessed: 2016-01-22.

[13] DATEX II Specification. http://www.datex2.eu/. Accessed: 2016-01-22.

[14] DBpedia Mobile. http://wiki.dbpedia.org/projects/dbpedia-mobile. Accessed: 2016-01-22.

[15] DBpedia Spotlight. http://spotlight.dbpedia.org/. Accessed: 2016-01-22.

[16] DERI Vocabularies. http://vocab.deri.ie/. Accessed: 2016-01-22.

[17] Drupal: Open Source CMS. http://drupal.org/. Accessed: 2016-01-22.

[18] European Environment Agency. http://www.eea.europa.eu/. Accessed: 2016-01-22.

[19] Financial Report Ontology. http://financialreportontology.wikispaces.com/. Accessed: 2016-01-22.

[20] Food Ontology. http://data.lirmm.fr/ontologies/food. Accessed: 2016-02-12.

[21] Fusepool P3 BatchRefine Transformer. https://fusepoolp3.github.io/batch-refine/. Accessed: 2016-03-23.

[22] General Transit Feed Specification (GTFS). https://developers.google.com/transit/gtfs/. Accessed: 2016-01-22.

[23] Global Open Drug Data (GODD). http://godd.finki.ukim.mk/. Accessed: 2016-07-20.

[24] Government Linked Data (GLD) Working Group. https://www.w3.org/2011/gld/wiki/Main_Page. Accessed: 2016-01-22.

[25] Health Insurance Fund of Macedonia. http://www.fzo.org.mk/. Accessed: 2016-01-22.

[26] ISO 3166-1 alpha-3 Standard. https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3. Accessed: 2016-03-23.

[27] ISO 4217 Standard. https://en.wikipedia.org/wiki/ISO_4217. Accessed: 2016-03-23.

[28] JSP Skopje. http://www.jsp.com.mk/. Accessed: 2016-01-22.

[29] Linked Data: Connect Distributed Data across the Web. http://linkeddata.org/. Accessed: 2016-01-22.

[30] Linked Open Data (LOD) Cloud. http://lod-cloud.net/. Accessed: 2016-01-22.

[31] Linked Open Data (LOD) Cloud cache instance. http://lod.openlinksw.com/. Accessed: 2016-01-22.

[32] Linked Open Data (LOD) Cloud: How To Join. http://lod-cloud.net/#how-to-join. Accessed: 2016-01-22.

[33] Linked Open Vocabularies (LOV). http://lov.okfn.org/. Accessed: 2016-01-22.

[34] LinkedBrainz: A Project to Provide MusicBrainz NGS as Linked Data. http://linkedbrainz.c4dmpresents.org/. Accessed: 2016-01-22.

[35] Linking Open Drug Data (LODD). https://www.w3.org/wiki/HCLSIG/LODD. Accessed: 2016-06-07.

[36] LOD2 Project. http://lod2.eu/. Accessed: 2016-03-23.

[37] LODRefine. https://github.com/sparkica/LODRefine/. Accessed: 2016-01-22.

[38] Macedonian Drug Registry. https://lekovi.zdravstvo.gov.mk/drugsregister/overview. Accessed: 2016-03-23.

[39] Macedonian Stock Exchange. http://www.mse.mk/en/. Accessed: 2016-01-22.

[40] Ministry of Interior, Republic of Macedonia. http://www.mvr.gov.mk/. Accessed: 2016-01-22.

[41] MusicBrainz: The Open Music Encyclopedia. https://musicbrainz.org/. Accessed: 2016-01-22.

[42] Open Calais. http://www.opencalais.com/. Accessed: 2016-01-22.

[43] Open Corporates. https://opencorporates.com/. Accessed: 2016-01-22.

[44] Open Data Commons. http://opendatacommons.org/. Accessed: 2016-01-22.

[45] Open Government Licence v3.0. https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/. Accessed: 2016-01-22.

[46] Open Government Partnership. http://www.opengovpartnership.org/. Accessed: 2016-01-22.

[47] Open Knowledge Ireland. http://opendata.ie/. Accessed: 2016-01-22.

[48] OpenRefine. http://openrefine.org/. Accessed: 2016-01-22.

[49] Permanent URI of the LinkedDrug Dataset. http://linkeddata.finki.ukim.mk/lod/data/drugs#. Accessed: 2016-04-12.

[50] RDF:Alerts. http://swse.deri.org/RDFAlerts/. Accessed: 2016-01-22.

[51] Schema.org in RDFS. http://schema.rdfs.org/. Accessed: 2016-01-22.

[52] Schema.org Vocabulary. http://schema.org/. Accessed: 2016-01-22.

[53] Semantic Web Health Care and Life Sciences Interest Group. https://www.w3.org/blog/hcls/. Accessed: 2016-01-22.

[54] Seminant: SPARQL Execution and Sharing. http://seminant.com/. Accessed: 2016-03-23.

[55] Silk Framework. http://silkframework.org/. Accessed: 2016-01-22.

[56] Sindice: Data Web Services. http://sindice.com/. Accessed: 2016-01-22.

[57] SPARQL Endpoint at the Faculty of Computer Science and Engineering in Skopje. http://linkeddata.finki.ukim.mk/sparql. Accessed: 2016-01-22.

[58] Swedish Transport Administration Data as Linked Data. http://sta.linkeddata.finki.ukim.mk/. Accessed: 2016-01-22.

[59] Swoogle: Semantic Web Search. http://swoogle.umbc.edu/. Accessed: 2016-01-22.

[60] Talis Aspire. https://talis.com/digitised-content/. Accessed: 2016-01-22.

[61] The 'Drug' class, from the Schema.org Vocabulary. http://schema.org/Drug. Accessed: 2016-01-22.

[62] The DrugBank RDF Dataset. http://wifo5-03.informatik.uni-mannheim.de/drugbank/. Accessed: 2016-06-07.

[63] The Music Ontology. http://musicontology.com/. Accessed: 2016-01-22.

[64] Trafikverket: Swedish Transport Administration. http://www.trafikverket.se/. Accessed: 2016-01-22.

[65] US Global Foreign Aid, 1946-2009. https://data-gov.tw.rpi.edu//demo/USForeignAid/demo-1554.html. Accessed: 2016-01-22.

[66] Virtuoso Instance at the Faculty of Computer Science and Engineering in Skopje. http://linkeddata.finki.ukim.mk/. Accessed: 2016-01-22.

[67] Virtuoso Sponger. http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger. Accessed: 2016-01-22.

[68] Virtuoso Universal Server. http://virtuoso.openlinksw.com/. Accessed: 2016-01-22.

[69] W3C: Hash vs. Slash. https://www.w3.org/wiki/HashVsSlash. Accessed: 2016-01-22.

[70] WordPress: Blog Tool, Publishing Platform and CMS. http://wordpress.org/. Accessed: 2016-01-22.

[71] World Bank Group Finances: Financial Datasets. https://finances.worldbank.org/all-datasets. Accessed: 2016-01-22.

[72] World Bank IBRD Loans and IDA Credits. http://data.worldbank.org/indicator/DT.DOD.MWBG.CD. Accessed: 2016-01-22.

[73] World Bank SPARQLer: General Purpose Processor. http://worldbank.270a.info/sparql. Accessed: 2016-01-22.

[74] Ben Adida, Mark Birbeck, Shane McCarron, and Ivan Herman. RDFa Core 1.1. http://www.w3.org/TR/rdfa-core/, March 2015. Accessed: 2016-01-22.

[75] Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. Flavor Network and the Principles of Food Pairing. *Scientific Reports*, 1, 2011.

[76] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing Linked Datasets with the VoID Vocabulary. https://www.w3.org/TR/void/, March 2011. Accessed: 2015-08-15.

[77] Keith Alexander and Michael Hausenblas. Describing Linked Datasets - On the Design and Usage of VoID, the Vocabulary of Interlinked Datasets. In *Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09)*. Citeseer, 2009.

[78] Grigoris Antoniou and Frank Van Harmelen. *A Semantic Web Primer*. MIT Press, 2004.

[79] Aaron Antrim, Sean J Barbeau, et al. The Many Uses of GTFS Data - Opening the Door to Transit and Multimodal Applications. *Location-Aware Information Systems Laboratory at the University of South Florida*, 2013.

[80] Marcelo Arenas, Alexandre Bertails, Eric Prud'hommeaux, and Juan Sequeda. A Direct Mapping of Relational Data to RDF. `https://www.w3.org/TR/rdb-direct-mapping/`, September 2012. Accessed: 2016-03-23.

[81] Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N Mendes, Bert Van Nuffelen, et al. Managing the Life-Cycle of Linked Data with the LOD2 Stack. In *The Semantic Web–ISWC 2012*, pages 1–16. Springer, 2012.

[82] L Barandovski, M Cekova, MV Frontasyeva, SS Pavlov, T Stafilov, E Steinnes, and V Urumov. Air Pollution Studies in Macedonia Using the Moss Biomonitoring Technique, NAA, AAS and GIS Technology. *Preprint E18-2006-160, Joint Institute for Nuclear Research, Dubna*, 2006.

[83] Cosmin Basca, Stéphane Corlosquet, Richard Cyganiak, Sergio Fernández, and Thomas Schandl. Neologism: Easy Vocabulary Publishing. In *Proceedings of the Workshop on Scripting for the Semantic Web, in conjunction with ESWC 2008*, 2008.

[84] Tim Berners-Lee. 5-star Open Data. `http://5stardata.info/`.

[85] Tim Berners-Lee. Linked Data. `https://www.w3.org/DesignIssues/LinkedData.html`, June 2009. Accessed: 2016-01-22.

[86] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

[87] Tim Berners-Lee and Nigel Shadbolt. There's Gold to Be Mined From All Our Data. *The Times, London*, 2011.

[88] Diego Berrueta, Jon Phipps, Alistair Miles, Thomas Baker, and Ralph Swick. Best Practice Recipes for Publishing RDF Vocabularies. `http://www.w3.org/TR/swbp-vocab-pub/`, August 2008. Accessed: 2015-08-15.

[89] Ted J Biggerstaff. The Library Scaling Problem and the Limits of Concrete Component Reuse. In *Software Reuse: Advances in Software Reusability, 1994. Proceedings., Third International Conference on*, pages 102–109. IEEE, 1994.

[90] Jennifer Billing and Paul W Sherman. Antimicrobial Functions of Spices: Why Some Like It Hot. *Quarterly Review of Biology*, pages 3–49, 1998.

[91] Chris Bizer, Richard Cyganiak, Tom Heath, et al. How to Publish Linked Data on the Web. `http://linkeddata.org/docs/how-to-publish`, 2007. Accessed: 2016-01-22.

[92] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story so Far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[93] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked Data on the Web (LDOW2008). In *Proceedings of the 17th International Conference on World Wide Web*, pages 1265–1266. ACM, 2008.

[94] Francesca Borrelli, Raffaele Capasso, and Angelo A Izzo. Garlic (Allium Sativum L.): Adverse Effects and Drug Interactions in Humans. *Molecular Nutrition & Food Research*, 51(11):1386–1397, 2007.

[95] MJ Bradley. *Comparison of Energy Use & CO_2 Emissions from Different Transportation Modes*. American Bus Association Foundation, 2007.

[96] Dan Brickley. Basic Geo Vocabulary. `http://www.w3.org/2003/01/geo/`. Accessed: 2016-06-07.

[97] Dan Brickley, Ramanathan V. Guha, and Brian McBride. RDF Schema 1.1. `https://www.w3.org/TR/rdf-schema/`, February 2014. Accessed: 2016-01-22.

[98] Rabia Bushra, Nousheen Aslam, and Arshad Yar Khan. Food-Drug Interactions. *Oman Med J*, 26(2):77–83, 2011.

[99] Maire Byrne Evans, Kieron O'Hara, Thanassis Tiropanis, and Craig Webber. Crime Applications and Social Machines: Crowdsourcing Sensitive Data. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 891–896. International World Wide Web Conferences Steering Committee, 2013.

[100] Kei-Hoi Cheung, Eric Prud'hommeaux, Yimin Wang, and Susie Stephens. Semantic Web for Health Care and Life Sciences: A Review of the State of the Art. *Briefings in Bioinformatics*, 10(2):111–113, 2009.

[101] Mayo Clinic. Nearly 7 in 10 Americans Are on Prescription Drugs. `https://www.sciencedaily.com/releases/2013/06/130619132352.htm`, June 2013. Accessed: 2016-01-22.

[102] National Research Council. Medication Errors Injure 1.5 Million People and Cost Billions of Dollars Annually. `http://www8.nationalacademies.org/onpinews/newsitem.aspx?RecordID=11623`, July 2006. Accessed: 2016-01-22.

[103] Richard Cyganiak, Christian Bizer, Jörg Garbers, Oliver Maresch, and Christian Becker. The D2RQ Mapping Language. `http://d2rq.org/d2rq-language`, March 2012. Accessed: 2016-01-22.

[104] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. `http://www.w3.org/TR/rdf11-concepts/`, February 2014. Accessed: 2016-01-22.

[105] A Dahan and H Altman. Food–Drug Interaction: Grapefruit Juice Augments Drug Bioavailability—Mechanism, Extent and Relevance. *European Journal of Clinical Nutrition*, 58(1):1–9, 2004.

[106] Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF Mapping Language. `https://www.w3.org/TR/r2rml/`, September 2012. Accessed: 2016-03-23.

[107] Paul Davidson. Designing URI Sets for the UK Public Sector. *UK Chief Technology Officer Council*, 2009.

[108] Tim Davies. Open Data Barometer: 2013 Global Report. *World Wide Web Foundation and Open Data Institute*, 2013.

[109] Ian Davis. TRANSIT: A Vocabulary for Describing Transit Systems and Routes. `http://vocab.org/transit/`. Accessed: 2016-01-22.

[110] Victor Epitropou, Lasse Johansson, Kostas D Karatzas, Anastasios Bassoukos, Ari Karppinen, Jaakko Kukkonen, and Mervi Haakana. Fusion of Environmental Information for the Delivery of Orchestrated Services for the Atmospheric Environment in the PESCaDO Project. In *Proceedings of the International Congress on Environmental Modelling and Software Managing Resources of a Limited Planet, Leipzig, Germany*, 2012.

[111] Orri Erling and Ivan Mikhailov. RDF Support in the Virtuoso DBMS. In *Networked Knowledge-Networked Media*, pages 7–24. Springer, 2009.

[112] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. HTTP Content Negotiation. `https://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html`, 1999. Accessed: 2016-01-22.

[113] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Hypertext Transfer Protocol–HTTP/1.1. Technical report, Internet Engineering Task Force, 1999.

[114] John E Gaffney Jr and RD Cruickshank. A General Economics Model of Software Reuse. In *Proceedings of the 14th International Conference on Software Engineering*, pages 327–337. ACM, 1992.

[115] Dieter Genser. Food and Drug Interaction: Consequences for the Nutrition/Health Status. *Annals of Nutrition and Metabolism*, 52(Suppl. 1):29–32, 2008.

[116] W3C OWL Working Group. OWL 2 Web Ontology Language. `https://www.w3.org/TR/owl-overview/`, December 2012. Accessed: 2016-01-22.

[117] W3C SPARQL Working group. SPARQL 1.1 Overview. `https://www.w3.org/TR/sparql11-overview/`, March 2013. Accessed: 2016-01-22.

[118] Ramanathan V. Guha. Introducing Schema.org: Search Engines Come Together for a Richer Web. `https://googleblog.blogspot.com/2011/06/introducing-schemaorg-search-engines.html`, June 2011. Accessed: 2016-01-22.

[119] Ramanathan V. Guha, Dan Brickley, and Steve Macbeth. Schema.org: Evolution of Structured Data on the Web. *Communications of the ACM*, 59(2):44–51, 2016.

[120] Michael Hausenblas. Linked Data Life Cycles. `http://www.slideshare.net/mediasemanticweb/linked-data-life-cycles`, July 2011. Accessed: 2015-08-15.

[121] Patrick J. Hayes and Peter F. Patel-Schneider. RDF 1.1 Semantics. `https://www.w3.org/TR/rdf11-mt/`, February 2014. Accessed: 2016-01-22.

[122] Tom Heath and Christian Bizer. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, 2011.

[123] Ian Hickson. HTML Microdata. https://www.w3.org/TR/microdata/, October 2013. Accessed: 2016-01-22.

[124] Robert Hoehndorf, Dietrich Rebholz-Schuhmann, Melissa Haendel, and Robert Stevens. Thematic Series on Biomedical Ontologies in JBMS: Challenges and New Directions. *Journal of Biomedical Semantics*, 5:15, 2014.

[125] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the Pedantic Web. In *Linked Data on the Web Workshop (LDOW2010) at WWW 2010*, 2010.

[126] Bernadette Hyland, Ghislain Atemezing, and Boris Villazón-Terrazas. Best Practices for Publishing Linked Data. http://www.w3.org/TR/ld-bp/, January 2014. Accessed: 2015-08-15.

[127] Bernadette Hyland and David Wood. The Joy of Data: A Cookbook for Publishing Linked Government Data on the Web. In *Linking Government Data*, pages 3–26. Springer, 2011.

[128] Renato Iannella and James McKinney. vCard Ontology - for describing People and Organizations. https://www.w3.org/TR/vcard-rdf/, May 2014. Accessed: 2016-06-07.

[129] Yuval Itan, Bryony L Jones, Catherine JE Ingram, Dallas M Swallow, and Mark G Thomas. A Worldwide Correlation of Lactase Persistence Phenotype and Genotypes. *BMC Evolutionary Biology*, 10(1):1, 2010.

[130] Milos Jovanovik. Drug Data from the Health Insurance Fund of Macedonia. https://datahub.io/dataset/drug-data-hifm, July 2013. Accessed: 2016-01-22.

[131] Milos Jovanovik. Drug Dataset. https://datahub.io/dataset/drug-dataset, April 2015. Accessed: 2016-01-22.

[132] Milos Jovanovik. Recipe Dataset. https://datahub.io/dataset/recipe-dataset, April 2015. Accessed: 2016-01-22.

[133] Milos Jovanovik. Visualizations of Cuisine-Drug Interactions. http://viz.linkeddata.finki.ukim.mk/, January 2015. Accessed: 2016-01-22.

[134] Milos Jovanovik. The LinkedDrugs Project on Datahub. https://datahub.io/dataset/linked-drugs, April 2016. Accessed: 2016-04-12.

[135] Milos Jovanovik. The LinkedDrugs Project on GitHub. https://github.com/etnc/linked-drugs, April 2016. Accessed: 2016-04-12.

[136] Milos Jovanovik. The LinkedDrugs Project Website. http://drugs.linkeddata.finki.ukim.mk/, April 2016. Accessed: 2016-04-12.

[137] Milos Jovanovik, Aleksandra Bogojeska, Dimitar Trajanov, and Ljupco Kocarev. Inferring Cuisine-Drug Interactions Using the Linked Data Approach. *Scientific Reports*, 5, 2015.

[138] Milos Jovanovik, Marjan Georgiev, and Dimitar Trajanov. Towards Consolidating Brand-Name Drug Data Using Linked Data: The Case Study of the Macedonian Drug Bureau. Submitted for publication.

[139] Milos Jovanovik, Bojan Najdenov, Gjorgji Strezoski, and Dimitar Trajanov. Linked Open Data for Medical Institutions and Drug Availability Lists in Macedonia. In *New Trends in Database and Information Systems II*, Advances in Intelligent Systems and Computing, pages 245–256. Springer International Publishing, 2015.

[140] Milos Jovanovik, Bojan Najdenov, and Dimitar Trajanov. Linked Open Drug Data from the Health Insurance Fund of Macedonia. In *Proceedings of the 10th Conference for Informatics and Information Technology*, pages 56–61. Faculty of Computer Science and Engineering, Skopje, 2013.

[141] Milos Jovanovik, Matej Petrov, Bojan Najdenov, and Dimitar Trajanov. Linked Music Data from Global Music Charts. In *Proceedings of the 10th International Conference on Semantic Systems (SEMANTiCS 2014)*, pages 108–115. ACM, 2014.

[142] Milos Jovanovik, Petar Ristoski, and Dimitar Trajanov. A System for Suggestion and Execution of Semantically Annotated Actions Based on Service Composition. In *ICT Innovations 2013*, Advances in Intelligent Systems and Computing, pages 97–109. Springer International Publishing, 2014.

[143] Aleksandar Kareski, Milos Jovanovik, and Dimitar Trajanov. Desktop Gateway: Semantic Desktop Integration with Cloud Services. In *Proceedings of the Sixth Balkan Conference in Informatics*, pages 162–168, 2013.

[144] Nick Kizoom Kizoom and P Miller. A Transmodel Based XML Schema for the Google Transit Feed Specification with a GTFS / Transmodel Comparison. *Kizoom Ltd., London*, 2008.

[145] Eduard Klein, Stephan Haller, Adrian Gschwend, and Milos Jovanovik. Sustainable Linked Open Data Creation: An Experience Report. In *Electronic Government and Electronic Participation*, volume 23 of *Innovation and the Public Sector*, pages 99–110. IOS Press, 2016.

[146] Julia Kollewe. The World's 10 Best-Selling Prescription Drugs. `https://www.theguardian.com/business/table/2014/mar/27/world-best-selling-prescription-drugs-pharmaceuticals-industry`, March 2014. Accessed: 2016-01-22.

[147] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-Driven Evaluation of Linked Data Quality. In *Proceedings of the 23rd international conference on World Wide Web*, pages 747–758. International World Wide Web Conferences Steering Committee, 2014.

[148] Martin Kostovski, Milos Jovanovik, and Dimitar Trajanov. Open Data Portal based on Semantic Web Technologies. In *Proceeding from the 7th South East European Doctoral Student Conference*, 2012.

[149] Charles W Krueger. Software Reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183, 1992.

[150] Vivek Kundra. *Digital Fuel of the 21st Century: Innovation through Open Data and the Network Effect*. Joan Shorenstein Center on the Press, Politics and Public Policy, 2012.

[151] Vivian Law, Craig Knox, Yannick Djoumbou, Tim Jewison, An Chi Guo, Yifeng Liu, Adam Maciejewski, David Arndt, Michael Wilson, Vanessa Neveu, et al. DrugBank 4.0: Shedding New Light on Drug Metabolism. *Nucleic Acids Research*, 42(D1):D1091–D1097, 2014.

[152] Steve Macbeth. Introducing Schema.org: Bing, Google and Yahoo Unite to Build the Web of Objects. `https://blogs.bing.com/search/2011/06/02/introducing-schema-org-bing-google-and-yahoo-unite-to-build-the-web-of-objects`, June 2011. Accessed: 2016-01-22.

[153] Josip Maras, Maja Štula, and Ivica Crnković. Towards Specifying Pragmatic Software Reuse. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*, page 54. ACM, 2015.

[154] M Douglas McIlroy, JM Buxton, Peter Naur, and Brian Randell. Mass-Produced Software Components. In *Proceedings of the 1st International Conference on Software Engineering, Garmisch Pattenkirchen, Germany*, pages 88–98. sn, 1968.

[155] Robert Meusel, Christian Bizer, and Heiko Paulheim. A Web-Scale Study of the Adoption and Evolution of the Schema.org Vocabulary over Time. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, page 15. ACM, 2015.

[156] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. `https://www.w3.org/TR/skos-reference/`, August 2009. Accessed: 2016-01-22.

[157] Kostadin Mishev, Angjel Kjosevski, Nikola Kalemdzhievski, Nikola Koteli, Milos Jovanovik, Kosta Mitreski, and Dimitar Trajanov. Publishing Skopje Air Quality Data as Linked Data. In *Proceedings of the 12th Conference for Informatics and Information Technology*, 2015.

[158] Elena Mishevska, Bojan Najdenov, Milos Jovanovik, and Dimitar Trajanov. Open Public Transport Data in Macedonia. In *Proceesings of the 11th Conference for Informatics and Information Technology*, 2014.

[159] Martin Mitrevski, Milos Jovanovik, Riste Stojanov, and Dimitar Trajanov. Open University Data. In *Proceedings of the 9th Conference for Informatics and Information Technology*, 2012.

[160] Elena Montiel-Ponsoda, Daniel Vila-Suero, Boris Villazón-Terrazas, Gordon Dunsire, Elena Escolano Rodríguez, and Asunción Gómez-Pérez. Style Guidelines for Naming and Labeling Ontologies in the Multilingual Web. In *Proceedings of the 2011 International Conference on Dublin Core and Metadata Applications*, DCMI'11, pages 105–115. Dublin Core Metadata Initiative, 2011.

[161] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, et al. The Open Provenance Model Core Specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

[162] Bojan Najdenov, Milos Jovanovik, and Dimitar Trajanov. VEO: an Ontology for CO_2 Emissions from Vehicles. *ICT Innovations 2014, Web Proceedings*, pages 269–278, 2014.

[163] Bojan Najdenov, Hristijan Pejchinoski, Kristina Cieva, Milos Jovanovik, and Dimitar Trajanov. Open Financial Data from the Macedonian Stock Exchange. In *ICT Innovations 2014*, pages 115–124. Springer International Publishing, 2015.

[164] Bojan Najdenov, Goran Petkovski, Milos Jovanovik, Riste Stojanov, and Dimitar Trajanov. Automated Linked Data Generation from the Transport Administration Domain. In *23rd Telecommunications Forum (TELFOR), 2015*, pages 827–830, 2015.

[165] Axel-Cyrille Ngonga Ngomo and Sören Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. *Integration*, 15:3, 2011.

[166] Intergovernmental Panel on Climate Change. *2006 IPCC Guidelines for National Greenhouse Gas Inventories*. Intergovernmental Panel on Climate Change, 2006.

[167] World Health Organization et al. *Anatomical Therapeutic Chemical (ATC) Classification With Defined Daily Doses (DDDs)*. WHO, Collaborating Centre for Drug Statistics Methodology, 2001.

[168] James Penman. *Good Practice Guidance and Uncertainty Management in National Greenhouse Gas Inventories*. Institute for Global Environmental Strategies (IGES) for the Intergovernmental Panel on Climate Change, 2000.

[169] Eric Prud'hommeaux, Carlos Buil-Aranda, Andy Seaborne, Axel Polleres, Lee Feigenbaum, and Gregory Todd Williams. SPARQL 1.1 Federated Query. `https://www.w3.org/TR/sparql11-federated-query/`, March 2013. Accessed: 2016-01-22.

[170] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. The Music Ontology. In *ISMIR*, pages 417–422. Citeseer, 2007.

[171] José Luis Redondo-García, Vicente Botón-Fernández, and Adolfo Lozano-Tello. Linked Data Methodologies for Managing Information about Television Content. *International Journal of Interactive Multimedia and Artificial Intelligence*, 1(6), 2012.

[172] Richard S Rivlin. Historical Perspective on the Use of Garlic. *The Journal of Nutrition*, 131(3):951S–954S, 2001.

[173] Marco Rospocher. An Ontology for Personalized Environmental Decision Support. In *FOIS*, pages 421–426, 2014.

[174] Anisa Rula and Amrapali Zaveri. Methodology for Assessment of Linked Data Quality. In *Proceedings of the 1st Workshop on Linked Data Quality, co-located with 10th International Conference on Semantic Systems (SEMANTiCS 2014)*, 2014.

[175] Matthias Samwald, Anja Jentzsch, Christopher Bouton, Claus S Kallesøe, Egon Willighagen, Janos Hajagos, M Scott Marshall, Eric Prud'hommeaux, Oktie Hassanzadeh, Elgar Pichler, et al. Linked Open Drug Data for Pharmaceutical Research and Development. *Journal of heminformatics*, 3(1):19, 2011.

[176] Leo Sauermann, Richard Cyganiak, Danny Ayers, and Max Völkel. Cool URIs for the Semantic Web. `https://www.w3.org/TR/cooluris/`, December 2008. Accessed: 2015-08-15.

[177] Leo Sauermann, Richard Cyganiak, and Max Vökel. Cool uris for the semantic web. Technical report, Saarländische Universitäts- und Landesbibliothek, Postfach 151141, 66041 Saarbrücken, 2007.

[178] François Scharffe, Ondřej Zamazal, and Dieter Fensel. Ontology Alignment Design Patterns. *Knowledge and Information Systems*, 40(1):1–28, 2014.

[179] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.

[180] Lars E Schmidt and Kim Dalhoff. Food-Drug Interactions. *Drugs*, 62(10):1481–1502, 2002.

[181] Shashi Seth. Introducing Schema.org: A Collaboration on Structured Data. `http://www.ysearchblog.com/2011/06/02/introducing-schema-org-a-collaboration-on-structured-data/`, June 2011. Accessed: 2016-01-22.

[182] William Smith, Alan Chappell, and Courtney Corley. Medical and Transmission Vector Vocabulary Alignment with Schema.org. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2015.

[183] Mirko Spasić, Milos Jovanovik, and Arnau Prat-Pérez. An RDF Dataset Generator for the Social Network Benchmark with Real-World Coherence. In *Proceedings of the Workshop on Benchmarking Linked Data (BLINK), 15th International Semantic Web Conference (ISWC)*, 2016.

[184] Riste Stojanov, Marjan Georgiev, Vladimir Zdraveski, Milos Jovanovik, and Dimitar Trajanov. Live Objects - Collaborative Window in the Corporate Documents. In *New Trends in Database and Information Systems II*, Advances in Intelligent Systems and Computing, pages 71–81. Springer International Publishing, 2015.

[185] Mari Carmen Suárez-Figueroa. *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. PhD thesis, Informatica, 2010.

[186] Mari Carmen Suárez-Figueroa and Asunción Gómez-Pérez. NeOn Methodology for Building Ontology Networks: A Scenario-Based Methodology. In *Proceedings of the International Conference on Software, Services & Semantic Technologies*. Sofia, 2009.

[187] Gabriel Svennerberg. *Beginning Google Maps API 3*. Apress, 2010.

[188] Dallas M Swallow. Genetics of Lactase Persistence and Lactose Intolerance. *Annual Review of Genetics*, 37(1):197–219, 2003.

[189] Ellen Tattelman. Health Effects of Garlic. *Am Fam Physician*, 72(1):103–106, 2005.

[190] Damjan Temelkovski, Milos Jovanovik, Igor Mishkovski, and Dimitar Trajanov. Towards Open Data in Macedonia: Crime Map Based on Ministry of Internal Affairs' Bulletins. In *Proceedings of the 9th Conference for Informatics and Information Technology*, 2012.

[191] Will Tracz. Where Does Reuse Start? *ACM SIGSOFT Software Engineering Notes*, 15(2):42–46, 1990.

[192] Dimitar Trajanov, Riste Stojanov, Milos Jovanovik, Vladimir Zdraveski, Petar Ristoski, Marjan Georgiev, and Sonja Filiposka. Semantic Sky: A Platform for Cloud Service Integration Based on Semantic Web Technologies. In *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS 2012)*, pages 109–116. ACM, 2012.

[193] Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked Open Vocabularies (LOV): A Gateway to Reusable Semantic Vocabularies on the Web. *Semantic Web*, 1(1):1–16, 2015.

[194] Boris Villazón-Terrazas, Mari Suárez-Figueroa, and Asunción Gómez-Pérez. A Pattern-Based Method for Re-Engineering Non-Ontological Resources into Ontologies. *International Journal on Semantic Web & Information Systems*, 6(4):27–63, October 2010.

[195] Boris Villazón-Terrazas, Luis M Vilches-Blázquez, Oscar Corcho, and Asunción Gómez-Pérez. Methodological Guidelines for Publishing Government Linked Data. In *Linking Government Data*, pages 27–49. Springer, 2011.

[196] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk - A Link Discovery Framework for the Web of Data. *LDOW*, 538, 2009.

[197] Timo Vuorisalo, Olli Arjamaa, Anti Vasemägi, Jussi-Pekka Taavitsainen, Auli Tourunen, and Irma Saloniemi. High Lactose Tolerance in North Europeans: A Result of Migration, Not in Situ Milk Consumption. *Perspectives in Biology and Medicine*, 55(2):163–174, 2012.

[198] Dydimus Zengenene, Vittore Casarosa, and Carlo Meghini. Towards a Methodology for Publishing Library Linked Data. In *Bridging Between Cultural Heritage Institutions*, pages 81–92. Springer, 2014.