LDBC

Cooperative Project

FP7 – 317548

D4.4.4 Benchmark Design for Mapping Management

Coordinator: [Antonis Loizou]

With contributions from: [Peter Boncz]

1st Quality Reviewer: Orri Erling
2nd Quality Reviewer: Evangelia Daskalaki

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M24
Actual delivery date:	M24
Version:	0.1
Total number of pages:	49
Keywords:	linked data, identity, mappings, linksets, resource equivalence

Abstract

The issue of entity equivalence is central to achieving true data integration on the Semantic Web. However current approaches rely on permanently asserting equivalence between two or more RDF resources. Moreover they tend to rely on manual, labour intensive ETL processes to generate such persistent linkage, and often do so using the `owl:sameAs` predicate; the strong semantics of which can be the cause of a plethora of issues. At the same time, in a variety domains, such equivalences are purely context dependant. This deliverable investigates 5 strategies to provide flexible entity equivalence, through leveraging the experience of the Open PHACTS data integration project in the pharmacology domain.

Executive Summary

This deliverable focuses on investigating various strategies to enable the provision of flexible resource equivalence across multiple datasets and identifying the major choke points associated with them. The main goal behind developing a benchmark for flexible equivalence reasoning is to stimulate innovation in this area by RDF data management software.

The Open PHACTS project [26] (to which LDBC is an Associated Partner) has developed an approach to capturing the meaning of such links in a compatible way to existing published linked data [10, 6, 5].

The approach allows a linked data application to apply different equivalence criteria for resources across various datasets through the use of *scientific lenses* – sets of rules that modify which links across dataset can be traversed. As the $meaning$ of the link needs to be published along with the mapping triples in order for the approach to be applicable, the project has also published a set of dataset descriptor guidelines[1] that specify the vocabularies and predicates which should be used in expressing the meaning of resource mappings across datasets.

The work described in this deliverable adopts the Open PHACTS approach and datasets to investigate different ways in which Linked Data stores may manage such mappings in order to provide flexible notions of operational equivalence between resources at query time.

A subset of the data made available by the Open PHACTS project, in terms of both data– and link–sets was used for the purposes of the mapping management benchmark, and 5 strategies for providing flexible entity equivalence within an RDF store are detailed herein.

In summary the 5 strategies are:

Query Expansion (QE): Given an input URI and Lens, first identify applicable mappings for each named graph and manually insert them to the query using `VALUES` clauses. This strateggy mirrors the approach taken by the Open PHACTS project. However, the need to explicitly list all mappings in the query text can quickly cause compilation errors when large number of mapping are retrieved. Finally, this approach is not applicable to analytical queries that take into account the majority of entities present in an RDF store.

Property Path (PP): Construct a SPARQL 1.1 Property Path expression based on the specified Lens to resolve mappings between an input URI and corresponding entity URIs in the other datasets. As the mappings do not need to be explicitly listed in the query text, the compilation issues mentioned above do not arise. However, the property path constructed needs to be evaluated once for each individual data graph, potentially causing a large overhead. However, this overhead can be alleviated through internal caching and reuse of intermediate results by the RDF stores.

Temporary Graph (TG): To avoid the repeated evaluation of the same property path, the TG approach investigates the performance inserting the mappings resulting form the application of the property path starting from the input URI in a temporary graph to be subsequently queried. Once the required queries are executed, the temporary graph is dropped, to be populated again using the next instance. A single predicate is used to link entities in the temporary graph, and all entities are directly connected to each other.

Named Graph (NG): The named Graph strategy takes TG to its logical conclusion: rather than creating and dropping a temporary graph for each instance, NG attempts to create one, persistent, graph per lens. Inside this graph, all compatible mappings applicable under the corresponding lens are directly connected using a single predicate.

Inferencing (OWL): The OWL strategy investigates whether the direct links between mappings applicable to a given lens can be pre–computed through the applications of OWL inference rules. This is implemented by asserting that the link predicates for linksets that hold under the given lens are equivalent

[1] `http://www.openphacts.org/specs/2013/WD-datadesc-20130912/`

(`owl:equivalentProperty`) to `owl:sameAs`. As there is no standard way of either isolating inferred triples in RDF stores (so that they could be subsequently dropped) or of declaring the RDF triples (through a named graph for example) a rule should be applied to, we investigated whether such rules, along with the linksets applicable for each lens can be loaded each in a separate repository.

The 5 strategies were evaluated using two state–of–the–art RDF stores, Ontotext OWLIM-SE and OpenLink Virtuoso OS and experimental results are reported upon.

Document Information

IST Project Number	FP7 – 317548	Acronym	LDBC
Full Title	LDBC		
Project URL	http://www.ldbc.eu/		
Document URL	http://www.ldbc.eu:8090/display/PROJECT/Deliverable+summary		
EU Project Officer	Carola Carstens		

Deliverable	Number	D4.4.4	Title	Benchmark Design for Mapping Management
Work Package	Number	WP4	Title	Semantic Choke Point Analysis

Date of Delivery	Contractual	M24	Actual	M24
Status	version 0.1		final ☐	
Nature	Report (R) ☒ Prototype (P) ☐ Demonstrator (D) ☐ Other (O) ☐			
Dissemination Level	Public (PU) ☒ Restricted to group (RE) ☐ Restricted to programme (PP) ☐ Consortium (CO) ☐			

Authors (Partner)	Antonis Loizou (VUA)			
Responsible Author	Name	Antonis Loizou	E-mail	a.loizou@vu.nl
	Partner	VUA	Phone	+31 20 59 87753

Abstract (for dissemination)	The issue of entity equivalence is central to achieving true data integration on the Semantic Web. However current approaches rely on permanently asserting equivalence between two or more RDF resources. Moreover they tend to rely on manual, labour intensive ETL processes to generate such persistent linkage, and often do so using the `owl:sameAs` predicate; the strong semantics of which can be the cause of a plethora of issues. At the same time, in a variety domains, such equivalences are purely context dependant. This deliverable investigates 5 strategies to provide flexible entity equivalence, through leveraging the experience of the Open PHACTS data integration project in the pharmacology domain.
Keywords	linked data, identity, mappings, linksets, resource equivalence

Version Log			
Issue Date	Rev. No.	Author	Change
15/09/2014	0.1	Antonis Loizou, Peter Boncz	First version

Table of Contents

1 Introduction

This deliverable focuses on investigating various strategies to enable the provision of flexible resource equivalence across multiple datasets and identifying the major choke points associated with them. Its main contribution is identifying such flexible equivalence reasoning as a problem and potential topic as for a future LDBC industry benchmark, which it motivates using experiences from one of the current flag-ship RDF data management projects, Open PHACTS. The current problem definition, and Open PHACTS datasets could be already regarded a benchmark, however in our opinion this report should be the starting point for more detailed benchmark development.

The main goal behind developing a benchmark for flexible equivalence reasoning is to stimulate RDF data management software (developers) to innovate in this area. All the various approaches tested here to address the need for flexible reasoning have significant drawbacks and certainly there is no ready–made solution for such flexible reasoning on the software market yet. On the technical level, there is a wide area open for exploring options. The two extremes for reasoning seem to be to either perform all reasoning offline during bulk–load (backward chaining, the OWLIM approach), though in case of flexible reasoning this leads to a need to re-import the data sets whenever the reasoning settings change; whereas the other extreme is to leave all reasoning for runtime (forward chaining, the Virtuoso approach). Yet, there must exist many technical alternatives in between these two, exploiting the notion that (i) different reasoning settings ("Scientific Lenses" in Open PHACTS - see later) may appear only infrequently and are re-used often, and (ii) different settings may be slight variations of each other which would allow to share the reasoning computational load between them, it could e.g. be possible to build indexing structures that serve multiple Scientific Lenses. All in all, the technological solution space is still rather unexplored. The lack of existing solutions is a hindrance for deployment of a full-fledged LDBC benchmark at this moment, as it could be too early on.

Let us now motivate the need for flexible equivalence reasoning in the field of RDF data management. Resource equivalence has always had a prominent role in knowledge representation; 'Leibniz's Law' states that for every x and every y, if whatever is true of x is true of y, then x is identical to [y] [1]. This notion of identity thus places the focus on the properties of resources, in that for two things to be the same *they must share all the same properties*. Such a view on identity however gives rise to a number of classical problems. Are clones of (digital) objects identical to each other ? What about mutable properties such as a person's age ? This focus on resource properties carried over to the Semantic Networks of the 80s[9], and now to the Semantic Web and Linked data [7, 18, 13, 20].

While the final star of Tim Berners–Lee's 5 star deployment scheme[1] is obtained by including links to other URIs ["so that they can discover more things"], it is interesting to note that in practical terms such linking can give rise to numerous issues if it is not accurately specified.

Using "Paul Allen" as an example, [13] shows that by resolving `owl:sameAs` links across Freebase[8] and DBPedia [3], a reasoner will infer the triple:

```
dbpedia:Paul_Allen's_house owl:sameAs dbpedia:Paul_Allen .
```

Such issues arise from the strong semantics attributed to the `owl:sameAs` predicate, namely that it is both $transitive$ and $symmetric$.

Halpin et.al. [18] have shown that the `owl:sameAs` predicate is being widely misused in practise. Having identified four different types of equivalence for which the predicate is typically used, they propose alternative predicates from the FOAF[11] and SKOS[23] vocabularies that provide more flexible notions of equivalence than `owl:sameAs`, along with the use of named graphs for denoting equivalence across different contexts.

While avoiding the misuse of `owl:sameAs` through using semantically weaker predicates can certainly alleviate problems associated with automatically deriving false equivalences between resources, the conditions under which such linking predicates apply remain largely unspecified. In turn, this creates significant issues in evaluating the compatibility of individual links in a data integration setting.

[1] `http://www.w3.org/DesignIssues/LinkedData.html`

The Open PHACTS project [26] has developed an approach to capturing the meaning of such links in a compatible way to existing published linked data [10, 6, 5]. The approach allows a linked data application to apply different equivalence criteria for resources across various datasets through the use of *scientific lenses* – sets of rules that modify which links across dataset can be traversed. As the $meaning$ of the link needs to be published along with the mapping triples in order for the approach to be applicable, the project has also published a set of dataset descriptor guidelines[2] that specify the vocabularies and predicates which should be used in expressing the meaning of resource mappings across datasets.

The work described in this deliverable adopts the Open PHACTS approach and datasets to investigate different ways in which Linked Data stores may manage such mappings in order to provide flexible notions of operational equivalence between resources at query time.

Chapter 2 provides an overview of the Open PHACTS project, detailing the mapping management approach developed. Chapter 3 then details the five strategies we investigated, while Chapter 4 describes the framework that was implemented to compare Linked Data store performance for each strategy, using the Open PHACTS datasets and API queries. Results obtained with Virtuoso and OWLIM are presented in Chapter 5. Finally, Chapter 6 provides a summary of our results, relates them to potential choke points for RDF engines, and outlines directions for future work in producing a mapping management benchmark.

[2] `http://www.openphacts.org/specs/2013/WD-datadesc-20130912/`

2 Identity Management in Open PHACTS

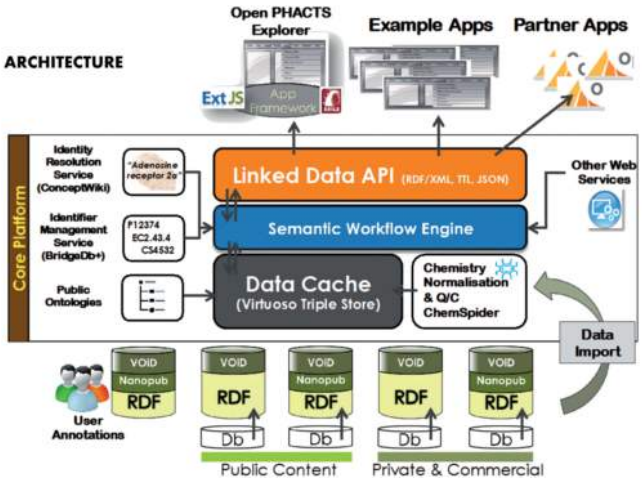

Figure 2.1: The Open PHACTS platform architecture overview[1]

This chapter provides an overview of the Open PHACTS project, focusing on the mapping management challenges and the approach developed to overcome them. The project has developed a semantic platform for the pharmacology domain that integrates data from a set of distributed linked open data sources [16].

Figure 2.1 presents the Open PHACTS platform architecture; at the bottom layer, the platform ingests a number of public and private datasets, described using the VoID vocabulary [2], as defined in [15]. In addition, data pertaining to chemistry is subject to normalisation and quality control, driven by the Royal Society of Chemistry[2]. The data is then loaded into the Open PHACTS Data Cache, currently an OpenLink Virtuoso RDF store.

The development of the platform was motivated through a set of pharmacology research questions provided by the domain experts [4]. These questions were used to obtain operational definitions for individual API end-points [17], implemented by extending Puelia[3], a PHP implementation of the Linked Data API (LDA)[4].

While the provision of an API is motivated by the same rationale that drove the development of the LDA vocabulary, a significant difference between the two is that Open PHACTS is strongly focused on data integration while the LDA itself aimed to provide easy to access representations of core resources inside single datasets[5] .

As datasets often provide numerous links to equivalent concepts in other datasets, these result in a profusion of "equivalent" identifiers for a concept. Other systems, such as Identifiers.org provide a single identifier that links to all the underlying equivalent dataset records for a concept. However, this approach constrains the system to a single view of the data, albeit a curated one.

However, domain scientists are willing to accept different forms of relationships between entities depending on the task at hand. For example, when trying to find the targets that a particular compound interacts with, some data sources may have created mappings to gene rather than protein identifiers. In such instances it may be acceptable to users to treat gene and protein IDs as being in some sense equivalent. However, in other situations this may not be acceptable and the platform needs to allow for this dynamic equivalence within a scientific context.

The Open PHACTS platform employes a novel approach to instance level links between datasets. Rather than hard coding the links into the datasets, the platform defers the instance level links to be resolved during query execution by the Identity Mapping Service (IMS). Thus, by changing the set of dataset links used to

[1] http://www.openphacts.org/images/stories/content/Architecture_graphic_500pxwide.png
[2] http://ops.rsc.org/
[3] https://code.google.com/p/puelia-php/
[4] https://code.google.com/p/linked-data-api/
[5] https://code.google.com/p/linked-data-api/wiki/AssumptionsAndGoals

execute the query, different interpretations over the data can be provided. The following sections provide details on how this behaviour is achieved, as well as identify limitations associated with this approach.

2.1 Dataset and Linkset descriptors

A dataset description is essential for data discovery and to enable consumers to use the data. It provides core metadata about a dataset such as its title, description, and license information. Information about how to access the dataset, e.g. a SPARQL endpoint location, which vocabularies have been used, and key statistics about the data can also be conveyed.

The Vocabulary of Interlinked Datasets (VoID) [2] provides a vocabulary of terms and a deployment model for dataset descriptions. A key feature of VoID is the ability to embed the dataset description with the data. This is achieved by the data linking back to its description using the `void:inDataset` predicate. The notion of a Linkset is also introduced by VoID as a collection of links between two datasets. In turn, a linkset description captures the context of such links; which datasets are linked, what predicate is used and so on. Some benefits of providing separate linksets are that the linksets can develop independently of the datasets and even be provided by third-parties enabling an ecosystem of links, as well as being used in co-reference services such as sameas.org[6].

However, VoID does not prescribe which properties must be provided to support discovery, linking and reuse. This makes the use of VoID dataset descriptions difficult as there is no guarantee that the information needed by an application will be provided. To overcome these challenges in the Open PHACTS project, a checklist of properties that must be provided [15] has been defined. Where needed, additional vocabulary terms for capturing the context linksets have been identified, for example the version number of a dataset is captured using the Provenance, Authoring and Version vocabulary (PAV) [12].

There can be many reasons to equate records across datasets. VoID linkset metadata captures details of the datasets linked, i.e. the context, and the link relationship. However, the link predicate tends to be either `owl:sameAs` or `skos:exactMatch`. The VoID linkset metadata does not capture the reason why the resources are considered equivalent, i.e. the *justification* for the links.

Many datasets contain links to other related datasets. For example, UniProt [22] includes links to several related datasets. However, the nature of these links are not captured; in the case of the RDF export of UniProt they are all stated as `rdfs:seeAlso`. One cannot distinguish between:

(i) two resources that capture different aspects of the same real–world concept

(ii) two resources that are highly related

(iii) a resource that is a relevant reference but not the same real–world concept.

It is therefore hard to automatically reuse such links due to the differing natures of the datasets and meaning of the link.

For users and applications to interpret and reuse equivalence relationships, they need to understand what notion of equivalence is being claimed. UniProt, in their RDF export, weaken their links to `rdfs:seeAlso` to avoid making inaccurate claims, but this reduces the knowledge conveyed. At the other extreme, the datasets in the Linked Data Cloud[7] widely use the predicate `owl:sameAs` [19], however they typically do not intend the strict semantics associated with it [18]. As such, these links need to be used with caution in many application domains, particularly in science. To alleviate such issues, the context of the links should be captured in the metadata of the link.

One approach for encoding the equivalence of a link between two data resources is to define a domain specific predicate. For example, one could define a predicate for stating that two chemicals share the same drug name and declare it as a sub–property of the `skos:exactMatch` predicate. This would allow standard inferencing rules to be applied. However there is a major social barrier to such an approach: gaining consensus on the

[6] `http://www.sameas.org`
[7] `http://linkeddata.org/`

required linking predicates. Additionally, there is the burden of updating the existing links in the datasets to use these new link predicates; a labour intensive task. Such an approach is unlikely to gain traction.

An alternative is to continue using existing link predicates such as `owl:sameAs` and `skos:exactMatch`, and annotate the linkset descriptions with additional contextual data that captures the equivalence criteria used to generate the links. This enables the use of the existing links unchanged. Thus, lowering the barrier to uptake as the annotations can be retrofitted to the existing links.

This additional metadata is refferred to as *justification* for the linkset; the notion captured is the scientific interpretation of the operational equivalence applied by the linkset. For example, the linkset relating ChemSpider and DrugBank because they have the same InChIKey string would express the justification in the corresponding VoID header with the following triples:

```
:Chemspider-Drugbank_Linkset void:linkPredicate skos:exactMatch ;
             bdb:linksetJustification cheminf:CHEMINF_000059 .
cheminf:CHEMINF_000059 rdfs:label ‘‘InChIKey’’ .
```

where `:Chemspider-Drugbank_Linkset` is the resource that describes the linkset, the link predicate is declared to be `skos:exactMatch` using the VoID predicate `void:linkPredicate`, and the justification is specified using the BridgeDB vocabulary (namespace `bdb`[8]). The value for an InChIKey is taken from the Chemical Information Ontology (namespace `cheminf`[9]).

The full set of supported justifications within the Open PHACTS Discovery Platform can be found in [15]. A key advantage of this approach is that it extends rather than changing the existing data, thus enabling additional metadata to be added later on with minimal effort.

2.2 Scientific Lenses

Users of data integration systems such as the Open PHACTS Discovery Platform expect answers to their queries despite discrepancies in the underlying data that make aligning the data in a fixed way difficult. For example, consider the retrieval of a record representing the cellular pathway for the human metabolism of aflatoxin. When mappings are based on entries sharing the same InChIKey, the record is only able to retrieved using one stereoisomer (*aflatoxin B116*) but not using another (*aflatoxin B117*).

This is due to one or more of the underlying data sources not containing details of the stereochemistry; this is common. However, the Open PHACTS users expect that the pathway would be returned for the call using either stereoisomer, since their notion of equivalence includes stereoisomers. Scientific Lenses are thus proposed as a means to enable different resource equivalence criteria to be applied for a given query by applying a different lens.

A lens defines a conceptual view over the data that varies the links between resources based on the notion of equivalence to be applied. Lenses are modelled in RDF and consist of the following:

- Identifier: Each lens is given a URI to identify it.
- Title (`dct:title`): Each lens is given a short descriptive title.
- Description (`dct:description`): Each lens has a textual description that explains the affect of the lens to a domain scientist.
- Documentation link (`dcat:landingPage`): A link to further explanation with illustrative examples of the affects of the lens.
- Creator (`pav:createdBy`): A link to a resource that represents the person that created the lens.
- Created (`pav:createdOn`): Timestamp of when the lens was created.

[8] `http://vocabularies.bridgedb.org/ops` (to appear soon)
[9] `http://semanticscience.org/resource/`

- Equivalence rules (`bdb:linksetJustification`): A set of URIs identifying the justifications that hold under the lens.

At present, minimal provenance information is captured using properties from the PAV ontology [12]. Detailed documentation of the effects of each of the lenses deployed on the Open PHACTS Discovery Platform is provided through the `dcat:landingPage` predicate, to demonstrate the effects of a lens using examples to show the changes in the results returned.

The Open PHACTS consortium is currently testing a number of lenses. One encapsulates a set of default expected behaviours; for instance this lens only equates chemicals that have the same InChI representation or where the datasets equate their identifiers. This lens provides the primary linking between resources and matches the behaviour of existing integration strategies and in particular that of the Open PHACTS Discovery Platform prior to the introduction of lenses.

Another lens, called the ChemistryParentChild, accepts a much larger set of relationships generated by the Chemical Registration Service. It is very permissive in its notion of equivalence, relating all resources that are variations of charge, isotopes, stereochemistry, salt forms, tautomers, and compounds in a mixture.

For the cellular pathway example given above, the users of the Open PHACTS API benefit from the use of lenses. Under the default lens, no pathways are returned due to individual datasets referring to different stereoisomers. Using the ChemistryParentChild lens, five pathways are returned.

Other lenses that only activate a subset of these variations, e.g. a stereochemistry lens, are available. However, it is noted that considerable effort is required to explain the behaviour of a given lens to the scientific users of the system.

2.3 Open PHACTS Identity Management Service (IMS), API and SPARQL query expansion

As stated above, the lenses functionality is provided within the Open PHACTS Discovery Platform by the Identity Mapping Service (IMS). The IMS provides a lookup service to return "equivalent" URIs for a given input URI. The notion of equivalence can be varied by supplying the URI for the lens to be applied. The IMS implementation is an extended version of the BridgeDb framework that maps database identifiers [25], extended to support cross-references over linked data sources. The source code is available on GitHub[10] and the service is accessible through the Open PHACTS API[11] [17].

Inferred linksets can be computed based on the justification of linkset. For example, a linkset between datasets A and C can be generated through some intermediary dataset B if there is a linkset between A and B and one between B and C such that both linksets have the same justification. Note that we do not require that the linksets have the same link predicate; the resulting inferred linkset is given the weaker of the two link predicates, and the same justification. Version 1.4 of the Open PHACTS IMS[12] currently indexes 117983250 mappings across 49 data sources. The service is implemented as a web service, exposing an API over a MySQL database [10]. The IMS API returns a list of mappings for a (set of) input URIs and lens specified; the mappings may be further restricted by specifying acceptable syntactical patterns through the `targetUriPattern` parameter.

In Open PHACTS, data integration is achieved through identifying the same entity across various datasets, and collating the available information together through various API [17] methods. This style of data integration is motivated by the pharmacology research questions [4] which provided the operational definitions for the Open PHACTS API methods.

In turn, Open PHACTS API methods are characterised by a corresponding SPARQL query, a mapping between HTTP parameters and `FILTER` or `ORDER BY` clauses, and one or more *Input URIs*. While these input URIs may not appear in the data themselves, Open PHACTS users expect that information stored against equivalent (under a particular lens) URIs is returned. To achieve this, Open PHACTS API methods invoke the IMS once

[10] `https://github.com/openphacts/IdentityMappingService`
[11] `https://dev.openphacts.org/`
[12] `http://openphacts.cs.man.ac.uk:9090/QueryExpander`

per each combination of input URI and target dataset, generating a `VALUES` clause for each named graph (i.e. dataset) in the corresponding SPARQL query. The clauses are inserted into the query which is then executed against the RDF store.

This approach is very efficient, as the computation of equivalence between URIs under different lenses is carried out offline, thus reducing the SPARQL queries issued to look–ups on properties of explicitly defined constants. This style of queries however can easily give rise to large cartesian products that can be problematic in their evaluation, especially when one of the more permissive lenses is specified. Another disadvantage of this approach is that the IMS cannot be invoked for intermediate results obtained in a query. While it is possible to achieve a similar effect by splitting the query so that the intermediate results are selected, and thus mappings for them obtained through the IMS, when intermediate results are numerous the resulting queries cannot be parsed by the RDF store due to their size. This prohibits the use of the IMS in evaluating complex analytic queries that involve large numbers of entities.

We note that such complex behaviour is rarely required by users of the Open PHACTS platform and that in such cases users are encouraged to interact with the API using workflow systems such as KNIME[13] or PipeLinePilot[14] [21].

[13] `https://www.knime.org/`

[14] `http://accelrys.com/products/pipeline-pilot/`

3 Mapping Management Benchmark Design

The Open PHACTS use–case provides a strong motivator for designing benchmarks that test the efficiency of different strategies for providing flexible resource equivalence. This chapter provides details on the five strategies investigated in this context.

3.1 Mapping Management Strategies

A similar setup to that used in Open PHACTS is assumed, namely that :

- Each dataset is loaded in a distinct named graph.
- Each linkset is loaded in a distinct named graph.
- Each linkset declares a justification[1] for the links included, and the link predicate used.[2]
- A set of lenses is defined, each one declaring which justifications are applicable.[3]

3.1.1 Query Expansion (QE)

The Query Expansion strategy is the one closest to the current Open PHACTS approach, in that preliminary queries are first evaluated in order to select appropriate mappings for a given input URI and lens. The mappings are in turn inserted in the final SPARQL query text as `VALUES` clauses.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX ops-ims: <http://openphacts.cs.man.ac.uk:9090/ontology/DataSource.owl#>
PREFIX void: <http://rdfs.org/ns/void#>

SELECT DISTINCT ?linkset_graph ?link_pred
WHERE {
  <LENS_URI> ops-ims:linksetJustification ?justification .
  GRAPH ?linkset_graph {
    ?linkset dul:expresses ?justification ;
      void:linkPredicate ?link_pred .
  }
}
```

Figure 3.1: SPARQL query to return linksets whose justification holds under a given lens, along with the link predicates used.

The process is split in three steps; first using the lens specified (`<LENS_URI>`), a SPARQL query is constructed using the template given in Figure 3.1. This query results in distinct pairs of applicable linkset graph names and corresponding link predicates. These are subsequently used to generate a `FROM` clause for each linkset named graph, and a SPARQL 1.1 property path [14] that traverses any of the resulting predicates, in either direction, zero or more times:

$$(\texttt{linkPredicate}_1|\hat{}\texttt{linkPredicate}_1|...|\texttt{linkPredicate}_n|\hat{}\texttt{linkPredicate}_n)^*$$

[1] `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#expresses`
[2] `http://rdfs.org/ns/void#linkPredicate`
[3] `http://openphacts.cs.man.ac.uk:9090/ontology/DataSource.owl#linksetJustification`

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>

SELECT DISTINCT ?g ?uri

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE

WHERE {
  VALUES ?g {
    <http://ops.rsc.org>
    <http://www.conceptwiki.org>
    <http://www.ebi.ac.uk/chembl>
    <http://linkedlifedata.com/resource/drugbank>
  }
  <INPUT_URI> PROP_PATH ?uri .
  GRAPH ?g {
    ?uri [] []
  }
}
```

Figure 3.2: SPARQL query to return mappings from each dataset given an input URI.

Figure 3.2 provides the SPARQL query template used in the second step. Here, `FROM_CLAUSE` and `PROP_PATH` are replaced respectively with the `FROM` clauses and property path obtained by the previous query, while `<INPUT_URI>` is replaced with the input URI given. The query is thus restricted to the named graphs that either contain a dataset, or a linkset whose justification holds under the lens specified through the `FROM` clauses and the resulting mappings are restricted to those which appear in the subject position of triples in one of the datasets. As the query selects both the resulting mapping and the graph it occurs in, a `VALUES` clause of the following form can easily be constructed for each dataset:

$$\texttt{VALUES ?mapping \{<URI}_1\texttt{> <URI}_2\texttt{> ... <URI}_n\texttt{> \}}$$

Finally, the resulting `VALUES` clauses are embedded inside additional SPARQL queries, that retrieve the required information from the RDF store. Such queries are presented in detail in Section 4.5. We note here that the QE strategy suffers from the same limitations as those identified above with respect to the Open PHACTS approach: mappings for intermediate results need to be resolved using intermediate queries, while the approach is not applicable for queries that involve a large number of entities such as those given in Section 4.5.2.

3.1.2 SPARQL 1.1 Property Paths (PP)

The PP approach investigates how the evaluation of queries is affected when the property path (and corresponding `FROM` clauses) obtained from the query template in Figure 3.1 is inserted directly into the final, information retrieval, queries.

While this approach does not suffer from the limitations described above regarding intermediate results and large numbers of entities, it does require that the property path appears once per dataset graph and is thus

evaluated multiple times. This is required since each solution sequence will contain bindings for the actual link predicates used to obtain each mapping (whether these are selected or not). However, to obtain mappings for one dataset may require the traversal of different link predicates, incompatible to those used in obtaining mappings for another. In contrast, the property path is used only once in the query provided in Figure 3.2 (not used in PP), as the formulation of the query allows for different link predicates to be bound in each solution sequence.

3.1.3 Temporary Graph (TG)

Under the Temporary Graph approach, all mappings applicable to a particular input URI and lens combination are first `INSERT`ed into a temporary graph, then the information retrieval queries are evaluated and subsequently the temporary graph is dropped. While the TG approach removes the need for multiple evaluation of a complex property pathin each query, it reintroduces the issues identified in QE regarding intermediate results and analytical queries. In addition, it is interesting to compare the performance of this approach to QE to evaluate how the overhead associated associated with the `INSERT` (see Figure 3.3) and `DROP` (see Figure 3.4) queries compares to evaluating queries obtained using the template of Figure 3.2. The preliminary query (see Figure 3.1) used to obtain applicable linksets and to generate the property path is used in the same way here as previously in PP and QE, with the exception that `USING` clauses are generated instead of `FROM`, in accordance to the SPARQL 1.1 Update specification [25, 24].

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

INSERT {
        GRAPH <http://www.ldbcouncil.org/mmb/tg> {
                <INPUT_URI> skos:exactMatch ?mapping
        }
}

USING_CLAUSE

WHERE {
        <INPUT_URI> PROP_PATH ?mapping
}
```

Figure 3.3: SPARQL query to insert all mappings for a given combination of input URI and lens in a temporary graph.

```
DROP SILENT GRAPH <http://www.ldbcouncil.org/mmb/tg>
```

Figure 3.4: SPARQL query to drop a previously constructed temporary graph.

3.1.4 Named Graph (NG)

The Named Graph strategy relies on pre–computing all transitive mappings applicable to each lens, prior to the execution of any information retrieval queries. The strategy is very similar to TG, however under NG the named graphs are persistent, and are constructed for all resource rather than with an explicitly defined starting point. The query used to create these graphs is given in Figure 3.5. The `USING_CLAUSE` and `PROP_PATH` are generated in the same way as previously (Fig. 3.1), while `LENS_GRAPH` is a distinct graph name for each lens.

3.1.5 Inference (OWL)

Rather than leveraging SPARQL Update to pre–compute all direct mappings applicable for each lens, the OWL strategy investigates whether the same effect can be achieved through OWL inference rules. This can

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

INSERT {
        GRAPH <LENS_GRAPH> {
                ?source skos:exactMatch ?mapping
        }
}

USING_CLAUSE

WHERE {
        ?source PROP_PATH ?mapping
}
```

Figure 3.5: SPARQL query to insert all mappings that hold under a given lens inside a named graph.

be achieved simply by asserting that the link predicates for linksets that hold under the given lens are equivalent (`owl:equivalentProperty`) to `owl:sameAs`.

However, the approach becomes problematic when a linkset whose justification does *not* hold under the given lens uses the same link predicate as others that do hold. The issue stems from the fact that there is no standard way of either isolating inferred triples in RDF stores (so that they could be subsequently dropped) or of declaring the RDF triples (through a named graph for example) a rule should be applied to.

This issue can be resolved in one of three ways :

- Maintain a separate RDF store instance for each lens, containing all datasets, but only the applicable linksets and corresponding rules.
- Load the datasets only in the RDF store and create a checkpoint. When a lens is specified at runtime, `INSERT` applicable linksets and corresponding rules. Restore the checkpoint whenever a different lens is specified.
- Load both the datasets and all linksets in the RDF store and create a checkpoint. When a lens is specified at runtime, `DROP` the named graphs for linksets with justifications that do not hold under the lens, and `INSERT` the appropriate rules. Restore the checkpoint whenever a different lens is specified.

4 Implementation

A first implementation of the mapping management benchmark was developed using a subset of the Open PHACTS datasets, linksets and lenses. The implementation also includes a set of scripts to generate and run workloads using each of the 5 strategies, and collect the results. The scripts are available on GitHub[1], while the full setup including the datasets is available online[2].

This chapter provides details on the first implementation of the benchmark.

4.1 Datasets

Four of the main datasets used in Open PHACTS were selected to implement the mapping management benchmark. This section provides a detailed description of each one.

4.1.1 Open PHACTS Chemistry Registration Service (OCRS)

The Open PHACTS Chemistry Registration service (OCRS) provides annotations for chemical substances, that are either assigned manually or computed based on a structural representation of the molecule in question. Figure 4.1 provides a graphical representation of the schema used in the RDF produced by the OCRS. The prefixes used in the figure are as follows :

```
cheminf:  <http://semanticscience.org/resource/>
obo:      <http://purl.obolibrary.org/obo#>
owl:      <http://www.w3.org/2002/07/owl#>
qudt:     <http://qudt.org/1.1/schema/qudt#>
rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs:     <http://www.w3.org/2000/01/rdf-schema#>
rdfs:     <http://www.w3.org/2001/XMLSchema#>
```

An entry for a chemical substance in OCRS (the node labelled `ocrs:{$id}` in Figure 4.1) is initially created using a set of `owl:annotationProperty` predicates, given at the top of the figure. At least one representation of the substances structure is required for the entry to be considered; this is can be achieved using the SMILES (`cheminf:CHEMINF_000018`) or InChI (`cheminf:CHEMINF_000396`) notations. This structural representation is first validated by the OCRS, and invalid chemical structures are rejected.

The OCRS then generates a set of 25 properties, computed based on the chemical structure provided using the ACD/Labs PhysChem[3] software library. A URI is generated for each property, which is attributed to the chemical substance entry using the 'is about' (`obo:IAO_0000136`) predicate.

The `rdf:type` predicate is used to denote the type of each property, whereas the numeric value, measurement unit used, and expected standard deviation are given respectively through the `qudt:numericValue`, `qudt:unit`, and `qudt:standardUncertainty` predicates from the Quantities, Units, Dimensions and Types ontology.[4]

Moreover, each computed property is asserted as the output (`obo:OBO_0000299`) of the execution of a particular software suite (`:{$id}execution`); in turn the software suite itself is given by the `rdf:type` of the execution, in this case, the ACD/Labs PhysChem software library, version 12.01 (`cheminf:CHEMINF_000354`).

The OCRS dataset consists of 216 451 600 triples, using 27 classes and 19 predicates in total.

[1] `https://github.com/antonisloizou/LDBC-MMB`
[2] `http://ldbc-mmb.ops.labs.vu.nl/`
[3] `http://www.acdlabs.com/`
[4] `http://qudt.org/1.1/schema/qudt`

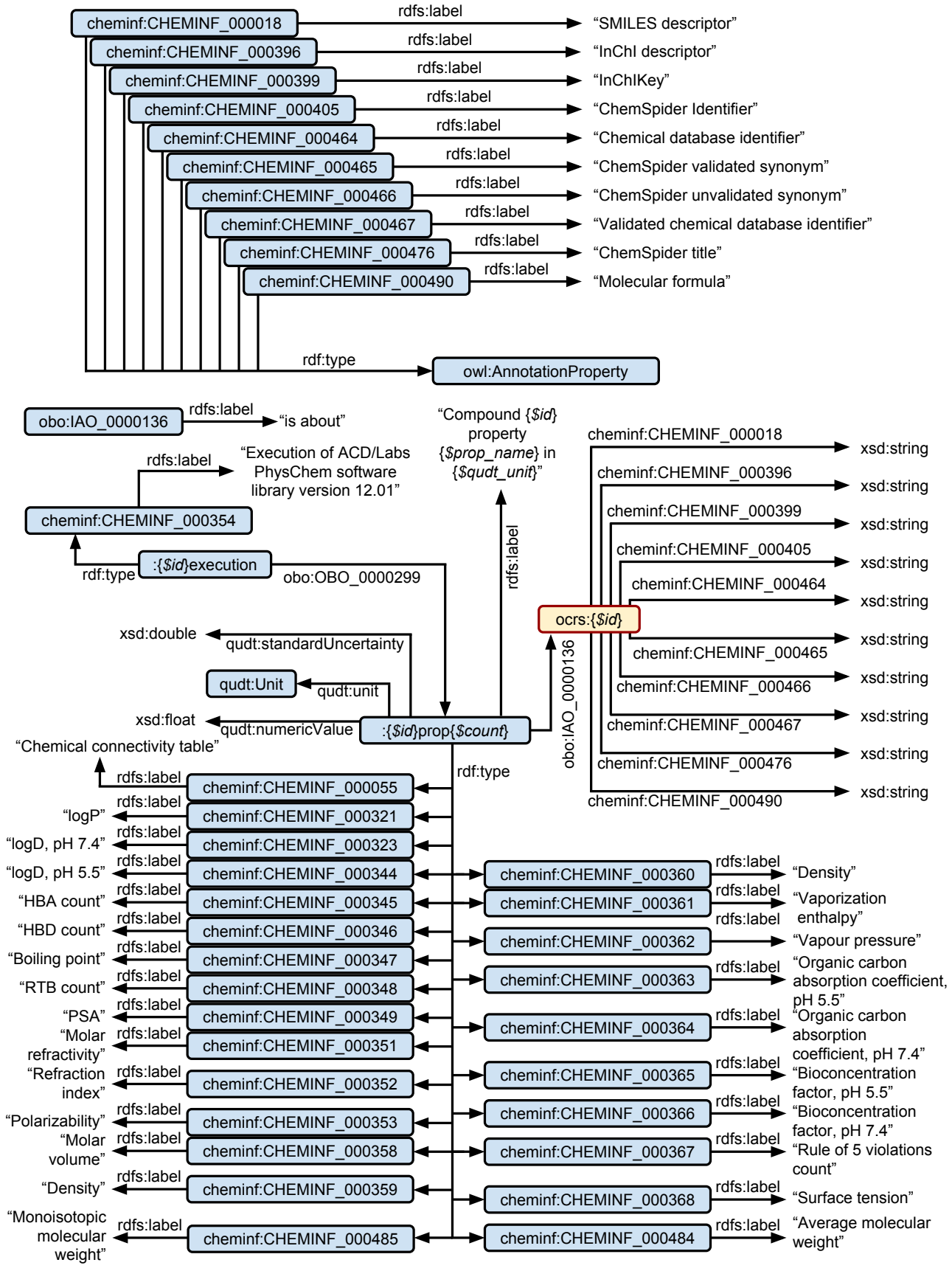

Figure 4.1: The OCRS RDF schema

4.1.2 ChEMBL

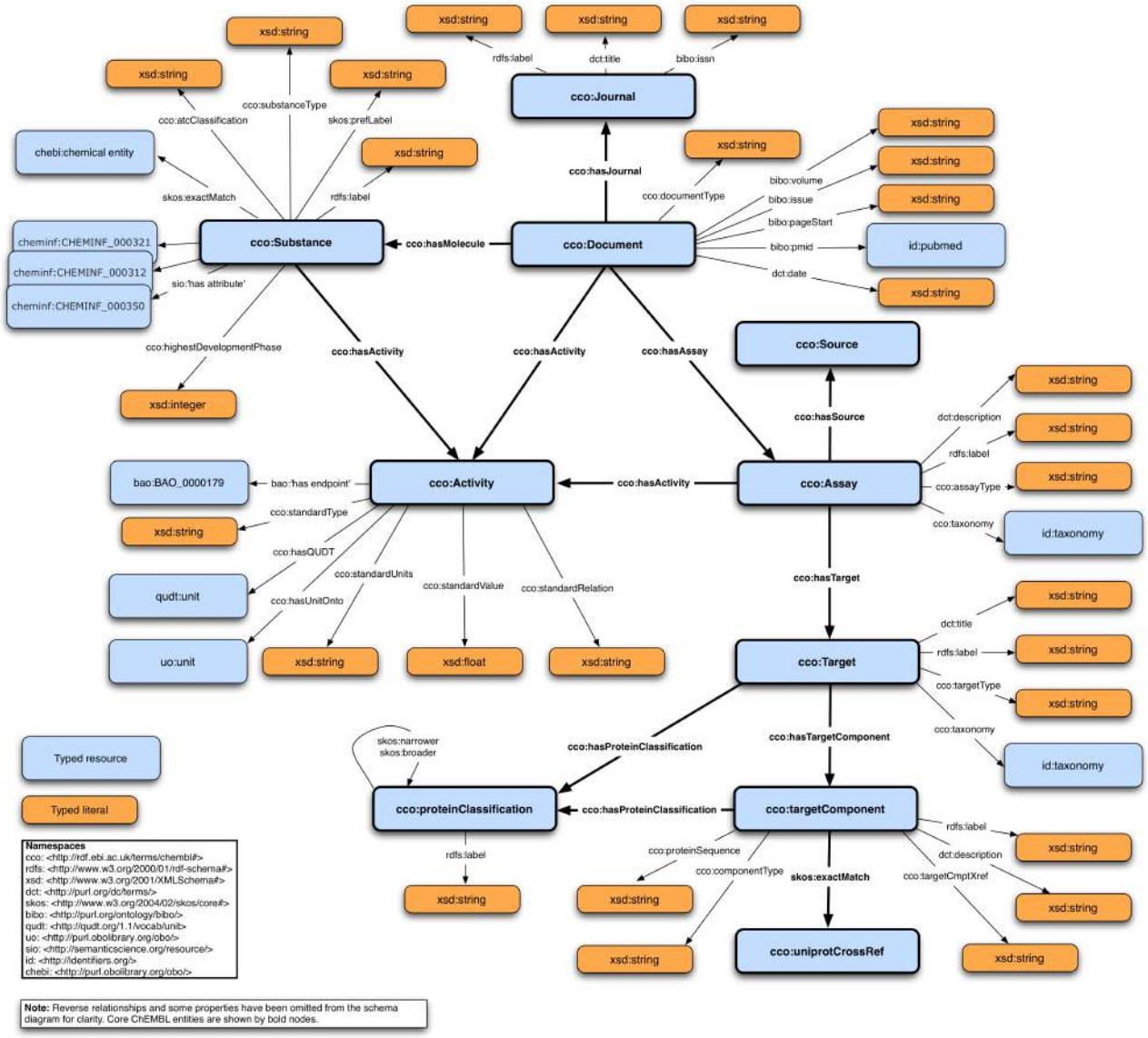

Figure 4.2: The ChEMBL RDF schema[5]

Figure 4.2 provides a graphical representation of the schema used to model the ChEMBL database in RDF. It focuses on pharmacology experiments, i.e. sets measurements that indicate the effects of a chemical substance (`cco:Substance`) on a biological entity (`cco:Target`).

The procedure through which the measurements are taken is defined as an `cco:Assay`, while the measurements themselves are modelled as `cco:Activity`. Activities are attributed to assays through the `cco:hasActivity` predicate, and are characterised by set of 4 datatype properties :

`cco:standardType`:	The type of experiment performed, for example “LD50” or “concentration”.
`cco:standardValue`:	The value recorded.
`cco:standardUnits`:	The unit used, e.g. “nanomolar”.
`cco:standardRelation`:	Denotes whether the measurement published is an exact value, or a lower/upper bound, using the strings ‘>’, ‘>=’, ‘=’, ‘<’, and ‘<=’

[5] `https://www.ebi.ac.uk/chembl/extra/RDF/chembl_rdf_schema.png`

Further, cco:Activity is the domain of three additional object properties:

- cco:hasQUDT and cco:hasUnitOnto provide respectively the QUDT[6] and Unit Ontology[7] URIs for the units in which the object of cco:standardValue is expressed in.
- bao:BAO_0000208 ("has endpoint") provides a URI from the BioAssayOntology[8], equivalent to the object of cco:standardType.

A chemical substance is connected to its cco:Activity measurements directly, through the cco:hasActivity predicate. Substances can be annotated using various properties, some of which are also available in the OCRS dataset. The quality of these annotations is generally considered to be lower than that of the OCRS data. In addition, each ChEMBL entry provides a mapping to a corresponding ChEBI[9] entry.

cco:Target resources are not connected directly to cco:Activity objects; instead they are associated with assays using the cco:hasTarget predicate, and cco:Assay resources refer to the corresponding activities using the cco:hasActivityPredicate. The cco:taxonomy predicate connects both targets and assays to the organism they belong to, while a label and description is assigned to them via rdfs:label and dct:description respectively. A natural language description of the type of the resource is given by cco:assayType for Assays, and cco:targetType for Targets.

Targets consist of one or more cco:targetComponents; the protein sequence of each component is given by cco:proteinSequence while rdfs:label,dct:description and cco:componentType are used in a similar manner as above. Mappings to Uniprot[10] are given by skos:exactMatch, while mappings to other databases are given by the cco:targetCmptXref predicate. In addition, both Targets and their Components are classified under the ChEMBL protein classification hierarchy using the predicate cco:hasProteinClassification

Finally, references to the literature where the experimental results have been published is provided through instances of the cco:Document class. Document resources provide references for Substances, Activities, and Assays through the predicates cco:hasMolecule, cco:hasActivity, and cco:hasAssay respectively. The following terms from the Bibliographic ontology[11] are used to annotate a cco:Document: bibo:volume, bibo:issue, bibo:pageStart and bibo:pmid (PubMed[12] ID), as well the date of publishing through dct:date. In addition a cco:Document may be attributed to a cco:Journal whis is in turn annotated with a title (dct:title), a label (rdfs:label), and an ISSN (bibo:issn).

Numerous subclasses of cco:Substance and cco:Target are used in the RDF, but are omitted from the diagram as well as numerous inverse properties. In total, the ChEMBL RDF consists of 120 083 551 triples, 81 classes, and 85 predicates.

4.1.3 DrugBank

Figure 4.3 provides a graphical representation of the schema used in the FU-Berlin[13] conversion of the DrugBank[14] database.

DrugBank focuses on launched drugs, the biological entities that these drugs modulate, and interactions between pairs of drugs. The RDF uses a total of 119 predicates and 5 classes. Due to the large number of predicates, and the fact that the vast majority are datatype properties with self–explanatory names, only the main elements of the schema will be described here.

Drugs are assigned one or more types (e.g. db:approved, db:experimental, db:illicit) and categories (e.g. db:antiCancerAgents, db:histamenAgonists, db:pesticides) through the predicates db:dugType

[6] http://qudt.org/1.1/schema/qudt
[7] http://www.berkeleybop.org/ontologies/uo.owl
[8] http://www.bioassayontology.org/bao/bao_complete.owl
[9] http://www.ebi.ac.uk/chebi
[10] http://www.uniprot.org
[11] http://purl.org/ontology/bibo/
[12] http://www.ncbi.nlm.nih.gov/pubmed
[13] http://www4.wiwiss.fu-berlin.de/drugbank/
[14] http://www.drugbank.ca/

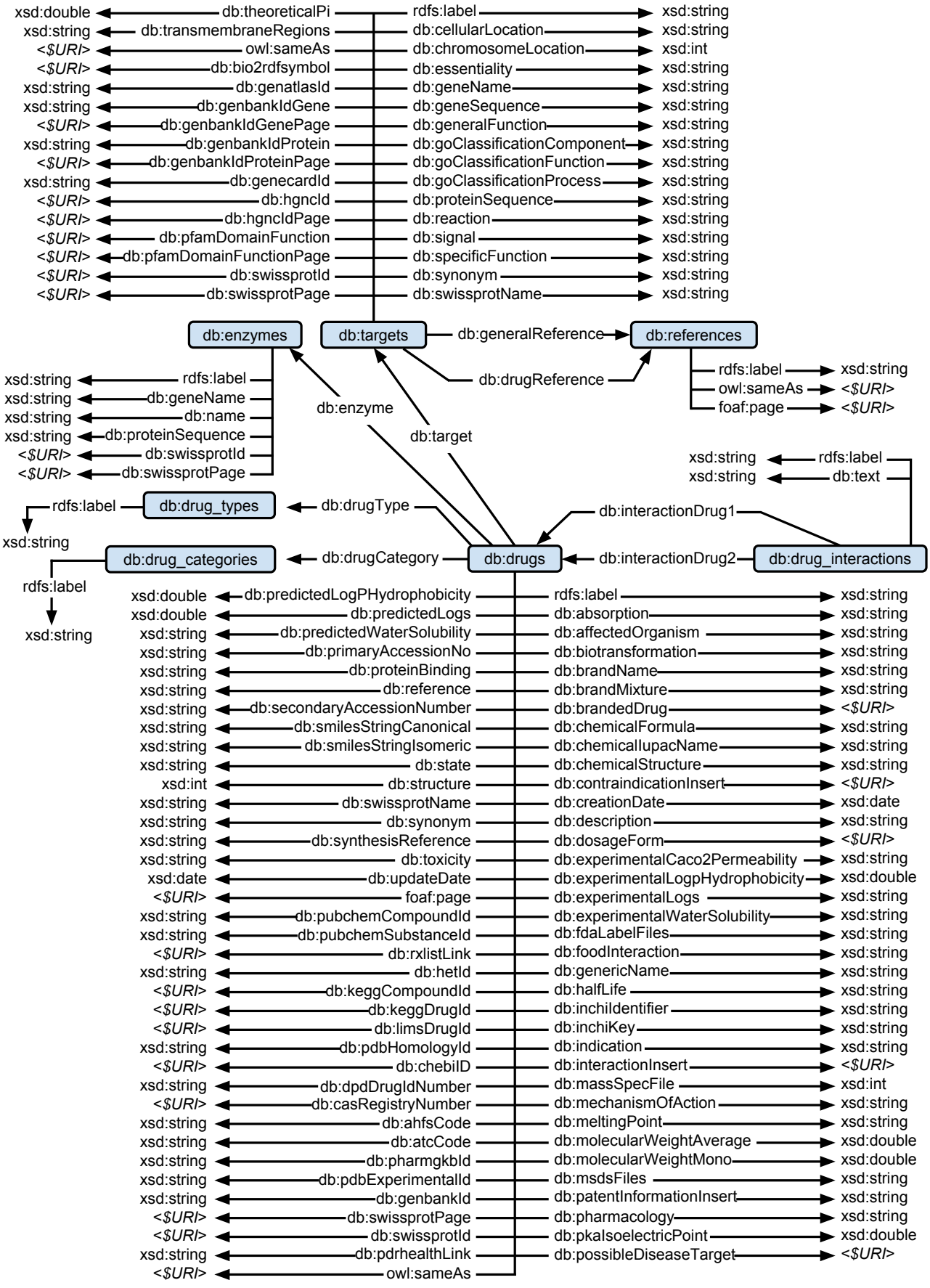

Figure 4.3: The DrugBank RDF schema

and db:drugCategory respectively. Binary interactions between drugs are modelled using db:drug_interactions resources, and the predicates db:interactionDrug1 and db:interactionDrug2 point to each of the two interacting drugs. Two db:drug_interactions resources are created for each pair of interacting drugs in the database, to cover both predicate–drug permutations.

Under the FU-Berlin DrugBank RDF model, drugs interact with biological entities of type db:targets or db:enzymes. db:targets are annotated using 36 predicates; a subset of 7 properties is used to annotate db:enzymes resources.

Finally, db:targets are associated with db:references via two predicates: db:generalReference and db:drugReference. db:generalReference is used to point to general literature about the target, while literature which discusses the interaction of the target with a drug is referred to using the db:drugReference predicate.

In total DrugBank contains 517 584 RDF triples.

4.1.4 ConceptWiki

ConceptWiki[15] is used as the "Identity Resolution Service (IRS)" in Open PHACTS, enabling the resolution of a textual label to a resource URI and vice–versa. The database itself is highly curated (manually), merging together concepts that represent the same entity across several sources, and assigning a canonical label to them. The RDF exported from ConceptWiki uses only a single predicate, skos:prefLabel, to connect a 3 024 385 resource URIs to their canonical labels. Additional information stored in the ConceptWiki system such as synonyms and external database identifiers is not yet available in RDF.

4.2 Linksets and justifications

All linksets available in Open PHACTS, both provided and computed, are available for download online.[16]. A subset has been selected for use in the mapping management benchmark, using IMS version 1.3.1. Any linkset pertaining to the type of entities (i.e. chemical substances and biological targets) described in the four datasets listed in the previous section is selected, resulting in 214 linksets in total.

These are split into two sets: primary and secondary linksets. Primary linksets are defined as those that provide mappings between any of the four datasets described in the previous section, while secondary linksets contain mappings for identifiers in other datasets. Table 4.1 provides a summary of the available linksets.

The table illustrates how the secondary linksets can sometimes necessary in resolving mappings between instances of the primary datasets; for example, the majority of the mappings between the OCRS and ConceptWiki datasets are obtained by combining the *ConceptWiki* $\rightarrow$ *ChemSpider* and *OCRS* $\rightarrow$ *ChemSpider* linksets. It is further noted that in order to combine these linksets, it is necessary to traverse one of the two link predicates in the opposite direction (from object to subject). In addition, Table 4.1 shows that identifiers present in any of the selected Open PHACTS linksets can potentially be used to obtain mappings for equivalent identifiers in the four primary datasets.

The selected linksets express 17 justifications, listed in Table 4.2. The table provides a short description for each justification and the number of linksets using it.

The most frequently used justification is 'Database cross–reference' (cheminf:SIO_001171). This justification encodes that the mappings were originally obtained directly from one or more databases without carrying out any validation.

In contrast, cheminf:CHEMINF_000059 is only used when the chemical structure (as expressed by the InChIKey) of both the resources being mapped is validated, and found to be identical. The six justifications cheminf:CHEMINF_000456, cheminf:CHEMINF_000458, cheminf:CHEMINF_000459, cheminf:CHEMINF_000460, obo:has_part and obo:is_tautomer_of then relax this requirement by allowing slight differences in the chemical structure regarding the stereochemistry, normalisation, isotopes, charge, subsumption and tautomerism.

[15] http://www.conceptwiki.org/

[16] http://openphacts.cs.man.ac.uk/

Subjects									Objects	
Primary			Secondary							
ChEMBL	ConceptWiki	OCRS	Ensembl	FAERS	HGNC	HMDB	EntrezGene	Uniprot		
1	4	7							ChEMBL	Primary
	5	7		1					DrugBank	
			1						BioGrid	Secondary
						1			CAS	
1		7							ChEBI	
	4	5							ChemSpider	
						1			DBPedia	
								1	Ensembl	
			9						EntrezGene	
								1	FlyBase	
								1	GeneID	
	3								GeneOntology	
			7						HGNC	
					1				HGNC Symbol	
		7							HMDB	
			1						Interpro	
			2					1	IPI	
						1			KEGG	
			2					1	MGI	
			2					1	MIM	
			9						miRBase	
	4	7							MSH	
	4								NCIM	
	2								OBO	
	3		10					1	PDB	
						1			PubChem	
			48					1	RefSeq	
			10						RFAM	
			1					1	RGD	
								1	SGD	
								1	UniGene	
			5						UniParc	
1	2		11				1		Uniprot	
	1								WikiPathways	
								1	WormBase	
			1					1	ZFIN	

Table 4.1: Overview of selected Open PHACTS linksets. Cell values indicate the number of linksets which provide mappings from the dataset indicated by the column to the dataset indicated by the row.

Justification URI	Description	# of Linksets
cheminf:CHEMINF_000059	The InChIKey of the two resources is the same	22
cheminf:CHEMINF_000456	Subclass relation between a class that has stereochemistry defined and an otherwise identical class that does not	5
cheminf:CHEMINF_000458	The object is the OPS normalised counterpart of the subject	5
cheminf:CHEMINF_000459	Subclass relation between a class that has isotopes specified and an otherwise identical class that does not	5
cheminf:CHEMINF_000460	Connects one molecule to another with identical heavy-atom connectivity which is neutral	5
cheminf:SIO_000985	Protein coding gene	39
cheminf:SIO_001107	Pathway	1
cheminf:SIO_001171	Database cross–reference	67
cheminf:SIO_010004	Chemical entity	4
cheminf:SIO_010035	Gene	27
cheminf:SIO_010043	Protein	18
cheminf:SIO_010299	Disease	3
cw:ConceptWikiGene	Manually curated gene–protein links	1
cw:ConceptWikiProtein	Manually curated protein–protein links	1
edam:data_2342	Pathway name	1
obo:has_part	The object is part of the subject	5
obo:is_tautomer_of	Tautomer: the two resources readily interconvert.	5

Prefixes:

```
cheminf:  <http://semanticscience.org/resource/>
obo:      <http://purl.obolibrary.org/obo#>
cw:       <http://example.com>
edam:     <http://edamontology.org/>
```

Table 4.2: Descriptions and frequency of use for linkset justifications used in Open PHACTS.

A set of further 5 justification are available to encode for identifiers which represent the same real–world entity: `cheminf:SIO_001107` for biological pathways, `cheminf:SIO_010004` for chemical entities, `cheminf:SIO_010035` for genes, `cheminf:SIO_010043` for proteins and `cheminf:SIO_010299` for diseases.

Next, `cheminf:SIO_000985`, `cw:ConceptWikiGene` and `cw:ConceptWikiProtein` provide different notions of equivalence between genes and proteins. `cheminf:SIO_001107` encodes a mapping relationship between a protein and the gene that transcribe it. As ConceptWiki creates distinct identifiers for both types of resources`cw:ConceptWikiGene` is used to equate gene identifiers form ConceptWiki to protein identifiers in other datasets. Reiterating that the ConceptWiki datasets consists of textual labels for resources in the domain, this linkset enables proteins to be associated with a gene name which is typically shorter and is preferred by certain communities of domain scientists. In contrast, `cw:ConceptWikiProtein` is used to encode mappings between protein only identifiers, between ConceptWiki and other datasets.

The final justification, `edam:data_2342`, is used to equate pathways that have been assigned the same name.

Justification	ops:default	ops:conceptWikiGene	ops:chemistryStrict	ops:chemistryTautomer	ops:chemistryParentChild	ops:chemistryChildParent	ops:chemistryParentChildTautomer	ops:chemistryChildParentTautomer	ops:chemistryAll	ops:all
cheminf:CHEMINF_000059	×	×	×	×	×	×	×	×	×	×
cheminf:CHEMINF_000456					×		×		×	×
cheminf:CHEMINF_000458					×		×		×	×
cheminf:CHEMINF_000459					×		×		×	×
cheminf:CHEMINF_000460					×		×		×	×
cheminf:SIO_000985	×	×		×	×	×	×	×	×	×
cheminf:SIO_001107	×	×	×	×	×	×	×	×	×	×
cheminf:SIO_001171	×	×	×	×		×		×	×	×
cheminf:SIO_010004	×	×	×	×	×	×	×	×	×	×
cheminf:SIO_010035	×	×	×	×	×	×	×	×	×	×
cheminf:SIO_010043	×	×	×	×	×	×	×	×	×	×
cheminf:SIO_010299										×
cw:ConceptWikiGene		×								×
cw:ConceptWikiProtein	×		×	×	×	×	×		×	×
edam:data_2342	×	×	×	×	×	×	×	×	×	×
obo:has_part					×	×	×	×	×	×
obo:is_tautomer_of				×			×	×	×	×

Table 4.3: Applicable justifications per lens defined in Open PHACTS. Prefixes for the justifications are given in Table 4.2, and ops: <http://www.openphacts.org/lenses/>.

4.3 Lenses

A set of 10 lenses have been defined in Open PHACTS, and Table 4.3 provides an overview of which justifications hold for each one.

The default lens (ops:default) is applied whenever a request is made without specifying a lens; the justification which hold were selected by the domain scientists to reflect justifications that are expected to provide 'common sense' mappings, applicable for the typical use cases in the pharmacology domain. Then, ops:conceptWikiGene supports the same justifications as the Default lens with one exception: the justification cw:ConceptWikiProtein is replaced with cw:ConceptWikiGene. As mentioned earlier, this facilitates the output of gene names rather than proteins, which is preferable in certain communities.

Next, the strict chemistry lens (ops:chemistryStrict) removes the equivalences between genes and the proteins they transcribe, as they are not strictly speaking the same chemical substance. The requirement for matching chemistry is then iteratively relaxed over the next six lenses, ops:chemistryTautomer through toops:chemistryAll where all justifications applicable to chemical substances hold.

Finally, ops:all is a lens under which all justifications hold, by definition.

4.4 Loading

Each dataset is loaded inside a distinct named graph as follows:

- OCRS: `<http://ops.rsc.org>`
- ConceptWiki: `<http://www.conceptwiki.org>`
- ChEMBL: `<http://www.ebi.ac.uk/chembl>`
- DrugBank: `<http://linkedlifedata.com/resource/drugbank>`

Similarly, each linkset is also loaded in a distinct graph. The names for these graphs are constructed by replacing spaces in the each filename with underscores, removing the file extension and adding the prefix `'http://www.openphacts.org/linksets/'`

Finally lenses are loaded into named graphs as well, using the URIs given in Table 4.3 as the graph names.

4.5 Queries

This section provides a description of the various SPARQL queries used in order to run the benchmark.

4.5.1 Lookup queries

The two most frequently used Open PHACTS API methods are "Compound Pharmacology: List" and "Target Pharmacology: List". These methods retrieve pharmacology experiments (`chembl:Activity`) for a given compound or target respectively, as well as additional information about the compound involved in each experiment. As paging is used in both methods, the response is an ordered list of a pre–specified number of activities. The number of activities that are returned is given by the 'page size' specified, and users may further specify the page number they would like to have displayed. To enable iterating through all pages for a given entity, two further methods are defined: "Compound Pharmacology: Count" and "Target Pharmacology: Count", which return the total number of experiments available.

The four queries have been reused in the mapping management benchmark, and are given by Figures 4.4 - 4.7. Mappings between identifiers for the entity in each datasets are resolved as described in Section 3.1, and depend on the strategy used. The queries listed herein are the variants used under the PP strategy.

These queries are referred to as 'Lookup queries' in the context of the mapping management benchmark: a single compound or target URI is provided, equivalent URIs from the various datasets are identified, and a set of properties for each one obtained. The two list methods (figures 4.4 and 4.6) make use of a SPARQL 1.1 subquery in order to limit the number of `chembl:Activity` resources returned, achieving the same effect as the 'page size' parameter of the Open PHACTS API.

It is important to note that the *compound* pharmacology queries only require mappings for the input URI in each of the dataset queried. However, in the case of the *target* pharmacology queries the input URI must first be mapped to a ChEMBL target URI, to retrieve interacting compounds in ChEMBL, which must in turn be mapped to instances in the other 3 datasets. Thus Q3 and Q4 are expected to be more costly than Q1 and Q2.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX ops: <http://www.openphacts.org/api#>

CONSTRUCT {

?activity chembl:hasMolecule ?chembl_uri ;
        chembl:standardType ?activity_type ;
        chembl:standardRelation ?activity_relation ;
        chembl:standardValue ?activity_value ;
        chembl:hasQUDT ?qudt_uri;
        chembl:publishedType ?published_type ;
        chembl:publishedRelation ?published_relation ;
        chembl:publishedValue ?published_value ;
        chembl:publishedUnits ?published_unit ;
        chembl:pChembl ?pChembl ;
        chembl:activityComment ?act_comment ;
        chembl:hasAssay ?assay_uri ;
        chembl:hasDocument ?doi ;
        bibo:pmid ?pmid ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?qudt_uri skos:prefLabel ?activity_unit .
?assay_uri chembl:hasTarget ?target_uri ;
        chembl:assayOrganismName ?assay_organism ;
        chembl:assayTestType ?assay_type ;
        dcterms:description ?assay_description  ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?target_uri dcterms:title ?target_name ;
        chembl:targetOrganismName ?target_organism ;
        chembl:hasTargetComponent ?protein ;
        a ?target_type ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?chembl_uri skos:exactMatch ?cw_compound_uri .
?cw_compound_uri skos:prefLabel ?compound_name ;
        void:inDataset <http://www.conceptwiki.org> .
?chembl_uri skos:exactMatch ?ocrs_uri .
?ocrs_uri ops:smiles ?smiles ;
        ops:inchi ?inchi ;
        ops:inchikey ?inchiKey;
        ops:molweight ?molweight ;
        ops:ro5_violations ?num_ro5_violations ;
        void:inDataset <http://ops.rsc.org> .
?chembl_uri skos:exactMatch ?db_uri ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?db_uri drugbank:drugType ?drugType ;
        drugbank:genericName ?drug_name ;
        void:inDataset <http://linkedlifedata.com/resource/drugbank> .

}

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE
```

Figure 4.4: Q1. SPARQL query corresponding to the 'Compound Pharmacology: List' method of the Open PHACTS API (PP variant).

```
WHERE {
        { SELECT ?activity ?chembl_uri ?target_uri WHERE {
                <INPUT_URI> PROP_PATH ?chembl_uri .
                GRAPH <http://www.ebi.ac.uk/chembl> {
                        ?activity chembl:hasMolecule ?chembl_uri ;
                                a chembl:Activity ;
                                chembl:hasAssay/chembl:hasTarget ?target_uri .
                }
        } ORDER BY ?activity LIMIT PAGE_SIZE }

        <INPUT_URI> PROP_PATH ?ocrs_uri .
        GRAPH <http://ops.rsc.org> {
                ?ocrs_uri cheminf:CHEMINF_000018 ?smiles ;
                        cheminf:CHEMINF_000396 ?inchi ;
                        cheminf:CHEMINF_000399 ?inchiKey .
                OPTIONAL { [] obo:IAO_0000136 ?ocrs_uri ;
                                a cheminf:CHEMINF_000484 ;
                                qudt:numericValue ?molweight . }
                OPTIONAL { [] obo:IAO_0000136 ?ocrs_uri ;
                                a cheminf:CHEMINF_000367;
                                qudt:numericValue ?num_ro5_violations . }
        }

        GRAPH <http://www.ebi.ac.uk/chembl> {
                ?activity chembl:hasAssay ?assay_uri .
                ?target_uri a ?target_type .
                OPTIONAL { ?target_uri dcterms:title ?target_name }
                OPTIONAL { ?target_uri chembl:organismName ?target_organism }
                OPTIONAL { ?target_uri chembl:hasTargetComponent ?protein }
                OPTIONAL { ?assay_uri dcterms:description ?assay_description }
                OPTIONAL { ?assay_uri chembl:assayTestType ?assay_type }
                OPTIONAL { ?activity chembl:publishedType ?published_type }
                OPTIONAL { ?activity chembl:publishedRelation ?published_relation }
                OPTIONAL { ?activity chembl:publishedValue ?published_value }
                OPTIONAL { ?activity chembl:publishedUnits ?published_unit }
                OPTIONAL { ?activity chembl:standardType ?activity_type }
                OPTIONAL { ?activity chembl:standardRelation ?activity_relation }
                OPTIONAL { ?activity chembl:standardValue ?standard_value }
                OPTIONAL { ?activity chembl:standardUnits ?activity_unit }
                OPTIONAL { ?activity chembl:hasQUDT ?qudt_uri }
                OPTIONAL { ?activity chembl:pChembl ?pChembl }
                OPTIONAL { ?activity chembl:activityComment ?act_comment }
                OPTIONAL { ?activity chembl:hasDocument ?doc_uri .
                        OPTIONAL { ?doc_uri owl:sameAs ?doi }
                        OPTIONAL { ?doc_uri bibo:pmid ?pmid }
                }
        }

        <INPUT_URI> PROP_PATH ?cw_compound_uri .
        GRAPH <http://www.conceptwiki.org> {
                ?cw_compound_uri skos:prefLabel ?compound_name .
        }

        OPTIONAL {
                <INPUT_URI> PROP_PATH ?db_uri
                GRAPH <http://linkedlifedata.com/resource/drugbank> {
                        ?db_uri drugbank:genericName ?drug_name ;
                                drugbank:drugType ?drugType_uri .
                        ?drugType_uri rdfs:label ?drugType.
                }
        }

}
```

Figure 4.4: (cont.) Q1. SPARQL query corresponding to the 'Compound Pharmacology: List' method of the Open PHACTS API (PP variant).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX ops: <http://www.openphacts.org/api#>

SELECT ( COUNT ( DISTINCT ?activity ) AS ?activity_count )

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE

WHERE {

        <INPUT_URI> PROP_PATH ?chembl_uri .
        GRAPH <http://www.ebi.ac.uk/chembl> {
                ?activity chembl:hasMolecule ?chembl_uri ;
                        a chembl:Activity ;
                        chembl:hasAssay ?assay_uri  .
                ?assay_uri chembl:hasTarget ?target_uri .
                ?target_uri a ?target_type .
        }

        <INPUT_URI> PROP_PATH ?ocrs_uri .
        GRAPH <http://ops.rsc.org> {
                ?ocrs_uri cheminf:CHEMINF_000018 ?smiles ;
                        cheminf:CHEMINF_000396 ?inchi ;
                        cheminf:CHEMINF_000399 ?inchiKey .
        }

        <INPUT_URI> PROP_PATH ?cw_compound_uri .
        GRAPH <http://www.conceptwiki.org> {
                ?cw_compound_uri skos:prefLabel ?compound_name .
        }

}
```

Figure 4.5: Q2. SPARQL query corresponding to the ‘Compound Pharmacology: Count’ method of the Open PHACTS API (PP variant).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX ops: <http://www.openphacts.org/api#>

CONSTRUCT {

?activity chembl:hasMolecule ?chembl_compound ;
        chembl:standardType ?activity_type ;
        chembl:standardRelation ?activity_relation ;
        chembl:standardValue ?activity_value ;
        chembl:hasQUDT ?qudt_uri;
        chembl:publishedType ?published_type ;
        chembl:publishedRelation ?published_relation ;
        chembl:publishedValue ?published_value ;
        chembl:publishedUnits ?published_unit ;
        chembl:pChembl ?pChembl ;
        chembl:activityComment ?act_comment ;
        chembl:hasAssay ?assay_uri ;
        chembl:hasDocument ?doi ;
        bibo:pmid ?pmid ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?qudt_uri skos:prefLabel ?activity_unit .
?assay_uri chembl:hasTarget ?chembl_uri ;
        chembl:assayOrganismName ?assay_organism ;
        chembl:assayTestType ?assay_type ;
        dcterms:description ?assay_description  ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?chembl_uri dcterms:title ?target_name ;
        chembl:targetOrganismName ?target_organism ;
        chembl:hasTargetComponent ?protein ;
        a ?target_type ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?chembl_compound skos:exactMatch ?cw_compound_uri .
?cw_compound_uri skos:prefLabel ?compound_name ;
        void:inDataset <http://www.conceptwiki.org> .
?chembl_uri skos:exactMatch ?ocrs_uri .
?ocrs_uri ops:smiles ?smiles ;
        ops:inchi ?inchi ;
        ops:inchikey ?inchiKey;
        ops:molweight ?molweight ;
        ops:ro5_violations ?num_ro5_violations ;
        void:inDataset <http://ops.rsc.org> .
?chembl_uri skos:exactMatch ?db_uri ;
        void:inDataset <http://www.ebi.ac.uk/chembl> .
?db_uri drugbank:drugType ?drugType ;
        drugbank:genericName ?drug_name ;
        void:inDataset <http://linkedlifedata.com/resource/drugbank> .

}

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE
```

Figure 4.6: Q3. SPARQL query corresponding to the 'Target Pharmacology: List' method of the Open PHACTS API (PP variant).

```
WHERE {

        { SELECT ?activity ?chembl_uri ?chembl_compound WHERE {
                <INPUT_URI> PROP_PATH ?chembl_uri .
                GRAPH <http://www.ebi.ac.uk/chembl> {
                        ?activity chembl:hasAssay/chembl:hasTarget ?chembl_uri ;
                                a chembl:Activity ;
                                chembl:hasMolecule ?chembl_compound .
                }
        } ORDER BY ?activity LIMIT PAGE_SIZE}

        GRAPH <http://www.ebi.ac.uk/chembl> {
                ?activity chembl:hasAssay ?assay_uri .
                ?chembl_uri a ?target_type .
                OPTIONAL { ?chembl_uri dcterms:title ?target_name }
                OPTIONAL { ?chembl_uri chembl:organismName ?target_organism }
                OPTIONAL { ?chembl_uri chembl:hasTargetComponent ?protein }
                OPTIONAL { ?assay_uri dcterms:description ?assay_description }
                OPTIONAL { ?assay_uri chembl:assayTestType ?assay_type }
                OPTIONAL { ?activity chembl:publishedType ?published_type }
                OPTIONAL { ?activity chembl:publishedRelation ?published_relation }
                OPTIONAL { ?activity chembl:publishedValue ?published_value }
                OPTIONAL { ?activity chembl:publishedUnits ?published_unit }
                OPTIONAL { ?activity chembl:standardType ?activity_type }
                OPTIONAL { ?activity chembl:standardRelation ?activity_relation }
                OPTIONAL { ?activity chembl:standardValue ?standard_value }
                OPTIONAL { ?activity chembl:standardUnits ?activity_unit }
                OPTIONAL { ?activity chembl:hasQUDT ?qudt_uri }
                OPTIONAL { ?activity chembl:pChembl ?pChembl }
                OPTIONAL { ?activity chembl:activityComment ?act_comment }
                OPTIONAL { ?activity chembl:hasDocument ?doc_uri .
                        OPTIONAL { ?doc_uri owl:sameAs ?doi }
                        OPTIONAL { ?doc_uri bibo:pmid ?pmid }
                }
        }

        ?chembl_compound PROP_PATH ?ocrs_uri .
        GRAPH <http://ops.rsc.org> {
                ?ocrs_uri cheminf:CHEMINF_000018 ?smiles ;
                        cheminf:CHEMINF_000396 ?inchi ;
                        cheminf:CHEMINF_000399 ?inchiKey .
                OPTIONAL { [] obo:IAO_0000136 ?ocrs_uri ;
                                a cheminf:CHEMINF_000484 ;
                                qudt:numericValue ?molweight . }
                OPTIONAL { [] obo:IAO_0000136 ?ocrs_uri ;
                                a cheminf:CHEMINF_000367;
                                qudt:numericValue ?num_ro5_violations . }
        }

        ?chembl_compound PROP_PATH ?cw_compound_uri .
        GRAPH <http://www.conceptwiki.org> {
                ?cw_compound_uri skos:prefLabel ?compound_name .
        }

        OPTIONAL {
                ?chembl_compound PROP_PATH ?db_uri .
                GRAPH <http://linkedlifedata.com/resource/drugbank> {
                        ?db_uri drugbank:genericName ?drug_name ;
                                drugbank:drugType ?drugType_uri .
                        ?drugType_uri rdfs:label ?drugType.
                }
        }
}
```

Figure 4.6: (cont.) Q3. SPARQL query corresponding to the ‘Target Pharmacology: List’ method of the Open PHACTS API (PP variant).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX ops: <http://www.openphacts.org/api#>

SELECT ( COUNT ( DISTINCT ?activity ) AS ?activity_count )

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE

WHERE {

        <INPUT_URI> PROP_PATH ?chembl_uri .
        GRAPH <http://www.ebi.ac.uk/chembl> {
                ?activity chembl:hasAssay  ?assay_uri ;
                        chembl:hasTarget ?chembl_uri ;
                        a chembl:Activity ;
                        chembl:hasMolecule ?chembl_compound .
                ?chembl_uri a ?target_type .
        }

        ?chembl_compound PROP_PATH ?ocrs_uri .
        GRAPH <http://ops.rsc.org> {
                ?ocrs_uri cheminf:CHEMINF_000018 ?smiles ;
                        cheminf:CHEMINF_000396 ?inchi ;
                        cheminf:CHEMINF_000399 ?inchiKey .
        }

        ?chembl_compound PROP_PATH ?cw_compound_uri .
        GRAPH <http://www.conceptwiki.org> {
                ?cw_compound_uri skos:prefLabel ?compound_name .
        }

}
```

Figure 4.7: Q4. SPARQL query corresponding to the 'Target Pharmacology: Count' method of the Open PHACTS API (PP variant).

4.5.2 Analytical queries

In addition to the Open PHACTS API queries described above, a set of 4 queries has been developed for the mapping management benchmark. We refer to these as *analytical* queries, as they are intended to traverse mappings between large numbers of instances. This characteristic renders the QE strategy not applicable, as expansion for large numbers of instances leads to queries that are too lengthy to parse. In addition, the TG strategy for these queries becomes identical to the NG strategy as the number of instances involved reaches the number of instances in the datasets.

Q5. Compound mappings statistics

This query computes the number of resources that can be reached in ConceptWiki, ChEMBL, and DrugBank, using OCRS compound URIs as a starting point. As the OCRS was developed to provide chemistry validation services, it is assumed that every compound entity in Open PHACTS has a corresponding OCRS URI. As such the query can be seen to evaluate the mapping coverage regarding compound entities across the four datasets.

```
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX db: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>

SELECT

( COUNT ( DISTINCT ?ocrs_compound ) AS ?ocrs_count)
( COUNT ( DISTINCT ?chembl_compound ) AS ?chembl_count )
( COUNT ( DISTINCT ?cw_compound ) AS ?cw_count )
( COUNT ( DISTINCT ?db_compound ) AS ?db_count )

FROM <http://ops.rsc.org>
FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE

WHERE {
        { SELECT DISTINCT ?ocrs_compound ?mapping WHERE {
                GRAPH <http://ops.rsc.org> {
                        ?ocrs_compound sio:CHEMINF_000405 []
                }
                ?ocrs_compound PROP_PATH ?mapping
        } }
        {
                GRAPH <http://www.ebi.ac.uk/chembl> {
                        [] chembl:hasMolecule ?mapping
                        BIND (?mapping AS ?chembl_compound)
                }
        } UNION {
                GRAPH <http://www.conceptwiki.org> {
                        ?mapping skos:prefLabel []
                        BIND (?mapping AS ?cw_compound)
                }
        } UNION {
                GRAPH <http://linkedlifedata.com/resource/drugbank> {
                        ?mapping a db:drugs
                        BIND (?mapping AS ?db_compound)
                }
        }
}
```

Figure 4.8: Q5. Compound mappings statistics (PP variant)

Q6. Target mappings statistics

This query computes the number of resources that can be reached in the OCRS, ChEMBL, and DrugBank, using ConceptWiki target URIs as a starting point. A similar assumption to the previous query is made, namely that ConceptWiki contains an entry for every target entity supported by the Open PHACTS platform. As such the query can be seen to evaluate the mapping coverage regarding target entities across the four datasets.

```
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX db: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>

SELECT
( COUNT ( DISTINCT ?cw_target ) AS ?cw_count )
( COUNT ( DISTINCT ?chembl_target ) AS ?chembl_count )
( COUNT ( DISTINCT ?db_target ) AS ?db_count )
( COUNT ( DISTINCT ?ocrs_target ) AS ?ocrs_count )

FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://linkedlifedata.com/resource/drugbank>
FROM <http://ops.rsc.org>

FROM_CLAUSE

WHERE {
        { SELECT DISTINCT ?cw_target ?mapping WHERE {
                GRAPH <http://www.conceptwiki.org> {
                        ?cw_target skos:prefLabel []
                        BIND (?mapping AS ?cw_compound)
                }
                ?cw_target PROP_PATH ?mapping
        } }
        {
                GRAPH <http://www.ebi.ac.uk/chembl> {
                        [] chembl:hasTarget ?mapping
                        BIND (?mapping AS ?chembl_target)
                }
        } UNION {
                GRAPH <http://linkedlifedata.com/resource/drugbank> {
                        ?mapping a db:targets
                        BIND (?mapping AS ?db_target)
                }
        } UNION {
                GRAPH <http://ops.rsc.org> {
                        ?mapping sio:CHEMINF_000405 []
                        BIND (?mapping AS ?ocrs_target)
                }

        }
}
```

Figure 4.9: Q6. Target mappings statistics (PP variant)

Q7. Target and corresponding active compound statistics

Q7 returns the CW target count per ChEMBL target type, and the number of *active* compounds in OCRS and their average molecular weight. For the purposes of this deliverable, *active* refers to compounds that have been experimentally tested against the target in activities of type *IC50*, and a value less than 50 nanomolars recorded. In addition, the query returns the number of active drugs per DrugBank compound type, for compounds that also appear in DrugBank.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX db: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX qudt-ops: <http://www.openphacts.org/units/>
PREFIX obo: <http://purl.obolibrary.org/obo/>

SELECT
?target_type
( COUNT ( DISTINCT ?chembl_target ) AS ?target_count )
( COUNT ( DISTINCT ?chembl_compound ) AS ?compound_count )
( AVG ( ?mol_weight ) AS ?avg_mol_weight )
?drug_type
( COUNT ( DISTINCT ?db_compound ) AS ?drug_count )

FROM <http://www.conceptwiki.org>
FROM <http://www.ebi.ac.uk/chembl>
FROM <http://ops.rsc.org>
FROM <http://linkedlifedata.com/resource/drugbank>

FROM_CLAUSE

WHERE {
        GRAPH <http://www.conceptwiki.org> {
                ?cw_target skos:prefLabel []
        }
        ?cw_target PROP_PATH ?chembl_target
        GRAPH <http://www.ebi.ac.uk/chembl> {
                ?chembl_target a ?target_type .
                ?assay_uri chembl:hasTarget ?chembl_target ;
                        chembl:hasActivity/chembl:standardType "IC50" ;
                        chembl:hasActivity/chembl:hasQUDT qudt-ops:Nanomolar ;
                        chembl:hasActivity/chembl:standardValue ?activity ;
                        chembl:hasActivity/chembl:hasMolecule ?chembl_compound .
                FILTER ( ?activity < 50)
        }
        ?chembl_compound PROP_PATH ?ocrs_compound
        GRAPH <http://ops.rsc.org> {
                [] obo:IAO_0000136 ?ocrs_compound ;
                        a cheminf:CHEMINF_000484 ;
                        qudt:numericValue ?mol_weight .
        }
        OPTIONAL {
                ?ocrs_compound PROP_PATH ?db_compound
                GRAPH <http://linkedlifedata.com/resource/drugbank> {
                        ?db_compound db:drugType ?drug_type
                }
        }
} GROUP BY ?target_type ?drug_type
```

Figure 4.10: Q7. Target and corresponding active compound statistics (PP variant)

Q8. Compound and corresponding active target statistics

Q8 returns the OCRS compound count and average molecular weight per DrugBank compound type. As above, *active* refers to targets that have been experimentally tested against the compound in activities of type *IC50*, and a value less than 50 nanomolars recorded. In addition, the query returns the number of active targets per ChEMBL target type, for compounds that also appear in ChEMBL.

```
PREFIX cheminf: <http://semanticscience.org/resource/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
PREFIX db: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX qudt-ops: <http://www.openphacts.org/units/>
PREFIX obo: <http://purl.obolibrary.org/obo/>

SELECT
?drug_type
( COUNT ( DISTINCT ?ocrs_compound ) AS ?compound_count )
( AVG ( ?mol_weight ) AS ?avg_mol_weight )
?target_type
( COUNT ( DISTINCT ?chembl_target ) AS ?target_count )

WHERE {
        GRAPH <http://ops.rsc.org> {
                [] obo:IAO_0000136 ?ocrs_compound ;
                        a cheminf:CHEMINF_000484 ;
                        qudt:numericValue ?mol_weight
        }
        ?ocrs_compound PROP_PATH ?db_compound
        GRAPH <http://linkedlifedata.com/resource/drugbank> {
                ?db_compound db:drugType ?drug_type
        }
        OPTIONAL {
                ?ocrs_compound PROP_PATH ?chembl_compound .
                GRAPH <http://www.ebi.ac.uk/chembl> {
                        ?activity_uri chembl:hasMolecule ?chembl_compound ;
                                chembl:standardType "IC50" ;
                                chembl:hasQUDT qudt-ops:Nanomolar ;
                                chembl:standardValue ?activity .
                        ?assay_uri chembl:hasActivity ?activity_uri ;
                                chembl:hasTarget ?chembl_target .
                        ?chembl_target a ?target_type
                        FILTER ( ?activity < 50)
                }
        }
} GROUP BY ?drug_type ?target_type
```

Figure 4.11: Q8. Compound and corresponding active target statistics (PP variant)

4.6 Software setup

The benchmark can be installed on Linux and Mac environments by cloning the GitHub repository[17] and running the `acquire_datasets.sh` bash script from the `./data` subdirectory. In turn the script will download the required files and populate the following directories :

- `./data/datasets/v1.4.0/` Contains 4 subdirectories, one for each dataset. The individual files that each datasets consists of are stored there.
- `./data/linksets/version1.3.1.alpha1/primary/` Contains a subdirectory for each linkset publisher. The linksets stored therein contain resources from one of the four datasets in either the subject or object position of mapping triples.
- `./data/linksets/version1.3.1.alpha1/secondary/` Contains a subdirectory for each linkset publisher. The linksets stored therein do not contain resources from any of the four datasets.

[17] `https://github.com/antonisloizou/LDBC-MMB.git`

- `./resources/compounds/primary/` The directory contains a file for each of the four datasets. Each file is a list of all compound URIs which appear in the corresponding dataset.
- `./resources/compounds/secondary/` The directory contains a file for each dataset that participates in a linkset. Each file is a list of all compound URIs which appear in the corresponding linkset.
- `./resources/targets/primary/` The directory contains a file for each of the four datasets. Each file is a list of all target URIs which appear in the corresponding dataset.
- `./resources/targets/secondary/` The directory contains a file for each dataset that participates in a linkset. Each file is a list of all target URIs which appear in the corresponding linkset.

4.6.1 Software dependencies

The following packages are required in order to run the benchmark (exact versions not necessary):

- bash shell
- cURL (v.7.35.0)
- RedLand Raptor[18] (v2.0.13)
- GNU sed (v4.2.2)
- GNU grep (v2.16)

4.6.2 Workload generator

The workload generator has been implemented as a bash script, `./wl_gen/generateWorkload.sh`. The script prompts for the following parameters :

- **Type bias:** The probability that a compound resource is selected, $P(compound)$. Default is 0.5, and $P(target) = 1 - P(compound)$.
- **Source bias:** The probability for selecting a resource URI from the primary datasets. Default is 0.5, and $P(secondary) = 1 - P(primary)$.
- **Compounds source:** The file to use in selecting compound resources. Default is picking a file randomly for each query.
- **Targets source:** The file to use in selecting target resources. Default is picking a file randomly for each query.
- **Lens URI:** The lens to be used. Default is picking a lens randomly for each query.
- **Page size:** The page size to use for lookup queries. Default is picking one of $[10, 25, 50]$ at random.
- **Workload size:** The number of individual sets of SPARQL query parameters that will be generated.
- **Output file:** The file to save the generated workload in. Consists of four columns: Resource type (C for compound and T for target), Resource URI, Lens URI and Page size.

[18] `http://librdf.org/raptor/`

4.6.3 Benchmark driver

A benchmark driver has been implemented for each mapping management strategy described in Section 3.1 as a collection of bash scripts, located in the relevant subdirectory of `./driver/`. While the number of scripts varies with each strategy, a workload can be run against a SPARQL Endpoint over HTTP using the `runWorkload.sh` script for each strategy. The following parameters are required for the script to execute:

- **Workload File:** The workload file used to generate requests.
- **SPARQL Endpoint URL:** The HTTP location of the SPARQL endpoint.
- **Result Dir:** Contains a file with response times for each request made. In addition, for `CONSTRUCT` queries the output is validated using Raptor, and the resulting triple count is stored in a separate file. For QE, the mappings obtained for each named graph are also recorded in individual files.
- **Log File:** All requests, including the SPARQL query text and the full responses are appended to this file.

In addition, a `generateTSV.sh` script is provided for each strategy. The script takes as parameters the workload file used and the results directory, and combines the results in a single tab–separated file for further analysis. The input parameters are given by the first 4 columns, followed by the response times obtained for each request made, the number of triples reported by Raptor for `CONSTRUCT` queries, and finally, for QE, the mappings retrieved for each named graph.

5 Results

Preliminary experimentation was carried out using two systems: OpenLink Virtuoso OS (version 07.10.3208-pthreads) and Ontotext OWLIM (version 5.5.7071). We also attempted running the experiments on SPARQL City's SPARQLverse RDF store, however we were unable to complete the data loading without errors.

For the lookup queries, a workload of 100 URIs was generated and used for all experiments described in this section, using the following parameters:

P(primary URI): `1`

P(compound): `0.5`

P(target): `0.5`

Lens: Selected randomly.

Limit: Selected randomly from `[10,25,50]`.

A timeout of 300s was applied to the execution of each single query (i.e. the total time can exceed 300s without incurring a timeout). Moreover, as only primary URIs were selected for the workload, all Pharmacology queries are expected to produce results. This was verified by executing the equivalent calls on the Open PHACTS API [1].

The following subsections provide a summary of the results, while the full TSV files can be found online[2].

5.1 OpenLink Virtuoso

OpenLink's Virtuoso was built from the `develop/7` branch on GitHub[3]. The only change made to the default configuration was to increase the total memory available to 32GB (`NumberOfBuffers = 2720000` and `MaxDirtyBuffers = 2000000`).

5.1.1 Loading

Data loading was carried out by populating the `LOAD_LIST` table, and running 8 loading threads in parallel (`rdf_loader_run();&`). The load was completed in 1690 seconds.

5.1.2 Query Expansion

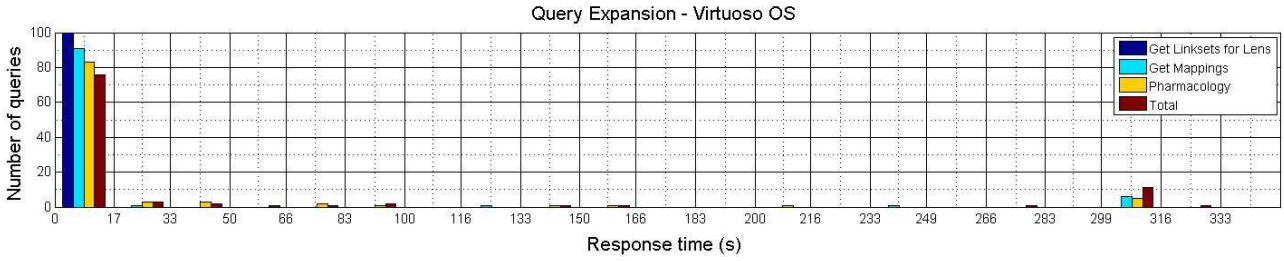

Figure 5.1: Histogram of the response times obtained using Virtuoso OS under the QE strategy.

Figure 5.1 provides a histogram of the response times obtained using Virtuoso under the QE startegy, using 20 evenly spaced bins. The first (blue) bar corresponds to the 'Get Linksets for Lens' query (Fig. 3.1), the

[1] `http://dev.openphacts.org/1.4/`
[2] `http://ldbc-mmb.d2s.labs.vu.nl/results/`
[3] `https://github.com/openlink/virtuoso-opensource`

second (cyan) is the execution time of the 'Get Mappings' query (Fig. 3.2), while the third (yellow) provides the execution for the Target/Compound Pharmacology `CONSTRUCT` queries. The fourth bar is the Total time, i.e. the sum of execution times for all 3 queries.

We observe that in the majority of instances (76/100), the total time taken is under 17 seconds. However, in certain cases both the 'Get Mappings for URI' query and the Pharmacology queries take substantially longer. In fact, 'Get Mappings for URI' exceeds the time–out threshold of 300 seconds 6 times, while 'Target Pharmacology' times out 4 times. No time outs occurred for 'Compound Pharmacology' queries. This is seen as empirical evidence that the target pharmacology query is significantly more expensive than the equivalent query based on a compound, as discussed theoretically above, in Section 4.5. In addition to the time-outs, Virtuoso failed to provide results under this strategy for a further 26 cases.

5.1.3 Property Path

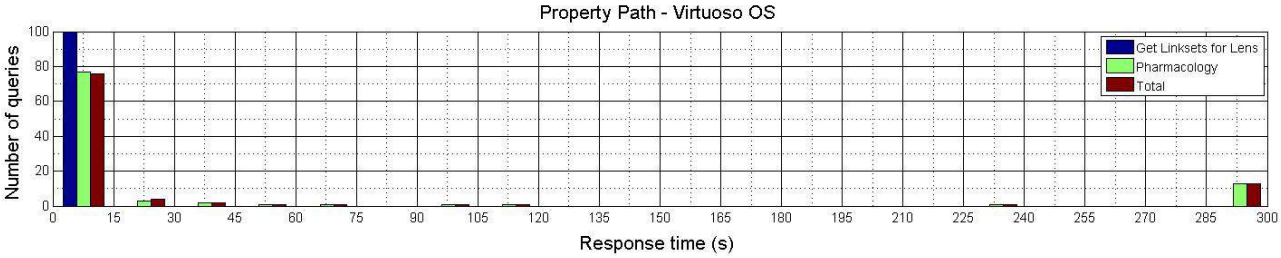

Figure 5.2: Histogram of the response times obtained using Virtuoso OS under the PP strategy.

The histogram of Figure 5.2 corresponds to the PP strategy, using 20 evenly spaced bins. Here, once the appropriate linksets are identified through the 'Get Linksets for Lens' query (first bar, blue), the pharmacology queries (second bar, green) are executed using an appropriately constructed property path, as discussed in 4.5. As with the QE strategy, the majority of queries return within 15 seconds total time (76/100). However the number of timeouts increases w.r.t. QE to 13 (as opposed to 10). These concern 9 targets and 4 compounds. Again, as with QE, 26 queries fail to produce results. These concern the same instances as before.

5.1.4 Temporary Graph

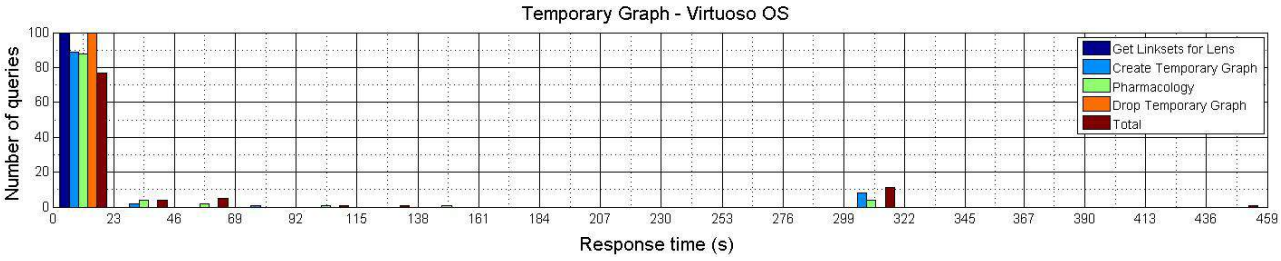

Figure 5.3: Histogram of the response times obtained using Virtuoso OS under the TG strategy.

As discussed in Section 4.5, and shown above in Figure 5.3, the Temporary Graph strategy consists of 4 queries:

- 'Get Linksets for Lens' : first bar; blue.
- 'Create Temporary Graph': second bar; cyan.
- 'Compound/Target Pharmacology': third bar; green.
- 'Drop Temporary Graph': fourth bar; orange.

As above, the final bar(5^{th}, red) is the total time taken to resolve the 4 aforementioned queries. However, as the total time taken here is longer, the 20 bins used become wider. Thus, 77 queries fall in the first bin, under 23 seconds total time. We observe 12 timeouts, 8 for 'Create Temporary Graph' and 4 'Target Pharmacology' queries (the same ones as with QE). Again, the same 26 queries produce no results.

5.1.5 Named Graph

Running the NG approach over Virtuoso has not been possible. The Lens instantiation query (Fig. 3.5) apparently requires more temporary transitive memory than is available (hard–coded) in Virtuoso, producing the error:

```
Message: TN...: Exceeded 100000000 bytes in transitive temp memory. use t_distinct, t_max or more T_MAX_memory options to limit the search or increase the pool
```

5.1.6 Inference

Virtuoso does not support OWL inferencing, hence the OWL strategy has not been executed.

5.2 Ontotext OWLIM

An 8-core license for OWLIM-SE (version 5.5.7071) was provided by Ontotext for the purposes of this deliverable. This was deployed on an Apache Tomcat application server (version 6.0.39.0), with the Java heap set to 32GB.

5.2.1 Loading

The data was loaded into an OWLIM-SE repository using the openrdf console (single threaded); the process took 35264.823 seconds. The 'OWL-Horst (optimised)' ruleset was used. The repository was initialised using the following parameters :

Total cache memory: 32GB

Main index memory: 24GB

Use predicate indices: True

Predicate index memory: 8GB

Use context index: True

Query time-out in seconds: 300

The remaining parameters were left to their default values. While no loading errors were reported, the Pharmacology queries failed to produce any results for all workloads. Regardless of this fact, the following sections report on the response times obtained with OWLIM-SE.

5.2.2 Query Expansion

As with Virtuoso, the first (blue) bar corresponds to the 'Get Linksets for Lens' query (Fig. 3.1), the second (cyan) is the execution time of the 'Get Mappings' query (Fig. 3.2), while the third (yellow) provides the execution for the Target/Compound Pharmacology `CONSTRUCT` queries. The fourth bar is the Total time, i.e. the sum of execution times for all 3 queries.

Under the QE strategy, only 55 out of the 100 instances in the workload returned in under 30s total time. 6 timeouts were observed for the 'Get Mappings for URI' query, while the Pharmacology queries timed out for 42 instances. Moreover, both the 'Get Mappings for URI' and 'Pharmacology' queries timed out in 5 cases.

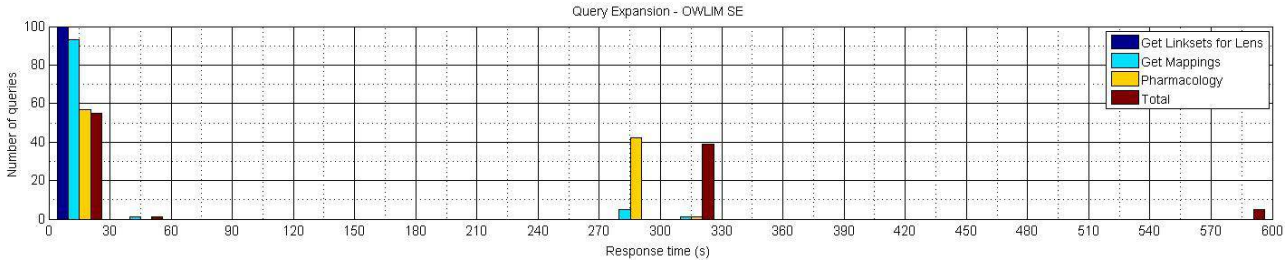

Figure 5.4: Histogram of the response times obtained using OWLIM-SE under the QE strategy.

However, when 'Get mappings for URI' fails (whether by timeout or by not producing any results), the string 'No mappings found' is bound to the subject position of the corresponding query, through a `VALUES` clause. As this by definition would not produce any results, it is interesting to note that OWLIM still executes the query until the 300s threshold is reached.

5.2.3 Property Path

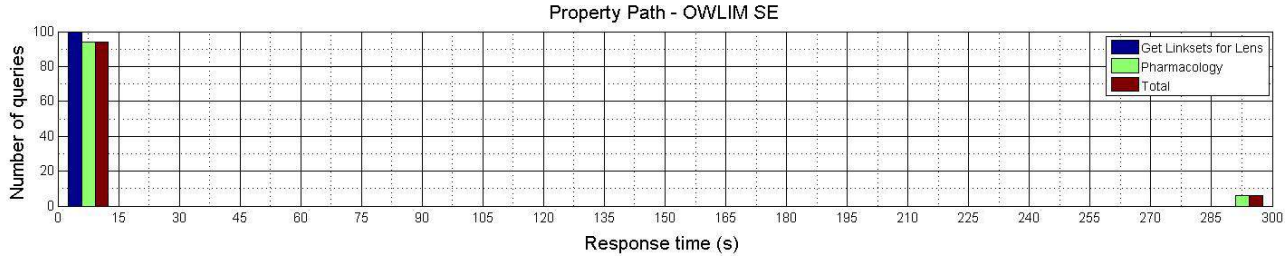

Figure 5.5: Histogram of the response times obtained using OWLIM-SE under the PP strategy.

The histogram of Figure 5.5 gives the response times obtained from OWLIM-SE. The first bar (blue) gives the response times for the 'Get Mappings for URI', the second (green) gives the response times for the two pharmacology queries, while the third (red) bar is the sum of response times for both queries.

The PP strategy appears to be particularly suited to OWLIM, as it outperforms the other strategies by a very large margin. Here, a striking 94 out of 100 instances are executed in less than 15s total time. The pharmacology queries time out for the remaining 6 instances, which are all targets. However, reiterating that none of the Pharmacology queries returned results for OWLIM-SE, it is not possible to assert that these queries were executed correctly.

5.2.4 Temporary Graph

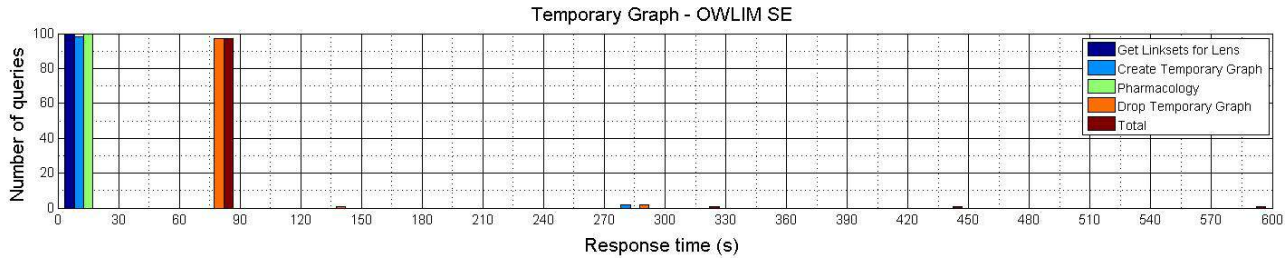

Figure 5.6: Histogram of the response times obtained using OWLIM-SE under the TG strategy.

The first 4 bars of Figure 5.6 give the response times for 'Get Linksets for Lens' (blue), 'Create Temporary Graph' (cyan), 'Compound/Target Pharmacology' (green), and 'Drop Temporary Graph' (orange), respectively. The final bar (red) gives the total time spent executing the aforementioned 4 queries.

We observe that the majority of time under this strategy is spent dropping the temporary graph; this is counter intuitive, as one would expect deletions to be faster that additions. Given that inferencing was enabled for this repository, we speculate that the time might be spent checking whether any inferred statements must be retracted as a consequence of the `DROP` operation. Further, we not that if the 'Drop Temporary Graph' query is excluded from the measurements, the response times obtained mirror those given by the PP strategy.

5.2.5 Named Graph

As with Virtuoso, the Named Graph approach was not possible to execute, as the 32GB Java Heap was exhausted for each of the 10 lenses.

5.2.6 Inference

While the OWL workload was developed with OWLIM in mind, once again we have not been able to obtain any results. The data was loaded to the repository for each lens, ignoring linksets that were not applicable. Then, loading the RDF statements declaring the applicable link predicates for each lens as `owl:equivalentProperty` to `owl:sameAs` was attempted through the OpenRDF console. A threshold of 10 hours (approx. 12 minutes longer than the total load time) was applied to the loading process. When this threshold was reached, the loading process was stopped, and the repository cleared and reloaded. The 10 hour threshold was exceeded for each of the 10 lenses.

6 Conclusions

This deliverable presented 5 strategies for providing flexible mappings between RDF resources, such that the criteria for equivalence between entities can be specified at run–time. Inspired by the approach used in the Open PHACTS project, these criteria are expressed through so called 'scientific lenses'; collections of link justifications that can be applied together under a certain context. Moreover, links between entities in different datasets are captured in linksets, each one declaring the justification for the links contained therein.

We used a subset of the data made available by the Open PHACTS project, in terms of both data– and link–sets. The 5 strategies were evaluated using two state–of–the–art RDF stores, Ontotext OWLIM-SE and OpenLink Virtuoso OS.

In summary the 5 strategies are:

Query Expansion (QE): Given an input URI and Lens, first identify applicable mappings for each named graph and manually insert them to the query using `VALUES` clauses. While this approach performs well with Virtuoso, with OWLIM we observe time–outs (at 300s) for almost half the instances tested. In addition, the need to explicitly list all mappings in the query text can quickly cause compilation errors when large number of mapping are retrieved. Finally, this approach is not applicable to analytical queries that take into account the majority of entities present in an RDF store.

Property Path (PP): Construct a SPARQL 1.1 Property Path expression based on the specified Lens to resolve mappings between an input URI and corresponding entity URIs in the other datasets. This approach is the fastest to execute using OWLIM, while the performance for Virtuoso is comparable to the QE strategy. Moreover, as the mappings do not need to be explicitly listed in the query text, the compilation issues mentioned above do not arise. However, the property path constructed needs to be evaluated once for each individual data graph, potentially causing a large overhead. This was not found to be the case in either of the two systems tested, as noted earlier with respect to performance, perhaps to the internal caching and reuse of intermediate results by the RDF stores.

Temporary Graph (TG): To avoid the repeated evaluation of the same property path, the TG approach investigates the performance inserting the mappings resulting form the application of the property path starting from the input URI in a temporary graph to be subsequently queried. Once the required queries are executed, the temporary graph is dropped, to be populated again using the next instance. A single predicate is used to link entities in the temporary graph, and all entities are directly connected to each other.

In terms of performance, TG is similar to both QE and PP for Virtuoso. For OWLIM-SE however, dropping the tempotary graph after each query execution introduces a substantially large overhead. This may be caused by various consistency check that may need to be carried out in an inference enabled store such as OWLIM-SE

Named Graph (NG): The named Graph strategy takes TG to its logical conclusion: rather than creating and dropping a temporary graph for each instance, NG attempts to create one, persistent, graph per lens. Inside this graph, all compatible mappings applicable under the corresponding lens are directly connected using a single predicate.

The execution of the Named Graph approach had not been possible for either system: Virtuoso has a hard–coded limit of "transitive temp. memory" which is consistently exceeded under this strategy, while the 32GB memory available to OWLIM was found insufficient for creating named graphs instantiating all applicable mappings under any of the 10 lenses.

Inferencing (OWL): The OWL strategy investigates whether the direct links between mappings applicable to a given lens can be pre–computed through the applications of OWL inference rules. This is implemented by asserting that the link predicates for linksets that hold under the given lens are equivalent (`owl:equivalentProperty`) to `owl:sameAs`. As there is no standard way of either isolating inferred

triples in RDF stores (so that they could be subsequently dropped) or of declaring the RDF triples (through a named graph for example) a rule should be applied to, we investigated whether such rules, along with the linksets applicable for each lens can be loaded each in a separate repository.

As Virtuoso does not support OWL inferencing, the OWL strategy was attempted to be executed only with OWLIM. However, loading the inference rules did not complete within the 10 hour timeout applied, for any of the ten lenses defined.

6.1 Future Work

Progress on this deliverable was halted due to unforeseen circumstances during the weeks leading up to the deadline. As such, we were only able to carry out limited experimentation. Future work is thus necessary to address the questions raised by the current results presented in Section 5.

In many instances, and in particular for the NG approach, the amount of memory available to the RDF stores was found to be the limiting factor. While for OWLIM this amounts simply to securing more hardware resources, for Virtuoso we need to investigate the possibility of raising the temporary transitive memory available, which is currently not a configurable parameter.

Similar issues affect the execution of the analytical queries presented in Section 4.5.2. Preliminary attempts at running these queries have been made, however these results are not reported herein, as we were not able to execute them using the experimental setup used for the look-up queries and time constraints prohibited setting up a second experimental environment.

Further, the fact that OWLIM-SE did not produce any results for the pharmacology queries, as reported in Section 5, requires further investigation, in collaboration with the RDF store vendor, Ontotext.

Finally, we intend to experiment with running workloads using these strategies with additional RDF stores and evaluated whether the same difficulties as those we observed using Virtuoso OS and OWLIM-SE are present.

REFERENCES

[1] HG Alexander. The leibniz-clarke correspondence, manchester, 1956. *AlexanderThe Leibniz-Clarke Correspondence1956*.

[2] Keith Alexander and Michael Hausenblas. Describing linked datasets-on the design and usage of void, theâĂŹvocabulary of interlinked datasets. In *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09*. Citeseer, 2009.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[4] Kamal Azzaoui, Edgar Jacoby, Stefan Senger, Emiliano Cuadrado Rodríguez, Mabel Loza, Barbara Zdrazil, Marta Pinto, Antony J Williams, Victor de la Torre, Jordi Mestres, et al. Scientific competency questions as the basis for semantically enriched open pharmacological space development. *Drug discovery today*, 18(17):843–852, 2013.

[5] Colin R Batchelor, Christian Brenninkmeijer, Christine Chichester, Mark Davis, Daniela Digles, Ian Dunlop, Chris Evelo, Anna Gaulton, Carole Goble, Alasdair J G Gray, Paul Groth, Lee Harland, Kenneth Karapetyan, Antonis Loizou, Jogn P Overington, Steve Pettifer, Jon Steele, Robert Stevens, Valery Tkachenko, Andra Waagmeester, Antony Williams, and Egon Willighagen. Scientific Lenses to support multiple views over linked chemistry data. In *The 13th International Semantic Web Conference (ISWC2014)*, Riva del Garda, Italy, October 2014.

[6] Colin R Batchelor, Christian Brenninkmeijer, Chris Evelo, Carole Goble, Alasdair J G Gray, Kenneth Karapetyan, Valery Tkachenko, and Egon Willighagen. Scientific Lenses over Linked Chemistry Data using BridgeDb and the Open PHACTS Chemical Registration System. In *10th Int. Conf. Chem. Struct.*, page 76, Noordwijkerhout, The Netherlands, June 2014.

[7] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *Intl Journal on Semantic Web and Information Systems (IJSWIS), Special Issue on Linked Data*, 2009.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[9] Ronald J. Brachman. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *Computer*, 16(10):30–36, 1983.

[10] Christian YA Brenninkmeijer, Chris Evelo, Carole Goble, Alasdair JG Gray, Paul Groth, Steve Pettifer, Robert Stevens, Antony J Williams, and Egon L Willighagen. Scientific lenses over linked data: An approach to support task specific views of the data. a vision. In *Proceedings of 2nd International Workshop on Linked Science*, 2012.

[11] Dan Brickley and Libby Miller. Foaf vocabulary specification 0.98. *Namespace Document*, 9, 2012.

[12] Paolo Ciccarese, Stian Soiland-Reyes, Khalid Belhajjame, Alasdair JG Gray, Carole A Goble, and Tim Clark. Pav ontology: provenance, authoring and versioning. *J. Biomedical Semantics*, 4:37, 2013.

[13] Li Ding, Joshua Shinavier, Zhenning Shangguan, and Deborah L McGuinness. Sameas networks and beyond: analyzing deployment status and implications of owl: sameas in linked data. In *The Semantic Web–ISWC 2010*, pages 145–160. Springer, 2010.

[14] Steve H Garlik, Andy Seaborne, and Eric PrudâĂŹhommeaux. Sparql 1.1 query language. *World Wide Web Consortium*, 2013.

[15] A Gray. Dataset descriptions for the open pharmacological space. working draft, open phacts (september 2013).

[16] Alasdair JG Gray, Paul Groth, Antonis Loizou, Sune Askjaer, Christian Brenninkmeijer, Kees Burger, Christine Chichester, Chris T Evelo, Carole Goble, Lee Harland, et al. Applying linked data approaches to pharmacology: Architectural decisions and implementation. *Nucleic Acids Res*, 40(D1):D1100–D1107, 2012.

[17] Paul Groth, Antonis Loizou, Alasdair JG Gray, Carole Goble, Lee Harland, and Steve Pettifer. Api-centric linked data integration: The open phacts discovery platform case study. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2014.

[18] Harry Halpin, Patrick J Hayes, James P McCusker, Deborah L McGuinness, and Henry S Thompson. When owl: sameas isn't the same: An analysis of identity in linked data. In *The Semantic Web–ISWC 2010*, pages 305–320. Springer, 2010.

[19] 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.

[20] Afraz Jaffri, Hugh Glaser, and Ian Millard. Uri disambiguation in the context of linked data. 2008.

[21] Hari K Machina and David J Wild. Laboratory informatics tools integration strategies for drug discovery integration of lims, eln, cds, and sdms. *Journal of laboratory automation*, 18(2):126–136, 2013.

[22] Michele Magrane, UniProt Consortium, et al. Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011:bar009, 2011.

[23] Alistair Miles and Sean Bechhofer. Skos simple knowledge organization system reference. *W3C recommendation*, 18:W3C, 2009.

[24] Simon Schenk, Paul Gearon, and Alexandre Passant. Sparql 1.1 update. *World Wide Web Consortium*, 2010.

[25] Andy Seaborne, Geetha Manjunath, Chris Bizer, John Breslin, Souripriya Das, Ian Davis, Steve Harris, Kingsley Idehen, Olivier Corby, Kjetil Kjernsmo, et al. Sparql/update: A language for updating rdf graphs. *W3C Member Submission*, 15, 2008.

[26] Antony J Williams, Lee Harland, Paul Groth, Stephen Pettifer, Christine Chichester, Egon L Willighagen, Chris T Evelo, Niklas Blomberg, Gerhard Ecker, Carole Goble, et al. Open phacts: semantic interoperability for drug discovery. *Drug discovery today*, 17(21):1188–1198, 2012.