A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information

Project acronym: MyHealthAvatar

Deliverable No. 4.2

Extension of the Semantic Core Ontology

Grant agreement no: 600929

Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D4.1
Document name:	Requirements analysis for semantic core ontology
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	0.6
Actual Submission Date:	28/02/2015
Editor: Institution: E-Mail:	Haridimos Kondylakis FORTH kondylak@ics.forth.gr

ABSTRACT:
This deliverable presents the next 2 phases of our methodology used for developing the MyHealthAvatar Semantic Core Ontology namely the conceptualization and the implementation phases of the ontology and the extension performed enable semantic lifting of all available data within the MyHealthAvatar project. In addition it presents the methodology and the technological approaches adopted for enabling data extraction and integration. More specifically the semantic integration methodology is presented and the mapping formalism adopted for this integration as well.

KEYWORD LIST:
Requirement Analysis, Semantic Core Ontology

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 600929.

---

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

MODIFICATION CONTROL			
Version	**Date**	**Status**	**Author**
0.1	01/10/2014	TOC	Haridimos Kondylakis
0.2	30/11/2014	First Draft	Haridimos Kondylakis
0.3	20/02/2015	Final Draft	Haridimos Kondylakis, Manolis Spanakis, Manolis Tsiknakis. Kostas Marias
0.4	21/02/2015	Review Version	Haridimos Kondylakis
0.5	25/02/2015	Commented Version	Haridimos Kondylakis
0.6	28/02/2015	Final	Haridimos Kondylakis

List of contributors

- Haridimos Kondylakis, FORTH-ICS
- Manolis Spanakis, FORTH-ICS
- Manolis Tsiknakis, FORTH-ICS
- Kostas Marias, FORTH-ICS

Contents

1 Executive Summary

This deliverable complements D4.1 Requirement Analysis for semantic core ontology [1] by presenting the next 2 phases of our methodology used for developing the MyHealthAvatar Semantic Core Ontology. More specifically, getting as input the analysis performed in D4.1 it presents the *conceptualization* and the *implementation* phases of the ontology and the extension performed enable semantic lifting of all available data within the MyHealthAvatar project.

In addition, although not foreseen in the DOW, this deliverable presents the methodology and the technological approaches adopted for enabling data extraction and integration. More specifically the semantic integration methodology is presented and the mapping formalism adopted for this integration as well.

2 Introduction

According to Wikipedia, Semantic Integration is the process of interrelating information from diverse sources [2] and has to resolve several heterogeneity problems. These problems are usually divided in syntactic heterogeneities and semantic heterogeneities. The former refer to those due to differences in the access interface, query language and database models. The latter are caused by different data representations for schemas or instances.

During the last 15 years, numerous systems have been developed, often targeting specific problems or areas. The main approaches, are either centralized – e.g. data warehouses, where data is stored locally – or federated – where data is left at the sources and accessed on demand. The selection of either approach depends on the type of solution to be deployed. Data warehouses might deal with data privacy issues and with outdated data. However, they provide better efficiency and allow tighter control to data managers over what data will be available. Federated approaches always access updated data, allow partial and non-managed data connections, but suffer from efficiency issues.

Either in virtual integration or data warehousing, during the last years, ontologies have been used in order to integrate structured and semi-structured data [3] , obtaining promising results, for example in the fields of biomedicine and bioinformatics [4] [5]. In MyHealthAvatar project the role of virtual schema has the MHA Semantic Core Ontology. It is a modular ontology consisting of an upper layer ontology named Translational Medicine Ontology [7] extended appropriately to cover the MyHealthAvatar project needs and several other modules. Those modules are well-known, established ontologies that are reused in the context of MyHealthAvatar. In Chapter 2, the aforementioned ontology is being described and explained.

Then we proceed further to describe the semantic integration methodology followed within MyHealthAvatar. We *adopt a hybrid semantic integration approach combining personal health information stored in a data warehouse and a federated semantic integration solution to integrate both internal and external data*. Federated approaches, also known as query translation rely on a virtual schema that represents the space of queries that the user can submit to the system. It is called 'virtual' because no data is stored centrally. Instead, each query is dynamically translated into a set of subqueries for the databases to integrate, and their single results are merged into a global result which is presented to the end-users as answer to his initial query. More specifically the data integration system that we will adopt and extend is the *exelixis* system [9]. The translated data are then stored in a Virtuoso triple store in order to enable efficient query answering.

In addition, when using ontologies to integrate data, one is required to produce mappings, to link similar concepts or relationships from the ontology/ies to the sources by way of an equivalence. This is the *mapping definition process* [6] and the output of this task is the *mapping*, i.e., a collection of mappings rules. In practice, this process is done manually with the help of graphical user interfaces and it is a *time-consuming*, *labour-intensive* and *error-prone* activity. In this deliverable we report on X3ML [8], a novel mapping formalism named X3ML, which is human readable and generic enough to cover most of the data models used nowadays. This model carefully distinguishes between mapping information from the domain experts who knows and provide the data and that created by the IT technicians, who actually implement data translation and integration solutions, and serves as an interface between both. This solves a major bottle neck in current information aggregation environments in terms of human resources and quality control. The aforementioned approach is described as part of Chapter 3.

2.1 Purpose of this document

According to the aforementioned claims, the purpose of this deliverable is to present the semantic backbone of the MyHealthAvatar project, namely the MHA Semantic Core Ontology, and the Data Integration methodology and mechanisms and the work performed on the mapping mechanisms between the ontology and the various data sources.

2.2 Structure of this document

The structure of this document is the following: Sections 3 describes the MyHealthAvatar Semantic Core ontology and presents its modules and components. Then Section 4 specifies the mechanisms and technologies that are used in order to enable the semantic integration of data within the project. Section 5 presents an example instantiation of the platform for the needs of the CHF scenario. Finally, Section 6 concludes this deliverable and presents directions for future work.

3 The MyHealthAvatar Semantic Core Ontology

As already stated in D4.1 [11] the development of the MHA Semantic Core ontology shall be based on the following three principles:

- **Reuse**: Avoid "reinventing the wheel" and reuse already established high quality ontologies.
- **Granularity**: Annotations or mappings cannot be extracted from a single ontological resource. So, multiple ontologies should be used.
- **Modularity**: Create a framework where different ontologies would be able to integrate many modules through mappings between ontologies.

Ontology construction is deemed to be a labour-intensive and a time-consuming process [9]. In addition, the development of new ontologies does not necessarily tap the full potential of existing domain-relevant knowledge sources. Due to these problems the latest years the tendency is not to create new ontologies from scratch but to try to integrate high quality, domain-specific ontologies that have already prove their value.

Projects like INTEGRATE[3] relied on a Common Information Model (CIM) based on HL7 and SNOMEDCT to represent the information, p-Medicine[4] used an ontology called HDOT to integrate already existing and well found ontologies, VPH NoE[5] identified mappings between different ontologies to enable interoperability and eHealthMonitor[6] extended an ontology named TMO to allow integrating several well-known ontologies.

In our case, the approach is similar to p-Medicine and eHealthMonitor, where a central ontology is used to integrate already existing ontologies. Since HDOT is a cancer specific ontology we are reusing and extending the more generic ontology named Translational Medicine Ontology. Similar to INTEGRATE our ontology is based on HL7 concepts and relies on SNOMEDCT ontology as well. However, besides SNOMEDCT other sub-ontologies are also present enabling the multilayer representation of available information.

In the following sections we provide a description of the MyHealthAvatar Semantic Core Ontology produced by extending the TMO in order to be aligned with the requirements of our use-cases.

3.1 MyHealthAvatar Semantic Core Ontology Modules

The MyHealthAvatar Semantic Core Ontology is consisted of 34 sub-ontologies integrated through the Extended TMO ontology (eTMO). The integration is achieved by introducing terms from these subontologies to the eTMO ontology or via equivalences identified between each sub-ontology and the eTMO ontology. TMO was initially integrated with 18 ontologies (in blue) and for the needs of

---

[3] http://www.fp7-integrate.eu/
[4] http://www.p-medicine.eu/
[5] http://www.vph-noe.eu/
[6] http://www.ehealthmonitor.eu/

MyHealthAvatar we integrated in 13 additional ontologies (in green) focusing more on the social, personal and life style information.

Figure 1. The modules of MyHealthAvatar Semantic Core Ontology (in green the extensions provided for MyHealthAvatar)

The explanation of the acronyms of the aforementioned ontologies and the domains they cover are shown in Table 1. The interested reader is forwarded to D4.1 [11] for a short description of each sub-ontology.

Table 1. Ontologies within MyHealthAvatar Semantic Core Ontology

Acronym	Ontology Title	Domain Described
ACGT	ACGT Master Ontology	Health Status & Clinical Information, Molecular Data, Medication Data
BFO	Basic Formal Ontology	Upper Layer Ontology
CHEBI	Chemical Entities of Biological Interest	Biological & Molecular Data
CIDOC-CRM	CIDOC Conceptual Reference Model	Upper layer ontology for documentation & Information Artifacts
CTO	Clinical Trial Ontology	Clinical Information
DO	Human Disease Ontology	Clinical Information
DTO	Disease Treatment Ontology	Clinical & Medical Information
FHHO	Family Health History Ontology	Personal Information & Lifestyle, Health Status & Clinical Information
FMA	Foundation Model of Anatomy	Health Status & Clinical Information

FOAF	Friend of a Friend Ontology	Social Information
GALEN	Galen Ontology	Medical Information
GO	Gene Ontology	Genomic Information
GRO	Gene Regulation Ontology	Genomic Information
IAO	Information Artifact Ontology	Upper Layer Ontology for Information Artifacts
ICD	International Classification of Diseases	Health Status & Clinical Information, Medication Data
ICO	Informed Consent Ontology	Health Status & Clinical Information
LOINC	Logical Observation Identifier Names and Codes	Health Status & Clinical Information
MESH	Medical Subject Headings	Health Status & Clinical Information, Medication Data
NCI-T	NCI therarurus	Health Status & Clinical Information
NIFSTD	Neuroscience Information Framework Standardized ontology	Molecular Data
NNEW	New Weather Ontology	Personal Information & Lifestyle
OBI	Ontology for Biomedical Investigation	BioMedical Infomation
OCRE	Ontology for Clinical Research	Clinical Information
OMRSE	Ontology of Medically Related Social Entities	Personal Information & Lifestyle
PATO	Phenotypic Quality Ontology	Clinical Information
PLACE	Place Ontology	Life-Style Information
PRO	Protein Ontology	Protein Information
RO	Relation Ontology	Upper layer Ontology
SBO	Systems Biology Ontology	Molecular Data, Systems Biology Models
SNOMED-CT	SNOMED clinical terms	Health Status & Clinical Information, Medication Data
SO	Sequence Ontology	Sequencing Information
SYMP	Symptom Ontology	Clinical & Health Information
TIME	Time Ontology	Lifestyle Information
UMLS	Unified Modeling Language System	Health Status & Clinical Information, Medication Data

3.2 The Extended Translational Medicine Ontology

The Extended TMO is an OWL compliant ontology and consists of about 10000 triples. In those triples we have 329 classes and 38 properties that represent the following entities relevant to biomedical studies:

- Materials: e.g. molecule, protein, cell lines, pharmaceutical preparations
- Processes: e.g. diagnosis, study, intervention
- Roles: e.g. subject target, active ingredient
- Informational Entities: e.g. dosage, mechanism of action, sign/symptom, family history

By contrast, particulars (e.g. “a patient with a given name” and “a blister package of a pharmaceutical product with a particular identifying code on it”) refer to individuals and are represented as instances of classes in the ontology. Consequently, a particular consultation at a given time and day, the particular patient role in that consultation, and the physician role in that consultation can be represented as instances of classes in the ontology.

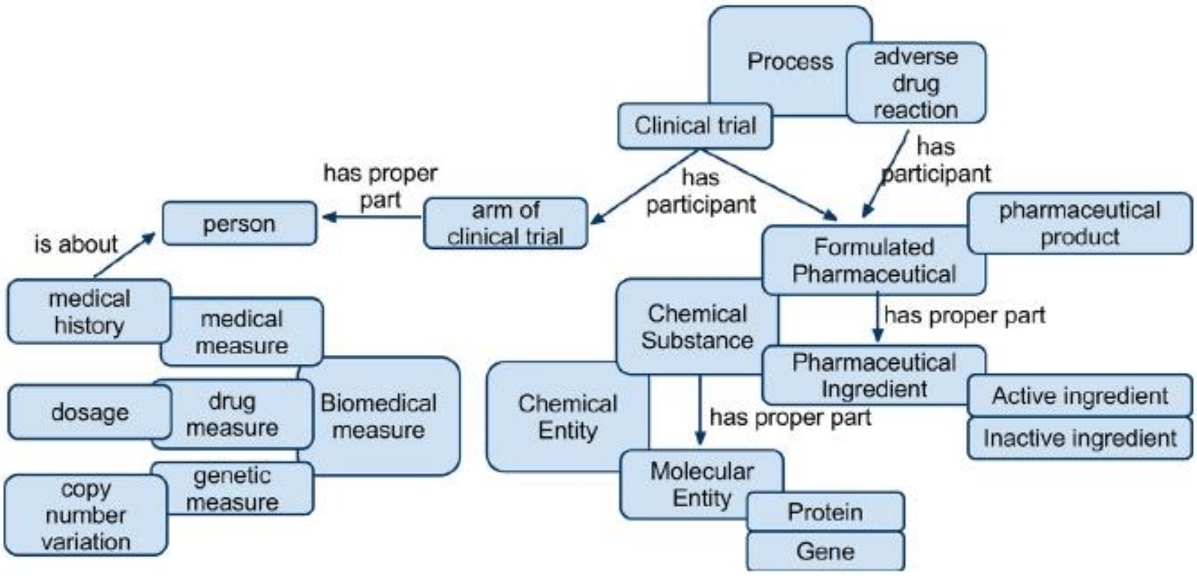


Figure 2. Overview of selected types, subtypes (overlap) and existential restrictions (arrows) in the TMO as presented in [7]

Figure 2 demonstrates a portion of the TMO and illustrates selected types, subtypes, and existential restrictions that hold between types. The TMO extends the basic types defined in the Basic Formal Ontology[7] (BFO) and uses relations from the Relation Ontology[8] (RO). Moreover it uses the Information Artifact Ontology[9] (IAO) as well.

Within MyHealthAvatar the TMO was extended by introducing terms from FOAF, ICO, DT, TIME and PLACE. A Figure showing some of the interconnections between those ontologies is shown in Figure 3. From the Figure we can see that a Person (FOAF:person) can participate in an event (TMO:processual_entity) and can have a specific risk (ICO:risk) about a disease (TMO:disease_progression). Events occur within a specific time period (OBO:temporal_region) and are located in a specific place (PLACE:place or obo:SpatialRegion). In addition a disease can have a specific treatment (DT:Treatment) and can have a specific symptom (TMO:Symptom). Finally a person can have several daily actions (TMO:daily_action) such as running, walking etc.

---

[7] http://www.ifomis.org/bfo

[8] http://www.obofoundry.org/ro/

[9] http://code.google.com/p/information-artifact-ontology/

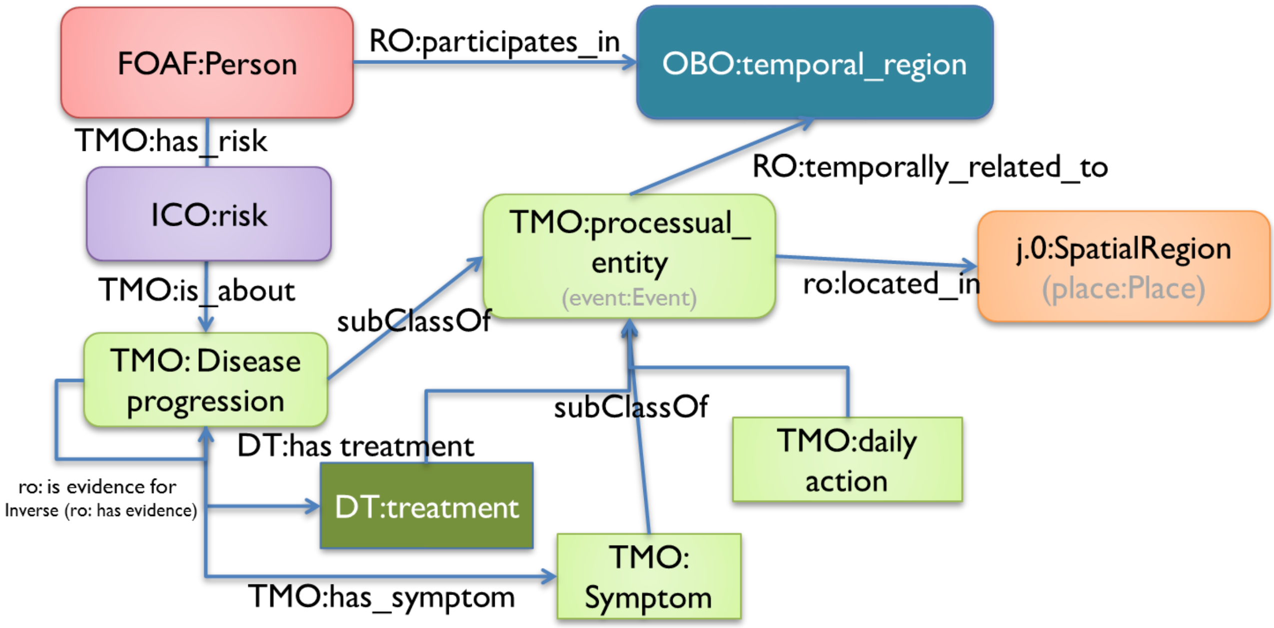


Figure 3. Interconnections between FOAF, OBO, ICO, TMO, DT, PLACE ontologies

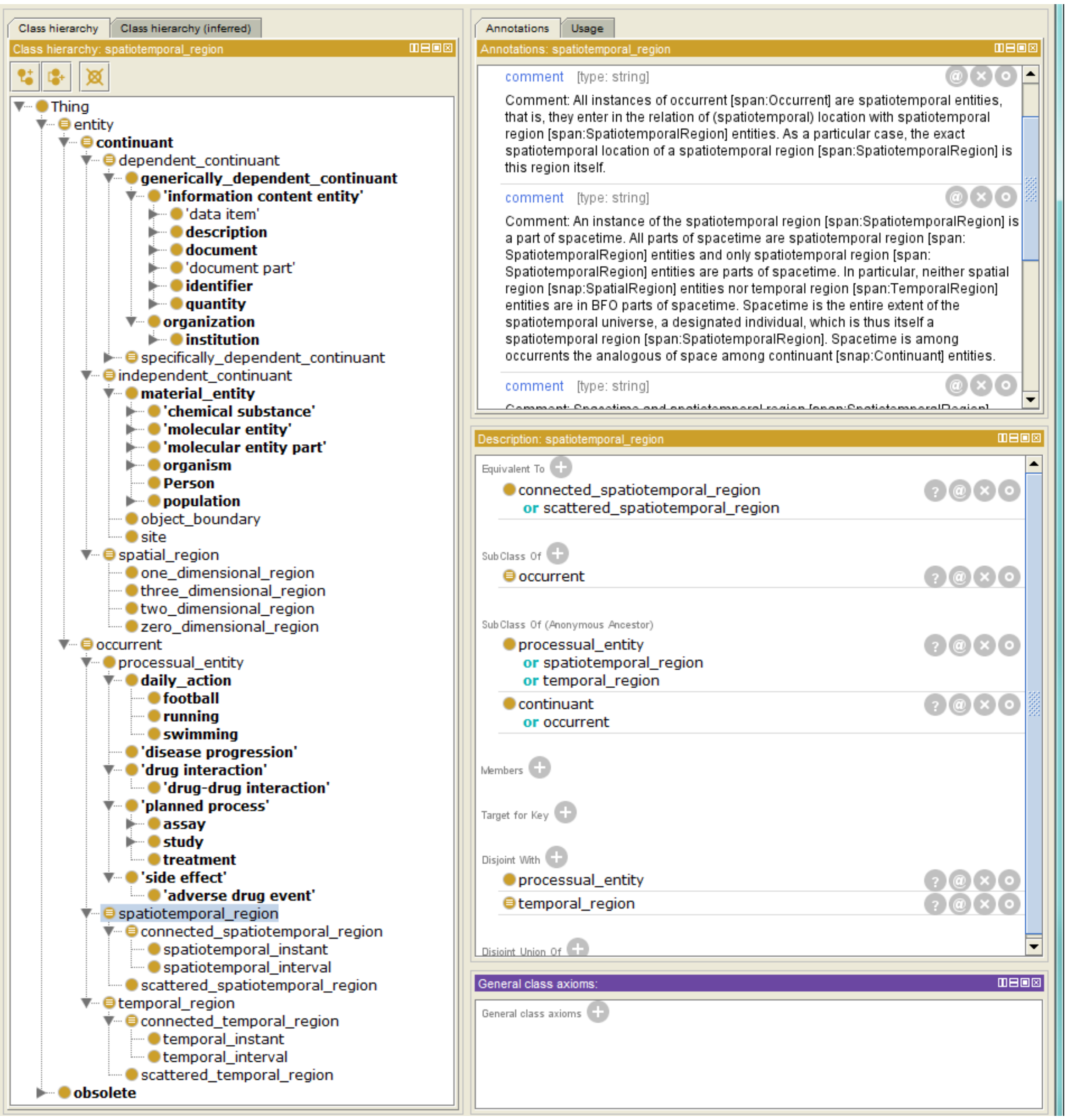
Bellow we provide the top four levels of the eTMO ontology with a short definition of those terms. In Figure 4 we provide a screenshot of the class hierarchy of the eTMO within the Protégé tool[10] that we used for ontology development.

- **Thing**
  - **entity**
    - **continuant**: An entity [bfo:Entity] that exists in full at any time in which it exists at all, persists through time while maintaining its identity and has no temporal parts.
      - **dependent_continuant:** A continuant [snap:Continuant] that is either dependent on one or other independent continuant [snap:IndependentContinuant] bearers or inheres in or is borne by other entities.
      - **independent_continuant:** A continuant [snap:Continuant] that is a bearer of quality [snap:Quality] and realizable entity [snap:RealizableEntity] entities, in which other entities inhere and which itself cannot inhere in anything.
      - **spatial_region:** A continuant [snap:Continuant] that is neither bearer of quality [snap:Quality] entities nor inheres in any other entities.
    - **occurent:** An entity [bfo:Entity] that has temporal parts and that happens, unfolds or develops through time. Sometimes also called perdurants.
      - **processual_entity:** An occurrent [span:Occurrent] that exists in time by occurring or happening, has temporal parts and always involves and depends on some entity.

---

[10] http://protege.stanford.edu/

- **spatiotemporal_region:** All instances of occurrent [span:Occurrent] are spatiotemporal entities, that is, they enter in the relation of (spatiotemporal) location with spatiotemporal region [span:SpatiotemporalRegion] entities. As a particular case, the exact spatiotemporal location of a spatiotemporal region [span:SpatiotemporalRegion] is this region itself.
- **temporal_region:** An instance of temporal region [span:TemporalRegion] is a part of time. All parts of time are temporal region [span:TemporalRegion] entities and only temporal region [span:TemporalRegion] entities are parts of time. Time is the entire extent of the temporal universe, a designated individual, which is thus a temporal region itself.



comment [type: string]

Comment: All instances of occurrent [span:Occurrent] are spatiotemporal entities, that is, they enter in the relation of (spatiotemporal) location with spatiotemporal region [span:SpatiotemporalRegion] entities. As a particular case, the exact spatiotemporal location of a spatiotemporal region [span:SpatiotemporalRegion] is this region itself.

comment [type: string]

Comment: An instance of the spatiotemporal region [span:SpatiotemporalRegion] is a part of spacetime. All parts of spacetime are spatiotemporal region [span: SpatiotemporalRegion] entities and only spatiotemporal region [span: SpatiotemporalRegion] entities are parts of spacetime. In particular, neither spatial region [snap:SpatialRegion] entities nor temporal region [span:TemporalRegion] entities are in BFO parts of spacetime. Spacetime is the entire extent of the spatiotemporal universe, a designated individual, which is thus itself a spatiotemporal region [span:SpatiotemporalRegion]. Spacetime is among occurrents the analogous of space among continuant [snap:Continuant] entities.

comment [type: string]

Equivalent To
connected_spatiotemporal_region
or scattered_spatiotemporal_region

SubClass Of
occurrent

SubClass Of (Anonymous Ancestor)
processual_entity
or spatiotemporal_region
or temporal_region

continuant
or occurrent

Members

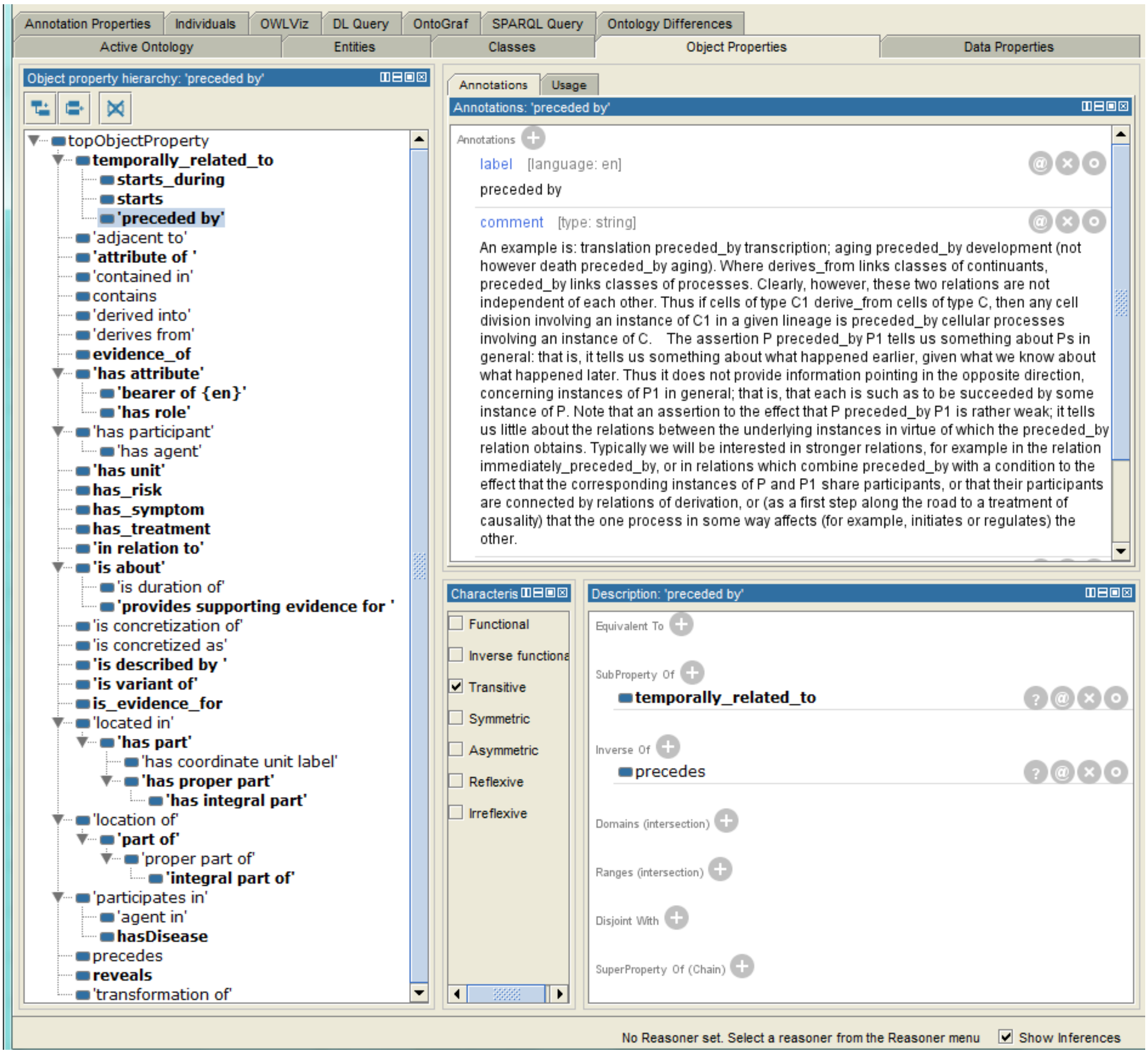
Target for Key

Disjoint With
processual_entity
temporal_region

Disjoint Union Of

Figure 4. Visualizing the class hierarchy of eTMO using Protégé

In addition in Figure 5 we visualize the object properties hierarchy of the eTMO using Protégé.



label [language: en]

preceded by

comment [type: string]

An example is: translation preceded_by transcription; aging preceded_by development (not however death preceded_by aging). Where derives_from links classes of continuants, preceded_by links classes of processes. Clearly, however, these two relations are not independent of each other. Thus if cells of type C1 derive_from cells of type C, then any cell division involving an instance of C1 in a given lineage is preceded_by cellular processes involving an instance of C. The assertion P preceded_by P1 tells us something about Ps in general: that is, it tells us something about what happened earlier, given what we know about what happened later. Thus it does not provide information pointing in the opposite direction, concerning instances of P1 in general; that is, that each is such as to be succeeded by some instance of P. Note that an assertion to the effect that P preceded_by P1 is rather weak; it tells us little about the relations between the underlying instances in virtue of which the preceded_by relation obtains. Typically we will be interested in stronger relations, for example in the relation immediately_preceded_by, or in relations which combine preceded_by with a condition to the effect that the corresponding instances of P and P1 share participants, or that their participants are connected by relations of derivation, or (as a first step along the road to a treatment of causality) that the one process in some way affects (for example, initiates or regulates) the other.

☐ Functional
☐ Inverse functiona
☑ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Equivalent To

SubProperty Of
temporally_related_to

Inverse Of
precedes

Domains (intersection)

Ranges (intersection)

Disjoint With

SuperProperty Of (Chain)

No Reasoner set. Select a reasoner from the Reasoner menu ☑ Show Inferences

Figure 5. Visualizing the object properties hierarchy of eTMO using Protégé

The interested reader is forwarded to the web site of the project[11] to download the latest version of the ontology.

3.3 *Equivalent Classes with Sub Ontologies*

Given the prevalence of the terms defined in the ontology and the desire to establish the eTMO as a global ontology, ~300 equivalence (using owl:equivalentClass) and subclass (rdfs:subClassof) mappings were created from eTMO to the various ontology modules. These mappings were manually identified and verified using the NCBO BioPortal[12] and UMLS[13].

---

[11] http://www.myhealthavatar.eu/

[12] http://bioportal.bioontology.org/

[13] http://www.nlm.nih.gov/research/umls/

The list of mappings is shown at the Appendix 2. Here we will describe just a few to demonstrate the idea. The following mappings allow us to use classes/properties from the NNEW ontology and to link it with the eTMO. We have to note that only the part of the NEEW ontology with high quality and relevant our use cases was mapped to the TMO and not the whole ontology.

Mapping:
NNEW:MereologicalPhenomena
rdfs:subClassOf
TMO:connected_spatiotemporal_region
Explanation: This mapping identifies the entire hierarchy bellow mereological phenomena from NNEW ontology as subclasses of the connected spatiotemporal region of the TMO.

Mapping:
NNEW:Measurement
rdfs:subClassOf
TMO:quantity
Explanation: As we can see in the eTMO ontology *TMO:quantity: A quantity is a quantitative measure of some feature* whereas in the NNEW we can see that *NNEW:Measurement: Measurement describes the general properties that can be measured.*

For ICD-10 we identified the following mapping with TMO:

Mapping:
ICD-10:Disease
owl:equivalence
TMO:Disease
Explanation: This means that the entire hierarchy bellow *ICD-10:Disease* can be placed below *TMO:Disease.*

For SYMP we identified the following mappings with eTMO:

Mapping:
SYMP:Symptom
owl:equivalence
TMO:Symptom
Explanation: The entire hierarchy bellow *SYMP:Symptom* can be placed below *TMO:Symptom.*

For FHHO we identified the following mappings with eTMO

Mapping:
FHHO:Diagnosis
owl:equivalence
TMO: Diagnosis
Explanation: Actually the entire hierarchy bellow *FHHO:Diagnosis* can be placed below *TMO: Diagnosis.*

3.4 Future Directions

Although it seems that the current state of MHA Semantic Core ontology is adequate to semantically describe the data of the different use cases we have to note that ontologies are living artifacts and subject to change. So we expect that the ontology will continue to be refined and evolve as more and more data providers and consumers appear and are implemented within the project. In D4.3

4 Semantic Integration Layer

The semantic integration layer consists of two main components shown in Figure 6. A Virtuoso RDF Triple Store[14] and the *exelixis* [9] [13] data integration system. The *exelixis* system semantically uplifts the available data through the MHA Semantic Core Ontology and the result is stored within the Virtuoso Triple Store for reasons of efficiency as we will explain in the following sections. Then the MHA platform is able to issue SPARQL queries through the SPARQL endpoint and retrieve the corresponding answers. To be able to translate data to their semantic representation a set of mappings is considered.

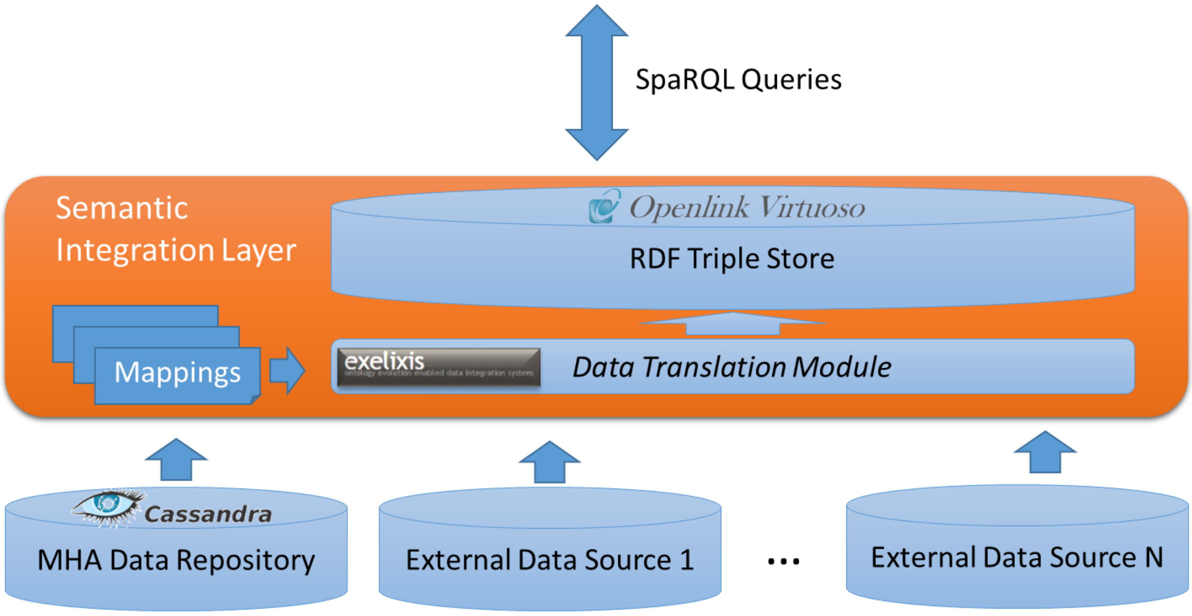


Figure 6. The Semantic Integration Layer of MyHealthAvatar platform

The architecture adopted is in a way based on the *command-query responsibility segregation* principle[15]. The core of that principle is that one used a different model to update information than the model one is using to read. Although the mainstream approach people use for interacting with an information system is to treat it as a CRUD datastore, as the needs become more sophisticated state of the art approaches steadily move away from that model since we may want to look at the information in a different way to the record store, perhaps collapsing mulptle records into one, or forming virual records by combining information for different places. On the update side we may find validation rules that only allow certain combinations of data to be stored, or may even infer data to be stored that are different from that we provide.

Specifically in our case, the NoSQL databases adopted to be the main data repository imposes strict restrictions on the way the information is stored and queried. The main restrictions are the following:

- Joins are now allowed
- You cannot project the value of a column without selecting first the key of the column.

---

[14] http://virtuoso.openlinksw.com/

[15] http://en.wikipedia.org/wiki/Command%E2%80%93query_separation

- Unlike the projection in a SQL SELECT, there is no guarantee that the results will contain all of the columns specified because Cassandra is schema-optional. An error does not occur if you request non-existent columns.

And although the *exelixis* system is able to reconcile those restrictions by enabling the combination of information in different tables and systems the aforementioned restrictions make the whole process really time consuming and non-efficient when large amounts of data are available. This is why we use *exelixis* system *offline* to integrate and reconcile the data and the result is then stored in a native RDF triple store.

Using the aforementioned architecture there are two main benefits. Firstly complexity is handled more efficiently since the two layers have different characteristics. The other main benefit is in handling high performance applications by separating the load from reads and writes allowing to scale each independently. In addition, different optimization strategies can be applied to the different components.

In the following sections we will describe in detail each one of the aforementioned components, i.e. the RDF Triple Store, the *exelixis* system and the mapping formalism used. Then we will show an example instantiation of the semantic infrastructure for the CHF Scenario.

4.1 RDF Triple Store

As the central RDF Triple store we considered Virtuoso[16] and OWLIM[17].

OWLIM, is a state-of-the-art RDF database management system developed by Ontotext. It is a native RDF engine, implemented in Java that supports the semantics of RDFS, OWL 2 RL and OWL 2 QL this makes it adequate for the reasoning choke points analysis that we undertake in this deliverable. According to their website, OWLIM offers great scalability features, and efficient loading and query evaluation times, even for large datasets. OWLIM is used in a large number of research projects and software tools. Although OWLIM came in three different versions, namely OWLIM-Lite, OWLIM-SE and OWLIM-Enterprise currently the platform was renamed to GraphDB and stopped offering support to the community edition. OWLIM will not be maintained any longer. Moreover, the free version simply only supports basic RDF query/reasoning and has specified that OWL based reasoning will be in commercial standard version e.g. sameAs inference.

Virtuoso on the other hand, is an innovative enterprise grade multi-model data server, developed by Openlink. The main innovation of Virtuoso is that it delivers a platform-agnostic solution for data management, access and integration. It supports the management of various types of data, including relational, RDF, XML, text documents and others. This way, the users can employ the hybrid server architecture of Virtuoso to get access to all these different types of data.

---

[16] http://virtuoso.openlinksw.com/

[17] http://www.ontotext.com/owlim

Both engines represent the current state-of-the-art in RDFS/OWL reasoning. Virtuoso uses backward reasoning whereas OWLIM uses forward reasoning. According to a recent report produced by LDBC project [14] OWLIM supports more reasoning constructs than Virtuoso and outperforms Virtuoso for reasoning intensive queries. However, OWLIM has stability problems. Virtuoso is more stable and is ideal for big data solutions, offering a lot of additional tools, libraries and it is more efficient in queries that do not require extensive reasoning. In addition the discontinued support of the OWLIM community edition lead us to the adoption of Virtuoso as the RDF triple store of the project although at the initial steps OWLIM was considered.

4.2 exelixis data integration engine

exelixis [13] [9] is a novel data integration engine that achieves query answering by accepting SPARQL queries. The queries are then rewritten according to the source schemata and forwarded to the sources to be answered. To achieve this rewriting the proper mappings should be established between the source schemata and the ontology. Instead of query rewriting there is also the possibility to materialize all generated information using the aforementioned mapping similarly to query rewrtiting. In both cases the results are provided as input to the RDF Triple Store where they are available for further queries.

In this Chapter we will explain formally how we changed the algorithms in order to word under the restrictions that are imposed by the NoSQL database. We 'll start by defining a a data integration system according to [15]:

Definition 4.1: *A data integration system* $\mathcal{I}$ *is a tuple of the form* ($\mathcal{T}$, $\mathcal{S}$, $\mathcal{M}$) *where* $\mathcal{T}$ *is the global schema,* $\mathcal{S}$ *is the source schema and* $\mathcal{M}$ *the mapping between the source schemata and the global schema.*

In our case, considering $\mathcal{T}$ we restrict ourselves to *valid RDF/S knowledge bases*, as most of the Semantic Web Schemas in nowadays are expressed in RDF/S. The representation of knowledge in RDF[18] is based on triples of the form *(subject predicate object)*. Assuming two disjoint and infinite sets $\mathcal{U}$, $\mathcal{L}$, denoting the URIs and literals respectively, $\mathcal{T} = \mathcal{U} \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L})$ is the set of all triples. An RDF Graph $\mathcal{V}$ is defined as a set of triples, i.e., $\mathcal{V} \subseteq \mathcal{T}$. In this section, we ignore unnamed resources, also called *blank nodes.* RDFS[19] introduces some built-in classes (class, property) which are used to determine the *type* of each resource. The typing mechanism allows us to concentrate on nodes of RDF graphs, rather than triples, which is closer to ontology curators' perception and useful for defining intuitive high-level changes.

RDFS provides also *inference semantics*, which is of two types, namely *structural inference* (provided mainly by the transitivity of subsumption relations) and *type inference* (provided by the typing system, e.g., if *p* is a property, the triple (*p*, *type*, *property*) can be inferred). The RDF Graph containing all triples that are either explicit or can be inferred from explicit triples in an RDF Graph $\mathcal{V}$ (using *both*

---

[18] http://www.w3.org/TR/REC-rdf-syntax/

[19] http://www.w3.org/TR/rdf-schema/

types of inference), is called the *closure* of $\mathcal{V}$ and is denoted by *Cl(*$\mathcal{V}$*)*. An *RDF/S Knowledge Base (RDF/S KB)* $\mathcal{V}$ is an RDF Graph which is closed with respect to *type inference*, i.e., it contains all the triples that can be inferred from $\mathcal{V}$ using type inference. Moreover, we assume that the RDF/S Knowledge Bases are *valid*. The notion of validity has been described in various fragments of the RDFS language. The validity constraints that we consider in this work concern the *type uniqueness,* i.e., that each resource has a unique type, the *acyclicity* of the *subClassOf* and *subPropertyOf* relations and that the subject and object of the instance of some property should be correctly classified under the domain and range of the property, respectively. For a full list of the validity constraints we adopt see [16]. Those constraints are enforced in order to avoid cycles when trying to consider constraints coming from the ontology.

Consider for example the following ontology, shown on top of Figure 7 that describes persons and their contact points. Moreover consider two column families in two Cassandra databases that store information about insured persons and patients shown also in Figure 7.

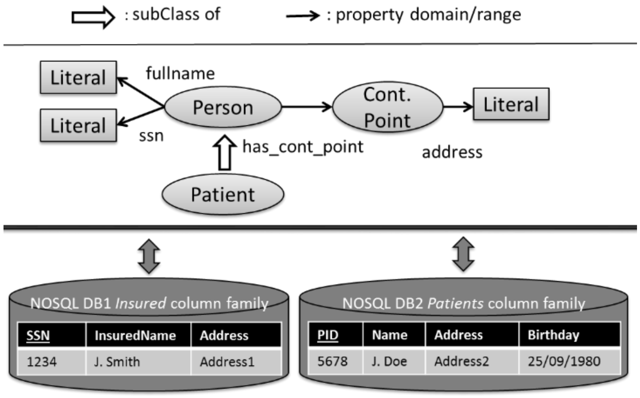


Figure 7. An example ontology & two NoSQL DBs

We have to note however, that the NoSQL database can be replaced by traditional relational DBMS systems, CSV files etc.

4.2.1 Semantics of $\mathcal{I}$

To define semantics for $\mathcal{I}$ we start by considering a *local database* for ($\mathcal{T}$, $\mathcal{S}$, $\mathcal{M}$), i.e., a database $\mathcal{D}$ that conforms to the *local sources of* $\mathcal{S}$.

Now, based on $\mathcal{D}$, we shall specify the information content of the global schema $\mathcal{T}$. We call a database for $\mathcal{T}$ a *global database.* However, since those global databases might be many, we are interested in the *legal global database.*

Definition 4.2 (Legal Global Database): *A global database* $\mathcal{G}$ *for* ($\mathcal{T}$, $\mathcal{S}$, $\mathcal{M}$) *is said to be legal with respect to* $\mathcal{D}$*, if*

- $\mathcal{G}$ *satisfies all subClass/subProperty constraints of* $\mathcal{T}$
- $\mathcal{G}$ *satisfies the mapping* $\mathcal{M}$ *with respect to* $\mathcal{D}$*.*

When we refer to the notion of $\mathcal{G}$ satisfying $\mathcal{M}$, we mean $\mathcal{G}$ satisfying the GLAV mappings produced from $\mathcal{M}$. In relational databases a GLAV mapping associates a table from the target schema to a query over the source schemata. So, in our case a GLAV mapping associates to a query q_T in $\mathcal{T}$ a CQL query q_S over $\mathcal{S}$, i.e., $q_T \rightarrow q_S$.

Definition 4.3: *A database G satisfies the mappings* $q_T \rightarrow q_s$ *with respect to* $\mathfrak{D}$ *if* $q_T \supseteq q_S^{\mathfrak{D}}$ *where* $q_S^{\mathfrak{D}}$ *is the result of evaluating the query* q_S *over* $\mathfrak{D}$.

Note that since Cassandra does not allow joins $q_S^{\mathfrak{D}}$ refers always to a single table. Moreover, in Cassandra in order to be able to select a field you always have to include also the key of that table. So in the mappings whenever a table is being selected special attention should be given on selecting also the primary key of the table. For example, the sequence of the GLAV mappings that corresponds to our sequence of changes is:

NoSQL DB1 Insured Column Family:

m_1: *rdfs:type(person_{SSN}, Person)* →

select SSN from Insured

m_2: *fullname(person_{SSN}, {InsuredName})* →

select SSN, InsuredName from Insured

m_3: *address(cont_pont_{SSN}_{Address}, {Address})* →

select SSN, Address from Patients

m_4:*has_cont_point(person_{SSN}, cont_pont_{SSN}_{Address})* →

select SSN, address from Patients

NoSQL DB2 Insured Column Family:

m_5: *rdfs:type(patient_{PID}, Patient)* →

select PID from Patients

m_6: *fullname(patient_{PID}, {Name})* →

select PID, Name from Insured

m_7: *address(cont_pont_{PID}_{Address}, {Address})* →

select PID, Address from Patients

m_8:*has_cont_point(patient_{PID}, cont_pont_{PID}_{address})* →

select PID, address from Patients

4.2.2 Query Processing

Queries to $\mathcal{I}$ are posed in terms of the global schema $\mathcal{T}$. For querying, we adopt a core subset of the SPARQL language corresponding to union of conjunctive queries[20]. We chose SPARQL since it is currently the standard query language for the semantic web and has become an official W3C recommendation. Essentially, SPARQL is a graph-matching language. Given a data source, a query consists of a pattern which is matched against, and the values obtained from this matching are processed to give the answer. A SPARQL query consists of three parts. The *pattern matching* part, which includes several features of pattern matching of graphs, like optional parts, union of patterns, nesting, filtering (or restricting) values of possible matchings. The *solution modifiers*, which once the output of the pattern has been computed (in the form of a table of values of variables), allows to modify these values applying classical operators like projection, distinct, order, limit, and offset. Finally, the *output* of a SPARQL query can be of different types: yes/no answers selections of values of the variables which match the patterns, construction of new triples from these values, and descriptions of resources.

In order to avoid ambiguities in parsing, we present the syntax of SPARQL graph patterns in a more traditional algebraic way, using the binary operators *UNION* (U), *AND* ($\wedge$), *OPT*, and *FILTER*. Assuming the existence of an infinite set of variables *Var* disjoint from $\mathcal{U}$, $\mathcal{L}$, a SPARQL graph pattern expression is defined recursively as follows:

- A tuple from $(\mathcal{U} \cup \mathcal{L} \cup Var) \times (L \cup Var) \times (\mathcal{U} \cup \mathcal{L} \cup Var)$ is a graph pattern (a triple pattern)
- If P_1 and P_2 are graph patterns, then expressions ($P_1 \wedge P_2$), (P_1 OPT P_2) and (P_1 U P_2) are graph patterns.
- If P is a graph pattern and R is a SPARQL built-in condition, then the expression (P *FILTER* R) is a graph pattern.

A SPARQL built-in condition is constructed using elements of the set $(\mathcal{U} \cup \mathcal{L} \cup Var)$ and constants, logical connectives, inequality symbols, the equality symbol etc.

In this paper we adopt a streamlined version of the core fragment of SPARQL as presented in [17] with precise syntax and semantics. Moreover, we restrict even more the specific fragment of SPARQL since we do not consider *OPT* and *FILTER* operators which we leave for future work. The remaining SPARQL fragment corresponds to union of conjunctive queries (this will not hold if we allow *OPT* and *FILTER* operations). Moreover, the application of the solution modifiers and the output is performed after the evaluation of the query. So, without loss of generality we will not present them in this paper. Continuing our example, assume that we would like to know the "*ssn*" and "*fullname*" of all persons stored on our DBs and their corresponding "*address*". The SPARQL query, formulated using the latter version of our example ontology is:

```
q1: select ?SSN ?NAME ?ADDRESS where {
        ?X type Person.
        ?X ssn ?SSN.
        ?X  fullname ?NAME.
        ?X has_cont_point ?Y.
```

---

[20] http://www.w3.org/TR/rdf-sparql-query/

?Y type Cont.Point.
?Y address ?ADDRESS}

Using the semantics from [17] the algebraic representation of q_1 is equivalent to:

q_1: $\pi_{?SSN,?NAME,?ADDRESS}$ (
(?X, type, Person) $\wedge$
(?X, ssn, ?SSN) $\wedge$
(?X, fullname, ?NAME) $\wedge$
(?X, has_cont_point, ?Y) $\wedge$
(?Y, type, Cont.Point) $\wedge$
(?Y, address, ?ADDRESS))

Now we define what constitutes an answer to a query over *O*. We will adopt the notion of *certain answers* [18] [19].

Definition 4.4 (Certain Answers): *Given a global database $\mathcal{D}$ for $\mathcal{I}$, the answer $q^{\mathcal{I},\mathcal{D}}$ to a query q with respect to $\mathcal{I}$ and $\mathcal{D}$, is the set of tuples t such that $t \in q^T$ for every global database T that is legal for $\mathcal{I}$ with respect to $\mathcal{D}$, i.e., such that t is an answer to q over every database T that is legal for $\mathcal{I}$ with respect to $\mathcal{D}$. The set $q^{\mathcal{I},\mathcal{D}}$ is called the set of certain answers to q with respect to I and $\mathcal{D}$.*

Although certain answers are mostly used in local-as-view data integration systems, still in our case the ontological constraints may introduce incompleteness, so certain answers have to be adopted [20]. In fact, it has been shown [22], that computing certain answers to union of conjunctive queries over a set of legal global databases, corresponds to evaluating the query over a special database called *canonical* which represents all possible global databases legal for the data integration system and which may be infinite in general. In order to define the *canonical database* we first should define the *retrieved database.*

Definition 4.5 (Retrieved database): *If $\mathcal{D}$ is a global database for $\mathcal{I}$, then the retrieved total database ret($\mathcal{I}$, $\mathcal{D}$) is the global database obtained by computing and evaluating, for every element of $\mathcal{T}$ the query associated to it by M over the global database $\mathcal{D}$.*

The query associated to an element of $\mathcal{T}$ is actually the GLAV mapping.

Definition 4.6 (Canonical database): *If $\mathcal{D}$ is a global database for $\mathcal{I}$, then the canonical global database can($\mathcal{I}$, $\mathcal{D}$) is the set of retrieved global databases ret($\mathcal{I}$, $\mathcal{D}$) that do not violate any constraint in $\mathcal{T}$.*

Now, instead of trying to construct the canonical database and then evaluate the query, another approach is to transform the original query *q* into a new query $exp^T(q)$ over the $\mathcal{T}$, (which is called the *expansion* of *q* w.r.t. $\mathcal{T}$) such that the answer to $exp^T(q)$ over the *retrieved global database* is equal to the answer to *q* over the *canonical database* [21].

In this step, the query is expanded to take into account the constraints coming from the ontology. Query expansion amounts to rewriting the query *q* posed to the ontology version into a new query q_p, so that all the knowledge about the constraints in ontology has been "compiled" into q_p.

Definition 4.7 (Query Expansion): *Let $\mathcal{I}$ a Data Integration system and let q be a query over $\mathcal{T}$. Then q_p is called an expanded query of q, i.e. $exp^T(q)$, w.r.t. $\mathcal{I}$ if, for every global database $\mathcal{D}$, $q^{\mathcal{I},\mathcal{D}} = q_p{}^{ret(\mathcal{I},\mathcal{D})}$.*

Query expansion is also known as perfect rewriting. Algorithms for computing the query expansion/perfect rewriting of a query *q* w.r.t to a schema, have been presented in [18], [20] and mainly use chase/backchase algorithms [22]. In our work, we use the *QuOnto* system [20] in order to produce the query expansion of our initial query. Query expansion is in our case PTIME in the size of ontology and NP in the size of query. For more general classes of logic it is complete for PSPACE and 2EXPTIME as proved in [18].

Continuing our example if we expand q_1 we get q_2:

$$\begin{array}{lcl}
\mathbf{q_2}: \pi_{?SSN,?NAME,?ADDRESS}(& & \pi_{?SSN,?NAME,?ADDRESS}(\\
(?X, type, Person) \wedge & & (?X, type, \mathbf{Patient}) \wedge \\
(?X, ssn, ?SSN) \wedge & \cup & (?X, ssn, ?SSN) \wedge \\
(?X, fullname, ?NAME) \wedge & & (?X, fullname, ?NAME) \wedge \\
(?X, has\_cont\_point, Y) \wedge & & (?X, has\_cont\_point, ?Y) \wedge \\
(?Y, type, Cont.Point) \wedge & & (?Y, type, Cont.Point) \wedge \\
(?Y, address, ?ADDRESS)) & & (?Y, address, ?ADDRESS))
\end{array}$$

This is produced by considering the transitive constraint of the *subClass* relation among the classes "*Person"* and *"Patient"*.

Now instead of evaluating $exp^T(q)$ over the retrieved total database, we transform it to a new query, named unfolding, i.e. *unfold*($exp^T(q)$). This is done, as already discussed, in order to avoid the construction of the retrieved total database.

When the retrieved total database is produced by GLAV mappings, as in our case, query rewriting is simply performed using *unfolding* [20]. This is a standard step in data integration which trivially terminates and it is proved that it preserves soundness and completeness [20] [21].

Now, we can state the main result of this section.

Theorem 1 (Soundness and Completeness): *Let $\mathcal{I}$ an Evolving Data Integration system, q a query posed to $\mathcal{I}$, $\mathcal{D}$ a global database for $\mathcal{I}$ such that $\mathcal{I}$ is consistent w.r.t. $\mathcal{D}$, and t a tuple of constants of the same arity as q. Then $t \in q^{\mathcal{I},\mathcal{D}}$ if and only if $t \in [unfold(exp(q))]^{\mathcal{D}}$.*

Proof: By soundness and completeness of unfolding $t \in [unfold\ (exp(q))]^{\mathfrak{D}}$ if and only if $t \in exp_{Om}(q)^{ret(\mathcal{I},\mathcal{D})}$. Now by the soundness of the query expansion step we have that $t \in exp_{Om}(q)^{ret(\mathcal{I},\mathfrak{D})}$ if and only if $t \in q^{can(\mathcal{I},\mathfrak{D})}$. By the canonical database $t \in q^{can(\mathcal{I},\mathfrak{D})}$ if and only $t \in q^{\mathcal{I},\mathfrak{D}}$. This proves the claim▪

Continuing our example we will show how the valid rewriting of q_2 is constructed using unfolding. In unfolding each subgoal in the query is replaced with its definition in the mapping. So, the following query is produced:

SELECT SSN, Insured-Name as Name, Address FROM Insured	U	*SELECT PID as SSN, Name, Address FROM Patients*

The first sub-query is forwarded to the first database to be answered whereas the second sub-query is forwarded to the second database. The results are unioned then and returned to the user.

4.2.3 Implementation

One of the assumptions of the data integration systems is that all data can be viewed as belonging to the same "federated" database. Then query answering mechanisms reformulate input query to sub-queries which are being sent to this "federated" database to be answered. This "federated" database has also the task to union the final results to perform the join operations to apply conditional selections etc. However, in our case we cannot rely on such an assumption. This is due to the fact that Cassandra does not allow union and join operations. These operations should be implemented by our system. Moreover, in Cassandra, for a given partition key, the clustering columns induce an ordering of rows and relations on them is restricted to the relations that allow to select a contiguous (for the ordering) set of rows. So condition on no key columns are not possible and should also be implemented by our system. All these functionalities are implemented by our system in the Splitter/Combiner component shown in Figure 8. The *exelixis* system can either produce the entire canonical database or specific query results in order to be published at the Virtuoso triple store or version. It is implemented using JAVA for the algorithms and HTML/JQuery for the presentation layer and can also be found online through a nice front-end[21].

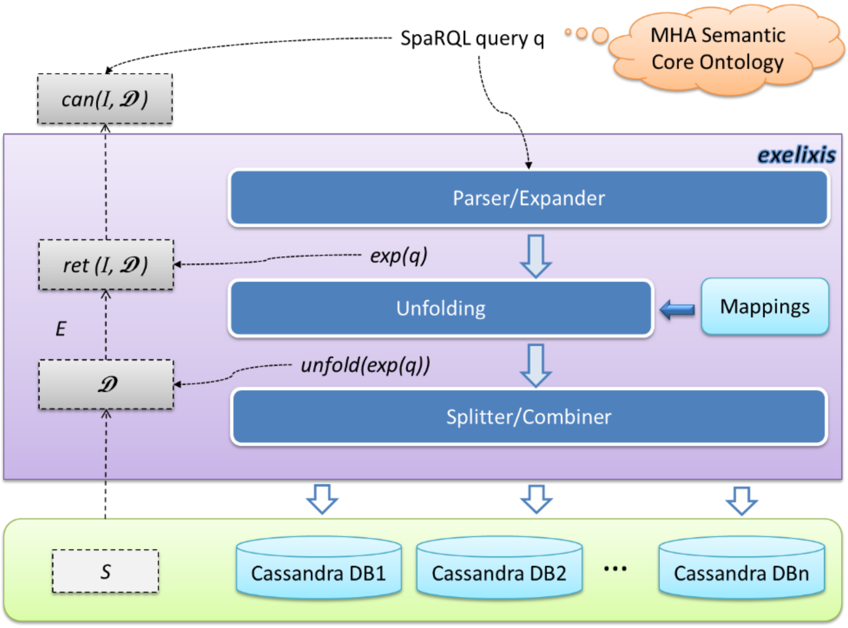


Figure 8. exelixis high-level architecture

---

[21] http://139.91.183.29:8080/exelixis/

4.3 X3ML Mapping Workflow

Independent of the specific approach selected for data integration and transformation mappings are required in order to link the individual data sources to the global schema - the MHA Semantic Core Ontology in our case. The definition of those mappings is a time-consuming, labor-intensive and error-prone activity [43].

Although the latest years mapping formalisms is an active field of research so far there is no no standard model to allow mapping of data sources, the technologies used are too complex to be used by the domain experts and the whole workflow is not well-defined. For that purpose we present a mapping workflow that manages the core processes needed to create, maintain and manage mapping relationships between different data sources over the long term, with high level of quality control. At the center of this workflow lies a novel matching formalism named X3ML, which is human readable and generic enough to cover most of the data models used nowadays. This model carefully distinguishes between information from the domain experts who knows and provide the data and that created by the IT technicians, who actually implement data translation and integration solutions, and serves as an interface between both. This solves a major bottle neck in current information aggregation environments in terms of human resources and quality control.

4.3.1 Existing Mapping formalisms Review and Assessment

In the latest years an active field of research is mapping relational databases (RDB) to RDF. This happens as the majority of data currently published on the web are still stored in relational databases with local schemata and local identifiers. Bridging the conceptual gap between the relational model and RDF is a key to make the data available as linked data, materializing the vision of Semantic Web [24].

One of these solutions is the Direct Mapping [25] which maps automatically relational tables to classes and the attributes of tables to properties in an RDF vocabulary maintaining local schema semantics The URIs of the instances and the classes are automatically generated based on the RDB schema and data. One implementation that uses Direct Mapping is SquirrelRDF [26]. The focus of these solutions however is on mapping discovery and URI generation is tied to the implementation of the systems. This limits the ways to use URIs and makes it unmanageable for complex structures and information integration.

Besides the aforementioned approach there are also XML-based solutions. For example, D2R MAP [27] is a declarative language to describe mappings between relational databases and OWL/RDFS ontologies. The specific approach is extended by eD2R [28] in order to cover databases that are not in first normal form and by D2RQ [29] which is formally defined by an RDFS schema. The mappings are based on SQL queries that extract records from the RDB and transformation functions that can be applied to the extracted values. Another XML-based approach focusing on expressivity is R2O [30] which is able to cope with complex mapping cases where one model is richer than the other. For that sake, Virtuoso RDF Views [31] is also created exposing RDBs using a declarative Meta Schema Language for defining the mapping of SQL data to RDF vocabularies. Triplify [32] on the other hand maps HTTP-URI requests onto RDB queries and translates the resulting relations into RDF statements. Finally

R2RML [33] is a mapping language proposed by W3C in order to standardize RDB to RDF mappings. There exist already several implementations [34] and is lately extended by RML [35] to simultaneously support different mapping sources into RDF. However RML misses the actual implementation of the mappings it defines. Moreover, most of the aforementioned approaches lack general conditions when mapping is dependent on particular data values in data structures and they limit the ways that the URIs can be constructed making it impossible to select other forms of URIs to be generated. The latter may frequently be required in order to adapt transformed data to different targets, such as LoD, or to look-up the identifiers used in the target system.

Similar to the previous solutions but mapping to more expressive data models is the Relational.OWL [36] which is a representation format for mapping relational models to OWL full ontologies. In all aforementioned approaches however, there is no distinction between IT and domain experts and an IT expert is assumed to be responsible for the entire process. This is reflected in the complexity and lack of user-friendliness of the employed mapping languages, even if some easy parts of the mapping are done graphically. IT experts on the other side will not easily understand the domain semantics they are dealing with. URI generation is still based on a sort of unique key generation from data, as in RDBMS

Besides approaches that try to map relational data mostly to the RDF models there are other approaches which try to map CSV files to RDF. For example XLWrap's mapping language [37] provides conversions from csv and spreadsheets to RDF data model, Mapping Master's M2 [38] converts data from spreadsheets into owl statements and Vertere [39] is a conversion tool based on a templating mechanism. Each line results in one or more RDF resources, while each column value can result in one or more triples about this resource. TaRQL [40] is used also for the same purpose. Other tools, generated to provide mappings from XML to RDF lead to mappings in the syntactic level rather on the semantic level and fail to provide solutions applicable to broader domains. Tools in this category include tools based on XSLT (Krextor [41], AstroGrid-D [42]), tools based on XPATH (Tripliser [43]) and XQUERY (XSPARQL [44]) and tools that are based on algorithms (GRDDL [45]) that provide links between data and RDF. Finally there are existing tools that provide mappings from several formats to RDF. Tools in this category include Datalift [46], The DataTank [47], OpenRefine [48], RDFizers [49], Virtuoso Sponger [50]. A fundamental problem with using highly expressive languages such as XSLT is the fact, that even the programmer himself has difficulties to understand the mapping logic from reading code written by a colleague. The only way to verify the mapping is testing its output behaviour.

Other approaches exploit mapping technologies to publish their data as linked data. For example the Smithsonian American Art Museum used KARMA [51] to publish their data as linked data, a tool trying to automate the mapping process allowing then the users to adjust the generated mappings. However, still there is no clear description on the work of the domain and the IT experts which perplexes the whole workflow. A similar work is the SIP creator [52], created for Europeana in 2009 to bring together more than a hundred fifty different sources. Whereas it only dealt with flat formats at the beginning it was later expanded to handle hierarchical formats as output, however only in XML. Motivated by the goal of transparency, the mapping file format created by the SIP-Creator was seen to come closest to the structure that was required for generic mapping, since it appeared in an easy-to-comprehend, human readable XML with the source and target paths clearly identified. However it combined interactive schema matching with resolving not formally structured elements (they call it “syntax normalization”), (e.g. putting life dates in parentheses behind a person name for better identification) on the basis of the rich Groovy [53] programming language. Groovy however, as a programming language cannot be used or understood by a domain expert.

All these different approaches prove that there is no standard model to allow mapping of data sources other than relational, the technologies used are too complex to be used by the domain experts and the whole workflow is not well-defined. More specifically our review on the aforementioned systems and techniques led to the identification of the following limitations:

Lack of a standard mapping model. There is no standard model to allow mapping of data sources other than relational (R2ML). However, even in R2ML, schema matching information again is not represented in a way a domain expert could understand and verify. In practice data are published in different source and target formats which, in turn, requires a more generic approach.

Mapping model tied to the implementations. In most of the cases, the mapping rules are tied to the implementation of the data transformation engine which prevents their extraction and reuse across different implementations. In particular, they do not allow harmonization of mappings from different providers to the same target by the domain experts. Most of these implementations are monolithic, i.e. each implementation has developed another subset of possible functionalities without a chance of integration.

Tied to particular data models: Moreover, the current solutions they do not allow for switching between XML, RDF and RDBMS support on the source and the target side. Besides RML, most of them allow only one source to be mapped at once or supporting only mapping from a certain source format.

Lack of flexibility: Most of the current solutions do not support incremental changes of source and target schema and URI generation policies. Even R2ML and its derivatives builds complex target structures by crafting URIs that fit and they cannot create UUIDs. This becomes unmanageable for deeper structures and limits the ways to use URIs. Similar flexibility and methods is required for generating datatype values, such as dates, geo-coordinates and numbers. Both have in common that they create representation of entity instances, in contrast to property instances and classification. We therefore talk in the following about “instance generation” for both.

Lack of sustainability: More than this, they do not foresee collaborative work of experts with different roles on the mapping process and the do not provide means to maintain an automated communication for data cleaning with the provider, to easily do incremental mapping following changes in source or target schema, use of URIs and use of terminology the mapping depends on.

4.3.2 A novel workflow for mapping process

To tackle all aforementioned limitation we designed and extensively used a new workflow and reference model. The basic design principles were the following:

I. **Simplicity**. One important way to achieve simplicity and clarity is by carefully naming things so that their meaning is as obvious as possible to the naked eye.

II. **Transparency.** The most important feature of the mapping workflow and the corresponding mapping models is its general application to mapping creation and execution and hopefully its longevity. People must be able to easily understand how it works. The cleaner the core design of this engine and X3ML language, and the clearer its documentation, the more readily it will get traction and become the basis for future mappings.

III. **Collaborative Mapping Memory.** The X3ML descriptions must lend themselves to be stored and handled by collaborative tools, as well as potentially be written by hand. This was the motivation for choosing a simple syntax in XML, and one which does not depend on implicit knowledge.

IV. **Separation between *Schema Matching* and *Mapping definition*.** Schema matching needs to be separated from the concern of generating proper URIs, labels and datatype values so that different expertise can be applied to these two very different responsibilities. The instance generation expert must ensure that the generated URIs match certain criteria such as consistency and uniqueness or Linked open Data qualities and that datatype encodings follow the target specifications such as date formats, while the Schema experts only need to concern themselves with the proper semantic interpretation of the source in terms of target concepts.

V. **Re-use of Standards and Technologies.** The best way to build a new software module is to carefully choose its dependencies, and keeping them as small as possible. Building on top of proven technologies is the quickest way to a reliable result.

In our approach the mapping definition process consists of two separate and well-defined steps shown in Figure 9.

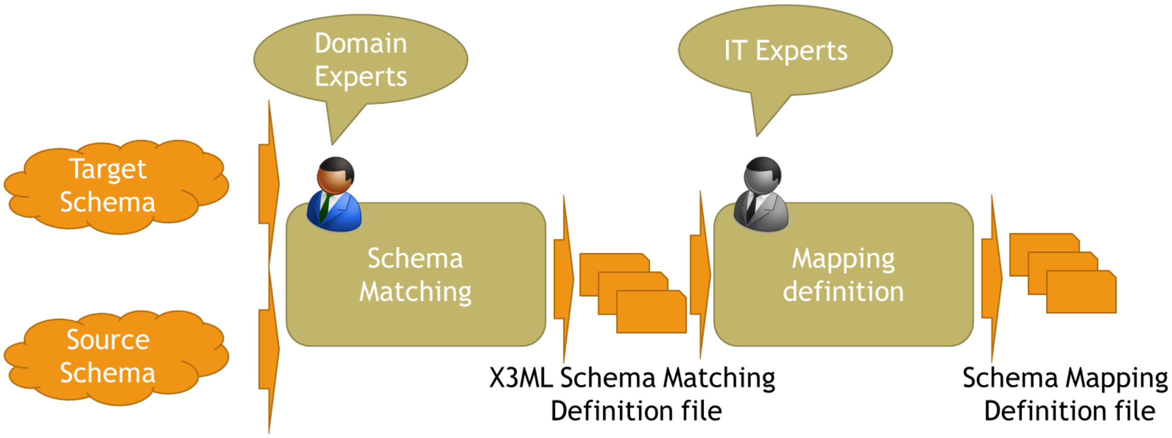


Figure 9. The mapping workflow

Bellow we explain in detail each one of the three steps:

I. **Schema matching**: Source and target schema experts (a.k.a the domain experts) define a schema matching which is documented in a *schema matching definition file*. This file should be human and machine readable and is the ultimate communication mean on the semantic correctness of the mapping. In order to do so, all source schema elements must be well understood and mapped to target schema paths. To achieve that, adequate tools could be used to enhance user understanding on the models through navigation and visualization, to suggest automatically mappings or to visualize them. All related tools should take into account the need for incremental mappings and guide the user through relevant changes. In this document we will not focus on these tools but only on the core layer.

II. **Mapping definition**: In this step the actual mappings are defined and generated based on the input of the first step. In this step only IT experts are involved and domain experts have no interest or knowledge about it. The translation from schema matching to schema matching can be also aided by proper tools however, we do not foresee that can be fully automated.

Below we describe in more detail our X3ML matching definition file and then we focus on the mapping definition file as well.

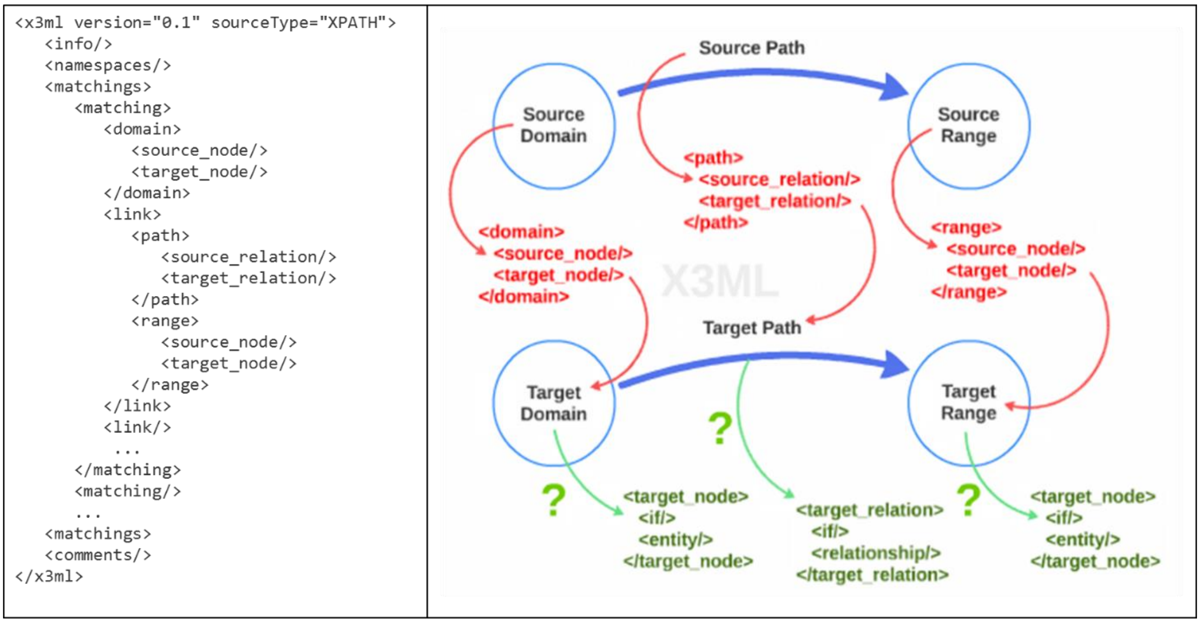
4.3.3 Schema Matching

The first step of the schema matching process is to identify the part of the source and the target schemata that are going to be used. It is a common case for database managers to offer only a small portion of the whole schema to take part in data integration since confidential parts of data or data of internal use only may exist. So, in the beginning, the domain expert identifies information of interest and then the IT expert finds the corresponding tables/files that are going to be shared. Then, in order to produce a simple and expressive language the basic matching construct we use is *proposition*. The idea is that we convert the target and the source schemata into elementary propositions and we match those propositions to each other. Every model can be represented as class-role-class triples (c-r-c) i.e. a set of propositions, using nodes and links. The conversion into propositions is a trivial syntactic procedure that is done automatically by a proper tool. Note however that attributes in E-R or XML elements may play a double role of role and class. For instance, a field “ID” may represent “has ID” and being of type “ID”, or a field “creator” may represent “has creator” and being of type “creator” – meaning “agent”.

Then the next step is to match schema entities. The inputs of this sub-process are the target and the source schemata and the output is a *schema matching definition file*. The person typically capable to carry out this step is the domain expert, which is possibly assisted by an IT expert in the beginning. He has to specify the matchings of each proposition of the source schema to the proper proposition from the target schema, preferably by using an appropriate graphical tool.

Such a tool should be available to make the whole process more intuitive, to ensure consistency of the statements with the target and source schemas and to help the experts with appropriate visualizations for the result of those statements. Moreover, such a tool may employ several algorithms and a background knowledge base to perform schema matching suggestions in order to aid the domain expert in his difficult work. The main difference here is that schema matching suggestion (or matching discovery) should be based on entire propositions in context and not on isolated classes as traditionally done.

The main arguments for matching triples instead of single classes/fields are: a) the simplicity of the approach and b) that problems occur when trying to match single classes/fields. Those problems appear since usually we can match a field to several partially overlapping classes (-depending on data instance values) from the target schema and we cannot make a clear choice. Matching discovery tools, as long as the matching rules are based on simple class correspondences, have real difficulties to identify to which level of abstraction a correspondence needs to be found. Even though it is easy to make a schema with only one class, such as “*Thing*“, it is impossible to make a schema without relationships. Since there is no proposition without a relationship we focus on the relationships as basic elements to define the matching correspondences. A first step to schema matching can be based exclusively on relationships, but not exclusively on classes. As long as the target is RDF, target propositions are context free and globally identified, and therefore can be merged into huge knowledge pools, ignoring the boundaries of the source documents they were derived from.



```
<x3ml version="0.1" sourceType="XPATH">
   <info/>
   <namespaces/>
   <matchings>
      <matching>
         <domain>
            <source_node/>
            <target_node/>
         </domain>
         <link>
            <path>
               <source_relation/>
               <target_relation/>
            </path>
            <range>
               <source_node/>
               <target_node/>
            </range>
         </link>
         <link/>
          ...
      </matching>
      <matching/>
      ...
   <matchings>
   <comments/>
</x3ml>
```

Figure 10. The structure of an X3ML mapping

The basic matching scheme and the corresponding XML structure is shown in Figure 10. and is called X3ML. Each c-r-c of the source schema is matched individually to the global schema. Each class-role-class can be seen as self-explanatory, context independent proposition. Having that in mind we need to define:

- The matching between the Source Domain and the Target Domain.
- The matching between the Source Range and the Target Range.
- The proper Source Path.
- The proper Target Path.
- The matching between Source Path and Target Path

Below we analyse in detail the aforementioned format of this XML structure.

4.3.3.1 The info and *comments* elements

Since X3ML is intended to bridge the gap between human authors and machines, it has to allow textual comments to be placed in the matching specification. This is intended for describing alternatives to be discussed between experts and for harmonizing parallel mappings between providers. For this purpose the *info* and the *comment* elements have been defined in the matching specification. These blocks contain relevant information for humans to understand the specific matchings and can also contain useful provenance information recorded automatically by the tools used to manage the X3ML files, such as the date of creation or the author of the matching file.

4.3.3.2 *The matching element*

Each matching element consists of a *domain* element and a number of *links*. It is common case, to have a single domain matching and some range matching, so by using the specific matching format the single domain matching doesn't have to be declared again. This is an ergonomic choice good for tree-dominated source schemata or sets of Relational tables, which helps user orientation. It further provides an intuitive default local scope to define that the same instance of a class in a matching rule appears as domain value in multiple target propositions.

The *domain* element is used to specify the matchings between a source (i.e. the *source_node*) entity (table, class non-leaf element) that can be regarded as domain of a source proposition and an equivalent target (i.e. the *target_node*) entity. The *source_node* provides information on how to navigate to the source record and in case of XML it is an XPATH expression. On the other hand, the *targed_node* might contains some *if* conditions the equivalence depends on and an *entity* element. The *entity* element leads to the generation of resource URIs or datatype values for the output graph.

Inside the *link* element there is a *path* element. It allows for matching a *source_relation* from the before defined source domain to a *target_relation* to the before defined target domain. The path element must be followed by a *range* element, which is used to map the source and the target entities that are the equivalent range of the respective paths. The *target_relation* might contain conditions as well. A source / target range pair may reappear as a subsequent domain in an X3ML matching.

The conditional expressions in X3ML can check for existence and equality of values and the corresponding syntax is shown in Figure 11. Moreover, they can be combined into boolean expressions. Further, testing for a "narrower term of" relation is foreseen, such "if X is narrower term of Physical Object".

```
<if>

   <exists>[xpath]</exists>

</if>

<if><not>

   <if><exists>[xpath]</exists></if>

</not></if>

<if>

   <equals value="[value-for-comparison]">[xpath]</equals>

</if>

<if><not>

   <if><equals value="[value-for-comparison]">[xpath]</equals></if>
```

```
</not></if>
```

Figure 11. X3ML Conditions

In addition, sometimes a path in the source schema needs to be matching with several entities and relationships in the target schema. The reason for this is that usually source schemata use “shortcuts” on the way the knowledge is constructed which should be analyzed when matching to richer data models. In such cases, an intermediate entity must be introduced. Consider the example shown in Figure 12 which inserts an intermediate element in order to formulate the target relation. In general, source and target paths may contain a series of intermediate nodes and subsequent relations. If the source path contains one or more intermediates, the intermediate entities are “skipped” in the mapping, i.e., they have no equivalent entity in the target. If the target path contains intermediates, these are logically inferred (such as “birth” from “birthdate”). The “range” element is the point where both paths meet again with equivalent entities.

```
Source proposition: “record - has_descriptor – term”
Target propositions: “man-made-object - was_produced_by - PRODUCTION - used_general_technique => type”

This is formulated using the intermediate element:
<path>
    <source_relation/>
    <target_relation>
        <relationship/>
        <entity/>
        <relationship/>
    </target>
</path>
```

Figure 12. Matching paths with intermediate nodes

4.3.3.3 Variables

Sometimes it is necessary to generate an instance in X3ML only once in the scope of a given domain entity, and then re-use it in a number of links of this domain. This is most frequently the case for intermediate target nodes. For examples, a description of a museum object may re-use the same production event for matching its “creator” link and its “date” link. In these cases, an entity can be assigned to a variable:

```
<entity variable="p1">

    [generate the value]

</entity>
```

Figure 13. Assigning an *entity* to a *variable*

Entity blocks with their variables set will only generate the associated values once, and then re-use it whenever the variable name (in this case *p1*) is used again, in the scope of the respective domain.

4.3.3.4 *Additional Nodes*

Regularly constant properties and entities are needed to be added to a target entity, either from background knowledge or in order to characterize the meaning of a classification by the source schema rather than by data. For instance, a database about museum objects may not mention at all the museum as current keeper. A table “coin” may be mapped to “physical object”, but each instance of “coin” must have the type “coin”. For that purpose, the *additional* element can be used. It contains the entity which will be attached to the target entity, and the relationship describing the link, and the respective constant values. The syntax is shown in Figure 14.

```
<range>
    <source_node/>
    <target_node>
        <entity>
            <additional>
                <relationship/>
                <entity/>
            </additional>
            ... more additionals ...
        </entity>
    </target_node>
</range>
```

Figure 14. Declaring additional nodes

Note that the target allows multiple additional nodes.

4.3.1 Mapping Definition

This step gets as input the X3ML file generated at the previous step. Typically an IT expert will carry it out and does not requires the involvement of the domain experts. Then the file is provided to the data integration engine that automatically is able to generate the corresponding RDF/S triples and to answer SPARQL queries.

Each mapping must also contain one or more mapping axioms. A mapping axiom is defined as a pair of source and target. The source is an arbitraty query over the database and the target is a triple template that contains placeholders that reference column names mentioned in the source query.

For example, the following code shown in Figure 15 is a valid mapping:

```
mappingID Book Collection
Source select id, title from books
target <http://www.example.org/library#BID_{id}> rdf:type :Book; :title {title}
```

Figure 15. example mapping axiom

Note that the target of a mapping looks like RDF triples in turtle syntax, except that the triple's subject or object may contain place holders. In the previous example we have two types of place holders, one is a variable, i.e., {title}, and another is a URI Template, <http://www.example.org/library#BID_{id}>.

4.3.1.1 Prefixes

Prefixes can be used when writing a URI templates. A prefix must be enclosed between the symbols '&' and ';' and it will be replaced by the prefix definition when the system initializes.

```
PREFIX   :           <http://www.example.org/ontology1#>
PREFIX   p:          <http://www.example.org/ontology2#>
```

Figure 16. example prefixes

4.3.1.2 Types

It is possible to create typed literals by specifying the type in the mapping. For example:

```
mappingId     Book titles

source        SELECT id, title, edition, comment FROM books

target        :BID_{id} :title {title}^^xsd:string; :edition {edition}^^xsd:int; :description {com-
ment} .
```

Figure 17. example declaration of specific type

The type used in the mapping has to agree with the type in the ontology.

5 Example Instantiation of the Semantic Infrastructure for CHF Scenario

5.1 Modelling data of CHF use-case using MHA Semantic Core Ontology

To demonstrate the expressivity of the MHA Semantic Core Ontology we will focus as a running example on the CHF use-case. In short, the target of this use-case is to create a service able to empower citizens, patients and doctors by providing a supportive environment for the self-management of patients/citizens with cardiovascular disease risks.

The idea is that a citizen will download the CHF app and this app will be linked to the MyHealthAvatar account of the citizen to download all available information already stored in the MHA platform. The data needed to be downloaded are the following.

- *Sex*: The sex of the citizen
- *Age*: The age of the citizen
- *Name*: The name of the citizen
- *Forced vital capacity*: The amount of air which can be forcibly exhaled from the lungs after taking the deepest breath possible.
- *Systolic blood pressure*: The blood pressure when the heart is contracting. It is specifically the maximum arterial pressure during contraction of the left ventricle of the heart.
- *Heart rate*: Heart rate, or heart pulse, is the speed of the heartbeat measured by the number of poundings of the heart per unit of time — typically beats per minute (bpm).
- *Left Ventricular Hypertrophy*: Left ventricular hypertrophy is enlargement (hypertrophy) of the muscle tissue that makes up the wall of your heart's main pumping chamber (left ventricle).
- *Coronary heart disease*: Coronary heart disease (CHD) is a disease in which a waxy substance called plaque (plak) builds up inside the coronary arteries. These arteries supply oxygen-rich blood to the heart muscle.
- *Valve disease*: Valvular heart disease is any disease process involving one or more of the four valves of the heart (the aortic and mitral valves on the left and the pulmonary and tricuspid valves on the right). These conditions occur largely as a result of aging.
- *Diabetes*: a disorder of the metabolism causing excessive thirst and the production of large amounts of urine.
- *Cardiomegaly*: abnormal enlargement of the heart.
- *BMI*: Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women.

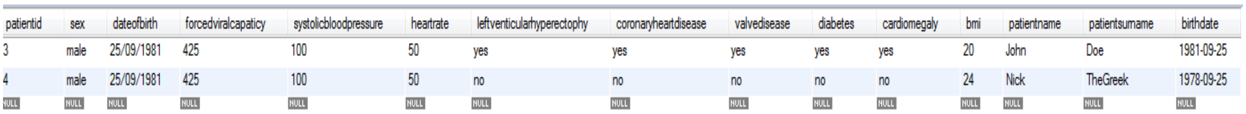
These data can be described using the following terms from the available subontologies withing the MHA Semantic Core Ontology.

- Sex ➔ TMO_0086
- Age ➔ TMO_0178
- Name ➔ foaf:name
- Forced vital capacity ➔ SNOMEDCT/50834005 (forced viral capacity)
- Systolic blood pressure ➔ SNOMEDCT/72313002 (systolic blood pressure)
- Heart rate ➔ SNOMEDCT/364075005 (heart rate)
- Left Ventricular Hypertrophy ➔ SNOMEDCT/55827005
- Coronary heart disease ➔ SNOMEDCT/56265001 (heart disease)
- Valve disease ➔ SNOMEDCT/8722008 (Aortic valve disorder)
- Diabetes ➔ SNOMEDCT/73211009 (diabetes mellitus)
- Cardiomegaly ➔ SNOMEDCT/8186001
- BMI ➔ SNOMEDCT/60621009

As we can observe *sex* and *age* are already within eTMO ontology whereas the *name* comes from the FOAF ontology. The rest of the fields are observations. Since SNOMED-CT is mapped to the phenotype class of the eTMO we can use all observation subclasses from the SNOMED-CT.

5.2 *Create Schema Matching Definitions and the corresponding Mapping Axioms.*

The aforementioned fields can be found in a NoSQL database table shown in Figure 18.



patientid	sex	dateofbirth	forcedviralcapaticy	systolicbloodpressure	heartrate	leftventicularhyperectophy	coronaryheartdisease	valvedisease	diabetes	cardiomegaly	bmi	patientname	patientsurname	birthdate
3	male	25/09/1981	425	100	50	yes	yes	yes	yes	yes	20	John	Doe	1981-09-25
4	male	25/09/1981	425	100	50	no	no	no	no	no	24	Nick	TheGreek	1978-09-25
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 18. CHF example table

So initially a domain expert has to create the matchings of the aforementioned fields with the MHA Semantic Core Ontology based also on the identified ontology terms from the previous section. Two examples matchings are shown in Figure 19 and Figure 20 graphically instead of the X3ML XML version for reasons of better understanding.

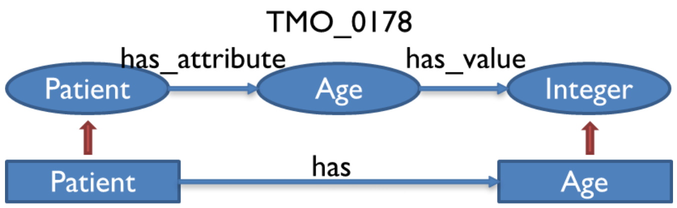


Figure 19. CHF example matching for the Age

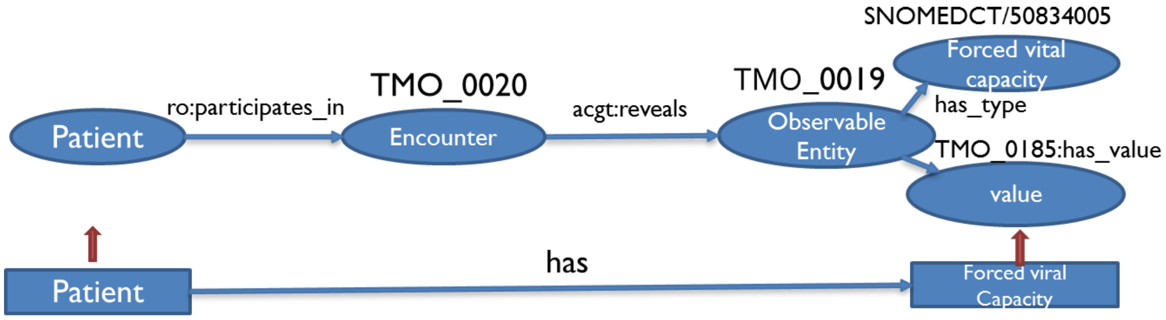


Figure 20. CHF example matching for the Forced Viral Capacity

Having generated the aforementioned X3ML matchings an IT expert has to define then the mapping axioms. The mapping axioms corresponding to Figure 19 are shown in Figure 21 whereas the mapping axioms corresponding to Figure 20 are shown in Figure 22.

```
mappingId patient has age object

target  transmed:TMO_0124_{patientid}               (patient)
        transmed:TMO_0196                           (has attribute)
        transmed:TMO_0179_{age} .                             (age)

source  select patientid, YEAR(now()) - Year(birthdate) AS age from patient



mappingId age object has age value
target    transmed:TMO_0179_{age}                             (age)
          transmed:TMO_0185                         (has value)
          {age}^^int .                                        (integer)
source    select YEAR(now()) - Year(birthdate) AS age from patient
```

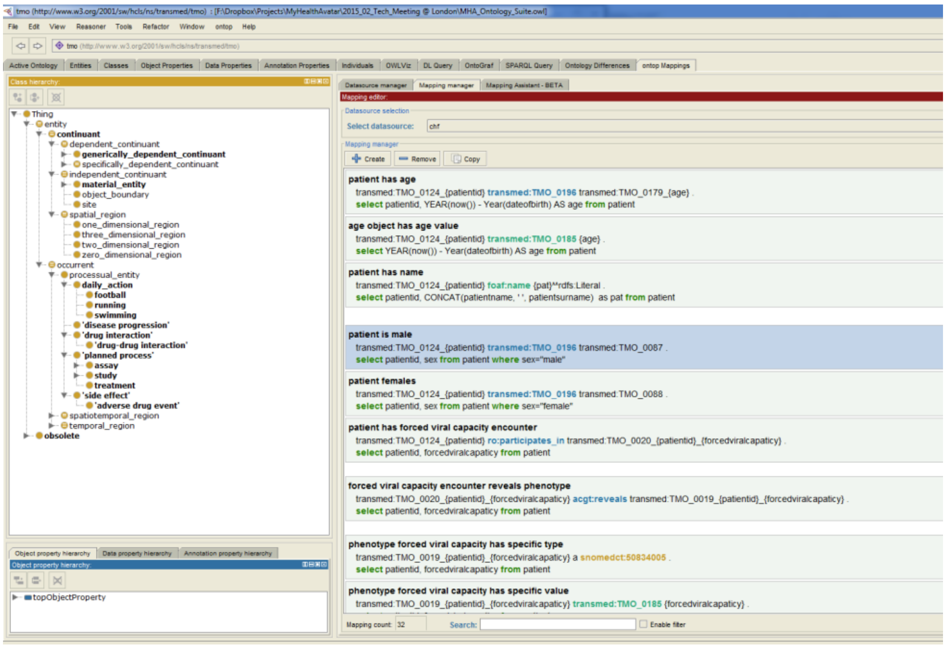
Figure 21. The mapping axioms for the Age

```
mappingId patient has forced viral capacity encounter

target  transmed:TMO_0124_{patientid}
        ro:participates_in
        transmed:TMO_0020_{patientid}_{forcedviralcapaticy} .
```

```
source  select patientid, forcedviralcapaticy from patient



mappingId  forced viral capacity encounter reveals phenotype

target  transmed:TMO_0020_{patientid}_{forcedviralcapaticy}
        acgt:reveals
        transmed:TMO_0019_{patientid}_{forcedviralcapaticy} .

source  select patientid, forcedviralcapaticy from patient



mappingId  phenotype forced viral capacity has specific type

target    transmed:TMO_0019_{patientid}_{forcedviralcapaticy} a snomedct:50834005 .

source    select patientid, forcedviralcapaticy from patient



mappingId  phenotype forced viral capacity has specific value

target  transmed:TMO_0019_{patientid}_{forcedviralcapaticy}
        transmed:TMO_0185
        {forcedviralcapaticy} .

source    select patientid, forcedviralcapaticy from patient
```

Figure 22. The mapping axioms for the Forced Viral Capacity

Note that there can be many mapping axioms defined for one X3ML Schema Matching. In addition the graphical mapping editor we are using is shown in Figure 23.



```
patient has age
  transmed:TMO_0124_{patientid} transmed:TMO_0196 transmed:TMO_0179_{age} .
  select patientid, YEAR(now()) - Year(dateofbirth) AS age from patient
age object has age value
  transmed:TMO_0124_{patientid} transmed:TMO_0185 {age} .
  select YEAR(now()) - Year(dateofbirth) AS age from patient
patient has name
  transmed:TMO_0124_{patientid} foaf:name {pat}^^rdfs:Literal .
  select patientid, CONCAT(patientname, ' ', patientsurname)  as pat from patient
patient is male
  transmed:TMO_0124_{patientid} transmed:TMO_0196 transmed:TMO_0087 .
  select patientid, sex from patient where sex="male"
patient females
  transmed:TMO_0124_{patientid} transmed:TMO_0196 transmed:TMO_0088 .
  select patientid, sex from patient where sex="female"
patient has forced viral capacity encounter
  transmed:TMO_0124_{patientid} ro:participates_in transmed:TMO_0020_{patientid}_{forcedviralcapaticy} .
  select patientid, forcedviralcapaticy from patient
forced viral capacity encounter reveals phenotype
  transmed:TMO_0020_{patientid}_{forcedviralcapaticy} acgt:reveals transmed:TMO_0019_{patientid}_{forcedviralcapaticy} .
  select patientid, forcedviralcapaticy from patient
phenotype forced viral capacity has specific type
  transmed:TMO_0019_{patientid}_{forcedviralcapaticy} a snomedct:50834005 .
  select patientid, forcedviralcapaticy from patient
phenotype forced viral capacity has specific value
  transmed:TMO_0019_{patientid}_{forcedviralcapaticy} transmed:TMO_0185 {forcedviralcapaticy} .
```

Figure 23. A graphical editor for generating mappings within Protégé

5.3 Issue Queries

After all mappings have been generated they are fed to the data translation module and queries can be issued. So the query shown in Figure 24 asks for the age and the name of all persons available in the database. There are two options here. Either the query is translated to the source databases to be answered or the entire canonical database is computed and stored at the RDF Triple Store. In both cases the results are returned to the final user.

```
PREFIX : <http://www.semanticweb.org/kondylak/ontologies/2013/1/untitled-ontology-86#>

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

PREFIX xsd: http://www.w3.org/2001/XMLSchema#

PREFIX foaf: http://xmlns.com/foaf/0.1/

PREFIX obo: <http://purl.obolibrary.org/obo/>

PREFIX ro: <http://www.obofoundry.org/ro/ro.owl#>

PREFIX transmed: <http://www.w3.org/2001/sw/hcls/ns/transmed/>

SELECT ?name ?age_value

WHERE {

          ?patient foaf:name ?name.

          ?patient transmed:TMO_0196 ?age .
```

```
          ?age transmed:TMO_0185 ?age_value .

}
```

Figure 24. Querying for the age of all users

6 Conclusion

In the present deliverable we focused on the development of the MyHealthAvatar semantic backbone and more specifically on the MHA Semantic Core Ontology and the Semantic Integration Layer and its components.

In Chapter 3 we presented the MHA Semantic Core ontology and the extensions performed to enable the representation of the various data within MyHealthAvatar. We have to note that the creation and the specification of the MHA Semantic Core ontology is an ongoing process. That means that in time and depending on the evolution and the refinement of the requirements of each use case the ontology might change or additionally extended.

Then in Chapter 4 we focused on the description of the Semantic Integration layer. Disparate data sources are mapped to the MHA Semantic Core Ontology in a two-step process. First the matchings between the schemata are established and this process can be executed by the domain experts. Then IT experts transform these matching in mapping axioms. Using these mapping axioms the Data Translation Layer is able either to directly answer SPARQL queries by rewriting those queries to use the source schemata or to materialize the entire canonical database and store in in the central RDF triple store. For reasons of efficiency usually the materialized canonical database is stored in a Virtuoso RDF triple store and the queries are directly answered by the Virtuoso SPARQL endpoint.

We finally demonstrated an instantiation of the semantic infrastructure in Chapter 5 for the use case of CHF to better understand the whole process and set up. In the following months we ‘ll focus more on assessing and evaluating the proposed solution and on developing mechanisms that will allow the uninterrupted evolution of the MHA Semantic Core Ontology and the summarization of the available knowledge within the RDF Triple store. The results will be reported in the next deliverable of WP4 the D4.3 Technical evaluation report of ontology including ontology evolution and summarization.

7 References

[1] Berners-Lee, T., Hendler, J. & Lassila, O. 2001. “The Semantic Web” Scientific American, 284 (5), pp. 34-43.

[2] Semantic Integration. 2013 [cited 2013 14/11/2013]; Available from: http://en.wikipedia.org/wiki/Semantic_integration.

[3] Calvanese D, De Giacomo G, Lembo D, Lenzerini M, Poggi A, Rodriguez-Muro M, et al. Ontol-ogies and Databases: The DL-Lite Approach. Reasoning Web. 2009:255-356.

[4] Martin L, Anguita A, Maojo V, Bonsma E, Bucur A, Vrijnsen J, Brochhausen M, Cocos C, Stenzhorn H, Tsiknakis M, Doerr M, Kondylakis H:. Ontology Based Integration of Distributed and Heterogeneous Data Sources in ACGT. HEALTHINF. 2008:301-6.

[5] Kondylakis, H., Plexousakis, D., Hrgovcic, V., Woitsch, R., Premm, M., Schuele, M.: Agents, Models and Semantic Integration in support of Personal eHealth Knowledge Spaces , Web Information System Engineering (WISE), 1, pp. 496-511, 2014, Thessaloniki, Greece

[6] Flouris G, Manakanatas D, Kondylakis H, Plexousakis D, Antoniou G. Ontology change: Classification and survey. The Knowledge Engineering Review. 2007:1-29.

[7] Luciano, J.S., Andersson, B., Batchelor, C., Bodenreider, O., Clark, T., Denney, C., Domarew, C., Gambet, T., Harland, L., Jentzsch, A., Kashyap, V., Kos, P., Kozlovsky, J., Lebo, T., Marshall, M.S., McCusker, J.P., McGuiness, D.L., Ogbuji, C., Pichler, E., Powers, R.L., Prud'hommeaux, E., Samwald, M., Schriml, L., Tonellato, P.J., Whetzel, P.L., Zhao, J., Stephens, S., Dumontier, M.: The Translational Medicine Ontology and Knowledge Base: Driving personalized medicine by bridging the gap between bench and bedside. Journal of Biomedical Semantics, 2011.

[8] Kondylakis, H., Doerr, M., de Jong, G., Oldman, D., Norton, B.: *X3ML: An interface for sustainable management of data mapping process*, International Journal on Digital Libraries, Springer, (Submitted February 2015 – review pending)

[9] Kondylakis, H., Dimitris, P.: Exelixis: Evolving Ontology-Based Data Integration System. SIGMOD, pp. 1283-1286 (2011)

[10] Gomez-Perez, A., Fernandez, M. and De Vicente, A.J.: Towards a Method to Conceptualize Domain Ontologies. In: ECAI-96 Workshop on Ontological engineering, Budapest (1996).

[11] MyHealthAvatar Consortium, D4.1 Requirement Analysis for Semantic Core Ontology, July 2013

[12] MyHealthAvatar Consortium, D7.1 Description of scenarios and use-cases for MyHealthAvatar, August 2014

[13] Kondylakis, H., Plexousakis, D.: Ontology Evolution without Tears, Journal of Web Semantics (2013), 19, pp. 42-58, Elsevier

[14] LDBC Consortium : D4.4.1 Use Case Analysis and Classification of Choke Points, September 2013

[15] Lenzerini, M.: Data integration: a theoretical perspective. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, Madison, Wisconsin (2002)

[16] Serfiotis, G., Koffina, I., VassilisChristophides, Tannen, V.: Containment and Minimization of RDF/S Query Patterns. ISWC, pp. 607-623 (2005)

[17] Perez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. ACM Trans. Database Syst. 34, 1-45 (2009)

[18] Cali, A., Gottlob, G., Lukasiewicz, T.: Datalog+-: a unified approach to ontologies and integrity constraints. ICDT, pp. 14-30. ACM, St. Petersburg, Russia (2009)

[19] Lenzerini, M.: Data integration: a theoretical perspective. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, Madison, Wisconsin (2002)

[20] Poggi, A., Lembo, D., Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Linking data to ontologies. Journal on data semantics X 133-173 (2008)

[21] Calì, A., Calvanese, D., Giacomo, G.D., Lenzerini, M.: Data Integration under Integrity Constraints. Advanced Information Systems Engineering, pp. 262-279 (2006)

[22] Deutsch, A., Popa, L., Tannen, V.: Query reformulation with constraints. SIGMOD Rec. 35, 65-73 (2006)

[23] Kondylakis, H., Doerr, M., de Jong, G., Norton, B., Oldman, D.: X3ML: An interface for sustainable management of data mapping process, IJDL Journal, 2015 (review pending)

[24] Hert M, Reif G, Gall HC (2011) A comparison of RDB-to-RDF mapping languages. In: Proceedings of the 7th International Conference on Semantic Systems. I-Semantics:25–32

[25] Berners-Lee T (2009) Relational Databases on the Semantic Web. Available online http://www.w3.org/DesignIssues/RDB-RDF.html.

[26] SquirrelRDF. Available: http://jena.sourceforge.net/SquirrelRDF/. Accessed February 2015

[27] Bizer C (2003) D2R MAP – A Database to RDF Mapping Language. In Proceedings of the 12th International World Wide Web Conference

[28] Barrasa J, Corcho O, Gomez-Perez A (2003) Fund Finder: A Case Study of Database-to-Ontology Mapping. In Proceedings of the Semantic Integration Workshop, 2003.

[29] Bizer C, Seaborne A (2004) D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. In Proceedings of the 3rd International Semantic Web Conference Posters, 2004.

[30] Barrasa J, Corcho O, Gomez-Perez A (2004) R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language. In Proceedings of the 2nd Workshop on Semantic Web and Databases:1069-1070

[31] Openlink Software: Mapping Relational Data to RDF with Virtuoso's RDF Views http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html. Accessed February 2015

[32] Auer S, Dietzold S, Lehmann J, Hellmann S, Aumueller D (2009) Triplify – Light-Weight Linked DataPublication from Relational Databases. In Proceedings of the 18th International World Wide Web Conference:621-630

[33] Das S, Sundara S, Cyganiak R (2010) R2RML: RDB to RDF Mapping Language. W3C Working Draft. Available online http://www.w3.org/TR/2010/WD-r2rml-20101028/,October 2010.

[34] RDB2RDF Implementations, Available: http://www.w3.org/2001/sw/rdb2rdf/wiki/Implementations, Accessed February 2015

[35]Dimou A, Vander Sande M, Colpaert R, Verborgh R, Mannens E, Van de Walle R (2014) RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data, 7th Workshop on Linked Data on the Web

[36]de Laborda CP, Conrad S (2005) Relational.OWL – A Data and Schema Representation Format Based on OWL. In Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling:89-96

[37]Langegger A, Woß W (2009) XLWrap – Querying and Integrating Arbitrary Spreadsheets with SPARQL. In Proceedings of the 8th International Semantic Web Conference:359–374

[38]O'Connor MJ, Halaschek-Wiener C, Musen MA (2010) Mapping Master: a flexible approach for mapping spreadsheets to OWL. In Proceedings of the 9th International Semantic Web Conference (ISWC):194–208

[39]Vertere RDF: https://github.com/knudmoeller/Vertere-RDF. Accessed February 2015

[40]TARQL: https://github.com/cygri/tarql. Accessed February 2015

[41]Lange C (2011) Krextor - an extensible framework for contributing content math to the Web of Data. In Proceedings of the 18th Calculemus and 10th international conference on intelligent computer mathematics (MKM):304–306

[42]AstroGrid-D: http://www.gac-grid.de/project-products/Software/XML2RDF.html. Accessed February 2015

[43]Tripliser: http://daverog.github.io/tripliser/. Accessed February 2015

[44]Bischof S, Decker S, Krennwallner T, Lopes N, Polleres A (2012) Mapping between RDF and XML with XSPARQL. Journal on Data Semantics 1(3):147–185

[45]GRDDL: http://www.w3.org/TR/grddl/. Accessed February 2015

[46]Scharffe F, Atemezing G, Troncy R, Gandon F, Villata S, Bucher B, Hamdi F, Bihanic L, Kepeklian G, Cotton F, Euzenat J, Fan Z, Vandenbussche PY, Vatant B (2012) Enabling Linked Data publication with the Datalift platform. In Proc. AAAI workshop on semantic cities

[47]The DataTank: http://thedatatank.com. Accessed February 2015

[48]OpenRefine: http://openrefine.org/. Accessed February 2015

[49]RDFizers: http://simile.mit.edu/wiki/RDFizers. Accessed February 2015

[50]Sponger: http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger

[51]Szekely PA, Knoblock CA, Yang F, Zhu X, Fink EE, Allen R, Goodlander G (2013) Connecting the Smithsonian American Art Museum to the Linked Data Cloud. ESWC: 593-607

[52]SIP-Creator: https://github.com/delving/delving/tree/master/sip-creator. Accessed February 2015

[53]Groovy Programming Language: http://groovy.codehaus.org/. Accessed February 2015

Appendix 1 – Abbreviations and acronyms

ACGT	Advancing Clinic-Genomic Trials on Cancer
ACGT MO	Advancing Clinical-Genomic Trials on Cancer Master Ontology
ACGT SM	Advancing Clinical-Genomic Trials on Cancer Semantic Mediation Layer
AEO	Adverse Event Ontology
AERO	Adverse Event Reporting Ontology
At	Anatomy Taxonomy
ASA	Anatomical Structural Abstractin
ATA	Anatomical Transformation Abstraction
ATC	Anatomical Therapeutic Chemical
BCO	BioCaster Ontology
BFO	Basic Formal Ontology
CCC	Clinical Care Classification System
CCD	Continuity of Care Document
CCR	Continuity of Care Record
CDM	Common Data Model
CHEBI	Chemical Entities of Biological Interest
CIM	Common Information Model
CL Ontology	Cell Type Ontology
CT	Clinical Trial
DDD	Defined Daily Dose
DO	Disease Ontology
DOW	Description of Work

EFO	Experimental Factor Ontology
EHR	Electronic Health Record
EMAP	Edinburgh Mouse Atlas
FHHO	Family Health History Ontology
FMA	Foundational Model of Anatomy
GO	Gene Ontology
HDOT	Health Data Ontology Trunk
HIS	Hospital Information system
HL7	Health Level 7
IAO	Information Artifact Ontology
ICD	International Classification of Diseases
ICF	International Classification of Functioning
ICT	Information & Communication Technology
IFOMIS	Institute for Formal Ontology and Medical Information Science
IGS	Institute for Genome Sciences
LOINC	Logical Observation Identifiers Names and Codes
LODD	Linking Open Drug Data
LUI	Lexical Unique Identifier
MA	Mouse Anatomical
MeSH	Medical Subject Headings
MHA	MyHealthAvatar
MK	Metaknowledge
MLOCC	Middle Layer Ontology for Clinical Care

MO	Master Ontology
MRI	Magnetic resonance imaging
NIF	Neuroscience Information Framework Project
NIFSTD	Neuroscience Information Framework Standardized Ontology
NLM	US National Library of Medicine
NNEW	Nextgen Network Enabled Weather
OBI	Ontology for Biomedical Investigation
OBO	Open Biomedical Ontologies
OGMS	Ontology for General Medical Science
OMRSE	Ontology of Medically Relevant Social Entities
OPB	Ontology for Physics in Biology
ORF	Original Release Format
OWL	Ontology Web Language
p-Medicine	Personalized Medicine
PATO	Phenotypic Quality Ontology
PHR	Personal Health Record
PRO	Protein Ontology
RICORDO	Researching Interoperability using Core Reference Datasets and Ontologies for the Virtual Physiological Human
RO	Relational Ontology
RRF	Rich Release Format
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
SNOMED RT	Systematized Nomenclature of Medicine Reference Terminology

SOA	Service Oriented Architecture
TMKB	Translational Medicine Knowledge Base
TMO	Translational Medicine Ontology
TOVE	TOronto Virtual Enterprise methodology
UMLS	Unified Medical Language System
UO	Units Ontology
VPH-NoE	Virtual Physiological Hyman Network of Excellence
WHO	World Health Organisation
WP	Work Package

Appendix 2 – Mappings between eTMO and subontologies

eTMO TERM	relationship	EXTERNAL ONTOLOGY TERM
<http://purl.obolibrary.org/obo/IAO_0000310>	owl:equivalence	<http://bio2rdf.org/drugbank:references> .
<http://purl.obolibrary.org/obo/IAO_0000310>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/references> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/dailymed:drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/dailymed/resource/dailymed/drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/medicare:drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/medicare/resource/medicare/drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/sider:drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/sider/resource/sider/drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/drugbank:drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugs> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Drug> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Drug> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0020>	owl:equivalence	<http://bio2rdf.org/pchr:Encounter> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0020>	owl:equivalence	<tag:eric@w3.org:2009/tmo/translator#Encounter> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/linkedct:agency> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/agency> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/dailymed:organization> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/dailymed/resource/dailymed/organization> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://bio2rdf.org/linkedct:intervention> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/intervention> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://bio2rdf.org/linkedct:trials> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/trials> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/diseasome:genes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/genes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0043>	owl:equivalence	<http://bio2rdf.org/sider:side_effects> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0043>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/sider/resource/sider/side_effects> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0046>	owl:equivalence	<http://bio2rdf.org/linkedct:trial_arm_group> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0046>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/trial_arm_group> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Disease> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Disease> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/linkedct:condition> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/condition> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/diseasome:diseases> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/diseases> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://bio2rdf.org/linkedct:secondary_outcomes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/secondary_outcomes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://bio2rdf.org/linkedct:primary_outcomes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://data.linkedct.org/resource/linkedct/primary_outcomes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0066>	owl:equivalence	<http://bio2rdf.org/drugbank:targets> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0066>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/targets> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0071>	owl:equivalence	<http://bio2rdf.org/dailymed:ingredients> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0071>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/dailymed/resource/dailymed/ingredients> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0078>	owl:equivalence	<http://bio2rdf.org/drugbank:enzymes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0078>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/enzymes> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Association> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://bio2rdf.org/pharmgkb:Association> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://bio2rdf.org/pharmgkb:DrugGeneVariantInteraction> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://bio2rdf.org/pharmgkb:DrugGeneVariantInteraction> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://bio2rdf.org/drugbank:drug_interactions> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0304>	owl:equivalence	<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drug_interactions> .
<http://www.ifomis.org/bfo/1.1/snap#Continuant>	owl:equivalence	<http://bio2rdf.org/foaf:Agent> .
<http://www.ifomis.org/bfo/1.1/snap#Continuant>	owl:equivalence	<http://bio2rdf.org/dcterms:Agent> .
<http://www.ifomis.org/bfo/1.1/snap#Continuant>	owl:equivalence	<http://bio2rdf.org/dcterms:AgentClass> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0000>	owl:equivalence	<http://bio2rdf.org/hl7v3.0:ACTI> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/ncit:Pharmaceutical_Formulation> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0001>	owl:equivalence	<http://bio2rdf.org/umls:C1705957> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0002>	owl:equivalence	<http://bio2rdf.org/ncit:Finished_Pharmaceutical_Product> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0002>	owl:equivalence	<http://bio2rdf.org/umls:C1708062> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0003>	owl:equivalence	<http://bio2rdf.org/hl7v3.0:IACT> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0006>	owl:equivalence	<http://bio2rdf.org/ncit:Target> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0006>	owl:equivalence	<http://bio2rdf.org/umls:C1521840> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0007>	owl:equivalence	<http://bio2rdf.org/galen:Prognosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0007>	owl:equivalence	<http://bio2rdf.org/msh:D011379> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0007>	owl:equivalence	<http://bio2rdf.org/ncit:Prognosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0007>	owl:equivalence	<http://bio2rdf.org/umls:C0033325> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0009>	owl:equivalence	<http://bio2rdf.org/ocre_clinical:Clinician> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://bio2rdf.org/birnlex:2364> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://bio2rdf.org/bro:Patent> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://bio2rdf.org/galen:patent> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://bio2rdf.org/iao:0000313> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000313> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0012>	owl:equivalence	<http://bio2rdf.org/msh:D020490> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0013>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0013>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0014>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0014>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0015>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0015>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0016>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0016>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0017>	owl:equivalence	<http://bio2rdf.org/ncit:Evidence> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0017>	owl:equivalence	<http://bio2rdf.org/umls:C0332120> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0017>	owl:equivalence	<http://bio2rdf.org/iao:0000123> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0017>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000123> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0018>	owl:equivalence	<http://bio2rdf.org/iao:0000123> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0018>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000123> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0020>	owl:equivalence	<http://bio2rdf.org/ncit:Personal_Medical_History> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0020>	owl:equivalence	<http://bio2rdf.org/umls:C0262926> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0022>	owl:equivalence	<http://bio2rdf.org/ncit:Sign> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0022> (label:sign)	owl:equivalence	<http://bio2rdf.org/snomedct:72670004> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0022>	owl:equivalence	<http://bio2rdf.org/umls:C0311392> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0023>	owl:equivalence	<http://bio2rdf.org/acgt:Symptom> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0023>	owl:equivalence	<http://bio2rdf.org/cpr:symptom> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0023>	owl:equivalence	<http://bio2rdf.org/ncit:Symptom> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0023>	owl:equivalence	<http://bio2rdf.org/umls:C1457887> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/acgt:Institution> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/birnlex:2085> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/lnc:LP76237-4> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/ncit:Institution> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025> (label: institution)	owl:equivalence	<http://bio2rdf.org/snomedct:385437003> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0025>	owl:equivalence	<http://bio2rdf.org/umls:C1272753> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0027>	owl:equivalence	<http://bio2rdf.org/ncit:Healthcare_Payer> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0027>	owl:equivalence	<http://bio2rdf.org/umls:C2348942> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/lnc:LP74341-6> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/ncit:Study> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/ocre_research:Study> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029> (label:study)	owl:equivalence	<http://bio2rdf.org/snomedct:224699009> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/umls:C0947630> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://bio2rdf.org/obi:0000471> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0029>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000471> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://bio2rdf.org/ncit:Intervention> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://bio2rdf.org/ocre_research:Intervention> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0030>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/acgt:Diagnosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/fhho:Diagnosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/galen:Diagnosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/lnc:LP72437-4> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/msh:D003933> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/ncit:Diagnosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/obi:0000075> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000075> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/ocre_clinical:Diagnosis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031> (label:diagnosis)	owl:equivalence	<http://bio2rdf.org/snomedct:439401001> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031>	owl:equivalence	<http://bio2rdf.org/umls:C0011900> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://bio2rdf.org/hl7v3.0:CLNTRL> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://bio2rdf.org/msh:D016430> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://bio2rdf.org/ncit:Clinical_Trial> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032> (lebel:clinical trial)	owl:equivalence	<http://bio2rdf.org/snomedct:110465008> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0032>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0034>	owl:equivalence	<http://bio2rdf.org/chebi:23367> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0034>	owl:equivalence	<http://purl.obolibrary.org/obo/CHEBI_23367> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0034>	owl:equivalence	<http://bio2rdf.org/gro:MolecularEntity> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/acgt:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/birnlex:23> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/chebi:36080> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://purl.obolibrary.org/obo/CHEBI_36080> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/fma:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/go:0003675> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://purl.obolibrary.org/obo/GO_0003675> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/gro:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/galen:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/ncit:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/odgi:Protein> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/pro:000000001> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035> (label: protein)	owl:equivalence	<http://bio2rdf.org/snomedct:88878007> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/so:0000358> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://purl.obolibrary.org/obo/SO_0000358> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0035>	owl:equivalence	<http://bio2rdf.org/umls:C0033684> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0036>	owl:equivalence	<http://bio2rdf.org/birnlex:11036> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0036>	owl:equivalence	<http://bio2rdf.org/efo:0000322> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0036>	owl:equivalence	<http://bio2rdf.org/msh:D002460> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0036>	owl:equivalence	<http://bio2rdf.org/ncit:Cell_Line> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0036>	owl:equivalence	<http://bio2rdf.org/umls:C0007600> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/fma:Structural_gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/gro:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/galen:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/lnc:LP32747-5> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/msh:D005796> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/ncit:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/ncit:Gene_Object> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/ndfrt:C242394> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/pro:Gene> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037> (label:gene)	owl:equivalence	<http://bio2rdf.org/snomedct:67271001> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/so:0000704> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://purl.obolibrary.org/obo/SO_0000704> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0037>	owl:equivalence	<http://bio2rdf.org/umls:C0017337> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0038>	owl:equivalence	<http://bio2rdf.org/msh:D020641> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0038>	owl:equivalence	<http://bio2rdf.org/ncit:Single_Nucleotide_Polymorphism> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0038>	owl:equivalence	<http://bio2rdf.org/so:0000694> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0038>	owl:equivalence	<http://purl.obolibrary.org/obo/SO_0000694> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0038>	owl:equivalence	<http://bio2rdf.org/umls:C0752046> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0039>	owl:equivalence	<http://bio2rdf.org/so:0001019> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0039>	owl:equivalence	<http://purl.obolibrary.org/obo/SO_0001019> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0040> (label: drug-drug interaction)	owl:equivalence	<http://bio2rdf.org/snomedct:404204005> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0040>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0040>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/msh:D018450> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/ncit:Disease_Progression> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/ndfrt:C7812> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/obi:0000354> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000354> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/umls:C0242656> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0041>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0042>	owl:equivalence	<http://bio2rdf.org/ncit:Mechanism_of_Action> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0043>	owl:equivalence	<http://bio2rdf.org/msh:D004362> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0043> (label:adverse drug event)	owl:equivalence	<http://bio2rdf.org/snomedct:406644009> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0044>	owl:equivalence	<http://bio2rdf.org/gro:Pathway> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0044>	owl:equivalence	<http://bio2rdf.org/galen:Pathway> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0044>	owl:equivalence	<http://bio2rdf.org/iev:0000001> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0044>	owl:equivalence	<http://bio2rdf.org/mi:0619> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0044>	owl:equivalence	<http://bio2rdf.org/ncit:Pathway> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0045>	owl:equivalence	<http://bio2rdf.org/ocre_research:Population> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0045>	owl:equivalence	<http://bio2rdf.org/dcterms:AgentClass> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0045>	owl:equivalence	<http://bio2rdf.org/iao:0000120> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0045>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000120> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0046>	owl:equivalence	<http://bio2rdf.org/iao:0000120> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0046>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000120> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/acgt:Disease> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/birnlex:11013> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/doid:4> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/gro:Disease> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/lnc:LP21006-9> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/msh:D004194> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/ncit:Disease_or_Disorder> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/ndfrt:C2140> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/obi:0000155> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000155> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047>	owl:equivalence	<http://bio2rdf.org/umls:C0012634> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0048>	owl:equivalence	<http://bio2rdf.org/so:0001023> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0048>	owl:equivalence	<http://purl.obolibrary.org/obo/SO_0001023> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://bio2rdf.org/chebi:25212> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://purl.obolibrary.org/obo/CHEBI_25212> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://bio2rdf.org/galen:Metabolite> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://bio2rdf.org/lnc:LP20702-4> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://bio2rdf.org/ncit:Metabolite> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0049>	owl:equivalence	<http://bio2rdf.org/umls:C0870883> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0050>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0050>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0051>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0051>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0052>	owl:equivalence	<http://bio2rdf.org/msh:D054316> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0052>	owl:equivalence	<http://bio2rdf.org/ncit:Biomarker> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0052>	owl:equivalence	<http://bio2rdf.org/umls:C1955886> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0052>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0052>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0053>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0053>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/msh:D008019> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/ncit:Lifestyle> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054> (label: lifestyle)	owl:equivalence	<http://bio2rdf.org/snomedct:134436002> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/snomedct:366925008> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/snomedct:367150003> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/umls:C0023676> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0054>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://bio2rdf.org/iao:0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0055>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000121> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0056>	owl:equivalence	<http://bio2rdf.org/acgt:Toxicity> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0056>	owl:equivalence	<http://bio2rdf.org/msh:Q000633> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0056>	owl:equivalence	<http://bio2rdf.org/ncit:Toxicity> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0056>	owl:equivalence	<http://bio2rdf.org/umls:C0600688> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0057>	owl:equivalence	<http://bio2rdf.org/efo:0000428> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0057>	owl:equivalence	<http://bio2rdf.org/lnc:LP6801-7> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0057>	owl:equivalence	<http://bio2rdf.org/ncit:Dose> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0057> (label: dosage)	owl:equivalence	<http://bio2rdf.org/snomedct:260911001> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0057>	owl:equivalence	<http://bio2rdf.org/umls:C0178602> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/acgt:Syndrome> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/doid:225> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/efo:0000428> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/efo:0000716> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/msh:D013577> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/ncit:Syndrome> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0058>	owl:equivalence	<http://bio2rdf.org/umls:C0039082> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0059>	owl:equivalence	<http://bio2rdf.org/ncit:Hypothesis> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0059>	owl:equivalence	<http://bio2rdf.org/obi:0000074> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0059>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000074> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0059>	owl:equivalence	<http://bio2rdf.org/umls:C1512571> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0060>	owl:equivalence	<http://bio2rdf.org/ocre_clinical:Subject> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0061>	owl:equivalence	<http://bio2rdf.org/ncit:Side_Effect> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0068>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0068>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0069>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0069>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0070>	owl:equivalence	<http://bio2rdf.org/iao:0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0070>	owl:equivalence	<http://purl.obolibrary.org/obo/IAO_0000124> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0075>	owl:equivalence	<http://bio2rdf.org/birnlex:11021> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0075>	owl:equivalence	<http://bio2rdf.org/obi:0000011> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0075>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000011> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/birnlex:11025> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/efo:0001455> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/galen:Assay> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/msh:Q000032> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/ncit:Assay> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/pato:0000003> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076> (label: assay)	owl:equivalence	<http://bio2rdf.org/snomedct:272392009> .

<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/umls:C1510438> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/cto_asyont:Assay> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://bio2rdf.org/obi:0000070> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0076>	owl:equivalence	<http://purl.obolibrary.org/obo/OBI_0000070> .
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0124>	owl:equivalence	<http://xmlns.com/foaf/0.1/person>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0019> (label:phenotype)	owl:equivalence	<http://purl.bioontology.org/ontology/SNOMEDCT/observable_entity>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_temporal instant>	owl:equivalence	<http://www.w3.org/TR/owl-time/Instant>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_temporal instant>	owl:equivalence	<http://purl.obolibrary.org/time/Interval>
<http://www.w3.org/2001/sw/hcls/ns/transmed/demographics>	rdfs:subClassOf	<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0017>
<http://purl.obolibrary.org/obo/RO_0002217>	rdfs:superPropertyOf	<http://www.ifomis.org/acgt/1.0#hasDisease>
<http://www.ifomis.org/bfo/1.1/span#ConnectedSpatiotemporalRegion>	rdfs:superClassOf	<http://www.ll.mit.edu/2009/weather-merged.owl#MereologicalPhenomena>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0094> (label:quantity)	rdfs:superClassOf	<http://www.ll.mit.edu/2009/weather-merged.owl#Measurement>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0047> (label:disease)	owl:equivalence	<http://purl.bioontology.org/ontology/ICD10/Disease>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0023> (label:symptom)	owl:equivalence	<http://purl.bioontology.org/ontology/SYMP/Symptom>
<http://www.w3.org/2001/sw/hcls/ns/transmed/TMO_0031> (label:diagnosis)	owl:equivalence	<http://purl.bioontology.org/ontology/FHHO/Diagnosis>