Consolidated approach for semantic mediation and integration of heterogeneous data sources for clinical trials

Project Number:	FP6-2005-IST-026996
Deliverable id:	D7.4
Deliverable name:	Consolidated approach for semantic mediation and integration of heterogeneous data sources for clinical trials
Submission Date:	12/05/2008

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	ACGT
Project Full Name:	Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery
Document id:	D7.4
Document name:	Consolidated approach for semantic mediation and integration of heterogeneous data sources for clinical trials
Document type (PU, INT, RE)	RE
Version:	1.0
Submission date:	12/05/2008
Editor: Organisation: Email:	Luis Martín Universidad Politécnica de Madrid lmartin@infomed.dia.fi.upm.es

Document type PU = public, INT = internal, RE = restricted

ABSTRACT: This deliverable describes the semantic mediation approach adopted in ACGT. The algorithms used in the implementation of the semantic mediator are shown. A case study has been included, providing insight of the full mediation process (since the moment is decided to include a new database in the system, until final queries are launched and results are retrieved) for a given example—a Pseudo-TOP scenario query.

This document contains the following sections: section 1 gives an introduction to the work carried out in mediation, section 2 shows the approaches and algorithms and section 3 contains the case study. The deliverable also includes 3 technical annexes with information related to the case study.

KEYWORD LIST: clinical trials, database integration, web services, semantic mediation, ontologies

MODIFICATION CONTROL			
Version	**Date**	**Status**	**Editor**
0.1	20/01/2008	Draft	Luis Martin
0.5	01/03/2008	Pre-Final	Luis Martin
0.9	12/05/2008	Pre-Final	Luis Martin
1.0	25/05/2008	Final	Luis Martin

List of contributors:

- Luis Martín, UPM
- Alberto Anguita, UPM
- Stefano Chiesa, UPM

Contents

1. Introduction

Database Integration is one of the key challenges in the ACGT project. The need of providing seamless access to heterogeneous, disparate, multilevel types of data sources have led to the dedication of specific efforts in this field. To this end, a software layer called the ACGT Semantic Mediator has been developed and integrated in the ACGT platform. The ACGT Semantic Mediator has one main functionality: answering client queries to an integrated set of databases. This database set is comprised by different, maybe disparate sources containing data of interest for a clinical trial.

The Semantic Mediator is placed in the Bioinformatics and Knowledge Discovery Services layer of the ACGT Architecture. This middleware receives data from different sources. Some of these sources act as clients—e.g. the Knowledge Discovery Tools—, some other as providers—e.g. the Data Access Services— of the tool. The Semantic Mediator can be considered a core in the flow of data across the different tools. For that reason, a simple yet powerful interface has been defined, based on OGSA-DAI web services technology, providing a means for every tool inside the platform to communicate with it, being able to send queries and receive the required data.

The Semantic Mediator is supported by a set of tools and resources integrated in the ACGT platform. These elements interact with the Semantic Mediator before, during and after the mediation process to guarantee the consistency of the results retrieved.The Semantic Mediator interacts with a variety of tools within the ACGT platform. It takes two different roles, namely *provider* or *client*, depending on the kind of interaction that is taking place. Different types of data flows take place depending on the type of interaction with the tool. The Mediator acts as a *provider* when another tool or user launches a query against it. In this case, the Mediator acts as a data provider. There are two main types of Mediator’s clients within the ACGT platform: the ACGT Query Tool and the Knowledge Discovery Tools. The Mediator acts as a *client* when he requires another’s tool services. This is mainly the case of the ACGT Data Access Wrapper. After the query translation process, the Semantic Mediator produces queries for the different database wrappers. In this case, the Mediator is the client. These queries are formulated in SPARQL language, and it accepts results in SPARQL Result Set format.

Besides the different tools in the platform that interact with the Mediator as clients or information providers, there exist other set of tools and resources needed to give support to different steps of the mediation process. These tools and resources are involved of different tasks, such as data pre-processing and mapping construction. The main resource used in the mediation process is the ACGT Master Ontology, that acts as global schema.

The ACGT Master Ontology is used in two main steps of the mediation process: i) the definition of the global schema, and 2) the mapping process. In the former, an RDF Schema extracted from the ontology is exposed and can be used by clients to build the queries for the tool, while in the latter, subsets of the ontology with a defined meaning are mapped with elements from database wrappers’ schemas. The ontology is a critical resource of the mediation process, and the ACGT Master Ontology is a core resource in the ACGT platform, not only for mediation and database integration, but to guarantee the semantic interoperability throughout all the tools and resources. The approach selected for mediation, then, must allow the using of a model defined entirely independent from the database integration domain.

This deliverable is organized as follows: Section 2 describes the approach adopted for semantic mediation, showing the algorithms and data formats used in the process. Section 3 describes a case study taken from the Pseudo-TOP scenario. This case has been validated together with the rest of the tools in the platform. The document includes three technical annexes with complementary information of the selected scenario.

2. The ACGT Database Integration and Semantic Mediation Approach

2.1. Background

From the theoretical point of view, we can identify three types of database integration methods [1], namely Information Linkage, Data Transformation and Query Translation. They differ in the way the information is treated. Information Linkage approaches use cross references pointing to the different data sources, Data Transformation methods centralize the data in a repository, and Query Translation is based on the transformation of a query expressed in terms of a given schema to queries for the mediated databases. Given the disparate and evolving nature of data in the biomedical domain, the size of the databases and other important issues like the security, Query Translation approaches seem to fit better for mediation solutions.

The methods to tackle Query Translation can be classified in two categories: 1) Global as View [2], and 2) Local as View [3]. The Global as View approach is based on the idea of creating an ad-hoc schema (called global schema) built by analyzing the schemas of the underlying databases. In Global as View query translation is straightforward, since all possible heterogeneities are taking into consideration when creating the global schema. The only drawback is that the system created is very sensible to changes—i.e. if the schema of an integrated database change, or a new database is needed to be integrated in the system, the complete global schema must be redefined. By contrast, in Local as View the global schema is used to define a local view for each database. In this case, the global schema does not need to take into account the characteristics of the database schemas. Local as View behaves better in dynamic domains, since only local views needs to be changed when databases change. However, it presents scalability problems in the query translation process.

2.2. Mapping between schemas

The discovery and annotation of elements with the same meaning between two schemas is called the mapping process. The ACGT Semantic Mediator query translation algorithm needs of these correspondences (between the ACGT Master Ontology and every one of the database wrappers' schemas) to produce the queries that are sent to the Data Access Services. These mappings can be of different granularities: for example, if we are talking about mapping ontologies, it would be feasible to think on mapping classes, relations or even complete sub-trees of the taxonomies. Given the characteristics and different cases of heterogeneity present in clinical trial databases, it has been decided to adopt a "view based" approach.

The next two subsections describe the mapping format and mapping process adopted in ACGT.

2.2.1. The Mapping Format

In order to cover all possible heterogeneities, a mapping format for semantic mediation has been defined. This format is based in the concept of view. We understand a view as a set of paths with constraints. The DTD of the mapping format can be seen in figure 1.

```
<!ELEMENT mapping (dbinfo,ontoclean?, map+)>
<!ELEMENT dbinfo (dbid,wrapperurl,description?)>
<!ELEMENT dbid (#PCDATA)>
<!ELEMENT wrapperurl (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT ontoclean (conceptualtophysical?,physicaltoconceptual?)>
<!ELEMENT conceptualtophysical (#PCDATA)>
<!ELEMENT physicaltoconceptual (#PCDATA)>
<!ELEMENT map (entrydescription?,condition*,path_list)>
<!ELEMENT entrydescription (#PCDATA)>
<!ELEMENT condition (#PCDATA)>
<!ELEMENT path_list (src_paths,target_paths)>
<!ELEMENT src_paths (path)+>
<!ELEMENT target_paths (path)+>
<!ELEMENT path (int_entity,rest)>
<!ELEMENT rest (int_link,int_entity)+>
<!ELEMENT int_link (#PCDATA)>
<!ELEMENT int_entity (#PCDATA)>
<!ATTLIST int_entity correspondenceid CDATA #IMPLIED>
<!ATTLIST int_entity composingid CDATA #IMPLIED>
<!ATTLIST int_entity restriction CDATA #IMPLIED>
```

Figure 1: DTD of the ACGT Mapping Format

As can be seen, a mapping file is comprised by entries. An entry is the mapping between a view in the global schema and a view in the local schema. An entry also includes a natural language description of the mapped views (that are semantically equivalent). This description is used in the query tool to identify parts of user queries.

2.2.2. The Mapping Process

The Semantic Mediator relies on the data access wrappers to solve syntactic heterogeneities. Semantic heterogeneities, however, must be tackled in a different manner. The basic approach for this is employing a global schema to which the RDF Schemas of the wrappers to integrate are mapped. These mappings contain pairs of views of RDF Schemas with semantically equivalent views of the global schema, as described in the previous section. Queries in the mediator will be created in terms of the mentioned global schema. The information contained in the mappings will guide the translation process carried out for each of these queries.

The element used as global schema is an ontology. Ontologies have been used as frameworks in semantic mediation in many previous works [4] [5] [6] [7]. Concretely, the Semantic Mediator makes use of the ACGT Master Ontology —however, its design allows to use any ontology written in OWL-DL [8]. This ontology has been developed specifically for the ACGT project, and describes the cancer domain. More detailed information regarding the Master Ontology can be found in deliverable D7.2.

The creation of mappings between the Master Ontology and a given RDF Schema—usually entitled mapping process—requires of the collaboration of a set of experts in different domains. The profiles required for successfully carrying out this task are, namely: i) a data expert, who can explain the meaning of the fields of the database to be integrated, ii) an Master Ontology expert, who can help finding specific concepts in the ontology, and iii) a technology expert, who understands the details of the mapping building inputs. For this purpose, the ACGT Mapping Tool has been developed. The Mapping Tool is an online tool for creating mappings supporting concurrent input, thus allowing the required collaboration described previously. The input for this tool is the global schema (Master Ontology), and an RDF Schema. The output is a file containing the mapping of the given RDF Schema with the Master Ontology. Figure 2 depicts the Mapping Tool as a black box.

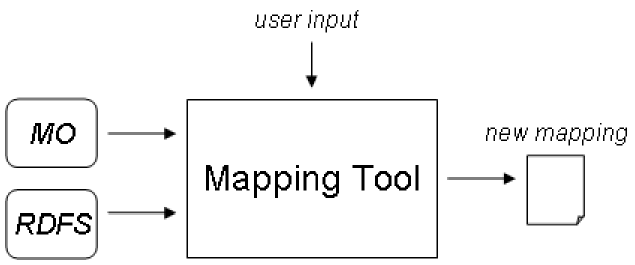


Fig 2. Process of building a new mapping through the Mapping Tool, seen as a black box

The Mapping Tool can be accessed through the ACGT Portal. At first, a user pretending to build a new mapping will have to create a new mapping session. She/he will automatically become its administrator. This enables several privileges, such as the possibility to invite other existing users to collaborate in the same session, or validate the mapping and submit it to the Semantic Mediator. After the proper session initialization, the users included in it will be able to start creating the mapping between the given RDF Schema and the Master Ontology. Graphical representation of these two schemas will let them navigate and construct the views that they consider semantically equivalent. The set of queries to be supported can be used in this step to identify such views.

Once the administrator of the session concludes that the mapping is correct and complete, he submits it to the Semantic Mediator. This involves the Mapping Tool making use of an available service of the Mapping Tool for updating the mapping information. Figure 3 shows this mechanism.

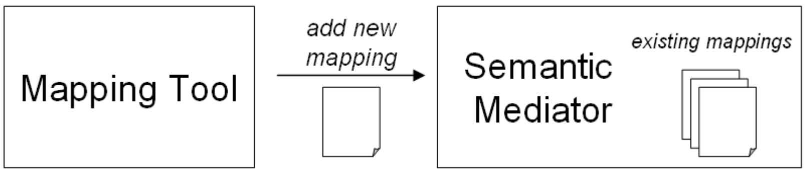


Fig 3. Addign a new mapping to the Semantic Mediator procedure, seen as a black box

As a result of this process, the Mapping Tool incorporates the new mapping file into its existing mappings. From that moment, queries performed in the Semantic Mediator are also redirected to the new data source

2.3. Query Translation Process

Query Translation is one of the core processes in Semantic Mediation and Database Integration. We understand as translation the process of, given a query expressed in terms of the global schema, producing a semantically equivalent set of queries for the mediated schemas. In ACGT, the original query is expressed in terms of the Master Ontology, and the Semantic Mediator produces a set of queries for the underlying Data Access Services.

The selected solution for Semantic Mediation and Database integration in ACGT is based in the query translation Local as View approach. This implies that no actual data are integrated before the execution of a query. This query is translated by mediation software, then sent to the corresponding databases (or database wrappers, in the case of ACGT). The results obtained are integrated to show to the user a set consistent with the original query. In Local as View no ad-hoc global schema is built to represent the possible queries. In ACGT, the Master Ontology, that has being developed with the aim of modelling the domain, independently from the mediation process, is used as Global Schema.

The construction of the queries that are sent to the Data Access Services is based in the analysis of three main sources of information, namely the original query, the Global Schema and the mapping information for each one of the wrappers. The query translation algorithm identifies which views inside the original query are of interest to obtain the final results and locates the corresponding information contained in the mapping in the final queries. This algorithm is show in detail in figure 4.

```
Given:
  • G the global schema
  • {$m_i$ | I ∈ [1..n]} a set of n Mappings
  • Q a query expressed in terms of G
  • {$s_i$ | I ∈ [1..n]} a set of empty queries
Do:
  ▪ P ← Views of interest in Q
  ▪ For each mapping $m_i$ do
      ▪ For each view p in P do
          ▪ P' ← all views subsumed by p in $m_i$
          ▪ Include views in P' in $s_i$
          ▪ Include variables of Q present in Views of P' in $s_i$
          ▪ Include split variables
      ▪ For each constraint c in Q
          ▪ if c has cross-reference then
              • Include cross-reference variables of Q in $s_i$
          ▪ else
              • include c in $s_i$
```

Fig 4: Query Translation Algorithm

As can be seen, the view is the semantic unit considered in the algorithm. A view is more expressive than a path, since a view is a set of paths with constraints. Finding mappings between views allows to overcome different semantic heterogeneities, and introduces "meaning" in the mapping, query formulation and query translation processes.

Another issue that needs to be solved when querying integrated repositories is the inconsistencies present in the identifiers. The OntoQueryClean has been specifically developed to deal with data inconsistencies contained in query literals. Due to different data formats coexisting in a mediation environment, a query translation process must take care of properly preprocessing any literal contained in an integrated query—i.e. in terms of a global schema. Not only values must be transformed, but also operators contained in restrictions are susceptible of requiring modifications. OntoQueryClean relies on the already described OntoDataClean tool to perform part of the work. It also makes use of a cleaning ontology which class instances describe the transformations to be performed over a data matrix. The data matrix is obtained from parsing SPARQL queries and extracting the literals contained in them. Figure 5 describes the basic functioning of the OntoQueryClean tool through its algorithm.

```
Given:
    • Q a SPARQL query
    • M a data matrix
    • CO a cleansing ontology
    • {$OPI_i$ | 0 < I < N} a set of Operator Preprocessing Instances in CO
Do:
    ▪ M ← literals contained in Q
    ▪ M ← preprocessed M using CO
    ▪ For each $OPI_i$ in CO do
        ▪ For each row $r_j$ in M do
            ▪ $r_j$ ← modify $r_j$ with $OPI_i$
Q ← include preprocessed literals from M in Q
Return Q
```

Fig 5. Algorithm of the OntoQueryClean tool

2.4. Data Integration Process

The integration of the data retrieved by the underlying databases to show the final user a consistent result set is the second key process in Semantic Mediation and Database Integration. When the user formulates the query, he wants to be retrieved with a coherent set of results, without taking care about the structure or even the existence of the integrated databases. To reach this end, both the expected results—i.e. information contained in the original query—and the results from the Data Access Services must be taking into consideration. Furthermore, results need to be pre-processed before being integrated in the central repository. This section describes both the data pre-processing approach and the data integration process.

OntoDataClean tool takes care of preprocessing the data contained in results retrieved from underlying databases. It makes use of an ontology-based approach in order to tackle the format inconsistencies that may be present in the mediation process. It takes as input a 2-dimensional data matrix containing the results to be preprocessed, and an instance of a cleaning ontology, which stores the cleaning methods that must be applied over the previous data matrix. Each instance of class of this ontology indicates a spefic cleaning method, and its properties store the details of the transformation to be applied. Figure 6 depicts the basic algorithm followed by OntoDataClean in order to, given a cleaning ontology, preprocess a data matrix containing raw results.

```
Given:
  • M a data matrix
  • CO a cleansing ontology
  • {$PI_i$ | 0 < I < N} a set of Preprocessing Instances in CO
Do:
  ▪ For each $PI_i$ in CO do
      ▪ For each row $r_j$ in M do
          ▪ $r_j \leftarrow$ modify $r_j$ with $PI_i$
Return M
```

Fig 6. Algorithm of the OntoDataClean tool

The Data Integration Process carried out by the mediator has the following inputs: 1) variables in the original query, 2) cross constraints in the original query, and 3) structure of the results retrieved by the wrappers. The system groups all the results retrieved by the Data Access Services, considering this set a database. These results, originally represented as SPARQL Result Set files, are stored in a single RDF repository. The system automatically generates a query for this RDF database, including the variables and cross constraints taken from the original query. Finally, this query is launched in the RDF repository, obtaining the final results.

3. Case Study: the Pseudo-TOP Scenario

This case study describes in detail the steps that are taken in order to integrate a new data repository into the ACGT Semantic Mediator. This process begins when a new data source—given by a new data access wrapper, an RDF Schema—is required to be integrated. At the end, the Semantic Mediator accepts queries for the new repository and retrieves results obtained from it. This process has been tested with the TOP Trial database, achieving successful results. Throughout this section, examples extracted from the integration of this database itself are presented.

The rest of the section is devoted to the description of each of the steps. Subsection 3.1 describes the inputs of the process. Subsection 3.2 shows how the Semantic Mediator deals with a specific example to illustrate the complete query generation and result integration mechanisms.

3.1. Process Input

The integration process of a new repository in the Semantic Mediator takes as input a data access wrapper to the repository—given itself by an URL and a Resource identifier—and an RDF Schema describing the set of accepted queries. This wrapper offers a homogeneous syntactic access to the data, as it hides its implementation details by providing a SPARQL [9] query mechanism to collect the data. This way, every database to integrate to the Semantic Mediator is exposed as an RDF [10] data repository. Figure 7 illustrates this mechanism.

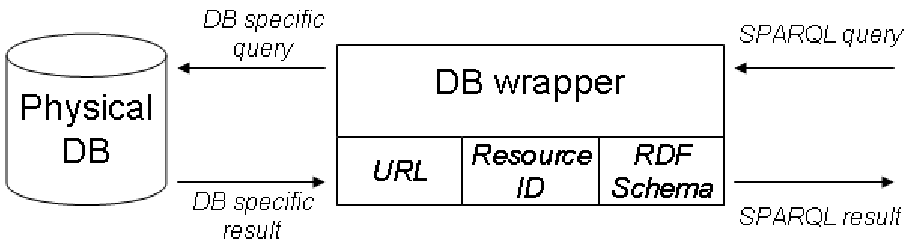


Fig 7. DB wrapper as a black box. Arrows indicate data flow

The RDF Schema of the wrapper developed for the TOP Trial database can be consulted in annex C.

In addition to the wrapper access information and the RDF Schema, a set of SPARQL queries to be supported is usually provided. These queries express direct user requirements for the final product of the integration process. This information, while not strictly necessary, gives very useful insights for the mapping creation process—as previously described in section 2.2.2. Figure 8 shows one of the queries employed as requirements during the integration of the TOP Trial database.

```
PREFIX vocab:
&lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;
PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;

SELECT ?pt ?birthDate ?regDate WHERE {
  ?patient vocab:patient_PT ?pt .
  ?demo vocab:PATIENT ?patient ;
        vocab:demo_BIRTH_DATE ?birthDate .
  ?rnd vocab:PATIENT ?patient ;
       vocab:rnd_RND_DATE ?regDate .
}
```

Fig 8. Queries are used as guideline during the mapping process

The complete set of queries is presented in annex B. The six SPARQL queries that it contains were used as a guideline during the integration of the TOP Trial database in the Semantic Mediator. This queries are used as expression of the user requirements to feed the mapping process (described in section 2.2.2).

3.2. Query execution

This section describes the interaction of the user creating and launching a query with the End User Interface and the tasks that the Semantic Mediator carries out when it receives such query. An example query is built using a graphical interface designed to offer high-level query creation capabilities and hide the details of the SPARQL language to non-technical users. The only mapping considered in this example is the one for the TOP Trial database wrapper.

3.3. Query Creation through the End User Interface

The ACGT End User Interface was specifically developed to offer high-level query capabilities over the Semantic Mediator. The goal was to allow non-technical users to easily create and perform queries for the Semantic Mediator, hiding—to some extent—the technical details of the SPARQL query language and the Master Ontology. The End User Interface is offered as an online tool through the ACGT Portal. It allows each registered user to save a set of preferred queries, as well as create new queries using natural language descriptions of specific parts of the Master Ontology.

In the example presented in this case study, a user pretends to retrieve the list of patients registered in the TOP Trial, together with their birth dates and registration dates. To do this, she/he selects the following four views from the available view list for the TOP database: i) “HumanBeings with their identifiers”, for retrieving the identifiers of the patients, ii) “HumanBeings with their birth dates”, for obtaining the birth dates of the patients, iii) “Registrations with their dates”, for retrieving the dates of the existing registrations, and iv) “Registrations with their HumanBeings”, for linking the existing registrations with the queried patients. Selecting these views and performing minor editing over the SPARQL query results in a query that can be executed in the mediator. This query, which is shown in figure 9, is semantically equivalent to the one shown in figure 8, but in terms of the Master Ontology.

```
PREFIX acgt:     <http://www.ifomis.org/acgt/1.0#>
PREFIX xsd:      <http://www.w3.org/2001/XMLSchema#>
PREFIX bfo_span: <http://www.ifomis.org/bfo/1.1/span#>

SELECT ?patientID ?birthDate ?regDate
WHERE {
  ?patient   acgt:hasIdentifier     ?patientID .
  ?patient   a                      acgt:HumanBeing .
  ?patientID a                      xsd:string .
  ?patient2  acgt:hasBirthDate      ?birthDate .
  ?patient2  a                      acgt:HumanBeing .
  ?birthDate a                      xsd:date .
  ?registration2 acgt:hasProcessEnd ?endOfReg .
  ?endOfReg      acgt:hasDate       ?regDate .
  ?registration2 a                  acgt:Registration .
  ?endOfReg      a                  bfo_span:ProcessBoundary .
  ?regDate       a                  xsd:date .
  ?registration  acgt:hasPatient    ?patient3 .
  ?registration  a                  acgt:Registration .
  ?patient3      a                  acgt:HumanBeing .
  FILTER ( ?patient = ?patient2 )
  FILTER ( ?patient = ?patient3 )
  FILTER ( ?registration = ?registration2 )
}
```

Fig 9. SPARQL query in terms of the Master Ontology created using the End User Interface

When the user click on the Submit button, the End User Interface generates a perform document that suits the requirements of the OGSADAI service in which the Semantic Mediator is hosted. This perform document is an XML file describing the activities request for the Semantic Mediator. Figure 10 shows such document.

```
<?xml version="1.0" encoding="UTF-8"?>
<perform xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
    <documentation>
      Simple query across all tables in the database.
    </documentation>
    <SemanticMediatorWIP name="myActivityInstance">
        <query>
PREFIX acgt:     <http://www.ifomis.org/acgt/1.0#>
PREFIX xsd:      <http://www.w3.org/2001/XMLSchema#>
PREFIX bfo_span: <http://www.ifomis.org/bfo/1.1/span#>

SELECT ?patientID ?birthDate ?regDate
WHERE {
  ?patient   acgt:hasIdentifier     ?patientID .
  ?patient   a                      acgt:HumanBeing .
  ?patientID a                      xsd:string .
  ?patient2  acgt:hasBirthDate      ?birthDate .
  ?patient2  a                      acgt:HumanBeing .
  ?birthDate a                      xsd:date .
  ?registration2 acgt:hasProcessEnd ?endOfReg .
  ?endOfReg      acgt:hasDate       ?regDate .
  ?registration2 a                  acgt:Registration .
  ?endOfReg      a                  bfo_span:ProcessBoundary .
  ?regDate       a                  xsd:date .
  ?registration  acgt:hasPatient    ?patient3 .
  ?registration  a                  acgt:Registration .
  ?patient3      a                  acgt:HumanBeing .
  FILTER ( ?patient = ?patient2 )
  FILTER ( ?patient = ?patient3 )
  FILTER ( ?registration = ?registration2 )
}
        </query>
        <SemanticMediatorOutput name="sparqlResults"/>
    </SemanticMediatorWIP>
</perform>
```

Fig 10. Perform document that the OGSADAI sevice hosting the Semantic Mediator receives

The OGSADAI service of the Semantic Mediator is invoked using the perform document. During the execution of the Semantic Mediator, the End User Interface waits for a response prior to show the final results to the user.

3.4. Query Translation

When the Semantic Mediator receives an integrated query—i.e. in terms of the Master Ontology—it constructs an equivalent query for each of the databases it integrates. This process involves itself the following steps:

- Query parsing: the query, received as a string, is parsed and stored into an internal representation.
- View generation: given a parsed query, generate all combinations of views that it contains.
- Subsumed views search: for each of the previously generated views, search for subsumed views in each of the existing mappings. Every subsumed view found is kept for subsequent generation of each of the subqueries. These views represent the parts of the original query that were found to have a translation in the mapping for a specific data wrapper.
- View translation: given the set of views that have translation for each query, use the mappings to translate them
- Query generation: given the translations of the previous views, merge them in order to create a complete query in terms of the underlying database wrapper.
- Query preprocessing: the generated queries are preprocessed by the OntoQueryClean tool. Any literals contained in the queries are transformed in order to adjust to the format employed in the physical database.
- Query optimization: optimize the generated queries in order to eliminate possible redundancy and reduce variable count.

Given the example query shown in figure 9, the query parsing process results in an internal representation depicted in figure 11—implementation details do not necessarily match this picture.

```
VOCABULARIES
    acgt=<http://www.ifomis.org/acgt/1.0#>
    xsd=<http://www.w3.org/2001/XMLSchema#>
    bfo_span=<http://www.ifomis.org/bfo/1.1/span#>

SELECT:
    patientID
    birthDate
    regDate

TRIPLES
  {patient, acgt:hasIdentifier, patientID}
  {patient2, acgt:hasBirthDate, birthDate}
  {registration2, acgt:hasProcessEnd, endOfReg}
  {endOfReg, acgt:hasDate, regDate}
  {registration, acgt:hasPatient, patient3}

CLASS_RESTRICTIONS
  {patient → acgt:HumanBeing}
  {patientID → xsd:string}
  {patient2 → acgt:HumanBeing}
  {birthDate → xsd:date}
  {registration2 → acgt:Registration}
  {endOfReg → bfo_span:ProcessBoundary}
  {regDate → xsd:date}
  {registration → acgt:Registration}
  {patient3 → acgt:HumanBeing}

FILTERS
  {patient = patient2}
  {patient = patient3}
  {registration = registration2}
}
```

Fig 11. Internal representation of the query received by the Semantic Mediator

Using this parsed query as input, the Semantic Mediator generates all possible views from it. For efficiency purposes, triples linked with the same variable are considered to form indivisible paths. This way the view generation is reduced to a random path combination. Figure 12 shows all generated views for the previous query—with a path count up to two.

```
V0 = {patient → acgt:hasIdentifier → patientID}

V1 = {patient2 → acgt:hasBirthDate → birthDate}

V2 = {registration2 → acgt:hasProcessEnd → endOfReg →
      acgt:hasDate → regDate}

V3 = {registration → acgt:hasPatient → patient3}

V3 = {patient → acgt:hasIdentifier → patientID}
     {patient2 → acgt:hasBirthDate → birthDate}

V4 = {patient → acgt:hasIdentifier → patientID}
     {registration2 → acgt:hasProcessEnd → endOfReg →
      acgt:hasDate → regDate}

V5 = {patient → acgt:hasIdentifier → patientID}
     {registration → acgt:hasPatient → patient3}

V6 = {patient2 → acgt:hasBirthDate → birthDate}
     {registration2 → acgt:hasProcessEnd → endOfReg →
      acgt:hasDate → regDate}

V7 = {patient2 → acgt:hasBirthDate → birthDate}
     {registration → acgt:hasPatient → patient3}

V6 = {registration2 → acgt:hasProcessEnd → endOfReg →
      acgt:hasDate → regDate}
     {registration → acgt:hasPatient → patient3}
```

Fig 12. Random view generation from the paths contained in the query. Only views with a path count up to two are shown

For each of these views, a search for subsumed views in the mapping of TOP is performed. In this case, the first four views return one result each. The found views are exactly the ones contained in the query. These are translated using the information contained in the mapping. These translations are shown in figure 13.

```
VIEW 0:
{vocab_0=http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#}
?var_0 vocab_0:patient_PT ?var_1 .
?var_0 a vocab_0:patient .

VIEW 1:
{vocab_1=http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#}
?var_2 vocab_1:PATIENT ?var_3 .
?var_2 vocab_1:demo_BIRTH_DATE ?var_4 .
?var_2 a vocab_1:demo .
?var_3 a vocab_1:patient .

VIEW 2:
{vocab_2=http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#}
?var_5 vocab_2:rnd_RND_DATE ?var_6 .
?var_5 a vocab_2:rnd .

VIEW 3:
{vocab_3=http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#}
?var_7 vocab_3:PATIENT ?var_8 .
?var_7 a vocab_3:rnd .
?var_8 a vocab_3:patient .
```

Fig 13. Translations of the found subsumed views in the mapping

These obtained parts of SPARQL queries are merged together, and the missing information—i.e. SELECT section and FILTERs for linking variables—added. The result, depicted in figure 14, is a complete SPARQL query in terms of the RDF Schema of the wrapper for the TOP Trial database.

```
PREFIX vocab_0:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_1:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_2:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_3:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
SELECT ?var_1 ?var_4 ?var_6
WHERE {
?var_0 vocab_0:patient_PT ?var_1 .
?var_0 a vocab_0:patient .
?var_2 vocab_1:PATIENT ?var_3 .
?var_2 vocab_1:demo_BIRTH_DATE ?var_4 .
?var_2 a vocab_1:demo .
?var_3 a vocab_1:patient .
?var_5 vocab_2:rnd_RND_DATE ?var_6 .
?var_5 a vocab_2:rnd .
?var_7 vocab_3:PATIENT ?var_8 .
?var_7 a vocab_3:rnd .
?var_8 a vocab_3:patient .

FILTER ( ?var_0 = ?var_3 )
FILTER ( ?var_0 = ?var_8 )
FILTER ( ?var_5 = ?var_7 )
}
```

Fig 14. Generated query from the previous four view translations

While this query is conceptually correct, optimizations can be performed in order to reduce the time the wrappers will take to compute it. The goal in this step is to produce a query semantically equivalent to the generated one, but that minimizes the wrappers response time. To do this, two actions are taken: i) variable count is reduced by merging equivalent variables, and ii) redundant triples are eliminated. Equivalent variables are those whose values are set to be equal by a FILTER restriction. In this case, one of the variable names is dropped in favour of the other. Transitivity must be taken into account to perform this optimization correctly. On the other hand, the translation of several views that overlap and the subsequent variable merging might produce redundant triples—information is simply repeated. In this case, the unnecessary triples are erased. In the generated query, the only possible optimization is reduction of variable count. The result after this process is shown in figure 15.

```
PREFIX vocab_0:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_1:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_2:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
PREFIX vocab_3:
<http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#>
SELECT ?var_1 ?var_4 ?var_6
WHERE {
?var_0 vocab_0:patient_PT ?var_1 .
?var_0 a vocab_0:patient .
?var_2 vocab_1:PATIENT ?var_0 .
?var_2 vocab_1:demo_BIRTH_DATE ?var_4 .
?var_2 a vocab_1:demo .
?var_5 vocab_2:rnd_RND_DATE ?var_6 .
?var_5 a vocab_2:rnd .
?var_5 vocab_3:PATIENT ?var_0 .
}
```

Fig 15. Query after the optimization process. Variables var_3, var_7 and var_8 have been eliminated

This is the final query that is performed against the wrapper of the TOP Trial database. The Semantic Mediator invokes the corresponding OGSADAI service and waits until a set of results is returned.

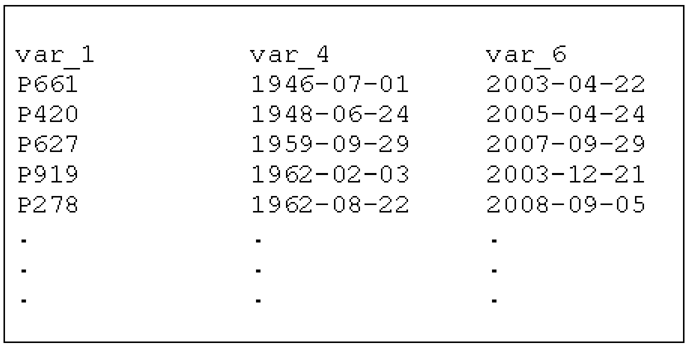
3.5. Result Integration

Data access wrappers return results in SPARQL Result Format [11]. In general, the Semantic Mediator obtains a set of results for each generated query. In the end, a single result set in CSV format must be retrieved to the user. To achieve this, the following steps are carried out:

- Each result obtained from the wrappers is parsed and stored into an internal representation.
- The parsed results are preprocessed by the OntoDataClean tool. Literals are modified in order to adjust to the format adopted by the Semantic Mediator—obtaining homogeneous results—, and variable names are modified to the terms of the original query.
- The complete set of results is analyzed and a new RDF document containing all the data together is automatically generated.
- A new SPARQL query containing the restrictions that involved variables from different wrappers is automatically generated and performed against the previous RDF document. This process generates a new SPARQL result document which contains the results of the original query.
- The definitive result document is parsed and an equivalent CSV document is generated.

- A metadata file for the results is generated.
- Both the results file and the metadata file are stored in a database. Links for obtaining them are returned as result of the query.

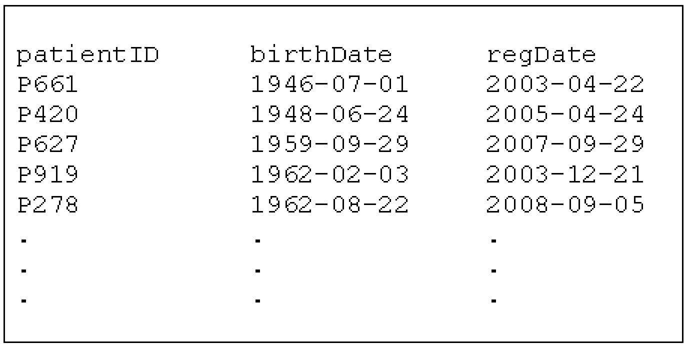
For the specific query generated for the TOP Trial database wrapper, there are 197 rows obtained. Figure 16 shows the first lines of these results—each variable is displayed in a column.



```
var_1          var_4          var_6
P661           1946-07-01     2003-04-22
P420           1948-06-24     2005-04-24
P627           1959-09-29     2007-09-29
P919           1962-02-03     2003-12-21
P278           1962-08-22     2008-09-05
.              .              .
.              .              .
.              .              .
```

Fig 16. Results returned by the TOP Trial database wrapper

The data format adopted by the Semantic Mediator for this scenario was simply the format employed by the TOP Trial database. Therefore, result preprocessing consists simply in properly renaming the variables. Figure 17 depicts the same results after being preprocessed.



```
patientID      birthDate      regDate
P661           1946-07-01     2003-04-22
P420           1948-06-24     2005-04-24
P627           1959-09-29     2007-09-29
P919           1962-02-03     2003-12-21
P278           1962-08-22     2008-09-05
.              .              .
.              .              .
.              .              .
```

Fig 17. Results after preprocessing of variable names

This result set is used as input for generating the RDF document that stores the complete set of results. Figure 18 describes such document for this case. Annex A contains the complete RDF document—in order to maintain a reasonable size, only the first five rows of results are considered.

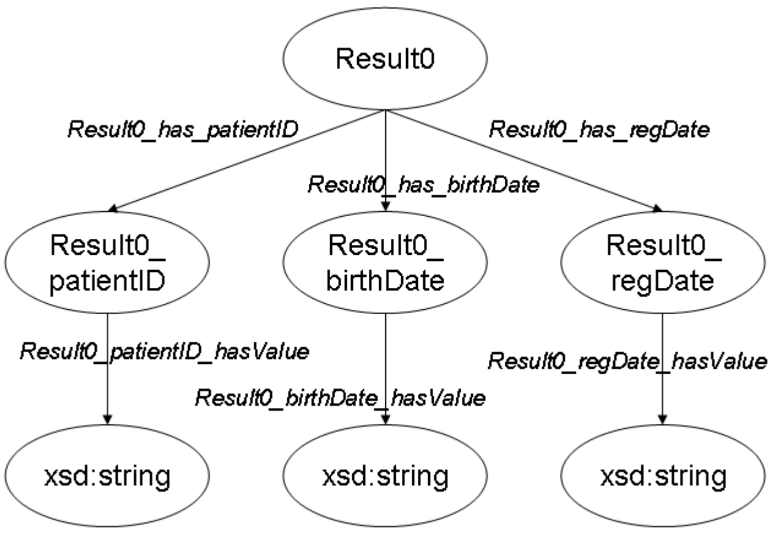


Fig 18. Graphical representation of the RDF Document storing the integration of the results. There is one instance of the Result0 class, and for each row of results, one instance of Result0_patientID, Result0_patientID and Result0_patientID are created.

Since no restriction involving variables from different database wrappers was contained in the original query, the SPARQL query for retrieving the final results simply retrieves the values stored in the previous RDF document, producing a result set equivalent to that shown in figure 16. Figure 19 shows this SPARQL query.

```
PREFIX v: <http://result_rdf#>
SELECT ?patientID ?birthDate ?regDate
WHERE {
    ?Result0 v:Result0_has_patientID ?patientID_ .
    ?Result0 v:Result0_has_birthDate ?birthDate_ .
    ?Result0 v:Result0_has_regDate ?regDate_ .
    ?patientID_ v:Result0_patientID_hasValue ?patientID .
    ?birthDate_ v:Result0_patientID_hasValue ?birthDate.
    ?regDate_ v:Result0_patientID_hasValue ?regDate.
}
```

Fig 19. SPARQL query for retrieving the final results

Finally, a CSV document is generated from the previous results. In addition to these results, a metadata file describing them must be generated. This metadata file consists on an RDF document containing the Master Ontology classes to which the retrieved variables belong. If for one variable results from more than one class in the Master Ontology were obtained, the common ancestor of these classes is used. For each variable, an individual with its name is created. This individual is an instance of the class to which the variable results belong. This way, the user gets a description in terms of the Master Ontology of the results obtained. Figure 20 depicts the RDF document for the results obtained.

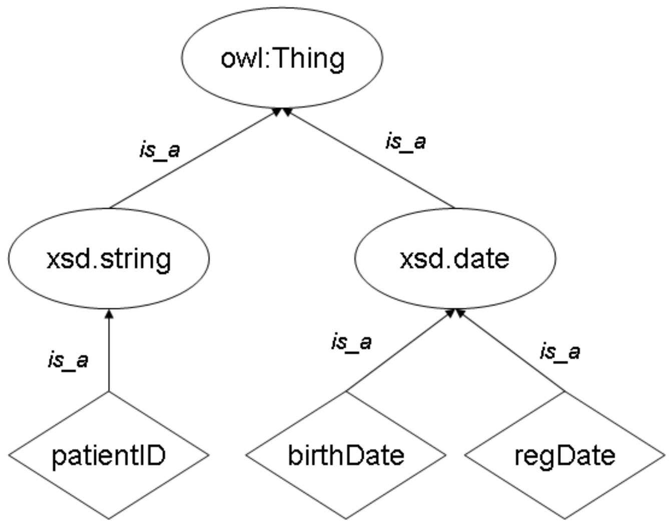


Fig 20. Graphical representation of the generated metadata file. In this case the variables retrived do not belong to classes of the Master Ontology, but are datatypes. Classes matching the datatype names are created

The CSV-formatted file containing the results of the original query, along with the related metadata document are both stored in a database with online access.

References

[1] Sujansky W. Heterogeneous database integration in biomedicine. J Biomed Inform, 2001; 34(4):285–98.

[2] Chawathe S, García-Molina H, Hammer J, Ireland K, Papakonstantinou Y, Ullman I, Widom J. The TSIMMIS project: Integration of Heterogeneous Information Sources. In Proceedings of the 10th Meeting of the Information Processing Society of Japan, Tokyo, Japan, October 1994; 7-18.

[3] Levy AY, Rajaraman A, Ordille JJ. Querying heterogeneous information sources using source descriptions. In Proceedings of the Twenty-second Internacional Conference on Very Large Databases (VLDB'96), Mumbai (Bombai), India, September 1996; 251-62.

[4] Perez-Rey, D., Maojo, V., Garcia-Remesal, M., Alonso-Calvo, R., Billhardt, H., Martin-Sanchez, F., Sousa, A.: ONTOFUSION: Ontology-Based Integration of Genomic and Clinical Databases. Computers in Biology and Medicine 36, 712--730 (2006)

[5] Kohler, J., Philippi, S., Lange, M.: Semeda: ontology based semantic integration of biological databases. Bioinformatics 19 (18), 2420--2427 (2003)

[6] Librelotto, G.R., Souza, W., Armalo, J.C., Henriques, P.R.: Using the Ontology Paradigm to Integrate Information Systems. Proceedings of the International Conference on Knowledge Engineering and Decision Support, pp. 497--504, Porto (2004)

[7] Bizer, C.: D2R MAP - A Database to RDF Mapping Language. Proceedings of the International World Wide Web Conference, Budapest (2003)

[8] OWL Web Ontology Language. Available at: http://www.w3.org/TR/owl-features/

[9] SPARQL Query Language for RDF. Available at: http://www.w3.org/TR/rdf-sparql-query/

[10] Resource Description Framework (RDF). Available at: http://www.w3.org/RDF/

[11] SPARQL Query Results XML Format. Available at: http://www.w3.org/TR/rdf-sparql-XMLres/

ANNEX A: RDF Document storing the integration of results

```
<?xml version="1.0"?>

<rdf:RDF

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns="http://result_rdf#"

    xml:base="http://result_rdf">

    <rdfs:Class rdf:ID="Result0_regDate">

        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

    </rdfs:Class>

    <rdfs:Class rdf:ID="Result0_birthDate">

        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

    </rdfs:Class>

    <rdfs:Class rdf:ID="Result0">

        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

    </rdfs:Class>

    <rdfs:Class rdf:ID="Result0_patientID">

        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>

    </rdfs:Class>

    <rdf:Description rdf:ID="Result0_has_birthDate">

        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0"/>

        <rdfs:range rdf:resource="#Result0_birthDate"/>

    </rdf:Description>

    <rdf:Description rdf:ID="Result0_birthDate_hasValue">

        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0_birthDate"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>

    </rdf:Description>

    <rdf:Description rdf:ID="Result0_patientID_hasValue">
```

```
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0_patientID"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

    </rdf:Description>

    <rdf:Description rdf:ID="Result0_regDate_hasValue">

        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0_regDate"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>

    </rdf:Description>

    <rdf:Description rdf:ID="Result0_has_regDate">

        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0"/>

        <rdfs:range rdf:resource="#Result0_regDate"/>

    </rdf:Description>

    <rdf:Description rdf:ID="Result0_has_patientID">

        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>

        <rdfs:domain rdf:resource="#Result0"/>

        <rdfs:range rdf:resource="#Result0_patientID"/>

    </rdf:Description>

    <Result0_birthDate rdf:ID="i14">

        <Result0_birthDate_hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1962-
02-03</Result0_birthDate_hasValue>

    </Result0_birthDate>

    <Result0_patientID rdf:ID="i13">

        <Result0_patientID_hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P919</Result0_patientID_hasValue>

    </Result0_patientID>

    <Result0 rdf:ID="i16">

        <Result0_has_birthDate rdf:resource="#i18"/>

        <Result0_has_regDate rdf:resource="#i19"/>

        <Result0_has_patientID rdf:resource="#i17"/>

    </Result0>
    <Result0_regDate rdf:ID="i15">

        <Result0_regDate_hasValue   rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2003-
12-21</Result0_regDate_hasValue>

    </Result0_regDate>

    <Result0_birthDate rdf:ID="i10">

        <Result0_birthDate_hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1959-
09-29</Result0_birthDate_hasValue>

    </Result0_birthDate>

    <Result0 rdf:ID="i12">

        <Result0_has_birthDate rdf:resource="#i14"/>

        <Result0_has_regDate rdf:resource="#i15"/>

        <Result0_has_patientID rdf:resource="#i13"/>

    </Result0>

    <Result0_regDate rdf:ID="i11">

        <Result0_regDate_hasValue   rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2007-
09-29</Result0_regDate_hasValue>

    </Result0_regDate>

    <Result0_patientID rdf:ID="i1">

        <Result0_patientID_hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P661</Result0_patientID_hasValue>

    </Result0_patientID>

    <Result0 rdf:ID="i0">

        <Result0_has_birthDate rdf:resource="#i2"/>

        <Result0_has_regDate rdf:resource="#i3"/>

        <Result0_has_patientID rdf:resource="#i1"/>

    </Result0>

    <Result0_patientID rdf:ID="i9">

        <Result0_patientID_hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P627</Result0_patientID_hasValue>

    </Result0_patientID>

    <Result0 rdf:ID="i8">

        <Result0_has_birthDate rdf:resource="#i10"/>

        <Result0_has_regDate rdf:resource="#i11"/>

        <Result0_has_patientID rdf:resource="#i9"/>
    </Result0>

    <Result0_regDate rdf:ID="i7">

        <Result0_regDate_hasValue    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2005-
04-24</Result0_regDate_hasValue>

    </Result0_regDate>

    <Result0_birthDate rdf:ID="i6">

        <Result0_birthDate_hasValue  rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1948-
06-24</Result0_birthDate_hasValue>

    </Result0_birthDate>

    <Result0_patientID rdf:ID="i17">

        <Result0_patientID_hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P278</Result0_patientID_hasValue>

    </Result0_patientID>

    <Result0_patientID rdf:ID="i5">

        <Result0_patientID_hasValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P420</Result0_patientID_hasValue>

    </Result0_patientID>

    <Result0_birthDate rdf:ID="i18">

        <Result0_birthDate_hasValue  rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1962-
08-22</Result0_birthDate_hasValue>

    </Result0_birthDate>

    <Result0 rdf:ID="i4">

        <Result0_has_birthDate rdf:resource="#i6"/>

        <Result0_has_regDate rdf:resource="#i7"/>

        <Result0_has_patientID rdf:resource="#i5"/>

    </Result0>

    <Result0_regDate rdf:ID="i19">

        <Result0_regDate_hasValue    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2008-
09-05</Result0_regDate_hasValue>

    </Result0_regDate>

    <Result0_regDate rdf:ID="i3">

        <Result0_regDate_hasValue    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2003-
04-22</Result0_regDate_hasValue>

    </Result0_regDate>

    <Result0_birthDate rdf:ID="i2">
```

```
        <Result0_birthDate_hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1946-
07-01</Result0_birthDate_hasValue>

    </Result0_birthDate>

</rdf:RDF>
```

ANNEX B: SPARQL requirement queries for the integration of TOP Trial database with the Semantic Mediator

```
Query 1:

--------

PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;

PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?pt ?birthDate ?regDate WHERE {

  ?patient vocab:patient_PT ?pt .

  ?demo vocab:PATIENT ?patient ;

        vocab:demo_BIRTH_DATE ?birthDate .

  ?rnd vocab:PATIENT ?patient ;

       vocab:rnd_RND_DATE ?regDate .

}


Query 2:

--------

PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;

PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?pt ?grade WHERE {

  ?patient vocab:patient_PT ?pt .

  ?dcmsub vocab:dcmsub_DCMSUBNM "CANC" .

  ?canc vocab:PATIENT ?patient ;

        vocab:DCMSUB ?dcmsub ;

        vocab:canc_GRADE ?grade .

}


Query 3:
```

```
--------

PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;

PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?pt ?assesResult WHERE {

  ?patient vocab:patient_PT ?pt .

  ?dcmsub vocab:dcmsub_DCMSUBNM "ASES1" .

  ?subevent vocab:subevent_EVENT ?event .

  ?event vocab:event_CPEVENT "SCREENING" .

  ?ases vocab:PATIENT ?patient ;

        vocab:DCMSUB ?dcmsub ;

        vocab:ACTEVENT ?subevent ;

        vocab:ases_ASSES_RSLT ?assesResult .

}


Query 4:

--------


PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;

PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?pt ?mtd WHERE {

  ?patient vocab:patient_PT ?pt .

  ?mtd vocab:PATIENT ?patient ;

       a vocab:mtd .

}


Query 5:

--------


PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;
PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?patient ?tumorWidth ?tumorHeight ?pcrYN {

  ?patient a acgt:HumanBeing ;

           acgt:undergoes ?chemotherapy .

  ?neoplasm a acgt:Neoplasm ;

            acgt:undergoes ?clinicalExamination ;

            acgt:partOf ?patient ;

            acgt:undergoes ?chemotherapy .

  ?chemotherapy a acgt:ChemoTherapy ;

                acgt:precedes ?tumorDiagnosis .

  ?tumorDiagnosis a acgt:DiagnosticProcess ;

                  acgt:measures ?widthQuality ;

                  acgt:measures ?heightQuality ;

                  acgt:reveals ?pcrStatus .

  ?widthQuality a acgt:Width ;

                acgt:hasFloatValue ?tumorWidth .

  ?heigthQuality a acgt:Height ;

                acgt:hasFloatValue ?tumorHeight .

  ?pcrStatus a acgt:pCRStatus ;

             acgt:hasBoleanValue ?pcrYN .

}


Query 6:

--------


PREFIX vocab: &lt;http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#&gt;

PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt;


SELECT ?pt ?geoid WHERE {

  ?patient vocab:patient_PT ?pt .
  ?map vocab:PATIENT ?patient ;

       vocab:ptop_ids_geoid ?geoid .

}
```

ANNEX C: RDF Schema of the TOP Trial database wrapper

```
<rdf:RDF

      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

      xmlns:code="http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/code#"

      xmlns:vocab="http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab#"

      xml:base="http://gridnode.ehv.campus.philips.com/temp/TOP_CRF_Norm/1.0/vocab">

  <rdfs:Class rdf:ID="CANC">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="dcm"/>

  <rdfs:Class rdf:ID="elig">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="dcmsub"/>

  <rdfs:Class rdf:ID="event"/>

  <rdfs:Class rdf:ID="ptop_clinfactors"/>

  <rdfs:Class rdf:ID="tr2">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="code_proc_type"/>

  <rdfs:Class rdf:ID="book"/>

  <rdfs:Class rdf:ID="diag">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="ae">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>
```

```
 <rdfs:Class rdf:ID="tr3">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="tr4">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="tr5">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="tr6">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="cmpl">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="canc2">

   <rdfs:subClassOf rdf:resource="#CANC"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="canc1">

   <rdfs:subClassOf rdf:resource="#CANC"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="rnd">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="fu">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="mtd">

   <rdfs:subClassOf rdf:resource="#DCM_View"/>

 </rdfs:Class>

 <rdfs:Class rdf:ID="demo">
```

```
    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="DCM_View"/>

  <rdfs:Class rdf:ID="patient"/>

  <rdfs:Class rdf:ID="tr">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="subevent"/>

  <rdfs:Class rdf:ID="chst">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>

  <rdfs:Class rdf:ID="ptop_ids"/>

  <rdfs:Class rdf:ID="code_asses_type"/>

  <rdfs:Class rdf:ID="ases">

    <rdfs:subClassOf rdf:resource="#DCM_View"/>

  </rdfs:Class>



  <rdf:Property rdf:ID="diag_DIAG_BEG_DATE">

    <rdfs:domain rdf:resource="#diag"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="demo_SUBJ_INIT">

    <rdfs:domain rdf:resource="#demo"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_SPM_DATE">

    <rdfs:domain rdf:resource="#fu"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="chst_TRUCUT_NO">

    <rdfs:domain rdf:resource="#chst"/>

  </rdf:Property>
  <rdf:Property rdf:ID="dcmsub_DCMSUBNM">

    <rdfs:domain rdf:resource="#dcmsub"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr3_TR_YN">

    <rdfs:domain rdf:resource="#tr3"/>

  </rdf:Property>

  <rdf:Property rdf:ID="cmpl_CYCLE_NO">

    <rdfs:domain rdf:resource="#cmpl"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_CARC_TYPE_YN1">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr3_TR3_TXEN">

    <rdfs:domain rdf:resource="#tr3"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_CARC_TYPE_YN2">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_AE_ONG">

    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr6_TRIAL_YN">

    <rdfs:domain rdf:resource="#tr6"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_MOD_TYPE">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="book_FIRST_BOOK_PAGE">

    <rdfs:domain rdf:resource="#book"/>

    <rdfs:range rdf:resource="&xsd;int"/>
  </rdf:Property>

  <rdf:Property rdf:ID="elig_ELI_YN">

    <rdfs:domain rdf:resource="#elig"/>

  </rdf:Property>

  <rdf:Property rdf:ID="cmpl_DOSE">

    <rdfs:domain rdf:resource="#cmpl"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_PRTP_YN">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_DIAG_YN">

    <rdfs:domain rdf:resource="#ases"/>

  </rdf:Property>

  <rdf:Property rdf:ID="cmpl_DENSE_YN">

    <rdfs:domain rdf:resource="#cmpl"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_SPM_SP">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_OBS_COM1">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="dcm_DCMNAME">

    <rdfs:domain rdf:resource="#dcm"/>

  </rdf:Property>

  <rdf:Property rdf:ID="cmpl_CMPL_RSP">

    <rdfs:domain rdf:resource="#cmpl"/>

  </rdf:Property>

  <rdf:Property rdf:ID="cmpl_CMPL_REAS">

    <rdfs:domain rdf:resource="#cmpl"/>
  </rdf:Property>

  <rdf:Property rdf:ID="ACTEVENT">

    <rdfs:domain rdf:resource="#DCM_View"/>

    <rdfs:range rdf:resource="#subevent"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_OBS_COM2">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_GCSF_DOSE">

    <rdfs:domain rdf:resource="#mtd"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_TR_YN">

    <rdfs:domain rdf:resource="#tr4"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_OBS_COM3">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_TR4_TXEN">

    <rdfs:domain rdf:resource="#tr4"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_TK_RSP">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_ASSES_TYPE">

    <rdfs:domain rdf:resource="#ases"/>

    <rdfs:range rdf:resource="#code_asses_type"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PT_PATH">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>
  <rdf:Property rdf:ID="event_VISIT_NUMBER">

    <rdfs:domain rdf:resource="#event"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_TR_TXEN_YN">

    <rdfs:domain rdf:resource="#tr4"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_DEATH_TYPE">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr5_TRIAL_YN">

    <rdfs:domain rdf:resource="#tr5"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_SER_YN">

    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr5_TR_YN">

    <rdfs:domain rdf:resource="#tr5"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PT0_YN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_TUMOR_SIZE1">

    <rdfs:domain rdf:resource="#canc2"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PN_PATH">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="dcmsub_SUBSETSN">

    <rdfs:domain rdf:resource="#dcmsub"/>
    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_TUMOR_SIZE2">

    <rdfs:domain rdf:resource="#canc2"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc1_PN_CLIN">

    <rdfs:domain rdf:resource="#canc1"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr5_CRF_NO">

    <rdfs:domain rdf:resource="#tr5"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_CYCLE_NO">

    <rdfs:domain rdf:resource="#tr4"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="patient_PATIENT_POSITION_ID">

    <rdfs:domain rdf:resource="#patient"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_SITE">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc1_LAB_RSLT1">

    <rdfs:domain rdf:resource="#canc1"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_FOCAL_YN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_GCSF_BEG_DATE">

    <rdfs:domain rdf:resource="#mtd"/>
    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc1_LAB_RSLT2">

    <rdfs:domain rdf:resource="#canc1"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc1_PM">

    <rdfs:domain rdf:resource="#canc1"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_AE_GRADE">

    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_ids_tbid">

    <rdfs:domain rdf:resource="#ptop_ids"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr6_TR_YN">

    <rdfs:domain rdf:resource="#tr6"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_TR_SP">

    <rdfs:domain rdf:resource="#tr4"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_LYNOD_POS">

    <rdfs:domain rdf:resource="#canc2"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_RES_STAT">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_PROC_DATE">

    <rdfs:domain rdf:resource="#ases"/>

    <rdfs:range rdf:resource="&xsd;date"/>
  </rdf:Property>

  <rdf:Property rdf:ID="chst_SITE">

    <rdfs:domain rdf:resource="#chst"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_DIAG_SP">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="subevent_SUBEVENT_NUMBER">

    <rdfs:domain rdf:resource="#subevent"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_DEATH_DATE">

    <rdfs:domain rdf:resource="#fu"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_INV_SIGN">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="patient_PT">

    <rdfs:domain rdf:resource="#patient"/>

  </rdf:Property>

  <rdf:Property rdf:ID="PATIENT">

    <rdfs:domain rdf:resource="#DCM_View"/>

    <rdfs:range rdf:resource="#patient"/>

  </rdf:Property>

  <rdf:Property rdf:ID="subevent_ACTEVENT">

    <rdfs:domain rdf:resource="#subevent"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr4_DOSE">

    <rdfs:domain rdf:resource="#tr4"/>
    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_AE_BEG_DATE">

    <rdfs:domain rdf:resource="#ae"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_clinfactors_ihc_top2a">

    <rdfs:domain rdf:resource="#ptop_clinfactors"/>

    <rdfs:range rdf:resource="&xsd;byte"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_MOD_RSP">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_DEATH_SP">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr_TR_BEG_DATE">

    <rdfs:domain rdf:resource="#tr"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="demo_HOSP_NO">

    <rdfs:domain rdf:resource="#demo"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_MT_REL">

    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="demo_BIRTH_DATE">

    <rdfs:domain rdf:resource="#demo"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_CENTRIC_YN">
```

```
    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_NPRTP_RSP">

    <rdfs:domain rdf:resource="#rnd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_FU_SP">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_PROC_TYPE">

    <rdfs:domain rdf:resource="#ases"/>

    <rdfs:range rdf:resource="#code_proc_type"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_TK_YN">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_clinfactors_ggi_score">

    <rdfs:domain rdf:resource="#ptop_clinfactors"/>

    <rdfs:range rdf:resource="&xsd;float"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_DIAG_YN1">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_DIAG_YN3">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="event_CPEVENT">

    <rdfs:domain rdf:resource="#event"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_DIAG_YN2">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>
  <rdf:Property rdf:ID="canc1_DIAG_BEG_DATE">

    <rdfs:domain rdf:resource="#canc1"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_RECR_DATE">

    <rdfs:domain rdf:resource="#fu"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="diag_DIAG_CMPL_ONG">

    <rdfs:domain rdf:resource="#diag"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr2_TR_TXEN_YN">

    <rdfs:domain rdf:resource="#tr2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PN0_YN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_TK_TYPE">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PCR_YN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="subevent_EVENT">

    <rdfs:domain rdf:resource="#subevent"/>

    <rdfs:range rdf:resource="#event"/>

  </rdf:Property>

  <rdf:Property rdf:ID="dcmsub_DCM">

    <rdfs:domain rdf:resource="#dcmsub"/>

    <rdfs:range rdf:resource="#dcm"/>

  </rdf:Property>
  <rdf:Property rdf:ID="DCMSUB">

    <rdfs:domain rdf:resource="#DCM_View"/>

    <rdfs:range rdf:resource="#dcmsub"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_DOSE">

    <rdfs:domain rdf:resource="#mtd"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="rnd_RND_DATE">

    <rdfs:domain rdf:resource="#rnd"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_ids_geoid">

    <rdfs:domain rdf:resource="#ptop_ids"/>

  </rdf:Property>

  <rdf:Property rdf:ID="elig_ELI_CRIT">

    <rdfs:domain rdf:resource="#elig"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_MT_MOD_YN">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_MARGIN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc1_PT_CLIN">

    <rdfs:domain rdf:resource="#canc1"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr3_TR_SP">

    <rdfs:domain rdf:resource="#tr3"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_SPM_TYPE">
```

```
    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="diag_DIAG_TXEN">

    <rdfs:domain rdf:resource="#diag"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr6_TRIAL_SP">

    <rdfs:domain rdf:resource="#tr6"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_AE_CONT">

    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_clinfactors_pcr">

    <rdfs:domain rdf:resource="#ptop_clinfactors"/>

    <rdfs:range rdf:resource="&xsd;byte"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_CARC_YN">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc_GRADE">

    <rdfs:domain rdf:resource="#CANC"/>

  </rdf:Property>

  <rdf:Property rdf:ID="REPEATSN">

    <rdfs:domain rdf:resource="#DCM_View"/>

  </rdf:Property>

  <rdf:Property rdf:ID="fu_FU_YN">

    <rdfs:domain rdf:resource="#fu"/>

  </rdf:Property>

  <rdf:Property rdf:ID="tr_TR_TXEN">

    <rdfs:domain rdf:resource="#tr"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_NONE">
```

```
    <rdfs:domain rdf:resource="#ae"/>

  </rdf:Property>

  <rdf:Property rdf:ID="mtd_GCSF_YN">

    <rdfs:domain rdf:resource="#mtd"/>

  </rdf:Property>

  <rdf:Property rdf:ID="canc2_PTIS_YN">

    <rdfs:domain rdf:resource="#canc2"/>

  </rdf:Property>

  <rdf:Property rdf:ID="diag_DIAG_YN">

    <rdfs:domain rdf:resource="#diag"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ptop_clinfactors_geoid">

    <rdfs:domain rdf:resource="#ptop_clinfactors"/>

  </rdf:Property>

  <rdf:Property rdf:ID="demo_INV_NAME">

    <rdfs:domain rdf:resource="#demo"/>

  </rdf:Property>

  <rdf:Property rdf:ID="chst_TRUCUT_DATE">

    <rdfs:domain rdf:resource="#chst"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ae_AE_END_DATE">

    <rdfs:domain rdf:resource="#ae"/>

    <rdfs:range rdf:resource="&xsd;date"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_ASSES_RSLT">

    <rdfs:domain rdf:resource="#ases"/>

    <rdfs:range rdf:resource="&xsd;int"/>

  </rdf:Property>

  <rdf:Property rdf:ID="ases_DIAG_INV_YN">

    <rdfs:domain rdf:resource="#ases"/>
```

```
</rdf:Property>

<rdf:Property rdf:ID="canc2_PT_SP">

  <rdfs:domain rdf:resource="#canc2"/>

</rdf:Property>

<rdf:Property rdf:ID="ae_AE_TXEN">

  <rdfs:domain rdf:resource="#ae"/>

</rdf:Property>

<rdf:Property rdf:ID="tr4_REGIMEN">

  <rdfs:domain rdf:resource="#tr4"/>

</rdf:Property>

<rdf:Property rdf:ID="canc2_LYNOD">

  <rdfs:domain rdf:resource="#canc2"/>

</rdf:Property>



<rdf:Description rdf:about="&code;PrimaryTumor">

  <rdf:type rdf:resource="#code_proc_type"/>

</rdf:Description>

<rdf:Description rdf:about="&code;ClinicalExamination">

  <rdf:type rdf:resource="#code_asses_type"/>

</rdf:Description>

<rdf:Description rdf:about="&code;LymphNode2">

  <rdf:type rdf:resource="#code_proc_type"/>

</rdf:Description>

<rdf:Description rdf:about="&code;LymphNode1">

  <rdf:type rdf:resource="#code_proc_type"/>

</rdf:Description>

<rdf:Description rdf:about="&code;PrimaryTumor2">

  <rdf:type rdf:resource="#code_proc_type"/>

</rdf:Description>

<rdf:Description rdf:about="&code;UltraSound">

  <rdf:type rdf:resource="#code_asses_type"/>
  </rdf:Description>

</rdf:RDF>
```