Electronic Health Record Data-as-a-Services Composition Based on Query Rewriting

Amarouche Idir Amine[1], Djamal Benslimane[2], Mahmoud Barhamgi[2], Michael Mrissa[2], and Alimazighi Zaia[1]

[1] Université des Sciences et de la Technologie Houari Boumediene
BP 32 El Alia 16111 Bab Ezzouar, Alger, Algeria
I.A.Amarouche@gmail.com, alimazighi@wissal.dz
[2] Université de Lyon, CNRS
Université Lyon 1, LIRIS UMR5205
43, bd du 11 novembre 1918, Villeurbanne, F-69622, France
{firstname.lastname}@liris.cnrs.fr

Abstract. Due to the large development of medical information systems over the last few years, there is today a strong need for an infrastructure that uniformly integrates the distributed and heterogeneous collections of patient data to deliver value-added information to healthcare professionals at the points of care. The adoption of Electronic Health Records (EHRs) and Web services as a software infrastructure has become an extremely important prerequisite for patient data integration. In this paper we propose a semantic-enabled architecture for the automatic composition of EHR (Electronic Health Record) DaaSs (Data-as-a-Service). In our architecture, DaaSs are selected and composed automatically to resolve the user queries (i.e. queries posed by physicians, nurses, etc) using a query rewriting approach. Our proposed approach can also handle the semantic conflicts of data exchanged among component services in an EHR DaaS composition by deriving and applying automatically the necessary data conversions.

Keywords: Electronic Health Record (EHR), Data as a Service (DaaS), query rewriting, semantic annotation, composition, mediation.

1 Introduction

Our current health environment is characterized by a shared and distributed localization of patient information. Patients' data are spread across several autonomous, proprietary and heterogeneous information systems. The adoption of electronically formatted patient data with Electronic Health Records (EHR) has become the primary concern for a broad range of health information technology applications and practitioners. According to Healthcare Information and Management Systems Society [22] the Electronic Health Record (EHR) of an individual consists of a collection of lifetime health data in electronic format, generated during relevant interactions with the healthcare system.

A. Hameurlain et al. (Eds.): TLDKS IV, LNCS 6990, pp. 95–123, 2011.

In this context, one of the big challenges of the health actors is the communication, the sharing (via exchange and integration) of EHR data through several independent and heterogeneous health systems. Thus, the EHR data need to be available, discoverable, searchable and comparable by a connected group of care providers and health organizations.

For this reason, many efforts have been undertaken to identify the requirements and information architectures needed to support shared Electronic Health Records. These research projects have focused on supporting the care given to patients by promoting good designs for EHR systems and standards for the secure communication of part or all of a patient's EHR between authorized systems [4].

Also, many e-health systems already provide the possibility to export their data in standardized formats such as CEN TC251, openEHR, and HL7-CDA [3] which provide different ways to structure and markup the EHR data for exchange purpose. For this, the adoption of standardized Electronic Health Records has become an extremely important prerequisite for bringing interoperability and effective data integration to the healthcare industry [22].

However, these EHR-related standards have centered on the communication of parts of the EHR of an individual subject of care and deal with the patient data at the document level. Furthermore, many of the users' query requirements target the contents of the clinical documents [26] and little work has been done to date on defining a generic means of querying EHR systems, as distributed repositories, in a consistent way [4].

Today there is increasing interest in moving towards a Service-Oriented Architecture for EHR data sharing among independent health information systems [28] [19] [23] [22]. Web service technology can be used as a standardized way for accessing and sharing the EHR data over healthcare information systems. This type of services is known as Data-as-a-Service (DaaS). DaaSs allow for a query-like access to organizations' data sources and do not change the state of the world [34] [37]. In this paper we use the term *EHR DaaS* to denote the DaaS that provides EHR data (or parts of).

While individual DaaSs may provide interesting medical information alone, in most cases, users' queries require the composition of multiple DaaSs. Furthermore, as there are several EHR DaaSs provided by several health actors, the user (e.g. physicians) needs an assistance to discover, select and compose the required EHR DaaSs. Therefore a solution is needed to select and compose EHR DaaS automatically for the purpose of retrieving and integrating the EHR data, which is the problem we tackle in this paper. This problem is very complex since EHR DaaSs are developed by independent organizations that may use different standards to describe their data.

To address these challenges, we propose a declarative query-rewriting based approach for the automatic composition of EHR DaaSs. The key idea behind the approach is to describe DaaSs as views over medical ontologies to capture their semantics in a declarative way. Defined views are then used to annotate the EHR DaaSs' description files (e.g. WSDLs) and exploited to compose DaaSs automatically. They are also used in resolving (on the fly) the semantic conflicts of data exchanged inside DaaS compositions. Our composition framework is based on an RDF query rewriting algorithm [6] inspired by the mature research work done in the data integration area [17].

The rest of paper is organized as follows: Section 2 provides a motivation example, highlights the challenges addressed in this paper and describes our contribution. Section 3 provides some background knowledge about EHR data and EHR DaaS. Section 4 outlines our service-oriented approach for EHR DaaSs composition. We also present in this section our results in terms of models, which include an ontology model, a model for services (EHR DaaS and mediation services) and a conjunctive query model. In Section 5, we introduce the generic algorithmic solution for query processing, which includes a query rewriting approach for EHR DaaS composition and the automatic invocation of mediation services for the resolution of semantic conflicts. Section 6 shows the system implementation and Section 7 is devoted to related works. Section 8 summarizes the results obtained in this work and discusses some possible extensions.

2 Motivation, Challenges and Contributions

In this section, we provide an illustrating example where the information needs of health actors are satisfied with a service oriented approach. This approach raises up many problems, which motivate our proposal to apply semantic Web technologies to support EHR DaaSs composition.

2.1 Motivation Example

Let us consider an e-health system exporting the set of EHR DaaSs in Table 1 to query the patient data. The description of EHR DaaS can be seen in Table 1, where the symbols "$" and "?" denote inputs and outputs of EHR DaaSs, respectively. We assume that a physician wants to consult the laboratory test results for his patient, "Joe, 35 years old man". Laboratory test results are helpful tools for evaluating the health status of an individual. In each laboratory test order, we find several tests (Cholesterol rate, Bilirubine rate, etc...). The physician submits the following query, as shown in Figure 1: Q_1: "What are the pathologies indicated by the results of the laboratory tests of Joe"

For the sake of simplicity, we assume that the e-health system does not provides functionality (service location record) to find any EHR DaaSs providing Joe's health data. Doing so, the physician has to invoke the EHR DaaS that provides the recent laboratory tests ordered for Joe namely S_{11} or S_{12}. He will not invoke S_{13} or S_{14} which return the ordered test made by Gynecologist and Paediatrician specialist respectively. Also he will not invoke S_{15} because this service returns the laboratory tests ordered but aborted for a patient. After invoking S_{11} and S_{12} he will obtain the list of recent and successful laboratory tests ordered for Joe and the laboratories' names in charge for performing the tests. Then he will invoke S_{21} and S_{22} to retrieve the results of the laboratory tests ordered for Joe and made by *laboratory* 1 and *laboratory* 2 respectively.

Table 1. Example of Electronic Health Record Data-as-a-Services

Service	**Functionality**	**Constraints and DaaS provider**	**The employed health standard**
$S_{11}(\$x, ?y)$	Returns laboratory tests y ordered for a given patient x	DaaS provider is hospital1	
$S_{12}(\$x, ?y)$		DaaS provider is hospital2	
$S_{13}(\$x, ?y)$		Patient gender (woman) , DaaS provider is maternity hospital	
$S_{14}(\$x, ?y)$		Patient age (< 14), DaaS provider is paediatric private hospital	
$S_{15}(\$x, ?y)$	Returns laboratory test y aborted for a given patient x	DaaS provider is hospital1	
$S_{21}(\$x, ?y, ?z)$	Return the name y and the value z of a given test belonging to lab test order x	z.unit (unit of measure) is mg/l, DaaS provider is laboratory 1	$y.code \in \{LOINC\}$
$S_{22}(\$x, ?y, ?z)$		z.unit (unit of measure) is mmol/l, DaaS provider is laboratory 2	$y.code \in \{SNOMED\}$
$S_{23}(\$x, ?y, ?z)$		DaaS provider is laboratory 3	
$S_3(\$x, ?y, ?z)$	Returns low reference value y and hight reference value z for given lab test x	z.unit and y.unit (unit of measure) is mg/l	$x.code \in \{SNOMED\}$
$S_4(\$x, ?z)$	Returns indicated disease z for abnormal value of lab-test x		$z.code \in \{ICD\}$ and $x.code \in \{SNOMED\}$

Furthermore, according to his own experience, the physician will not invoke S_{23} because of the inferior quality test results returned by the *laboratory* 3. After that, he will invoke S_3 which returns the reference or normalized values for each laboratory test parameter, in order to compare with the values returned by S_{21} and S_{22}. Then, if he found any suspicious values he invokes S_4 to retrieve the pathology indicated by each abnormal value. The list of pathologies returned by S_4 will indicate to him the pathologies of Joe may suffer from, and for which a treatment must be applied or another investigation is needed.

It is necessary to mention that during the comparison between the values of the test results returned by S_{21} or S_{22} and the references values returned by S_3, the physician must operate a conversion between value units (mmol/l and mg/l)[1] because each piece of data is interpreted differently. Also, the physician may need to convert exchanged data between selected services. For example, he has to change the laboratory test code returned by S_{21} (codified using the LOINC standard) to codes acceptable by S_3 (codified using the SNOMED standard)

In short, the physician needs to discover and select services, to invoke them in a certain order, to make sure that the parameters of the services are compatible, to consolidate the results returned by each EHR DaaSs and to manually perform an ordered set of operations like joins, selections and projections.

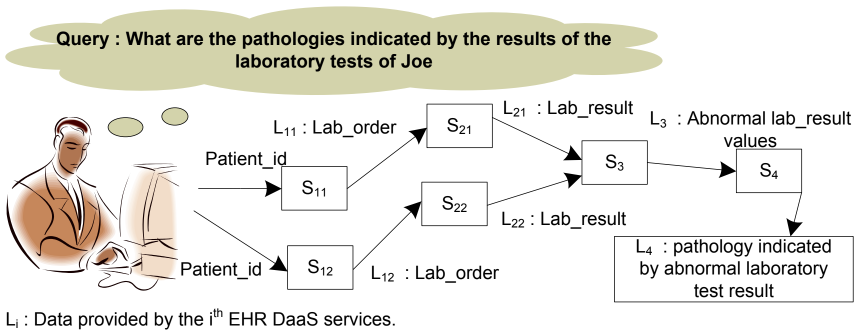

Fig. 1. Physician query scenario

2.2 Challenges

As shown previously, the manual process of composing EHR DaaSs for answering a query is painful and tedious; it may not possible for non expert users (eg. physicians, users...) to compose EHR DaaSs. Thus, automating the composition of EHR DaaS raises the following challenges:

1. *Understanding the Electronic Health Record Data-as-a-Service Semantics*: For the physician, confusion occurs in correctly understanding the functionalities provided by several EHR DaaSs. For instance, EHR DaaS like S_{11} and S_{15} have the same signature (input, output) but provide different functionalities, the former provides the laboratory test ordered for patient, whereas the latter provides the laboratory tests aborted for patient. Also, with several similar EHR DaaSs that offer the same functionality (like S_{11} or S_{12} or S_{13}) but that are subject to different constraints on data (patient age, patient gender,...), the physician must manually choose the ones that contribute to answering the query. Thus, the semantic annotation of EHR DaaS description files with the relationship between input and output on the one hand, and its content on another hand, will contribute mainly to automatic processing of EHR DaaSs discovery and composition and will spare the physician from selecting between several EHR DaaSs manually.

[1] In laboratory test results, mmol/l and mg/l are mass concentration unit measures.

2. *Electronic Health Record Data-as-a-Service discovery and composition*: The physician must select the services that are relevant to his query and compose them in the right order.
3. *Data level conflicts detection and resolution in Electronic Health Record Data-as-a-Services composition*: EHR DaaS parameters use concepts from different health ontologies such as SNOMED[2], ICD[3], LOINC[4], UMLS[5], ICPC[6] ...etc. For instance, S_{21} provides laboratory test coded in LOINC, but S_3 requires the laboratory test result specified using SNOMED terminology. Also, S_{22} returns test result measured in $mmol/l$ but S_3 measured its test result in mg/l. Consequently, the data will be passed during composition from S_{21} and S_{22} to S_3 will provoke an incompatibility problem. Then, how to leverage semantic conflict (health ontologies concept, unit,....etc) to enable the unrestricted composition of EHR DaaSs in a generic EHR environment becomes another challenge.
4. *Data trustworthiness*: Beside the semantic description of EHR DaaS, the physician need to have a mean to ensure the consistency and trustworthiness of data that are returned by different EHR DaaSs. Trustworthiness depends a lot on where data came from and which parties were involved in the establishment, change and forwarding of the data.

2.3 Contributions

Among the above mentioned challenges, we will focus only on the first three ones and we let the last one for future work. In a nutshell, we propose a new approach to integrate EHR data provided by several EHR DaaSs using a two-layer mediated ontology. The first layer, named *Generic Domain Ontology*, forms the core ontology (core concepts and relations) and it is the same for all EHR DaaSs providers. This layer is used to automate the composition of EHR DaaSs. The second layer, named *Specific Domain Ontology*, represents the (contextual) semantics (measuring units, scales, etc) of the data attached to EHR DaaS (called input/output parameters) and is used to detect and resolve the semantic conflicts of data exchanged among composed services.

The use of this two-level ontology allows deriving automatically the EHR DaaSs compositions that incorporate necessary mediation services (to carry out data conversion between interconnected services) during EHR DaaS composition. Our main contributions in the paper are summarized as follows:

- Firstly, we handle the first challenge by proposing an RDF-based modeling for EHR DaaSs. Specifically, we represent EHR DaaSs as *RDF views* over a mediated ontology. RDF views allow capturing the semantics of a DaaS in

[2] SNOMED: The Systematized Nomenclature of Medicine.
[3] ICD: International Codification of Disease.
[4] LOINC: Logical Observation Identifiers Names and Codes.
[5] UMLS: Unified Medical Language System.
[6] ICPC: The International Classification of Primary Care.

a declarative manner using concepts and relations whose semantics are formally defined in ontologies. We adopt SPARQL, the de facto query language for the Semantic Web, for posing queries over EHR DaaS services.

– Secondly, we propose a query rewriting approach to automatically select and compose the EHR DaaSs. In our approach composition queries, specified as SPARQL queries over a mediated ontology, are reformulated in terms of available EHR DaaSs based on the defined RDF views. Query reformulations are then translated into composition plans (i.e. orchestrations) that can be executed to answer the posed queries.
– Thirdly, since the data provided and required by individual EHR DaaSs may be bound to different (contextual) semantics (e.g. measuring units, scales, etc), we propose a mechanism that inserts automatically mediation services in compositions in order to resolve the semantic incompatibilities detected in the generated EHR DaaS compositions.

3 Background

In this section, we describe the features of EHR data published by EHR DaaSs. We also look at the relevant aspects of EHR data integration using Web service technology and the standards that EHR systems utilize nowadays. Furthermore, we look at the query rewriting approach for view based data integration, which is utilized in this work for EHR DaaS composition.

3.1 Electronic Health Record Standards

There have been various definitions of EHRs. According to the Healthcare Information and Management Systems Society[7] "EHR is a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting". A comprehensive survey and analysis of the electronic healthcare record is available in [14].

EHR data is stored in many kinds of systems and proprietary formats, inducing different internal structures. This situation leads to severe interoperability problems in the healthcare informatics domain. For this purpose, several EHR standards have been developed in order to structure the EHR data for the purpose of exchange. The standardization effort focused around two considered areas.

– The structuralization of EHR documents for the exchange of clinical data by supporting meaningful information representation between clinical information system within or between health care organizations. These standards include Health Level 7 (HL7) Clinical Document Architecture (CDA) [13], openEHR [35] and Cross-Enterprise Document Sharing (XDS) integration profile IHE[8], .. etc.

[7] HIMMS, http://www.himss.org

[8] IHE, "Integrating the Healthcare Enterprise" http://www.ihe.net.

– The health ontology used to represent the EHR data. To name a few examples of health ontology that model parts of the medical domain : ICD, LOINC, SNOMED, UMLS,..etc.

In sum, these approaches focus on the way of accessing the data rather than standardizing the data itself. A common feature of all emerging EHR standards is that the clinical concepts are modeled and expressed independently from how the data is actually stored in underlying data source. This challenge necessitated to:

– Select an appropriate technological infrastructure for making EHR data available at the point of care when authorized users need. In this context, Web services technology has been largely applied in healthcare domain by encapsulating existing EHR data within the Web service model and providing access to clinical data in a standard way.
– Adopt a novel modeling approach namely two-level modeling[14][7]. Two-level modeling approach in EHR system development divides the EHR data models into two separate ones. A generic information model and domain knowledge model. The domain knowledge model contains a set of constraints model ((simple and complex type), internal consistency (type, interval values, scale, unit, range), Reference Data (XML Format, health ontology)) on instance of the generic model entities.

3.2 Electronic Health Record Data Integration and Web Services

As explained previously, the introduction of Web service technology is motivated by the need to encapsulate the patient data in case of a EHR data exchange with another organization to perform a specialized medical procedure or for continuation of care. During this exchange, the execution of many operations are performed on EHR data before their exchange that concern (see figure 2).

– Retrieving relevant authorized patient data from the health organization information system, for instance : "Problem list", "past illness", "medication use", "present illness", "Family history", "Past surgical", "allergies",..etc.
– Coding this data using numerous standards that support ontological control at instance and type level by interlinking such health ontology (ICD-10, SNOMED,...) with the data definitions in standardized EHR documents (HL7,..etc) [24]. Using the coded data for creating a EHR document complaint XML clinical model (HL7/CDA, ...etc) ;
– Sending this document as SOAP encoded message to an appropriate destination.

However, the problem of EHR data integration is central in these systems. These latters deal with patient data at the document level, but health care data usage often is data centric, meaning that data should be extracted from various documents and then integrated according to specific criteria. As depicted by [24], even with a service approach, many interoperability problems still arise during EHR data integration.

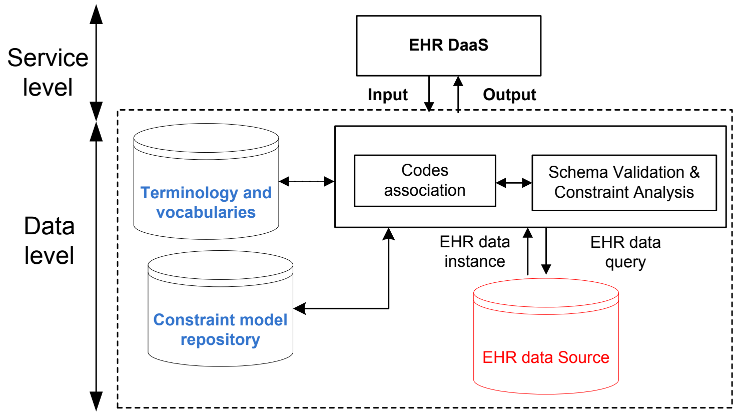

Fig. 2. Electronic Health Record data, publication through Electronic Health Record Data-as-a-Service

First, matching clinical data to codes in concurrent and semantically overlapping health ontology causes severe interoperability problems. Second, the semantic integration of heterogeneous systems in healthcare will have to deal with volatile medical concepts. For this reason, there is not, until now, a unique and comprehensive ontology of the medical domain [1] [30] [24].

In order to solve this problem, EHR data integration systems have to consider two levels; (1) generic information model and (2) domain knowledge (vocabularies, terminologies,..). These two levels must inter operate to integrate EHR data from disparate healthcare systems [33].

As every standard has its information reference model upon which domain knowledge is built, we will focus only on features of domain knowledge where EHR complaint XML documents at entry or section level are characterized by the frequent association with:

- Ontological concepts defined in some health ontologies (ICD, LOINC, SNOMED) for coding EHR;
- Semantic and structural constraint for maintaining internal consistency of EHR data;

Thus, the additional knowledge provided by the reference models upon which domain knowledge is built will not be addressed in this work.

3.3 Query Rewriting

The query rewriting problem has been extensively studied in the areas of query optimization and data integration. In the hereafter we report some definitions (based on the works [17,18]) to terms we use extensively throughout the paper.

Query Rewriting: Given a query Q and a set of view definitions $V = \{V1, ..., Vm\}$, the query Q' is an rewriting of Q using V if: Q' is contained in Q and Q' refers only to the views in V.

Query Containment and Equivalence: Query containment and equivalence enable comparison between different rewriting of a query. It will be used when we test the correctness of a rewriting of a query in terms of a set of views. Thus, a query $Q1$ is said to be contained in a query $Q2$, denoted by $Q1 \subseteq Q2$, if for all databases D, the set of tuple computed for $Q1$ is a subset of those computed for $Q2$, i.e., $Q1(D)$ and $Q2(D)$. The two queries are said to be equivalent if $Q1 \subseteq Q2$ and $Q2 \subseteq Q1$ or $Q1 \equiv Q2$.

Maximally Contained Rewriting: Equivalent rewritings may not always exist under the open world assumption. Finding the maximally-contained rewriting will be the only alternative for resolving a query. Thus, Let Q be a query, $V = \{V1, ..., Vm\}$ be a set of view definitions, and L be a query language. The query Q' is a maximally-contained rewriting of Q using V with respect to L if: Q' is a query in L that refers only to the views in V, Q' is contained in Q, and there is no rewriting $Q1 \in L$, such that $Q' \subseteq Q1 \subseteq Q$ and $Q1$ is not equivalent to Q.

4 Overview of the Approach

In this section we define our reference architecture for EHR DaaSs composition that is independent from specific data standards.

4.1 General Architecture

Our reference architecture defines four logical tiers, as shown in Figure 3.

- **Data Level:** The lowest level of the architecture contains information stored in different components. These components can be databases that store all the medical information concerning patients or documents that preserve all official documents generated during healthcare process. Also there are several clinical terminological and documentary resources that provide means to search and share clinical knowledge.

- **Service Level:** The service level publishes the different services provided by several systems to e-health actors. Services are either simple (one provider) or complex (multiple providers). This level provides two services categories.
  1. **Electronic Health Record Data-as-a-Services** provide information about patients. We can find two kinds of EHR DaaS in this category according to the nature of the data provided: EHR DaaSs that provide specific patient information (diseases, symptoms, medications or family history and so on) or EHR DaaSs that retrieve a clinical document complaint model (discharge summaries,...).

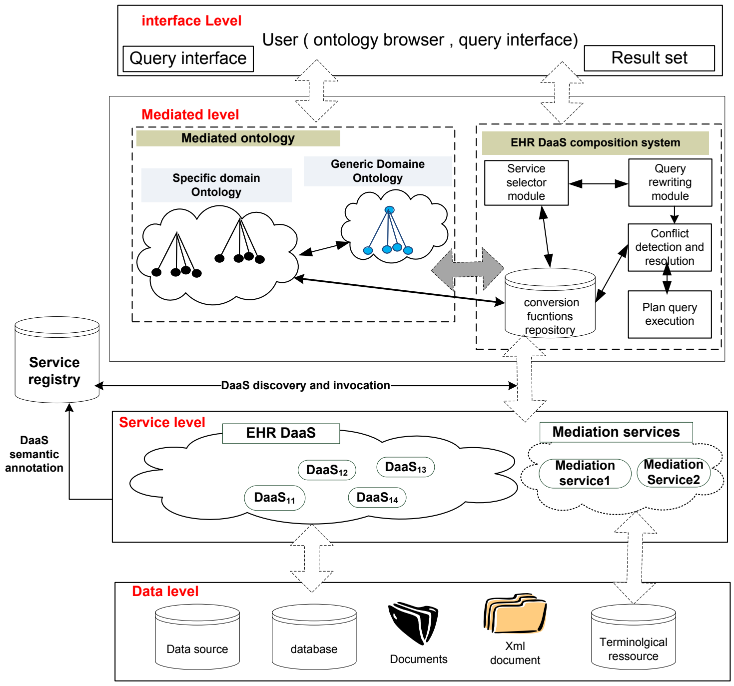

Fig. 3. Overview of the Electronic Health Record Data-as-a-Service composition architecture

2. **Mediation Services** is used mainly for mapping and converting the output parameter of a specific EHR DaaS to the input parameter of a subsequent EHR DaaS during service composition. Mediation services provide the definition of cross-mappings between terminologies (e.g; UMLS Terminology Services) and extract from a EHR DaaS output parameters what the inputs of subsequent EHR DaaS need.

These services advertise their WSDL definitions into a service registry. WSDL provides an XML-based grammar for describing a service interface. For automatic discovery, selection and composition of service, WSDL files are annotated with semantic entities from a mediated ontology. The service registry includes a set of services descriptions (WSDL files) semantically annotated with RDF views expressed in term of mediated ontology as in [6].

– **Mediated Level:** The mediated level is composed of two modules:
  1. *Mediated ontology*: The mediated ontology contains all the concepts and relations defined in EHR domain. It will be used to annotate and query services (EHR DaaSs, mediation services) in an environment of heterogeneous inter-working health information systems. We divide the ontology into two ontological levels which cuts the concept space into a

generic domain ontology and a set of extensions named domain specific ontologies.

1-1. The Generic Domain Ontology (**GDO**) defines the generic concepts and relations covered the EHR domain and it is the same for all EHR DaaSs providers. For example, patients, disease, laboratory test are entities of the generic domain ontology. The generic ontology is different from a specific ontology in that it only contains basic shared concepts and their properties. For example, the generic ontology states that a `laboratory-test-result` has a unit and a code, without specifying any specific unit or health ontology code (this information will be specified using the SDO detailed later). Doing so, the GDO is used mainly for EHR DaaS discovering and composition.

1-2. The Specific Domain Ontology (**SDO**) is created mainly to allow the semantic extension of generic ontology concepts for detecting and resolving semantic data conflicts. For example, the SDO states that a `Laboratory-test-code` or `Disease-code` are expressed in SNOMED or LOINC; or `laboratory-test-value` has `mg/l` as a unit of measure.

2. *Electronic Health Record Data-as-a-Service composition system:* Contains four sub-modules: a service selector module, a query rewriting module, a conflict detection and resolution module and a query plan execution module. The first module receives the query from the user interface and analyzes it based on the mediated ontology for discovering the appropriate EHR DaaSs. The second module receives the set of semantic descriptions of discovered EHR DaaSs and applies a query rewriting algorithm that generates a set of valid and executable of EHR DaaS compositions. The third one iteratively processes each rewriting previously generated in order to detect incompatibilities (semantic conflicts at the data level) and invokes the appropriate mediation services for their resolution. Mediation services offer the conversion functions defined in the conversion repository and referenced by SDO concepts. The last module arranges the selected EHR DaaSs along with the added mediation services in a composition plan which will be executed to return the results to users.

– **Interface Level:** The aim of this layer is to provide the interface for user whereby he can perform a query and receive results sets.

4.2 Models for Electronic Health Record Data-as-a-Services Composition

In this section, we propose models to address the issues related to query processing (query rewriting and conflict resolution) for EHR DaaSs composition. First, we formalize the notion of mediated ontology, with the introduction of generic and specific domain ontologies, which are useful as a support for semantic-aware querying and annotation of EHR DaaSs. Then, we propose a model for representing conjunctive queries over a mediated ontology. Finally, we develop models for EHR DaaS and mediation services.

Mediated Ontology

Mediated ontology includes two ontologies, namely, the generic and specific domain ontologies which have 'GDO' and 'SDO' as namespaces for their respective concepts. Such ontology should be defined by domain experts and specified using RDF/RDFS. The generic and specific ontologies models are inspired from [6] [31] [25]. In order to provide a precise semantic annotation for our service model we use these two models.

Definition 1. *(Generic Domain Ontology) : A RDFS generic ontology is 6-tuple $<$ C,D,OP, DP, SC, SP$>$ where*

- *C is a set of classes.*
- *D is a set of data types.*
- *OP is a set of object properties. Each object property has its own domain and range in C.*
- *DP is a set of data type properties. Each data type property has a domain in C and range in D.*
- *SC is a relation over $C \times C$, representing the sub-class relationship between classes. For example C_2 SC C_1 expresses that C_2 is subclass of C_1.*
- *SP is a relation over $(OP \times OP) \cup (DP \times DP)$, representing the sub-property relationship between homogeneous properties. For example DP_2 SP DP_1 means that DP_2 is a sub-property of DP_1.*

Figure 4 depicts the Generic Domain Ontology, in which class nodes are represented by ovals and data type nodes are represented by rectangles. In the GDO ontology, the `GDO:Patient` class is a core concept that characterizes patient information, such as name, SSN, etc. The `GDO:Laboratory-test-order` class captures information on laboratory tests ordered for a patient. It is related to the `GDO:Patient` class through the object property `GDO:Has-order`. The `GDO:Laboratory-test-Result` class captures results of laboratory tests ordered for patient and is related to the `GDO:Laboratory-test-order` class via the `GDO:Has-Result` object property. Individuals of `GDO:Laboratory-test-result` such as `LDL`, `AST`, `ALT`, `TotalBilirubin`, etc. are subclasses of the `GDO:Panel` class. The individuals that belong to the `GDO:Laboratory-test-Result` class may be related to multiple panels and each panel has several laboratory tests. The `GDO:Disease` class characterizes the disease which can be indicated by patient `GDO:Laboratory-test-result` and is related to the laboratory test class via the `GDO:indicate` object property.

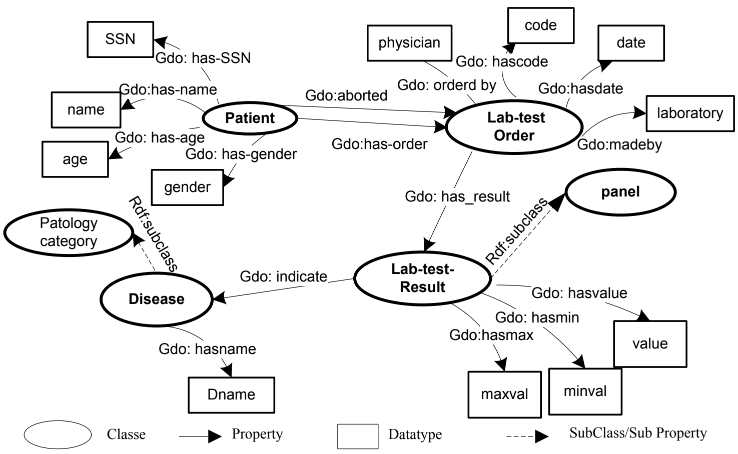

Fig. 4. Generic Domain ontology

Definition 2. *(Specific Domain Ontology) : A RDFS specific ontology is 3 tuple* $< C_g, C_i, \tau >$*, where:*

- C_g *is a set of concepts that represent the different conflictual aspects of a generic concept in Generic Domain Ontology (GDO). Each* C_g *has a name and a set of specialized concepts (i.e. sub concepts); the name represents a conflictual aspect of the associated generic concept. In the example depicted in Figure 5,* `SDO:Laboratory-test-code` *and* `SDO:disease-code` *are* C_g *concepts.*
- C_i *is a distinct set of concepts having the same super-concept* C_g*. By definition,* C_i *are not allowed to have sub-concepts. The properties of* C_i *are defined as follows :*
  - *name of concept.*
  - *Seqno is the property that represents the sequence number of a* C_i *concept among its siblings.*
  - *A couple of properties reference the conversion functions between orderly organized object of* C_i*. The function name denotes the conversion from* C_i *to subsequent or precedent sibling, for instance* `snomed-to-loinc`*,* `loinc-to-snomed` *or* `mg/l-to-mmol/l`*, as it follows the mapping direction. Supported conversions between sibling subclasses are* $n \longrightarrow 1$ *and* $1 \longrightarrow 1$.
- τ *refers to the sibling relationships on* C_i *and* C_g*. The relationships among elements of* C_g *is disjoint. However elements of* C_i *of a given* C_g *have peer relationship. They have similar data semantics, so that conversion or mapping can be performed among them.*

Let us illustrate this definition with an example in Figure 5. The concept `GDO:Laboratory-test-Result` in that figure has a conflictual aspect called "unit" that is described as a member of C_g in SDO (i.e. $SDO : unit$). The defined concept `SDO:unit` is linked to `GDO:Laboratory-test-Result` via the object property `SDO:has-Unit` which is also defined in SDO. `SDO:unit` has different measurement units represented as sub classes $C_i = \{mg/l, mmol/l, ..., n\}$. The code is also a conflictual aspect to both `GDO:Laboratory-test-result` and `GDO:Disease` concepts; i.e. codes can be represented differently in different health ontologies using $C_i = \{SNOMED, ICD, ..., n\}$. Note that, we can use an rdfs:collection to denote the sequence relationships between elements of C_i and typical processing will be to select one of the members of the container.

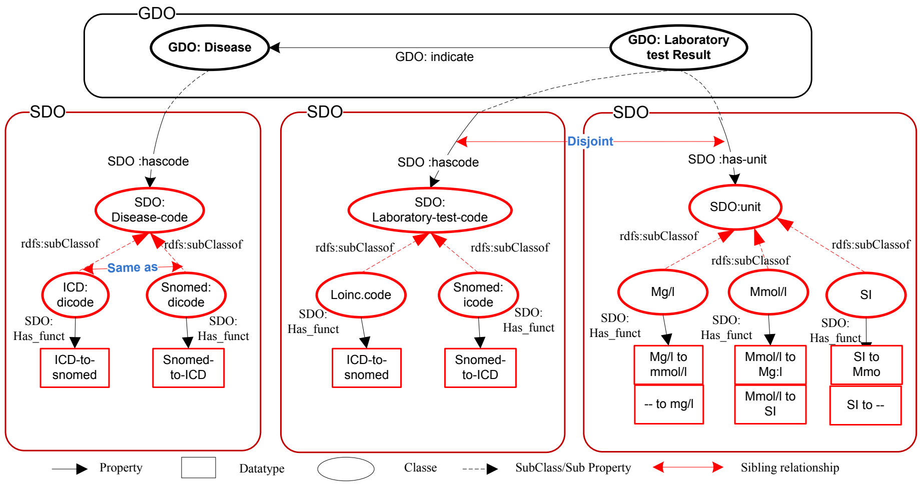

Fig. 5. Specific Domain ontology

Conjunctive Queries: In this paper we address conjunctive queries expressed using SPARQL, the do facto query language for the Semantic Web[9].

Definition 3. *A conjunctive queries Q has the form: Q(X):-$< G(X,Y), C_q >$ where :*

- $Q(X)$ *is the head of* Q*, it has the form of relational predicate and represents the result of query.*
- *G(X,Y) is the body of* Q*, it contains a set of RDF triples where each triple is of the form (subject. property.object). X and Y are called the distinguished and existential variables respectively. X and Y are subjects and objects in the RDF triples.*
- $C_q = \{C1_q, C2_q, ..., Cn_q\}$ *is a set of constraints expressed on X and Y variables in terms of traditional intervals or arithmetic expression like* $x\theta constant$ *,* $y\theta constant$ *and where* $\theta \in \{<, > \leq, \geq\}$ *.*

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

In our work, queries are formulated in SPARQL and use concepts from the mediated ontology (GDO ontology) and properties from the specific ontologies (SDO ontologies). Thus, a query can be seen as a graph with two types of nodes; class and literal nodes. Class nodes refer to classes in the ontology. They are linked via object properties. Literal nodes represent data types and are linked with class nodes via data type properties. Figure 6 depicts the RDF graph of the query Q_1 described in our scenario. The graph shows that $Q1$ has four class nodes P, LO, LR, D linked by object property `GDO:has-order(P,LO)`, `GDO:has-result(LO,LR)` and `GDO:indicate(LR,D)` respectively.

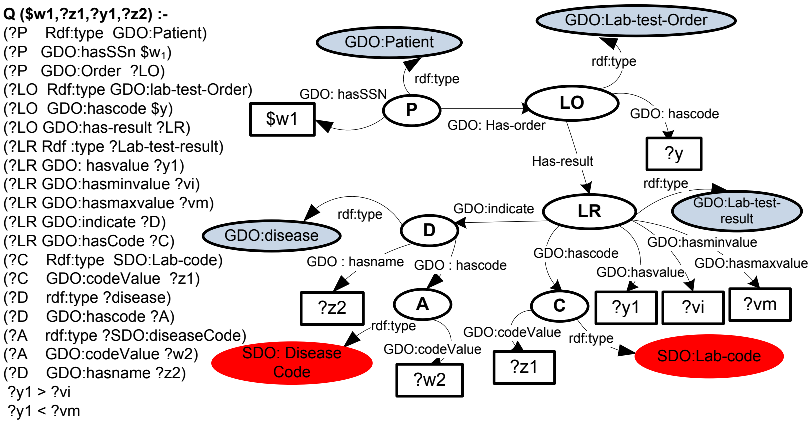

Fig. 6. Query in the running example

Electronic Health Record Data-as-a-Service model

We deem appropriate to follow the work of [6,34,10] to formalize the modeling of EHR DaaS as RDF views over a mediated ontology.

Definition 4. *EHR DaaS S_j is described as view in a Datalog-like notation over a GDO and SDO thus S_j model is :*

$$S_j(\$X_j, ?Y_j) : - < G_j(X_j, Y_j, Z_j), Co_j > |\alpha_{X_j}, \alpha_{Y_j} \text{ where:}$$

- *X_j and Y_j are the sets of input and output variables of S_j, respectively,*
- *G_j represents the functionality of the EHR DaaS which is described as a semantic relationship between input and output variables.*
- *Z_j is the set of existential variables relating X_j and Y_j.*
- *$Co_j = \{Co_{j_1}, ..., Co_{j_n}\}$ is a set of constraints expressed on X_j, Y_j or Z_j variables like $x\theta constant$ and $y\theta constant$ where $\theta \in \{<, >\leq, \geq\}$.*
- *α_{X_j} and α_{Y_j}, named adornment, are a set of RDF triplets describing the semantic (ontological reference, unit...etc) or domain expression of X_j and Y_j respectively. Each adornment α is indicated by the 2-tuple; $< C_g, C_i >$ where : C_g : is an SDO concept that represent the different conflictual aspects X_j and Y_j); C_i : is a concept from SDO inherited from C_g .*

An EHR DaaS model is described over a GDO and adorned by the entities from SDO. As an EHR DaaS is modeled uniquely over the entities of GDO, it does not provides explicit semantics about its input-and output parameters, so we extend its description with additional information describing more precisely how the semantics of the GDO concepts are described according to the SDO ontology. Then, each EHR DaaS model will be expressed as an adorned query [10]. The adornment is an annotation on variables, appearing in input and output parameters of a given EHR DaaS and expressed in term of SDO.

Figure 7 gives RDF view of EHR DaaS S_{21} and S_{22} services depicted in Table 1 with an adornment depicted in red color.

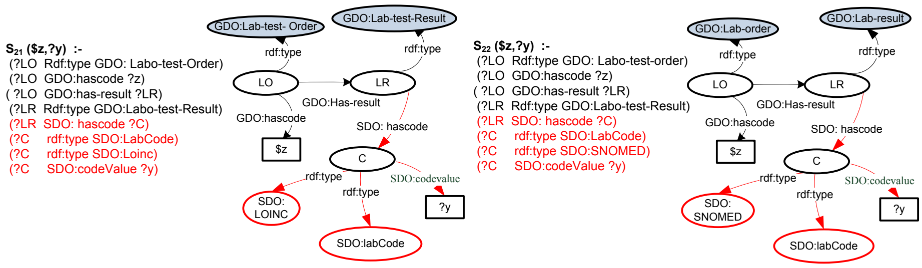

Fig. 7. Electronic Health Record Data-as-a-Service model

Mediation Service Model

Mediation Services are also represented as an EHR DaaS model (expressed in term of SDO ontology) whereas their adornments are described as a set of RDF triples that define the conversion function between peers of $SDO : C_i$ sub-concepts from the same $SDO : C_g$ concept in a declarative way. We remind the reader that the different $SDO : C_i$ are organized as an ordered list, hence a conversion from one to another is always a concatenation of conversion functions.

Definition 5. *Mediation Service S_j is modeled as below :*
$S_j(\$I_j, ?O_j) : - < G_j(I_j, O_j, Z_j) > |\alpha_{Func<I_j,O_j>}$*; Where :*

- $\$I_j$ *defines the input parameter required for using mediation service;*
- $?O_j$ *defines the output parameter required for using mediation service;*
- Z_j *represents variables or constants generated inside a conversion or required during conversion.*
- $\alpha_{Func<I_j,O_j>}$ *represents the conversion function from $SDO : I_j$ to $SDO : O_j$.*

Figure 8 illustrates the RDF view of a mediation DaaS service $S_{LOINC-SNOMED}$ utilized for converting a labcode from S_{21} to S_3.

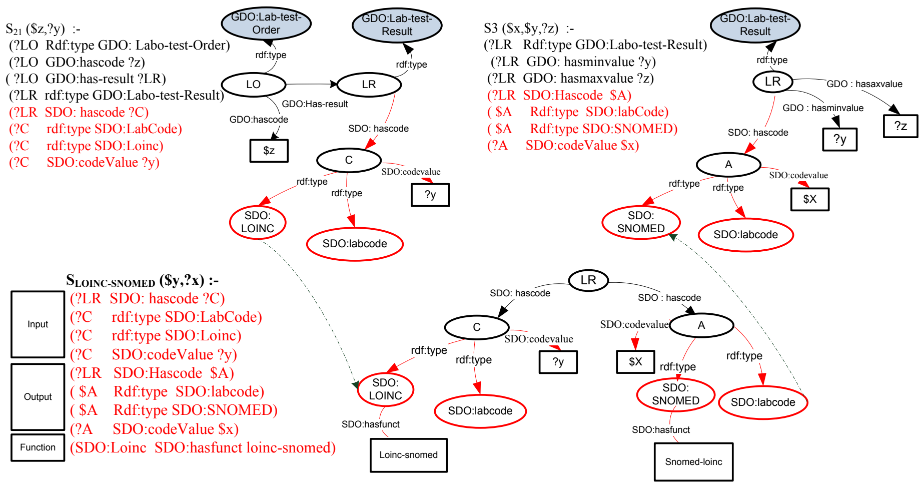

Fig. 8. Mediation Service model

5 Query Processing for Electronic Health Record Data-as-a-Services Composition

In this section, we outline the development of query processing for EHR DaaSs composition and we give a detailed description of the key phases of query rewriting and conflict detection and resolution phases.

5.1 Query Processing Phases

The complete query processing steps are depicted in Figure 9. They include four processes. First, query formulation and service discovery, second, query rewriting, third, conflict detection and resolution, and finally query execution and result restitution.

1. Firstly, query formulation and service discovery: In (1) and (2) the user issues SPARQL queries in terms of mediated ontology. Doing so, in (3) and (4) the service selector discovers EHR DaaSs from the service registry that partially or completely matches the query entities (class nodes, object property nodes).
2. Secondly, query rewriting: In (5), the query sent over the mediated ontology is rewriten into a query that refers directly to the set of discovered EHR DaaSs. The query rewriting module of EHR DaaSs composition component uses an approach in the spirit of the "bucket algorithm" [18] that returns the maximally contained rewritings of a query. The algorithm computes for each query entity (class node, object property node) called bucket or subgoal in the bucket algorithm, the EHR DaaS that are relevant to it. Thus,

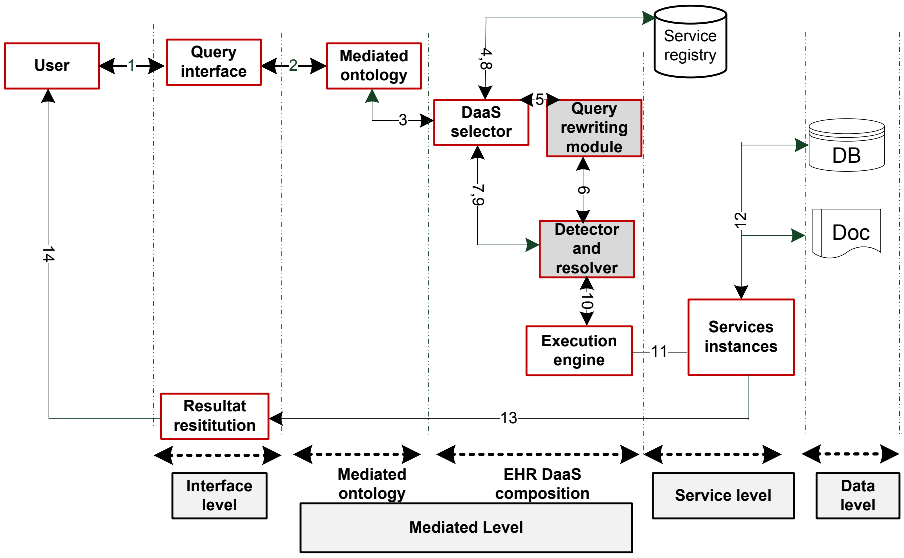

Fig. 9. Electronic Health Record Data-as-a-Services composition process

the rewriting is constructed by combining one element of every bucket. Candidate solutions generated by the query rewriting algorithm must be tested by applying the conjunctive query containment to validate it.

3. Thirdly, conflict detection and resolution: In (6), considering each combination generated by the query rewriting module, which may encompasses semantic conflicts at the data level (7), requires testing any conflict by comparing output and input of subsequent EHR DaaS in each query rewritings. The conflict is resolved with the insertion of a call to mediation services (8,9). Thus, besides EHR DaaSs, in each query rewriting combination mediation services are added to resolve conflicts.
4. Fourthly, query execution and result restitution: In (10, 11, 12), orchestrating the conflict-free composite service that has been generated requires a translation into an execution plan describing the data and control flows. Finally, (in 13 and 14) the result restitution module synthesizes results and returns them to users through user interface.

5.2 Query Rewriting

Given a query Q and a set of EHR DaaSs, the query rewriting module rewrites Q as composition of EHR DaaSs whose union of RDF graphs covers the RDF graph of Q. The query rewriting phase is preceded by a preprocessing step (carried out prior to receiving the composition query) in which the RDF views are extended with the RDFS semantic constraints (i.e. *subClassOf*, *subPropertyOf*, *Domain* and *Range*) to obtain a better matching with the composition queries.

Our composition query rewriting algorithm [6] has two main phases detailed as follows:

(1) **Finding the Covered Query's Sub-graphs:** In this phase the query is compared to the different RDF views to determine the class nodes and the object properties that are covered by the views. The term covers means that there is a containment mapping between classes nodes and object properties of Q and those of the views. The result of this step is a table summarizing for each EHR DaaS, the covered classes nodes and object properties. For example, the table2 shows the covered classes and properties for the services S_{11}, S_{21}, S_3, S_4.

Table 2. The covered classes nodes and object properties for the services S_{11}, S_{21}, S_3, S_4

Service	**Covered classes and properties**
$S_{11}(\$w_1, ?y)$	Patient(w_1), has-order(Patient, LabOrder), LabOrder(y)
$S_{21}(\$y, ?y_1, ?z_1)$	LabOrder(y), hasResult(LabOrder, LabResult), LabResult(y_1), hasCode(LabResult,LabCode), LabCode(z_1)
$S_3(\$z_1, ?v_i, ?v_m)$	LabResult(v_i, v_m), hasCode(LabResult, LabCode), LabCode(z_1)
$S_4(\$z_1, ?z_2, ?w_2)$	LabCode(z_1), hasCode(LabResult, LabCode), indicate (LabResult, Disease), Disease(z_2), hasCode(Disease, DiseaseCode), DiseaseCode(w_2)

(2) **Composition Generation:** In the second phase, the query rewriting module combines the different lines in the generated table in such a way all the classes nodes and object properties of the query are covered by the combination and the combination is executable. A combination is executable if the inputs of involved services are bound or can be made bound by other services (whose inputs are bound). For example, the combination of the services S_{11}, S_{21}, S_3, S_4 covers the whole set of classes nodes and properties; all the inputs of theses services are bound or can be made bound. Therefore the set S_{11}, S_{21}, S_3, S_4 is considered as a valid composition.(see Figure 10)

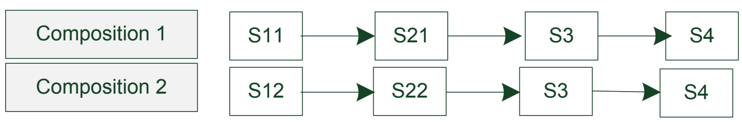

Fig. 10. Composition generation

5.3 Conflicts Detection and Resolution

In this phase we detect and resolve the semantics conflicts in the EHR DaaS compositions generated in the previous phase. This phase includes the following steps:

(1) The localization of conflicts between interconnected EHR DaaSs. Conflicts arise when data elements that have to be exchanged between two interconnected EHR DaaSs are interpreted differently by these services. Conflicts localization is accomplished by the *conflict detection module*. This kind of conflict is named *Attribute level incompatibilities* by the classification of structural and semantic message level heterogeneities proposed in [29]. According to that work, attribute level incompatibilities arise when semantically similar attributes are modeled using different descriptions. To detect the conflicts we define some rules that define some conflicts between EHR DaaSs expressed in RDF as indicated bellow:

- let $SDO : R_i$ and $SDO : E_i$ be subclasse of the same conflictual class $SDO : C_g$, such as $SDO : lab - code$, thus :

```
SDO:Ei  rdfs:subClassof  SDO:Cg
SDO:Ri  rdfs:subClassof  SDO:Cg
```

- Then, if we have two EHR DaaS S_i and S_j including concepts $SDO : E_i$ and $SDO : R_i$ respectively in their RDF descriptions as an adornment, expressed as depicted by the following triples :

```
LR  SDO:hascode ?N
?N  rdf:type   SDO:Ei
?N  rdf:type   SDO:LabCode
LR  SDO:hascode  ?A
?A  rdf:type   SDO:Ri
?A  rdf:type   SDO:labCode
```

then we have code laboratory test conflict.

In sum, the set of conflict types identified in our solution is the set of conflictual concepts $SDO : C_g$. For instance, code disease conflict or unit conflict. Other conflict types (e.g. data representation, data precision ,... etc) can be added to SDO in order to resolve more semantic conflicts at the data level. As a consequence, this step will provide the set of conflict objects where each conflict object will be identified as 3 tuple $< O(S_i), I(S_j), C_g) >$ where $O(S_i)$ is an adorned output parameter of a given EHR DaaS source S_i , $I(S_j)$ is an adorned input parameter of a given EHR DaaS target S_j, conflict type is a member of set of conflictual concepts $C_g = \{unit, labCode, ..., n\}$.

To detect the conflicts, the algorithm depicted (algorithm 1) will take each composition (represented as Directed Acyclic Graph) and iteratively verify the rules expressed previously for each parameter (adornment only) exchanged between interconnected services to find out all possible conflicts which will be stored in the conflicts-objects set.

(2) The conflict objects detected previously will be resolved by the automatic invocation of an appropriate mediation service. The latter is identified through:

- the input parameter $SDO : O(S_i)$, which is the output of S_i;
- the output parameter $SDO : I(S_j)$ which is the input of S_j;
- the conversion function as an adornment, defined as property of $SDO : O(S_i)$ and targets $SDO : I(S_j)$.

Algorithm 1. semantic conflict detection and resolution

Require: $M_{i,j}$ {Matrix is a graph of EHR DaaS combination with conflict}; $i, j, k, z \in \mathbb{N}$, $\mathcal{CO}$ a set of Conflict Object,
{Detection conflict step}
for $i = 1$ to n **do**
  for $j + 1$ to n **do**
    if $M[i][j] = 1$ **then**
      if $Output.S_i$ AND $Input.S_j$ have the same conflictual concept as type and differents SDO subclasses **then**
        $\mathcal{CO}z$ = New conflict object(output.$\mathcal{S}_i$, input.$\mathcal{S}_j$, conflictual concept),
        Add ($\mathcal{CO}$, $\mathcal{CO}z$)
      end if
    end if
  end for
end for
{Resolution conflict step}
for each $\mathcal{CO}z$ in $\mathcal{CO}$ **do**
  {according to conflict object identify mapping function ($Output.S_i$,$Input.S_j$) from SDO ontology}
  $M[i, j] = 0$ {delete S_i and S_j arc}
  ADD S_K {ADD mediation DaaS service}
end for
Ensure: $M'_{i,j}$ graph of EHR DaaS combination without conflict

As a consequence of this phase, the mediation services $S_{LOINC-SNOMED}$ and $S_{mmol/l-mg/l}$ are added to the first and second EHR DaaS compositions to resolve conflict as depicted in figure (see Figure 11). Afterwards, the obtained conflict-free compositions will be translated into execution plans (i.e. orchestrations) describing the data and control flows as depicted in the same Figure 11. For space limitation, we do not detail this step in the paper.

6 Implementation and Evaluation

To illustrate the viability of our approach to EHR DaaS composition, we implemented about /411/ EHR DaaS Web services on top of a set of medical data sources containing synthetic data about patients, including information like diseases, medical tests, allergies, medications lists, vaccination records, ongoing treatments, consultations, personal information (e.g., date of birth, sex, etc), etc. All of these data are usually represented by the commonly used types of the EHR information model. We built a medical ontology based on the building blocks and the data-types defined in the HL7 and the openEHR standards. The ontology included /81/ ontological concepts and /413/ properties (i.e., both datatype and object properties). We modeled all services as RDF views over that ontology. These views were used to annotate the description files (WSDLs) of corresponding DaaS services. We implemented also a set of mediation services; these services were used to convert the values of exchanged data from HL7 to

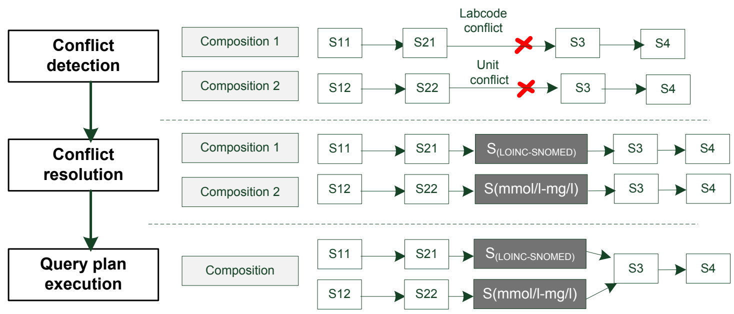

Fig. 11. Conflict detection and resolution

openEHR specific datatypes (and vice versa). These services allowed for example to convert between different medical data's measurements units, precisions, etc. All services were implemented in Java, and hosted on a GlassFish Web server.

Figure 12 depicts our implemented evaluation system. In that figure, the *DaaS Web services layer* plays the role of an abstraction layer on top of heterogeneous medical data sources; i.e., medical data located in heterogeneous data sources (e.g., relational data bases, silos of data-centric homegrown or packaged applications (e.g. SAP, PeopleSoft, Siebel, etc), files XML files, etc.) are all accessed by the same interface, the Web service interface. These services can be composed by the *Web Service Management System*. The system's users are assisted in formulating their queries (i.e., SPARQL queries) over the ontology.

We tested our system with a set of real-life queries (including that of the running example), examples included: "Q_1: check whether the medication ABC identified by the code "801" to be prescribed to patient John Doe interacts with the ones currently taken by that patient", "Q_2: For any given social security number X of a patient and a medication code Y representing the medication to be prescribed, verify whether the medications taken by the patient may interact with Y", "Q_3: What are the tests performed by patients that have been administered a given medication?", etc.

Throughout our tests, we made the following observations: (*i*) the system was able to process hundreds of services in a reasonable time (411 services in less than one second), the reported time is the time to create the composite services (this involves parsing the WSDL files of services, determining the relevant ones and building the composition); the created compositions included both the EHR DaaS services and the necessary mediation services. (*ii*) in all of the considered queries (20 queries), the system was able to insert the necessary data mediation services to transform data between heterogeneous component services in a composition. The system users (mainly physicians in our tested examples) have expressed their satisfaction, as they were able to answer their queries on the fly without any programming involved. Users used the created compositions in their daily clinical scenarios (e.g., prescribing a medication, studding the risks of a medication, etc.).

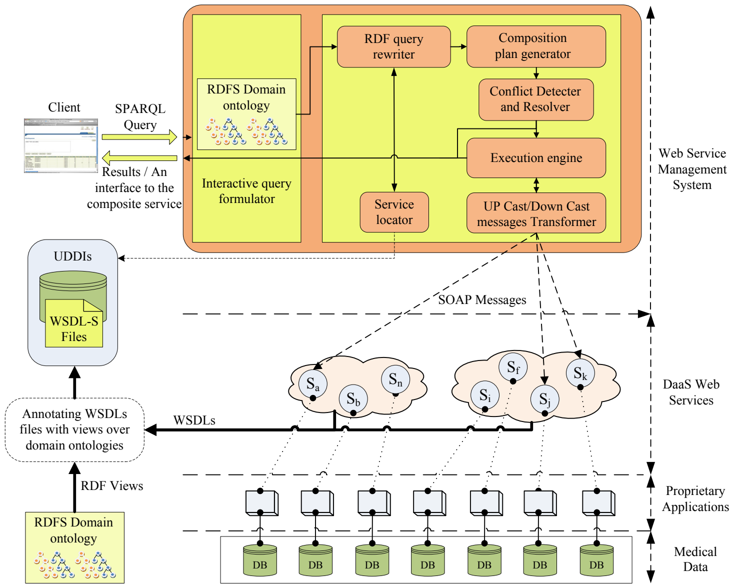

Fig. 12. The implemented system for evaluating our approach

7 Related Works

In this section, we give an overview of the main research works related to our subject. We have categorized these works into the following areas: EHR data integration, EHR Web service composition and query rewriting approach for automatic Web service composition.

7.1 Electronic Health Record Data Integration

Data integration is the problem of combining data residing at different sources to provide the user with a unified view of data. There is a large body of research work on data integration not only in the area of computer science but also in health and biomedical informatics[1] [11]. Broadly speaking data integration approaches can be classified into two main approaches: "data warehouses" and "mediation" approaches.

The mediation approach, unlike data warehouse, does not materialize data; it integrates data at the query processing time. In the e-health area most data integration projects have adopted this approach like, to name a few, Synapses [16], Synex [16] and Pangea-LE [1], etc. Most of these projects provide a global XML schema over structured XML views of EHR documents stored for a patient in existing health data repositories. The mediated system accepts requests for data from clients, decomposes them into queries against the connected data sources, and integrates the responses dynamically. In contrast with these projects, we adopt a service-oriented data mediation architecture and a service composition approach to integrate data on the fly by composing autonomous EHR DaaSs. Also, unlikely to these projects we handle the semantic conflicts at the data value level when data is exchanged among healthcare systems.

7.2 Medical Web Services Composition

A large number of research work have addressed the problem of WS composition in the healthcare application domain[2], [28], [39] and [21]. However, the bulk of these works have focused only on workflow oriented compositions; i.e. compositions that implement the different steps involved in a given business task (e.g. patient admission in a hospital, patient discharge, etc). We review some of these works in the following:

Authors in [2] defined a model-driven approach for semi-automatic Web service orchestration with run-time binding in the healthcare domain. Information related to medical Web services can be found in the corresponding standardization documents for instance HL7, DICOM and IHE. Unfortunately, WS composition in that work does not allow to integrate heterogeneous medical data sources.

Budgen et al. [9] propose data Integration Broker for Heterogeneous Information Sources (IBHIS). The proposed broker adopts a service-based model to query data at autonomous healthcare agencies. The broker achieves its goals through the use of semantic data descriptions, a semantic registry and a query engine. The semantic registry is based upon an extended form of UDDI, incorporating a matchmaker to match OWL-S data descriptions from the registry to the users' queries. The query engine formulates the users' queries, interacts with the matchmaker to answer the query, and displays the final results to users. Unlink to our work, in that work there is no way to compose services to address user's complex queries (i.e. the work assumes that a query can be always resolved by one service). Furthermore, the use of OWL-S language alone does not allow to specify explicitly the relationship between inputs and outputs of a DaaS service which may lead to errors in the service matching phase. Further, IBHIS relies on SNOMED as a mediated ontology which is a restriction, as in our vision we should remain independent from a specific ontology.

Hristoskova et al. [20] present an implementation of a dynamic and automatic composer for medical support services in the ICU (The Intensive Care Unit). The composition is achieved by semantically described Web services in order to provide automatic WS-BPEL composition. In comparison with our work, this solution does not address the EHR data integration using WS composition. Also, it is restricted to intensive care unit in hospital where data heterogeneity is not a real concern.

In ARTEMIS project [12] [8] ensures the semantic interoperability of Electronic Health Records and medical applications through using the Web service technology. It presents a mechanism for publishing, discovering and invoking semantically enriched Web services in Peer-to-Peer medical data sharing environments. Web services are annotated with OWL medical ontologies. However, the ARTEMIS project does not provide means to compose individual medical data services to answer the user's complex queries. Furthermore, heterogeneities at the data value level was not addressed in that work.

7.3 Query Rewriting Approach for Web Service Composition

Automatic Web service composition approaches can be classified according to the techniques adopted to solve the composition problem into: AI planning based approaches, workflow based approaches and query rewriting based approaches [6,40,27,5,36,38,32]. We review in the following some works in the last category as they relate to our approach.

Lu et al. [27] provide a framework for answering queries with a conjunctive plan that includes input and outputs of participating Web services annotated with Datalog expressions. In [36], a combination of inverse rules query reformulation algorithm and tuple filtering are used to generate a universal integration plan or a composition to answer user queries. However, those works do not take into account the semantics of the services during the matching phase; i.e. services are matched based on types matching. In order to address this problem, Bao et al. [5] proposed a semantic query rewriting approach using the CARIN [15] language. However, unlike to our work the semantic query rewriting in that work is limited to one-to-one matching; i.e. the work assumes that a query can be resolved by one service and there is now need to service combination.

Also, Zhou et al. [40] introduce an ontology-based approach to publishing and composing data-intensive Web services. They propose an extension to the description capability of OWL-S. This extension has the form of a SPARQL query defining semantic content and constraints on data published by the service. Also, an algorithm to generating service composition based on ontology language and graph-based planning are outlined. However, our work relies on a more efficient RDF query rewriting algorithm [6] that uses many optimization heuristics to out speed their proposed algorithm. Furthermore, we address data values heterogeneities in the obtained compositions.

Furthermore, Vaculin et al. [38] describes mechanisms for specification of generic data providing services using RDF views. They provided a characterization of matching conditions for DaaS services and developed an algorithm for matching with calculation of a matching degree between service requests and service advertisements. However, that work overlooks the relationships between matched outputs and inputs in the RDF graph of the DaaS service which may lead to erroneous matching results.

8 Perspectives and Conclusions

In this paper, we proposed an approach to automatically compose EHR DaaS Web services published by heterogeneous health information systems that employ different EHR data standards. The proposed approach follows a local-as-view paradigm by explicitly requiring a two-level mediated ontology. The first level models the generic data concepts and their inter-relationships while the second allows specifying in a declarative way how a concept of the first level is represented in different health ontologies and data standards. These ontologies are utilized to annotate EHR DaaS and mediation services on the one hand, and to specify user queries on the other hand. User queries are rewritten in terms of EHR DaaS services using an efficient query rewriting algorithm. Furthermore, our approach makes use of mediation services to handle the semantic heterogeneity of exchanged data.

As future work, we intend to improve our mediation approach in order to address structural-level incompatibilities, as well as complex data transformations between input and output parameters of EHR DaaSs. In order to realize this objective, we plan to study different solutions for the composition of mediation services to ensure a complete mediation in EHRs DaaS composition. In addition, we intend to include that data quality aspects (e.g. data trustworthiness and provenance, etc) of data provided by the EHR DaaSs in our composition approach. We will investigate the use of RDF reification and named graphs in that respect.

References

1. Angulo, C., Crespo, P., Maldonado, J.A., Moner, D., Pérez, D., Abad, I., Mandingorra, J., Robles, M.: Non-invasive lightweight integration engine for building ehr from autonomous distributed systems. International Journal of Medical Informatics 76(suppl.3), 417–424 (2007)
2. Anzböck, R., Dustdar, S.: Modeling and implementing medical web services. Data & Knowledge Engineering 55(2), 203–236 (2005)
3. Arguello, M., Des, J., Perez, R., Fernandez-Prieto, M., Paniagua, H.: Electronic health records (ehrs) standards and the semantic edge: A case study of visualising clinical information from ehrs. In: International Conference on Computer Modeling and Simulation, pp. 485–490 (2009)
4. Austin, T., Kalra, D., Tapuria, A., Lea, N., Ingram, D.: Implementation of a query interface for a generic record server. I. J. Medical Informatics 77(11), 754–764 (2008)
5. Bao, S., Zhang, L., Lin, C., Yu, Y.: A semantic rewriting approach to automatic information providing web service composition. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 488–500. Springer, Heidelberg (2006)
6. Barhamgi, M., Benslimane, D., Medjahed, B.: A query rewriting approach for web service composition. IEEE Transactions on Services Computing 3, 206–222 (2010)
7. Beale, T.: Archetypes: Constraint-based Domain Models for Futureproof Information Systems
8. Bicer, V., Laleci, G.B., Dogac, A., Kabak, Y.: Artemis message exchange framework: semantic interoperability of exchanged messages in the healthcare domain. SIGMOD Rec. 34, 71–76 (2005)
9. Budgen, D., Rigby, M., Brereton, P., Turner, M.: A data integration broker for healthcare systems. Computer 40, 34–41 (2007)

10. Calvanese, D., Giacomo, G.D., Lenzerini, M., Nardi, D., Rosati, R.: A principled approach to data integration and reconciliation in data warehousing. In: Proceedings of the International Workshop on Design and Management of Data Warehouses, DMDW 1999 (1999)
11. Cruz-Correia, R., Vieira-Marques, P., Ferreira, A., Almeida, F., Wyatt, J., Costa-Pereira, A.: Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs. BMC Medical Informatics and Decision Making 7(1), 14 (2007)
12. Dogac, A., Laleci, G.B., Kirbas, S., Kabak, Y., Sinir, S.S., Yildiz, A., Gurcan, Y.: Artemis: Deploying semantically enriched web services in the healthcare domain. Information Systems 31(4-5), 321–339 (2006); The Semantic Web and Web Services
13. Dolin, R., Alschuler, L., Beebe, C., Biron, P., Boyer, S., Essin, D., Kimber, E., Lincoln, T., Mattison, J.: The HL7 clinical document architecture. Journal of the American Medical Informatics Association 8(6), 552 (2001)
14. Eichelberg, M., Aden, T., Riesmeier, J., Dogac, A., Laleci, G.B.: A survey and analysis of electronic healthcare record standards. ACM Comput. Surv. 37(4), 277–315 (2005)
15. Goasdoué, F., Lattès, V., Rousset, M.-C.: The use of carin language and algorithms for information integration: The picsel system. Int. J. Cooperative Inf. Syst. 9(4), 383–401 (2000)
16. Grimson, J., Stephens, G., Jung, B., Grimson, W., Berry, D., Pardon, S.: Sharing health-care records over the internet. IEEE Internet Computing 5, 49–58 (2001)
17. Halevy, A.Y.: Theory of answering queries using views. SIGMOD Rec. 29(4), 40–47 (2000)
18. Halevy, A.Y.: Answering queries using views: A survey. The VLDB Journal 10, 270–294 (2001)
19. Hori, M., Ohashi, M.: Applying xml web services into health care management. In: Hawaii International Conference on System Sciences, vol. 6, p. 155a (2005)
20. Hristoskova, A., Moeyersoon, D., Hoecke, S.V., Verstichel, S., Decruyenaere, J., Turck, F.D.: Dynamic composition of medical support services in the icu: Platform and algorithm design details. Computer Methods and Programs in Biomedicine 100(3), 248–264 (2010)
21. Kart, F., Miao, G., Moser, L., Melliar-Smith, P.: A distributed e-healthcare system based on the service oriented architecture. In: IEEE International Conference on Services Computing, SCC 2007, pp. 652–659 (2007)
22. Katehakis, D.G., Sfakianakis, S., Kavlentakis, G., Anthoulakis, D.N., Tsiknakis, M.: Delivering a lifelong integrated electronic health record based on a service oriented architecture. IEEE Transactions on Information Technology in Biomedicine 11(6), 639–650 (2007)
23. Lee, Y., Patel, C., Chun, S.A., Geller, J.: Towards intelligent web services for automating medical service composition. In: Proceedings of the IEEE International Conference on Web Services, ICWS 2004, p. 384. IEEE Computer Society Press, Washington, DC, USA (2004)
24. Lenz, R., Beyer, M., Kuhn, K.A.: Semantic integration in healthcare networks. International Journal of Medical Informatics 76(2-3), 201–207 (2005); Connecting Medical Informatics and Bio-Informatics - MIE 2005
25. Liu, Q., Huang, T., Liu, S.-H., Zhong, H.: An ontology-based approach for semantic conflict resolution in database integration. Journal of Computer Science and Technology 22, 218–227 (2007), doi:10.1007/s11390-007-9028-4
26. Liu, S., Ni, Y., Mei, J., Li, H., Xie, G., Hu, G., Liu, H., Hou, X., Pan, Y.: ismart: Ontology-based semantic query of cda documents

27. Lu, J., Yu, Y., Mylopoulos, J.: A lightweight approach to semantic web service synthesis. In: WIRI 2005, Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration, pp. 240–247 (2005)
28. Mykkänen, J., Riekkinen, A., Sormunen, M., Karhunen, H., Laitinen, P.: Designing web services in health information systems: From process to application level. International Journal of Medical Informatics 76(2-3), 89–95 (2005)
29. Nagarajan, M., Verma, K., Sheth, A., Miller, J.: Ontology driven data mediation in web services. International Journal of Web Services Research 4(4), 104–126 (2007)
30. Orgun, B., Vu, J.: Hl7 ontology and mobile agents for interoperability in heterogeneous medical information systems. Computers in Biology and Medicine 36(7-8), 817–836 (2006); Special Issue on Medical Ontologies
31. Ram, S., Park, J.: Semantic conflict resolution ontology (scrol): an ontology for detecting and resolving data and schema-level semantic conflicts. IEEE Transactions on Knowledge and Data Engineering 16(2), 189–202 (2004)
32. Rao, J., Su, X.: A survey of automated web service composition methods. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005)
33. Sachdeva, S., Bhalla, S.: Semantic Interoperability in Healthcare Information for EHR Databases. In: Kikuchi, S., Sachdeva, S., Bhalla, S. (eds.) DNIS 2010. LNCS, vol. 5999, pp. 157–173. Springer, Heidelberg (2010)
34. Saleh, I., Kulczycki, G., Blake, M.B.: Demystifying data-centric web services. IEEE Internet Computing 13, 86–90 (2009)
35. Stroetmann, K., Stroetmann, V.: Towards an Interoperability Framework for a European e-Health Research Area–Locating the Semantic Interoperability Domain. In: EC Workshop on semantic interoperability, Brussels, pp. 14–15 (February 2005)
36. Thakkar, S., Ambite, J.L., Knoblock, C.A.: A data integration approach to automatically composing and optimizing web services. In: Proceedings of the ICAPS Workshop on Planning and Scheduling for Web and Grid Services (2004)
37. Truong, H.L., Dustdar, S.: On analyzing and specifying concerns for data as a service. In: APSCC, pp. 87–94 (2009)
38. Vaculín, R., Chen, H., Neruda, R., Sycara, K.: Modeling and discovery of data providing services. In: Proceedings of the 2008 IEEE International Conference on Web Services, pp. 54–61. IEEE Computer Society, Washington, DC, USA (2008)
39. Wright, A., Sittig, D.F.: Sands: A service-oriented architecture for clinical decision support in a national health information network. J. of Biomedical Informatics 41, 962–981 (2008)
40. Zhou, L., Chen, H., Wang, J., Zhang, Y.: Semantic web-based data service discovery and composition. In: International Conference on Semantics, Knowledge and Grid, pp. 213–219 (2008)