RESEARCH ARTICLE

Ontology-Based Data Integration between Clinical and Research Systems

Sebastian Mate[1]*, Felix Köpcke[2], Dennis Toddenroth[1], Marcus Martin[3], Hans-Ulrich Prokosch[1,2], Thomas Bürkle[4‡], Thomas Ganslandt[2‡]

1 Institute for Medical Informatics, University Erlangen-Nuremberg, Erlangen, Germany, **2** Center for Medical Information and Communication, Erlangen University Hospital, Erlangen, Germany, **3** Tumor Centre, Erlangen University Hospital, Erlangen, Germany, **4** Institute for Medical Informatics, Bern University of Applied Sciences, Bern, Switzerland

‡ These authors contributed equally to this work.
* sebastian.mate@imi.med.uni-erlangen.de

Citation: Mate S, Köpcke F, Toddenroth D, Martin M, Prokosch H-U, Bürkle T, et al. (2015) Ontology-Based Data Integration between Clinical and Research Systems. PLoS ONE 10(1): e0116656. doi:10.1371/journal.pone.0116656

Academic Editor: Pal Bela Szecsi, Gentofte University Hospital, DENMARK

Received: April 7, 2014

Accepted: December 6, 2014

Published: January 14, 2015

Data Availability Statement: The software and source code is now available for download on our website (http://www.imi.med.uni-erlangen.de/tools/ontoimportsuite/). In addition we supply a demonstration subset of the diverse ontology contents which is needed to replicate the methodology. Furthermore we supply the SQL script used to generate the HIS ontology.

Funding: The authors acknowledge support by Deutsche Forschungsgemeinschaft (DFG) and Friedrich-Alexander-University Erlangen-Nuremberg within the funding programme Open Access Publishing. The funders had no role in study design, data collection

Abstract

Data from the electronic medical record comprise numerous structured but uncoded elements, which are not linked to standard terminologies. Reuse of such data for secondary research purposes has gained in importance recently. However, the identification of rele-vant data elements and the creation of database jobs for extraction, transformation and loading (ETL) are challenging: With current methods such as data warehousing, it is not feasible to efficiently maintain and reuse semantically complex data extraction and trans-formation routines. We present an ontology-supported approach to overcome this challenge by making use of abstraction: Instead of defining ETL procedures at the database level, we use ontologies to organize and describe the medical concepts of both the source system and the target system. Instead of using unique, specifically developed SQL statements or ETL jobs, we define declarative transformation rules within ontologies and illustrate how these constructs can then be used to automatically generate SQL code to perform the desired ETL procedures. This demonstrates how a suitable level of abstraction may not only aid the interpretation of clinical data, but can also foster the reutilization of methods for un-locking it.

Introduction and Background

Reusing clinical routine care data in single source projects [1] has gained in importance recently [2–5]. The data are used for feasibility studies, patient recruitment, the execution of clinical trials [6–10], clinical research [11–14] and biobanking [15].

Routine care data can roughly be classified into three categories: *(1) unstructured free text,* which is used for flexible documentation items such as discharge letters, clinical notes and findings, *(2) structured and coded data elements,* which are coded according to standardized terminologies and are typically used for billing and *(3) structured but uncoded data elements,* which are used in assessment forms of electronic medical records (EMRs).

The first type, unstructured free text, provides the most comprehensive information, because it does not restrict the clinical user during the documentation process [16]. An

and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

automated analysis, however, requires complex *natural language processing* methods [17]. The second type, structured and coded data, is easier to process, but is limited in terms of expressiveness and credibility [18, 19]. In this manuscript we concentrate on structured but uncoded data, the third type. It encodes information with enumerable value lists that are, however, not linked to any standard terminology. Many EMR forms comprise such data. Its reuse is challenging due to the following four reasons:

A **It is difficult to process non-standardized data elements:** Assessment forms are often designed to mimic classic paper sheets. They are typically used to record events during the hospitalization, e.g. the medical history, examinations, surgical procedures and different pathological findings and are often highly customized. According to [20], assessment forms *"[. . .] have been developed to fulfill the specific requirements of the hospital unit, and the data is described according to the definitions of assessments and concepts that are used locally"*. Thus, the value sets are often not linked to standard classifications. For example, the data element 'sex' might be encoded with the value set {female, male} in one form and with {F, M} or {1, 2} in another. Although some of these concepts could be separately linked to standard terminologies (in the example we could link {female, male} to the standardized SNOMED-CT [21] codes {248152002, 248153007}), many other value lists are use-case-specific and cannot be mapped.

B **EMRs lack knowledge management functions:** Most EMR systems do not offer data dictionaries [22] with clear concept definitions to enable the reuse of data elements in multiple forms [23], although their advantages have been known for a long time [24]. Instead of defining and reusing concepts such as *weight, height* or *smoker status*, these elements are frequently redefined for each form. Over time, this results in an accumulation of inconsistent concept naming and value sets within new EMR forms and complicates data extraction and interpretation, because the redundant data elements have to be identified and merged.

C **Contextual semantic relationships between data elements and forms are lost:** A clinical user considers the medical context, the structure of the assessment form and the neighboring data elements when entering new data. He would, for example, understand a TNM [25] documentation field in a pathology form as a *pathological TNM* and not as a *clinical TNM*. The TNM is a classification to describe a patient's cancer status in terms of tumor size, affected lymph nodes and metastases. However, such implicit relationships are not stored in most clinical systems. When extracting the data, the data engineer has to manually review the forms and remodel these semantic relationships in his database transactions.

D **It is a challenge to integrate data from different institutions:** Previous efforts to integrate data from different EMRs [26–31] demonstrated success, but also identified challenges if the semantic representation between the EMR sources differed. Merging data between a hospital EMR and a cancer registry record for instance turned out to be difficult, because EMR data was linked to the patient but the cancer registry distinguished between cancer treatment applied to the main tumor and treatment of metastases and recurrence [31]. Such problems resulted in large manual efforts spent for the data integration in recent cross-institutional research projects [26, 27, 30].

Objective

Today a data engineer is required to address these challenges while preparing EMR data for reuse. The implicit knowledge gained in the extraction process, e.g. about data context and provenance, is conventionally not recorded in a universally machine-processible format and therefore is lost. The data extraction, transformation and loading (ETL) procedures are unique for each database system and cannot be reused. The complex ETL procedures are difficult to understand and to maintain.

Our goal was to develop a method that is based on declarative, universally machine-processible but also human-readable and easily maintainable ETL definitions that can be translated into automated database transactions. Our approach incorporates the ETL know-how in an ontological system that governs the correct extractions and data transformations.

Methods

The EMR in Erlangen

The Erlangen University Hospital is a 1,360-bed tertiary care unit in southern Germany. It deployed the EMR system Soarian Clinicals by Siemens [32] in 2003. In the following years, the system was rolled out in all clinical specialties for order entry, results reporting as well as medical and nursing documentation. Today, the EMR is used in more than 90 wards, in functional units such as echocardiography and in outpatient clinics by more than 2,800 registered users. It supports the design of custom assessment forms and workflows for specialized purposes. Using this toolbox, extensive electronic documentation instruments had been established for many clinical specialties in recent years. For example, detailed assessment forms for prostate, mamma, thorax, and colorectal carcinoma have been provided to support patient care in the Erlangen comprehensive cancer center [33]. Today, the EMR comprises 785 different assessment forms, which contain 28,055 data elements with 35,301 distinct selectable values. The system stores data of approximately 1,150,000 patients.

Several projects at Erlangen University Hospital reuse structured but uncoded EMR data in cross-institutional research settings [33–35]. In these projects we were confronted with the problems described in the introduction.

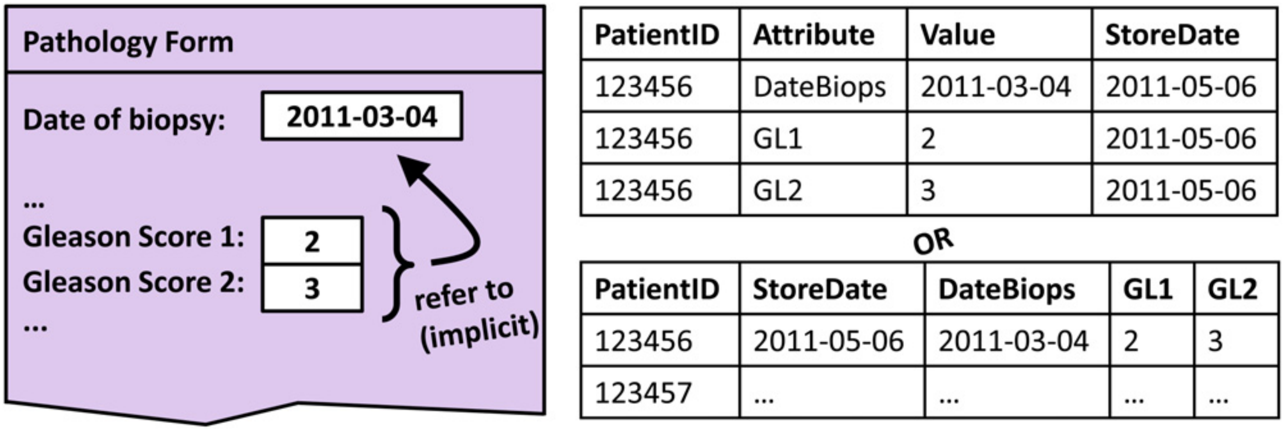
Fig. 1 illustrates a typical example. The Gleason Score describes the microscopic appearance of prostate tumors. Cell differentiation of the most common and the second most common tumor pattern are rated on a five-point scale from grade 1 (well differentiated) to grade 5 (poorly differentiated). The sum of both is the *Gleason Score*. Each Gleason Score thus consists of three parts (e.g. 2 + 3 = 5), which are denoted as *Gleason Score 1*, *2* and *3* in the EMR. An additional date field stores the time stamp of the biopsy. The EMR database, however, does not store this relationship explicitly, but treats all data elements separately. Thus the scores are attributed with the storage date of the assessment form (2011–05–06), while according to the clinical meaning, the reference date should be the value of the biopsy date element (2011–03–04).

While this exemplary ETL task may be easy to solve, one must bear in mind that typical single source projects require dozens or even hundreds of patient characteristics. Thus, the identification of all relevant data elements from large EMRs with tens of thousands of data elements, their semantic harmonization and the continuous maintenance of this ETL is a Sisyphean struggle.

We now describe our ontology-based approach that aims to simplify and support the mapping, extraction and data transfer processes.

Ontological representation of source and target systems and mappings

In a first step, we define the ETL process as a declarative representation that is stored in ontologies. In the scope of this paper we understand an ontology to be a directed graph. The graph's nodes represent entities while the edges describe relationships between them. Two nodes that are connected via an edge are called a triple. We use the Semantic Web [36] standard *Resource Description Framework* (RDF) [37, 38], where nodes are termed *resources* and edges are termed



Pathology Form

Date of biopsy: 2011-03-04

...

Gleason Score 1: 2

Gleason Score 2: 3

...

PatientID	Attribute	Value	StoreDate
123456	DateBiops	2011-03-04	2011-05-06
123456	GL1	2	2011-05-06
123456	GL2	3	2011-05-06

OR

PatientID	StoreDate	DateBiops	GL1	GL2
123456	2011-05-06	2011-03-04	2	3
123457	...	...	...	...

Figure 1. Generic databases do not reflect semantic relationships between data elements. The two (equivalent) database tables on the right side (EAV-like style on top, column-oriented style below) do neither reflect the form's structure as it is visible to the application user (left side), nor the semantic relationships between different input elements.

doi:10.1371/journal.pone.0116656.g001

properties. We also use some constructs from RDF Schema [39] and the Web Ontology Language (OWL) [40], although these are generously simplified for the readability of this paper and its illustrations. For example, we omit the distinction between classes and instances. However, this has no impact on the validity of our approach. The supplied appendix in S1 File distinguishes between classes and instances.

The upper part of Fig. 2 shows a typical ETL process, where data records have to be extracted from a source system (EMR database, left), transformed (black arrow, center) and then loaded into a target system (research database, right). The lower part of the figure illustrates our abstraction approach with ontologies. Our key concept is to express each of the three ETL steps with an ontology: A *source ontology* corresponds to the extraction, whereas a *target ontology* corresponds to the loading. By creating connections between these two ontologies in a *mapping ontology*, the user defines how data is to be transformed between both. Later, the mapping ontology can be automatically translated into executable SQL transactions.

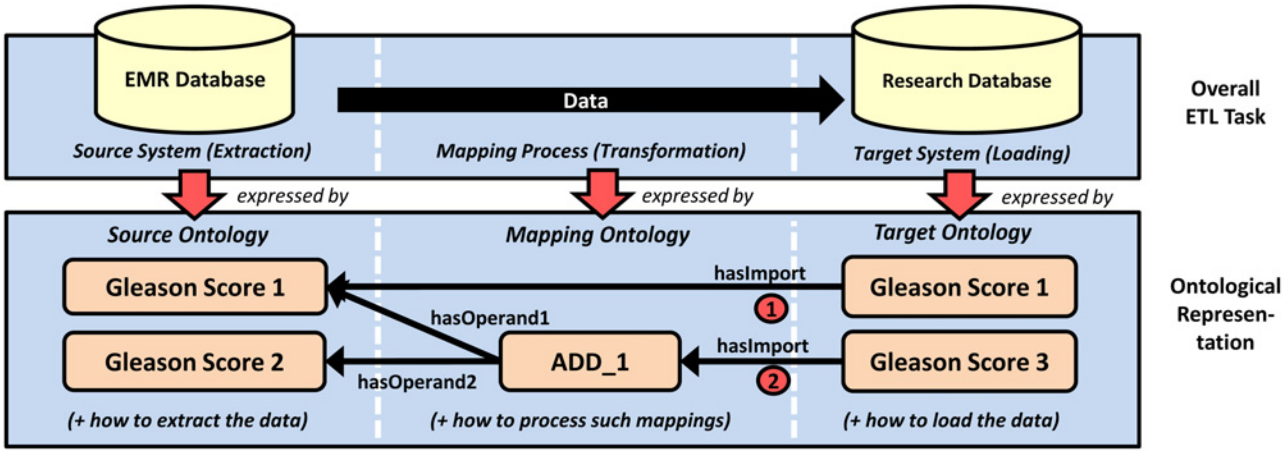


Figure 2. ETL steps are represented with ontologies. Components and processes involved in the extraction, transformation and loading of data are represented with ontologies. The mappings (1) and (2) illustrate "simple" and "complex" mappings, respectively.

doi:10.1371/journal.pone.0116656.g002

The source ontology

The *source ontology* is used to describe the contents of the source system's database. It serves two important functions. First, it acts as an inventory of all available medical data elements in the source system. These concepts can be organized in hierarchies to reflect the content structure of the source system. This allows the user to easily navigate the ontology and to select relevant concepts while creating the mappings.

Second, it provides an inventory-to-database-schema mapping by abstracting database record sets with ontology concepts. A source ontology concept *Gleason Score 1,* for example, represents the set of all Gleason Score 1 data in the source system. This record set is a list of all patient IDs, for which at least one *Gleason Score 1* is available. Additional columns for value and timestamp next to the patient IDs later allow comparisons and computations between multiple lists, e.g. it will become possible to sum the records of the lists "Gleason Score 1" and "Gleason Score 2" to derive a new "Gleason Score 3" list. We call this schema of three columns (PatientID, Value, Date) our *internal data model.*

Fig. 3 illustrates how this inventory-to-database-schema mapping is achieved using ontologies. The medical concept *Gleason Score 1* from the source ontology's "inventory" is connected to a table instance *MyEMRTable* with a *hasSourceTable* relationship, which is linked to a database connection instance *MyDBConnection*. These instances use RDF datatype properties (i.e. string values) to store information about the database connection and the table schema. A software component can process this information and map the source database schema to the internal data model. The source table column *PATID* from the source system for example is translated to the *PatientID* column of the internal data model using the statement *MyEMRTable hasPatientIDColumn "PatID"*. Respectively, the properties *hasValueColumn* and *hasDateColumn* provide the mappings to the columns *Value* and *Date* in the internal data model.

Some clinical EMR systems, such as Siemens Soarian [32] or Epic [41], store data in an entity-attribute-value format [42, 43]. In this case additional filter criteria are necessary to retrieve only the data records that are associated with the desired concept. They are implemented with *hasSelectFilter* datatype properties, which are connected to each source ontology concept

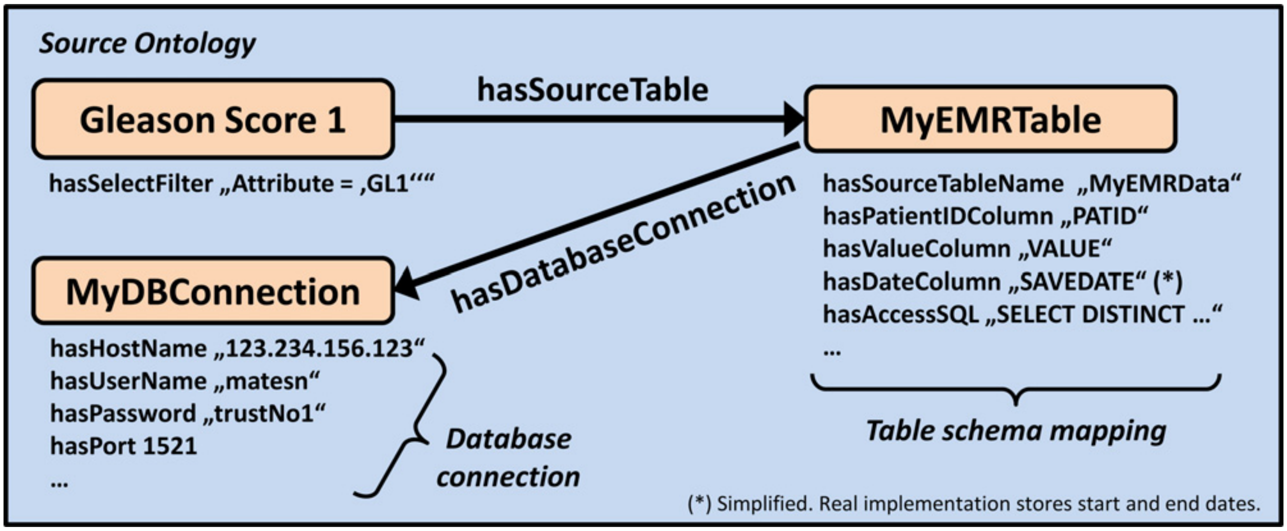

Source Ontology
Gleason Score 1
hasSourceTable
MyEMRTable
hasSelectFilter „Attribute = ‚GL1'"
hasDatabaseConnection
MyDBConnection
hasSourceTableName „MyEMRData"
hasPatientIDColumn „PATID"
hasValueColumn „VALUE"
hasDateColumn „SAVEDATE" (*)
hasAccessSQL „SELECT DISTINCT …"
...
hasHostName „123.234.156.123"
hasUserName „matesn"
hasPassword „trustNo1"
hasPort 1521
...
Database connection
Table schema mapping
(*) Simplified. Real implementation stores start and end dates.

Figure 3. Database bindings are described with ontologies. By linking the medical concept "Gleason Score 1" to additional ontology concepts that describe the database schema and connection, a software component can construct SQL to retrieve the data records. *Note: To save space within the figures, datatype properties are printed below the concepts. These statements have to be read like this: MyEMRTable hasSourceTableName "MyEMRData".*

doi:10.1371/journal.pone.0116656.g003

(see Fig. 3, below the *Gleason Score 1* concept). We can now construct an SQL statement that returns all desired Gleason Score 1 records in the internal data model:

SELECT DISTINCT **PATID** PatientID, **VALUE** Value, **SAVEDATE** Date

FROM **MyEMRData** WHERE **Attribute = 'GL1'**

The underlined parts are retrieved from the ontologies (see Fig. 3) and inserted into an SQL template, which is also stored in the ontology (*hasAccessSQL* property). Different templates can be created to access different database schemas. This template-based SQL code generation is used throughout our approach to achieve the necessary database bindings.

The target ontology

The *target ontology* is similar to the source ontology, but used for loading instead of extracting data. It represents a domain ontology, because it describes the collection of medical concepts to be loaded into the target system (the target dataset). In cross-institutional settings where multiple sites share their data in a central research database, the target dataset has to be defined before the creation of the target ontology. Such data elements are also called *common data elements* [44, 45].

The target ontology contains syntactic and semantic information that is linked to each concept and is used to generate the metadata for the target system. For demonstration purposes we chose *Informatics For Integrating Biology And The Bedside* (i2b2) [46], an open source research platform that can be used to identify patient cohorts, as an exemplary target system. Therefore, the target ontology has to implement the semantic features of i2b2. These include, for example, the data type of the concept, a short textual description, and, if applicable, the unit of measurement for numeric values and further attributes such as lab value ranges. Tables A-D in S1 File list all ontology constructs.

The mapping ontology

The *mapping ontology* connects the target ontology to the source ontology with manually created semantic relationships between medical concepts. We distinguish between simple and complex mappings. Simple mappings with a *hasImport* property express that the connected concepts share the same meaning (see Fig. 2, mapping (1)). Complex mappings are used whenever data transformation is required. In the mapping ontology, they are represented by intermediate nodes that express a filter operation or data transformation between the target node and exactly two operand nodes (e.g. *ADD_1*, see Fig. 2, mapping (2)). The different properties *hasOperand1* and *hasOperand2* allow the definition of non-commutative operations. Mapping nodes can be cascaded to full expression trees to support composed operations as shown in Fig. 4.

In addition, we have to define the processing method of complex mapping nodes. Fig. 5 shows once again the complex mapping node *ADD_1* from Fig. 2, which was used to define *Gleason Score 3* as the summation of the two operands *Gleason Score 1* and *Gleason Score 2.* It illustrates that *ADD_1* is connected to a command definition, *ADD*. The value of the *hasOutputTransformation* datatype property is an SQL database operation that adds the entries of the *Value* column of the two operand record sets OP1 and OP2 (as stated above, the *Value* column is part of the internal data model). The content of *hasSelectFilter* ensures that values for both operands exist. The *hasDateValue* property expresses that the time stamp for the result set (Gleason Score 3) has to be taken from OP1 (Gleason Score 1). The relevant ontology constructs, including the currently implemented operations, are shown in Tables E-J in S1 File.

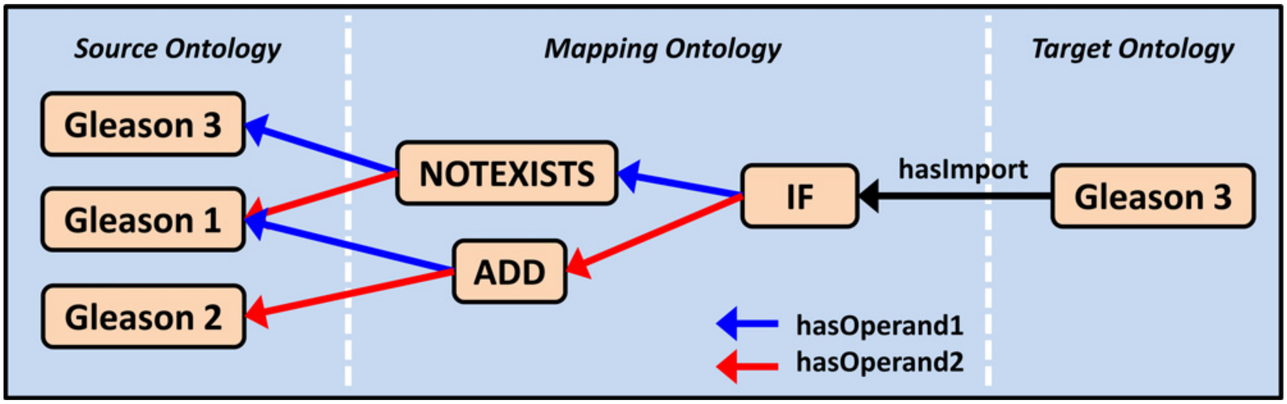


Figure 4. Cascading of mapping nodes. Cascaded mapping nodes allow the definition of arbitrary data transformations. The illustration has to be read from the right to the left, hasOperand1 before hasOperand2. Paraphrased, it means: If no data for Gleason 3 (left side) exist, add the Gleason 1 and Gleason 2 data and export these as Gleason 3 (right side) records. Details about the NOTEXISTS, ADD and IF nodes' semantics and why NOTEXISTS requires a second operand are given in Tables F and G in S1 File.

doi:10.1371/journal.pone.0116656.g004

Second step of the automated process: Translation of the ontologies into SQL

A software component populates an SQL template with information stored in the ontologies. It generates an SQL statement for each mapping node in the mapping ontology. Fig. 6 shows a populated SQL template, which processes the mapping node *ADD_1* from Fig. 2 and stores the result in a temporary database table. The SQL template is the same for all node types, including arithmetic, relational and string processing operations (see Tables F-H in S1 File for additional examples). The statement initially fetches the data records for both operand nodes (result sets OP1 and OP2) according to the definition in the source ontology (lines 21–24 and 28–31). Both result sets are retrieved in an internal data model, which comprises six columns *DocumentID*, *PatientID*, *CaseID*, *DateStartValue*, *DateEndValue* and *Value*. The SQL statement joins both result sets on the entity (*DocumentID*, lines 26 and 33). This allows computations between data elements from the same form. To perform the data transformation of the mapping node, the statement applies the specified database operation (line 15–17) and filter (line 35), which were described in the *hasOutputTransformation* and *hasSelectFilter* properties. The result is written to a temporary database table (lines 1–2) that is also defined by the *hasSourceTable* property inside the ontologies.

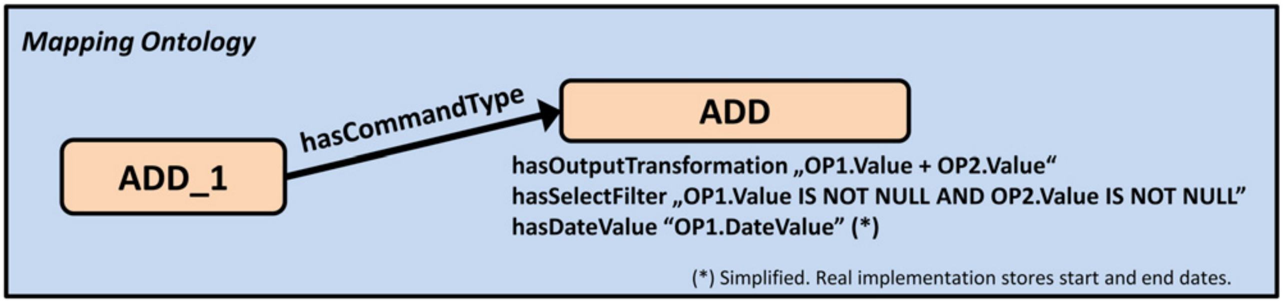
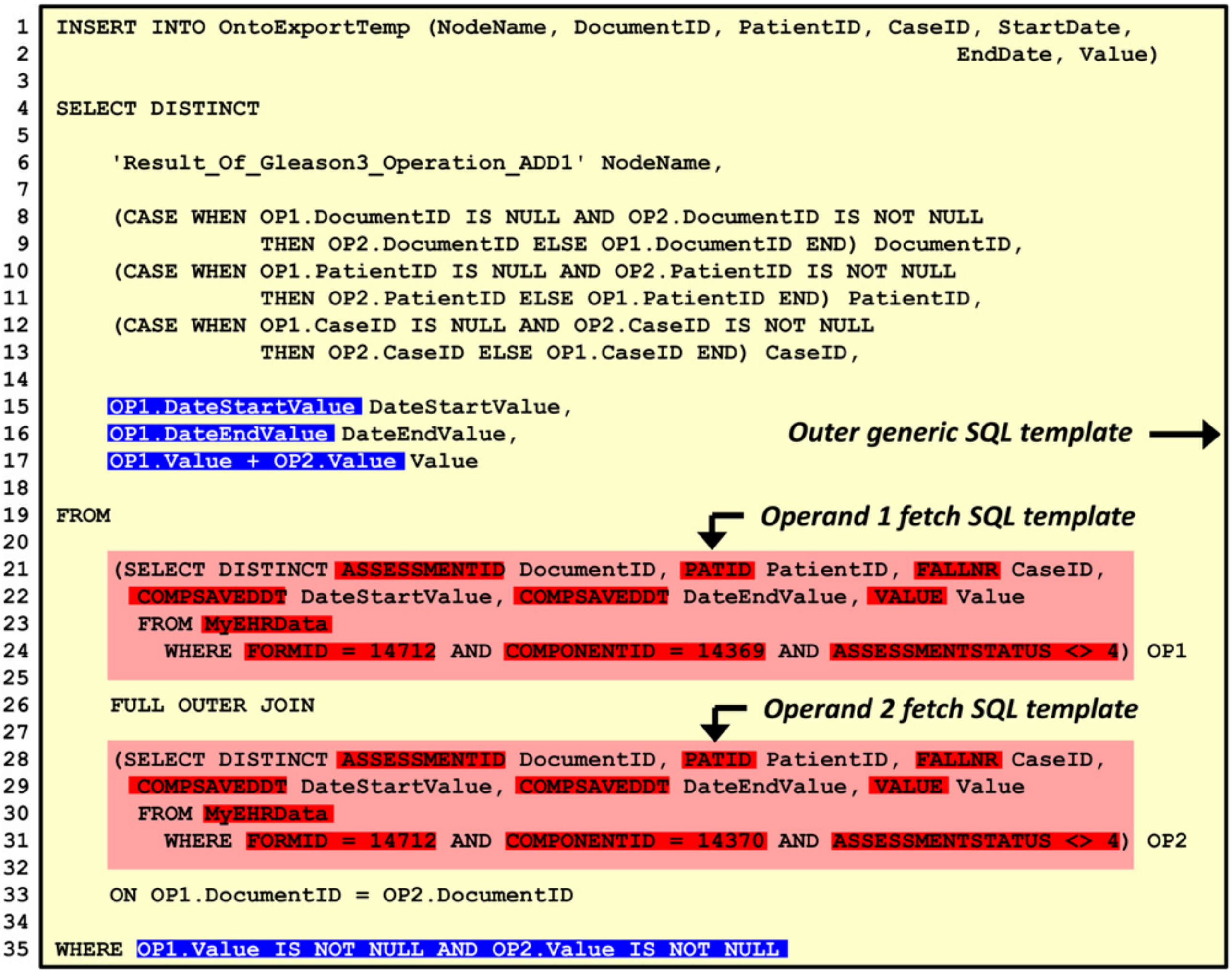

Mapping Ontology
ADD_1
hasCommandType
ADD
hasOutputTransformation „OP1.Value + OP2.Value“
hasSelectFilter „OP1.Value IS NOT NULL AND OP2.Value IS NOT NULL”
hasDateValue “OP1.DateValue” (*)
(*) Simplified. Real implementation stores start and end dates.

Figure 5. Command type definitions describe how to process mapping nodes from the mapping ontology. All intermediate nodes in the mapping ontology are connected to a command type definition. They contain SQL code fragments, which describe how to filter and transform the facts data derived from operands 1 and 2 (OP1 and OP2).

doi:10.1371/journal.pone.0116656.g005



```
 1  INSERT INTO OntoExportTemp (NodeName, DocumentID, PatientID, CaseID, StartDate,
 2                                                              EndDate, Value)
 3
 4  SELECT DISTINCT
 5
 6      'Result_Of_Gleason3_Operation_ADD1' NodeName,
 7
 8      (CASE WHEN OP1.DocumentID IS NULL AND OP2.DocumentID IS NOT NULL
 9                 THEN OP2.DocumentID ELSE OP1.DocumentID END) DocumentID,
10      (CASE WHEN OP1.PatientID IS NULL AND OP2.PatientID IS NOT NULL
11                 THEN OP2.PatientID ELSE OP1.PatientID END) PatientID,
12      (CASE WHEN OP1.CaseID IS NULL AND OP2.CaseID IS NOT NULL
13                 THEN OP2.CaseID ELSE OP1.CaseID END) CaseID,
14
15      OP1.DateStartValue DateStartValue,
16      OP1.DateEndValue DateEndValue,
17      OP1.Value + OP2.Value Value
18
19  FROM
20
21      (SELECT DISTINCT ASSESSMENTID DocumentID, PATID PatientID, FALLNR CaseID,
22        COMPSAVEDDT DateStartValue, COMPSAVEDDT DateEndValue, VALUE Value
23        FROM MyEHRData
24          WHERE FORMID = 14712 AND COMPONENTID = 14369 AND ASSESSMENTSTATUS <> 4) OP1
25
26      FULL OUTER JOIN
27
28      (SELECT DISTINCT ASSESSMENTID DocumentID, PATID PatientID, FALLNR CaseID,
29        COMPSAVEDDT DateStartValue, COMPSAVEDDT DateEndValue, VALUE Value
30        FROM MyEHRData
31          WHERE FORMID = 14712 AND COMPONENTID = 14370 AND ASSESSMENTSTATUS <> 4) OP2
32
33      ON OP1.DocumentID = OP2.DocumentID
34
35  WHERE OP1.Value IS NOT NULL AND OP2.Value IS NOT NULL
```

Outer generic SQL template
Operand 1 fetch SQL template
Operand 2 fetch SQL template

Figure 6. Software-generated SQL transaction that processes one intermediate mapping node. This real-world example shows the SQL code constructed from the example in Fig. 2 (complex mapping (2)). The inserted SQL fragments taken from the ontologies are printed in bold and highlighted in blue (how to transform the data) and red (how to access the data). The software that created this SQL code also generates the NodeName column entry in line 6.

doi:10.1371/journal.pone.0116656.g006

Cascaded mapping networks (as shown in Fig. 4) are processed sequentially during the export. To find the next node, we apply a simple rule: A mapping node can only be processed if the data it accesses (both operands' data records) are already available. All nodes inside the source ontology are considered to be ready for processing, because their data is already available in the source system's database. During the export, the export software uses SPARQL queries [47] to find a random, but valid next node. SPARQL is a query language similar to SQL, but used for RDF ontologies. The export software creates an SQL script with one SQL statement as shown in Fig. 6 for each mapping node. When executing the script on the database, it automatically extracts, transforms and transfers the data records into the target database.

Overloading internal data model properties with values from other data elements

A special mechanism allows replacing the values in the *DocumentID*, *PatientID*, *CaseID*, *DateStartValue*, *DateEndValue* columns of the internal data model with values from other concepts.

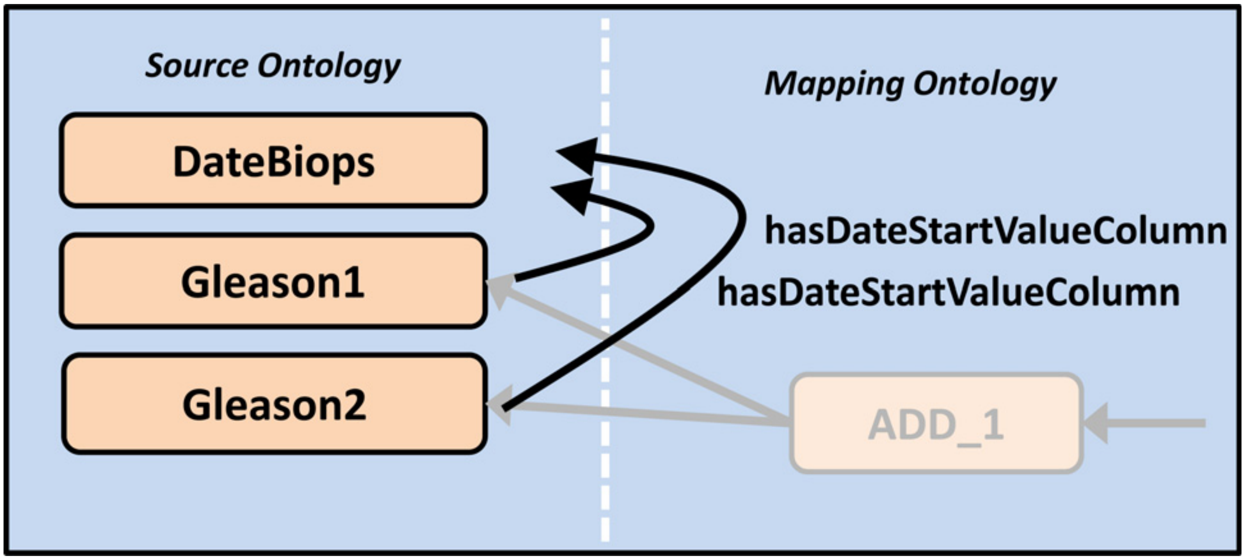


Figure 7. Overloading internal data model properties. This real-world example illustrates how semantic relationships between source data elements are stored explicitly in the ontologies and how they can be processed: Stating that *Gleason1 hasDateStartValueColumn DateBiops* e.g. tells the export software to use the data entry in the Value column of DateBiops as DateStartValue in Gleason1. Gleason2 is processed the same way.

doi:10.1371/journal.pone.0116656.g007

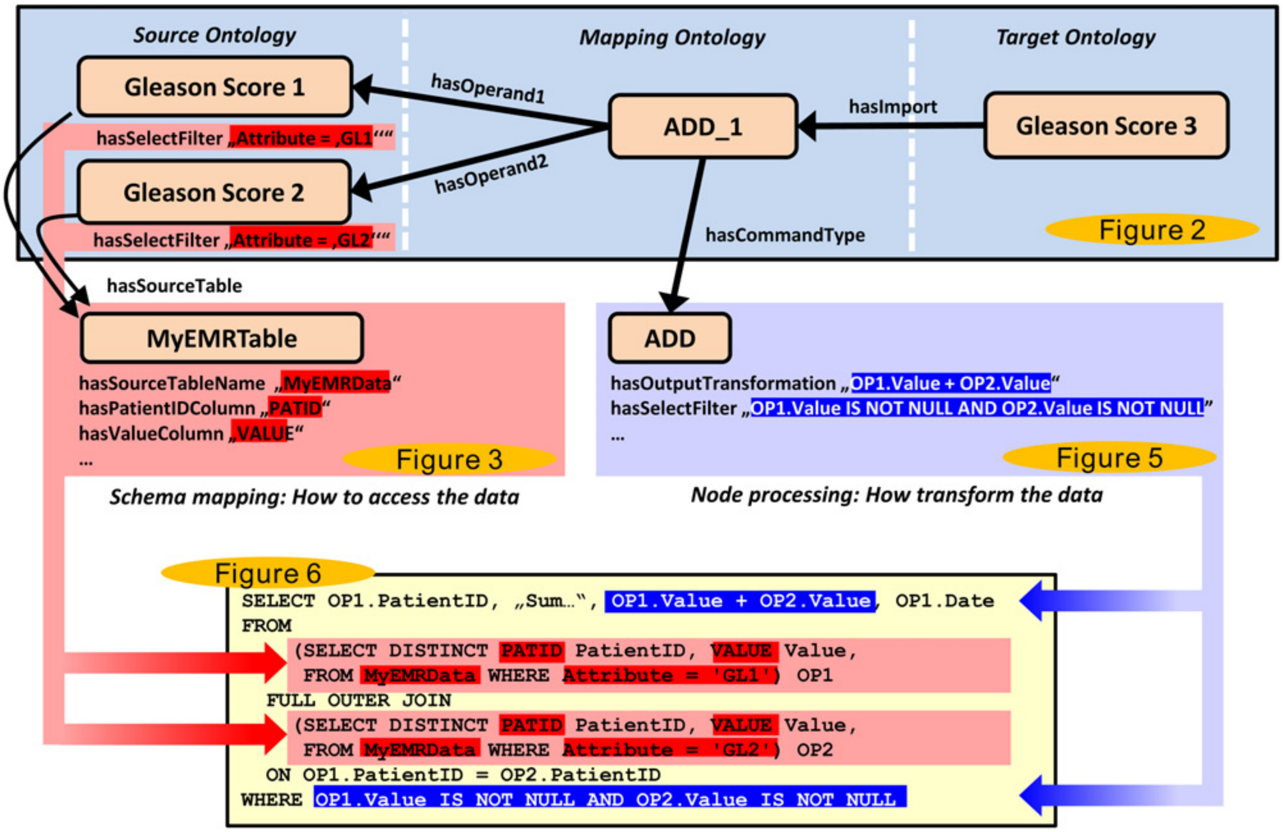
This can be done by creating ontology statements that follow the convention *ConceptA hasX-Column ConceptB*, where *X* is one of *DocumentID*, *PatientID*, *CaseID*, *DateStartValue* or *DateEndValue*.

This approach can deal with the time stamping problem that was shown in Fig. 1, where the correct time stamp of the data elements *Gleason1* and *Gleason2* was stored in a separate *DateBiops* field. Normally, the template-generated SQL code would use the storage date of the form for all data records. This is often acceptable under the assumption that the clinical documentation follows promptly the medical interventions and observations. However, in our example, a more timeliness data element *DateBiops* is available,which indicates the time when the biopsy was taken. By using the above-mentioned mechanism and by stating that *Gleason1 hasDateStartValueColumn DateBiops* and *Gleason2 hasDateStartValueColumn DateBiops* (see Fig. 7) the export software can replace the original operand-fetch SQL with other sub-selects. This in turn replaces the default *hasDateStartValueColumn* value ("2011–05–06") with the *hasValueColumn* value of *DateBiops* ("2011–03–04") of the data during the export.

Once such relationships have been defined in the source ontology, they are automatically considered in other mapping projects.

Handling of missing and erroneous values

Our approach is also capable of dealing with erroneous and missing ("NULL") values. In the example given above, the ADD node requires both operands to have existing data (see line 35 in Fig. 6 and the *hasSelectFilter* property in Fig. 5) because we specified that a Gleason Score 3 could not be calculated if one of the two operands is missing. However, we also defined "tolerant" node types, which explicitly allow one of the operands to have missing values. Depending on the operation, such NULL entries are replaced by the neutral element (0 or 1 for arithmetic operations, empty string for string operations). The use of tolerant or more stringent node types depends on the medical background of the mapping.


Source Ontology
Mapping Ontology
Target Ontology
Gleason Score 1
hasSelectFilter „Attribute = 'GL1'“
Gleason Score 2
hasSelectFilter „Attribute = 'GL2'“
hasOperand1
hasOperand2
ADD_1
hasImport
Gleason Score 3
hasCommandType
Figure 2
hasSourceTable
MyEMRTable
hasSourceTableName „MyEMRData“
hasPatientIDColumn „PATID“
hasValueColumn „VALUE“
...
Figure 3
Schema mapping: How to access the data
ADD
hasOutputTransformation „OP1.Value + OP2.Value“
hasSelectFilter „OP1.Value IS NOT NULL AND OP2.Value IS NOT NULL“
...
Figure 5
Node processing: How transform the data
Figure 6

```
SELECT OP1.PatientID, „Sum…“, OP1.Value + OP2.Value, OP1.Date
FROM
    (SELECT DISTINCT PATID PatientID, VALUE Value,
     FROM MyEMRData WHERE Attribute = 'GL1') OP1
  FULL OUTER JOIN
    (SELECT DISTINCT PATID PatientID, VALUE Value,
     FROM MyEMRData WHERE Attribute = 'GL2') OP2
  ON OP1.PatientID = OP2.PatientID
WHERE OP1.Value IS NOT NULL AND OP2.Value IS NOT NULL
```

Figure 8. Overview of the approach. The illustration shows an overview of our approach by combining several of the previous figures in a simplified fashion. The upper part (blue box) represents a mapping, which is visible to the user. The parts in the middle are internal ontology concepts that are hidden for the user. The SQL code in the lower part has been automatically compiled from the above ontologies.

doi:10.1371/journal.pone.0116656.g008

Overview

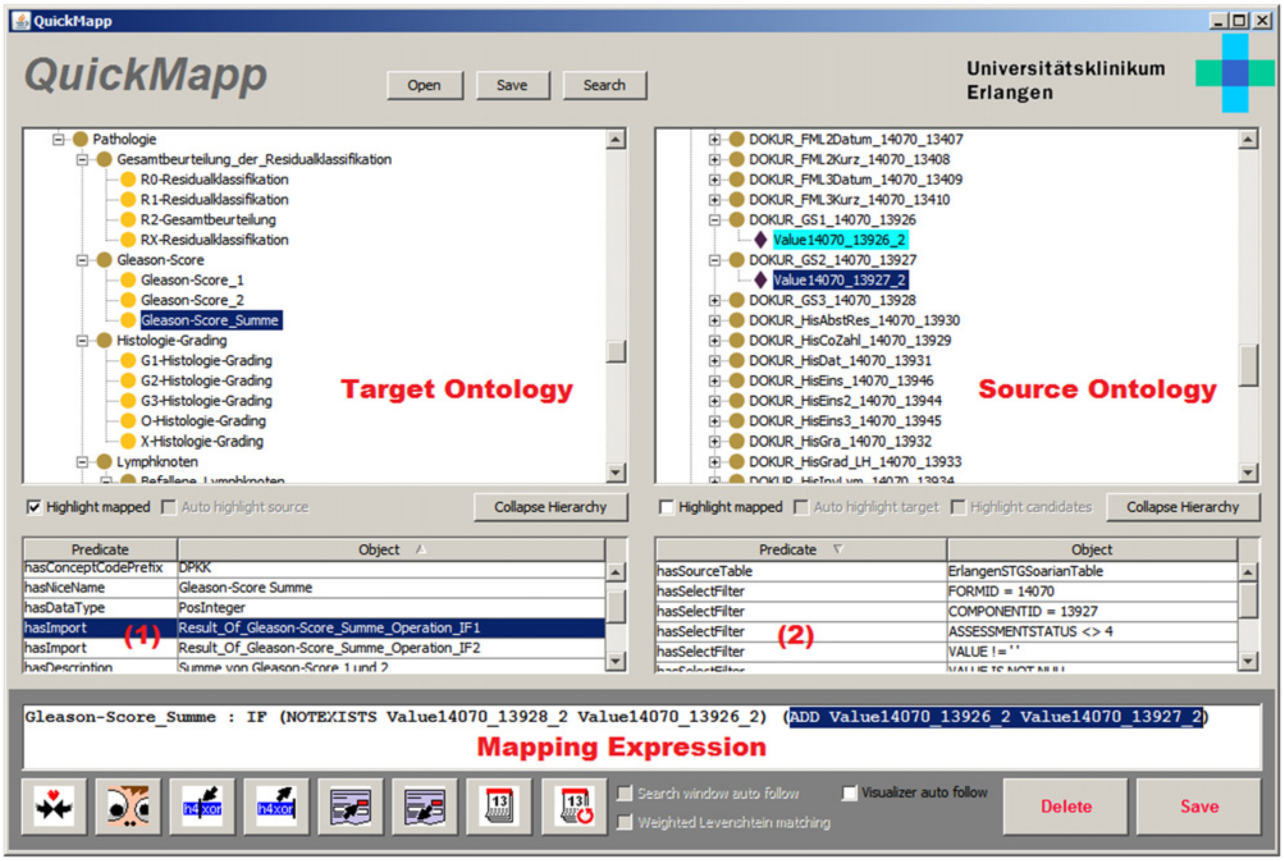
Fig. 8 provides an overview of the general approach by combining the information from the previous sections and figures. It shows how the information that is encapsulated in the ontologies (upper and middle part) is used to construct the SQL statement (lower part). The highlighted red parts represent the database schema mapping for the two operand nodes (Gleason Score 1 and 2) and describe how to perform the extraction of the source data, whereas the blue parts describe how to process the data (corresponding to the mapping).

Generation of source ontologies

An important prerequisite for our system is the generation of the source ontology. There are different options, depending on the database schema and the complexity that is used to store the metadata (i.e. names of forms and data elements). If this metadata is available in an EAV-like format, an SQL script can be used to query the contents and to create the ontology triples (an example is available online).

For relational database schemas it is more difficult to access the metadata, because it is part of the database schema (i.e. column names). One could either model the ontology manually or make use of tools for metadata discovery (e.g. [48]).

An important feature of the source ontology is the syntactic separation between the source ontology concepts and the actual source data. Syntactic separation introduces an abstraction



- Pathologie
  - Gesamtbeurteilung_der_Residualklassifikation
    - R0-Residualklassifikation
    - R1-Residualklassifikation
    - R2-Gesamtbeurteilung
    - RX-Residualklassifikation
  - Gleason-Score
    - Gleason-Score_1
    - Gleason-Score_2
    - Gleason-Score_Summe
  - Histologie-Grading
    - G1-Histologie-Grading
    - G2-Histologie-Grading
    - G3-Histologie-Grading
    - O-Histologie-Grading
    - X-Histologie-Grading
  - Lymphknoten
    - Befallene_Lymphknoten

- DOKUR_FML2Datum_14070_13407
- DOKUR_FML2Kurz_14070_13408
- DOKUR_FML3Datum_14070_13409
- DOKUR_FML3Kurz_14070_13410
- DOKUR_GS1_14070_13926
  - Value14070_13926_2
- DOKUR_GS2_14070_13927
  - Value14070_13927_2
- DOKUR_GS3_14070_13928
- DOKUR_HisAbstRes_14070_13930
- DOKUR_HisCoZahl_14070_13929
- DOKUR_HisDat_14070_13931
- DOKUR_HisEins_14070_13946
- DOKUR_HisEins2_14070_13944
- DOKUR_HisEins3_14070_13945
- DOKUR_HisGra_14070_13932
- DOKUR_HisGrad_LH_14070_13933
- DOKUR_HisInvLym_14070_13934

Predicate	Object
hasConceptCodePrefix	DPKK
hasNiceName	Gleason-Score Summe
hasDataType	PosInteger
hasImport	Result_Of_Gleason-Score_Summe_Operation_IF1
hasImport	Result_Of_Gleason-Score_Summe_Operation_IF2
hasDescription	Summe von Gleason-Score 1 und 2

Predicate	Object
hasSourceTable	ErlangenSTGSoarianTable
hasSelectFilter	FORMID = 14070
hasSelectFilter	COMPONENTID = 13927
hasSelectFilter	ASSESSMENTSTATUS <> 4
hasSelectFilter	VALUE != ''
hasSelectFilter	VALUE IS NOT NULL

```
Gleason-Score_Summe : IF (NOTEXISTS Value14070_13928_2 Value14070_13926_2) (ADD Value14070_13926_2 Value14070_13927_2)
```

Figure 9. Screenshot of the QuickMapp tool. As indicated with the red notes, the tool shows the target ontology on the left and the source ontology on the right side. The statement browsers below the ontologies show the statements connected of the selected above concept, e.g. incoming mappings (1) or hasSelectFilter statements (2). Mappings can be created with a mapping expression editor in a bracketed prefix notation. The mapping shown is the same as in Fig. 4. Without the EMR-specific local naming, the statement would be: Gleason3: IF (NOTEXISTS Gleason3 Gleason1) (ADD Gleason1 Gleason2).

doi:10.1371/journal.pone.0116656.g009

layer from the database schema, i.e. that the user no longer has to deal with the database structures and how the data is stored. Whenever the source system changes, the source ontology has to be modified to reflect these changes. For database schema changes (e.g. table or column names), a small modification of the routine that generates the source ontology can be sufficient. If the contents of the source system change, e.g. due to the versioning of data elements or the creation of new assessment forms, only the new data items have to be added.

Implementation

The whole approach has been implemented as a set of Java tools called "OntoImportSuite" using NetBeans (http://netbeans.org) and the Apache Jena framework (https://jena.apache.org/). It comprises an ontology editor (OntoEdit) for editing of i2b2-specific target ontologies, a manual mapping application (QuickMapp, see Fig. 9) and an OWL-to-SQL processor (OntoExport).

Results

Creation of a Soarian EMR source ontology

We created a comprehensive source ontology for the complete Soarian EMR of the Erlangen University hospital, which comprises 28,840 classes (with 785 first-level classes denoting the

Table 1. Types and numbers of intermediate nodes that were used in complex mappings in the urology project.

Node type	Times used	Description
ADD	9	Adds the numeric values of both operands
EQUALS	2	Returns 'TRUE' if operand 1 = operand 2
GREATER	2	Returns 'TRUE' if operand 1 > operand 2
GREATERVT	32	Returns 'TRUE' if operand 1 > operand 2, operand 1 can be NULL
EXISTS	26	Returns 'TRUE' if operands 1 and 2 exist (with operand 1 = operand 2)
NOTEXISTS	8	Returns 'TRUE' if operand 1 does not exist and operand 2 exists (see Table G in S1 File for detailed explanation why operand 2 is necessary)
IF	64	Returns the value from operand 2 if operand 1 has the value 'TRUE'

doi:10.1371/journal.pone.0116656.t001

forms and 28,055 second-level classes denoting the EHR data elements) and 69,841 instances representing the values, including versioning. In addition, the ontology contains what we call "abstract concepts". These are ontology concepts that do not describe original data in the source system but rather information that was derived from these original data. For example, we created concepts that permit to discover patients for whom a given documentation form has been created, independently whether the form was completed or not. Thus we may e.g. detect cancer patients in our database, because a radiotherapy form exists.

Data integration in a cross-institutional setting

In a multicenter research scenario we faced the task to build a shared tissue sample database [34]. The target data set comprised elements such as: *clinical study participation, clinical and pathological TNM, time between diagnosis and surgery, Gleason scores* and many more. This dataset has been modeled as an OWL target ontology with 51 classes and 189 leafs using the OntoEdit tool.

We then mapped this target ontology to our EMR source ontology. In this case we identified 143 relevant data elements from 20 different forms such as anamnesis, tumor board report and therapy reports.

The mapping ontology for this scenario comprises 648 mappings including 50 complex mappings (see Table 1 for the use of complex operators). To identify patients with a Gleason Score greater two, we added eight custom string values to deal with such classifications. In the mentioned example the custom string value "2" has been used to export data only for patients with a Gleason Score greater two. Other operators listed in Table 1, such as GREATERVT, EXISTS or NOTEXISTS were required for either temporal selections of the appropriate first or last data value respectively for selecting patients with the correct type of cancer.

The named scenario is in continuous use since 2010. Thus, the SQL statements have been used to fill the research database of currently 529 patients and 21,878 observation facts. Changes in the EMR have been successfully addressed by updating the source and mapping ontologies.

In this scenario we also integrated source data of the clinical information system from another University hospital. In that case the EMR comprised effectively 3 fields for the Gleason score, namely Gleason 1, Gleason 2 and the summary Gleason. To perform this task we implemented an additional source and mapping ontologies but reused the same target ontology.

Discussion

Overcoming the challenges of accessing and processing heterogeneous EMR data

This paper presents a novel approach for the ontology-based integration of heterogeneous medical data between clinical and research databases. It makes heavy use of abstraction by shifting the database-centered, technical thinking based on tables, columns and rows to a focus on medical concepts and their relations. The structural view of the data source, the data target and the mapping undergo an explicit externalization within three ontology constructs.

Our approach does not eliminate the mapping effort itself, although we strive to make the mapping more sustainable. Relying on machine-processible Semantic Web standards, our proposed method enables the re-use of the captured mapping knowledge to support, for example, the automated SQL code generation for the physical ETL process from one system to another.

Our methodology could act as an extension for research databases by providing a universally machine-readable, semantic ETL framework with a fine granularity that reaches down to the level of versioned value sets in the source systems. We have shown that our approach is compatible with i2b2 and that it is capable of processing highly heterogeneous EMR assessment form data. Furthermore, this method provides solutions for the challenges that were originally mentioned in the introduction:

A) Our method supports the mapping of non-standardized data elements to standard terminologies. Custom data elements are typical features of modern EMR systems that support the definition of customized clinical documentation forms. For secondary use research projects it is vital to standardize these data, e.g. by mapping them to standard terminologies. In our approach, these mappings form part of the permanent and reusable mapping ontology. We currently map to custom domain ontologies (e.g. the one described in section 3.2), but mappings to standard terminologies are possible. Some nomenclatures, such as SNOMED-CT, post-coordinate medical concepts. This means that one concept is actually a composition of others [49]. We support the integration of such concepts by making use of complex one-to-many or many-to-one mappings, because they allow the arbitrary merging, splitting and logical linking of concepts. Thus mappings to ontologies such as the NCI Thesaurus [50] or SNOMED-CT are possible. Once more terminologies become available in OWL (see e.g. [51, 52]) it will be easier to store and maintain such mappings in this format.

B) Our system provides knowledge management functions for EMRs. By manually mapping semantically equal concepts from the source system to single concepts in the target ontology, the users of our system create a verified, machine-processible knowledge repository, which is similar to a medical data dictionary. Due to the manual mapping process and the use of intermediate nodes, it is possible to explicitly define the semantic relationships between similar data elements, whereas others (especially automated mapping methods) are limited to only describing the level of similarity (e.g. [53]).

Apart from the automated SQL code generation for the ETL process, the ontologies of our approach can be post-processed and queried for other purposes as well, e.g. to derive provenance information of data. The mapping ontology can be evaluated in terms of node types, performed transformations and filter mechanisms used, as shown in Table 1. In conventional ETL tools, such identification would be very difficult if not impossible.

The provenance information is useful for the maintenance of the source system. It can be evaluated in order to identify redundant data elements, which is typically the case if two or more source system concepts are mapped to a single concept in the target ontology. When creating new content in a source system, e.g. when a new EMR form has to be created, a quick look-up in the source ontology enables the identification of already existing data elements.

This avoids the accumulation of inconsistent concept naming and value sets because already existing data elements can be reused.

C) Our approach provides means for the semantic annotation of EMR systems. We recreate and preserve contextual medical relationships between data elements within the mapping ontologies. This comprises also medical knowledge that may be hidden within the EMR. An example has been given in Fig. 1. The mentioned pathology form contains the hidden medical knowledge that Gleason is a compound score with two components, which refer to the same date of biopsy. Our mapping ontology makes this explicit.

D) Our approach facilitates data integration between institutions. The target ontology may be shared between several institutions. Every institution can define its own source ontology and mappings to the target ontology. The generated SQL statements then perform the data extraction and processing. Although we cannot eliminate semantic gaps between source and target, we are able to model individual as well as reusable scenarios to deal with such gaps in a formalized and reproducible fashion.

Related research

Current state-of-the-art single source research platforms such as *Informatics For Integrating Biology And The Bedside* (i2b2) [46], the *Shared Health Research Network* (SHRINE) [26] or *Electronic Health Records For Clinical Research* (EHR4CR) [30] use a data warehouse approach. Such data warehouses are based on common information models and allow the storage of heterogeneous medical data. To transfer clinical data into a research data warehouse an ETL process is required to extract and transform data from a clinical source system and to load it. The usual approach comprises copying table structures from the clinical system to a staging area, transforming them to a given target structure with the help of a mapping or ETL tool and to finally load the source system contents into the data warehouse. The complete mapping process remains more or less hidden within the respective ETL tool. In contrast, we make both the structure of source and target system and the mapping explicit in reusable triple structures within the ontologies.

Some data warehouses permit, similar to our approach, the automated generation of SQL statements (e.g. [54–63]). Upon a first glance, our implementation shares several similarities with these tools. They all feature the abstract and often graphical modeling of ETL jobs, which are then automatically translated into SQL code or another representation that processes the data. The popular Talend Open Studio ETL software [56] for example generates Java code. Furthermore, they contain useful features such as error tracking and volume auditing. However, these modeled ETL jobs are specific to the respective ETL software and do not permit external processing. In comparison, our approach uses machine-processible ETL definitions that can be reused outside the ETL environment. The advantage is that thus we can e.g. support sustainable mapping to external terminologies (see chapter 4.1) as well as external statistics of the mapping effort and mapping performance.

Bache et al. [30] describe how they connect to different DWHs using an SQL-template-based query mechanism in order to achieve a mapping from their source system to the data model of the EHR4CR platform. The use of predefined SQL queries is similar to our approach. However, while Bache et al. use different static queries for different medical data categories, our approach permits the use of dynamic templates attached to each medical concept. We extend this feature down to the value level using unique *hasSelectFilter* properties for the templates.

The development of ETL jobs for heterogeneous data is a difficult task and some researchers aim to partially automate it. The research area of *schema matching* and *mapping* develops algorithms that try to find correspondences between two different database schemas [64, 65]. For example, Sun's MEDIATE [53], which also uses semantic networks to store mappings between semantically equivalent concepts in different databases, is such an automatic schema matcher. The system is also capable of automatically creating SQL code for data retrieval by including "database links" into the semantic network. It is worth noting that such matching methods cannot create complex mappings which support data transformations between multiple concepts. To our knowledge, no such implementation exists yet, and we believe it would be very challenging to develop one in the case of uncoded EHR data, due to its extreme heterogeneity.

Others propose the use of Semantic Web technologies [37] to ease the challenge of heterogeneous data integration. In most implementations the complete, originally relational research data is made available in RDF triples [27, 66–70]. In this context, tools such as D2RQ [71] or Quest [72] have been developed and have been used e.g. in [73]. Such on-the-fly conversions, however, do not ease the challenges of reusing the intrinsic EMR data and the semantic annotation of the EMR, because the generated RDF is almost an exact copy of the original database schema ([37], p.345). This means that that original data is only represented in a different syntax (triples), but with no semantic value added. To continue to work with such data, technologies such as SPARQL [47] would have to be used in the same way as SQL for relational databases, and a semantic integration would have to take place afterwards. We believe it is better to convert the metadata of the source system to RDF, separated from the facts data. This allows a flexible representation and modeling of local specialties, such as the data element versioning and can be used to provide a true semantic mapping between source and target systems.

Integration with conventional ETL environments

The generated SQL code automatically handles the extraction, transformation and loading of the mapped data elements into i2b2. ETL methodologies for data warehouses can be considerably complex (e.g. [74, 75]), and depending on the character of the data different tools are used. Our proposed method could complement conventional data warehousing setups by simplify the integration of highly heterogeneous medical data, such as EMR assessment form data. In such a case the generated SQL would become a parallel track in the transformation pipeline. With the generated SQL scripts integrated into commercial or free ETL solutions (e.g. [54–63]), the approach would also benefit from error tracking, volume auditing and other features.

Portability to other institutions and environments

The proposed semantic ETL method can be transferred to another environment or research institution, provided that this institution has access to the metadata of its EMR database:

1. As described in 2.4, a process is required that generates the source ontology for the source system. For EMRs with relational EAV-like databases, this can be achieved with SQL scripts.
2. The manual mapping process, which may be supported with the QuickMapp tool, must be performed to define mappings and conversions between source and target ontology items.
3. The OntoExport tool reads all information from the source, target and mapping ontology and automatically produces the SQL to extract and transform the required data items to the target system.

While our approach is generic and should work with any relational database system, our OntoExport tool currently generates SQL code for Oracle. By modifying the SQL code fragments in one of the ontologies it is possible to generate SQL code compatible with other SQL-based database systems (see Fig. 5, *OntoMappingSystem.owl*). Besides Oracle we have also tested Microsoft SQL-Server.

Limitations

The creation of some complex mappings is inconvenient in our approach. We discussed mappings between different surgical interventions and body parts for which these interventions could be applied. This would have resulted in 108 rather ineffective partial mappings, because we do not yet support mappings at hierarchical levels, e.g. between the class of all interventions and the class of all body parts.

Today, our supported target system is i2b2. Thus, the current implementation incorporates some i2b2-specific features related to the semantics of the target ontology and the internal data model of the facts data. While the generic and pragmatic i2b2 system offers extensive research capabilities, additional standardization would simplify the data export to other research platforms. It might even enable us to develop our approach towards a comprehensive semantic data integration software suite. Development towards the ISO/IEC 11179 MDR metadata repository standard [76], openEHR archetypes [77], HL7 RIM [78], ISO 13606 [79] or the CDISC standards [80] could be a future task. Even less complex standards such as SKOS [81] could be beneficial, as shown in [82, 83]. Complying with such standards would simplify interfacing with non-i2b2 systems and the adaption to other sites.

A current practical limitation of our system is the storage of ontological knowledge in local OWL files. Therefore target ontologies must be copied between institutions even if they are identical. Switching to a central triple store or a terminology server, such as LexEVS [84], would remedy this issue.

Outlook and Future Research

We have presented a novel approach for semantic ETL in single source projects. Future work should concentrate on standardizing the target ontology and internal data model as well as the integration of additional mappings towards standardized terminologies, such as the NCI Thesaurus or SNOMED CT. Additional research concerning the ontological modeling of advanced properties within assessment forms will be necessary, e.g. to enable the creation of hierarchical and other abstract relationships between different form elements.

Availability of the software (OntoImportSuite)

The software and source code (licensed under the GPL3) are available on GitHub (https://github.com/sebmate/OntoImportSuite). In addition we have supplied a demonstration subset of the diverse ontology contents, which is needed to replicate the methodology and the SQL script that was used to generate the source ontology for our Soarian EMR system as described in this paper. The installation requires an i2b2 1.6.x instance or database schema. Please note that the software is of prototypical character and provided "as is", without any warranties.

Supporting Information

S1 File. Appendix containing Tables A-J. Table A. Classes of the ontology MDR-System.owl. **Table B.** Instances of class MDR-DataType in MDR-System.owl. **Table C.** Datatype properties of the ontology in MDR-System.owl. **Table D.** Object properties of the ontology in MDR-System.owl. **Table E.** Class hierarchy of the ontology OntoMappingSystem.owl. **Table F.** Instances of the class ArithmethicOperation in the ontology OntoMappingSystem.owl. **Table G.**

Instances of the class RelationalOperator in the ontology OntoMappingSystem.owl. **Table H.** Instances of the class StringOperation in the ontology OntoMappingSystem.owl. **Table I.** Object properties of the ontology OntoMappingSystem.owl. **Table J.** Datatype properties of the ontology OntoMappingSystem.owl.
(DOCX)

Author Contributions

Conceived and designed the experiments: SM. Performed the experiments: SM FK MM. Analyzed the data: SM HUP DT FK MM TB TG. Wrote the paper: SM TB DT FK HUP TG. Contributed to the approach and implementation: TB FK MM TG. Provided the infrastructure and managed the project: TG TB HUP.

References

1. Kush RD, Alschuler L, Ruggeri R, Cassells S, Gupta N, et al. (2007) Implementing Single Source: The STARBRITE Proof-of-Concept Study. J Am Med Inform Assoc 14: 662–673. doi: 10.1197/jamia.M2157 PMID: 17600107
2. Jensen PB, Jensen LJ, Brunak S (2012) Mining Electronic Health Records: Towards Better Research Applications and Clinical Care. Nat Rev Genet 13: 395–405. doi: 10.1038/nrg3208 PMID: 22549152
3. Safran C, Bloomrosen M, Hammond WE, Labkoff S, Markel-Fox S, et al. (2007) Toward a National Framework for the Secondary Use of Health Data: an American Medical Informatics Association White Paper. J Am Med Inform Assoc 14: 1–9. doi: 10.1197/jamia.M2273 PMID: 17077452
4. Prokosch H-U, Ganslandt T (2009) Perspectives for Medical Informatics. Reusing the Electronic Medical Record for Clinical Research. Methods Inf Med 48: 38–44. doi: 10.3414/ME9132
5. Pearson JF, Brownstein CA, Brownstein JS (2011) Potential for Electronic Health Records and Online Social Networking to Redefine Medical Research. Clinical Chemistry 57: 196–204. doi: 10.1373/clinchem.2010.148668 PMID: 21159898
6. Fadly El A, Rance B, Lucas N, Mead CN, Chatellier G, et al. (2011) Integrating Clinical Research with the Healthcare Enterprise: From the RE-USE Project to the EHR4CR Platform. J Biomed Inform 44: S94–S102. doi: 10.1016/j.jbi.2011.07.007
7. Dugas M, Lange M, Müller-Tidow C, Kirchhof P, Prokosch H-U (2010) Routine Data From Hospital Information Systems Can Support Patient Recruitment for Clinical Studies. Clin Trials 7: 183–189. doi: 10.1177/1740774510363013 PMID: 20338903
8. Fadly El A, Lucas N, Rance B, Verplancke P, Lastic P-Y, et al. (2010) The REUSE Project: EHR as Single Datasource for Biomedical Research. Stud Health Technol Inform 160: 1324–1328.
9. Embi PJ, Jain A, Clark J, Bizjack S, Hornung R, et al. (2005) Effect of a Clinical Trial Alert System on Physician Participation in Trial Recruitment. Arch Intern Med 165: 2272–2277. doi: 10.1001/archinte.165.19.2272 PMID: 16246994
10. Köpcke F, Kraus S, Scholler A, Nau C, Schüttler J, et al. (2012) Secondary Use of Routinely Collected Patient Data in a Clinical Trial: an Evaluation of the Effects on Patient Recruitment and Data Acquisition. Int J Med Inform. doi: 10.1016/j.ijmedinf.2012.11.008 PMID: 23266063
11. Weiner MG, Lyman JA, Murphy SN, Weiner M (2007) Electronic Health Records: High-Quality Electronic Data for Higher-Quality Clinical Research. Inform Prim Care 15: 121–127. PMID: 17877874
12. Dean BB, Lam J, Natoli JL, Butler Q, Aguilar D, et al. (2009) Review: Use of Electronic Medical Records for Health Outcomes Research: a Literature Review. Med Care Res Rev 66: 611–638. doi: 10.1177/1077558709332440 PMID: 19279318
13. Brownstein JS, Murphy SN, Goldfine AB, Grant RW, Sordo M, et al. (2010) Rapid Identification of Myocardial Infarction Risk Associated with Diabetes Medications Using Electronic Medical Records. Diabetes Care 33: 526–531. doi: 10.2337/dc09-1506 PMID: 20009093
14. Breil B, Semjonow A, Müller-Tidow C, Fritz F, Dugas M (2011) HIS-Based Kaplan-Meier Plots—a Single Source Approach for Documenting and Reusing Routine Survival Information. BMC Med Inform Decis Mak 11: 11. doi: 10.1186/1472-6947-11-11 PMID: 21324182
15. McCarty CA, Chisholm RL, Chute CG, Kullo IJ, Jarvik GP, et al. (2011) The eMERGE Network: a Consortium of Biorepositories Linked to Electronic Medical Records Data for Conducting Genomic Studies. BMC Med Genomics 4: 13. doi: 10.1186/1755-8794-4-13 PMID: 21269473

16. Morrison Z, Fernando B, Kalra D, Cresswell KM, Sheikh A (2013) National Evaluation of the Benefits and Risks of Greater Structuring and Coding of the Electronic Health Record: Exploratory Qualitative Investigation. J Am Med Inform Assoc. PMID: 24186957

17. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, et al. (2010) Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, Component Evaluation and Applications. J Am Med Inform Assoc 17: 507–513. doi: 10.1136/jamia.2009.001560 PMID: 20819853

18. O'Malley KJ, Cook KF, Price MD, Wildes KR, Hurdle JF, et al. (2005) Measuring Diagnoses: ICD Code Accuracy. Health Serv Res 40: 1620–1639. doi: 10.1111/j.1475-6773.2005.00444.x PMID: 16178999

19. Weiner MG, Embi PJ (2009) Toward Reuse of Clinical Data for Research and Quality Improvement: the End of the Beginning? Ann Intern Med 151: 359–360. doi: 10.7326/0003-4819-151-5-200909010-00141 PMID: 19638404

20. Papatheodorou I, Crichton C, Morris L, Maccallum P, METABRIC Group, et al. (2009) A Metadata Approach for Clinical Data Management in Translational Genomics Studies in Breast Cancer. BMC Med Genomics 2: 66. doi: 10.1186/1755-8794-2-66 PMID: 19948017

21. International Health Terminology Standards Development Organisation (n.d.) SNOMED CT. Available: http://ihtsdo.org/snomed-ct/. Accessed 2014 Oct 25.

22. Cimino JJ (1998) Desiderata for Controlled Medical Vocabularies in the Twenty-First Century. Methods Inf Med 37: 394–403. PMID: 9865037

23. Lee MK, Park H-A, Min YH, Kim Y, Min HK, et al. (2010) Evaluation of the Clinical Data Dictionary (CiDD). Healthc Inform Res 16: 82–88. doi: 10.4258/hir.2010.16.2.82 PMID: 21818428

24. Michel A, Prokosch H-U, Dudeck J (1989) Concepts for a Medical Data Dictionary. MEDINFO 89: 805–808. PMID: 10384592

25. Sobin LH, Compton CC (2010) TNM Seventh Edition: What's New, What's Changed: Communication From the International Union Against Cancer and the American Joint Committee on Cancer. Cancer 116: 5336–5339. doi: 10.1002/cncr.25537 PMID: 20665503

26. McMurry AJ, Murphy SN, MacFadden D, Weber GM, Simons WW, et al. (2013) SHRINE: Enabling Nationally Scalable Multi-Site Disease Studies. PLoS ONE 8: e55811. doi: 10.1371/journal.pone.0055811

27. Hussain S, Ouagne D, Sadou E, Dart T, Jaulent M-C, et al. (2012) EHR4CR: A Semantic Web Based Interoperability Approach for Reusing Electronic Healthcare Records in Protocol Feasibility Studies. Semantic Web Applications and Tools for Life Sciences. Available: http://ceur-ws.org/Vol-952/paper_31.pdf.

28. Bleich HL, Slack WV (1992) Design of a Hospital Information System: a Comparison Between Integrated and Interfaced Systems. MD Comput 92: 293–296.

29. Ohmann C, Kuchinke W (2009) Future Developments of Medical Informatics From the Viewpoint of Networked Clinical Research. Methods Inf Med 48: 45–54. PMID: 19151883

30. Bache R, Miles S, Taweel A (2013) An Adaptable Architecture for Patient Cohort Identification From Diverse Data Sources. J Am Med Inform Assoc. doi: 10.1136/amiajnl-2013-001858 PMID: 24064442

31. Bürkle T, Schweiger RK, Altmann U, Holena M, Blobel B, et al. (1999) Transferring Data From One EPR to Another: Content-Syntax-Semantic. Methods Inf Med 38: 321–325. doi: 10.1267/METH99040321 PMID: 10805022

32. Haux R, Seggewies C, Baldauf-Sobez W, Kullmann P, Reichert H, et al. (2003) Soarian (TM)—Workflow Management Applied for Health Care Vol. 42. pp. 25–36.

33. Prokosch H-U, Ries M, Beyer A, Schwenk M, Seggewies C, et al. (2011) IT Infrastructure Components to Support Clinical Care and Translational Research Projects in a Comprehensive Cancer Center. Stud Health Technol Inform 169: 892–896. PMID: 21893875

34. Mate S, Bürkle T, Köpcke F, Breil B, Wullich B, et al. (2011) Populating the i2b2 Database with Heterogeneous EMR Data: a Semantic Network Approach. Stud Health Technol Inform 169: 502–506. PMID: 21893800

35. Prokosch H-U, Mate S, Christoph J, Beck A, Köpcke F, et al. (2012) Designing and Implementing a Biobanking IT Framework for Multiple Research Scenarios. Stud Health Technol Inform 180: 559–563. PMID: 22874253

36. Berners-Lee T, Hendler JA, Lassila O (2001) The Semantic Web. Scientific American 284: 28–37. doi: 10.1038/scientificamerican0501-34

37. Hebeler J, Fisher M, Blace R, Perez-Lopez A (2009) Semantic Web Programming. Wiley. 651 pp.

38. Klyne G, Carroll JJ, McBride B (2004) Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation.

39. Brickley D, Guha RV, McBride B (2004) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation.

40. Smith MK, Welty C, McGuinness DL (2004) OWL Web Ontology Language Guide. W3C Recommendation.

41. Nadkarni PM (2012) Using Electronic Medical Records Systems for Clinical Research: Benefits and Challenges. Presentation at the University of Iowa.

42. Nadkarni PM, Brandt CA (1998) Data extraction and ad hoc query of an entity-attribute-value database. AMIA 5: 511–527.PMID: 9824799

43. Nadkarni PM, Marenco L, Chen R, Skoufos E, Shepherd GM, et al. (1999) Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. J Am Med Inform Assoc 6: 478–493. doi: 10.1136/jamia.1999.0060478 PMID: 10579606

44. Lin C-P, Black RA, Laplante J, Keppel GA, Tuzzio L, et al. (2010) Facilitating Health Data Sharing Across Diverse Practices and Communities. AMIA Summits Transl Sci Proc 2010: 16–20. PMID: 21347138

45. Nadkarni PM, Brandt CA (2006) The Common Data Elements for Cancer Research: Remarks on Functions and Structure. Methods Inf Med 45: 594–601. PMID: 17149500

46. Murphy SN, Weber GM, Mendis ME, Gainer VS, Chueh HC, et al. (2010) Serving the Enterprise and Beyond with Informatics for Integrating Biology and the Bedside (i2b2). J Am Med Inform Assoc 17: 124–130. doi: 10.1136/jamia.2009.000893 PMID: 20190053

47. Prud'hommeaux E, Seaborne A (2009) SPARQL Query Language for RDF. W3C Recommendation.

48. Krogh B, Weisberg A, Bested M (2011) DBLint: A Tool for Automated Analysis of Database Design.

49. Pathak J, Wang J, Kashyap S, Basford M, Li R, et al. (2011) Mapping Clinical Phenotype Data Elements to Standardized Metadata Repositories and Controlled Terminologies: the eMERGE Network Experience. J Am Med Inform Assoc 18: 376–386. doi: 10.1136/amiajnl-2010-000061 PMID: 21597104

50. Sioutos N, de Coronado S, Haber MW, Hartel FW, Shaiu W-L, et al. (2007) NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. J Biomed Inform 40: 30–43. doi: 10.1016/j.jbi.2006.02.013 PMID: 16697710

51. Noy NF, de Coronado S, Solbrig HR, Fragoso G, Hartel FW, et al. (2008) Representing the NCI Thesaurus in OWL DL: Modeling Tools Help Modeling Languages. Appl Ontol 3: 173–190. PMID: 19789731

52. Musen MA, Shah NH, Noy NF, Dai B, Dorf M, et al. (2008) BioPortal: Ontologies and Data Resources with the Click of a Mouse. AMIA Annu Symp Proc: 1223–1224. PMID: 18999306

53. Sun Y (2004) Methods for Automated Concept Mapping Between Medical Databases. J Biomed Inform 37: 162–178. doi: 10.1016/j.jbi.2004.03.003 PMID: 15196481

54. Pentaho Corporation (2014) Kettle Data Integration. Available:http://www.pentaho.com/. Accessed 2014 Jul 29.

55. Javlin AS (2014) CloverETL Rapid Data Integration. Available: http://www.cloveretl.com/. Accessed 2014 Jul 29.

56. Talend Inc (2014) Talend Open Studio. Available: http://www.talend.com. Accessed 2014 Jul 29.

57. IBM Corporation (2014) InfoSphere Information Server. Available: http://www-01.ibm.com/software/data/integration/info_server/. Accessed 2014 Jul 29.

58. Informatica Corporation (2014) PowerCenter Big Data Edition. Available: http://www.informatica.com/us/products/big-data/powercenter-big-data-edition. Accessed 2014 Jul 29.

59. SAP Aktiengesellschaft (2014) Business Objects Data Integrator. Available: http://www.sap.com/pc/tech/enterprise-information-management/software/data-integrator/index.html. Accessed 2014 Jul 29.

60. Oracle International Corporation (2014) Data Integrator. Available: http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html. Accessed 2014 Jul 29.

61. Oracle International Corporation (2014) Warehouse Builder. Available: http://www.oracle.com/technetwork/developer-tools/warehouse/overview/introduction/index.html. Accessed 2014 Jul 29.

62. Microsoft Corporation (2014) SQL Server Integration Services. Available: http://msdn.microsoft.com/library/ms141026.aspx. Accessed 2014 Jul 29.

63. IBM Corporation (2014) Cognos. Available: http://www-01.ibm.com/software/analytics/cognos/. Accessed 2014 Jul 29.

64. Rahm E, Bernstein PA (2001) A Survey of Approaches to Automatic Schema Matching. The VLDB Journal 10: 334–350. doi: 10.1007/s007780100057

65. Massmann S, Raunich S, Aumüller D, Arnold P, Rahm E (2011) Evolution of the COMA Match System. Ontology Matching: 49.

66. Farley T, Kiefer J, Lee P, Hoff Von D, Trent JM, et al. (2012) The BioIntelligence Framework: a New Computational Platform for Biomedical Knowledge Computing. J Am Med Inform Assoc 20: 128–133. doi: 10.1136/amiajnl-2011-000646 PMID: 22859646

67. Cheung K-H, Frost HR, Marshall MS, Prud'hommeaux E, Samwald M, et al. (2009) A Journey to Semantic Web Query Federation in the Life Sciences. BMC Bioinformatics 10 Suppl 10: S10–. doi: 10.1186/1471-2105-10-S10-S10 PMID: 19796394

68. Williams AJ, Harland L, Groth P, Pettifer S, Chichester C, et al. (2012) Open PHACTS: Semantic Interoperability for Drug Discovery. Drug Discovery Today 17: 1188–1198. doi: 10.1016/j.drudis.2012.05.016 PMID: 22683805

69. Harland L (2012) Open PHACTS: A Semantic Knowledge Infrastructure for Public and Commercial Drug Discovery Research. Knowledge Engineering and Knowledge Management. Springer. pp. 1–7.

70. Tao C, Jiang G, Oniki TA, Freimuth RR, Zhu Q, et al. (2012) A Semantic-Web Oriented Representation of the Clinical Element Model for Secondary Use of Electronic Health Records Data. J Am Med Inform Assoc 20: 554–562. doi: 10.1136/amiajnl-2012-001326 PMID: 23268487

71. Bizer C, Seaborne A (2004) D2RQ—Treating Non-RDF Databases as Virtual RDF Graphs. Proceedings of the 3rd international semantic web conference (ISWC2004) 2004.

72. Rodrigues-Muro M, Calvanese D (2012) Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access. OWLED 2012.

73. Assélé Kama A, Primadhanty A, Choquet R, Teodoro D, Enders F, et al. (2012) Data Definition Ontology for Clinical Data Integration and Querying. Stud Health Technol Inform 180: 38–42. PMID: 22874148

74. Anitha J, Babu M (2014) ETL Work Flow for Extract Transform Loading. International Journal of Computer Science and Mobile Computing 3: 610–617.

75. Kimball R (1998) The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses. Wiley.

76. Ngouongo SM, Stausberg J (2011) Integration of Classifications and Terminologies in Metadata Registries Based on ISO/IEC 11179. Stud Health Technol Inform 169: 744–748. PMID: 21893846

77. Garde S, Hovenga E, Buck J, Knaup-Gregori P (2007) Expressing Clinical Data Sets with openEHR Archetypes: a Solid Basis for Ubiquitous Computing. Int J Med Inform 76 Suppl 3: S334–S341.

78. Eggebraaten TJ, Tenner JW, Dubbels JC (2007) A Health-Care Data Model Based on the HL7 Reference Information Model. IBM Systems Journal 46: 5–18. doi: 10.1147/sj.461.0005

79. Veseli H, Kopanitsa G, Demski H (2012) Standardized EHR Interoperability—Preliminary Results of a German Pilot Project using the Archetype Methodology. Stud Health Technol Inform 180: 646–650. PMID: 22874271

80. Lastic P-Y (2012) CDISC SHARE: A BRIDG-based Metadata Repository Environment pp. 1–16. Available: http://www.tmf-ev.de/News/articleType/ArticleView/articleId/1228.aspx.

81. W3C (2009) SKOS Simple Knowledge Organization System Primer. Isaac A, Summers E, editors W3C Recommendation. Available: http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/.

82. Löbe M, Stäubert S (2010) Bringing Semantics to the i2b2 Framework. GI Jahrestagung (2) 2010: 734–738.

83. Majeed RW, Röhrig R (2013) Using the i2b2-Web Frontend to Query Custom Medical Data Repositories: Emulation of a Virtual i2b2 Server. GMDS 2013 58 Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie eV (GMDS): 1–2.

84. Ethier J-F, Dameron O, Curcin V, McGilchrist MM, Verheij RA, et al. (2013) A Unified Structural/Terminological Interoperability Framework Based on LexEVS: Application to TRANSFoRm. J Am Med Inform Assoc 20: 986–994. doi: 10.1136/amiajnl-2012-001312 PMID: 23571850