ICT-2010-270253

INTEGRATE

Driving excellence in Integrative Cancer Research through Innovative Biomedical Infrastructures

STREP
Contract Nr: 270253

Deliverable: D3.3 Solution providing uniform access to relevant external sources

Due date of deliverable: (09-30-2012)
Actual submission date: (10-12-2012)

Start date of Project: 01 February 2011 Duration: 36 months

Responsible WP: FORTH

Revision: accepted

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Haridimos Kondylakis	FORTH	kondylak@ics.forth.gr
Manolis Tsiknakis	FORTH	tsiknaki@ics.forth.gr
David Perez-Rey	UPM	dperez@infomed.dia.fi.upm.es
Juan Manuel Moratilla	UPM	jmmoratilla@infomed.dia.fi.upm.es
Santiago Aso Lete	UPM	saso@infomed.dia.fi.upm.es
Jasper van Leeuwen	PHILIPS	jasper.van.leeuwen@philips.com
George Manikis	FORTH	gmanikis@ics.forth.gr

0.2 Documents history

Document version #	Date	Change
V0.1	08/01/2012	Starting version, template
V0.2	09/13/2012	Definition of ToC
V0.3	09/30/2012	First complete draft
V0.4	09/30/2012	Integrated version (send to WP members)
V0.5	10/05/2012	Updated version (send PCP)
V0.6	10/05/2012	Updated version (send to project internal reviewers)
Sign off	12/05/2012	Signed off version (for approval to PMT members)
V1.0	12/05/2012	Approved Version to be submitted to EU

0.3 Document data

Keywords	Uniform Access, External Data Sources
Editor Address data	Name: Haridimos Kondylakis Partner: FORTH Address: Vassilika Vouton P.O Box 1385 GR-71110 Heraklion, Crete, Greece Phone: +302810391449 Fax: +30 2810 391428 E-mail: kondylak@ics.forth.gr
Delivery date	10-12-2012

0.4 Distribution list

Date	Issue	E-mailer

10-15-2012	V1.0	Robert.BEGIER@ec.europa.eu
		Agnes.MAROFKA@ec.europa.eu
		Gisele.Roesems@ec.europa.eu
		fp7-integrate@listas.fi.upm.es

Table of Contents

1 Introduction

The subject of this deliverable is the design and realization of a solution providing access to external sources that will be required by the INTEGRATE project. The next section provides more details about the context in which the work was carried out. It is followed by three sections that explain the objective of this deliverable, the intended audience, and how the remainder of the report is structured.

1.1 Context

In order to make this document self-contained the following subsections provide the minimal background that is needed to understand the subsequent chapters. Figure 1 shows the high level view of the INTEGRATE architecture as proposed in D2.4 [1].

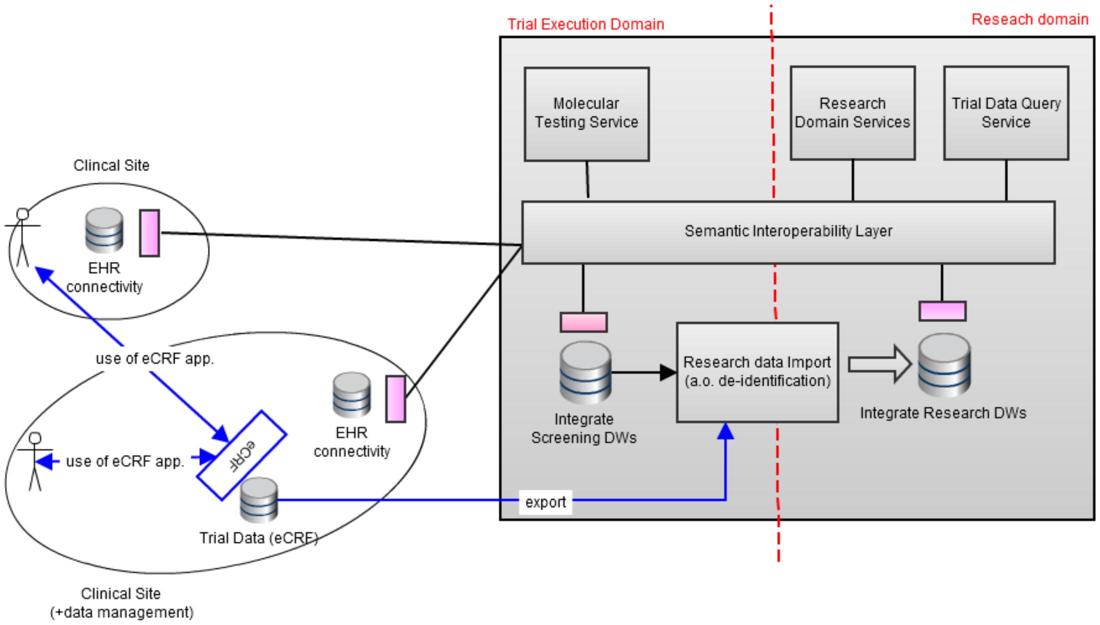

Figure 1. High-level view of the INTEGRATE architecture

In the architecture we can see that various clinical sites contribute data to the different data warehouses. Moreover, there is a separation between research domain and trial execution domain guaranteed by the INTEGRATE security framework. Data is allowed to be imported from the trial conduct domain to the research domain only through an export/import service that enforces de-identification and pseudonymization of data.

The semantic interoperability layer abstracts the different data sources, presenting a common information model to the upper application layers. These data sources include the central INTEGRATE data warehouses and the distributed EHR data exports at the participating trial sites.

The following types of data sources are important for the INTEGRATE project:

- **Electronic Health Record Systems**: These, amongst others, typically store the data gathered using the Clinical Report Forms (CRF).

- **Clinical Trials Repositories**: Repositories like these provide data on clinical trial. More specifically CRFs of patients participating in clinical trials are stored here and clinical trial designs as well.
- **Synapse Commons Repositories**: This repository provides raw data and corresponding phenotypic information for publicly available genomic data sets.

To access the relevant data sources the INTEGRATE project needs to provide several functions:

- Provide a uniform solution to load external data to the warehouse
- Use a common model to query data when those data have been transferred to the warehouse.
- Enforce the data source access policy, and audit access to data sources. For clinical data there are strict legal and ethical requirements that need to be adhered to.

1.2 Objective

This deliverable describes the design of a solution for providing uniform access to external data sources for the INTEGRATE project. More specifically, this report has the following objectives:

- To document the design decisions made, together with their rationale.
- To provide a basic understanding of the implementation of the data access services.
- To illustrate the possibilities and limitations of providing homogeneous access to heterogeneous data sources.

1.3 Structure of this deliverable

The remainder of this deliverable is structured as follows. Chapter 2 describes the external data sources as they were identified by INTEGRATE users. We describe the information they include alongside the current methods of exposing data. Next, chapter 3 describes the technologies that we have chosen for the realisation of the data access services in more detail. It provides the required background information and describes the design of the implementation of the data access services for the aforementioned data sources. Finally, chapter 4 concludes by summarising the achievements and discussing outstanding issues.

2 External Data Sources & Requirements

This chapter describes the external data sources that are of interest to the INTEGRATE project, and describes the means of accessing these resources.

2.1 Synapse Commons Repository

2.1.1 Description of the repository

The “Synapse Commons Repository” [2] is a collaborative computer space and part of the “Synapse” computational platform that allows scientists to share and analyse data together. The platform consists of a web portal, web services, and integrations with data analysis tools and is organized around novel “Analysis Communities” that scientists can create or join. Through this collaborative platform, access to raw data and corresponding phenotypic information for publicly available genomic data sets is available for use, facilitating cooperative compilation, comparison and evaluation of network models of disease under a unique framework. The “Synapse platform” is under the umbrella of a non-profit biomedical research organization named “Sage Bionetworks” [3], created to revolutionize how researchers approach the complexity of human biological information and the treatment of disease.

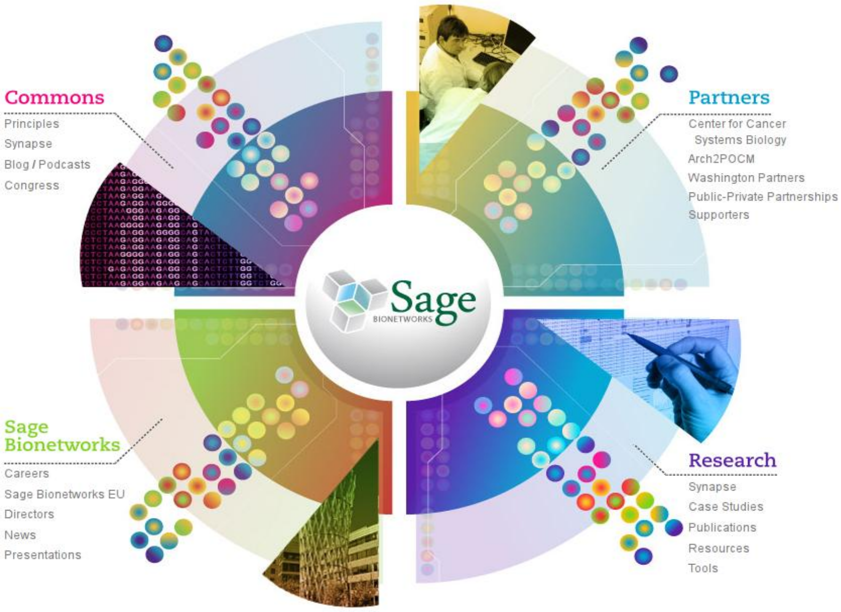

Figure 2. Sage platform

Among other currently running projects of the “Sage Bionetworks”, DREAM Breast Cancer Prognosis Challenge [4] focuses on the accuracy assessment of computational models designed to predict breast cancer survival based on clinical information about the patient's tumor as well as genome-wide molecular profiling data including gene expression and copy number profiles. A common dataset is provided by the “Synapse Commons Repository” to all participants, with a validation dataset held out for model evaluation. A novel dataset will be generated at the end of the challenge and used to provide a final, unbiased score for each model. Briefly described, the provided data comprises information of both survival and multi-modal feature data. Information about the survivability of each patient is given by the time from diagnosis to last follow up and whether the patient was alive at last follow up time. On the other hand, feature data consist of a large pool of gene expression, single nucleotide polymorphism (SNP), and clinical covariates for describing the tumor histology, size and cellularity, the received treatment, etc.

The scope and the goals of the “DREAM” challenge are closely related to the scientific questions that need to be answered from the INTEGRATE project and precisely from the predictive model that will be implemented for the needs of WP5. Therefore, the breast cancer dataset, available for the purposes of the “DREAM” challenge, seems to be a valuable additive information for designing, building and evaluating the INTEGRATE predictive model. Contributed data can be provided through public repositories like NCBI GEO[1], ArrayExpress[2] from EBI, and TCGA[3], as well as any source that can be accessed through the web, including a URL made available through an individual lab, a core facility, or a cloud based storage site like an Amazon S3 bucket.

2.1.2 Possible Methods of Accessing data

The synapse commons repository offers two methods of accessing data.

- **File Download:** Users can select the data and model of interest and then download them as a file into their local computer. These files can be parsed by an Extract-Transform-Load (ETL) tool and then be loaded to a local database, or used “as is” by the tools required for processing them.

- **APIs:** The repository provides the necessary APIs to load the datasets directly into the R Client application. It allows direct programmatic interactions with the Synapse repository.

2.2 Trial Management Databases

This section describes how the data originally residing in the trial management databases ends up being exposed in the INTEGRATE platform. The clinical data that is being collected in a clinical trial does only get analysed at the end of a clinical trial (or at very specific time points during a clinical trial). This implies that there is no necessity for a real-time interaction with the databases in which the clinical trial data is being collected. In INTEGRATE, the clinical trial data is ETL-ed to the INTEGRATE platform in batch mode.

2.2.1 Possible Methods of Accessing data

As described in *Deliverable 4.1 Specification of the model, data and annotation repositories* [5], the clinical trial data supplied by the clinical partner for development purposes is stored in Oracle Clinical.

[1] http://www.ncbi.nlm.nih.gov/geo/

[2] http://www.ebi.ac.uk/arrayexpress/

[3] http://cancergenome.nih.gov/

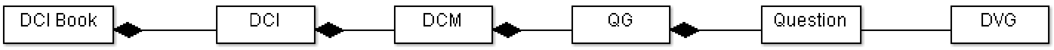

Figure 3 - Oracle Clinical structure

Figure 3 shows an overview of the structure of the Oracle Clinical database.

- *Question* represents a (particular) question on a CRF and its answer. When the answer can be populated from a set of possible answers (e.g. a code list), the answers are store in *DVG*, a Discrete Value Group.

- Questions are grouped in *QG* – Question group – which can be used to group (medically) related questions (which can be handy in order to reuse groups of questions of different CRFs).

- The question groups are used in *DCM* – Data Collection Modules – which represent (the sections of) the CRF screens that are used to collect the data and these sections should be answered during a single clinical visit.

- *DCI* – Data Collection Instrument – corresponds to a CRF. Typically, one DCI corresponds to one DCM, but the DCI construct allows for CRFs that collect data during multiple visits.

- Finally, a DCI Book specifies the order of the DCI’s.

```
<ClinicalData StudyOID="P2006-101" MetadataVersionOID="101.01">
   <SubjectData SubjectKey="1000" TransactionType="Insert">
      <StudyEventData StudyEventOID="Screen">
         <FormData FormOID="DEMOG">
            <ItemGroupData ItemGroupOID="DM">
                <ItemDataString ItemOID="USUBJID">101-001-
001</ItemDataString>
                <ItemDataString ItemOID="SEX">F</ItemDataString>
            </ItemGroupData>
        </FormData>
        <FormData FormOID="LABDATA">
            <ItemGroupData ItemGroupOID="LB">
                <ItemDataDatetime ItemOID="LBDTC">2006-07-
14T14:48</ItemDataDatetime>
                <ItemDataString
ItemOID="LBTESTCD">ALT</ItemDataString>
                <ItemDataString ItemOID="LBORRES">245</ItemDataString>
            </ItemGroupData>
        </FormData>
      </StudyEventData>
  </SubjectData>
</ClinicalData>
```

Table 1 - CDISC ODM excerpt

The data has can be exported by a clinical partner in Microsoft Excel format and follows the clinical database design. The exported data contains a sheet per DCI. The aggregated sub-tables are joined to provide the rows for the excel sheets, resulting in a row each time data is entered (a question is answered) on a Clinical Report Form.

In future, CDISC ODM (also described in *Deliverable 4.1 Specification of the model, data and annotation repositories* [5]) will be used to export clinical trial data. It provides a format for representing the study metadata, study data, and administrative data associated with a clinical trial and is designed to facilitate the archive and interchange of metadata and data for clinical research. The clinical data is collected per "study event" which is typically a patient visit. During this visit, the required forms are filled in and the data recorded on these forms is captured

CDISC ODM is an XML[4] specification as can be seen in Table 1, which shows an example from the CDISC documentation. The XML format offers an easy format for use in the ETL process of the INTEGRATE platform.

2.2.2 External data sources for trial criteria

An important objective of the INTEGRATE project is to build tools that support the efficient execution of post-genomic multi-centric clinical trials in breast cancer, which includes the automatic assessment of the eligibility of patients for available trials. Eligibility criteria describe characteristics of the patient population that need to be matched against the data items that are known for an individual patient. This process would be facilitated by the ability to identify semantic entities that sufficiently describe the meaning of the criteria and by establishing links to the relevant available data. Building these links (also known as mappings) is a partially manual process and it would be beneficial to be able to reuse them, whenever possible, across trials and systems.

Within INTEGRATE, we have analysed a large amount of clinical trials to identify subsets of widely used medical ontologies that sufficiently cover the content of the eligibility criteria of trials in the clinical domain of interest. Selecting only subsets of the ontologies facilitates the linkage of eligibility criteria to actual patient record, as defining mapping (or other processing steps) for entire ontologies is not feasible due to their sizes.

For this work, we have acquired a large selection of clinical trials from clinicaltrials.gov (4232 breast cancer trials, 6691 cancer trials other than breast cancer and 12255 trials on heart and blood diseases). Clinicaltrials.gov provides a simple RESTful API to retrieve information about clinical trials in XML (containing among others the eligibility criteria). Unfortunately, the eligibility criteria are specified in plain text. Therefore, we have used the bioportal annotator[5] to annotate the eligibility criteria with the relevant medical ontologies (mainly SNOMED, MedDRA and LOINC).

[4] http://www.w3.org/TR/REC-xml/

[5] http://bioportal.bioontology.org/annotator

From the results of the analysis, we can conclude that the reuse of concepts across trials is very significant, with a relative small number of concepts occurring in many trials. That allows us to prioritize concepts in the implementation of mappings.

2.3 Electronic Health Records

EHRs are defined as “a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting. Included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports” [6]. Some of the basic benefits associated with EHRs include being able to easily access computerized records and the elimination of poor penmanship, which has historically plagued the medical chart [7]. EHR systems can include many potential capabilities, but three particular functionalities hold great promise in improving the quality of care and reducing costs at the health care system level: clinical decision support (CDS) tools, computerized physician order entry (CPOE) systems, and health information exchange (HIE). These and other EHR capabilities are requirements of the “meaningful use” criteria set forth in the HITECH Act of 2009 [8].

Researchers have examined the benefits of EHRs by considering clinical, organizational, and societal outcomes. Clinical outcomes include improvements in the quality of care, a reduction in medical errors, and other improvements in patient-level measures that describe the appropriateness of care. Organizational outcomes, on the other hand, have included such items as financial and operational performance, as well as satisfaction among patients and clinicians who use EHRs. Lastly, societal outcomes include being better able to conduct research and achieving improved population health.

A large problem in both basic and clinical research is the lack of sufficient data, while the large amounts of patient data collected in clinical care in the EHR systems are seldom properly accessible for secondary use in research.

2.3.1 Possible Methods of Accessing data

Due to heterogeneity in data and communication standards in various eHealth systems, many aspects should be taken into consideration such as consistent nomenclature, data access, data transmission, data formats and storage. There are three main standards bodies currently active in international standards directly related to the EHR. These are ISO, the European Committee for Standardization (Comité Européen de Normalisation – CEN), and HL7. Within the United States there are many other standards development organizations that are involved in the development of EHR-related standards, most notably ASTM and the Object Management Group (OMG) Health Domain Task Force (HDTF). The following presented standards and profile aspects are the most widely used in integrating EHR data:

- **HL7 Standards:** As soon as the terms (“words”) are found to describe medical findings, a standard for how to combine these words into “sentences” is needed. These aspects are covered by Healthcare Information Technology (HIT) messaging standards. HL7 is the most commonly used HIT messaging standard worldwide. It has the aim to support hospital workflows by enabling different systems to communicate with each other. Early HL7v2 messages used a textual syntax whereas HL7v3 messages use an XML syntax.

- **The ASTM Continuity of Care Record (CCR)** standard[6] is a core data set of the most relevant administrative, demographic, and clinical information facts about a patient's healthcare, covering one or more healthcare encounters. It provides a means for one healthcare practitioner, system, or setting to aggregate all of the pertinent data about a patient and forward it to another practitioner, system, or setting to support the continuity of care. The primary use case for the CCR is to provide a snapshot in time containing the pertinent clinical, demographic, and administrative data for a specific patient. The Continuity of Care Document (CCD) is an HL7 CDA implementation of the CCR.

- **IHE Profiles:** Integrating the Healthcare Enterprise (IHE) is an international voluntary collaboration of vendors, healthcare providers, regulatory agencies, and independent experts working on improving medical data interoperability in a number of subject areas (domains) [9]. IHE Profiles organize and leverage the integration capabilities that can be achieved by coordinated implementation of communication standards, such as DICOM, HL7, W3C and security standards. They provide precise definitions of how standards can be implemented to meet specific clinical needs.

[6] http://www.astm.org/Standards/E2369.htm

3 Uniform Access Solution

3.1 Approaches for uniform access

Two different approaches exist for providing uniform access to heterogeneous data sources: data transformation and query translation. Using *data transformation*, data is taken from the original data sources, and converted and stored in a database specific to the data access service. This may involve mapping the data to a unified and normalised schema. Furthermore, all data can be stored in the same type of database, e.g. a relational database, irrespective of the type of data source that the data is coming from. In contrast, in the *query translation* approach the data remains in the original databases. Here, queries are translated instead. Both approaches have their advantages and drawbacks, which are summarised in Table 2. For the query translation approach, it is assumed that the data access services need to support all syntactically valid queries, irrespective of the query limitations of the underlying data source.

Data Transformation	**Query Translation**
− Large storage requirements at the data access service. Data from the original data sources is duplicated.	+ Minimal storage requirements at the data access service. For handling certain queries, it may be necessary to temporarily store data, but this is a subset of the data, and only has to be stored for the duration of the query
− It is difficult to keep the data synchronised. If the original data source cannot provide notifications when data has changed, the entire contents of the data need to be periodically converted, which is an expensive operation, and data will be outdated.	+ Retrieved data is always up to date.
+ Queries can be handled efficiently and easily.	− Translating queries can be a complex and expensive operation. If data sources have limited query capabilities, certain queries cannot be handled efficiently. Clients may experience a high latency, and the underlying data source and their network connection may experience a high load.
+ Answers do not depend on external sources availability	- Answers depend on external sources availability/policy changes

Table 2. A comparison of the Data Translation and Query Translation approach.

Based on the strengths and weaknesses given in Table 2, and the idiosyncrasies of the INTEGRATE project we have chosen to adopt a data translation approach for integrating external sources to the project. Handling only queries without having consistent access to the real data sources, proved to be a really difficult problem in pasts projects, whereas the imposed overhead in time make it prohibiting in our case.

3.1.1 General description of an ETL process

Intuitively, an ETL process can be thought of as a directed acyclic graph, as shown in Figure 1, with activities and record sets being the nodes of the graph and input-output relationships between nodes being the edges of the graph. Observe in the example that the sources can be HL7 messages or CDAs that must be propagated to the data warehouse fact tables on the right of the figure. The whole process of populating the fact tables is facilitated by a workflow of activities that perform all the appropriate filtering, intermediate data staging, transformations and loadings.

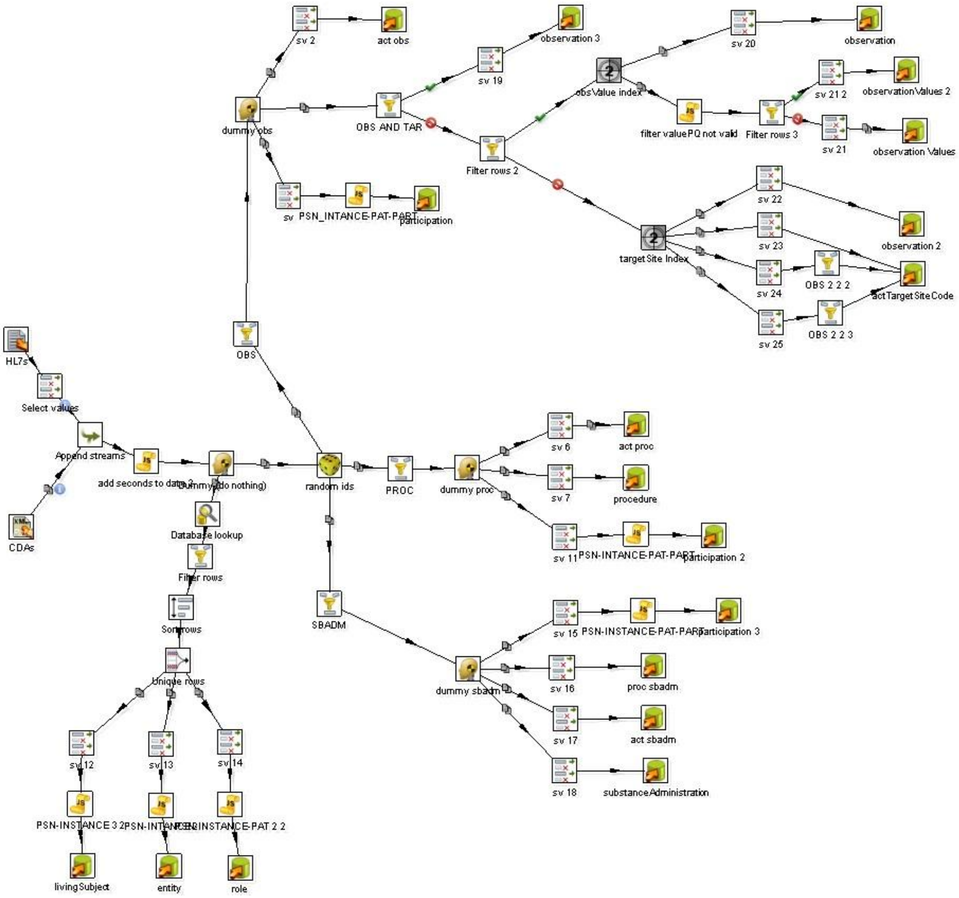

Figure 4. Example ETL process

The three main steps that should be followed in such a process are the following:

Extraction: The extraction step is conceptually the simplest task of all, with the goal of identifying the correct subset of source data that has to be submitted to the ETL workflow for further processing. As with the rest of the ETL process, extraction also takes place at idle times of the source system - typically at night. Practically, the task is of considerable difficulty, due to two technical constraints:

a. The source must suffer minimum overhead during the extraction, since other administrative activities also take place during that period
b. Both for technical and political reasons, administrators are quite reluctant to accept major interventions to their system's configuration; therefore, there must be minimum interference with the software configuration at the source side.

Depending on the technological infrastructure and the nature of the source system (relational database, COBOL file, spreadsheet, web site etc.) as well as the volume of the data that has to be processed, different policies can be adopted for the extraction step, which usually is also called “change data capture". The most naive possibility involves extracting the whole source and processing it as if the original first loading of the warehouse was conducted. A better possibility involves the extraction of a snapshot of data, which is subsequently compared to the previous snapshot of data (either at the source, or the DSA side) and insertions, deletions and updates are detected. In this case, there is no need to further process the data that remain the same. Another possibility, involves the usage of triggers in the source that are activated whenever a modification takes place in the source database. Obviously, this can be done only if the source database is a relational system; most importantly though, both the interference with the source system and the runtime overhead incurred are rather deterring factors with respect to this option. An interesting possibility, though, involves the “log sniffing", i.e., the appropriate parsing of the log file of the source. In this case, all modifications of committed transactions are detected and they can be “replayed" at the warehouse side. A final point in the extraction step involves the necessity of encrypting and compressing the data that are transferred from the source to the warehouse, for security and network performance reasons, respectively.

Transformation: Depending on the application and the tool used, ETL processes may contain a plethora of transformations. In general, the transformation and cleaning tasks deal with classes of conflicts and problems that can be distinguished in two levels [10]: the schema and the instance level. A broader classification of the problems is the following:

a. Schema-level problems: The main problems with respect to the schema level are (a) naming conflicts, where the same name is used for different objects (homonyms) or different names are used for the same object (synonyms) and (b) structural conflicts, where one must deal with different representations of the same object in different sources, or converting data types between sources and the warehouse.

b. Record-level problems. The most typical problems at the record level concern duplicated or contradicting records. Furthermore, consistency problems concerning the granularity or timeliness of data occur, since the designer is faced with the problem of integrating data sets with different aggregation levels (e.g., sales per day vs. sales per year) or reference to different points in time (e.g., current sales as of yesterday for a certain source vs. as of last month for another source).

c. Value-level problems. Finally, numerous low-level technical problems may be met in different ETL scenarios. To mention a few, there may exist problems in applying format masks, like for example, different value representations (e.g., for sex: `Male', `M', `1'), or different interpretation of the values (e.g., date formats: American `mm/dd/yy' vs. European `dd/mm/yy'). Other value-level problems include assigning surrogate key management, substituting constants, setting values to NULL or DEFAULT based on a condition, or using frequent SQL operators like UPPER, TRUNC, and SUBSTR.

To deal with such issues, the integration and transformation tasks involve a wide variety of functions, such as normalizing, de-normalizing, reformatting, recalculating, summarizing, merging data from multiple sources, modifying key structures, adding an element of time, identifying default values, supplying decision commands to choose between multiple sources, and so forth.

Loading: The end of the source records' journey through the ETL workflow comes with their loading to the appropriate table. A typical dilemma faced by inexperienced developers concerns the choice between bulk loading data through a DBMS-specific utility or inserting data as a sequence of rows. Clear performance reasons strongly suggest the former solution, due to the overheads of the parsing of the insert statements, the maintenance of logs and rollback-segments (or, the risks of their deactivation in the case of failures). A second issue has to do with the possibility of efficiently discriminating records that are to be inserted for the first time, from records that act as updates to previously loaded data. DBMS's typically support some declarative way to deal with this problem (e.g., Oracle's MERGE command [11]). In addition, simple SQL commands are not sufficient since the `open-loop-fetch' technique, where records are inserted one by one, is extremely slow for the vast volume of data to be loaded in the warehouse. A third performance issue that has to be taken into consideration by the administration team has to do with the existence of indexes, materialized views, or both, defined over the warehouse relations. Every update to these relations automatically incurs the overhead of maintaining the indexes and the materialized views.

3.2 Architectural Design & Design Choices

The approach of the INTEGRATE project for uniform access to relevant external data sources is shown in Figure 5. The general process is the following:

A. HL7 messages are extracted from these sources
B. Data are transformed to the Common Information Model
C. Data are loaded to the different INTEGRATE warehouses.

Then data can be accessed using a common query language. These steps are described in detail in the following sections. However, in order to perform data translation for integrating external sources to the project a Common Information Model (CIM) has been established in the INTEGRATE project.

It is based on the Common Data Model (CDM) and the Core Dataset (CD) as shown in Figure 5:

- Common Data Model: This is the common schema of the patient information stored at the different data warehouses. It is based on HL7-RIM. A common data model is used to resolve schema and record level problems that might occur in the ETL process.

- Core Dataset: It is the medical vocabulary used within the INTEGRATE platform. It standardizes the concepts used within INTEGRATE platform, including relationships to perform semantically-aware queries. It is based on SNOMED-CT, MEDRA and LOINC. Bu using a medical vocabulary the value-level problems are minimized

The selection of both CDM and CD is described in D3.1 [12].

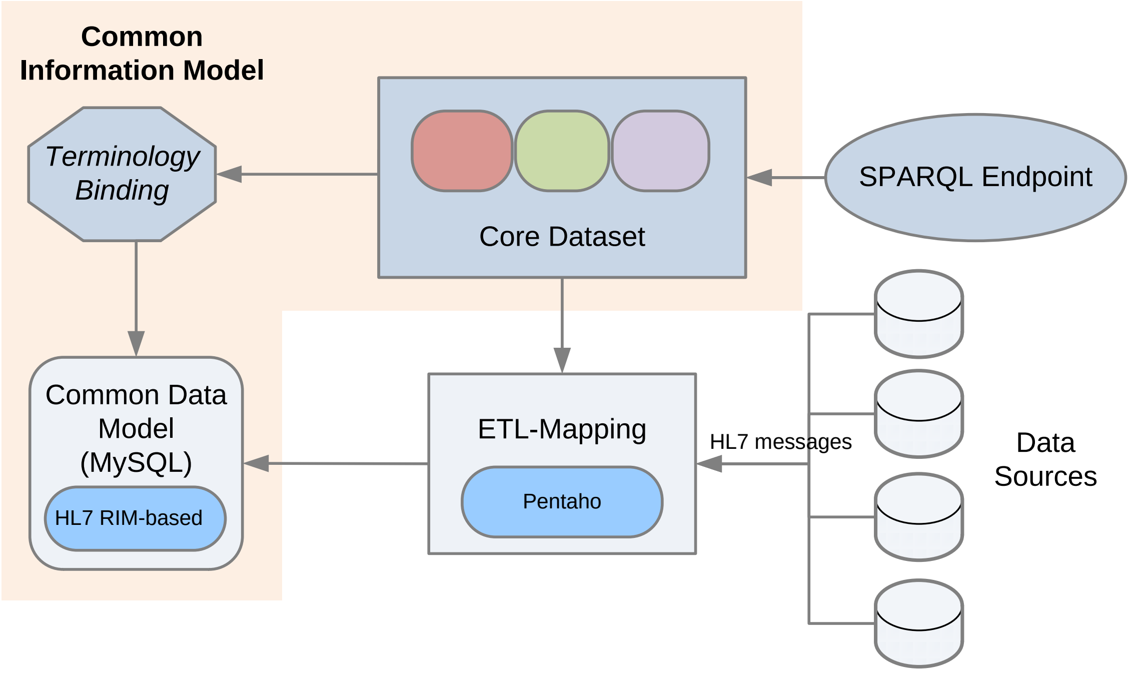

Figure 5. Integrate Common Information Model

3.2.1 Extraction

The main data sources for INTEGRATE project come from two operational domains: Trial Conduct Domain and Research Domain. To uniformly access data from external sources, we need to define a standard that such repositories need to be able to export. Following the project philosophy of reusing previous work, and to facilitate data exporting from external sources, we have selected the HL7 interoperability standard to provide the first step of accessing structured external sources within the INTEGRATE platform. We are aware that not every system in the health care domain will provide such exporting capabilities, but developing a new exporting standard would imply that no system would have such capabilities.

The proposed approach requires the export of the clinical data in HL7 compliant formats from the external data sources. The methodology on how this can be done cannot be generic since it varies among the clinical data management systems. In some cases, where the HL7 array of technologies is natively supported by the external system, this export operation can be easily supported, but in general we foresee that this process will require some system specific export functionality to be developed and installed as the “bridge”/”gateway” to the INTEGRATE platform.

3.2.1.1 HL7 messages

HL7 message standard, as mentioned on the previous section and it will be further detailed on Deliverable 3.4, has two versions that are currently in use, HL7 v2 and HL7 v3. Both versions define how information is packaged between the parts involved on a transaction. They set the language, structure and data types required for seamless integration from one system to another. Hence, they provide a specification for health and medical transactions and a common agreement among the parts involved.

Due to the fact that the HL7 International is a well-known organization that has created several standards in the healthcare environment, other software, from different companies allows export the information in the standards created by HL7. These standards aim to support hospital workflows, but each version of the standard is based on different interoperability formats. Therefore, processes to tackle these differences are specific for each version of the standard, and an approach will be given on the following sections.

3.2.1.2 HL7 V2

The HL7 v2 message standard is a human-readable stream of segments and delimiters. This syntax is not based on a common eXtensible Markup Language (XML) and has its own syntax.

HL7 v2 message consists of one or more segments that are separate by a carriage return character (\r) and consequently each segment will be displayed on a different line of text. Each segment consists of one or more fields separated by a pipe character (|). A field could contain other fields (components) that are normally separated by circumflex accent character (^), and also a component could contain a subcomponent that will be separated by ampersand (&). Tilde character (~) is the default repetition separator. If a message contains a special delimiter character, a different special escape sequence is used to specify the delimiter character in each case. Additionally, more special escape sequences are define for highlighting sequences, formatting sequences and character set codes. The name of each segment is specified by the first field of the segment, which is always a three character long code (Message Header MSH, Patient information PID, Next of Kin NK1, Patient Visit PV1, etc) that identifies the message type.

Complete understanding of HL7 v2 message syntax, may facilitate the creation a process which transforms these messages into a tabular file which will be the common pattern input for the ETL (extract, transform and load) tools. Consequently, an additional ETL mapping process takes place before the main ETL process.

3.2.1.3 HL7 V3 and Common Data Model XML Schema

The HL7 v3 message standard, as opposed to version two, is based on a XML (eXtensible Markup Language) encoding syntax that can be directly translated into a tabular file which will be the common pattern input for the ETL mapping tool. Therefore, the XML encoding provide an easier start for the ETL mapping tool, involving an initial less complex process from a data source based on a HL7 v3 message standard.

An HL7 v3 message stored as XML is specified by the HL7 Clinical Document Architecture (CDA), which is intended to specify the encoding, structure and semantics of clinical documents for exchange. The schema for any XML-based message must follow an XML schema that constrains the structure and content of an XML file. Any clinical document must have a minimal structure in order to satisfy the CDA standard. Thus, the list of minimal elements required by the CDA standard is:

- Header
  - CDA schema identification
  - Elements from "*ClinicalDocument*": typeID, id, code, effectiveTime, confidentialityCode
  - recordTarget
  - author
  - custodian
  - component
- Body
  - For a structured body: component-> StructuredBody->component->Section
  - For a non-structured body: component->nonXMLBody->text

In addition to the previous list of minimal elements required by the CDA standard, a XML Schema (XSD) for the INTEGRATE platform has been generated only with the information required by the Common Data Model (CDM) – describe in the next section. This XSD file will be used to validate the HL7 v3 messages, checking their structure, refusing messages if the structure is not correct, or even if there are other information that cannot be stored in the CDM due to its structure.

3.2.2 Transformation & Loading

Once HL7 messages have been exported from the external source, and validated against the corresponding XSD, an ETL tool has to be used to transform and store the information into the CDM. Different ETL tools were presented in the Deliverable 2.2 and finally the selected solution was Pentaho Data Integration (Kettle). This tool offers a graphical interface where could be constructed the ETL process, as a sequence of different operations (example shown in Figure 6). Moreover, the tool provides solutions for the resolution of all transformation and loading problems that we expect to occur.

With HL7 non-XML messages, and due to its structure, the ETL process has to first split the different fields, transforming the message in a tabular form file for further processes. Moreover, it is necessary to prepare the HL7 v3 messages from XML format to a tabular form file, but in this case, is done directly by selecting the different labels that compose the file and the data is extracted automatically.

Then, the tabulated HL7 v2 and HL7 v3 messages should follow a new process where the data will be analysed and stored into the different tables of the CDM, maintaining the consistency. There are different aspects that have to be considered to keep the consistency of the data in the created CDM. A list with the most relevant examples available is the following:

- A SubstanceAdministration instance cannot be stored without creating a Procedure instance and an Act instance.

- A Person instance cannot be stored without creating a LivingSubject instance and an Entity instance.
- An Observation instance cannot be stored without creating an Act instance.
- The different TargetSiteCodes have to be related with the instance of the Observation or Procedure that they belongs to.
- To relate a Person with an Act have to create an instance in the Role table and another instance in the Participation table.
- Each Patient should be checked if exists in the CDM before storing it and its new data.

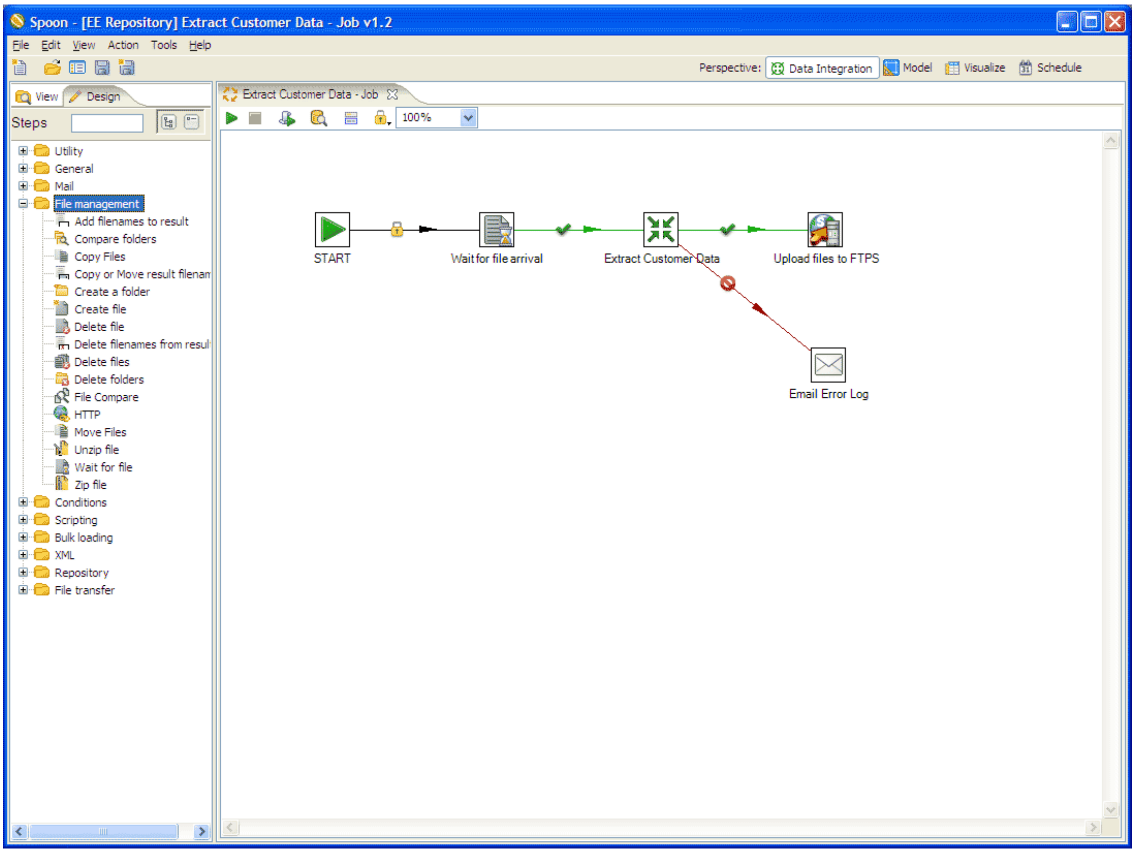

Figure 6. Pentaho Data Integration Example

Once data from the external source is stored using the INTEGRATE CDM, applications can homogeneously query such information using the semantic interoperability layer.

3.2.3 Common Query Language

In order for the data access services to provide a homogeneous interface, they need to use a common query language. The following main requirements were identified for the common query language:

- It must be sufficiently expressive. It should support the types of queries that clinicians and clinical researchers want to carry out.
- It must be attainable, with acceptable effort, to map the query language to those used by the various data sources that need to be accessed.

- It should be a community accepted standard. This ensures that there are sufficient support tools available (such as parsing and query engines).

There is a large number of query languages. Fortunately, most query languages can be quickly discarded, given the requirements above. Only the following query languages were seriously considered:

- SQL, as the most commonly used query language. It is the de facto standard for querying relational databases, which is currently the most popular database storage technology.
- XQuery [13], as the upcoming standard query language for XML data sources. XML is used as the message syntax for Web Services, so XQuery should be considered for this reason alone.
- SPARQL [14], as a query language for RDF data. RDF is the data model that is most frequently used for semantic mediation. Out of the many RDF query languages[7], SPARQL was chosen because it is in the process of becoming a standard, and more expressive than its predecessor RDQL [15].

Out of these, SPARQL meets our requirements best. One of the main reasons is that it is based on the RDF data model [16], which is a general graph-based model. It can be applied to data that is stored as an RDF graph, but also to data that is stored and represented using other data models such as relational or hierarchical ones. In contrast, both SQL and XQuery are very specific to their underlying logical data models, respectively relational and XML, and cannot easily be applied to others. Furthermore, SQL and XQuery are both substantially more complex than SPARQL.

SPARQL has an intermediate level of expressiveness. It is less expressive than either SQL or XQuery. Most notably, SPARQL does not support any form of aggregation[8]. It can only return the values that are in the underlying database, not any derived values obtained through counting, averaging, summation, etc. This is a drawback because it means that end-user queries that use aggregation are not directly supported. They can still be carried out, but aggregation will have to be performed client-side which can be significantly less efficient. It also means that users cannot ask how many “hits” their query has before deciding whether or not to retrieve these all. However, SPARQL does support the LIMIT keyword in order to limit the number of results that are to be returned. Furthermore, certain data sources do not support aggregation either, and may provide query functionality that is significantly more basic than the functionality provided by SPARQL.

[7] There are at least ten different ones, e.g. RDQL, Triple, SeRQL,Versa, N3, RQL, RDFQL, RxPath, SPARQL and SquishQL.

[8] Aggregates are supported in the forthcoming 1.1 version of SPARQL, which is currently a “Working Draft” available at http://www.w3.org/TR/sparql11-query/, but there are already many implementation sof it.

4 CONCLUSIONS

In this document we have described a solution for uniform access to external data sources. Firstly, our end-users identified the external data sources that are of interest to the INTEGRATE projects. Those are: The synapse commons repository, EHR & Clinical Trial systems. Then we described the technological choices we made in order to provide uniform access to those sources. Our solution at first uses well-established international standards (HL7 messages and documents) to export data from the aforementioned external sources. Then ETL tools parse those messages and load data to the Common Data Model. Subsequently, all data are made available within the INTEGRATE data warehouse and can be accessed using the SPARQL language.

5 REFERENCES

[1] D2 4 Initial system architecture and implementation status, The INTEGRATE Consortium, 2012

[2] https://synapse.sagebase.org/

[3] http://www.sagebase.org/

[4] http://www.the-dream-project.org/

[5] D4.1 Specification of the model, data and annotation repositories, The INTEGRATE Consortium, 2012

[6] HIMSS. EHR definition. http://www.himss.org/ASP/topics_ehr.asp. Accessed January 31, 2011.

[7] Gunter T.D., Terry N.P.:The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions. J Med Internet Res 7: 1, 2005.

[8] Blumenthal D, Tavenner M.: “The meaningful use” regulation for electronic health records. N Engl J Med. 2010;363(6):501–504.

[9] Rhoads, J.G., Cooper, T., Fuchs, K., Schluter, P., Zambuto, R.P.: Medical device interoperability and the Integrating the Healthcare Enterprise (IHE) Initiative. IT Horizons AAMI, 2010

[10] Lenzerini, M.: Data integration: A theoretical perspective. PODS, 233-246, 2002.

[11] Oracle. Oracle9i SQL Reference. Release 9.2, 2002.

[12] D3.1 Canonical models of CTMS and HER systems, The INTEGRATE Consortium, 2012

[13] Boag, S., Chamberlin, D., Fernández, M.F., Florescu, D., Robie, J., Siméon, S. (Editors): XQuery 1.0: An XML Query Language”, W3C Recommendation 23 January 2007, http://www.w3.org/TR/2007/REC-xquery-20070123/

[14] Prud’hommeaux, E., Seaborne A.: SPARQL Query Language for RDF, W3C Candidate Recommendation 14 June 2007, http://www.w3.org/TR/2007/CR-rdf-sparql-query-20070614/

[15] Seaborne, A.: RDQL - A Query Language for RDF, W3C Member Submission 9 January 2004, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

[16] Resource Description Framework (RDF), http://www.w3.org/RDF/

6 Appendix 1 - Abbreviations and acronyms

ASTM	American Society for Testing and Materials
CCD	Continuity of Care Document
CCR	Continuity of Care Record
CDA	Clinical Document Architecture
CDS	Clinical Decision Support
CIM	Common Information Model
CPOE	Computerized Physician Order Entry
CRF	Clinical Report Form
DCI	Data Collection Instrument (Oracle Clinical)
DCM	Data Collection Module (Oracle Clinical)
DICOM	Digital Imaging and Communications in Medicine
DVG	Discrete Value Group (Oracle Clinical)
EHR	Electronic Health Record
ETL	Extract Transform Load
HIE	Health Information Exchange
HL7	Health Level 7
IHE	Integrating the Healthcare Enterprise
QC	Question Group (Oracle Clinical)
RDF	Resource Description Framework
RIM	Reference Information Model (HL7)
SPARQL	SPARQL Protocol and RDF Query Language
XML	Extensible Markup Language