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

Project acronym: MyHealthAvatar

Deliverable No. D3.1
User Requirements

Grant agreement no: 600929

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

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D3.1
Document name:	User Requirements
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	1.0
Actual Submission Date:	30/06/2013
Editor: Institution: E-Mail:	Emmanouil G. Spanakis FORTH-ICS spanakis@ics.forth.gr

ABSTRACT:

This deliverable will review existing and emerging standards that are pertinent to the definition and building of the project's technological platform. The aim of this deliverable is to feed the subsequent tasks in the building of the system by providing a range of important technologies, standards, etc with their evaluation as enabling blocks of the MyHealthAvatar's architecture.

KEYWORD LIST:

User Requirements, Standards, Guidelines, Protocols, Formats, IT, State-Of-the-Art, Review

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

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

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

The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

MODIFICATION CONTROL			
Version	**Date**	**Status**	**Author**
1.0	08/07/2013	Draft	Emmanouil G. Spanakis
1.1	13/7/2013	Draft	Haridimos Kondylakis
1.2	20/7/2013	Draft	Manolis Tsiknakis
1.4	25/8/2013	Draft	Kostas Marias
2.0	09/09/2013	Final	Emmanouil G. Spanakis

List of contributors

- Emmanouil G. Spanakis (FORTH-ICS)
- Youbing Zhao (University of Bedfordshire)
- Xia Zhao (University of Bedfordshire)
- Feng Dong (University of Bedfordshire)
- Haridimos Kondylakis (FORTH-ICS)
- Tsiknakis Manolis (TEI-CRETE)
- Kostas Marias (FORTH-ICS)

Contents

1 Executive Summary

European healthcare systems have been subject to a long and complex history of independent evolution among many different countries. As a result, the picture is highly fragmented with differences between member states, regions, and even between hospitals within the same country. So, from the perspective of the individual patient, maintaining a clinical record in a consistent manner is difficult, and the problem is being exacerbated by the increased population movement within Europe. This situation poses as a threat to the provision of high quality healthcare services, and this is particularly true for the prediction and treatment of major and long-term diseases (e.g. cancer) where a consistent record of individual patients is of great importance. To this end, information collection, access, sharing and analysis have become a key to the problem that we are all facing in Europe. On the other hand the rapid progress of computing power and ICT technology offers great potential for addressing challenges in information access, collection, sharing and analysis for new knowledge discovery, and has led to a huge amount of valuable data becoming available on the web. These newly available technologies grant us unprecedented opportunities to support next-generation healthcare in tackling, among other things, the ageing population and the impact of its growth on the numbers of patients suffering from chronic diseases.

MyHealthAvatar proposes a solution for access, collection and sharing of long term and consistent personal health status data through an integrated environment, which will allow more sophisticated clinical data analysis, prediction, prevention and *in silico* treatment simulations tailored to the individual citizen. The proposed solution will be able to support: Information collection and access (Internal data repositories to store individual data for the avatars; links to external sources; model repositories, information extraction from the web and data collection using mobile apps; semantics and linked data to support the data/model searching and reasoning.), Data management and sharing, as well as Information analysis using integrated toolboxes.

In this deliverable we review existing and emerging standards that are pertinent to the definition and building of the project's technological platform. The aim of this deliverable is to feed the subsequent tasks in the design and development of the system by providing a review of important and relevant technologies and related standards, Together with their evaluation as enabling building blocks of the MyHealthAvatar's integrated architecture. This document describes all the activities planned for Task3.1 in respect to the user requirement analysis and review of existing technologies and emerging standards pertinent to the definition and implementation of the project's technological platform. If necessary this document will be dynamically updated in the course of the project to reflect project's findings, and will extend the list of technologies, standards, protocols, etc. that will be required for the architectural design and implementation of MyHealthAvatar technological platform.

2 Introduction

This document includes all related activities that form the core of the architecture definition process for an effective cloud based platform to support MyHealthAvatar. We will describe latest technology advance in cloud technologies and we will follow the IEEE 1471 recommendations [IEEE00]. The steps that will be taken to complete the related task include: capturing stakeholder needs; making a series of architectural design decisions that result in a solution to meet these needs, assessing it against the stakeholder needs, and refining this solution until it is adequate and capturing the architectural design decisions in an Architectural Description. We consider standardization issues with respect to the data storage and security, encouraging interoperability and data reliability and integrity. We also address computational requirements for simulation and 3D imaging data analysis. We plan for the architecture to be able to support multiple platforms, including Windows, Linux, iPhones, Android, etc. We pay special interest to develop a local cloud to serve as a pool of resources. This will be built on the existing facilities (computers, clusters) within an organization and support dynamic resource allocation.

2.1 Purpose of this document

The amount of data in healthcare that needs to be managed and the computational challenges introduced by MyHealthAvatar are ground breaking and call for innovative system design strategies. These requirements have given rise to High Performance Computing (HPC) and there are multiple instantiations of it, including cluster computing; grid computing and cloud computing [MEL11]. The cloud platform is currently the preferred solution for offering on-demand computing and data storage resources. Many commercial public cloud providers are already well-known [AMA, GGRI, NASA]. A number of cloud APIs are available. However, these systems and services are not ready for providing services and platform for MyHealthAvatar due to their performance restrictions and inherent legal and ethical implications. In contrast, community (private) clouds [AMVPC] become very useful for biomedical scientific research in order to have a full control on data access, reliability and storage management. A cloud platform involves a great number of technical issues, including virtualization [XEN, KVM] and software stack [EUC] for the infrastructure, optimization of resource management and allocation for deploying an application [ARM09], access to external HPC to gain extra computing resource that is not available within the cloud [DEI], support of scalable data repository and multiple cloud storages (via data federation) [VUK10, CAC10], data reliability and integrity [KAL07,JUE09], and securities over the network and infrastructure. From the perspective of a web service, an emerging trend is the use of simple REST web services [FIE02] that present a small entry barrier and a transition from the SOAP and WSDL Web Services technologies. These architectures are more bound to the existing web and also are more Semantic Web friendly since they share common basic infrastructure and interaction protocols. Also, architecture linking to external sources for data exchange concerns standardization and interoperability issues. Typical standards include EHR, HL7, OpenEHR]. MyHealthAvatar will utilize the latest architecture technology on cloud to lay down the foundation and allow high information security and effective integration of different components of the avatar to achieve high performance. Also, links to external sources will be explored together with the standardization issues.

The objective of this deliverable is to analyze user requirements and identify architectural components and standards for the realization of the architectural methodology, implementation and design of the envisaged project’s platform and is related with T3.1 on which all other tasks of WP3 will depend for providing: methodologies for the integration with external sources such as hospital records, existing data and model warehouse, social network, standards, guidelines and techniques in order to achieve system integration, techniques to build a local cloud infrastructure to support data processing by utilizing resources within individual institutions and market review of open source APIs.

User Requirements, Standards and guidelines

The function of user requirement analysis is to serve as the mandate or terms of reference for the design, development and realisation of MyHealthAvatar platform. This document is produced by analysing user requirements standards, processes and guidelines based on the stipulations of the project definitions, the implementation plan and selected scenarios of use.

2.1.1 What is a standard?

A standard is an agreed way of performing an action. Standards cover a huge range of activities and can be described as tools of knowledge. They can be divided into: *Technical standard*: “a formal document that establishes uniform engineering or technical criteria, methods, processes and practices” and *De-facto standard*: “a custom, convention, product, or system that has achieved a dominant position by public acceptance or market forces”[3].

Standards might also be divided broadly by what they govern (from Wikipedia):

- *A standard specification is an explicit set of requirements for an item, material, component, system or service. It is often used to formalize the technical aspects of a procurement agreement or contract. For example, there may be a specification for a turbine blade for a jet engine which defines the exact material and performance requirements.*
- *A standard test method describes a definitive procedure which produces a test result. It may involve making a careful personal observation or conducting a highly technical measurement. For example, a physical property of a material is often affected by the precise method of testing: any reference to the property should therefore reference the test method used.*
- *A standard practice or procedure gives a set of instructions for performing operations or functions. For example, there are detailed standard operating procedures for operation of a nuclear power plant.*
- *A standard guide is general information or options which do not require a specific course of action.*
- *A standard definition is formally established terminology.*
- *Standard units of measurement, in physics and applied mathematics, are commonly accepted measurements of physical quantities.*

In this document we are primarily concerned with standard specifications, practices and guidelines (*software, guidelines, protocols, formats and laws*). We are primarily concerned with IT and non IT standards that might influence MyHealthAvatar’s project architectural design. Especially we focus on state of the art visualization technologies and methodologies for the envisaged 4D health avatar implementation.

2.2 Motivation for using standards

There are several advantages of making use of a standard:

[3] *HTML is a good example of both: it started out as a de-facto standard and was transformed into a technical standard.* (http://www.whatwg.org/specs/web-apps/current-work/)

- Enable interoperability with other platforms which uses the same standard. Make a software system/subsystem potentially a drop-in replacement encouraging adoption of technological innovations.
- Avoid *redoing* work developing new protocols, formats, software and other entities which are typically associated with a standard
- Make software systems easier to adopt, maintain and develop by new people who are familiar with the standard
- Adopt legal requirements (or the standard may take the form of a legal requirement)
- Enforce minimum quality standards lending towards credibility to a software system, making it more likely to be adopted, particularly in mission critical or life-or-death applications

But there are also some disadvantages:

- It is not widely adopted, nullifying many of the above advantages
- It may be complex or heavyweight and therefore hard to comply to
- It may be lightweight that few benefits are gained from adhering to it, compared with the effort
- It might not be a good fit for the software system being developed - it may lack required elements, or make demands unnecessary for the task at hand
- It is in development or constantly changing, making compliance difficult
- There are many competing standards and there is no clear picture of which will ultimately win

2.3 VPH model toolkit

The VPH initiative is expected to deliver a wide range of content, including tools, data and models that will assist in the dissemination of content through a central point of access. The **VPH network of excellence (VPH-NoE)** project is developing a series of guidelines[4] that will address relevant issues to content deliverers who wants to share their content through the VPH-NoE toolkit portal. These guidelines range from toolkit, model and data characterization to ontology, interoperability, licensing, usability as well as legal & ethics guidelines. The VPH guidelines provide a useful set of information, a starting point on how to classify content, and ancillary information related to the content itself.

2.3.1 VPH Toolkit/Model/Data and ethics

The purpose of the **toolkit** characterization guidelines[5] is to present a set of criteria that will identify the functionalities of the toolkit in its domain as well as a method to assess the quality of the toolkit. The toolkit guideline identifies a number of areas of standards that are applicable including input/output format, language, operating system, third party library, documentation, maintenance & versioning, license and certification. These guidelines also discuss methods for verification, ownership, training, maintenance as well as suggested method of ranking. Although this guideline is not complete, it contains much useful information and a starting point on the approach and methods to classify toolkits for MyHealthAvatar project.

[4] http://toolkit.vph-noe.eu/toolkit-guidelines

[5] http://toolkit.vph-noe.eu/component/docman/doc.download/2-g01-toolkit-tool-characterisation-guideline-v10

The goal of the **model** characterization guideline[6] is to ensure that models can be understood by the end user, coded and solved. This has become increasingly important as models have become more complex. Furthermore, to be used in a clinical environment, a model must be able to be validated and demonstrated to be reproducible. This guideline focuses on the development of specification of the minimum information required to describe a model as well as the development of model encoding standards. The minimum information specification described in this guideline includes the MIRIAM (Minimal Information Required in Annotation of biochemical model), MIASE (Minimum information about a simulation experiment) and MIBBI (Minimum information about a biomedical or biological investigation). Several encoding standards have also been included: CellML, FieldML, SBML as well as NeruoML, InSilicoML. The typical strategy for developing an encoding standard include the development of *a markup language for metadata and data*; *an application programming interface based on the MLs; libraries of tools that can read and write ML files; data and model repository based on MLs; and a metadata framework that demonstrates model reproducibility*. The guideline also describes validation methods, training, maintenance, as well as ranking methods. Not all sections are completed, however this guideline provides a general overview on common used standards and specifications in the area of modelling that will be adopted by MyHealthAvatar platform.

The exchange of **data**, on the other hand, between different users is an important goal of the VPH toolkit portal. The goal of this guideline[7] is to facilitate the exchange of the data through a trusted mechanism. It defines the scope of data characterization principles including: ethics, law, licensing, reproducibility, interoperability and sustainability. It includes data sharing based on quality standards and good practice such as technical characterization, provenance and standard formats. The Data guideline recommends licenses which are compliant with the *Open Knowledge* definition developed by the *Open Knowledge* foundation. **Ethics** are described in a number of areas including the *ethical* use of personal and health data within the VPH project. The importance of ethics within the VPH project is considered to be an important factor on the longevity of the project, as such the use of material within the VPH should acknowledge the right of those that contributed the data including specific right to publish, the role of all contributors as well as compliance with relevant legal requirements. The **European Data protection directive (Directive 95/46/EC)** is designed to protect the privacy of personal data and with freedom of information measure (**European Union Directive 2003/98/EC**) supports the right of individual to inspect the nature of information held about them. Ethics in terms of health data requires that contributors can: *withdraw or refuse the use of their data at any time, treat data with respect and* confidentially and that *exploitation will not expose them to unnecessary level of risk.* These principles can be found at the **Research Ethics, Committees, Data Protection and Medical Research in European Countries, Directive 2001/20/EC**. The data guideline characterizes data standards as well as inclusion of metadata which complements

[6] http://toolkit.vph-noe.eu/component/docman/doc.download/3-g02-toolkit-model-characterisation-guideline-v10

[7] http://toolkit.vph-noe.eu/component/docman/doc.download/8-g03-toolkit-data-characterisation-guideline-v10

the raw information; processing which describes how data may have been processed to allow repetition of a study; interoperability which describes how easy is it to read or write the data including documentation and openness and sustainability which related to the data format used and how it will be stored. The criteria of sustainability for file formats includes: extensibility, user base, licensing, backward compatibility, stability. The data characterization also provide use case sharing plans from bio-sharing, cardiac atlas project, and medical research council data sharing initiative as well as the Open Provenance Model.

2.3.2 Ontology description and guidelines

The ontological guidelines are intended to support the verifiability and re-use of data and model resources. A key aspect is to develop a strategy for resources that is both human and machine readable. These include metadata to hold resource annotation and use of knowledge representation and ontologies in annotation. Within the VPH project, the key approach is to develop and support semantic interoperability between data and model resources.

Semantic interoperability aims to provide a standardized as well as machine readable context to data. This allows interpretation of different resources more easily. This guideline uses a dedicated semantic integration framework developed through the RICORDO VPH project. This guideline provides a useful overview of the scope of ontology, semantic integration and the relevance of these to the VPH project. This guideline makes the following recommendation for the following areas:

- Biological structures:
  - Foundational model of anatomy
  - Edinburgh Mouse Atlas (EMAP)
  - Mouse Anatomical dictionary (MA)
  - The Cell Type ontology
  - Gene ontology cell component
  - Protein ontology
  - Chemical Entities of Biological Interest
- Biological Events
  - Go Biological Process
  - Mammalian Pathology
- Qualities
  - Go Molecular Function
  - PATO
  - Ontology for Physics in Biology
- Classes of entities in Experiments, Modeling and Simulation
  - Units Ontology
  - Ontology for Biomedical Investigation

The ontology guideline also lists a range of knowledge representation tools relevant to the VPH community:

- Knowledge Representation Languages
  - Prolog-based: in SWI-Prolog system
  - LISP-based: OCML,
  - Common Logic
  - OBO format
- Web Languages
  - RDF
  - RDF-Schema
  - OWL

The ontology guideline mention tools in authoring, browsing & validation, reasoning and management as well as methods of deployment and acknowledges the difficulty in assessing and ranking the usefulness of any ontology. Some criteria to assess ontologies include:

- Absolute generic criteria (Availability, Usability, Documentation and training, Robustness of validation procedures)
- Context dependent criteria (Integration and interoperability, Adoption of communal standards, Strength and commitment of developer community and users, Quality, Generic application)

2.3.3 Interoperability

Interoperability according to International Standards and Organization (ISO/IEC 2382-01, Information Technology Vocabulary and Fundamental Terms) can be defined as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristic of those units". The interoperability guideline[8] attempts to provide a set of practical interoperability recommendation in specific domains. Interoperability is seen as an important aspect in the effort to integrate tools, data and model to support the development of patient specific computer models and their applications.

According to the Interoperability guideline, interoperability can be classified in the following manner:

- Level 0: Standalone system with no interoperability
- Level 1: Technical interoperability with communication protocol available for exchanging data.
- Level 2: Syntactic Interoperability where a common structure is available to exchange information
- Level 3: Semantic Interoperability where a common information exchange model is used
- Level 4: Pragmatic Interoperability where a system is aware of procedures of other systems.
- Level 5: Dynamic Interoperability where a system can comprehend the state changes that occur in the assumptions and constraints each is making over time.
- Level 6: Conceptual Interoperability where the system is fully specified and be able to be evaluated by other engineers.

The interoperability guidelines list out the following standards or commonly used tool in the following domains: Modelling (CellML, FieldML, SBML, NeuroML), Data (DICOM, HL7, JPEG, TIFF, VTK file format, BioSignalML, GDF, Analyse 7.5, Nifti), Ontology (FMA, SNOMED CT, GO, LOINC, MIASE) and Infrastructure (JSDL, OGSA-BES, OGSA-DAI, GridFTP, SSH, MTOM). The guidelines outline the need to deploy common standards as well as issues related to each specific area including legacy systems. The guideline suggests the use of IHE (Integrating the Healthcare Enterprise) as a means of verification. IHE Gazelle Tools has been developed to test interoperability according to the IHE standards.

2.3.4 Licensing

Licensing guidelines[9] highlight issues involved with software, data and content licensing. They provide a concept of licensing as well as standards and standard bodies such as: Open Source

[8] http://toolkit.vph-noe.eu/component/docman/doc.download/5-g05-toolkit-interoperability-guideline-v10

[9] http://toolkit.vph-noe.eu/component/docman/doc.download/1-g07-toolkit-licensing-guideline-v10

Initiative (software licensing); Free Software Foundation; Open Knowledge Foundation; Creative Commons; Science Commons.

The guidelines recommend specific licenses, with relevant characteristics. Software, data and content licenses are characterized according to 5 criteria, including OSI approval, business-friendliness, GNU GPL (viral copyleft license) compatibility, attribution and the legal jurisdiction specified in the license.

- Software licenses: AGPL - The GNU Affero General Public License v3, Apache v2 - The Apache License, 3-clause BSD - The 3-clause BSD License, 4-clause BSD - The 4-clause BSD License, CeCILL v2 - The CeCILL License, EUPL - The European Union Public License, GPL v2 - The GNU General Public License version 2, GPL v3 - The GNU General Public License version 3, LGPL v2.1 - The GNU Lesser General Public License version 2.1, MIT - The MIT License, MPL v1.1 - The Mozilla Public License 1.1
- Data licenses: Open Data Commons public domain dedication and licenses, Open Data Commons attribution license, Open Data Commons Open Database Licenses, Creative Commons CCZero
- Content Licenses: Creative Commons Attribution license family, Creative Commons CCZero, GNU Free Documentation License

Licensing guideline is particularly useful to the project due to the development environment as well as the amount of data involved.

2.3.5 Legal and ethical

The Legal, Ethical & Provenance guidelines describes the legal and ethical requirements when interacting with the VPH toolkits, including copyright, data protection and freedom of information. The close association of VPH with industry and clinic requires both provenance and ethical standards to be followed. The guideline raises practical legal issues relevant to the delivery of content to the toolkit. It is categorized into sections on simulation & data, security, research & innovation, tools & techniques, interoperability & workflow and standards. The guidelines point out that the common legal issues which arise regularly includes whether licensing has been clearly stated and attached, whether all involved parties have been acknowledged and the copyright conditions clearly stated and the whether an adequate disclaimer have been attached to the toolkit content. These legal issues are relevant to all users. Personal interaction with the toolkit will result the user being classified either as a user or author and hence certain legal responsibility will fall on the user.

The ethical guidelines highlight the issues relevant to VPH toolkit users which can then be used as a framework on the definition of acceptability and equality. Relevant ethical issues are categorized into sections on software & data, community, sustainability of infrastructure, exploitation and sustainability of interoperability and workflow. The ethical guidelines can be considered as a best practice guide. Given the close association of VPH with industry and clinicians, the need to follow ethical standards closely is emphasized. A basic principle of consent, authorization and anonymization should be adopted.

The guidelines also a brief background on the significance and importance of provenance. The use of Open Provenance Model is recommended. The issues of provenance is discussed in the following categories: protocol, quality & provenance in data and software, security, simulation, provenance protocol, research and innovation, tools and techniques, interoperability and workflows, sustainability in community, ontology, documentation, exploitation.

The legal and ethical guidelines provide a number of use case studies. Given the similarities and overlapping of some VPH projects and p-medicine, the issues and standards raised in these guidelines are particularly relevant to the scope of p-medicine given the close association to clinical data and software development

3 Software and data standards/requirements

The various steps and processes in software engineering are typically categorized by the IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK), ISBN 0769523307 [11] as :

- Analysis/Specification
- Architecture/Design
- Implementation/Testing
- Evaluation/Maintenance/Documentation

In the sections below we present the existing standards for each category as well as selected de facto standards/technologies in the software engineering industry. MyHealthAvatar project has a large consortium of partners with backgrounds ranging from clinical to academic to industry. The sections below will be used as a reference guide for creating the necessary engineering software models that will be adopted by different patterns and institutions to create the envisaged platform.

3.1 Software standards and requirements

3.1.1 Requirement Analysis

The analysis process in software engineering comprises the elicitation specification and validation of the requirements for software engineering. The software requirements are properties which must be exhibited to solve particular problems (i.e. software components should be verifiable). In order to achieve this, requirements should be expressed as clearly as possible, and where appropriate with quantity metrics and measurable properties. There are various methodologies to gather and document the software requirements, there are also defined standards for the requirements specification and there are also a great number of tools for the requirements management, tracking, verification and documentation, which we outline below.

3.1.2 Design requirements and methodologies

To the best of our knowledge, there is not a standard or de facto standard on how to collect the software requirements of a system. Although there are existing guidelines and standards on how to record and express requirements, there is not a standard process on how to acquire them. The first and most important step in the requirements elicitation process is to correctly identify the stakeholders of the software under development. Different groups of stakeholders, such as end users, customers, software engineers, government regulators, might have different, or even conflicting, requirements. If we fail to identify all the correct stakeholders in the requirements elicitation process, we usually end up delivering a product which has not taken in to account the correct set of requirements, thus we build a wrong or problematic piece of software. Apart from the stakeholders, there are also other sources from where the software architect or the requirements engineer can draw the requirements, such as the specific domain knowledge he might have or acquire from an advisor with expertise on the subject, the operational and organizational environment of the software and others. There are various methodologies which we use for gathering the software requirements, each one having its advantages or specific cases in which may be more appropriate.

Interviews (and follow up interviews) is the classic way of asking users what they need from the end product, how they imagine it and what particular preferences they have from it.

Questionnaires can be used to gather requirements from large user groups, since they can be distributed to many users simultaneously and processed automatically (e.g. online questionnaires). We can also give specific directions to the users through the questions and acquire more meaningful answers than the vague requirements which an interview might come up with, but there is also the danger to have biased questions and thus biased answers, so designing a questionnaire demands good knowledge of the problem domain.

Prototypes is a valuable tool for providing a fast insight of the users to the end product and understand better what kind of input information they need to provide. Paper or screen mock ups are an especially useful type of prototype when building user interfaces.

Brainstorming is used for acquiring requirements, especially when we have a good knowledge of the domain or when we can consult domain experts on it. It can also help when we consult in parallel different stakeholder groups, although it needs a good organization and agenda of the discussion, else the conversations might end up with vague views and no clear requirements.

Scenarios/use cases can provide a framework to identify the usual operational and functional requirements of software without explicitly asking the users for the requirements but rather for their usual tasks, work routine and workflows. We can also introduce hypothetical scenarios of the kind "what if" and design the software accordingly.

Modelling a specific problem and splitting it up into various subcategories, tasks, stakeholders we can gather the requirements needed to solve the problem. This usually might imply that we also need to sketch or apply existing models about the domain, the usage, the user interface etc. Conceptual modeling usually helps in understanding the problem, not designing the solution. Modeling also allows for a more formal definition both of the problem and of the requirements, which allows also for a more systematic tracking and verification of the requirements.

The requirements of a typical system usually are numerous. Requirements also have several properties, such as quantification of desired attributes, preconditions and post conditions, and other defining elements. Consequently, it is difficult to record, keep track, verify, refine, and manage in any other way all the requirements of a system, unless an automated Requirements Management tool is used to help in this task. Numerous such tools exist, both proprietary or provided as free software. Each of these tools might have also advanced capabilities such as testing the software against the defined requirements, or be part of more general set of tools for project management, software design and so on. Based on the requirements of all the stakeholders, we proceed to the design of the system, which can be split up to two steps; the high level Architectural design of the main elements and components of the system and the low level Detailed design which defines the specific behavior, interfaces and algorithms of the architectural components.

3.2 Implementation Methodologies & Environments

The software construction, or implementation, is the detailed creation of software, based on its requirements, through a combination of coding, verification, testing, integration, and debugging. Throughout construction, software engineers constantly test their product against errors and defects, so the software implementation is closely linked to the testing procedure, although it might be necessary in many systems to have a follow up testing phase in order to ensure and guarantee some quality attributes or test the overall integration status of the system.

A common consideration in the software implementation is that, more often than not, the software requirements might change. Consequently, the software construction methodology should take this into account and be flexible enough to adapt to changing requirements, to be able to adopt new requirements added from the testing and evaluation phase and to verify that the end product is in accordance with its requirements. For that reason various methodologies have been developed over time, which set some guidelines on how to implement software. Below we outline the models of some of the most known techniques, although many other variations of them exist.

Waterfall Sequential design process is a sequential development approach in which development flows directly from one phase to the other (analysis, design, implementation, maintenance). Emphasis is given to strict planning, time schedule, milestones and deadlines of each phase, thus not allowing backtracking or flexibility on the development scheduling.

Iterative Cyclic process of prototyping, testing and refining is a cyclic process as a response to the weaknesses of the (sequential) Waterfall model. It starts with an initial planning which is refined through iterations of waterfall-like procedures of analysis, design, implementation and evaluation. In these iterations, each phase builds on the previous ones by taking into consideration the difficulties, problems and modifications which have been introduced since the previous planning.

Spiral Design and prototype in stages combines elements of design and prototyping in stages. Starting from a minimal project, the software development progresses into setting more requirements, adding stakeholders, refining the implementation and adapting the design in order to gradually expand into a fully defined and implemented system.

Agile Changing requirements is a group of methodologies, such as Scrum, Extreme Programming, Adaptive Software Development, Feature Driven Development and others, which share some common characteristics. The core characteristics, of this process, are the continuous adaptation of the software development based on a set of changing requirements, the rapid delivery of software by frequent releases of new working versions and the close cooperation of the end-users with the developers.

Rapid prototyping Development is a methodology which facilitates the software construction through iterations and refinements of prototypes. Starting from GUI (graphical user interface) mock ups or prototypes with reduced functionality, this methodology prioritizes the business needs and the requirements and proceeds to next versions of prototypes with added functionality. These throwaway prototypes usually have little or no documentation and have active user involvement and continuous evaluation in order to achieve fast delivery with low cost.

Test Driven Development is a model which relies on late implementation of functionality, until it is actually needed in the system. It starts with prototype implementations, which are gradually developed based on failing unit tests or integration tests, which introduce newly needed functionality. It can be considered as a variation of the RAD, although it usually does not deliver working prototypes but it builds over the failing ones. Although it is accredited for development speed, flexibility and extensibility, it is also criticized for the lack of clear management strategy.

Cleanroom Based on formal methods, certifiable reliability and adherence to specifications is intended to produce software with a certifiable level of reliability and validation against its specification. The main principles of this methodology is that it uses formal methods for the specification and designing of a system, automated tools or standard based quality control, and testing with statistical models which can statistically verify the reliability of the software and the estimated confidence which is implied.

The formal methods for specification, development and verification of computer and software systems are driven by the expectation that mathematical analysis and formalism can contribute to the reliability and robustness of a system. However, the high cost of using formal methods to systems development, usually restricts their usage only to high integrity systems where safety, security or robustness is of utmost importance. Some examples include: *Petri Nets* net is one kind of mathematical modeling language for the description of systems; *Z Notation* is a formal specification language which is used for describing and modeling computing systems. Since 2002 it has been standardized, under the ISO/IEC 13568:2002 [21] standard; and *Vienna Development Method* (VDM) is a formal method for computer system development, which supports modeling, testing and proving specific properties of the models and code generation from the validated models.

When the software engineering process comes to the point of actual implementation for a specific platform or environment, many practical considerations apply. These considerations vary from pure subjective preference over one environment or the other, to actual limitations, or benefits in comparison between environments. These considerations can be about interoperability, reusability, community support, cost, documentation, available tools, openness and standards compliance, previous technical expertise of the software development team and many other factors.

Only a few environments or platforms have been standardized in the software engineering industry, because the industry needs are always pressing for refined and improved versions with new or modified functionality. Such examples of standards include programming languages or environments which have met wide acceptance by the computer science community during decades, such as the C (ISO/IEC 9899:1990) and the C++ programming language (ISO/IEC 14882:2003), FORTRAN (ISO/IEC 1539-1:2004), and various versions of POSIX family of standards, which is an IEEE standard of system interfaces for the Unix operating system (e.g. POSIX:2008 or IEEE:1003.1-2008). However, some platforms are considered to be de facto standards nowadays, because they maintain a large user and developer group. From the commercial and proprietary environments, the most prominent are the technologies of Microsoft Corporation (MSWindows [24], VisualStudio [25], .NET framework [26]) and Apple Inc. (Mac OS X [27], Xcode [28], Cocoa [29]). From the open source, free software or community based technologies the most prominent are the various Linux (Unix-type) distributions (Ubuntu, Fedora, Debian, openSUSE, CentOS, Knoppix etc), the Java [30] programming language, the Eclipse [31] and NetBeans [32] IDEs and the LAMP [33] (Linux, Apache, MySQL, Perl/Python/PHP) collection of free, open source software.

Although the architecture and design of a system should not rely on implementation issues such as the underlying execution environment, and although the architecture definition should be made based on communication and data exchange standards and protocols, abstraction, interfaces and separation of implementation issues, many times the exact execution environment defines various architectural decisions because of the availability or not of various libraries, protocol implementations, adherence to standards, interoperability issues and other practical reasons. Consequently, knowing the exact development and execution platform, sometimes can enhance or degrade the development process.

3.2.1 Source Code utilization and evaluation

The software development aims to deliver a product which satisfies user requirements. Since the user requirements change or evolve through time, defects are uncovered and technology evolves, the maintenance phase is a significant part of a product’s lifecycle even though it usually does not receive as much attention as it should. The driving needs for maintenance can vary, such as correcting faults, improving the design or the implementation, interface with other systems, adaptations for different hardware and software requirements and so on. Thus, we can categorize the types of software maintenance as:

Corrective: reactive modification of a software product performed after delivery to correct discovered problems.

Adaptive: modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.

Perfective: modification of a software product after delivery to improve performance or maintainability.

Preventive: modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

All these are described in the ISO/IEC 14764-1999 Software Engineering - Software Maintenance standard which describes standardized techniques for software maintenance. Other standards which are related to software maintenance, evaluation or quality metrics which affect maintenance and evaluation processes are:

- IEEE 610.12-1990 (R2002), IEEE Standard Glossary of Software Engineering Terminology.
- IEEE 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.
- IEEE 1219-1998, IEEE Standard for Software Maintenance.
- IEEE/EIA 12207.0-1996//ISO/IEC12207:1995, Industry Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology -Software Life Cycle Processes.
- IEEE Std 14143.1-2000//ISO/IEC14143-1:1998, Information Technology – Software Measurement-Functional Size Measurement - Part 1: Definitions of Concepts.
- ISO/IEC 9126-1:2001, Software Engineering-Product Quality - Part 1: Quality Model.
- ISO/IEC 14764-1999, Software Engineering - Software Maintenance.
- ISO/IEC TR 15271:1998, Information Technology - Guide for ISO/IEC 12207, (Software Life Cycle Process).

3.2.2 Cloud software

Cloud computing has been receiving much attention as an alternative to both specialized grids and to owning and managing one's own servers. Currently available articles, blogs, and forums focus on applying clouds to industries outside of biomedical informatics. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS).Some vendors use terms such as IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) to describe their products. The line between "low-level" infrastructure and a higher-level "platform" is not crisp. We believe the two are more alike than different, and we consider them together and we elaborate on them in section 6.15 where we discuss interoperability issues concerning general public and private cloud infrastructures. The data center hardware and software is what we will call a cloud.

Cloud computing shares characteristics (from en.wikipedia.org/wiki/Cloud_computing) with:

- Client–server model / distributed application model
- Grid computing / distributed and parallel computing
- Mainframe computer / Powerful computers, critical applications
- Utility computing / packaging of computing resources as a metered service
- Peer-to-peer distributed
- Cloud gaming / on-demand gaming

3.2.3 Mobile distributed environments

The rapidly expanding technology of cellular communications allows users the capability of accessing information regardless of the location of the user or of the information. It is expected that in the near future, tens of millions of people in the U.S. alone will carry a lightweight, inexpensive terminal that will give them access to a world-wide information network called PCN (Personal Communication Network). These users will be constantly relocating between cells of size much smaller than today (future cell size might be a building or a floor of a building).

Wireless communications and mobility, on the other hand, introduce a new paradigm of distributed computing[10]. Today's computer systems often depend heavily for their operation on the rest of the network. Mobile computers, however, are very susceptible to network disconnections. The majority of these disconnections are voluntary. Frequently, users will deliberately avoid use of the network for cost or power consumption or because no networking capability is available at their current location. Thus, many users will be only occasionally connected to the rest of the network. Handling disconnections has been discussed extensively in the context of network partition[11]. Furthermore, in network partitions, disconnections are involuntary and mostly unpredictable since they result from network or host failures.

Wireless networks deliver much lower bandwidth than wired networks, and have higher error rates[12]. Mobile systems are also characterized by high variation in network bandwidth that can shift one to four orders of magnitude, depending on whether the host is plugged in or using wireless access and on the type of connection at its current cell. As a consequence, concurrency control schemas for mobile distributed systems should meet novel objectives such as:

1. support the autonomous operation of mobile hosts during disconnections,
2. reflect a greater concern for bandwidth consumption and constraints,
3. adapt to varying connectivity conditions, and
4. take into account the changing locality.

Future health informatics for personalized e-Health services must rely on technologies and systems for transparent and continuous collection of evidence-based medical information at any time, from anywhere, and despite the coverage and availability of communication means. Disruption and Delay Tolerant Networking (DTN) is a novel approach for next-generation e-Health information exchange where end-to-end homogeneous networking connectivity is not available[13]. This setting can be

[10] T. Imielinksi and B. R. Badrinath. Wireless Mobile Computing: Challenges in Data Management. Communications of the ACM, 37(10), October 1994.

[11] S. B. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in Partitioned Networks. ACM Computing Surveys, 17(3):341{370, September 1985.

[12] G. H. Forman and J. Zahorjan. The Challenges of Mobile Computing. IEEE Computer, 27(6), April 1994.

[13] Spanakis, E.G., & Voyiatzis, A.G. (2012). DAPHNE: A Disruption-Tolerant Application Proxy for e-Health Network Environments. 3rd International Conference on Wireless Mobile Communication and Healthcare, Paris, France, November 21-23, 2012.

applied in both rural and urban environments and in both disaster events and normal day-to-day life. The ability of DTN to provide in-transit persistent information storage allows the uninterruptible provision of crucial e-Health services overcoming network instabilities, incompatibilities, or even absence for a long duration.

3.3 Software Quality

The notion of quality is sometimes perceived as a matter of subjective criteria. Various definitions about the software quality exist, which can be summarized as the degree of conformance to user requirements, or the degree of customer satisfaction from the end product. Quality characteristics may be required or not, or may be required to a greater or lesser degree, and trade-offs may be made among them.

In the industry there are various standards about quality procedures and metrics which can also be, partly or fully, applied to the software engineering process, such as the ISO900 [43] family of standards about quality management, the CMMI [44] (Capability Maturity Model Integration) which deals with process improvement approaches and the ISO/IEC 15504 [45] standard (also known as SPICE - Software Improvement and Capability Determination) which is a set of technical standards for the computer software development processes and business management functions.

The quality assurance process on software development is based on verification and validation of the end product based on testing, inspections, audits, technical and management reviews, as well as adhering to standard processes which ensure quality management via the lifecycle of software.

There are many practical considerations which affect the quality of the end software product, such as the particular domain of the software, its requirements, the external components which are used in the system, the methods and tools which are used during the development, maintenance and evaluation of the system, the budget, the staff, the project organization and the scheduling of all the processes. Even the defect characterization and handling can reveal the degree of quality in a system, such as the distinction and handling between different types of defects and the so-called fault-tolerance of a system to situations such as Errors (the difference between a computed result and the correct result), Faults (incorrect step, process, or data definition in a computer program), Failures (the (incorrect) result of a fault) and Mistakes (a human action that produces an incorrect result).

The models of software product quality often include measures to determine the degree of each quality characteristic attained by the product. If they are selected properly, measures can support software quality (among other aspects of the software life cycle processes) in multiple ways. They can help in the management decision process. They can find problematic areas and bottlenecks in the software process and they can help the software engineers assess the quality of their work for longer-term process quality improvement. There are also a few topics where measurement supports software quality management directly. These include unit testing results, reliability models and benchmarks, fault and failure data, statistical tests and analysis of prediction models.

Some standards which are related with software quality assessment, measurement, definition, methodology and guidelines, are:

- IEEE 730-2002, IEEE Standard for Software Quality Assurance Plans.
- IEEE 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.
- IEEE 1008-1987 (R2003), IEEE Standard for Software Unit Testing.
- IEEE 1012-1998, Software Verification and Validation.
- IEEE 1028-1997 (R2002), IEEE Standard for Software Reviews.
- IEEE 1044-1993 (R2002), IEEE Standard for the Classification of Software Anomalies.
- IEEE 1059-1993, IEEE Guide for Software Verification and Validation Plans.
- IEEE 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.
- IEEE 1228-1994, Software Safety Plans.
- IEEE 1462-1998//ISO/IEC14102, Information Technology - Guideline for the Evaluation and Selection of CASE Tools.
- ISO/IEC12119:1994, Information Technology-Software Packages - Quality Requirements and Testing.
- ISO 9001:2000, Quality Management Systems - Requirements.
- ISO/IEC 9126-1:2001, Software Engineering - Product Quality, Part 1: Quality Model.
- ISO/IEC 14598:1998, Software Product Evaluation.
- ISO/IEC 15026:1998, Information Technology - System and Software Integrity Levels.
- ISO/IEC TR 15504-1998, Information Technology - Software Process Assessment (parts 1-9.
- ISO/IEC 15939:2000, Information Technology - Software Measurement Process.
- ISO/IEC 90003:2004, Software and Systems Engineering - Guidelines for the Application of ISO9001:2000 to Computer Software.

3.4 Web standards and protocols

Many work packages will develop human interface web servers and programmatic web services. This section describes standards which relate to this development.

3.4.1 Protocols description

3.4.1.1 HTTP and HTTPS

HTTP is the basic communication protocol of the web, and its latest version (HTTP/1.1) was defined in RFC 2616[14] in 1999. It is widely implemented in both servers and clients and is clearly the only standard which is relevant for web services, therefore details of the standard are not given here.

[14] http://tools.ietf.org/html/rfc2616

HTTPS, defined in RFC 2818[15] builds upon HTTP, adding a layer of encryption to secure data against eavesdropping and man-in-the-middle attacks during transmission. It relies upon trusted certification authorities (CAs), the most popular of which are typically included with browsers. Several attacks upon the security of HTTPS have been documented, but its ubiquity and integration with current web infrastructure make it the only obvious choice for securing data destined for web browsers. It is a less obvious choice for web services interfaces, because alternatives exist according to the web services protocol being used - see section on web services.

Some of the MyHealthAvatar platform services are to be deployed on cloud infrastructure. HTTPS in its most basic form supports only one certificate per port per IP address. Therefore the presence of multiple virtual servers sharing a port on a single real server prevents the use of HTTPS. Since most data transmitted in will be sensitive, HTTPS should always be used, rather than HTTP.

3.4.1.2 Web services protocols: SOAP and REST

Although HTTP was primarily designed to transmit content for human consumption, it has been re-purposed for communication between electronic devices. A key motivation for this was to allow RPC type communication to tunnel through firewalls, which are typically configured to allow HTTP traffic.

SOAP (Simple Object Access Protocol)[16] was the first popular approach, although its transport layer was not restricted to HTTP. Messages are structured using XML with a combination of a standard SOAP envelope schema and an application-specific schema. This is contained in a GET or POST HTTP request with a SOAP-specific content type. Many standards have built up around SOAP addressing application-specific needs. In particular, Web Services Security (WS-Security)[17] defines standards for signing and encrypting SOAP messages, using various common formats and algorithms, and implementations exist for most web services platforms.

REST (Representational State Transfer)[18] is technically and conceptually an architectural style, rather than an alternative protocol to SOAP. However, HTTP is the major architecture which conforms to the principles of REST (HTTP/1.1 and REST were developed in parallel), and when one speaks of a REST service, this typically refers to the use of HTTP and related web standards to implement RPC style communications. This is more precisely and correctly referred to as a RESTful web service. Technically, then, SOAP can be RESTful if it follows the design principles of REST, although they are usually spoken of as being distinct. Less philosophically, RESTful web services make more use of all the verbs defined in HTTP. SOAP services typically use just POST (and sometimes GET) methods when using HTTP as a container, giving details of the precise nature of the request in an application-specific manner within the SOAP envelope. REST services attempt to reveal some of the semantics of a request by making use of the full HTTP verb vocabulary, URIs to refer to resources upon which

[15] http://tools.ietf.org/html/rfc2818

[16] http://www.w3.org/TR/soap12-part1/

[17] http://www.ibm.com/developerworks/library/specification/ws-secure/

[18] http://www.ics.uci.edu/.fielding/pubs/dissertation/rest.arch.style.htm

verbs act, and response status codes. This allows standard web intermediaries (caches, proxies, tunnels, firewalls) to act upon messages more intelligently. REST is the current trend in web services, although there is little evidence that the stated advantages of REST are real (although there is a great deal of opinion on the matter). Usually the justification for using REST is that there's no good reason to not use it.

3.4.2 Markup languages and styling

HTML is the dominant standard for text and content markup on the web. Whilst HTML 4.01 is the latest official standard, HTML5 is currently in advanced stages of development and many new features in the latest draft specifications are supported by the latest web browsers. HTML5 adds many useful elements to HTML, including video, audio and canvas tags, and official inclusion of SVG (for vector graphics) and MathML (for expressing mathematical formulae) in the standard. There is an increased focus upon standardized APIs for scripting. Lack of support in older browsers means that using HTML5 at this stage may be premature, especially given legacy systems deployed in clinical environments and the slow pace of change in those environments. As with many web standards, use of the latest standards and facilities may need to pair with a less advanced but more widely supported backup solution.

HTML is intended as a markup language for interlinked hypertext documents. As such, elements are generally presentational rather than semantic in intent. HTML has been extended to allow the inclusion of non-document semantic information, mainly using the class attribute of most elements. This can be turned into presentational information using CSS. XML can be regarded as a purer semantic representation of information. It is a generalized markup language which is not specific to hypertext documents. Whilst most web browsers can display XML documents, the lack of any presentational information means it is typically in a very technical brows-able tree form which is not user-friendly. An XML document may be transformed into a more human-readable form with domain-specific browsers, or more generally using an XSLT description.

XHTML is a parallel standard to HTML, designed to mirror all of its functionality but using XML as its basis rather than the more flexible SGML. XML, and therefore XHTML, is somewhat stricter and more consistent on how tags, attributes and content are arranged, and ambiguous formatting is reduced. All XML documents are valid SGML documents, but not vice-versa. There are XHTML versions to mirror HTML 4.01 and HTML5. All browsers which support normal HTML also support the equivalent XHTML, since it is a subset of the standard. XHTML documents can be parsed with an XML parser, which is relatively simple compared to a general HTML parser which must deal with ambiguous markup.

3.4.3 Content styling and presentation languages

CSS (Cascading Style Sheets) describes the visual presentation of HTML documents. CSS descriptions can be included in the style attribute of an HTML tag, but this is generally regarded as bad practice. It is better to separate style from the content of an HTML document. The style for a set of HTML tags which match a certain pattern can be placed in the header of an HTML document, or in a separate CSS file which the document links to. A set of such patterns is termed a stylesheet, and several stylesheets can be associated with a given HTML document, which can be chosen according to user taste, accessibility requirements, or the presentation medium (for example, a standard computer screen, a small-format smartphone, a projector, or a hard-copy printer). CSS is mostly supported in all modern browsers, with a few omissions in the less-used functionally. For example, support for paged media (hard-copy) is often incomplete.

XSLT (Extensible Stylesheet Language Transformation) is a declarative XML-based language for transforming an XML document into some other form, most often HTML, but possibly any other textual format. Although it is a more general tool than CSS, its primary intent is to transform XML documents into a more user-friendly human-readable form. Most web browsers support XML transformation via XSLT on the client side. Additionally, an XSLT transformation may be applied server-side so that only HTML is served. XML in combination with XSLT therefore represents the most flexible and pure solution to marking up content semantically, but presenting it in a user-friendly manner. The key disadvantages are is that developers may be unfamiliar with this paradigm, XSLT is a fairly verbose and ugly language which is difficult to develop and debug, and it is not a commonly used paradigm and therefore there maybe issues regarding its robustness and future support.

3.5 Software standards examples

3.5.1 STANDARD WIDGET TOOLKIT (SWT)

SWT is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. To display GUI elements, the SWT implementation accesses the native GUI libraries of the operating system using JNI (Java Native Interface) in a manner that is similar to those programs written using operating system-specific APIs. Programs that call SWT are portable, but the implementation of the toolkit, despite part of it being written in Java, is unique for each platform. The toolkit[19] is licensed under the Eclipse Public License, an open source license approved by the Open Source Initiative.

3.5.2 SWT/XML

SWT/XML[20] is a lightweight XML markup language for describing Eclipse SWT / RCP user interfaces. It includes an Eclipse Web Tools based IDE editor plug-in. Its features include:

- Very readable and compact XML grammar for describing SWT user interfaces.
- Easy to learn and use. SWT/XML provides a 1:1 mapping from SWT widget classes to XML tags. If you know SWT, you just need to learn few general rules.
- I18N support using label properties and resource bundles (same as for plugin.xml files):

[19] http://www.eclipse.org/swt/

[20] https://code.google.com/p/swtxml/

- Eclipse IDE plug-in with Content Assist and Preview:
- Strict markup validation with meaningful exception messages.
- Easy integration in application, the runtime library of SWT/XML is very small (~ 170KB) and has almost no external dependencies (except for Eclipse RCP plug-ins).

3.5.3 JAVAFX (2D/3D)

JavaFX[21] is the next step in the evolution of Java as a rich client platform. It is designed to provide a lightweight, hardware-accelerated Java UI platform for enterprise business applications. With JavaFX, developers can preserve existing investments by reusing Java libraries in their applications. They can even access native system capabilities, or seamlessly connect to server-based middleware applications.

3.5.4 APACHE PIVOT

Apache Pivot[22] is an open-source platform for building installable Internet applications combining enhanced features for user interface building. It allows developers to easily construct visually-engaging, cross-platform, connected applications in Java or any other JVM language, such as JavaScript, Groovy, or Scala. Pivot is also the only truly open IIA framework: it is completely open source, and is driven entirely by the software development community.

3.5.5 WPF

Windows Presentation Foundation (or WPF)[23] is a Computer Software graphical subsystem for rendering user interfaces in Windows-based applications developed by Microsoft. WPF attempts to provide a consistent programming model for building applications and separates the user interface from business logic. It resembles similar XML-oriented object models, such as those implemented in XUL and SVG. WPF employs XAML, an XML-based language, to define and link various UI elements.

3.5.6 Other Web Application Frameworks

Other web application technologies are listed here for future reference.

[21] http://docs.oracle.com/javafx/

[22] http://pivot.apache.org/

[23] http://msdn.microsoft.com/en-us/library/ms754130.aspx

- HTML 5.0
- JQUERY
- FLASH
- GOOGLE WEB TOOLKIT (GWT)
- APACHE CLICK
- APACHE VELOCITY
- APACHE TAPESTRY
- APACHE WICKET

4 The envisaged role of MyHealtAvatar– the users and platform

4.1 Avatar interface overview

The interface design of MyHealthAvatar will need to look into user needs. As a result of ageing population, healthcare and associated social welfare costs are growing exponentially in recent years and they will soon become unsustainable unless we change the way in which people are supported. In many cases, there is a need to shift medical care from institutions to the home environment. To this end, ICT tools are being used to reform the traditional ways in which medical data are recorded, tested and analyzed, without in any way reducing its quality. MyHealthAvatar will make it possible to set up new interactions between doctors and patients to maintain the quality and intensity of treatment at a more sustainable cost. It complies with the trend to patient-centered healthcare and offers a pathway to enhance the self-awareness of citizens and to empower them to play more significant roles in taking care of their own health, which is regarded as an effective way of dealing with the increased challenges anticipated in future healthcare. Through the self-management of their own avatars, patients will be better informed about their treatment, condition, and on improved lifestyle, which will contribute to the awareness of their own health problems, and hence make the future healthcare system more efficient. It will also make it easier for patients to discover and talk with fellow patients suffering from similar diseases/conditions and to exchange experiences and hence raise their spirits collectively in the fight against the illness.

MyHealthAvatar will take advantage of a range of ICT advances, such as simulation models, semantics, visual analytics technology, web-based technologies. It can be seen as a user interface to allow clinicians and citizens to access modern ICT technologies for their healthcare. These may include:

- an interface to collect, store and manage individual citizen information
- an interface for data sharing and information exchange with other people
- an interface to access external resources and information
- an interface a range of clinically proven tools to support the display of clinical information and to assist in clinical analysis and decision making.
- a means to contribute data to biomedical and clinical research for new knowledge discovery.

From the perspective of user interfaces, different user interfaces will need to be designed to accommodate different user needs. Especially, interface with citizens will need to be simple and require little training.

4.2 Interface for citizens/patients

The interface design of MyHealthAvatar should allow self-management of individual citizens. It will allow collection and access to medical history and all the risk factors for the development of major diseases of individual patients, which will provide extremely useful information for healthcare.

MyHealthAvatar will need to be presented as a 4D health avatar, which is a unique interface that will allow data access, collection, sharing and analysis by utilizing modern ICT technology. It should function as an interface to provide:

- Long-term and consistent health status information of individual citizens.
- Different tools to allow valuable information retrieval for patients, health risk analysis and prediction, patient care and monitoring.

The user interface of the health avatar should allow effective data management and sharing by individual citizens. Through the user interface, the citizens will decide how the data is shared by stakeholders. The interface should facilitate highly self-motivated data management and user-centered data collection, supported by the necessary data integrity measures.

The user interface should allow users to log in their own accounts. The users will be able to input their information for healthcare, such as name, age, gender, contact, information of their GPs, etc. They will also be able to build links with their friends and followers to allow information sharing. The information need to be grouped (e.g. under different Tabs) to allow effective information management.

Information that can be included are given below (but not limited to):

- User image, gender, birthday, location
- Medical information, e.g. hospital admission (Reason, Duration, Discharge
- Friends and followers
- My health Status, including Patient Diary (using Daily questionnaire regarding "Physical, Mental and Social Functioning"), weight, medical condition, symptoms and treatments.

Users will be able to use the above system menu to:

- Exam their own data
- Find patients with similar condition, symptom and treatments
- Find out symptoms and treatment for their conditions by looking at other fellow patients
- Find out possible conditions for their symptoms by looking at other fellow patients
- Find out possible treatments for their conditions by looking at other fellow patients

4.3 Interface for clinicians

MyHealthAvatar will provide different tools to facilitate clinical data analysis and knowledge discovery for clinicians.

The user interface of the health avatar should support information analysis using integrated toolbox, which will be a vehicle by which medical professionals can augment their clinical knowledge with heterogeneous information from the avatar for clinical decision making and knowledge exploration. It will offer significant assistance to doctors by:

- displaying related information in a body-centric view around the avatar and by performing visually assisted data analysis (i.e. visual analytics) to extract clinically meaningful information from the heterogeneous data of individual/shared avatars, such as the patterns of symptoms, experience of treatments, medicines, self-care guidelines, risk factors etc.., supported by the computing resource that will be provided by the architecture.
- Allowing access of integrated predictive computer simulation models which foresee the growth of the disease and the effect of treatment. Data sharing will be encouraged, which will potentially provide to an extensive collection of population data to offer extremely valuable support to clinical research.

By functioning as a personalised metaphor, the 4D health avatar should bring explicit benefit that matches the initiative of VPH, including personalised, predictive, and integrative treatment. It also interfaces with modern technologies to bring down the cost of the current healthcare system by involving self-management and self-monitoring of patients. The avatar is supported by an infrastructure to maximise the yield of biomedical research expenditure. It supplies healthcare providers with ICT capacity in terms of integrating the patient information into a coherent entity, which will subsequently offer medical professionals and researchers an interface for the access of a large set of patient information through the sharing of the avatar data, and for blending information with extreme heterogeneity, including those from different data sources, different models, organ systems, space-time scales and modalities.

4.4 Data collection interface

Data collection is one of the keys to MyHealthAvatar. We should utilize the latest ICT technology for data gathering and information searching. Mobile phone techniques will be also used to collect the data from patients.

The Internet resource is an effective means of engaging users in terms of attracting their input. Web information extraction (IE or WI) should be used for information gathering from social network and other websites in a semi-automated way. In particular, we shall mainly focus on information extraction from the social media as a novel way for data collection. Exponentially increased amount of valuable information is buried in social networks nowadays owing to their ever growing popularity. There are rich evidences to show that social media has provided more valuable data than new. Connecting to the social network is also an ideal way to engage users who are willing to provide their information through the networks. Due to the constant user engagement, the information extracted from the social network is often more completed and up to date.

4.5 Platform interface

European healthcare systems have been subject to a long and complex history of independent evolution among many different countries. As a result, the picture is highly fragmented with differences between member states, regions, and even between hospitals within the same country. So, from the perspective of the individual patient, maintaining a clinical record in a consistent manner is difficult, and the problem is being exacerbated by the increased population movement within Europe. The number of European countries with a positive migration balance, meaning more people enter than leave the country, has grown over the last decades. In many cases, the size of net migration determines whether a country has population growth or is entering a stage of population decline. This situation poses as a threat to the provision of high quality healthcare services, and this is particularly true for the prediction and treatment of major and long-term diseases (e.g. cancer) where a consistent record of individual patients is of great importance. To this end, MyHealthAvatar offers a consistent interface for information collection, access, sharing and analysis, which has become a key to the problem that we are all facing in Europe.

4.6 Review of current avatar modelling and rendering technologies

4.6.1 WebGL

From the early days of the internet, Web-based 2D and 3D graphics attracted much of people's interest. Java started as applets in the browser to show interactive graphics. Today, Flash players have dominated most of the browsers on demonstrating interactive 2D graphics. VRML plugins had been popular in rendering 3D objects in web browsers. With the evolution of OpenGL and the advent of HTML5 and WebGL (Web Graphics Library) , the 3D graphics capability of web browsers has been unbridled. WebGL is a JavaScript API for rendering interactive 3D graphics in web browsers. WebGL elements can be mixed with other HTML elements and composited with other parts of the page.

Latest versions of Firefox, Chrome and Safari Desktop all supports WebGL while Internet Explorer will only support WebGL from version 11.

4.6.2 Three.js and SceneJS

With WebGL one can write OpenGL programs that render in a web page. However, from OpenGL 3.1, OpenGL no longer supports backward compatibility, which implies that the old programming paradigms in OpenGL 1.x and 2.x are no longer favoured and WebGL users need to write shaders by themselves. This is inconvenient for some programmers who know little about OpenGL shaders.

Three.js is an open-source javascript library based on WebGL designed to fill the gap by simulating the old OpenGL style programming paradigms such as lighting, material, camera, etc. It is easier to use and doesn't require knowledge on shaders.

SceneJS , another open-source javascript library based on WebGL, provides a JSON-based scene graph API and supports scenegraph management. It was created for efficient rendering of large numbers of objects.

4.6.3 Web-based Human Body Visualisation

There are two major 3D human body websites, one is Zygote Body and the other is BioDigital Human . Both of them provide interactive visualistion of the whole body male and female anatomy. The model quality of Zygote Body is better than BioDigital Human . Zygote Body shows the whole body after model loading while BioDigital Human displays a skeleton initially and let the user select other parts to show. The memory management of Zygote Body also outperforms BioDigital Human. The latter also provides some special visualisations of conditions, nevertheless, none of them aims at health data visualisation.

The user interface of Zygote Body and BioDigital Human re shown in Figure. 1 and Figure. 2 respectively. BioDigital Human uses SceneJS as its WebGL engine.

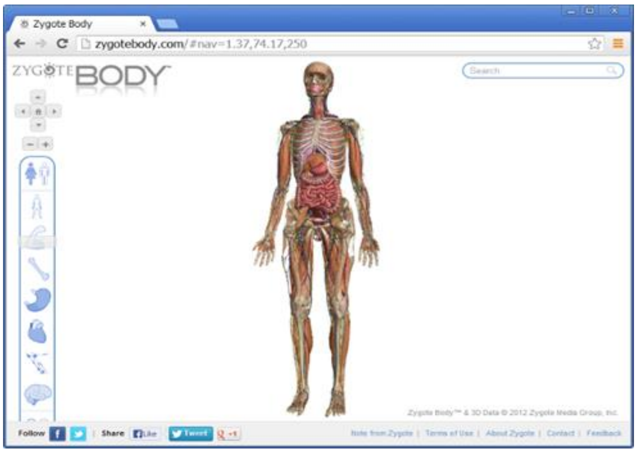

Zygote Body
zygotebody.com/#nav=1.37,74.17,250
ZYGOTE BODY
Search
Zygote Body™ & 3D Data © 2012 Zygote Media Group, Inc.
Follow
Share
Like
Tweet
+1
Note from Zygote | Terms of Use | About Zygote | Contact | Feedback

Figure 1: The user interface of Zygote Body

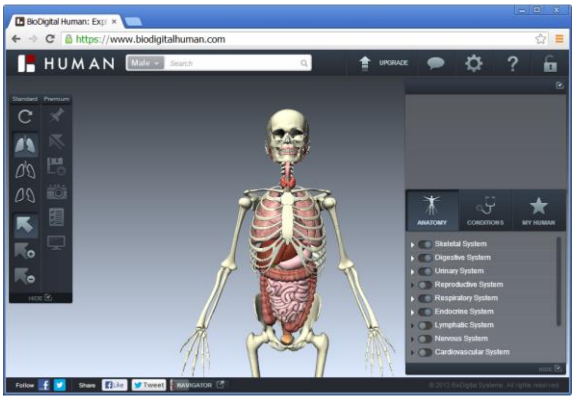


Figure 2: The user interface of BioDigital Human

4.7 Web based avatar framework

In the MyHealthAvatar project, 4D avatars are represented in a Web-based framework, in which online interfaces are provided to support the collection of, and access to, the complete medical information relating to individual citizen's longitudinal health status, gathered from both internal and external sources.

Figure 3 shows the overall avatar framework. A 4D avatar is formed by two key parts: personalised 3D body models and health related citizen's longitudinal information. Avatar data collected from various sources are stored in Cassandra big data repositories to facilitate fast query and data analytics. The relationships and schemas among the data are stored into semantic RDF repository to provide further semantic reasoning. A main Web based user interface along with several mobile apps are provided for accessing the 3D model, collecting, updating and managing user specific health data.

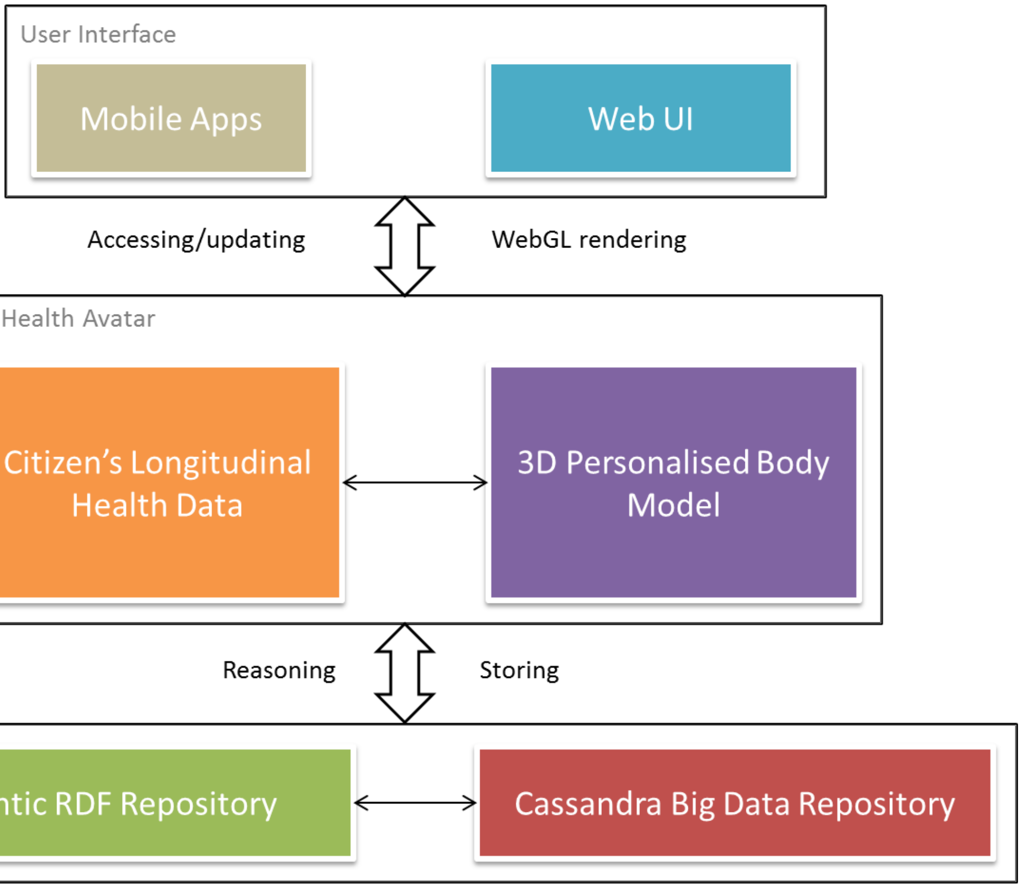


Figure 3: An overview of the Web Avatar framework.

4.8 Visual Analytics

Visual Analytics (VA) is an important way to graphically present data and information within MyHealthAvatar at desired level of details. It combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data set. As a major multi-disciplinary field, VA includes the science of analytical reasoning, interactive graphical representations and interaction techniques, data representations and transformations, production, presentation and dissemination.

VA plays an important role in the interaction between users and the avatars, allowing information comprehension and data analysis. Automatic data analysis has achieved great success in the last few decades. Visualization is effective in terms of helping domain experts to understand data by offering crucial visual information. However, due to the rapid evolution of data complexity, existing simulation and fully automatic data analysis is often not able to reach an optimal solution. It is found that a more effective approach can be to involve the power and versatility of human decision making by integrating simulation and data analysis with interactive visualization.

4.8.1 Information visualization

A critical challenge in the representation of MyHealthAvatar is the large data set, as well as the level of complexity involved within the clinical cases. More specifically:

- Management of large data and models
- Visually controlled data mining to support the sound integration of user interaction for effective data mining.
- New interaction techniques to support effective model/data exploration through user interaction, e.g. eye tracking, human gesture, touch screen etc. Special attention could be paid to interaction techniques for collaborative and distributed working environments.
- Evaluation techniques to test the results of the above techniques.

Visualization techniques that are particularly suitable for the exploration of data and model relationships within MyHealthAvatar include the interactive visualization of very large models and data, multiscale visualization (in both spatial and temporal domains), visualization of uncertainty within models and data, and collaborative visualization.

Visualization of high dimensional datasets has been of research interest for many years in information visualization. However, the problem still remains largely unsolved and a convincing way to represent high dimensional information is still lacking.

Visualization of multiscale data set has achieved great success in many areas such as Google Map, human neural systems, etc. However, such a demand also exists at multiple scales in the biomedical area, and suitable techniques have not been fully elaborated.

4.8.2 Visualization of avatar models

The avatar should include a full set of anatomical parts of human body, including (not limited to):

- Body Shape
- Skeletal System
- Muscular System
- Ligament System
- Integumentary System
- Digestive System
- Respiratory System
- Immune System
- Nervous System
- Circulatory System
- Lymphatic System
- Endocrine System
- Vestibular System
- Urinary System
- Reproductive System

A range of manipulations will be allowed to the avatar models, such as

- Reset Avatar: (Reset function to bring the view of the 3D avatar back to its original state)
- Standard:(Add body object and search for relative information)
- Transparency: (view through the transparent 3D avatar body the selected object)
- Extract: extract the selected object from the 3D body
- Select: Tool which will helps to select individual anatomy object and provides associated information (e.g. by mining web information)
- Multi Select: Enables more than one anatomy object selections
- Hide/Reveal: Hide or reveal underlying structures
- Note: Label body parts and assign custom description
- Cross Section Tool: slice the Human anatomy by imaging planes
- Save: Save the view of the screen, share it to the public
- Picture: Take a picture of the 3D avatar as it appears on the screen and share it with the public

The avatar must be able to rotate, zoom in/out. From a technical point of view, the avatar must be a light application in order to permit quick upload upon each refresh of the page. User experience is expected to be invariant across different device type or navigation browser.

5 Data exchange and interoperability

5.1 Interoperability

The report "Semantic Interoperability for Better Health and Safer Healthcare"[24] [103] produced by the FP6 Semantic HEALTH project provides a number of relevant definitions, standards, and application domains for semantic interoperability. SemanticHEALTH developed a longer-term research and deployment roadmap for semantic interoperability. Its vision is to identify key steps towards realizing semantic interoperability across the whole health value system, thereby focusing on the needs of patient care, biomedical and clinical research as well as of public health through the re-use of primary health data.

The Recommendation, COM(2008)3282[25] provides the following definition: "Semantic interoperability means ensuring that the precise meaning of exchanged information is understandable by any other system or application not initially developed for this purpose", whereas "interoperability of electronic health record systems means the ability of two or more electronic health record systems to exchange both computer interpretable data and human interpretable information and knowledge", (page 14). In the context defined above, semantic interoperability (SIOp) addresses issues of how to best facilitate the coding, transmission and use of meaning across seamless health services, between providers, patients, citizens and authorities, research and training. Its geographic scope ranges from local interoperability (within, for example, hospitals or hospital networks) to regional, national and cross border interoperability. The information transferred may be at the level of individual patients, but also aggregated information for quality assurance, policy, remuneration, or research.

Interoperability is split according to the Semantic Health report in following 4 incremental levels:

- Level 0: no interoperability at all
- Level 1: technical and syntactical interoperability (no semantic interoperability)
- Level 2: two orthogonal levels of partial semantic interoperability
- Level 2a: unidirectional semantic interoperability
- Level 2b: bidirectional semantic interoperability of meaningful fragments
- Level 3: full semantic interoperability, sharable context, seamless co-operability

The investigations undertaken by SemanticHEALTH suggest that full semantic interoperability (Level 3) is required in order to take full advantage of computerized medical records. It is however also recognized that due to steep investments needed, the highest level of semantic interoperability should only be sought in specific areas with the high potential for improvements, while in other areas a lower interoperability level may suffice.

[24] http://ec.europa.eu/information.society/activities/health/docs/publications/2009/2009semantic-health-report.pdf
[25] http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32008H0594:EN:NOTEC

In order to achieve complete technical and semantic interoperability, existing standards have to be harmonized and bridged. In the US major consortia have been formed to provide semantic interoperability between the standards (e.g. BRIDG covering CDISC, HL7 (RCRIM), FDA, NCI) and to provide core sets of data collection fields (CDISC CDASH). Furthermore, efforts exploring the potential to improve interoperability between the electronic health record and electronic data capture (e.g. CDISC eSDI, eClinical Forum/PhRMA EDC/eSource Taskforce) have been initiated.

5.1.1 Generic protocols for web services and network communication

The World Wide Web Consortium (W3C) produces protocols and standards for Web technologies. OWL-based Web Service Ontology (OWL-S) is an OWL based Ontology, within the OWL-based framework of the Semantic Web, for describing Web services. OWL-S ontology is also sometime considered as a language for describing services, reflecting the fact that it provides a standard vocabulary that can be used together with the other aspects of the OWL description language to create service descriptions.

Web Service Modeling Ontology (WSMO) is an ontology for semantically describing Semantic Web Services. It is a model for the description of semantic web services that tries to overcome the limit of the existing technologies for the service description; in particular OWL-S. Web Service Modeling Language (WSML) is a language that formalizes the WSMO. It uses well-known logic formalisms, namely, Description Logics, First-Order Logic and Logic Programming, in order to enable the description of various aspects related to Semantic Web Services. Web Service Semantics - WSDL-S specification is a W3C Member Submission that defines how to add semantic information to Web Services Description Language (WSDL) documents. WSDL is a XML based language that is used to describe the functionalities of a web service. WSDL-S tries to overcome the lack of semantics in WSDL by adding new extensibility elements to the WSDL standards to annotate the semantic of web services. Each service description refers one or more Semantic Model. A semantic model captures the terms and concepts used to describe and represent an area of knowledge or some part of the world, including a software system. A semantic model usually includes concepts in the domain of interest, relationships among them, their properties, and their values. WSDL-S provide mechanisms to annotate the service and its inputs, outputs and operations. Additionally, it provides mechanisms to specify and annotate preconditions and effects of Web Services. These preconditions and effects together with the semantic annotations of inputs and outputs can enable automation of the process of service discovery.

Semantic Annotation for WSDL (SAWSDL) defines how to add semantic annotations to various parts of a WSDL document such as input and output message structures, interfaces and operations. The extension attributes defined in this specification fit within the WSDL 2.0 extensibility framework. It provides mechanisms by which concepts from the semantic models that are defined either within or outside the WSDL document can be referenced from within WSDL components as annotations. The annotations on schema types can be used during Web service discovery and composition. In addition, SAWSDL defines an annotation mechanism for specifying the data mapping of XML Schema types to and from an ontology; such mappings could be used during invocation, particularly when mediation is required.

5.1.2 Medical Document Classification

One of the main techniques of information extraction applied to medical documents is classification. Classification of documents based on free-text can be performed using text categorization methods from Information retrieval or machine learning techniques. Efforts have been made to automatically classify the radiology reports based on their content.

Wilcox et al[26], assigned 6 different clinical conditions to narrative x-ray reports using different machine learning techniques. Patients' cancer stage has been inferred by classification of histology reports using statistical machine learning methods[27]. Comparison of IR, ML and rule based approaches for ICD-9 CM code assignment to radiology reports carried out by Goldstein et al[28]. Also the role of domain knowledge in automatic classification of medical reports is investigated[29]. In addition to classifying the medical documents as a whole, machine learning techniques has been applied to classify a part of the documents like the Medical Abstracts sentences[30] or the assertions of findings in medical reports[31]. The effect of feature representation on classification of medical documents has been investigated by representing the document in Bag of Words and Bag of Phrase format[32]. An example of a common information extraction tool for biomedical domain is MedLEE. It is a natural language processing tool for radiology reports which uses a medical vocabulary for mapping words and phrases to standard medical terms while also adding some modifier information like body location, region, certainty, etc. to them. It works based on a sentence at a time and does not capture the context. MedLEE is in routine use for encoding radiology reports, and its ability to identify medical findings in radiology reports has been investigated in several studies[33]. Also, the encoded output of Medlee has been used in many studies. For example, Medlee was used to encode radiology reports for creating feature vectors[17] or structured output generated by Medlee consisting of findings and modifiers has been used to automatically assign Unified Medical Language System (UMLS) codes to a clinical document[34].

[26] Classification Algorithms Applied to Narrative Reports. Adam Wilcox, George Hripcsak. 1999. AMIA

[27] Classification of Cancer Stages from Histology Reports. Iain McCowan, Darren Moore, Mary-Jane Fry. 2006. IEEE.

[28] Three Approaches to Automatic Assignment of ICD-9-CM Codes to Radiology Reports. Ira Goldstein, Anna Arzumtsyan,Ozlem Uzuner. s.l.: AMIA, 2007

[29] The Role of Domain Knowledge in Automating Medical Text Report Classification. Adam Wilcox, George Hripcsak. s.l. : JAMIA, 2003, JAMIA.

[30] Categorization of Sentence Types in Medical Abstracts. Larry McKnight, Padmini Srinivasan. 2003. AMIA. p. 440

[31] Machine learning and Rule-based Approaches to Assertion Classification. Ozlem Uzuner,Xiaoran Zhang,Tawanda Sibanda. s.l.: JAMIA, 2009.

[32] The Effect of Feature Representation on MEDLINE Document Classification. Meliha Yetisgen, Wanda Pratt. 2005. AMIA. pp. 849-853.

[33] Identification of Findings Suspicious for Breast Cancer Based on Natural Language Processing of Mammogram Reports. Nilesh L.Jain, Carol Friedman. 1997. AMIA. pp. 829-833.

[34] Automated Encoding of Clinical Documents Based on Natural Language Processing. Carol Friedman, Lyudmila Shagina,Yves Lussier,George Hripcsak. 2004, JAMIA, pp. 392-402.

5.1.3 Clinical Decision Support Tools

In Kawamoto et al[35], efforts and issues concerning use of standards in the CDS area to enable interoperability are discussed. The paper argues that a critical need for enabling CDS capabilities on a much larger scale is the development and adoption of standards that enable current and emerging CDS resources to be more effectively leveraged across multiple applications and care settings. Standards required for such effective scaling of CDS include:

1. Standard terminologies and information models to represent and communicate about health care data.
2. Standard approaches to representing clinical knowledge in both human-readable and machine-executable formats.
3. Standard approaches for leveraging these knowledge resources to provide CDS capabilities across various applications and care settings.

Item 1 above has been addressed in other sections of this report and is not specific to CDS, but refers to more general requirements for clinical data representation and exchange and for semantic interoperability.

Several languages exist which could be applied for representing and sharing clinical knowledge in a formalized way (executable clinical knowledge according to the above classification), such as Arden syntax[36], Gello (M. Sordo et al. Software specifications for gello: An object-oriented query and expression language for clinical decision support), ERGO (S. Tu et al. Ergo: A template based expression language for encoding eligibility criteria). The Arden Syntax is a standardized language used to represent and share clinical knowledge through Medical Logic Modules (MLMs). The Arden Syntax was introduced in 1989 and was first adopted in 1992 by ASTM; it is not a full-fledged programming language, it is meant to be written and maintained by clinicians. Version 2.0 was adopted in 1999 by both its current sponsor, HL7, and ANSI version 2.1 is the current standard. The Guideline Elements Model (GEM) is an example of standard for non-executable representation of clinical knowledge. It is an ASTM standard and is used for representation of the contents of clinical practice guidelines in a structured way. GEM II includes and XML Schema that defines a structured format for extracting relevant content out of a clinical practice guideline, an object-oriented data model and an editor for GEM guidelines. While several, mainly HL7-supported, standardization efforts emerge that address the use of executable clinical knowledge and the CDS delivery, their adoption is low and significant gaps still exist.

[35] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3097480/pdf/TOMINFOJ-4-235.pdf
[36] http://www.hl7.org/special/Committees/arden/index.cfm

5.1.4 Semantic Web Standards

5.1.4.1 Resource Description Framework (RDF)

The Resource Description Framework[37] or RDF, is a data model developed by the World Wide Web Consortium (W3C) for representing semi-structured information in the Web. It enables users to describe Web content with arbitrary vocabularies and associate semantics to the data through the definition of metadata. It has since grown into a general data model and is very widely used outside the realm of web resource description. The RDF data model represents information in terms of statements, commonly known as RDF triples. Each statement in the model describes a resource (the triple's subject) by specifying a property (the triple's predicate) and its value (the triple's object). A triple can also be seen as a directed graph with a subject and object being represented as nodes and the predicate encoded as an arc between the two nodes. A set of statements describing a set of resources can be seen as an RDF graph and can optionally be associated with a name (URI) to uniquely identify them (the so-called Named Graphs). RDF is a model rather than a format: graphs can be serialized using multiple notations. One of the most common ones is RDF/XML that serializes an RDF graph in XML, described in RFC 3870. Other, more efficient and less verbose, notations also exist such as N3, N-Triples and Turtle. All formats have their benefits and are easily supported for both reading and writing. Since its conception RDF has become the de facto standard for representing semi-structured information in the Web in machine-processable form and has become the foundation of the Semantic Web. RDF as a data exchange format also enables interoperability between applications.

5.1.4.2 Resource Description Framework Schema (RDFS)

RDF does not make any assumption for the vocabulary used for describing resources nor does it assumes or define the semantics of a given domain. In order to do that, users need to define their own vocabularies using a vocabulary description language. RDFS[38] is a W3C recommendation of a basic ontology language for defining vocabularies. It enables users to define concepts (classes of resources) and properties that can be used for describing the semantics of a given application domain. Classes and properties defined in RDFS can then be used to describe an application domain with RDF. In addition, RDFS provide a semantics for specialization/generalization of classes and properties.

5.1.4.3 Ontology Web Language (OWL)

OWL (also known as OWL 1.0)[39] is a Web ontology language developed by the Web Ontology working group of the W3C. It has been a W3C Recommendation since 2004. It enables users to

[37] Manola F, Miller E. RDF Primer. W3C Recommendation 10 February 2004.http://www.w3.org/TR/rdf-primer/

[38] Brickley D, Guha R.V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-schema/

[39] Dean M, Schreiber, G. OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-ref/

specify ontologies by defining classes of objects, properties and relations between classes. It also allows for the specification of individual objects in the domain of discourse. OWL has a higher expressiveness than that of RDFS. Three fragments or OWL dialects exists. The first is OWL-Lite, the least expressive fragment of the three. OWL Lite is meant for users in need of classification hierarchies and simple constraints. The second fragment is OWL-DL, a more expressive language with good computational properties. Finally, OWL-Full is the most expressive fragment of the three although it provides no guarantees with respect to its computational properties. Other variations of OWL also exist including OWL Horst and OWL DLP. Many ontologies, including biomedical ontologies, are expressed using some variant of OWL.

5.1.4.4 Ontology Web Language 2 (OWL 2)

OWL 2[40] is a revision and extension of OWL 1.0. It includes three versions to accommodate the needs of different type of users. OWL 2 EL is a subset of OWL 2 for which the basic reasoning services can be solved in polynomial time. It is particularly useful for expressing ontologies with large number of classes and/or properties. Many biomedical ontologies can be expressed with OWL 2 EL and many efficient reasoning systems for this language have been developed in recent years. OWL-QL is tailored for applications that use large amount of instance data and whose main reasoning service is query answering. OWL 2 RL is a profile aimed at applications that require scalable reasoning without losing too much expressive power. Reasoning in OWL 2 RL can be implemented using rule-based approaches.

5.1.4.5 RDF Query Languages (SPARQL)

SPARQL (SPARQL Protocol and RDF Query Language)[41] is a W3C Recommendation of a query language for RDF. It is intended to replicate the functionality of SQL as used in conjunction with a relational database. It is based upon describing multiple templates for sub-graphs, set operations upon the sets of matching sub-graphs, and selecting which nodes or edges within the matches are returned as results. It is a mature and widely supported querying language, and is being actively extended and refined to replicate some of SQL's more advanced functionality. Since its inception it has become one of the most widely used query languages for semi-structured data expressed in RDF and it is nowadays a Semantic Web standard query language. Other RDF querying languages exist, but they are usually specific to a particular RDF store or interface implementation. Virtually all RDF repositories nowadays provide an implementation of the language and many data repositories in the Web provide query services backed by SPARQL, the so-called SPARQL endpoints.

[40] Motik B, Patel-Schneider P.F, Parsia B. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation 27 October 2009. http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/

[41] Prud'hommeaux E, Seaborne A. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. http://www.w3.org/TR/rdf-sparql-query/

5.1.4.6 Assembly of workflows

A scientific workflow is a series of connected steps that describes a sequence of operations with regards to computational or data related tasks using a scientific workflow management systems as well as general business process management tools.

5.1.5 Triple stores

A triple store is a system capable of storing and querying RDF data. Triple stores are widely used because they allow to track easier provenance information than relational databases, it is easier to be accessed and linked to other data sources and provide more flexibility in querying and extending. In recent years, numerous triple stores have been developed. Based on their architecture, they usually fall into three main categories, In-memory, Native, Non-native Non-memory [RUS] [DIN03]. In-memory systems store RDF data in main memory. Native triple stores provide persistent storage with their own implementation of the databases while non-native, non-memory have been built on top of existing commercial relational database engines such as MySQL, PostsgreSQL and Oracle. From an architectural point of view, the native triple stores seem to be superior in relation to in-memory and non-native non-memory triple stores. The in-memory approach has scalability issues, when dealing with large volumes of data. The non-native non-memory approach suffers from the difficulty of querying efficiently the RDF data graph via SQL. On the other hand the native approach seems to have the advantage in performance due to its ability to be optimized for RDF Data.

Below you can find a brief description of the prominent open-source triple stores in use today ([HAS11] [TRI],[TRIP],[COM]). The purpose of this comparison is to highlight shortly the advantages and the drawbacks of each approach.

- **Virtuoso**
  Virtuoso, is a native triple store provided by OpenLink Software. It is available in both open source and commercial licenses and provides command line loaders, a connection API, and support for SPARQL and web server to perform SPARQL queries and uploading of data over HTTP. Virtuoso is scalable to the region of 15.4 Billion triples. In addition to this, it provides bridges to be used with other RDF Data frameworks such as Jena and Sesame.

- **Apache Jena**
  Jena is an open source java framework for building semantic web applications. Jena implements APIs for dealing with Semantic Web building blocks such as RDF and OWL. Jena SDB provides scalable storage and query of RDF datasets using conventional SQL databases such as PostgreSQL, MySQL, Oracle, MS SQL Server, HSQLDB and Apache Derby. Jena SDB scales up to 650 Million triples. Jena TDB is a native persistent graph storage layer for Jena. Jena TDB works with the Jena SPARQL query engine (ARQ) to provide complete SPARQL support. Jena TDB scales up to 1.7 Billion triples.

- **Mulgara**

Mulgara is an open source RDF triple store written in Java. Mulgara instances can be queried via the iTQL query language and the SPARQL query language. Mulgara scales up to 500 Million triples. Mulgara is not based on a relational database due to the large numbers of table joins encountered by relational systems when dealing with metadata. Instead, Mulgara is a new database that stores metadata in the form of short subject-predicate-object statements, much like the W3C's Resource Description Framework (RDF) standard. Metadata may be imported into or exported from Mulgara in RDF or Notation 3 form.

- **Sesame**

Sesame is an open source framework for storage, and querying of RDF data. Sesame matches the features of Jena with the availability of a connection API, inferencing support, availability of a web server and SPARQL endpoint. Like Jena SDB it provides support for multiple backends like MySQL and Postsgresql. Sesame Native is the native triple store offering from Sesame which has been tested with upto approximately 70 Million triples.

- **BigData**

Bigdata is a scale-out storage and computing fabric for ordered data (B+Trees). Scale-out is achieved via dynamic key-range partitioning of the B+Tree indices. Index partitions are split (or joined) based on partition size on disk and moved across data services on a cluster based on server load. The entire system is designed to run on commodity hardware, and additional scale can be achieved by simply plugging in more data services dynamically at runtime, which will self-register with the centralized service manager and start managing data automatically. Much like Google's BigTable, there is no theoretical maximum scale. The bigdata RDF store is an application written on top of the bigdata core. The Bigdata RDF store is fully persistent, Sesame 2 compliant, supports SPARQL, and supports RDFS and limited OWL inference. The single-host RDF database is stable and is used at the core of an open-source harvesting system for the intelligence community.

Other prominent proprietary closed source triple stores are Allegro Graph that is able to scale up to 1 Trillion triplets and supports SPARQL and built in SPARQL endpoint, OWLIM that can scale up to 12 Billion triplets and can also support SPARQ, built in SPARQL endpoint, RDFS and OWL.

5.1.6 Data Reasoning Services

One of the pillars in the technology stack that is required to help fulfill the Semantic Web vision is ability to be able to use the knowledge of a given domain, which is formally described by ontologies, to infer implicitly stated knowledge from the explicitly represented information. In the Semantic Web jargon this process is referred to as *semantic reasoning*.

Over the past several years several approaches, techniques and tools have been proposed for efficient reasoning over ontological data expressed in various ontology representation languages. Implementation of reasoners can be classified according to the employed mechanisms and the language used. Pellet [SIR07], FACT++ [HOR06] and RacerPro [MOL03] are description logic reasoners that implement tableau algorithms. Rule-based reasoners such as Bossam [MIN04] and OWLIM [KIR05], use rule engines for inferencing. On the other hand KAON2 [MOT05] can be classified as datalog driven since it reduces a DL knowledge base to a disjunctive datalog program. As far as resource-constrained reasoning is concerned, μOR [SAF09] is a micro OWL DL Reasoning system that supports a subset of OWL-Lite. It consumes much less memory resources than its competitors. Pocket KRHyper [KLE05], targeted for mobile devices, is a 1st Order Logic (FOL) theorem prover and model generator. However it does not support direct DL reasoning. Bellow we analyze in more detail the three most widely used ones.

5.1.6.1 Pellet

Pellet is an OWL reasoner developed by Clark & Parsia[42] that provides reasoning services for OWL ontologies with support for the latest revision of the Ontology Web Language, OWL 2.0. It is distributed under the terms of the AGPL v3 license for open source applications and under alternative license terms for proprietary, commercial closed-source applications. Pellet is regarded as the first reasoner to support OWL-DL and over the years it has become the de-facto standard for implementing applications that require reasoning over OWL ontologies.

In terms of usability Pellet is implemented in Java and its reasoning services can be accessed through its own Java API or by using one of the many bindings to common programming toolkits such as Jena and the OWL API[43]. Pellet has also been integrated in the Protege ontology editor[44]. Additionally, Pellet implements the DIG interface which allows users to access the reasoner's services through HTTP requests.

Reasoning in Pellet is based on *tableau methods*. In order to the derive the truth value of a logical formula tableaux-based methods try to build a model of the formula by exploiting its structure and by applying a series expansion rules until no more rules can be applied or, a contradiction is found. Pellet is able to reason over OWL-DL ontologies, a syntactic variant of the DL $\mathcal{SHOIN}(\mathcal{D})$.

As an OWL-DL reasoner Pellet implements all the basic DL reasoning services including *consistency checking*, *concept satisfiability*, *classification* and *realization*. Consistency checking, the process of checking whether a (set of) logical formulae is consistent, is the basic reasoning service upon which all other services are implemented. In addition, Pellet supports other non-standard services that have been identified as important for practical applications. One such service is the explanation and debugging of ontologies. The current version of Pellet (as of the time of writing this report) supports also reasoning over OWL 2 EL.

Pellet also provides support for *non-monotonic* reasoning, a form of reasoning that is capable of capturing several forms of common-sense and database reasoning, through the implementation of a

[42] http://clarkparsia.com/pellet/

[43] http://owlapi.sourceforge.net/

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

non-monotonic language called $\mathcal{ALCK}$. This feature allows Pellet's users to "turn on" the closed world assumption on demand at query time by means of an extension to the SPARQL language. Knowledge bases can also include, in a restricted form, a non-monotonic rule that makes use of the $\mathcal{K}$ operator in $\mathcal{ALCK}$ to represent non-monotonic information. Using this feature Pellet should be able to treat knowledge bases as database-like repositories.

Moreover, Pellet provides limited support for *rule-based reasoning*. This is achieved by implementing a decidable fragment of SWRL (Semantic Web Rule Language (17)), called $\mathcal{AL}$-*Log* that combines Datalog rules with DL knowledge bases allowing DL concepts to be used in the body of the rules. Pellet also provides support for reasoning with standard and user-defined XML Schema-based data types, nominals and over individuals. Another feature of Pellet is that it supports reasoning with individuals (ABox reasoning) by answering conjuctive queries over assertions. Those queries can be issued in any of the supported languages such as SPARQL, RDQL and KIF.

Finally an important non-standard service of Pellet is the ontology analysis and repair, known also as ontology debugging. Using this technique Pellet is able to identify or pinpoint the source of inconsistencies in an ontology and extract that part from the ontology. This service is important for the design, debugging and evolution of ontologies.

5.1.6.2 Racer Pro

RACER stands for Renamed *ABox and Concept Expression Reasoner* and is a reasoner with support for terminological and assertional reasoning over knowledge bases specified in the DL $\mathcal{SHIQ}(\mathcal{D})$, i.e. $\mathcal{SHIQ}$ with concrete domains, extended with simple data types. More specifically, RACER supports OWL Lite and OWL DL. In addition, RACER implements a decision procedure for satisfiability in Modal Logics. It was the first DL reasoning system to support a very expressive logic. RACER PRO is the commercial version of RACER. The system is maintained and released by Racer Systems GmbH & Co. KG [45] and can be accessed programmatically through Java, C and C++ APIs and, through TCP/IP.

Reasoning in RACER (Pro) is based also on tableau-methods that implement the reasoner's core reasoning service, namely Abox consistency. In addition to the standard reasoning services provided by state-of-the-art DL systems RACER implements a series of non-standard services that have been identified as important in many practical applications. These include services to *retrieve the list of concepts and individuals* in the knowledge base, *retrieval of the set of roles and sub-roles*, etc. The full list of services can be found at the reasoner's website[46].

A distinctive feature of Racer is its support for *reasoning over multiple Tboxes and Aboxes*. RACER also supports knowledge base management activities by providing services to add and retract

[45] http://www.racer-systems.com

[46] http://www.racer-systems.com/products/racerpro/index.phtml

axioms from (possibly) multiple Tboxes and Aboxes. Moreover, RACER provides support for algebraic reasoning including *concrete domains over integers*, *min/max cardinality restrictions over integers* and *(in)equalities over strings*.

The architecture of Racer reflects the advances in optimization techniques for DL systems. This is manifested in the fact that Racer implements a series of optimizations for improving the performance and efficiency of reasoning.

In addition to being seen as a Description Logics reasoner RACER can also be seen as a Semantic Web reasoner. As such RACER *supports reasoning over OWL Lite and OWL DL ontologies* with approximations for nominals in class expressions. RACER is also *capable of reasoning over certain extensions of OWL* such as OWL-E and, of handling rules with its implementation of SWRL (17). From this point of view RACER allows for:

- Checking the consistency of OWL ontologies.
- Computing and querying the specialization/generalization hierarchy induced by the declarations in the ontology.
- Finding synonyms for resources 9both for classes and instances).
- Retrieving extensional information by means of OWL-QL.
- Information retrieval based on incremental query answering.
- Accessing reasoning services using the DIG interface through HTTP.

A limitation of the system, however, is the *lack of support for user-defined XML data types*. Moreover, RACER support queries written in nRQL (*new Racer Query Language*), which supports *negation as failure*, *numeric constraints wrt. attribute values of different individuals*, *substring properties between string attributes*, etc. RACER's implementation of nRQL is the basis for implementing many of the features of OWL-QL.

5.1.6.3 FACT++

FaCT++[47] is a *tableau-based* reasoner that supports reasoning over $\mathcal{SHOIQ}$ DL knowledge bases and, in its latest version provides limited reasoning support for OWL 2 ($\mathcal{SROIQ}$). In addition, FaCT++ provides support for reasoning with XML Schema data types although it lacks support for built-in primitive types. It was originally designed as a reasoning platform for experimenting with novel tableau methods and optimization techniques and for reasoning over ontologies that use inverse roles.

It is an open source project distributed under GNU LGPL. It is available as a Protege plug-in and its services can be accessed through the DIG interface and the OWL API. The core reasoning service is KB satisfiability, so the core component of the reasoner is a satisfiability checker. Every other reasoning service is reduced to this. The basic workflow of the reasoner loads an ontology into the

[47] http://owl.man.ac.uk/factplusplus/

knowledge base and then classifies the ontology using the satisfiability component for deciding subsumption between pairs of concepts.

The reasoner applies several optimization techniques for enabling efficient reasoning over ontologies. Many of the optimizations are well-known in the DL community and implemented by several DL reasoning systems. In addition to incorporating these techniques FaCT++ implements novel optimization techniques and heuristics. These are applied at different stages of the reasoning and ontology management process and include optimization strategies used while loading data to the reasoner, strategies for optimizing satisfiability checking and those used in the classification task geared towards reducing the number of subsumption tests.

Although FaCT++ is able to reason over OWL 2 ontologies ($\mathcal{SROIQ}$ DL) its reasoning capabilities are limited. Specifically, FaCT++ cannot properly handle Top/Bottom Object and Data property semantics and has partial data type support; the only supported data types are *literal*, *string*, *anyURI*, *boolean*, *float*, *double*, *integer*, *int*, *dateTime* and *nonNegativeInteger*.

5.1.7 Linked Life Data

Linked Life Data[48] (LDD) is a data-as-service platform that provides access to 25 public biomedical databases through a single access point. The service accepts complex queries and can answer complex questions related to bioinformatics. The service can be offered either through public endpoints or through premium endpoints that are commercial and provide access to more mature applications with extra features.

LDD uses a distributed graph data model to represent complex heterogeneous offered as linked data. Figure 4 presents an example of how the data are interlinked using URIs in LDD. LDD is actually a RFD warehouse, it uses OWLIM for storage and offers 10B+RDF statements describing 1,5B+ resources integrating 32 biomedical data sources. The service covers the full path of data - gene, protein, molecular interaction, pathway, target, drug, disease and clinical trial related information. These are the primary entities of a knowledge base composed by structured databases (NCBI Gene, Uniprot, DrugBank, BioPAX and many more), terminologies (UMLS, OBO), and semi-structured documents (Pubmed, ClinicalTrials.gov). The public service integrates only free databases whereas in Enterprise edition further options are offered.

[48] http://linkedlifedata.com/

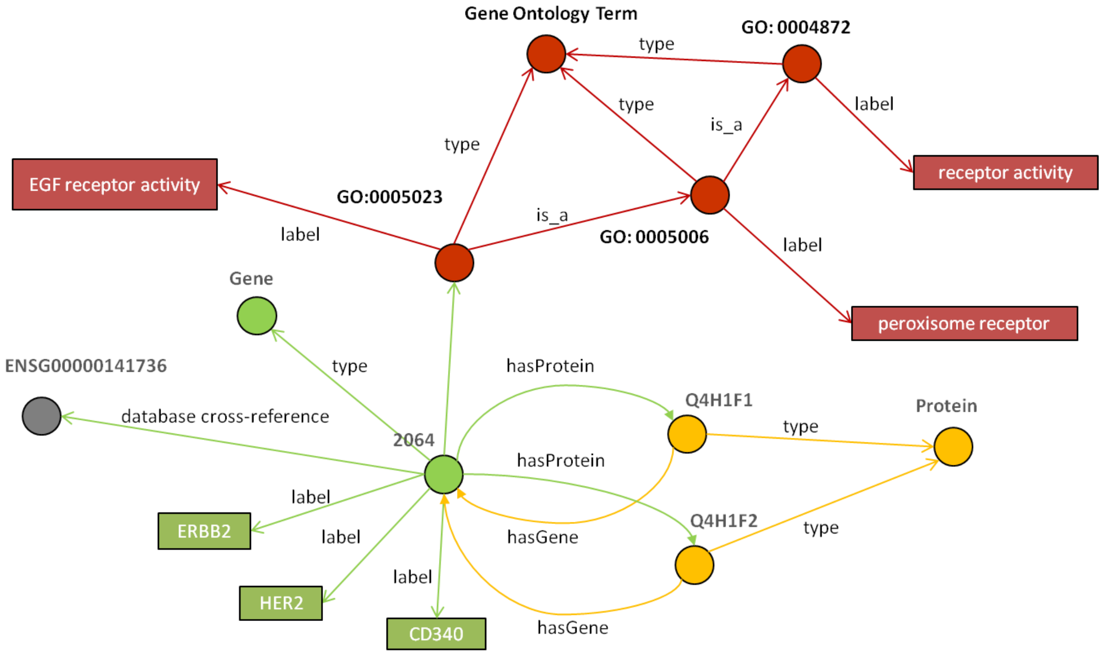


Figure 4. An example of linked information in LDD

All this information can be accessed instantly either using online faceted search, or using SPARQL queries that are issued to the LDD SPARQL endpoint. All information is stored using OWLIM database which is suitable for handling massive volumes of data. OWLIM allows the loading of all RDF statements in a single machine and guarantees very fast query response time. The database also supports federated queries using URIs hosted by external systems.

5.1.8 Cloud server standards

While many existing (e.g. security, virtualization) and emerging standards are important in cloud computing and several emerging efforts towards standardization exist, the adoption of standards in cloud computing is currently low. Some of these, such as security-related standards, apply to distributed computing environments. In this section we introduce commonly used definitions and classifications and present several relevant standardization initiatives. In Recommendations of the National Institute of Standards and Technology[49] cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The report states that this cloud model promotes availability. Five essential characteristics, three service models, and four deployment models are proposed. All current cloud implementations can be described according to these dimensions.

[49] http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\cloud-definition.pdf

The essential characteristics are: A) On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider. B) Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., smart phones, laptops, and tablets). C) Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines. D) Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out, and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time. E) Measured Service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

The service models described are: A) Cloud Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. B) Cloud Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations. C) Cloud Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

Finally, the following Deployment Models are defined: A) Private cloud: The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise. B) Community cloud: The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise. C) Public cloud: The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. D) Hybrid cloud: The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

The Open Cloud standards initiative[50], emerging from the Open Grid Forum is among the first to attempt standardization in cloud computing with the Open Cloud Computing Interface specification. Their goal is to develop an open specification and API for cloud offerings. The standards will adhere to the Open Cloud Principles, which are open formats, open interfaces, open data, and open source. The current focus is on Infrastructure-as-a-Service but the interface can be extended to support Platform and Software as a Service as well.

The Open Cloud Standards Incubator was formed by the DTMF (Distributed Management Task Force) to assess the impacts of cloud computing on management and virtualization standards and to make recommendations for extensions to better align with the requirements of cloud environments. It aims to enable portability and interoperability between private clouds within enterprises and hosted or public cloud service providers. The OCSI has published a white paper entitled Inter-operable Clouds[51] in which they defined a set of architectural semantics to unify the inter-operable management of enterprise and cloud computing. The report proposes usage scenarios, a service life-cycle and a conceptual Cloud Service Reference Architecture which describes key components (actors, interfaces, data artifacts, and profiles) and the relationships among these components.

The Cloud Management Working Group (CMWG) develops prescriptive specifications that deliver architectural semantics as well as implementation details to achieve inter-operable management of clouds between service requesters/developers and providers. This WG will propose a resource model that at minimum captures the key artifacts identified in the Use Cases and Interactions for Managing Clouds document produced by the Open Cloud Standards Incubator[52].

The Storage Networking Industry Association (SNIA) proposes the Cloud Data Management Interface (CDMI)[53], which defines the functional interface that applications may use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the

[50] http://occi-wg.org/

[51] http://dmtf.org/sites/default/files/standards/documents/DSP-IS0101.1.0.0.pdf

[52] http://cloud-standards.org/wiki/index.php?title=Main.Page

[53] http://www.snia.org/cdmi

data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.

In general, the adoption of cloud and use of cloud resources in different environments may demand compliance with specific requirements of those environments. This would also be the case in healthcare, where due to specific needs concerning security, privacy, legal, availability, performance and conformance with the clinical workflow not all models are applicable and compliance with additional requirements may be necessary. Although not directly cloud-related, the generic requirements for infrastructures for the federal government in the US are a very relevant example[54]. Their compliance with these additional requirements enabled Google to provide cloud services to the US federal government.

5.2 Clinical interoperability standards

With the increased adoption of electronic patient records there is an increase opportunity for creating longitudinal patient records that span many decades and aggregate data from multiple institutions and for collaboration among healthcare organizations in the process of delivering care. Within this trend the need for standards for the representation and exchange of patient data has become apparent. As defined by the Health Information Management Systems Society (HIMSS): “The Electronic Health Record (EHR) is 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. The EHR automates and streamlines the clinician’s workflow. The EHR has the ability to generate a complete record of a clinical patient encounter, as well as supporting other care-related activities directly or indirectly via interfaces including evidence-based decision support, quality management, and outcomes reporting.” EHRs use both technical and clinical standards. However, EHR vendors have only implemented some standards, while having a great deal of variation in their implementations, which results in systems that cannot interoperate and for which secondary use of data for research, epidemiology, etc. is difficult. Current EHR systems, due to their evolution over time, are often just an electronic representation of the previously used paper records. “Electronic patient records today are highly idiosyncratic, vendor-specific realizations of patient record subsets. They adopt few, if any, health information standards, and very rarely accommodate controlled terminologies where they might be sensible. The reason for this epidemic of incompatible data has more to do with the limitations of available information standards and machine-able vocabularies than with any fundamental unwillingness to adopt standards. A compelling business case, for system vendors or patient providers, simply has not emerged to foster standards adoption and systems integration.”

[54] http://csrc.nist.gov/groups/SMA/fisma/index.html

According to the report of the National Institutes of Health (National Center for Research Resources) report on EHRs[55], three main organizations create standards related to EHRs: Health Level Seven (HL7), Comite Europeen de Normalization - Technical Committee (CEN TC) 215, and the American Society for Testing and Materials (ASTM) E31. HL7, which operates in the United States, develops the most widely used healthcare-related electronic data exchange standards in North America. CEN TC 215, which operates in 19 European member states, is the preeminent healthcare IT standards developing organization in Europe. Both HL7 and CEN collaborate with the ASTM, which operates in the United States and is mainly used by commercial laboratory vendors.

5.2.1 HL7 (model / architecture)

The main aim of the HL7 messaging standard is to ensure that health information systems can communicate their information in a form which will be understood in exactly the same way by both sender and receiver. Whereas HL7 version 2 was a pure messaging standard for interoperability, version 3 (V3) not only specifies how to send a message, but also what a message can contain. To achieve this goal, V3 makes use of vocabularies and ontologies like SNOMED and LOINC. At the basis of all HL7 V3 messages is the Reference Information Model (RIM), an abstract model of the concepts which underlie healthcare information.

The RIM is defined as an object-oriented model, and the following are the definitions of the six core HL7 RIM classes:

- **Act** an action of interest that has happened, can happen, is happening, is intended to happen, or is requested/demanded to happen. An Act instance is a record of such an action.
- **Entity** a class or specific instance of a physical thing or an organization/group of physical things capable of participating in Acts. This includes living subjects, organizations, material, and places. The Entity hierarchy encompasses human beings, organizations, living organisms, devices, pharmaceutical substances, etc.
- **Role** establishes the roles that entities play as they participate in an Act. Each role is “played” by one Entity (the Entity that is in the role).
- Participation an association between a Role and an Act. The Participation represents the involvement of the Entity playing the Role with regard to the associated Act. A single Role may participate in multiple Acts and a single Act may have multiple participating Roles.
- **Act Relationship** an association between a pair of Acts. This includes Act-to-Act associations such as collector/component, predecessor/successor, and cause/outcome. The class has two associations to the Act class, one named source the other named target.
- **Role Link** a connection between two roles expressing a dependency between those roles

The HL7 Clinical Document Architecture (CDA) is a document markup standard, based on the HL7 RIM, that specifies the structure and semantics of clinical documents for the purpose of exchange. Examples of such clinical documents are referral notes, discharge summaries, and clinical

[55] http://www.ncrr.nih.gov/publications/informatics/ehr.pdf

summaries. The CDA document is defined such that it can include text, images, sounds, and other multimedia content. A CDA document is encoded in the XML language, and basically consists of a header and a (structured) body. Examples of such clinical documents are referral notes, discharge summaries, and clinical summaries. The CDA document is defined such that it can include text, images, sounds, and other multimedia content. A CDA document is encoded in the XML language, and basically consists of a header and a (structured) body. The header classifies the document and provides information on the encounter, the patient, and other involved entities. The body contains the actual clinical report, and often is divided into nestable document sections. Each of these sections can contain a single "narrative block" (that is, human readable content) and any number of CDA entries. These entries represent structured content and can be an envelope for information described in each of the HL7 domains, as the CDA is RIM-compliant.

5.2.2 HL7 Clinical Genomics

In the Clinical Genomics working group, HL7 V3 standards are developed that enable the exchange of interrelated clinical and personalized genomic data. Currently, the domain consists of three main topics: Genotype, Genetic Variation, and Pedigree (Family History). The latter topic aims at describing a patient's pedigree based on genomic data. As such, it utilizes the models from the Genotype topic to contain the genomic data for the patient's relatives. The Genotype topic consists of two (HL7 RIM-based) models: the Genetic Locus and the Genetic Loci. The latter model groups several genetic locus instances, e.g. in case of a genetic test of several genes. The first model describes data related to a genetic locus: the position of a particular given sequence in a genome or linkage map. The model includes sequencing and expression data, and can be linked to clinical information or phenotypes. Existing bioinformatics mark-up languages such as MAGE-ML and BSML are utilized to represent the raw genomic data.

The Genetic Variation topic defines a model that is a constraining of the Genotype topic models. The focus of this model is on variations in the DNA of individuals, derived using methods such as SNP probes, sequencing and genotype arrays that focus on small scale genetic changes. However, gene expression analysis, e.g. based on microarray data, is not suitable for the Genetic Variation model and will be addressed by different models within the HL7 Clinical Genomics working group. As the costs of generating genomic data, such as microarray-based gene expression, and extracting information from that data is still quite high it does not make economic sense to discard all the collected data (and to preserve only the test result) instead of storing and re-using it, especially that it is assumed that there is much more information in the data than what can be obtained in a single test or experiment. New standards for storing and exchanging genomic data such as MIAME and MAGE also require that all data should be preserved and annotated, including the raw microarray image. Of course, MIAME's and MAGE's main focus is research, but the reasoning behind the storage and full annotation of all genomic data is precisely based on the fact that the data encapsulates information far beyond the scope of a single experiment and should be preserved and shared. In this context, there is no reason to miss on such a valuable source of data that is represented by clinical practice. Many research-focused healthcare organizations have already identified this opportunity, and they invest in infrastructure to support this approach.

5.2.3 IHE

IHE Integration Profiles[56] describe the solution to a specific integration problem, and document the system roles (Actors), standards and design details for implementers to develop systems that cooperate to address that problem. IHE has introduced the first standards-based approach to cross-enterprise exchange of health information through its Cross-Enterprise Document Sharing (XDS) integration profile. XDS provides a specification for managing the sharing of documents between any healthcare enterprises, and handles a broad range of clinical documents including diagnostic imaging, medical summaries and scanned documents leveraging established standards such as DICOM and HL7. With the iHistory platform, HxTI provides a certified, XDS-compliant infrastructure for interoperable health information exchange. iHistory includes the ability to transform legacy PACS and RIS into XDS-compliant sources of clinical data.

IHE Profiles describe the solution to a specific integration problem, and document the system roles (Actors), standards and design details for implementers to develop systems that cooperate to address that problem. IHE Integration and Content Profiles are a convenient way for implementers and users to be sure they're talking about the same solution without having to restate the many technical details that ensure actual interoperability. Each IHE Profile has a short acronym. 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.

IHE is organized across a growing number of clinical and operational domains. Each domain produces its own set of Technical Framework documents, in close coordination with other IHE domains. Committees in each domain review and republish these documents annually, often expanding with supplements that define new profiles. Initially each profile is published for public comment. After the comments received are addressed, the revised profile is republished for trial implementation: that is, for use in the IHE implementation testing process. If criteria for successful testing are achieved, the profile is published as final text and incorporated. Each domain defines and publishes profiles to address interoperability issues related to its clinical and operational scope. The currently active IHE domains are:

[56] http://wiki.ihe.net/index.php?title=Profiles

- IHE Cardiology (CARD)[57]
- IHE Eye Care (EYECARE)[58]
- IHE IT Infrastructure (ITI)[59]
- IHE Laboratory (LAB)[60]
- IHE Anatomic Pathology (ANAPATH)[61]
- IHE Patient Care Coordination (PCC)[62]
- IHE Patient Care Device (PCD)[63]
- IHE Pharmacy (PHARM)[64]
- IHE Quality, Research and Public Health (QRPH)[65]
- IHE Radiation Oncology (RO)[66]
- IHE Radiology (RAD)[67]
- IHE Dental (DENT)[68]
- IHE Endoscopy[69]

[57] http://wiki.ihe.net/index.php?title=Profiles#IHE_Cardiology_Profiles

[58] http://wiki.ihe.net/index.php?title=Profiles#IHE_Eyecare_Profiles

[59] http://wiki.ihe.net/index.php?title=Profiles#IHE_IT_Infrastructure_Profiles

[60] http://wiki.ihe.net/index.php?title=Profiles#IHE_Laboratory_Profiles

[61] http://wiki.ihe.net/index.php?title=Profiles#IHE_Anatomic_Pathology_Profiles

[62] http://wiki.ihe.net/index.php?title=Profiles#IHE_Patient_Care_Coordination_Profiles

[63] http://wiki.ihe.net/index.php?title=Profiles#IHE_Patient_Care_Device_Profiles

[64] http://wiki.ihe.net/index.php?title=Profiles#IHE_Pharmacy_Profiles

[65] http://wiki.ihe.net/index.php?title=Profiles#IHE_Quality.2C_Research.2C_and_Public_Health_Profiles

[66] http://wiki.ihe.net/index.php?title=Profiles#IHE_Radiation_Oncology_Profiles

[67] http://wiki.ihe.net/index.php?title=Profiles#IHE_Radiology_Profiles

[68] http://wiki.ihe.net/index.php?title=Dental

[69] http://wiki.ihe.net/index.php?title=Endoscopy

New Domains are added as more fields of healthcare adopt the IHE process.

5.2.4 OpenEHR

openEHR is an open, detailed, and tested specification for a comprehensive interoperable health information computing platform for the EHR. It is based on a two level methodology concerning the development of information systems that separates the semantics of information and knowledge into two levels. The Reference Model corresponds to the information level and consists of a relatively small number of non-volatile abstract concepts. At the knowledge level models defining domain concepts by expressing constraints on instances of the underlying Reference Model are built, called archetypes. Next to the use of standards for data representation and exchange, the use of standard clinical vocabularies and of widely adopted ontologies would greatly enhance the ability of clinical information systems, such as EHRs to interoperate in a meaningful way. While syntactic interoperability is essential, the real added value with respect to automatically understanding the meaning of data from different systems, reasoning about the data and integrating that data into meaningful applications will come from enabling semantic interoperability.

5.2.5 DICOM

The DICOM Standards Committee create and maintain international standards for communication of biomedical diagnostic and therapeutic information in disciplines that use digital images and associated data. The goals of DICOM are to achieve compatibility and to improve workflow efficiency between imaging systems and other information systems in healthcare environments worldwide. DICOM is a cooperative standard. Connectivity works because vendors cooperate in testing via either scheduled public demonstrations, over the Internet, or during private test sessions. Every major diagnostic medical imaging vendor in the world has incorporated the Standard into its product design, and most are actively participating in the enhancement of the Standard. Most of the professional societies throughout the world have supported and are participating in the enhancement of the Standard as well.

DICOM is used or will soon be used by virtually every medical profession that utilizes images within the healthcare industry. These include cardiology, dentistry, endoscopy, mammography, ophthalmology, orthopedics, pathology, pediatrics, radiation therapy, radiology, surgery, etc. DICOM is even used in veterinary medical imaging applications. DICOM also addresses the integration of information produced by these various specialty applications in the patient’s Electronic Health Record (EHR). It defines the network and media interchange services allowing storage and access to these DICOM objects for EHR systems.

5.2.6 CDISC

The CDISC[70] was initiated in 1997 by the FDA (Food and Drug Administration) in order to develop standards for the acquisition, exchange, submission and storage of clinical data in clinical research. The primary goal of CDISC is to develop rules in order to submit standardized records to regulatory authorities. The other CDISC objectives are acquisition, exchange and storage of data. The records follow CDISC reporting rules and protocols, simplifying their interpretation and auditing by regulatory agencies, as well as decreasing the burden on trial physicians and speeding the entire clinical development cycle. This homogenisation also improves internal training and simplifies the set-up of new tests, both accelerating the adoption of systems of data capture and improving data exchange with partners and long term storage of clinical data. To ensure full neutrality with respect to the market economy in the world of research, the development of these standards is independent of computer platforms and solution vendors. CDISC has identified eXtensible Markup Language (XML) as the cornerstone of its plan because this language has a good reputation in industry.

The most relevant data-related standard in clinical research is CDISC[71]. Currently, CDISC is the leading standards development organization for the medical research domain. Its mission is to develop industry standards to support acquisition, exchange, submission and archiving of clinical trials data for medical and biopharmaceutical product development. The various standards are based on the CDISC Operational Data Model (ODM) standards. The standards wildly vary in maturity and uptake into practice. According a communication (Feb 2010) from CDISC, the FDA's Center for Drug Evaluation and Research (CDER) and Center for Biologics Evaluation and Research (CBER) give the following advice:

1. Use CDISC Standards (SDTM and ADaM) NOW in eSubmissions to CDER.
2. If you want high quality eSubmissions (which will be reviewed better and more quickly), start with CDISC CDASH for your case report forms when you plan your study.
3. The expected transport format for the foreseeable future is SASXPT with Define.xml

In order to facilitate an efficient submission of trial results to regulatory bodies, CDISC has defined the Study Data Tabulation Model (SDTM). The SDTM is a general framework describing the organization of the information that is collected during clinical trials. The SDTM consists of a set of clinical data file specifications and underlying guidelines. These different file structures are referred to as domains. Each domain describes a type of data associated with clinical trials, such as demographics, vital signs or adverse events. CDISC also provides a standard for running a clinical trial, namely the Operational Data Modelling (ODM) standard. ODM supports interchange between applications used in collecting, managing, analysing and archiving data.

[70] http://www.cdisc.org/

[71] http://www.cdisc.org

ODM provides a format for representing study metadata, study data and administrative data associated with a clinical trial. Various clinical trial management systems like SAS , OracleClinical support the ODM standard for data exchange. ODM provides a format for representing study metadata, study data and administrative data associated with a clinical trial. It represents the data that would be transferred between different software systems during a trial, or archived after a trial. There are two types of ODM files: snapshots files and transactional files. A snapshot file is completely self-contained, containing the current state of the source database. A transactional file shows, for each included entity, both the latest state of the source database, and (optionally) some prior states of that entity in the source database. The transactional files can provide an audit trail.

Clinical Data Acquisition Standards Harmonization (CDASH) is focused on data collection (as opposed to data reporting). The CDASH project identifies the basic data collection fields needed from a clinical, scientific and regulatory perspective to enable more efficient data collection at the investigative sites. The CDASH data collection fields (or variables) can be mapped to the SDTM structure. When the data are identical between the two standards, the SDTMIG variable names are presented in the CDASH domain tables. In cases where the data are not identical, CDASH has suggested new variable names. The CDASH recommendation also includes some data collection fields that are not included in the SDTMIG, these collection fields are intended to assist in the cleaning of data and in confirming that no data are missing. In some instances the optimal data collection method conflicts with the SDTM for reporting data, in these cases additional transformations and derivations may be needed to create the final SDTM compliant datasets. Some of the caBIG results, such as Common Data Elements (used for exchanging annotation, for example) need to be followed. There are also tools for CDEs. These are used in the clinical research community linked to caBIG and have the potential to become de-facto standards. For clinical research SAS[72] [60] may also be relevant due to its adoption by the pharmaceutical industry.

CDISC is now considered the common language for clinical trials. **CDASH**[73] (Clinical Data Acquisition Standards Harmonization) is a data standard mainly used to manage clinical trials and clinical data collection, and ensure interoperability between healthcare and clinical research.

Submission Data Tabulation Model[74] (SDTM) and Analysis Data Tabulation Model[75] (ADaM) are used for the submissions to regulatory authorities (FDA, EMA, etc.):

- **SDTM** is used for data reporting, and conflicts in some instances with CDASH. Thereby some transformations/derivations could be used to define STDM compatible data collection. CDISC defines a set for 16 safety SDTM streams (figure 5), by harmonising sections' names, definitions and metadata. The objective is to set up a standardized datasets model for all submissions.

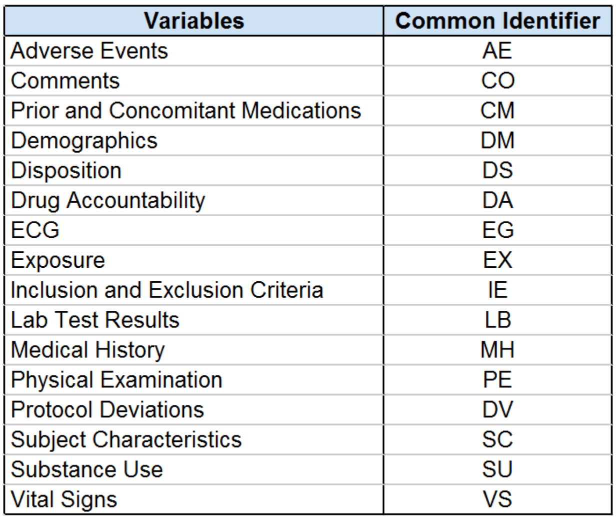
[72] http://www.sas.com/industry/life-sciences/cdisc/index.html

[73] http://www.cdisc.org/cdash/

[74] http://www.cdisc.org/sdtm/

[75] http://www.cdisc.org/adam

- **CDASH to SDTM mapping:** When SDTM variables are defined from data collection based on CDASH, mapping tables are needed for the variables that are to be reported. CDISC provides these mapping tables, which also defines some variables that are internal to SDTM (which are not reported), as well as an explanation about why they are not used in CDASH.
- **ADaM** is used to perform statistical analysis on clinical trial results, and comes together with SDTM for the submission to regulatory authorities.
- **ODM**[76] (Operational Data Model) The Operational Data Model enables the acquisition of data from a paper or electronic CRF (Case Report Form) that comes from a central laboratory or CRO (Clinical Research Organisation). It also enables storage of clinical trial data, as well as data exchange between different stakeholders in the areas of biomedical research. This model very precisely defines three types of information related to a clinical trial: metadata about a clinical trial, administrative data from a clinical trial, and clinical data for a test. The ODM standard is specified as a Document Type Definition (DTD) format, a dictionary for a specific type of XML document. This DTD also defines reference data, used for normal values of laboratory tests. Each section also includes information for specifying the elements that define the model. All this information is organized hierarchically in order to take into account the monitoring of different revisions of the model.



Variables	Common Identifier
Adverse Events	AE
Comments	CO
Prior and Concomitant Medications	CM
Demographics	DM
Disposition	DS
Drug Accountability	DA
ECG	EG
Exposure	EX
Inclusion and Exclusion Criteria	IE
Lab Test Results	LB
Medical History	MH
Physical Examination	PE
Protocol Deviations	DV
Subject Characteristics	SC
Substance Use	SU
Vital Signs	VS

Figure 5: CDASH / SDTM Common Variables

[76] http://www.cdisc.org/stuff/contentmgr/files/0/919cb4ef843829170d470b37eb662aeb/misc/odm1_1_0.html

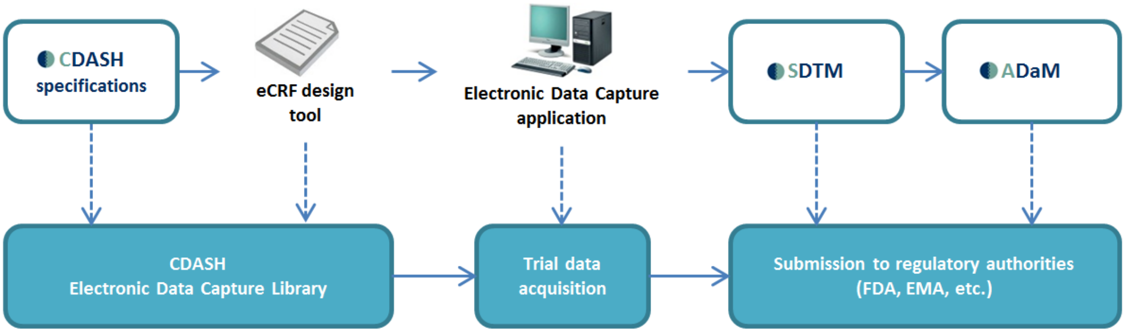


Figure 6: CDISC implementation flow[77] *(CDASH, SDTM, ADaM)*

5.3 Big Data and NoSQL

5.3.1 NoSQL vs. SQL

NoSQL standing for "Not only SQL" was originally motivated by Web 2.0 applications which require high scalability for managing thousands or millions of users doing updating while reading. Together with the development of Cloud Computing, there are increasing demands on efficient Big Data storage and processing, high performance reads and writes, which makes the traditional relational database and SQL face many challenges.

The relational database is difficult to be expanded and replicated on distributed machines; it has limit capacity, thus is inadequate for large scale, high concurrent Big Data processing such as search engines. Because the relation database has its own schema and complexity logic, when the amount and the size of the data increase, it is prone to produce deadlocks and other concurrency issues, which leads to slow read and write speed.

In contrast, NoSQL shows advantages when dealing with Big Data in many aspects such as high concurrent reads and writes with low latency, high scalability and available, easy for expansion and replication and massive amount of data storage.

Although the MyHealthAvatar is a proof-of-concept project, we foresee the future of such systems will manage large amount of patient data with high demands on updating and reading. Therefore, in this project, we invest our effort on NoSQL rather than the traditional relational database.

5.3.2 Review of Existing NoSQL Solutions

Four categories of data models for NoSQL implementations are available: Key-value, Column, Document and Graph. There are a number of existing literatures on the surveys of these implementations[78,79,80,81]. In the MyHealthAvatar project, we aim to use open source solutions as

[77] J. De Bondt, "CDASH The Rising Star," 2009.

[78] Strauch, C., Sites, U. L. S., & Kriha, W. (2011). NoSQL databases. *URL: http://www. christof-strauch. de/nosqldbs. pdf (дата обращения 07.11. 2012).*

possible, so this report mainly reviews the state-of-art open source NoSQL solutions in each category.

5.3.2.1 Key-value Stores

Key-value stores use a hash table where a unique key is used to refer to a particular item of data. The key is normally stored as a string, and the data itself is usually some kind of primitive data types (e.g. a string, an integer, and an array) or an object. Although the model structure is simple, querying a single value by the key is straightforward and faster than relational databases. However, as the data stored in the key-value stores has no schema at all, it is inefficient when querying or updating part of a value.

5.3.2.1.1 Redis

Redis[82] is a Key-value memory database. When Redis runs, data are entirely load into memory, so all the operations were run in memory, then periodically save the data asynchronously to the hard disk. The characteristics of pure memory operation makes it very good performance, it can handle more than 100,000 read or write operation per second. Redis support List and Set and various related operations and its maximum of value limit is 1GB. The main drawback is that capacity of the database is limited by physical memory, so Redis cannot be used as big data storage, and scalability is poor.

5.3.2.1.2 Project Voldemort

Project Voldemort[83] is a key-/value-store initially developed for and still used at LinkedIn. Voldemort provides multi-version concurrency control (MVCC) for updates. It updates replicas asynchronously, so it does not guarantee consistent data. However, it can guarantee an up-to-date view if you read a majority of replicas.

Both, keys and values can be complex, compound objects as well consisting of lists and maps. Compared to relational databases, the simple data structure and API of a key-value store does not provide complex querying capabilities: joins have to be implemented in client applications while constraints on foreign-keys are impossible; besides, no triggers and views may be set up.

5.3.2.2 Column Stores

Column stores are created to store and process large amount of data stored on distributed file systems. Keys are still used but refer to multiple columns which are arranged by column families. In contrast to relational database systems, data are still stored in tables, but are organised by columns

[79] Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on* (pp. 363-366). IEEE.

[80] Hecht, R., & Jablonski, S. (2011, December). NoSQL evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on* (pp. 336-341). IEEE

[81] Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, *39*(4), 12-27.

[82] http://redis.io

[83] http://www.project-voldemort.com/

rather than rows, which make key-based data aggregation much easier. However, there are no joins in column stores, relations have to be managed when developing applications. Column stores usually offer good concurrent read and write performance, fast key-based queries and scalable distributed store over the network. Column stores are typical used for data analytics and business intelligence in distributed environment.

5.3.2.2.1 Apache Cassandra

Apache Cassandra[84] is a distributed database system which is operated on clusters. Cassandra has been used widely (http://en.wikipedia.org/wiki/Apache_Cassandra) and been considered with good performance, availability and scalability. The key characteristics of Cassandra are: 1) no database schema design is required, and add or delete field is very convenient; 2) support various types of queries: row key queries, all row queries and range queries; 3) high scalability: a single point of failure does not affect the whole cluster.

In Cassandra, a write operation is replicated to multiple nodes, and read request is routed to a certain node. For a Cassandra cluster, only to add node can achieve the goal of scalability. Cassandra also supports rich data structure and powerful query language and it is compatible with Hadoop and can run MapReduce. Writes can be much faster reads, so Cassandra is normally considered for real-time data analysis.

5.3.2.2.2 Apache HBase

HBase[85] is tabular style store, where data are stored in tables. One table can have multiple column families, and similar to Cassandra, each Column Family may have multiple columns and each record is referenced by its row key. It supports row key query, range scan, filter query. HBase is built within Hadoop, so it naturally integrates well with Hadoop and can run MapReduce. As opposed to Cassandra, in HBase clusters, nodes are divided into different types. HBase has been used widely as well (http://wiki.apache.org/hadoop/Hbase/PoweredBy) and been considered with good scalability.

5.3.2.3 Document Stores

Document stores also use keys (or indexes) to refer to a collection of documents which are typically in semi-structure and stored in formats such as JSON and XML. Document stores usually focus on big data storage and query performance rather than high concurrent read and write performance.

5.3.2.3.1 MongoDB

MongoDB[86] is a database between relational databases and non-relational database, its key features are: 1) it support complex data types such as binary JSON format (BSON); 2) it supports index and

[84] http://cassandra.apache.org/
[85] http://hbase.apache.org/
[86] http://www.mongodb.org/

allows most of functions like query in single-table of relational databases; 3) it offers high-speed access to mass data; 4) it supports field search, regular expression search and range query.

MongoDB has been used by a number of enterprises as well, e.g. sourceforge, SAP. It has been considered with good performance in moderate scale.

5.3.2.3.2 CouchDB

Apache CouchDB[87] is a flexible, fault-tolerant database, which supports data formats such as ISON and AtomPub, it provides REST-style API. To ensure data consistency, CouchDB comply with ACID properties. In addition, CouchDB provides a P2P-based distributed database solution that supports bi-directional replication. However, it also has some limitations, such as only providing an interface based on HTTP REST, concurrent read and write performance is not ideal and so on.

5.3.2.4 Graph Stores

Graph stores are designed to store data with their relations can be represented in graphs. Social networks are typical examples can be represented by graph stores. Graph stores can take advantage of graph algorithm for application functionalities; however, it usually requires traverse the entire graph to achieve a definitive answer, so the query performance gets lower when the size of the graph increases.

5.3.2.4.1 Neo4j

Neo4j[88] is a high performance, robust and scalable graph database solving queries with multiple relationships storing data in the nodes and relationships [9]. Neo4j is fully written in Java and can be deployed on multiple systems.

The database is queried through Cypher Query Language. It is a new query language that has been recently added to the Neo4j. Cypher is a declarative language. Using Cypher, efficient querying of the graph is possible, without having to write traversers in the code.

5.3.3 Two-tier Data Repositories for MyHealthAvatar

In the MyHealthAvatar project, there will be a large amount of data in various types such as unstructured social data collected from daily life, semi-structured retrieved from available health records and structured data created for clinical modelling. In order to store and query all types of data efficiently, we introduce a two-tier data repository structure: one is a distributed NoSQL database for storing all the data that do not have strong schemas; the other is a semantic RDF repository for storing semi or structured data as well as the relationship schemas among the data.

[87] http://couchdb.apache.org/

[88] http://www.neo4j.org/

Cassandra is chosen for the NoSQL implementation considering its high scalability, no single point failure and potential for real time data analytics.

5.4 Data mining standards

Generally speaking, data mining lacks specific standards with only a few exceptions. It does not rely on proprietary formats but uses standards developed for other purposes. A review from the Indian Institute of Technology Kanpur[89] provides a comprehensive survey and classification of relevant standards supporting data mining. The data mining standards are classified based on one or more of the following issues:

1. The overall process by which data mining models are produced, used, and deployed: This includes, for example, a description of the business interpretation of the output of a classification tree.
2. A standard representation for data mining and statistical models: This includes, for example, the parameters defining a classification tree.
3. A standard representation for cleaning, transforming, and aggregating attributes to provide the inputs for data mining models: This includes, for example, the parameters defining how zip codes are mapped to three digit codes prior to their use as a categorical variable in a classification tree.
4. A standard representation for specifying the settings required to build models and to use the outputs of models in other systems: This includes, for example, specifying the name of the training set used to build a classification tree.
5. Interfaces and Application Programming Interfaces (APIs) to other languages and systems: There are standard data mining APIs for Java and SQL. This includes, for example, a description of the API so that a classification tree can be built on data in a SQL database.
6. Standards for viewing, analyzing, and mining remote and distributed data: This includes, for example, standards for the format of the data and metadata so that a classification tree can be built on distributed web-based data.

Specific standards are evolving for the first two of these categories. The 1999 European Cross Industry Standard Process for Data Mining[90] (CRISP-DM 1.0) is an effort to capture the various steps in a data mining process including Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment.

The Java Data Mining (JDM) is a standard Java API for developing data mining applications and tools. The JDM 1.0 standard was developed under the Java Community Process as JSR 73. As of 2006, the JDM 2.0 specification is being developed under JSR 247[91]. Various data mining functions and techniques like statistical classification and association, regression analysis, data clustering, and attribute importance are covered by the 1.0 release of this standard. These are evolving standards; later versions of these standards are under development. Independent of these standardization

[89] http://www.dataminінggrid.org/wdat/works/att/standard01.content.08439.pdf

[90] http: //community.udayton.edu/provost/it/training/documents/SPSS.CRISPWPlr.pd

[91] http://www.jcp.org/en/jsr/detail?id=73

efforts, freely available open-source software systems like the R language, Weka, KNIME, RapidMiner, jHepWork and others have become an informal standard for defining data-mining processes. Notably, all these systems are able to import and export models in Predictive Model Markup Language.

The Predictive Model Markup Language[92] (PMML) is a more formal standardization activity concerned with representing data mining models in terms of an XML-based language. It is a vendor-independent method of defining models so that proprietary issues and incompatibilities are no longer a barrier to the exchange of models between applications. It was developed by the Data Mining Group (DMG), an independent group composed of many data mining companies. PMML version 4.0 was released in June 2009.

A PMML model captures several aspects:

- Header
  - Version and timestamp
  - Model development environment information
- Data dictionary defining:
  - Variable types
  - Valid, invalid and missing values
- Data transformations
  - Normalization, mapping and discretization
  - Data aggregation and function calls
- Model
  - Description and model-specific attributes
- Mining schema
  - Usage type
  - Outlier and missing value treatment and replacement
- Targets
  - Score post-processing, scaling
  - Definition of model architecture and parameters

Many data mining systems support exchanging PMML models (for a list see Wikipedia)[93]. One notes that the support in terms of versions is quite heterogeneous) but the extent of usage of PMML as an exchange mechanisms is not very well documented.

5.4.1 Data Mining Tools and Systems

Notably, data mining rather relies on quasi-standards as defined by data tools and data mining systems rather than on standards. All these systems have internal proprietary data formats for the exchange between the components of the system and for storage. Examples for data mining tools and systems are described below.

[92] http://www.dmg.org/v4-0-1/GeneralStructure.html

[93] http://en.wikipedia.org/wiki/Predictive_Model_Markup_Language

RapidMiner[94] provides many data mining and machine learning procedures including data pre-processing and visualization. These can be nested to construct complex data mining processes. It integrates learning schemes and attributes evaluators of the machine learning environment WEKA and statistical modeling schemes of the R-Project. Weka[95] is an alternative to RapidMiner also providing many machine learning and data mining algorithms.

KNIME[96] is an open source data analytics, reporting and integration platform that integrates various components for machine learning and data. It provides a graphical user interface allowing quick and easy assembly of nodes for data preprocessing (Extraction, Transformation, Loading), for modeling and data analysis, and visualization.

jHEpWork[97] is a full-featured multiplatform data-analysis framework that incorporates many open-source math software packages into a coherent interface using the concept of Java scripting, rather than only-GUI or macro-based concept. jHepWork uses Jython, the Python language for the Java platform in order to call Java numerical and visualization libraries, which brings more power and simplicity for scientific computing. Other scripting languages (like BeanShell etc.) and, of course, Java itself, can also be used.

R[98] is a programming language and development environment for statistical computing and as a de facto standard for developing statistical software. In Dicode use case 1, R is used extensively but on the level of scripts. Dicode will split these scripts into reusable services using Rserve as enactment machine. Rserve[99] is a TCP/IP server that allows other programs to use the functionality of R. This allows combining R-based algorithms with other services.

Java Data Mining Package[100] (JDMP) is an open source Java library for data analysis and machine learning. It facilitates the access to data sources and machine learning algorithms (e.g. clustering, regression, classification, graphical models and optimization) and provides visualization modules. Import and export interfaces are provided for JDBC data bases, TXT, CSV, Excel, Matlab, Latex, MTX, HTML, WAV, BMP and other file formats. JDMP provides a number of algorithms and tools, but also interfaces to other machine learning and data mining packages. Many other data mining packages are available such as, e.g., Rattle, Apache Lucene, LibSVM and tm for Text Mining.

5.4.2 Data Exchange Formats

Data mining algorithms typically access data stored in databases using SQL-like languages. Explicit data exchange on file level takes place using proprietary formats like that of Excel or the ubiquitous

[94] http://rapid-i.com/content/view/181/190/lang,en/

[95] http://en.wikipedia.org/wiki/Weka.(machine.learning)

[96] http://www.knime.org/

[97] http://jwork.org/jhepwork

[98] http://en.wikipedia.org/wiki/R-Project

[99] http://www.rforge.net/Rserve/

[100] http://www.jdmp.org

comma separated values (CSV)[101], typically with a first row consisting of attribute names and subsequent rows being corresponding values. Another common file format is ARFF[102] (Attribute-Relation File Format). ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes and was initially developed for the use with Weka but currently is supported by other data mining tools such as Rattle and RapidMiner.

5.4.3 Data Mining Performance

Performance measurement of data mining relies on benchmark data sets as, for instance, provided by the UC Irvine Machine Learning Repository[103] or the Edinburgh Data Sets for Data Mining[104]. Alternatively, many data mining conferences or institutions set up data mining challenges or contests (see, e.g., ECML PKDD 20 Discovery Challenge[105], Hearst Challenge 2011[106]).

5.4.4 Large Scale Data Mining

Data mining on really large data sets is demanding in terms of computation time and storage that often exceeds the capacity of a single work station (with, e.g., a storage of even 128 Gbyte and computation time of, e.g., 4 month). Distributed data mining is a possible option for speed up and more storage. Several systems are under development with Apache Mahout[107] possibly developing into a quasi-standard though there are competing systems.

Apache Mahout currently has a very active development community. It has been reported to be used as part of the spam filtering pipeline at Yahoo![108], for matching couples at Speeddate, as well as a part of the recommendation modules at AOL and Foursquare. Mahout's goal is to provide scalable machine learning libraries for the Java programming language. Mahout implements most of its algorithms on top of Apache Hadoop[109] using the Map/Reduce paradigm. Therefore, it is perfectly suited for massive data that is stored in a Hadoop Cluster.

A very recent project called Radoop[110] integrates Hadoop in RapidMiner by providing a Hadoop extension for RapidMiner. Since Mahout uses Hadoop, it will then be possible to use Mahout class library from within RapidMiner.

[101101] http://en.wikipedia.org/wiki/Comma-separated.values

[102] http://weka.wikispaces.com/ARFF+.28stable+version.29

[103] http://archive.ics.uci.edu/ml/

[104] http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

[105] http://www.ecmlpkdd2011.org/challenge.php

[106] http://hearstchallenge.com/

[107] http://mahout.apache.org/

[108] http://www.slideshare.net/hadoopusergroup/mail-antispam

[109] http://hadoop.apache.org/

[110] http://prekopcsak.hu/index.php?slug=radoop

5.5 Communication standards

InMedica, a medical electronics market research group within IMS research (a leading independent provider of market research and consultancy to the global electronics industry), came up with a quantitative market assessment for the world Telehealth market in 2011[111]. In that assessment report it is estimated that Remote Patient Monitoring (RPM) is already highly relevant in the treatment of chronic diseases but that it also will increase over the next coming years. Medical devices that are deployable within the patient homes and make part of Telehealth services (e.g. Blood Glucose Meters, Pulse Oximeters, Weighing scales, Blood Pressure Monitors) are regarded to play a key role in the management of diabetes and hypertension which are two out of four main chronic disease managements. This all sounds like a good foundation to build any business models for MyHealthAvatar but what happens when a software platform becomes target for deployment restrictions? In Sweden, there is a rise in the number of software-related injuries and even deaths. Through the Swedish Medical products agency, the country asked the European Commission (EC) if it could be possible to handle unregulated software within EU.

What is then available as guidance in avoiding any possible future restrictions on qualified medical devices and software is that the classification itself depends on whether or not the device or software falls within the scope of the Medical Devices Directive (i.e. MDD; 2007/47/EC). Currently, the directive states that software can indeed be qualified as a medical device but unfortunately the directive does not specify what exact kind of software will meet the medical device definition per se. This uncertainty has led to various interpretations among EU member states and private vendors and developers, creating an uneven playing field for companies. The upcoming EC guideline will provide clarity in this issue.

5.5.1 Communication protocols

In order to make any kind of a device communicate with a gateway or a platform client of some kind it is important that it can seamlessly be integrated. The project, will adopt, the Continua Health Alliance guidelines[112] that describe a set of standards that are developed in order to allow vendors, solution developers and sharers of various types of health related information to easily share the data. Some of the communication protocols supported are described here:

5.5.1.1 Bluetooth

Bluetooth is an open wireless technology standard for exchanging data over short distances from fixed and mobile devices, by using short length radio waves and creating personal area networks (PANs). Bluetooth was initially created by telecoms vendor Ericsson in 1994, and today is managed by the Bluetooth Special Interest Group. It can, in its current specification, connect several devices, forming small short range networks.

[111] http://in-medica.com/press-release/Global_Telehealth_Market_Set_to_Exceed_1_Billion_by_2016

[112]http://www.continuaalliance.org/static/cms_workspace/Continua_Free_Guidelines_Release_1.11.12_Continua_Health_Alliance.pdf.

5.5.1.2 ZigBee

ZigBee is a WPAN standard for a suite of high level communication protocols using small, low-power digital devices with short range radios. ZigBee is typically used for industrial automation and domestic light control applications.

5.5.1.3 USB

The Universal Serial Bus (USB) is a wired point to point connection technology that is capable of high throughput (480Mbit/s for USB 2.0 and 4800Mbit/s for USB 3.0). The USB signals are transmitted through a twisted-pair data (channels) cable and prior to USB 3.0 these commonly used half-duplex differential signalling to reduce the electromagnetic noise effects in long lines. The USB 3.0 uses far more complex by introducing two additional pairs of shielded twisted wires and interoperable contacts. These data channels permit a higher data rate as well as full duplex operation. A USB connection is always between a host (or a hub, e.g. PDA) at the connector end and a device (or hub's upstream port, e.g. a biosensor's transport gate) at the other end.

5.5.1.4 Wi-Fi

A wireless fidelity (Wi-Fi) network is used to connect computers to each other, to the Internet, and also to wired networks. Current Wi-Fi networks operate in the ISM bands (2.4GHz, 5GHz), offer speeds up to 54Mbps and support quality of service (QoS) with managed levels for data, voice as well as video applications.

5.5.1.5 Li-Fi

Li-Fi (Light Fidelity) is the optical version of Wi-Fi and the newly launched Li-Fi Consortium[113] says that the technology will provide a secure, reliable and ultra high-speed wireless communication interfaces that relies on GigaSpeed technology, an optical mobile technology with several Li-Fi environmental features and service. It will use already existing communication technology or technical topics regarding optical wireless communication in order to increase available services in the consumer area, industry, medical area or in logistics. Li-Fi for medical devices in the future is a force to count on.

5.5.2 Tools and Resources

There are some tools and resources available online although these are more or less similar and build on the same technical approach. They could possibly assist in realising the communication strived by the Continua Guidelines.

5.5.2.1 Wipro Continua Toolkit

This toolkit enables medical device to get compliant to the Continua specified protocol, i.e. IEEE 11073-XXXXX, and it contains a Wipro Continua Agent and a Wipro Continua Manager. The Agent is a library component with a well-defined APIs and portable ANSI C source code for multiple sensor

[113] http://www.lificonsortium.org/index.html

and device specializations. The Manager is on the other hand capable of supporting multiple Continua Agent enabled medical devices in order to retrieve data via USB, Serial, Bluetooth and TCP/IP. The toolkit works on various types of platform both handheld devices and desktops. It works with OEM devices such as pulse Oximeters, glucose meters, weight scales and blood pressure monitors.

5.5.2.2 Stollman BlueHDP+USB dongle

This dongle can simply be used to add Health Device Profile (HDP) functionality to any PC with standard USB slot as there is no Bluetooth stack required on the PC. The Stollman[114] BlueHDP+USB dongle uses the Continua manager software CESL and works around of it enabling features such as embedded HDP and embedded Serial Port Profile. It uses the SPP to enable Continua compliant communication with agent devices but also proprietary communications. The dongle works over a virtual COM port to the application running on the PC which can use the Local Transport Protocol (LTP) to control the dongle and the data communicated.

5.5.2.3 Toshiba Bluetooth HDP stack and API

The Toshiba Bluetooth Stack for Windows has a HDP API layer that is implemented upon the (MCAP) IEEE 11073 Data Exchange Protocol layer and it (i.e. TosHdpApi) provides an interface to user applications for the HDP processing. It relies on the initial Toshiba Bluetooth stack which is only restricted to three types of HDP profiles: the blood pressure monitor (IEEE 11073-10407), the weighing scale (IEEE 11073-10415) and the cardiovascular fitness and activity monitor (IEEE 11073-10441).

5.5.2.4 ANT+

ANT+[115] is a managed network that uses ‘device profiles’ to define how to send data over the network in a consistent way. The target applications for the ANT+ managed network include sport, wellness, and home health. Of course in reality the ANT+ managed network is made up of many smaller networks spread around the world; these networks exist wherever a group of ANT+ sensors and receivers can be found, and are separated simply by the physical distance between them. In a nutshell: ANT+ devices use the ANT+ network key to access the ANT+ network, and they implement at least one of the ANT+ device profiles. At a high level that’s really all there is to it.

[114] http://www.stollmann.de/en/modules/bluetooth-products/bluehdp-usb.html

[115] http://www.thisisant.com/

6 Security standards

High security and privacy are necessary in modern medical applications. Strict policies are defined by official medical instances to ensure that the confidential medical data can be accessed and manipulated in a secure way. For this the data is protected by different security and privacy mechanisms like authentication, identification, authorization, anonymization, protected data transport and storage. It is clear that the p-medicine platform must use these mechanisms to reach a high level of security.

6.1 SAML

The Security Assertion Markup Language (SAML)[116] defines an XML-based protocol, making it possible to exchange authorization and authentication data between one or more security domains. This exchange is done by using signed assertions containing identity information. The entity that provides the assertions is called the asserting party while the relying party is the entity that consumes and verifies the assertions. A level of trust is required between the assertion providers and the relying parties. The current version of SAML is 2.0, which is a combination of three predecessor identity federation standards: SAML 1.1, ID-FF 1.2 and Shibboleth (Figure 7). This resulted in SAML 2.0 not being compatible with SAML 1.1. SAML mainly focusses on solving the problem of web browser single sign-on. For this it offers a single sign-on profile.

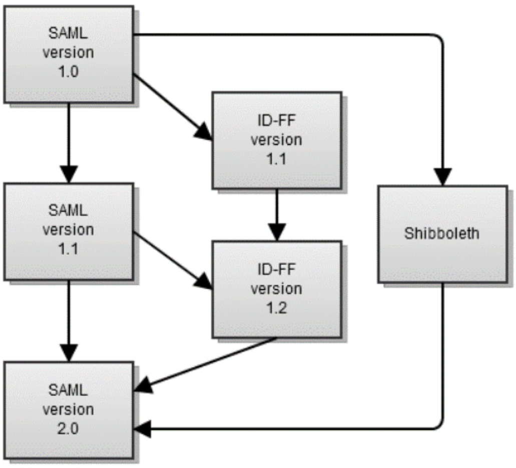


Figure 7: SAML 2.0 is a combination of three predecessor standards

[116] http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

6.2 Liberty-Alliance

The Liberty Alliance[117] is an effort of more than 150 organizations that try to establish open standards, guidelines and best practices for identity management. The main keywords in the project are federated identity, single sign-on, global logout, circle of trust and Web Services.

The Liberty Alliance Project contains three main components:

- The ***Liberty Identity Federation Framework*** (ID-FF) specifies core protocols (single sign-on (SSO), single logout (SLO), federation, name registration), schemata, bindings (HTTP and SOAP) and concrete profiles (Browser/Artifact, Browser/Post,...) that allow implementers to create a standardized, multi-vendor, identity federation network. It enables identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management. ID-FF is an extension of SAML and served as input for SAML 2.0.
- The ***Liberty Identity Web Services Framework*** (ID-WSF) consists of a set of schemata, bindings (SOAP, PAOS), protocols (Discovery and Interaction) and profiles (Security mechanisms ...) for providing a basic framework of identity services, such as identity service discovery and invocation. ID-WSF provides the framework for building interoperable identity services, permission based attribute sharing, identity service description and discovery, and the associated security profiles.
- The ***Liberty Identity Service Interface Specifications*** (ID-SIS) utilize the ID-WSF and ID-FF to provide networked identity services, such as contacts, presence detection or wallet services that depend on networked identity. ID-SIS enables interoperable identity services such as personal identity profile service, alert service, calendar service, wallet service, contacts service, geo-location service, presence service and so on.

6.3 WS-*

WS-* is a collective noun for a variety of specifications associated with web services[118],[119],[120],[121],[122]. These specifications form together the basic framework for web services build on the first-generation standards of SOAP, WSDL and UDDI. This association does not mean they are developed by a main standard body, the specifications are maintained by a diverse set of bodies or entities. The specifications are also not strictly disjoint, they may complement, overlap, and compete with each other.

6.4 OpenID

OpenID[123] is a lightweight HTTP-based protocol for single sign-on and attribute exchange. The OpenID specification is widely adapted and implemented by internet companies that have large user

[117] http://projectliberty.org/

[118] http://www.oasis-open.org/committees/tc.home.php?wg.abbrev=wss

[119] http: //docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf

[120] http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.4/os/ ws-secureconversation-1.4-spec-os.pdf

[121] http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ ws-securitypolicy-1.3-spec-os.pdf

[122] http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1. 2-spec-os.pdf

[123] http://openid.net/specs/openid-authentication-2.0.html

bases like Google, Yahoo, WordPress and Facebook. OpenID Authentication uses only standard HTTP(S) requests and responses, so no special capabilities are required from the user client. OpenID is not tied to the use of cookies or any other specific mechanism of consumer or OpenID Provider session management. Extensions to user clients can simplify the end user interaction but are not required by the protocol. The OpenID Authentication protocol messages are a defined fixed set of key-value pairs which are included in the HTTP(S) requests as HTTP parameters.

6.5 PKIX

PKIX is a standard, specified by the IETF's Public-Key Infrastructure working group, describing a public key infrastructure. It specifies public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm. PKIX is a derivation of the X.509[124] standard in order to adapt it to the more specific domain of internet standards. The term X.509 certificate usually refers to the IETF's PKIX Certificate and CRL Profile of the X.509 v3 certificate standard.

6.6 XACML

XACML (full name: eXtensible Access Control Markup Language)[125] is a XML based declarative access control policy language defining both a policy, decision request and decision response language. It is based on the Attribute Based Access Control (ABAC) model which incorporates Role Based Access Control (RBAC). Currently version 2.0 of XACML is adopted; the planned 3.0 version (which is currently in first draft) will support a broad range of new features. In the following paragraphs version 2.0 is used as reference unless otherwise stated.

6.7 Other

Other security standards are listed here for reference:

- U-Prove is a claims-based identity management framework that aims to offer anonymity, security, scalability and privacy based on cryptographic technologies.
- Shibboleth is an architecture and implementation of a federated single sign-on authentication and authorisation infrastructure heavily coupled with SAML.
- CAS is a centralized ticked based single sign-on HTTP-based protocol. Unlike most of the single sign-on systems, CAS lacks the use of attributes.
- Kerberos can be seen as the first widely distributed single sign-on system. It is a ticket oriented system that allows a user to authenticate him/herself in a non-secure network domain.
- OAuth is an open standard for authorization. It allows a resource owner to grant access to his/her private resources on one site (which is called the server), to another site (called client) without the need to share his/her personal credentials.
- In an attribute-based authorization model, identity information on a subject is exchanged from one site to another site in support of some action.

[124] http://tools.ietf.org/html/rfc5280

[125] http://www.oasis-open.org/committees/xacml/

- Ponder is a declarative, object-oriented language for specifying different types of policies, grouping policies into roles and relationships and finally defining configurations of roles and relationships as management structures.
- PERMIS (PrivilEge and Role Management Infrastructure Standards) is an authorization infrastructure that is based on two underlying technologies: role based access control (RBAC) and Policy based Management.
- The Gridge Toolkit is a set of integrated middleware services, based on GridLab, used to build grid environments.
- Cassandra is a role-based policy specification for access control in a distributed system in which the expressiveness (and the computational complexity) can be tuned according to need by choosing an appropriate constraint domain.

7 LEGAL AND ETHICAL STANDARDS

This chapter shall give an overview of the legal (and ethical) requirements for the transfer and processing of medical data, focusing on the transfer and processing of medical data. We provide an analysis of the legal requirements established on a European level in order to give an overview of the legal standards to be respected to lawfully establish the envisaged MyHealthAvatar platform.

Within EU legislation we will focus on the general rules and principles for processing of personal data stated by the Directive 95/46/EC on the protection of individuals with regard to the processing personal data and on the free movement of such data (Directive 95/46/EC, Data Protection Directive). The Data Protection Directive sets out the rights of the data subject and control mechanisms, establishes general rules on the lawfulness of the processing of personal data, and regulates the transfer of personal data into third countries. Directive 95/46/EC, thus, introduces the rules applicable to every processing of personal data throughout the EU. As it only covers the processing of personal data, whereas the processing of anonymous data does not fall into its scope, special attention shall be laid on the definition of these terms.

There may be more specific rules governing the use of patient data under specific circumstances. This might be the case, when data are used in the context of clinical trials. Therefore, Directive 2001/20/EC on the approximation of the laws, regulations and administrative provisions of the Member States relating to the implementation of good clinical practice in the conduct of clinical trials on medicinal products for human use (Directive 2001/20/EC, Clinical Trials Directive) that seeks to simplify and harmonize the administrative provisions governing clinical trials by establishing a clear, transparent procedure as well as the Directive 2001/83/EC on the Community code relating to medicinal products for human use. These two Directives, therefore, will also have to be analyzed in order to identify possible implications for the data protection and data security framework.

Ethical requirements will additionally be taken into consideration. These ethical standards are not always written in a legal reference document and may go beyond the legal rules. The main ethical issues to be discussed are related to the notion of "informed consent" and the possibility to access personal information ("right to know"), the "duty to inform" the patient of research results and the "quality of feedback" given by the researchers.

7.1 Data Protection Directive 95/46/EC

Directive 95/46/EC has two main purposes:

1. To allow for the free flow of data within the EU in order to prevent the Member States from blocking cross-border data flows on grounds of data protection within the EU and
2. To establish a minimum level of data protection throughout all Member States.

It had to be transposed to national law by all EU Member States, so that all national laws within the EU reflect the basic rules set by this Directive[126]. In the respective landmark ruling of the European Court of Justice, C-101/01, Lindquist, 06/11/2003[127], the ECJ clarified that Directive 95/46/EC envisages complete harmonization of the data protection regime within its scope. Accordingly Member States in principle have to adopt national legislation conforming to regime of the Directive. However, certain provisions of the Directive can explicitly authorize the Member States to adopt more constraining data protection rules. In doing so they, however, have to maintain a balance between free movement of personal data and protection of private life. Accordingly the Member States are allowed to set higher standards under specific circumstances, so that the data protection law is not completely harmonized within the EU. With regard to areas excluded from scope of application of Directive 95/46/EC Member States are free to regulate these areas in their own way, whenever there is no other rule of Community law providing otherwise.

7.1.1 Scope of the Directive and categories of data

Directive 95/46/EC distinguishes between several categories of data: “personal data”, “pseudonymous data”, “sensitive data” and “anonymous data”. The main distinction is made between personal data and anonymous data, since according to Art. 3 para. 1 Directive 95/46/EC, the Directive is applicable only to the processing of “personal data”, whereas “anonymous data” is not subject to the processing-restrictions of the Directive. Pseudonymous data and sensitive data define special categories of personal data, so that these categories of data in general underlie the scope of the Directive.

7.1.2 Territorial application

The Data Protection Directive is based on the territoriality principle. Art. 4 para. 1 lit. a Directive 95/46/EC provides that the Member State shall apply the national provisions it adopts pursuant to this Directive to the processing of personal data where the processing is carried out in the context of the activities of an establishment of the controller on the territory of the Member State. Accordingly the processing of personal data underlies the national law of the country in which the controller is established, regardless of where the actual processing takes place.

7.1.3 Requirements for the fair and lawful processing of data

The term “processing of personal data” (processing) is extraordinarily broad, covering “any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction“. Hence, this definition includes virtually

[126] See Art. 32 Directive 95/46/EC

[127] European Court of Justice, C-101/01, Lindquist, 06/11/2003, ECR 2003, I-12971.

any operation performed on personal data. Accordingly, also the anonymisation of personal data is to be regarded as processing of personal data.

7.2 Clinical Trials Directive 2001/20/EC

Directive 2001/20/EC45, also known as Clinical Trials Directive, states the requirements for the conduct of clinical trials in the EU. It is concretized by the Commission Directive 2005/28/EC[128] laying down principles and detailed guidelines for good clinical practice as regards investigational medicinal products for human use, as well as the requirements for authorization of the manufacturing or importation of such products (Good Clinical Practice Directive). By adopting this Directive the European legislator aimed to facilitate the performance of multi-national clinical trials in Europe through the harmonization of the regulatory procedures. It, however, only provides for minimum standards the national rules transposing the Directive in the Member States can vary considerably.

The Directive 2001/20/EC „establishes specific provisions regarding the conduct of clinical trials, including multi-centre trials, on human subjects involving medicinal products ..., in particular relating to the implementation of good clinical practice“ (Art 1 para. 1). It is applicable to clinical trials performed in the European Union. The term “clinical trial” is defined by Art. 2 lit. a of the Directive. It covers „any investigation in human subjects intended to discover or verify the clinical, pharmacological and/or other pharmacodynamics effects of one or more investigational medicinal product(s), and/or to identify any adverse reactions to one or more investigational medicinal product(s) and/or to study absorption, distribution, metabolism and excretion of one or more investigational medicinal product(s) with the object of ascertaining its (their) safety and/or efficacy“[129]. Directive 2001/20/EC, however, does not apply to non-interventional trials, meaning studies, where the medicinal product(s) is (are) prescribed in the usual manner in accordance with the terms of the marketing authorization. Directive 2001/20/EC applies to trials on medicinal products, covering trials on pharmaceuticals, which are in development, and “investigational medicinal products”. The Directive does not apply to other health related research, such as physiological research, research into medical devices, observational studies, research into tissue, organs or blood, or embryo research.

7.3 Medicinal Products Directive 2001/83/EC

Furthermore the Directive 2001/83/EC on the Community code relating to medicinal products for human use has to be taken into account. This Directive deals with the disparities between certain national provisions, in particular between provisions relating to medicinal products. A medicinal

[128] Commission Directive 2005/28/EC of 8 April 2005 laying down principles and detailed guidelines for good clinical practice as regards investigational medicinal products for human use, as well as the requirements for authorisation of the manufacturing or importation of such products (Text with EEA relevance), published in Official Journal L 91, 9.4.2005, p. 13.

[129] An “investigational medicinal product” according to Art. 2 lit. d Directive 2001/20/EC is „a pharmaceutical form of an active substance or placebo being tested or used as a reference in a clinical trial, including products already with a marketing authorisation but used or assembled (formulated or packaged) in a way different from the authorised form, or when used for an unauthorised indication, or when used to gain further information about the authorised form“.

product is defined as any substance or combination of substances presented for treating or preventing disease in human beings or any substance or combination of substances which may be used in or administered to human beings either with a view to restoring, correcting or modifying physiological functions by exerting a pharmacological, immunological or metabolic action, or to making a medical diagnosis (Art. 1 para. 2 Directive 2001/83/EC as amended by Directive 2004/27/EC). The Directive provides inter alia rules regarding the placing of medicinal products on the market, such as marketing authorization, specific provisions applicable to homeopathic medicinal products, procedures relevant to the marketing authorization. Furthermore the Directive regulates the manufacture and importation of medicinal products, their labelling and the package leaflet as well as the wholesale distribution of and advertisement for medicinal products.

Directive 2001/83/EC however only applies to industrially produce medicinal products for human use intended to be placed on the market in Member States.58 Art. 3 of the Directive provides several limitations of the scope. Accordingly the Directive does inter alia not cover medicinal products intended for research and development trials. Furthermore the Directive does not apply to whole blood, plasma or blood cells of human origin. Hence this Directive does not apply to the use of medicinal products within the project if these products are merely used for research and development trials. In the event that industrially produced medicinal products for human use intended to be placed on the market shall be tested the Directive provides inter alia for a clinical documentation.

7.4 Proposal on a General Data Protection Regulation 2012/0011(COD)

On 25 January 2012, the EU Commission published its draft General Data Protection Regulation 2012/0011(COD)[130]. The intention is that, in due course, this Regulation will replace current data protection laws across Europe. The Regulation's stated intention is to build a stronger and more coherent data protection framework in the EU that will resolve current legal uncertainties, put individuals in control of their own data and bring greater legal and practical certainty for organizations that are subject to the legislation. Once ratified by the EU member states and the European Parliament, the Regulation will replace the current EU Data Protection Directive (which is the basis of the UK's Data Protection Act 1998) and will become law across the EU without the need for further domestic implementation. The EU Commission has indicated that it aims to have in place a revised legislative framework by 2014.

7.5 Informed Consent

Similar to the legal concept of informed consent, the ethical approach provides specific requirements or preconditions in order to classify an informed consent as valid[131]. The three fundamental preconditions are (1) informed, (2) voluntarily and (3) capable to take decisions. With

[130] http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf

[131] Faden/Beauchamp, The History and Theory of Informed Consent

respect to the requirement “informed” the Declaration of Helsinki determines that participants in research projects should be provided with information on “the aims, methods, sources of funding, any possible conflicts of interest, institutional affiliations of the researcher, the anticipated benefits and potential risks of the study and the discomfort it may entail. In addition the subject should be informed of the right to abstain from participation in the study or to withdraw consent to participate at any time without reprisal“. What information has to be provided specifically, in the meaning of how much information is needed, depends on different criteria of which the most important one is the capability of the respective patient. The information should therefore be not only provided, but also provided in a way that the respective patient or participant can understand and is not overwhelmed by the information – that is why the relevant information necessary for a valid consent will differ in each particular case. Consent likewise has to be given “voluntarily” or “freely” to be valid. This means “that the individual is free from external pressure to make a particular decision”65. External pressure is to be assumed where consent is given under coercion or duress, under pressure, or under manipulation or undue influence. Last but not least the patient must be – in general - capable to take decisions. Decisional capacity is discussed in both law, as well as in ethics. Two levels of discussion can be recognized – the theoretical legal framework and the practical assessment of capacity.

8 Requirements Management (RM) tools

List of Requirements Management Tools:

- Enterprise Architect: http://www.sparxsystems.com/platforms/requirements.management.html
- Lighthouse RM: http://www.workspace.com/workspace/requirements.html
- MKS Requirements: http://www.mks.com/solutions/discipline/rm/requirements-management
- Open Source RM: http://sourceforge.net/projects/osrmt/
- Projectricity: http://www.projectricity.com/requirements.management.tool.htm
- SoftREQ: http://www.softreq.com/features.cfmTopTeam Analyst
- Accompa: http://www.accompa.com/requirements-management-software.html
- CASE Spec: http://www.casespec.net/requirementsmanagement.htm
- objectiF: http://www.microtool.de/instep/en/grafisches.anforderungsmanagement.asp
- RALLY Agile: http://www.rallydev.com/agile.products/lifecycle.management/
- RMTrak: http://www.rmtrak.com/
- TRUEreq: http://www.truereq.com/requirement-management.html
- ARCWAY Cockpit: http://www.arcway.com/en/product/arcway-cockpit-3/
- Bright Green Projects: http://www.brightgreenprojects.com/
- Case Complete: http://www.casecomplete.com/
- OneDesk requirements management: http://www.onedesk.com/features/requirements-management/
- Modelio Requirement Analyst: http://www.modeliosoft.com/en/modules/modelio-requirement analyst.html
- Objectiver: http://www.objectiver.com/

10 References

[IEEE00] IEEE Computer Society. Recommended Practice for Architectural Description. IEEE Std-1471-2000. October 9, 2000. http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html.

[ENC] http://www.eucalyptus.com/

[Had] http://hadoop.apache.org/

[MEL11] P. Mell and T. Grance. The NIST Definition of Cloud Computing. National Institute of Science and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[AMA] Amazon Web Services, http://aws.amazon.com/, Amazon, 2011.

[GGRI] GoGrid, http://www.gogrid.com/, GoGrid, 2011

[NASA] Nebula Cloud . http://nebula.nasa.gov

[AMVPC] Amazon Virtual Private Cloud (Amazon VPC), http://aws.amazon.com/vpc/2011

[KVM] Kernel Based Virtual Machine, http://www.linux-kvm.org/page/Main_Page, 2011.

[XEN] X. Hypervisor, http://www.xen.org/, Citrix Systems, 2011.

[EUC] Eucalyptus, http://open.eucalyptus.com/, Eucalyptus Systems, 2011.

[ARM09] M. Armbrust, A. Fox, R. Grifth, AD. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report, University of California at Berkeley, February 2009.

[DEI] Distributed European Infrastructure for Supercomputing Applications (DEISA): http://www.deisa.org (accessed May 9, 2011)

[CAC10] C. Cachin, R. Haas, M. Vukolic. Dependable Storage in the Intercloud, Aug 2010.

[VUK10] M. Vukolic , The Byzantine Empire in the Intercloud, SIGACT News, 41:105–111, September 2010.

[FIE02] R.T. Fielding and R.N. Taylor. Principled design of the modern Web architecture, ACM Transactions on Internet Technology (TOIT), 2(2), 115 – 150, 2002.

[EHR] http://en.wikipedia.org/wiki/Electronic_health_record

[HL7] http://www.hl7.org/

[OpenEHR] http://www.openehr.org/home.html

[RUS] Rusher, J., Triple Store, Semantic Web Advanced Development for Europe (SWAD-Europe), Workshop on Semantic Web Storage and Retrieval - Position Papers

[DIN03] Dingley, A.P., 2003. Storage and Management of Semi-structured Data (Use of SQL relational databases as an RDF triple store), US2003145022 Patent

[HAS11] Haslhofer, B., Momeni R.E., Schandl, B., Zander, S., 2011. Europeana RDF Store Report, project EuropeanaConnect

[TRI] Large TripleStores: http://www.w3.org/wiki/LargeTripleStores

[TRIPL] Triplestore: http://en.wikipedia.org/wiki/Triplestore#cite_note-4

[COM] Comparison of Triple Stores: http://www.bioontology.org/wiki/images/6/6a/Triple_Stores.pdf

[SIR07] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., Katz, A., 2007. Pellet: A practical OWL-DL reasoner, International Journal of Web Semantics, Vol 5 (2), pp 51-53.

[HOR06] Horrocks, I., Tsarkov, D., 2006. FaCT++ Description Logic Reasoner: System Description. In: Proc. Third Int'l. Joint Conference of Automated Reasoning

[MOL03] Moller, R., Harrslev, V., 2003. Racer: A Core Inference Engine for the Semantic Web. In: Proc. 2nd Int'l. Workshop Evaluation of Ontology Based Tools, pp. 27–36

[MIN04] Minsu, J., Sohn, J., 2004. Bossam: An extended rule engine for OWL Inferencing. In: Antoniou, G., Boley, H. (eds.) RuleML 2004. LNCS, vol. 3323, pp. 128–138. Springer

[KIR05] Kiryakov, A., Ognyanov, D., Manov, D., 2005. OWLIM - a Pragmatic Semantic Repository for OWL, Proc. Workshop Scalable Semantic Web Knowledge Base Systems

[MOT05] Motik, B., Studer, R., 2005. KAON2 – A Scalable Reasoning Tool for the Semantic Web, Proc. 2nd ESWC, Heraklion, Greece.

[SAF09] Safdar, A., Stephan, K., 2009. µOR – A Micro OWL DL Reasoner for Ambient Intelligent Devices , In Proc. of 4th International IEEE Conference on Grid and Pervasive Computing, Geneva, Switzerland, Lecture Notes in Computer Science 5529, pp 305-316.

[KLE05] Kleemann, T., Sinner, A., 2005. KR-Hyper - In Your Pocket. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 452–457. Springer

Appendix 1 – Abbreviations and acronyms

ADaM Analysis Data Model CDISC standard supporting efficient generation, replication, and review of analysis results

AES Advanced Encryption Standard a specification for the encryption of electronic data based on a design principle known as a Substitution permutation network

AGPL Affero General Public License refers to two free software licenses. Affero General Public License, Version 1 and GNU Affero General Public License, version 3.

API application programming interface is a particular set of rules (’code’) and specifications that software programs can follow to communicate with each other

ASCII American Standard Code for Information Interchange a character-encoding scheme based on the ordering of the English alphabet

BMP Bitmap a raster graphics image file format used to store bitmap digital images

BSD Berkeley Software Distribution is a Unix operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley

CAS Central Authentication Service a single sign-on web protocol

CA Certification Authority an entity that issues digital certificates

CBC Cipher-Block Chaining a cryptographic mode of operation in which each block of plaintext is XORed with the previous ciphertext block before being encrypted

CCM Counter with CBC-MAC Mode a mode of operation for cryptographic block ciphers

CCZero Creative Commons licenses are several copyright licenses that allow the distribution of copyrighted works

CDASH Clinical Data Acquisition Standards Harmonization CDISC standard describing the basic recommended (minimal) data collection fields for 18 domains, including common header fields, and demographic, adverse events, and other safety domains that are common to all therapeutic areas and phases of clinical research

CDA Clinical Document Architecture is an XML-based markup standard intended to specify the encoding, structure and semantics of clinical documents for exchange

CDE Clinical Document Architecture an XML-based markup standard defined by HL7 intended to specify the encoding, structure and semantics of clinical documents for exchange

CDISC Clinical Data Interchange Standards Consortium - a global, open, multidisciplinary, non-profit organization that has established standards to support the acquisition, exchange, submission and archive of clinical research data and metadata

CDMI Cloud Data Management Interface - defines a functional interface that applications can use to create, retrieve, update and delete data elements from the Cloud

CDS Clinical Decision Support decision support software designed to assist physicians and other health professionals with decision making tasks, as determining diagnosis of patient data

CMWG Cloud Management Work Group focused on standardizing interactions between cloud environments by developing specifications that deliver architectural

	semantics and implementation details to achieve interoperable cloud management between service providers and their consumers and developers
CRISP-DM	Cross Industry Standard Process for Data Mining a data mining process model that describes commonly used approaches that expert data miners use to tackle problems
CRL	Certificate Revocation List a list of certificates that have been revoked, and therefore should not be relied upon
CSS	Cascading Style Sheets is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language
CSV	Comma-Separated Values a set of file formats used to store tabular data in which numbers and text are stored in plain-text form that can be easily written and read in a text editor
CWM	Common Warehouse Metamodel a specification for modeling metadata for relational, non-relational, multi-dimensional, and most other objects found in a data warehousing environment
CeCILL	CEA CNRS INRIA Logiciel Libre is a free software license adapted to both international and French legal matters, in the spirit of and retaining compatibility with the GNU General Public License
CellML	Cell Markup Language is an XML based markup language for describing mathematical models
DICOM	Digital Imaging and Communications in Medicine - a standard for handling, storing, printing, and transmitting information in medical imaging
DTMF	Distributed Management Task Force brings the IT industry together to collaborate on the development, validation and promotion of systems management standards
EHR	Electronic health record is an evolving concept defined as a systematic collection of electronic health information about individual patients or populations
EULA	End-user licensing agreements An EULA is a legal contract between the manufacturer and/or the author and the end user of an application
EUPL	European Union Public License the first European Free/Open Source Software (F/OSS) license
FMA F	Foundational Model of Anatomy it is concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body in a form that is understandable to humans and is also navigable, parseable and interpretable by machine-based systems
FieldML	Field Markup Language is an XML based markup language for describing field models
GAS	Gridge Authorization Service provides functionality that would be able to fulfill most authorization requirements of grid computing environments
GCM	Galois/Counter Mode a mode of operation for symmetric key cryptographic block ciphers that has been widely adopted because of its efficiency and performance
GEM	Guideline Elements Model an XML-based guideline document model that can store and organize the heterogeneous information contained in practice guidelines

GNU Gnu’s Not Unix is a Unix-like computer operating system developed by the GNU project, ultimately aiming to be a ”complete Unix-compatible software system” composed wholly of free software.

GO Gene Ontology is a major bioinformatics initiative with the aim of standardizing the representation of gene and gene product attributes across species and databases

GPL General Public License is the most widely used free software license, originally written by Richard Stallman for the GNU Project

GridFTP GridFTP is an extension of the standard File Transfer Protocol (FTP) for use with Grid computing

HL7 Health Level Seven is an all-volunteer, non-profit organization involved in development of international healthcare informatics interoperability standards

HMAC Hash-based Message Authentication Code a mechanism for message authentication using cryptographic hash functions

HTML Hypertext Markup Language is the predominant markup language for web pages. HTML elements are the basic building-blocks of webpages.

HTTPS Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server

IBM International Business Machines

ID-FF Liberty Identity Federation Framework an approach for implementing a single sign-on with federated identities based on commonly deployed technologies

ID-WSF Liberty Identity Web Services Framework a framework for identity-based web services in a federated network identity environment

IEC International Electrotechnical Commission is the world’s leading organization that prepares and publishes International Standards for all electrical, electronic and related technologies

IEEE Institute of Electrical and Electronics Engineers is a non-profit professional association headquartered in the United States that is dedicated to advancing technological innovation and excellence

IETF Internet Engineering Task Force a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet

IHE Integrating the Healthcare Enterprise - an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information

IPSec Internet Protocol Security a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session

ISBN International Standard Book Number is a unique numeric commercial book identifier based upon the 9-digit Standard Book Numbering (SBN) code created by Gordon Foster

ISO International Organization for Standardization is an international standard-setting body composed of representatives from various national standards organizations

InSilicoML InSilico Markup Language is a markup language that can explicitly describe the multi-level hierarchical structures of the physiological functions in mathematical models

JDMP Java Data Mining Package an open source Java library for data analysis and machine learning

JPEG Joint Photographic Experts Group is a commonly used method of lossy compression for digital photography

JSDL Job Submission Description Language is an extensible XML specification from the Global Grid Forum for the description of simple tasks to non-interactive computer execution systems

KNIME Konstanz Information Miner a user-friendly and comprehensive open source data integration, process, analysis and exploration platform

LGPL Lesser General Public License is a free software license published by the Free Software Foundation

LOINC Logical Observation Identifiers Names and Codes is a database and universal standard for identifying medical laboratory observations

MAGE-ML Microarray and Gene Expression - Markup Language markup language format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MAGE-OM Microarray and Gene Expression - Object Model data exchange model for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MAGE-TAB Microarray and Gene Expression - Tabular tabular format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MIAME Minimum Information About a Microarray Experiment needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment

MIASE Minimal Information About a Simulation Experiment common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models

MIASE Minimum Information About a Simulation Experiment is an effort to list the common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models.

MIBBI Minimum Information for Biological and Biomedical Investigations maintains a web-based, freely accessible resource for ”Minimum Information” checklist projects, providing straightforward access to extant checklists (and to complementary data formats, controlled vocabularies, tools and databases), thereby enhancing both transparency and accessibility

MIT MIT License is a free software license originating at the Massachusetts Institute of Technology

ML Markup Language is a modern system for annotating a text in a way that is syntactically distinguishable from that text

MOF MetaObject Facility the foundation of OMG’s industry-standard environment where models can be exported from one application, imported into another, transported across a network, stored in a repository and then retrieved, rendered into different formats

MPL Mozilla Public License is a free and open source software license

MS Microsoft is an American public multinational corporation headquartered in Redmond, Washington

MTOM Message Transmission Optimization Mechanism is the W3C Message Transmission Optimization Mechanism, a method of efficiently sending binary data to and from Web services

MedLEE Medical Language Extraction and Encoding system System to extract, structure, and encode clinical information in textual patient reports so that the data can be used by subsequent automated processes

NeuroML Neuro Markup Language is an XML (Extensible Markup Language) based model description language that aims to provide a common data format for defining and exchanging models in computational neuroscience

OASIS Organization for the Advancement of Structured Information Standards a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society

OBO Open Biomedical Ontologies is an effort to create controlled vocabularies for shared use across different biological and medical domains

OGSA-BES Open Grid Services Architecture - Basic Execution Services defines Web Services interfaces for creating, monitoring, and controlling computational entities such as UNIX or Windows processes, Web Services, or parallel programswhat we call activities within a defined environment

OGSA-DAI Open Grid Service Architecture-Data Access and Integration allows data resources (e.g. relational or XML databases, files or web services) to be federated and accessed via web services on the web or within grids or clouds. Via these web services, data can be queried, updated, transformed and combined in various ways.

OSI Open Source Initiative is an organization dedicated to promoting open source software

OS Operating System is a set of programs that manages computer hardware resources, and provides common services for application software

OWL-S Ontology Web Language for web Services an ontology of services to discover, invoke, compose, and monitor Web resources offering particular services and having particular properties

OWL Web Ontology Language is a family of knowledge representation languages for authoring ontologies.

OpenID Open Identity provider of web-based SSO services

PAOS Reverse HTTP Binding for SOAP a binding that enables HTTP clients to expose services using the SOAP protocol, where a SOAP request is bound to a HTTP response and vice versa

PATO PATO an ontology of phenotypic qualities, intended for use in a number of applications, primarily defining composite phenotypes and phenotype annotation.

PHP PHP: Hypertext Preprocessor is a general-purpose server-side scripting language originally designed for web development to produce dynamic web pages

PKIX Public-Key Infrastructure Working Group was established in the fall of 1995 with the goal of developing Internet standards to support X.509-based Public Key Infrastructures

PMML Predictive Model Markup Language an XML-based language which provides a way for applications to define statistical and data mining models and to share models between PMML compliant applications

PNG Portable Network Graphics is a bitmapped image format that employs lossless data compression

POST POST is one of many request methods supported by the HTTP protocol used by the World Wide Web

RAD Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping

RDF Resource Description Framework is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model

REST Representational state transfer is a style of software architecture for distributed hypermedia systems such as the World Wide Web

RFC Request for Comments is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems

RICORDO RICORDO is focused on the study and design of a multiscale ontological framework in support of the Virtual Physiological Human community to improve the interoperability amongst its Data and Modelling resources

RIM Reference Information Model is the cornerstone of the HL7 Version 3 development process and an essential part of the HL7 V3 development methodology

SAML Security Assertion Markup Language a standard, XML-based framework for creating and exchanging security information between online partners

SAS Business analytics software and service developer, and independent vendor in the business intelligence market

SAWSDL Semantic Annotations for WSDL defines mechanisms using which semantic annotations can be added to WSDL components

SBML System Biology Markup Language is a representation format, based on XML, for communicating and storing computational models of biological processes

SDTM Study Data Tabulation Model CDISC defining a standard structure for human clinical trial (study) data tabulations that are to be submitted as part of a product application to a regulatory authority

SED-ML	Simulation Experiment Description Markup Language an XML-based format for encoding simulation experiments, following the requirements defined in the MIASE guidelines
SHA	Secure Hash Algorithm a number of cryptographic hash functions published by the National Institute of Standards and Technology as a U.S. Federal Information Processing Standard
SLO	Single Log-Out termination of a SSO action
SNIA	Storage Networking Industry Association not-for-profit trade organization for companies and individuals in various sectors of the storage industry
SNOMED-CT	Systematized Nomenclature of Medicine - Clinical Term is a systematically organised computer processable collection of medical terminology covering most areas of clinical information such as diseases, findings, procedures, microorganisms, pharmaceuticals etc
SOAP	Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks
SOAP	Simple Object Access Protocol a lightweight XML-based protocol for exchange of structured information in a decentralized, distributed environment
SOA	Service-Oriented Architecture s a set of principles and methodologies for designing and developing software in the form of interoperable services
SPARQL	SPARQL Protocol and RDF Query Language - query language for RDF
SQL	Structured Query Language a standard language for accessing and manipulating databases
SSH	Secure Shell is a network protocol for secure data communication, remote shell services or command execution and other secure network services between two networked computers that it connects via a secure channel over an insecure network
SSL	Secure Sockets Layer a cryptographic protocol that provides communication security over the Internet, predecessor of TLS
SSO	Single Sign-On a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords
TCP/IP	Transmission Control Protocol/Internet Protocol the first two networking protocols defined in the Internet Protocol Suite standard
TDD	Test-driven development is a software development process that relies on the repetition of a very short development cycle.
TLS	Transport Layer Security a cryptographic protocol that provides communication security over the Internet, successor of SSL
UML	Unified Modelling Language a specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems
VDM	Vienna Development Method is one of the longest-established Formal Methods for the development of computer-based systems

VPH-NoE	Virtual Physiological Human - Network of Excellence is a project which aims to help support and progress European research in biomedical modelling and simulation of the human body
VPH	Virtual Physiological Human is a methodological and technological framework that, once established, will enable collaborative investigation of the human body as a single complex system
WAV	Waveform Audio File Format is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs
WS-*	Web Services-* common prefix for the family of Web Services specifications
WSDL	Web Services Description Language a way to describe the abstract functionalities of a service and concretely how and where to invoke it
WSMO	Web Service Modelling Ontology ontology for describing Semantic Web Services
XFree86	A freely redistributable open-source implementation of the X Window System
XHTML	Extensible HyperText Markup Language is a family of XML markup languages that mirror or extend versions of the widely-used Hypertext Markup Language (HTML), the language in which web pages are written
XML	Extensible Markup Language - a format for encoding documents in machine-readable form, similar in syntax to HTML
XTS	XEX-based Tweaked Codebook a mode of operation for cryptographic block ciphers
caBIG	Cancer Biomedical Informatics Grid a virtual network of interconnected data, individuals, and organizations that work together to redefine how cancer research is conducted