Building a Semantic Repository for Outpatient Sheets

Maria Nisheva[1, 2], Hristo Georgiev[1] and Pavel Pavlov[1]

[1] Faculty of Mathematics and Informatics, Sofia University St. Kliment Ohridski,
5 James Bourchier Blvd., 1164 Sofia, Bulgaria
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

marian@fmi.uni-sofia.bg

Abstract. The paper analyzes some issues in the area of semantic interoperability of health information systems and in particular, the issues related to the provision of semantic interoperability of health data. The design principles and some implementation details of a semantic repository for outpatient sheets are discussed in the context of the suggested ideas.

Keywords: eHealth, semantic data models, semantic interoperability of health data.

1 Introduction

The rapid development and use of various health information systems and networks raises the issue of semantic interoperability between heterogeneous health informatics applications. Semantic interoperability may be characterized as the capability of different software systems to share information and to have that information properly interpreted by the receiving system in the same sense as intended by the creators or maintainers of the transmitting system. It involves:

- the processing of the shared information in all systems so that it is consistent with the intended meaning of this information;
- the encoding of queries and presentation of information so that it conforms to the intended meaning regardless of the source of information.

Standardization and utilization of semantic technologies are indicated as most effective instruments for providing and maintaining interoperability in information systems and especially in healthcare information systems.

Semantic technologies or Semantic Web technologies such as Linked Data, Resource Description Framework (RDF/RDFS), SPARQL and different kinds of domain ontologies are increasingly being used in the health informatics community to respond to the knowledge integration and semantic interoperability needs [1]. In particular, semantic repositories can be used to achieve various goals, such as

ability of managing large volumes of heterogeneous data, significant analytical power that is based on abilities of interlinking long-chain evidences, effective data interoperability.

According to [2], "a repository (also referred to as a data repository or digital data repository) is a searchable and queryable interfacing entity that is able to store, manage, maintain, and curate data/digital objects". A repository is a managed location where digital data objects are registered, permanently stored, made accessible and retrievable, and curated [3]. Repositories preserve, manage, and provide access to many types of digital resources, available in a variety of formats. Resources in online data repositories are curated to provide their search, discovery, and reuse.

This paper aims to present some results achieved in the development of a pilot version of semantic repository for clinical data received in formats used by the information system of the Bulgarian National Health Insurance Fund (NHIF).

2 Key features of a semantic repository for clinical data

Semantic repositories are software systems similar to database management systems (DBMSs). They allow the storage, retrieval and management of structured or semi-structured data. The main differences between semantic repositories and traditional DBMSs can be summarized as follows:

- the data in the semantic repositories are presented in RDF format and ontologies are used as data schemas that allow for automatic analysis and formal reasoning;
- the physical data model is flexible and schema independent, which simplifies the integration of additional knowledge and schemas, in particular the integration of relevant subject ontologies.

The minimum functionality of a software system that plays the role of a semantic repository of clinical data should include:

- reading XML documents, validating them and presenting them as RDF documents in different formats;
- loading RDF documents into the semantic repository;
- retrieving particular documents, specified by flexible queries;
- providing an interface that allows defining and executing SPARQL queries to the system;
- adding a single document or a set of documents to the semantic repository. The addition of documents should be implemented in compliance with the requirement of sustainability over time, and assistance that unplanned interruption of the system operation should not cause loss of information;
- editing existing documents and loading their modified versions to the repository;
- retrieving the entire document set so that it can be easily integrated with different external systems using various RDF formats.

In addition to meeting these general requirements, a semantic repository of clinical data oriented to real-world practical applications should also provide

- maintaining an interface to the corresponding core information system while maintaining the ability to handle documents in real time;
- identifying ways to connect the system to other (external) systems, in particular with appropriate graph or relational databases;
- developing additional systems to provide means of visualizing the documents used by the core system.

Fully to exploit the capabilities of a semantic repository, and in particular a semantic repository for electronic patient records, it should include multiple domain ontologies of various types. First, appropriate subject ontologies describing socially significant diseases, their diagnosis, planning of appropriate therapy, etc. should be selected or in some cases developed especially for this purpose. The inclusion of such ontologies enables the development of semantically interoperable information systems, decision support systems, data mining systems, and other types of intelligent software systems in the healthcare domain.

3 System design

As a part of the work on the more general task of defining and analyzing key requirements for clinical data processing and data exchange systems, a project has been developed and a pilot version 0of a semantic repository for outpatient sheets has been implemented, which covers all functionalities mentioned above. The project is oriented towards solving the following specific tasks:

- providing a convenient application interface, enabling opportunities for working with documents in real time;
- determining a technology of connecting the system with other (external) systems – graph or relational databases;
- development of an additional system that can provide means for visualization of the documents with which the main system works.

The structure of the repository management system of is shown in Fig. 1. It consists of a number of packages:

- The Commands package is the main entry point for working with the system. It implements the commands that the system can execute from the command line.
- The Converter package implements a library that takes care of working with files in XML, XSD and RDF formats, defining methods for transformations between XML files and sets of RDF triples. The library also supports operations that can retrieve subsets of triples of an RDF graph. The structures in this package are used by both the Commands and the Server package.
- The Server package contains two main parts: an implementation of a REST API server, which defines the operations that can be performed with the content of the semantic repository, and a number of static files, which are Swagger documentation of the API server. In this way, a convenient user interface is supported to manipulate the stored resources.

- The Settings package contains a description of the system settings.
- The Data Store object is a folder with the data files stored by the system.

To provide the functionalities of the system, the following workflows are implemented in a dynamic way:

- "XML" → "python dictionary" – converts XML messages to serialized python format in the form of dictionaries;
- "python dictionary" → "RDF graph" – converts python dictionaries to RDF graphs;
- "RDF graph" → "RDF subgraph" – constructs an RDF граф according to a set of criteria.

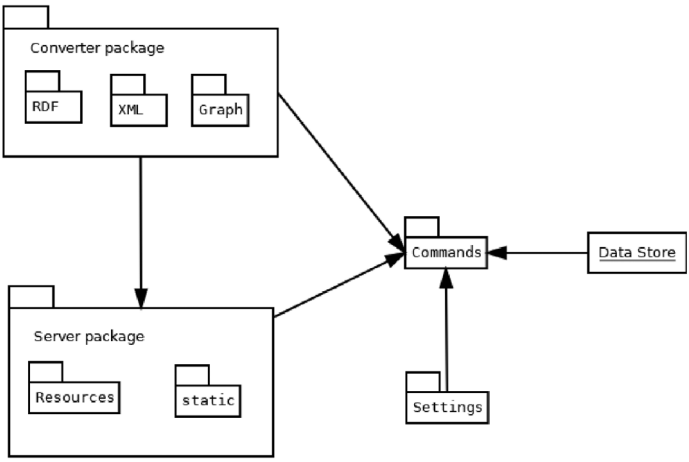

Fig. 1. Structure of the software implementation of the repository.

The auxiliary intermediate python dictionaries have the following internal representation:

```
Dict
(
  tag:        element tag,
  value:      element value,
  children:   list of children-elements,
  attributes: list of attributes
)
```

They are designed to implement the main functionalities of the system in the form of consecutive steps, passing through the selected intermediate serialized python format.

For example, for the purpose of document transformation in the "XML" → "python dictionary" direction, the XmlParser class is used, which has the

following methods:

__init__(self, schema_path)	constructor
parse(self, element)	*Reads an XML element in a dictionary structure*
build_node(self, data)	*Builds an XML element as a dictionary structure*
build(self, data)	*Builds an XML string from a dictionary structure*
validate(self, content)	*Validates an XML string*
validate_file(self, path)	*Validates the content of an XML file*
convert(self, value)	*Converts an XML string to a dictionary structure*
convert_file(self, path)	*Converts an XML file to a dictionary structure*
get_value(data, path)	*Gets the value of a dictionary element*

and for the transformation of a document in the direction "python dictionary" → "RDF graph", the RdfBuilder class is applicable with the following methods:

__init__(self, namespace, identifiers)	constructor
get_child_value(name, children)	*Gets the value of a list element given by its name*
parse_uri(self, uri)	*Converts a URI to node name and identifier*
get_children_signature(item)	*Gets the cryptographic signature of the adjacent elements of the node*
get_node(self, item, parent)	*Gets the URIRef of a graph node from its own dictionary and the parent structures of the dictionary*
parse(self, item, parent, graph)	*Converts the elements at a given level in a graph and adds URIRefs to the graph*
parse(self, data, parent, graph)	*Converts a dictionary structure and loads it into an RDF graph*

Thus, the transformation in the "XML" → "RDF graph" direction is performed in two steps using the XmlParser and RdfBuilder classes, respectively.

4 Data flow and document representation

The input data with which the system works are documents - anonymized monthly reports to the NHIF of doctors (general practitioners and specialists) for their outpatient activities. They are automatically transformed into the XML format of the NHIF for an outpatient list [4] and its equivalent RDF graphs in various serialized RDF formats [5]: N-Triple, N3, RDF / XML. In this way, the repository, and in particular the data from the outpatient sheets in it can be used for creating different types of semantically compatible healthcare information systems.

The overall data flow in the system is shown in Fig. 2.

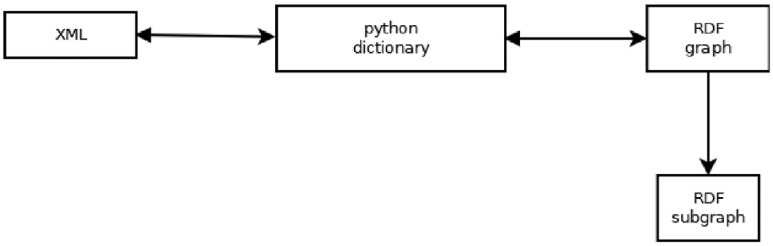

Fig. 2. Data flow in the system.

The generated RDF graphs are stored as files that describe an individual doctor as a "central point" and present multiple reports and outpatient sheets that are associated with this doctor (Fig. 3).

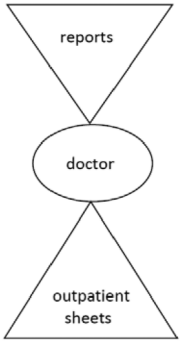

Fig. 3. Doctor-centered structure of the RDF graphs generated by the system.

Means for defining and executing the main types of SPARQL queries [6] (SELECT, CONSTRUCT, ASK, DESCRIBE) to the contents of the repository are supported. Fig. 4 shows the results of the execution of the following exemplary CONSTRUCT query:

```
CONSTRUCT
{
  ?amblist <http://amblist.com/hasDiagnose> ?diag
}
WHERE
{
   {
      ?amblist <http://amblist.com/AmbListMainDiag> ?diag.
      ?diag <http://amblist.com/MKB> ?code.
      filter regex(?code, „^E03.*", „i")
   }
```

```
    UNION
    {
       ?amblist <http://amblist.com/AmbListDiag> ?diag.
       ?diag <http://amblist.com/MKB> ?code.
       filter regex(?code, „^E03.*“, „i“)
    }
}
```

This query is intended to construct the set of available RDF triples, which integrates the outpatient sheets containing a primary or secondary diagnosis with a code beginning with E03.

	subject	predicate	object
1	http://amblist.com/AmbList/000031@yFmdCWRITIDITmdxVhgDEYbeGtsiisNIsBWirqfTP0U=	http://amblist.com/hasDiagnose	http://amblist.com/Diag/E03.8
2	http://amblist.com/AmbList/000084@yFmdCWRITIDITmdxVhgDEYbeGtsiisNIsBWirqfTP0U=	http://amblist.com/hasDiagnose	http://amblist.com/Diag/E03.8
3	http://amblist.com/AmbList/000099@yFmdCWRITIDITmdxVhgDEYbeGtsiisNIsBWirqfTP0U=	http://amblist.com/hasDiagnose	http://amblist.com/MainDiag/E03.8
4	http://amblist.com/AmbList/000130@yFmdCWRITIDITmdxVhgDEYbeGtsiisNIsBWirqfTP0U=	http://amblist.com/hasDiagnose	http://amblist.com/MainDiag/E03.8
5	http://amblist.com/AmbList/000143@yFmdCWRITIDITmdxVhgDEYbeGtsiisNIsBWirqfTP0U=	http://amblist.com/hasDiagnose	http://amblist.com/MainDiag/E03.8

Fig.4. Result of executing a CONSTRUCT query.

5 Conclusion

This paper discusses the concept and the design of a semantic repository for clinical data of patients, obtained from periodically received documents – monthly reports of doctors, providers of outpatient care. The presented pilot version of the semantic repository is ready for use. It can be integrated into various types of intelligent software systems, such as:

- information systems in the field of healthcare;
- software systems for data analysis and knowledge discovery in medical research data;
- healthcare decision support systems.

One of the next goals of our study is the development of an intelligent decision support system. It is designed to assist physicians in diagnosing and recommending treatment and diet plans for patients ill or prone to type 2 diabetes – a socially significant disease that affects a high percentage of the world's population (8.5% in the adult population according to World Health Organization 2016 data [7]). The analysis of the information about freely available medical ontologies indicates that among the most appropriate ontologies for the purposes of this study is the combination of DDO and DMTO [8]. DDO[1] is an ontology

[1] https://bioportal.bioontology.org/ontologies/DDO

for diagnosis of diabetes including knowledge about symptoms, lab tests, drugs, complications, etc. DMTO[2] is an ontology for creating customized treatment plans for type 2 diabetic patients. DMTO extends the DDO ontology by adding treatment classes and axioms to the existing diagnosis part. For this purpose, it is planned to develop a software module designed to read anonymized patient data from the repository and automatically create on their basis instances of appropriate classes of DMTO.

Acknowledgement

The research presented in this paper is supported by the National Scientific Program "eHealth" in Bulgaria.

References

1. Merrill, E., Corlosquet, S., Ciccarese, P., Clark, T., Das, S.: Semantic Web Repositories for Genomics Data Using the eXframe Platform. Journal of Biomedical Semantics 2014 5(Suppl 1):S3, doi:10.1186/2041-1480-5-S1-S3.
2. Austin, C., Bloom, T., Dallmeier-Tiessen, S. et al.: Key components of data publishing: using current best practices to develop a reference model for data publishing. International Journal on Digital Libraries 8, 77–92 (2017). https://doi.org/10.1007/s00799-016-0178-2.
3. Berg-Cross, G., Ritz, R., Wittenburg, P.: RDA Data Foundation and Terminology - DFT: Results RFC (Version 1.5). RDA DFT Working Group, 2015, https://www.rd-alliance.org/sites/default/files/DFT%20Core%20Terms-and%20model-v1-6.pdf (visited on May 20, 2020).
4. Първични медицински документи. https://www.nhif.bg/get_file?uuid=3f8425ef-5a98-469e-ac92-def5871cac37 (visited on May 20, 2020).
5. RDF Serialization Formats. https://help.poolparty.biz/pp6/developer-guide/general-information-on-the-poolparty-api/rdf-serialization-formats (visited on May 20, 2020).
6. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language, W3C Recommendation 21 March 2013. W3C, 2013, https://www.w3.org/TR/sparql11-query/ (visited on May 20, 2020).
7. World Health Organization: Global Report on Diabetes. WHO Library Cataloguing-in-Publication Data (2016), ISSN 978 92 4 156525.
8. El-Sappagh, S., Kwak, D., Ali, F., Kwak, K.: DMTO: a realistic ontology for standard diabetes mellitus treatment. Journal of Biomedical Semantics 9 (2018), https://doi.org/10.1186/s13326-018-0176-y.

[2] https://bioportal.bioontology.org/ontologies/DMTO