Towards Semantic Exchange of Clinical Documents

Juha Puustjärvi
Helsinki University of Technology
Espoo, Finland
juha.puustjärvi@tkk.fi

Leena Puustjärvi
The Pharmacy of Kaivopuisto
Helsinki, Finland
leena.puustjärvi@kolumbus.fi

Abstract— **Nowadays healthcare institutions have major problems with accessing and maintaining the large amounts of data that are continuously being generated. In addition, system interoperation is of prime importance as much of the patients' relevant information may be historic, and may have been gathered over many encounters with healthcare providers in different locations using heterogeneous healthcare information systems. In order to promote system interoperation several organizations in the healthcare sector have produced standards and representation forms using XML. However, the introduction of these XML-based technologies is not enough to provide a means to interpret the semantics of the exchanged messages. As a result, extending systems by new parties as well as introducing new message types is inconvenient. Replacing existing hard-coded medical information systems by open healthcare information systems that support semantic interoperation, are extensible, and maintainable is a challenging research problem. In this article we have restricted ourselves on this problem. In particular, we described our work on using RDF in exchanged clinical documents. Such documents themselves describe their semantics, and so they are in a machine understandable form. Hence RDF-based messaging represents an open, easily maintainable and extensible way for developing interoperable open systems.**

Keywords- e-health; open healthcare systems; semantic interoperability; ontologies.

I. INTRODUCTION

Information and communication technology has not only changed the way that clinical documents are stored and generated across and within healthcare organizations but it has also increased the efficiency and cost-effectiveness of healthcare organizations.

In particular, during the past few years the technology developed for interoperable autonomous systems has significantly changed. In particular, XML is rapidly becoming the key standard for data representation and transportation. However, the existing medical information systems that have been built during the past decades are based on proprietary solutions, developed in piecemeal way, and tightly coupled through ad hoc means [1]. These systems have many duplicated functions, and they are monolithic, non-extensible and non-interoperable [2, 3, 4, 5]. Such systems are commonly called *stovepipe systems* [6] as their components are hard-coded to only work together.

How to replace the stovepipe systems by the open healthcare information systems that support semantic interoperability, are extensible and maintainable is a challenging problem for the healthcare sector.

In our research we have focused on this problem. In particular, our focus is semantic exchange of pharmaceutical information between medical information systems. By semantic exchange we refer to the ability that the communicating parties can unambiguously (based on medicinal ontologies) interpret the exchanged messages [7]. By a medical information system [8] we refer to any system that processes medical data.

The starting point for our work has been the goal to develop an experimental infrastructure for exchanging pharmaceutical information, which satisfies the following goals: (i) The system supports semantic interoperability. (ii) Communicating information systems can independently introduce new message types. (iii) The system is open in the sense that new participating medical information systems can be easily introduced.

Whether these goals can be achieved by utilizing Semantic Web-technologies [7] is the main topic of this article. The article extends the work presented in [1].

The rest of the article is organized as follows. First, in Section II, we characterize open systems as they comprise the cornerstone of our approach. Then, in Section III, we give a short overview of the state of the art with respect to exchanging medical information. We first consider how semantic interoperability is achieved in the CDA (Clinical Data Architecture) [9] by hard-coding the semantics of the messages in communicating systems. Then, we consider the use of XML-based messaging in electronic prescription systems, and illustrate why XML-based messaging [10] requires hard-coding. In Section IV, we consider RDF-based messaging, i.e., message exchange where the messages include RDF-statements [11]. In such messages the semantics of the message is available from external sources, and so there is no need for hard-coding. The architectural and technical aspects of RDF-based messaging are considered in Section V. Finally, Section VI concludes the article by discussing the advantages and disadvantages of the deployment of the RFF-based technology in exchanging clinical documents.

II. OPEN SYSTEMS

O*pen systems* are computer systems that provide some combination of interoperability, portability, and open software standards [12]. In this article we consider open systems from interoperability point of view. By interoperability we refer to the ability of diverse systems and organizations to work together.

A. Semantic interoperability

Shared understanding of the exchanged messages can be achieved by *semantic interoperability*, which means that after data were transmitted from a sender system to a receiver, all implications made by one party had to hold and be provable by the other [12].

There are two thoroughly different approaches for achieving semantic interoperability: hard-coding and semantic messaging.

By *hard-coding* we refer to the software development practice of embedding the semantics input-messages into the application program, instead of obtaining the semantics from external sources. Hard-coding is proven to be a valuable and powerful way for exchanging structured and persistent business documents. However, if we use hard-coding in the case of non- persistent documents and non-static environments we will encounter problems in deploying new document types and extending the system by new participants.

By *semantic messaging* we refer to the practice of including the semantics of the exchanged document in a machine understandable form in the messages. Exchanging semantic messages represents an open, easily maintainable and extensible way for developing interoperable open systems.

B. Autonomy and heterogeneity in open systems

The emerging open information systems are co-operative where autonomous and heterogeneous components enable the components collectively to provide solutions. This requires that the information systems have components that cross organizational boundaries, and in this sense are open. In open systems the components are autonomous and heterogeneous, and the configuration of the whole system can change dynamically.

Fundamentally components´ autonomy means that they function under their own control. The reason for this is that the components reflect the autonomy of the organization interests that they represent. In addition there may be technical reasons for the autonomy, e.g., as a result of a hardware failure or error in a software.

In open systems heterogeneity can arise in a variety of formats, e.g., in networking protocols, in encoding information, and in used data models. Heterogeneity may also arise at semantic levels, e.g., the same concept is used for different meanings, or two different concepts are used for the same meaning. The reason for heterogeneity is historical: the components may have arisen out of legacy systems that are initially developed for local uses, but are eventually expanded to participate in open environments.

C. Document-centric Web services

SOA (Service Oriented Architecture) is an architectural design pattern that concerns with defining loosely-coupled relationships between producers and consumers [12]. It provides flexible methods for connecting information systems themselves as well as to other relevant systems. SOA relies on Web services as its fundamental design principle.

Technically Web services are self-describing modular applications that can be published, located and invoked across the Web. Once a service is deployed, other applications can invoke the deployed service.

There are two ways of using Web services: the RPC-centric view (Remote Procedure Call–centric) and the document-centric view. The *RPC-centric view* treats services as offering a set of methods to be invoked remotely while the *document–centric view* treats Web-services as exchanging documents with one another. Although in both approaches transmitted messages are XML-documents, there is a conceptual difference between these two views.

In the RPC-centric view the application determines what functionality the service will support, and the documents are only business documents on which the computation takes place. Instead the document-centric view considers documents as the main representation and purpose of the distributed computing: each component of the communicating system reads, produces, stores, and transmits documents. The documents to be processed determine the functionality of the service. Therefore, document centric view corresponds better with our goal of applying services in open environments. Furthermore, as the RDF-based messages are also represented by XML, the document-centric view suits well for semantic exchange of clinical documents.

III. EXCHANGING CLINICAL DATA

Health care systems are designed to meet the health care needs of target populations. The goals for health systems are good health, responsiveness to the expectations of the population, and fair financial contribution.

There are a wide variety of health care systems. In many countries health care system has evolved and has not been planned. Most of the health care systems that are developed during the past decades are closed. Typically they are proprietary and only serve one specific department within a healthcare institution [13]. Such standalone systems are developed by many different suppliers, and thus they are incompatible with one another. However, this is regrettable as system interoperation is crucial since much of the patients' relevant information may be historic, and may have been gathered over many encounters with healthcare providers in different locations using heterogeneous healthcare systems.

In order to improve interoperation, several organizations in the healthcare sector have produced standards and representation forms using XML. For example, patient records, blood analysis and electronic prescriptions [14, 15, 16, 17, 18] are typically represented as XML-documents. The introduction of these XML-based technologies alleviates the stovepipe problem but they are not enough to achieve semantic interoperability. Instead for achieving semantic interoperability it is necessary to provide standardized ways to describe the meanings of the exchanged XML-documents.

A. Interoperation in HL7 CDA

The Clinical Document Architecture (CDA) is an ANSI approved HL7 standard [19] for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services. "Health level seven" refers to the seventh (application) level of the International Organization for Standardization (ISO) seven-layer communications model for Open Systems Interconnections.

The HL7 Clinical Document Architecture (CDA) is a document markup standard that specifies the structure and semantics of clinical documents for the purpose of exchange. Release One (CDA R1), became an American National Standards Institute (ANSI)–approved HL7 Standard in 2000. Release Two (CDA R2) [20, 21], became an ANSI-approved HL7 Standard in 2005.

A CDA document is a defined and complete information object that can include text, images, sounds, and other multimedia content. CDA documents are encoded in Extensible Markup Language (XML), and they derive their meaning from the HL7 Reference Information Model (RIM) and use the HL7 Version 3 Data Types.

RIM is static object-oriented model in UML notation. It serves as the source from which all specialized HL7 version 3 information models are derived and from which all HL7 data ultimately receives its meaning.

HL7 is proven to be a valuable and powerful standard for a structured exchange of persistent clinical documents between different software systems. However, in the case of non persistent documents with CDA we encounter many problems.

The reason for this is that the semantics of the documents is bound to the shared HL7 Reference Information Model (RIM) [19]. The developers of the CDA-compliant systems are familiar with the RIM and use that information in developing CDA-compliant systems. That is, HL7 compliancy means that the knowledge of the relationship between the XML-elements in the received CDA document and the conceptual schema given in RIM is hard-coded in the systems receiving the messages. Therefore HL7 CDA compliant systems are able to understand each other as long as they exchange CDA-documents.

The semantics of the CDA-compliant message cannot be interpreted just based on the message and the conceptual schema given in RIM. Therefore introducing a new message-type (i.e., a CDA document) and corresponding extensions to RIM is a long lasting process requiring standardization and the modifications of the communicating software modules. As a result, applying HL7 standards to a new domain, (e.g., for pharmacy) is problematic. Therefore the solutions made in the HL7 CDA standard do not satisfy the goals of open, extensible healthcare information systems that support semantic interoperability

B. Electronic Prescriptions

Electronic prescription is the electronic transmission of prescriptions of pharmaceutical products from legally professionally qualified healthcare practitioners to registered pharmacies [22]. The scope of the prescribed products varies from country to country as permitted by government authorities or health insurance carriers.

The information in an electronic prescription includes for example, prescribed products, dosage, amount, frequency and the details of the prescriber. A simple prescription is presented in Fig. 1 in XML.

```
<prescription>
    <prescription_id>abc123</prescription_id>
    <patient>
        <name>John Smith </name>
        <id> 1465766677</id>
    </patient>
    <medicinal_product>Panadol</medicinal_product>
    <disease>fever</disease>
    <quantity>30</quantity>
    <dose>One tablet three times a day</dose>
    <physician>
        <name>Lisa Taylor </name>
        <id> 98765432</id>
    </physician>
</prescription>
```

Figure 1. A simplified prescription in XML.

C. The Semantics of Prescriptions

XML (Extensible Markup Language) is a set of rules for encoding documents electronically. XML's design goals emphasize simplicity, generality, and usability over the Internet.

Although XML-documents are commonly used for information exchange they do not provide any means of talking about the semantics (i.e., meaning) of data. For example there is no meaning associated with the nesting of the tags presented in the XML-coded prescription in Fig. 1. It is up to the applications that receive the XML-messages to interpret the nesting of the tags. Even if there is a conceptual schema or ontology [23, 24] having the modeling primitives having the same naming (e.g., class patient having attribute name) as the tags in the XML-message it is up to the application to interpret the nesting of tags. To illustrate this consider the statement:

"Physician Lisa Taylor cares for patient John Smith".

We can present this sentence by the following two nesting ways:

```
(1)  <patient name='John Smith'>
         <physician>Lisa Taylor</physician>
     </patient>

(2)  <physician name='Lisa Taylor'>
         <patient>John Smith</patient>
     </physician>
```

These formalizations include an opposite nesting although they represent the same information. Hence, there is no standard way of assigning meaning to tag nesting.

Therefore the semantics of the documents in the messages (e.g., the prescription) must be specified by binding it to a conceptual schema (ontology), e.g., to a conceptual schema presented in Fig. 2.

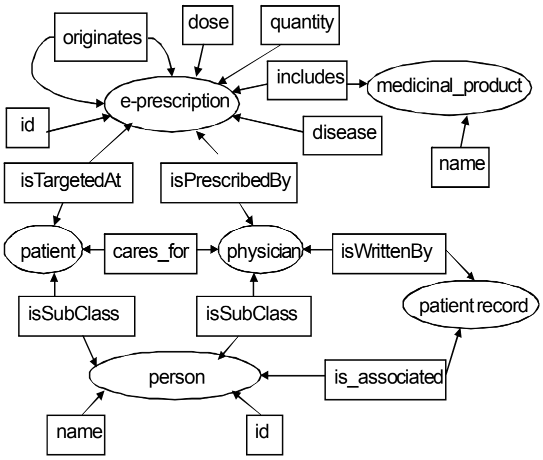

Figure 2. A medicinal ontology.

D. *Introducing New Message-Types*

In the case of hard-coded message exchange, the introduction of a new XML-message type requires that the syntax and the semantics of the message must be first standardized, and then the communicating systems' Web services can be updated by a new message type, i.e., the semantics of the messages can be hard-coded to the communicating applications.

In order to illustrate the problems of such hard-coded solutions, assume that the communicating medicinal systems do not only exchange electronic prescriptions but also renewed prescriptions. A renewed prescription deviates from other prescription in that it equals with the original prescription with respect to medicinal product but may deviate with respect to prescribing physician, quantity and dose. Such a renewed prescription of the prescription of Fig. 1 is presented in Fig. 3.

```
<prescription>
        <originates_id>abc123</originates_id>
        <patient>
                <name>John Smith </name>
                <id> 1465766677</id>
        </patient>
        <medicinal_product>Panadol</medicinal_product>
        <disease>fever</disease>
        <quantity>50</quantity>
        <dose>Two tablet three times a day</dose>
        <physician>
                <name>Paul Goodman </name>
                <id> 66765555</id>
        </physician>
</prescription>
```

Figure. 3. A renewed prescription presented in XML.

In order that the communicating medical information systems (e.g., electronic prescription writer, medical expert system, medical database system and electronic prescription holding store) would understand the syntax and semantics of the renewed prescription the structure of the XML-document should be standardized and its semantics should be specified by the conceptual schema. In addition, the semantics of the renewed prescription should be hard-coded in communicating information systems.

Another approach for deploying renewed prescriptions is that the renewed prescription message itself describes its semantics, and hence no message standardization process is needed. How this can be done by RDF is the topic of the next sections.

IV. RDF-BASED MESSAGING OF MEDICINAL DATA

A. *RDF-Based Prescriptions*

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is also a data model. Principally the RDF data model is not different from classic conceptual modeling approaches such as Entity-Relationship or Class diagrams, as it is based upon the idea of making statements about resources. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources.

RDF provides a simple language in which to capture knowledge. It incorporates a number of well-known ideas from knowledge representation. RDF is built on top of the Web notion of a URI (Universal Resource Identifier). URIs need not be absolute in that they need not correspond to the name of any actual object to be accessed via any specific protocol.

RDF's modeling primitive is an object-attribute-value triple, which is called a statement [7]. For example, the preceding sentence "Physician Lisa Taylor cares for patient John Smith" is such a statement.

There are various ways in capturing knowledge with RDF, e.g., as natural language sentence as above, in a simple triple notation called N3, in RDF/XML serialization format, and by as a graph of the triples [7]. In Fig. 4, the prescription of Figure 1 is presented as a graph of triples, whereas in Fig. 5 it is presented in RDF/XML serialization format.

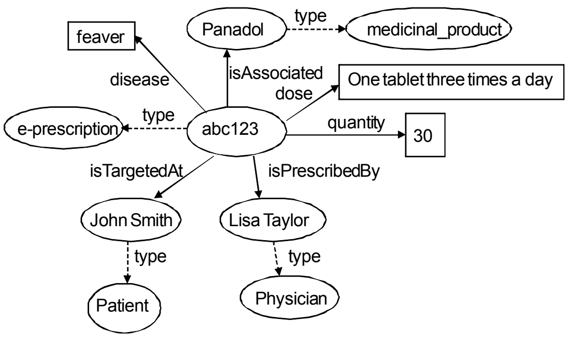

Figure 4. RDF-based prescription in a graphical form.

```
<rdf:RDF
    xmlns : rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns : xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns : mo="http://www.lut.fi/ontologies/montology#"
    <rdf:Description rdf:about="abc123">
      <rdf:type rdf:resource="&mo;e-prescription"/>
         <mo : dose>One tablet three ti mes a day</mo : dose>
         <mo : quantity rdf:datatype="&xsd;integer">30</mo : quantity>
         <mo: includes>Panadol</mo: includes>
    </rdf : Description>
    <rdf:Description rdf:about="1465766677">
      <rdf:type rdf:resource="&mo;patient"/>
         <mo : name>John Smith</mo : name>
    </rdf : Description>
    <rdf:Description rdf:about="98765432">
    <rdf:type rdf:resource="&mo;physician"/>
         <mo : name>Lisa  Taylor</mo : name>
    </rdf : Description>
</rdf:RDF>
```

Figure 5. An electronic prescription in RDF-format.

B. *RDF-Schema and RDF-Typing*

RDF is domain-independent in that no assumptions about a particular domain of use are made. It is up to users to define their own domain specific terminology (vocabulary) by RDF Schema (RDFS) [7].

Our defined medicinal vocabulary (ontology) includes concepts patient, physician, e-prescription and patient record as well as their relationships. Basically it deviates from the ontology presented in Figure 2 in that it is presented by RDFS. Using the medicinal vocabulary we can state for example "Physician Lisa Taylor cares for patient John Smith" in a machine understandable way. Particularly by using the RDF-type element we tie the subject, predicate and the object of the statement "Physician Lisa Taylor cares for patient John Smith" to the RDF Schema. To illustrate this consider Fig. 6, which includes a subset of the ontology presented in Fig. 2 and the RDF-statement "Physician Lisa Taylor cares for patient John Smith".

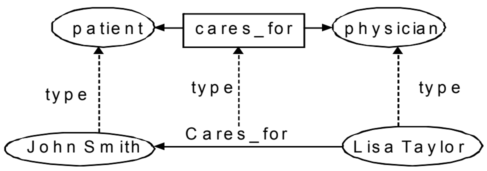

Figure 6. Typing an RDF-statement by RDFS.

V. THE ARCHITECTURE OF THE COMMUNICATING MEDICAL INFORMATION SYSTEMS

A. *The Components of the Architecture*

In our used architecture medical information systems communicate through Web services by the SOAP -protocol. The semantic exchange of clinical documents is carried out by the SOAP-messages [12], which include RDF-statements. The components of the architecture are presented in Fig. 7.

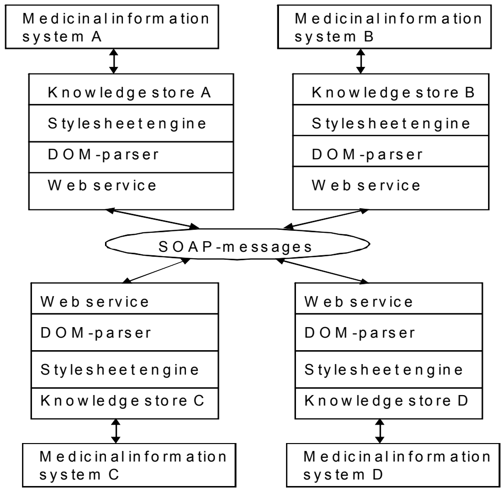

Figure 7. The components of the communicating systems

We next consider the components of the architecture from technology point of view.

B. *Web services*

As we have already stated the document-centric view of Web services suits for our purposes. The implementation of document-centric Web services of a prescription holding store is illustrated in Fig. 8. Here the Web service supports three kinds of requests: e-prescription requests, requests on patient's records and requests on patients' prescriptions. Each type of request is presented by specific document that is presented in RDF. However, the Web service does not support separate operations for these requests but rather a single operation, which just receives the documents and stores them in the Knowledge store. Further, processing the requests is the function of the Prescription management application.

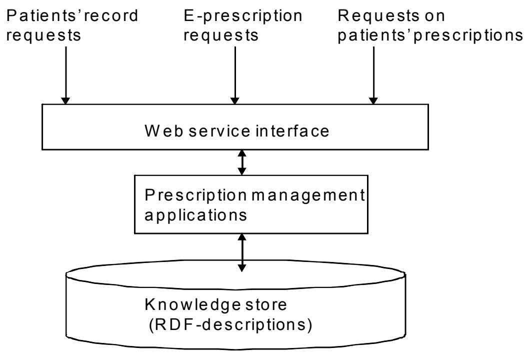

Figure 8. The structure of a Document-centric Web service.

A consequence of our used document-centric view is that we have to model the requests in the ontology of the Knowledge store; otherwise we could not store and retrieve the requests.. As the schema of the Knowledge store is specified by RDFS [7, 11], we have to model the requests also in RDFS. That is, we have RDFS class Request and its subclasses E-prescription request, Request on patient's record and Request on patients' prescription.

C. *SOAP-messaging*

SOAP was originally intended to provide networked computers with remote-procedure call services written in XML. It has since become a simple protocol for exchanging XML-messages over the Web.

A SOAP-message is comprised of a SOAP header, SOAP envelope and SOAP body. In particular, the SOAP body contains the application-specific message that the backend application will understand. As illustrated in Fig. 9, we incorporate our used RDF-formatted clinical documents in the SOAP body.

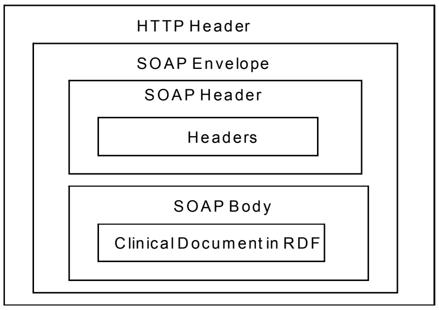

Figure 9. RDF-formatted clinical document in a SOAP-message.

An example of XML-coded SOAP-message which contains an RDF-formatted clinical document is presented in Fig. 10.

```
<SOAP-ENV: Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encodig/">
    <SOAP-ENV:Body>
      <clinical-document>
         <rdf:RDF
           xmlns : rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns : xsd="http://www.w3.org/2001/XMLSchema#"
           xmlns : mo="http://www.lut.fi/ontologies/montology#"
          <rdf:Description rdf:about="abc123">
             <rdf:type rdf:resource="&mo;e-prescription"/>
             <mo : dose>One tablet three ti mes a day</mo:dose>
             <mo : quantity rdf:datatype="&xsd;integer">30</mo:quantity>
             <mo: includes>Panadol</mo: includes>
           </rdf : Description>
           <rdf:Description rdf:about="1465766677">
               <rdf:type rdf:resource="&mo;patient"/>
               <mo : name>John Smith</mo:name>
            </rdf : Description>
            <rdf:Description rdf:about="98765432">
               <rdf:type rdf:resource="&mo;physician"/>
               <mo : name>Lisa  Taylor</mo:name>
            </rdf : Description>
         </rdf:RDF>
      </clinical-document>
    </SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

Figure 10. An RDF-formatted prescription in a SOAP-message

The RDF-coded clinical document of Fig. 10 is the prescription presented in Fig. 5. The namespaces "mo" specifies the used ontology. That is, the namespace "mo" refers to the URL where the ontology of Fig. 2 is stored in RDFS.

D. *Processing exchanged clinical documents*

In order that the medicinal information systems are able to handle the clinical documents of the SOAP-messages they have to use the DOM-parser and the Stylesheet engine. The DOM parser transforms input text (i.e., RDF-statements) into a tree, which is suitable for the Stylesheet engine to process. DOM (Document Object Model) [6] refers to a language-neutral data model and application programming interface (API) for programmatic access and manipulation of XML-coded data. Generally, parsing (also called syntactic analysis) is the process of analyzing a sequence of tokens to determine its grammatical structure with respect to a given formal grammar.

As illustrated in Fig. 11, the Stylesheet engine takes the RDF-document from the DOM-parser, loads it into a DOM source tree, and picks out the needed information by transforming the RDF-document with the instructions given in the style sheet.

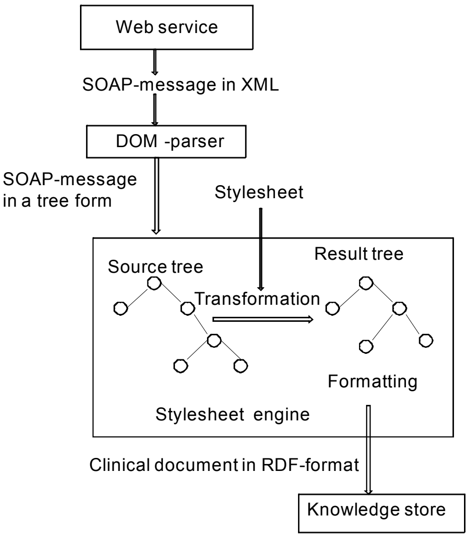

Figure 11. Transforming the representation formats.

In transforming the source tree the Stylesheet engine use XPath [6] expressions to reference portions of the tree and capture information to place it into the result tree. The result tree is then formatted, and the resulting RDF-document is stored in the Knowledge store.

E. *Knowledge store*

In our used architectural terminology Fig. 6 represents a overly simplified knowledge store in the sense that it includes and ontology represented in RDFS and one RDF-statement. In reality the knowledge stores have much wider ontology and thousands or millions of RDF-statements.

A salient feature of our used architecture is that the communicating medical information systems maintain their own knowledge store by picking out the interested knowledge from the messages they receive (i.e., RDF-statements) and then storing that knowledge to their own knowledge store.

To motivate this kind of message exchange strategy assume that the medical information system A sends a prescription to medical information systems B, C and D. These medical information systems may have different interests on the prescription.

For example, assuming that system B represents a pharmacy, so it is needs all the information in the prescription. On the other hand, assuming that system C represents government authorities, then it is obvious the system does not need information concerning the dose of the medicinal products; and assuming that system D represents health insurance authorities, then the system needs only the information of the patient and the prices of the medicinal products included in the prescription.

That is, each medicinal system has its own interest on the prescription, and they will only store in their knowledge store that part of the prescription. As illustrated in Fig. 11 the part on which a system has its interest is specified by the stylesheet it uses.

VI. CONCLUSIONS

Today healthcare institutions have major problems with accessing and maintaining the large amounts of data that are continuously being generated. At the same time the recent developments in the field of information technology have promised to bring improvements in the quality of managing and exchanging medicinal information. Also the technology developed for interoperable autonomous systems has significantly developed giving chances for implementing open healthcare information systems, which are easily extensible and maintainable.

In particular, the technology developed for interoperable autonomous systems has significantly changed. XML is rapidly becoming the key standard for data representation and transportation. However, the existing medical information systems that have been built during the past decades are based on proprietary solutions, developed in piecemeal way, and tightly coupled through ad hoc means

In this article, we have considered how to replace the hardcoded medical information systems by the open healthcare information systems that support semantic interoperability, and which are easily extensible and maintainable.

Semantic interoperability is the ability of computer systems to communicate information and have that information properly interpreted by the receiving system in the same sense as intended by the transmitting system. Semantic interoperability requires that any two systems will derive the same inferences from the same information.

The corner stone of our approach in achieving semantic interoperation is the medicinal ontology on which the communicating medical information systems have to commit in their mutual communication, i.e., the used medicinal ontology must be shared and consensual terminology as it is used for information sharing and exchange. It, however, does not suppose the introduction of a universal ontology for the healthcare sector. This situation is analogous with natural languages: a pharmacy, or any medicinal organization, may communicate in Finnish with medicinal authorities and in English with pharmaceutical companies. Just as there is no universal natural language, so there is no universal ontology.

A challenging situation for the health care organizations is also the introduction of new technologies. The introduction of semantic interoperation in healthcare sector is challenging as it incorporate semantic web technologies into many part of the work life cycle, including information production, presentation, analysis, archiving, reuse, annotation, searches and versioning. The introduction of these technologies also changes the daily duties of the many ICT-employees of the organization. Therefore the most challenging aspect will not be the technology but rather changing the mind-set of the ICT-employees and the training of the new technology.

The introduction of a new technology is also an investment. The investment on new Semantic Web-technology includes a variety of costs including software, hardware and training costs. Training the staff on Semantic Web-technology is a big investment, and hence many organizations like to cut on this cost as much as possible. However, the incorrect usage and implementation of a new technology, due to lack of proper training, might turn out to be more expensive in the long run.

REFERENCES

[1] J. Puustjärvi, and L. Puustjärvi. Semantic Exchange of Medicinal Data: a Way Towards Open Healthcare Systems. In the proc. of the Third International Conference on Digital Society (ICDS 2009), p.168-173.

[2] C. Liu, A. Long, Y. Li, K. Tsai, and H. Kuo, "Sharing patient care records over the World Wide Web", International journal of Medical Informatics, 61, 2001, p. 189-205.

[3] R. Batenburg, and E. Van den Broek, "Pharmacy information systems: the experience and user satisfaction within a chain of Dutch pharmacies", International Journal of Electronic Healthcare. Vol. 4, No.2 , 2008, p.119-131.

[4] K. Khoumbati, S. Shah, Y.K. Dwivedi, and M.H. Shah", Evaluation of investment for enterprise application integration technology in healthcare organisations: a cost-benefit approach", International Journal of Electronic Healthcare. Vol. 3, No.4, 2007, p.453-467.

[5] M.S. Raisinghani, and E. Young, "Personal health records: key adoption issues and implications for management", International Journal of Electronic Healthcare. Vol. 4, No.1, 2008, p.67-77.

[6] M. Daconta, L. Obrst, and K. Smith. The semantic web. Indianapolis: John Wiley & Sons. 2003.

[7] G. Antoniou, & F. Harmelen. A semantic web primer. The MIT Press. 2004

[8] F. Jung, "XML-based prescription drug database helps pharmacists advise their customers", http://www.softwareag.com/xml/aplications/sanacorp.htm, 2005

[9] HL7 Overview. http://www.interfaceware.com/manual/hl7.

[10] E. Harold, and W. Scott Means W., XML in a Nutshell. O'Reilly & Associates, 2002.

[11] J., Davies, D. Fensel and F. Harmelen., Towards the semantic web: ontology driven knowledge management. West Sussex: John Wiley & Sons.2002.

[12] M. Singh, & M. Huhns. Service Oriented Computing: Semantics, Processes, Agents. John Wiley &Sons, Ltd. 2005.

[13] J. Puustjärvi, and L. Puustjärvi. Managing Medicinal Instructions. In the proc. of the International Conference on Health Informatics (HEALTHINF 2009), p.105-110.

[14] P. Woolman, XML for electronic clinical communication in Scotland. International journal of Medical Informatics, 64, 2001, p. 379-383.

[15] G. Stalidis, A. Prenza, N. Vlachos, S. Maglavera, D. Koutsouris, "Medical support system for continuation of care based on XML web technology", International journal of Medical Informatics, 64, 2001, p. 385-400.

[16] R. Keet, "Essential Characteristics of an Electronic Prescription Writer", Journal of Healthcare Information Management, vol 13, no 3.1999.

[17] J. Puustjärvi, and L. Puustjärvi "The challenges of electronic prescription systems based on semantic web technologies", In Proc. of the 1st European Conference on eHealth (ECEH'06). pages 251-261. 2006.

[18] J. Puustjärvi, and L. Puustjärvi, "Automating the coordination of electronic prescription processes", In Proc. of the 8th International Conference on e-Health Networking Applications and Services (HealthCom2006). p. 147-151. 2006.

[19] R.H. Dolin, L. Alschuler; C. Beerb, P.V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J.E. Mattison. "The HL7 Clinical Document Architecture", J. Am Med Inform Assoc 2001:8(6), p.552-569.

[20] Robert H. Dolin, MD, Liora Alschuler, Sandy Boyer, BSP, Calvin Beebe, Fred M. Behlen, Paul V. Biron and Amnon Shabo (Shvo), HL7 Clinical Document Architecture, Release 2, http://www.jamia.org/cgi/content/abstract/13/1/30

[21] Gerdsen F, Müller S, Jablonski S, Prokosch HU. Standardized exchange of clinical documents--towards a shared care paradigm in glaucoma treatment.. Methods Inf Med. 2006;45(4):359-66.

[22] J. Puustjärvi, and L. Puustjärvi, "Developing an application integration strategy for electronic prescription system", In proc. of the International Workshop on Semantic Information Integration on Knowledge Discovery (SIIK2006), p. 253-262. 2006

[23] M. Gruber, Thomas R., "Toward principles for the design of ontologies used for knowledge sharing", Padua workshop on Formal Ontology, 1993.

[24] E. Mattocks, "Managing Medical Ontologies using OWL and an e-business Registry / Re-pository", http://www.idealliance.org/proceedings/xml04/papers/85/MMOOEBRR.html, 2005.