On the Structural and Semantic Interoperability of the Electronic Healthcare Records

Mustafa Yuksel and Asuman Dogac

***Abstract*— Interoperable cross-border clinical data exchange is an ambitious goal with several challenges, the most prominent ones being the structural differences in the EHR standards and the different terminologies used for semantic encodings of the data elements used in the EHRs.**

In this article, we address the structural interoperability problem of the EHRs by adapting the UN/CEFACT Core Component Technical Specification to define a mapping among the document components of EHRs as specified through HL7 CDA R2 and CEN EN 13606-1 standards. Then in order to address the semantic interoperability, we describe how to use the HL7 Common Terminology Services (CTS) to translate the codes from different code systems. For this purpose, we give implementations of CTS functions in terms of the Web service operations of one of the prominent terminology servers, namely, UMLSKS. Finally, we describe a method to map locally developed codes to their international counterparts.

***Index Terms*— Electronic Healthcare Records, National Health Information System, semantic interoperability, HL7 CDA, CEN EN 13606**

I. Introduction

An Electronic Healthcare Record (EHR) is defined as "digitally stored health care information about an individual's lifetime with the purpose of supporting continuity of care, education and research, and ensuring confidentiality at all times" [1]. There are several EHR content standards which aim to provide standard interfaces to existing proprietary systems. The most prominent efforts include the Health Level Seven Clinical Document Architecture [2], CEN EN 13606 Electronic Health Record Communication [3] and openEHR [4]. Such standards define the structure and markup of the clinical content. In [5], an extensive survey and analysis of EHR standards are presented.

In exchanging EHRs, because EHRs contain clinical information critical for patient safety, there is a general agreement that the EHRs must contain human readable text. For example, a valid HL7 CDA document cannot contain authenticated entries without textual representation [6].

On the other hand, CDA has a very powerful information model that is derived from the HL7 RIM. HL7 RIM is, in fact, developed to provide machine to machine interoperability of exchanged messages in the healthcare domain. By the help of RIM entry classes such as Act, Observation or SubstanceAdministration and supporting code systems, many complicated clinical statements can be represented in a CDA document in a fully machine-understandable way. For instance, by using the "SubstanceAdministration" class, it is possible to model even the most detailed prescription. Furthermore, new eHealth applications are emerging that necessitate to use the EHRs for machine to machine interoperability. The example applications include sharing the data collected from the medical sensors [7], [8]; sharing the "Emergency Data Set"s of patients containing mainly coded elements showing the current allergies and medications [9], [10]; gathering statistics from the coded elements in EHRs to be used in decision support systems for efficient management and monitoring of national healthcare policies [11]; updating personal healthcare records through EHRs [12]; and bio-surveillance by checking the critical coded elements in EHRs [13].

Additionally, today many countries or regions are developing their national or regional electronic healthcare infrastructures with the ultimate aim of sharing the EHRs of the patients across borders or regions. For example, in the USA, sharing EHRs through National Health Information Network (NHIN) [14] by providing interoperability among the Regional Health Information Organizations (RHIOs) has become a priority. Similar efforts include the Health Infoway in Canada [15], the NHS Connecting for Health in the United Kingdom [16], the Dossier Medical Personnel (DMP) in France [17] and the National Health Information System (Saglik-Net) in Turkey [11].

The EHR exchange brings up the following challenges:

- *The interoperability of EHR structures*: The first problem in cross border EHR schema interoperability is that even when the countries use the same standard, there can be many different ways of organizing the clinical information: the same information can be expressed through different document components and the components can be nested differently.
  In order to address this problem, we propose to adapt the UN/CEFACT Core Component Technical Specification (CCTS) [18] to the eHealth domain. CCTS recommends to build the documents starting from the basic core components and organizing them to aggregate core components in the business domain. This corresponds to building a Health Data Dictionary defining the commonly used healthcare data elements and creating Minimum Health Data Sets from these data elements in the eHealth domain. This approach standardizes the basic components and how they are composed to create the Minimum Health Data Sets. Finally, the related Minimum Health Data Sets can be composed to create the EHRs.

Mustafa Yuksel is with the Software Research, Development and Consultancy Ltd., Ankara, Turkey; Asuman Dogac is with the Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey.

This work is supported by the European Commission through IST-027065 RIDE project and in part by the Ministry of Health, Turkey.

The second problem is the use of different EHR standards. In order to address this problem, we identify the correspondences between the basic document components of the two prominent EHR standards namely, HL7 v3 CDA R2 and CEN EN 13606-1 and then describe how to transform a source EHR schema in one standard into a target EHR schema in another standard by using Extensible Stylesheet Language Transformation (XSLT) rules [19].

- *Semantic Interoperability*: Semantic interoperability implies that not only the structure of the EHRs, but the exchanged content is interoperable. In the healthcare domain semantic encoding of the data elements is achieved by using coded terms from code systems like LOINC [20] or SNOMED CT [21]. Different countries may be using different code systems and may additionally have local code systems. Therefore, even when the document schema can be processed without a problem at the receiving end, the terms used may be undecipherable.
  To be able to map different code systems, we propose to use HL7 Common Terminology Services (CTS). In order to access the existing terminology servers using CTS, CTS interfaces must be mapped to the interfaces of the terminology servers. For this purpose, we give implementations of CTS functions in terms of the Web Service operations of one of the prominent terminology servers, namely, Unified Medical Language System Knowledge Source Server (UMLSKS) [22]. Finally, we describe a method to map locally developed codes to their international counterparts.

Cross border interoperability of EHRs also involve the communication and transport protocols, patient identifiers, the professional identifiers as well as security and privacy provisions. However, within the scope of this article, we focus on the EHR schema and semantic interoperability.

The paper is organized as follows: Section II summarizes the enabling technologies and the standards. In Section III, the interoperability of the EHRs are addressed at the structural level and the use of the UN/CEFACT CCTS approach to handle structural differences in the EHRs conforming to the same standard is described. In order to address the interoperability of different EHR standards, the correspondences between the basic document components of the two prominent EHR standards namely, HL7 v3 CDA R2 and CEN EN 13606-1, are identified and then the required transformations are realized by using Extensible Stylesheet Language Transformation (XSLT) rules. In Section IV, we address the interoperability of EHRs at the semantic level by describing the implementation of CTS functions in terms of the Web service operations of UMLSKS [22]. Section V presents a summary of experimental results. Finally, Section VI concludes the article.

II. Enabling Standards and Technologies

This section briefly introduces the standards and technologies used in this work.

A. HL7 CDA

HL7 Clinical Document Architecture (CDA) is a document markup standard that specifies the structure and semantics of a clinical document (such as a discharge summary or progress note) for the purpose of exchange [2]. By definition, a valid CDA document is encoded in Extensible Markup Language (XML) and validates against the CDA Schema once all extensions have been removed from the instance. However, CDA Schema cannot validate every aspect of conformance because XML Schema is incapable of describing the full abstract CDA Reference Information Model (RIM) [23] based model. Conformance to the CDA specification is best tested using a Model Interchange Format (MIF) based validation tool [24] given that the MIF is a full expression of the abstract CDA model [25].

A CDA document has two main parts, the header and the body. So far, HL7 has released two versions of CDA. In the CDA Release 1 (R1), only the header part is derived from the RIM. In the CDA Release 2 (R2), in addition to the header part, the clinical content in the document body is also derived from the RIM. Therefore the CDA R2 model enables the formal representation of clinical statements through CDA Entry classes, namely, Act, Observation, ObservationMedia, SubstanceAdministration, Supply, Procedure, RegionOfInterest, Encounter, Organizer.

A CDA header provides information on the authentication, the encounter, the patient, and the involved providers whereas the CDA body includes the clinical report. The body part can be either an unstructured blob or a structured hierarchy which involves one or more section components. Within a section, narrative blocks and CDA entries are defined. Machine-processable clinical statements are represented by these CDA entries whereas the narrative blocks are human readable forms.

B. CEN EN 13606

CEN EN 13606, also known as Electronic Health Record Communication (EHRcom) consists of five parts:

1) *Reference Model* specifies the information architecture of the EHR data exchanged between systems and services.
2) *Archetypes Interchange Specification* specifies the Archetype Model and the language which are used to constrain the data.
3) *Reference Archetypes and Term Lists* presents some reference archetypes and specifies the code lists which are used with the standard.
4) *Security* specifies the privileges and regulations necessary to access data.
5) *Interface Specification* defines a set of interfaces by which the artifacts defined in 13606 Parts 1, 2 and 4 may be requested and provided.

Currently, Parts 1 to 4 are published standards and Part 5 is at the final stages of the balloting phase [26]. The root class of CEN EN 13606-1 [3] is the "EHR_Extract" for representing part or all of the EHRs of a single patient. It contains zero or more "Folder"s while "Folder"s may have nested "Folder"s. A "Composition" resides in a "Folder" via links or directly in the

"EHR_Extract". "Section"s are contained in "Composition"s while "Section"s may have nested "Section"s. An "Entry" is either contained within a "Composition" or located under a "Section". An "Entry" contains "Element" instances which are optionally contained within a "Cluster" hierarchy. The leaf node of the "EHR_Extract" hierarchy is the "Element" class which holds a single data value.

Part 3 of the 13606 standard also provides mappings for some of the HL7 Clinical Statement classes such as Act, Procedure and Supply.

C. Unified Medical Language System (UMLS)

The Unified Medical Language System (UMLS) [27] is a controlled compendium of many medical vocabularies, also providing a mapping structure between them. One of the main knowledge components of UMLS is the Metathesaurus that forms the base of the UMLS and comprises over one million biomedical concepts and five million concept names, all of which are from over a hundred controlled vocabularies and classification systems such as ICD10 [28], SNOMED [21] and LOINC [20]. The purpose of the Metathesaurus is to provide a basis of context and inter-context relationships between these various coding systems and vocabularies to provide a common basis of information exchange among the variety of clinical systems. UMLS also provides secure Web Services for querying its Knowledge Source Server (UMLSKS) for finding clinical concepts, retrieving concept details and mapping clinical terms.

D. XSL Transformations

Extensible Stylesheet Language Transformations (XSLT) [19] is an XML-based language used for the transformation of XML documents into other XML or "human-readable" documents such as HTML or plain text. A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source tree. A template is instantiated to create part of the result tree. The result tree is separate from the source tree. The most recent version of XSLT is XSLT 2.0 [29].

III. Providing the Interoperability of the EHRs at the Structural Level

Electronic Healthcare Records are "documents" describing the patient related healthcare data and, as such, they are organized into document components such as sections, entries and data elements. Indeed, the prominent EHR standards like HL7 CDA and 13606-1 structure the clinical documents from basic document components like "Sections" and "Entries" in HL7 CDA or "Compositions", "Sections" and "Entries" in 13606-1 and nesting these basic components as needed.

Clearly, there can be many different ways of organizing the same clinical information even when the same EHR standard is used: the same information can be expressed through different components and the components can be nested differently. Hence, adhering to an EHR standard interface can make the EHR content machine processable but not interoperable unless a common schema is used. Although it is possible to resolve such structural differences by providing the detailed mappings among the XML schemas, this is a very tedious and expensive process requiring human expertise and labor.

A. Addressing the Structural Differences in the same EHR Standard

We propose to prevent such structural differences in the EHRs by taking the UN/CEFACT Core Component Technical Specification (CCTS) [18] approach in constructing the EHRs. CCTS provides a methodology to identify a set of reusable building blocks, called *Core Components* which represent the common data elements of business documents such as "Address", "Amount", or "Line Item". These reusable building blocks are then assembled into business documents such as "Order" or "Invoice".

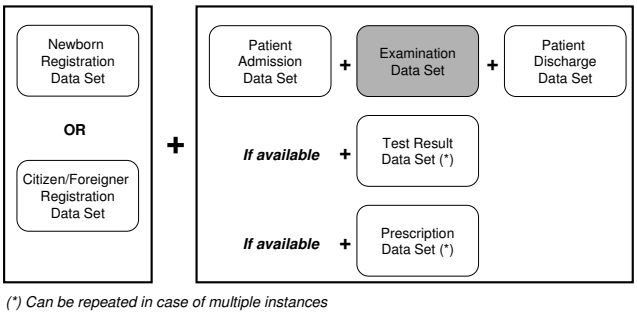

Fig. 1. The Examination EHR of NHIS, Turkey

This approach is applicable to the healthcare domain and is in fact applied in realizing the National Health Information System (NHIS), Turkey. Turkey started by first building its National Health Data Dictionary (NHDD) [30] defining the commonly used healthcare data elements such as Address, Name, Main Diagnosis, Vaccination, and Treatment Method. In defining these data elements, HL7 Data Types are used [31]. Then, the Minimum Health Data Sets (MHDSs) are formed using these data elements. Some example MHDS are: Citizen/Foreigner Registration MHDS, Medical Examination MHDS, Prescription MHDS, Pregnant Monitoring MHDS, Cancer MHDS and Inpatient MHDS. Currently, there are 261 data elements and 46 Minimum Health Data Sets in the NHDD.

The EHR standard used in NHIS, Turkey is HL7 CDA R2. Apart from the Patient Registration Data Sets which correspond to "recordTarget" participant of CDA, all the Minimum Health Data Sets correspond to HL7 CDA sections. In other words, by combining the related Minimum Health Data Sets, different EHRs are created. Figure 1 shows the "Examination" EHR which is composed by re-using the related Minimum Health Data Sets: first, for the patient demographics information either the "Newborn Registration Data Set" or the "Citizen/Foreigner Registration Data Set" is used. "Examination Data Set" is complemented by the "Patient Admission Data

Set" and the "Patient Discharge Data Set". When available "Examination" EHR contains one or more "Test Result Data Set" and "Prescription Data Set".

In this way, that is, by specifying both the EHR schemas and their basic components as local standards, Turkey has achieved structural interoperability of the EHRs exchanged nationwide. Furthermore, NHIS, Turkey is open to extensions: Whenever there is a need for a new EHR document, the existing Minimum Health Data Sets are used; the new Minimum Health Data Sets are constructed by using the existent Data Elements and whenever the need arises, the NHDD is expanded by defining the new Data Elements.

B. Addressing the Structural Differences among Different EHR Standards

Different countries, different regions within the same country or different healthcare organizations within the same region may be using different EHR standards. In this section, we address the structural differences between the two prominent EHR standards, namely, HL7 CDA R2 and CEN EN 13606-1. We describe how to map them and automatically generate 13606-1 instances from CDA instances. In order to generate conformant 13606-1 instances from HL7 CDA R2 instances, first we define a mapping between the building blocks of CDA and CEN EN 13606-1.

In 13606-1, the "EHR_Extract" is used to represent part or all of the health record information of a single subject of care extracted from an EHR provider system for the purposes of communication. "Folder" enables high-level logical grouping and organization of the "Composition"s. A "Composition" corresponds to a single clinical session of record interaction by definition [32]. Considering this 13606-1 document component structure semantics, an HL7 ClinicalDocument may correspond to an 13606-1 "Folder" or "Composition" and the first-level CDA sections correspond to 13606-1's "Composition"s. In Figure 2, the mapping between HL7 CDA R2 and CEN EN 13606-1 components used in this work are given. Note that here, the CDA instances used are valid against both the CDA Schema and the MIF based validation tool [24]. The label "link" in the figure indicates that the source class on the left side is mapped to two classes on the right side: the first class stores the actual content of the source class and the second one holds a link to the first. For example, the actual content of "recordTarget" in CDA is held in a "demographic_extract" in 13606-1. To state that this "demographic_extract" keeps the information about the patient, it is linked by the "subject_of_care" element.

A 13606-1 "Section" represents the clinical data within a "Composition" such as "abdominal examination" or "reason for encounter" and hence corresponds to a "Section" of CDA. A CDA "Section" contains "Entry" classes and an "Entry" class represents the clinical statement class of the Reference Information Model [23]. Examples include observations, evaluations, or a prescribed drug. 13606-1 "Entry"s directly map to the CDA "Entry" classes (Act, Observation, SubstanceAdministration, etc.) which are also termed as clinical statements. CEN 13606-3, which describes the "Reference Archetypes and

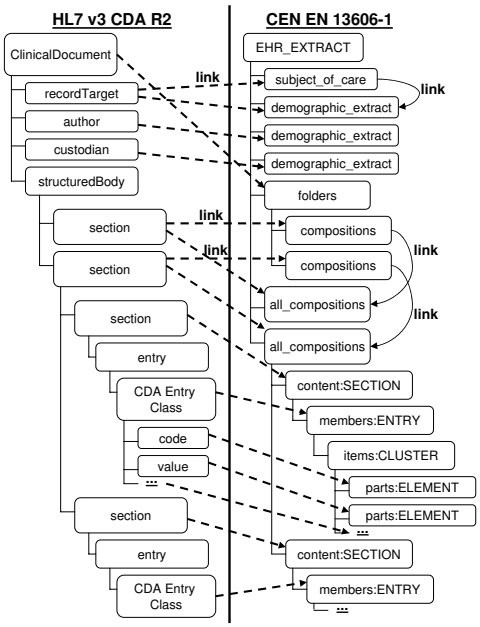

Fig. 2. Mapping HL7 CDA R2 to CEN EN 13606-1

Term Lists" [33], provides information on how to map the HL7 entry classes (Act, Observation, Procedure, SubstanceAdministration, Supply and Encounter) to CEN 13606 classes where an HL7 Entry class maps to a combination of "Entry", "Cluster" and "Element" classes of 13606-1.

In transforming the CDA "Entry" classes to CEN 13606, the mapping definitions in 13606-3 are used. However, the two CDA "Entry" classes, namely, the "Consumable" and the "Organizer" classes are missing from the mapping definitions and therefore extensions are made to cover these classes. In 13606-1, "Element" is the leaf node of the EHR hierarchy containing a single value. "Element"s can optionally be grouped within a "Cluster" to form multi-part data structures. These two classes correspond to attributes and elements of the CDA "Entry" classes.

In 13606-1, "demographic_extract" element is used to store information about the participant of the EHR, including the patient, the author or the healthcare organization. Therefore participants such as "author", "recordTarget", "indirectTarget" and "custodian" in CDA map to this element.

Once this mapping is completed, the second step is to develop the necessary XSLT rules for automatic transformation of CDA instances to valid 13606-1 instances. However, unlike HL7, CEN/TC251 does not provide a machine-processable schema for 13606 reference model. CEN/TC251 models the 13606-1 Reference Model as a UML class diagram and it is clearly stated in the normative content of the standard specification that just being compatible with this class diagram is necessary for conformance. There are some efforts to develop the XML Schema Definitions (XSD) of 13606-1 such as the LinkEHR XSDs [34]. We used the LinkEHR XSDs with some extensions for the missing parts. The developed CEN 13606 XSDs and the complete XSLT file are available in our Web page [35].

In order to provide insight to the transformation process, we present a simple example for the transformation of a CDA entry class, namely observation, to 13606-1. An example observation instance is presented in Figure 3. This instance has four elements two of which are simple attributes. The “classCode” declares that this is an Observation class. The “moodCode” value “EVN” states that this observation is an already completed event. Then the instance has two elements, “code” for annotating the meaning of the observation, which is “Main diagnosis” from SNOMED-CT and “value” for presenting this diagnosis as a coded value from ICD-10 meaning “malaria”.

```
<observation classCode="OBS" moodCode="EVN">
  <code code="8319008" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMEDCT"
   displayName="Main diagnosis"/>
  <value code="B54" codeSystem="2.16.840.1.113883.6.3" codeSystemName="ICD-10"
   codeSystemVersion="1.0" displayName="Malaria" xsi:type="CV"/>
</observation>
```

Fig. 3. An Example observation from CDA

CEN 13606-3 models this CDA “Entry” class as an “Entry” in the 13606 reference model. This “Entry” involves one “Cluster” composed of “Element”s each of which are used for modeling one property (attribute/element) of the CDA “Entry” class. There are four properties to map to the “Element”s. Due to space limitations, only a part of 13606-1 counterpart of the CDA observation, namely, the “Element” for “code” of the observation is given in Figure 4. A detailed example transforming the complete examination CDA document to CEN 13606 is given in [35].

```
<parts xsi:type="ELEMENT">
  <rc_id>
    <extension>NA</extension>
    <assigningAuthorityName>NHIS, Turkey</assigningAuthorityName>
    <validTime/>
    <root><oid>NA</oid></root>
  </rc_id>
  <name>
    <originalText>Act code</originalText>
  </name>
  <meaning>
    <codingScheme><oid>987654321.1</oid></codingScheme>
    <codingSchemeName>CEN/TC251/EN13606-3:RECORD_COMPONENT_MEANING</codingSchemeName>
    <codingSchemeVersion>1.0</codingSchemeVersion>
    <codeValue>HL7-04</codeValue>
    <displayName>Act_code</displayName>
  </meaning>
  <synthesised>false</synthesised>
  <item_category>
    <codingScheme><oid>987654321.3</oid></codingScheme>
    <codingSchemeName>CEN/TC251/EN13606-3:ITEM_CATEGORY</codingSchemeName>
    <codingSchemeVersion>1.0</codingSchemeVersion>
    <codeValue>IC01</codeValue>
  </item_category>
  <value xsi:type="CV">
    <codingScheme><oid>2.16.840.1.113883.6.96</oid></codingScheme>
    <codingSchemeName>SNOMEDCT</codingSchemeName>
    <codingSchemeVersion/>
    <codeValue>8319008</codeValue>
    <displayName>Main diagnosis</displayName>
  </value>
</parts>
```

Fig. 4. CEN 13606 Conformant “ELEMENT” Corresponding to “code” of “observation” in CDA

Finally, we note that when different EHR standards are used, the interoperability of the data types may become a problem. What is promising in this respect is the ongoing standardization effort for the harmonization of data types from HL7 Version 3, ISO 11404, CEN 13606 and the past ISO work on healthcare data types and the joint efforts of ISO/TC 215, HL7 and Connecting for Health [36].

IV. Providing the Interoperability of the EHRs at the Semantic Level

Semantic interoperability in eHealth is the ability for information shared by systems to be understood at the level of formally defined domain concepts so that the information is computer processable by the receiving system [37]. The semantic encoding of the data elements is achieved by using coded terms from code systems such as LOINC [20] or SNOMED CT [21].

When the code systems to be used with each data element is fixed, semantic interoperability can be achieved. For example, in NHIS, Turkey, if a data element is defined in the National Health Data Dictionary as coded or classified, then within the definition of the data element, the related code/classification system is given in the “HCRS System Code” field. Additionally, all these code systems are available at the Health Coding Reference Server (HCRS) [38] as Web services to be directly used by the applications generating EHRs to check whether the code used in a data element exists in the related code system.

However, cross border semantic interoperability of the EHRs is a challenge because different countries may be using different code systems and may additionally have local code systems. Therefore, even when the document schema can be processed without a problem at the receiving end, the coded terms used may be undecipherable.

The need for common terminology services has been recognized by HL7 and ISO and an initiative has started for defining and standardizing Common Terminology Services (CTS) [39]. CTS focuses on the common functional characteristics that an external terminology system must be able to provide, rather than specifying what an external terminology system must look like [39]. In other words, CTS defines an Application Programming Interface (API) that can be used when accessing and exchanging terminological content. The Web Service Description Language (WSDL) bindings of the APIs are also provided as normative content.

CTS is categorized into three main models: Message, Vocabulary and Translation (Code Mapping). The Message API is specific to HL7. Its purpose is to allow variety of HL7 message processing applications to create, validate and translate coded data types. On the other hand, the Vocabulary API is generic and is utilized by the Message API to allow applications to access different terminologies. As the name implies, Code Mapping API is used for code translation between different coding systems. Additionally, Message and Vocabulary models have “browsing” and “run time” sections. While “run time” functions are used for querying and retrieving from the terminology servers, “browsing” functions are used for defining and creating content for them.

Having a standard interface to access the terminology services is important to achieve semantic interoperability: the CTS allows applications to use a single interface to access different terminology servers such as the local terminology servers or the Unified Medical Language System Knowledge Source Server (UMLSKS) [22] or GALEN [40].

TABLE I
MAPPING CTS FUNCTIONS TO UMLSKS WEB SERVICE OPERATIONS

	CTS-Function	Description	UMLSKS Metathesaurus Operation
Vocabulary Runtime	getSupportedCodeSystems	Return the identifier, name and release versions of all code systems that are supported by the service	describeSources
	lookupCodeSystemInfo	Return detailed information about a specific code system	describeSources
	isConceptIdValid	Determine whether concept code is currently valid in the specified code system	findCUIBySabCode (Find Concept Unique Identifier by Source Abbreviations and Code)
	lookupDesignation	Return the preferred designation for the concept code in the supplied language	findCUIBySabCode and getConceptProperties
	areCodesRelated	Determine whether the named relationship exists between the source and target codes	findCUIBySabCode and getConceptProperties
Vocabulary Browsing	lookupConceptCodesByDesignation	Return a list of concept codes that have designations that match the supplied match string in the supplied language, if any	findLUI (Find Lexical Unique Identifier) and getTermsForLUI
	lookupConceptCodesByProperty	Return a list of concept codes that have properties that meet the supplied criteria	findCUIByExact or findCUIByNormString or findCUIByNormWord or findCUIByWord or findCUIByRightTruncation or findCUIByLeftTruncation or findCUIByApproximateMatch or findCUIBySemType (by semantic type) or findCUIBySabCode
	lookupCompleteCodedConcept	Return a complete description of the supplied concept code	findCUIBySabCode and getConceptProperties
	lookupDesignations	Return all designations for the supplied concept code that match the supplied criteria	findCUIBySabCode and getConceptProperties
	lookupProperties	Return the properties of the given code system id / concept code that match the supplied criteria	findCUIBySabCode and getConceptProperties
	lookupCodeExpansion	Recursively list the concept codes that are related to the supplied concept, including the preferred designation for the codes	getConceptProperties
Code Mapping	getSupportedMaps	Return a list of mappings that are supported by this service	getMapping
	mapConceptCode	Return the mapping of the supplied concept code from the source code system to the target code system using the named mapping resource	findCUIBySabCode and getConceptProperties

A. Using Common Terminology Services (CTS) for Translation between the Code Systems

In order to access the existing terminology servers using CTS, CTS interfaces must be mapped to the interfaces of the existing terminology servers. Although the terminology servers all implement similar functionalities, they have different interfaces.

In this section, we describe how to implement the CTS functions with the operations of one of the prominent and publicly available terminology servers, namely, UMLSKS [22]. Such an implementation is essential to get content from UMLSKS using a CTS interface.

UMLS provides secure Web Services for querying the Knowledge Source Server to find clinical concepts, to retrieve concept details, or to map clinical terms. In Table I, we provide the correspondences between the CTS Vocabulary Runtime, Vocabulary Browsing and Code Mapping Functions and the UMLSKS Web Service Operations. In order to implement a CTS function with the UMLSKS operations, the following steps are necessary:

- As shown in Table I, a CTS function may correspond to a number of UMLSKS operations. While implementing a CTS function through the UMLSKS operations, the arguments of the CTS function must be properly mapped to the arguments of the UMLSKS operations.
- In CTS, all the code systems are identified by their Object Identifiers (OID), for example the OID for ICD10 is “2.16.840.1.113883.6.3”. However, UMLSKS uses the abbreviations of the code systems as identifiers, such as SNOMEDCT or ICD10, rather than the object identifiers. The solution we propose is to maintain a table, namely, the “OID-Abbreviations” table, giving the correspondences between the OIDs of code systems used in CTS and their abbreviations.

As an example, Figure 5 gives the procedure that implements the CTS *mapConceptCode* function with the UMLSKS operations *findCUIBySabCode* and *getConceptProperties*. *mapConceptCode* takes three arguments: “fromCodeSystemOID” is the OID of the code system and “code” is the code to be mapped. “toCodeSystemOID” is the OID of the target code system. Given the OID of a code system, the function *findNameFromOID* returns the corresponding abbreviation by using the “OID-Abbreviations” table. Then, the *findCUIBySabCode* (Find Concept Unique Identifier by Source Abbreviations and Code) UMLSKS operation is called to find the UMLS concept corresponding to the input code and the code system. UMLS has its own generic concepts and the codes from various code systems such as ICD10, SNOMEDCT are bound to these concepts. Once the generic UMLS concept

```
mapConceptCode(fromCodeSystemOID, code, toCodeSystemOID) {
   // First map the OIDs of code systems to abbreviations in UMLSKS
   fromCodeSystemName = findNameFromOID(fromCodeSystemOID);
   toCodeSystemName = findNameFromOID(toCodeSystemOID);

   // Find the UMLS concept id for the corresponding input code
   umlsConceptId = findCUIBySabCode(umlsRelease, code, fromCodeSystemName);

   // Retrieve the details for the found UMLS concept id
   umlsConcept = getConceptProperties(umlsRelease, umlsConceptId);

   // Get the terms of the UMLS concept from various code systems
   termsForTheUMLSConcept = umlsConcept.getTermGroup();

   // Iterate over each term
   for aTerm in termsForTheUMLSConcept {
      // For each term, compare the name of the code system with toCodeSystemName
      // These may seem complicated since they are low-lewel programming details
      if aTerm.getTV().getSS().getSAB() == toCodeSystemName
         result = aTerm.getTV().getSS().getCode();
   }

   return result;
}
```

Fig. 5. A Procedure Implementing *mapConceptCode* CTS function with UMLS operations

is accessed, it is possible to get all the "terms" that are bound to this concept. *findCUIBySabCode* returns the id of the UMLS concept. The details of this concept is retrieved by calling *getConceptProperties*. The procedure, as shown in Figure 5, iterates over all the terms bound to this UMLS concept to find the matching code in the target code system.

As an example, suppose that we wish to map "J45" which is the code for "asthma" in ICD10 to its SNOMEDCT equivalent. *mapConceptCode* is called with "2.16.840.1.113883.6.3", "J45" and "2.16.840.1.113883.6.96" as the arguments. "2.16.840.1.113883.6.3" is the OID of ICD10, "J45" is the code for "asthma" in ICD10 and "2.16.840.1.113883.6.96" is the OID of SNOMEDCT. Then, *findCUIBySabCode* is called with "2009AB" (UMLS release version), "J45" and "ICD10" as the arguments. It returns the UMLS concept id for "asthma", namely "C0004096". Then the procedure retrieves the details for this concept by calling *getConceptProperties* and searches the terms bound to this concept until it finds the one whose abbreviation is SNOMEDCT. The result is "195967001" which is the code of "asthma" in SNOMEDCT. All of the CTS-UMLS mappings provided in Table I can be implemented in this way.

Another problem to be addressed in cross-border semantic EHR interoperability is mapping the locally developed codes and code systems to their target counter parts. Since the local codes are developed in the local language of a country, when mapping them to the target code system, the language problem must also be addressed.

It is clear that the *mapConceptCode* procedure cannot handle such cases because the locally developed code systems do not exist in a global terminology server.

In order to address this problem, we designed an additional layer: we maintain a table, called "DesignationCodeMappingTable" (Figure 6) containing the designations of the local codes and the corresponding English terms. This gives us the ability to query a terminology server just with the designation of a code, that is, with its human-readable name in English, such as "Main diagnosis" instead of the code and the code system. Then, by implementing a new function, namely, *mapCodeThroughDesignation* which returns the code in the target code system when a designation and target code system is given, it becomes possible to get the equivalent code in the target code system. Note that CTS may consider making this useful function a part of its Code Mapping API.

code	orgDesignation	engDesignation	tarCodeSystem	variant
ANATANI	Ana Tanı	Main diagnosis	SNOMEDCT	{1, 2}
EKTANI	Ek Tanı	Secondary diagnosis	SNOMEDCT	{3, 4, 5, 6}
SEVKTANISI	Sevk Tanısı	Referral diagnosis	SNOMEDCT	{7}
CIKISTANISI	Çıkış Tanısı	Discharge diagnosis	SNOMEDCT	{8}
KOMPLIKAS-YONTANISI	Komplikasyon Tanısı	Complication diagnosis	SNOMEDCT	*NULL*

Fig. 6. "DesignationCodeMappingTable" for "Diagnosis Type" Code System

```
mapCodeThroughDesignation(orgDesignation, toCodeSystemOID) {
   // First map the OID of the code system to its abbreviation in UMLSKS
   toCodeSystemName = findNameFromOID(toCodeSystemOID);

   // If the input designation is not in English, then retrieve
   // its English translation from the DesignationCodeMappingTable
   engDesignation = translateDesignation(orgDesignation);

   // Find the UMLS terms from various code systems corresponding
   // to the English translation of the designation string
   termIdsForTheDesignation = findLUI(umlsRelease, "ENG", engDesignation);

   // Iterate over each term
   for aTermId in termIdsForTheDesignation {
      // For each term, compare the name of the code system with toCodeSystemName
      // These may seem complicated since they are low-lewel programming details
      if aTermId.getStringInfos().getSS().getSAB() == toCodeSystemName
         result = aTermId.getStringInfos().getSS().getCode();
   }

   return result;
}
```

Fig. 7. A Procedure Implementing *mapCodeThroughDesignation* function with UMLS operations

Figure 7 gives the procedure that implements the *mapCodeThroughDesignation* function with *findLUI* UMLSKS operation. *findLUI* (Find Lexical Unique Identifier) Metathesaurus operation finds all the coded terms for a designation string according to a given language and the UMLS version. *mapCodeThroughDesignation* takes two arguments: "orgDesignation" is the human-readable name of the source code to be mapped in the local language and "toCodeSystemOID" is the OID of the mapping target. *findNameFromOID* function uses the "OID-Abbreviations" table to return the abbreviation of a code system. *translateDesignation* function outputs the English translation of a given designation in the local language by using "DesignationCodeMappingTable". The result is stored in the "engDesignation" variable. Then, the *findLUI* UMLSKS operation is called to find the UMLS "terms" corresponding to the input "engDesignation" in the English language by specifying a UMLS version as well. We prefer querying the UMLSKS in English since a great majority of code systems are available in English in UMLS. Note that *findLUI* UMLSKS operation returns all the "terms" directly rather than finding the corresponding UMLS concept. Once the terms related with the source "engDesignation" are gathered, the procedure, as shown in Figure 7, iterates over all the terms to find the matching code in the target code system. After the code in an international code system is retrieved, it is possible to find designations of this code in other languages. For instance, UMLS provides German localization of ICD10 and Spanish localization of

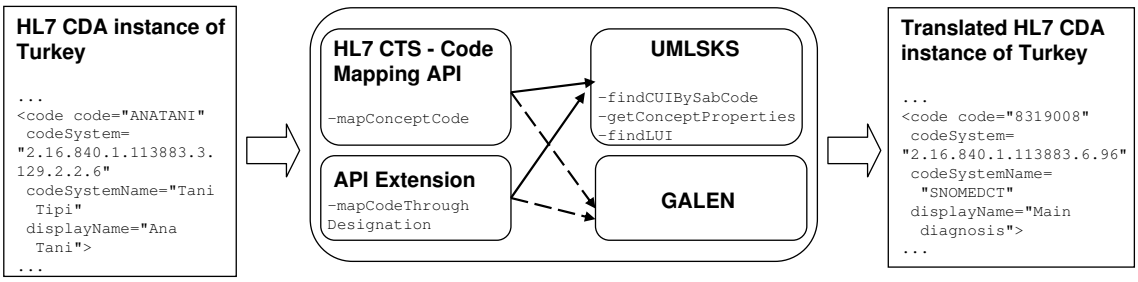

Fig. 8. Translating the Turkish Local Codes to their International Counterparts

SNOMEDCT.

Turkey is one of the countries that developed its own code systems in addition to using the international code systems. In Section IV-B, we explain how to implement our approach for automatically translating Turkish local codes to codes from international target code systems. Note that all the translated codes are checked by healthcare professionals.

B. An Example Application of the Semantic Translation Environment

As stated earlier, the majority of the data elements defined in the CDA based EHR documents of Turkey are coded locally, and they get their values from the coding systems in the Health Coding Reference Server (HCRS) [38]. The HCRS is comprised of 178 coding systems only four of which are international, such as ICD10 [28] and ICPC2 [41].

In order to provide cross-border semantic interoperability support to NHIS, Turkey, we use the code mapping approach presented in Section IV-A, as shown in Figure 8. First, all the coded values in a CDA document are retrieved with simple XPath [42] expressions. Then the *mapCodeThroughDesignation* function is called with the designations of these values and the corresponding international counterparts are fetched from the terminology server. Finally, the local codes are replaced with codes from target international code systems according to the user preferences.

The “DesignationCodeMappingTable” for the “Diagnosis type” code system in the HCRS is presented in Figure 6. The manual steps of the environment are entering the English translations of the Turkish designations into the “engDesignation” field and checking the results returned for consistency and correctness. The “tarCodeSystem” field stores the name of the target code system that the code will be mapped to. It is possible to specify distinct target code systems for each code in a local code system or enforce just one for the complete code system. If a value from the target code system occurs in the *mapCodeThroughDesignation* function response, then it will be chosen. Else, the decision is done based on the user preferences. If the user provides alternative target code systems, then they are checked. Otherwise, just the English translations of the designations are replaced in the CDA document. In the “Diagnosis type” code system, the SNOMEDCT is preferred as the target code system because of its extensive content and wide usage such as, the UK [43] and the USA [44].

id	codeSystem	code	designation
1	SNMI	G-1007	Main diagnosis
2	SNOMEDCT	8319008	Main diagnosis
3	LOINC	MTHU021495	Secondary diagnosis
4	MTH	U000281	Secondary diagnosis
5	SNMI	G-1008	Secondary diagnosis
6	SNOMEDCT	85097005	Secondary diagnosis
7	SNOMEDCT	406523004	Referral diagnosis
8	HL7V3.0	DISDX	discharge diagnosis

Fig. 9. International Counterparts for the “Diagnosis Type” Code System

In order to increase the system performance by reducing the number of Web Service calls to the UMLSKS, the retrieved results are stored locally to a table as shown in Figure 9 which corresponds to the “variant” field of the “DesignationCodeMappingTable”. When a local code is to be translated, this table is checked first to avoid an unnecessary Web service call to the UMLSKS.

V. Experimental Results

The HL7 CDA R2 based Electronic Healthcare Records are collected from the healthcare providers via National Health Information System (NHIS) Web Services in Turkey. There are 25 HL7 Web Services for different EHR formats such as “Examination”, “Pregnant Observation” or “Diabetes” that are composed of Minimum Health Data Sets, and they conform to the HL7 Web Services Profile [45]. There are some non-HL7 Web Services for the rarely used EHR formats as well. All of these Web Services have been heavily tested and evaluated during the development stage of the NHIS, Turkey.

The testing phase is composed of two main sub-phases: functional testing and performance testing. NHIS Web Services do not just get the EHR documents from healthcare providers and store somewhere; instead they first run validity constraints on the documents against schemas, business rules, Health Coding Reference Server (HCRS), etc. Functional testing covered this part, i.e. it evaluated whether the Web Services act as they supposed to do in every possible case. Functional testing started in February 2008 and ended in August 2008. During this period, thousands of EHR documents have been prepared and exchanged with the Web Services. Some of the functional testing items are: conformity of the Web Services themselves to the defined EHR XML schemas; control of mandatory, optional and conditional elements; control of coded

elements against the HCRS; HL7 data type checks and conformity of the EHR documents against the defined business rules such as the examination beginning time should be prior to ending time. Numerous errors have been detected in the Web Services as a result of this testing process and all of them are corrected before the Web Services went public.

After we are satisfied with the functional responsibilities of the Web Services, we started performance testing under heavy load. Since healthcare providers and general practitioners from all over the country are obliged to send their EHRs, high performance and 7/24 availability of the NHIS is essential to achieve. Performance testing started in September 2008 and ended in December 2008. During this period, millions of EHR documents have been exchanged with the NHIS Web Services. Some example performance testing frequencies are 20 EHR documents a second and million documents a night. These tests revealed that in the beginning Web Services had severe performance issues, such as memory, logging and response time problems. As a result, several enhancements most of which are software related have been done on the services.

Starting from January 15, 2009 the healthcare providers are sending real data to the NHIS Web Services. By January 2010, almost all public hospitals and a great majority of private and university hospitals are sending the EHRs constructed for the patients on a daily basis. Soon, the remaining healthcare providers will complete their integration phase. The overall EHR exchange rate is intense; as of January 2010, 150-200 thousand EHR documents are being sent to the servers of Ministry of Health on a daily basis. These numbers present the success of the NHIS and naturally the followed approach. The outcome of the UN/CEFACT Core Component Technical Specification (CCTS) approach is the comfortable understanding of rather complex EHR schemas by the developers and thus fast adaptation of the Hospital Information System (HIS) vendors. There are over 65 registered HIS vendors in Turkey and currently all of them are equipped with invaluable experience about international healthcare informatics standards such as HL7, CDA and ICD10.

VI. Conclusions

Cross-border interoperability of Electronic Healthcare Records are on the agenda of many countries. For example, in EU when a patient is taken ill while abroad, in an EU country other than his own, arrangements exist for the payment of costs. However, when it comes to making the most out of such arrangements, it is also necessary to access his previous clinical history through machine processable and interoperable EHRs.

In this article, we address the two important issues in cross-border EHR interoperability: interoperability of EHR structures and the interoperability of the code systems used.

For the structural interoperability, we propose to adapt the UN/CEFACT Core Component Technical Specification (CCTS) to the eHealth domain and build the EHRs starting from the basic data elements. Then the Minimum Health Data Sets are created from these data elements and EHRs from the Minimum Health Data Sets in return. This approach standardizes the basic components and how they are composed. Additionally, in order to address the interoperability among different EHR standards, we identify the correspondences between the basic document components of the two prominent EHR standards namely, HL7 v3 CDA R2 and CEN EN 13606-1 and then describe how to transform a source EHR schema in one standard into a target EHR schema in another standard by using Extensible Stylesheet Language Transformation (XSLT) rules.

It should be noted that there is an ongoing joint work by HL7 and CEN for modeling EHRcom with the HL7 Reference Information Model [46] where Domain Message Information Model (D-MIM) [47] and Refined Message Information Model (R-MIM) [48] of EHR_Extract reference model is developed. When this work is concluded, achieving interoperability of CEN EHRcom and HL7 CDA will be facilitated.

Semantic interoperability in eHealth is achieved through formally defined code systems. Since it does not seem plausible for all the EHR systems to use the same code system, we propose to use HL7 Common Terminology Services (CTS) to translate between different code systems. In this work, we use UMLSKS for as the terminology server because it is publicly available. However, the approach proposed can be used with any terminology server including the locally developed ones or other global ones such as GALEN [40].

The implementation is realized on the Java Platform, Standard Edition 6. In the transformation module, Saxon XSLT processor [49] is used as the built-in XSLT processor. Saxon offers open-source implementation with a very good performance. In the translation module, Axis [50] is used as the Web Service Framework.

References

[1] I. Iakovidis, "Towards personal health records: Current situation, obstacles and trends in implementation of Electronic Healthcare Records in Europe ," *International Journal of Medical Informatics*, vol. 52(1-3), pp. 105–115, 1998.

[2] "Clinical Document Architecture, Release 2," http://www.hl7.org-/v3ballot/html/infrastructure/cda/cda.htm.

[3] CEN EN 13606-1, "Health informatics – Electronic health record communication – Part 1: Reference model," European Committee for Standardization, Brussels, Belgium, Tech. Rep. EN 13606-1, 2007.

[4] "openEHR," http://www.openehr.org/.

[5] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, and G. Laleci, "A survey and analysis of Electronic Healthcare Record standards," *ACM Computing Surveys*, vol. 37, no. 4, pp. 277–315, 2005.

[6] "Human Readability and Rendering CDA Documents," http://www.hl7.org/v3ballot/html/infrastructure/cda/cda.htm#Human-_Readability_and_Rendering_CDA_Documents.

[7] C. Xanthopoulakis, "Interconnection of Philips Motiva and Catharina Hospital Eindhoven using the Clinical Document Architecture," in *9th International HL7 Interoperability Conference (IHIC)*, October 2008.

[8] M. M. et.al., "Sustainable Ubiquitous Home Health Care - Architectural Considerations and First Practical Experiences," in *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics*, 2007.

[9] eHealth Stakeholders' Group, "eHealth Interoperability - proposed activities," http://www.srdc.metu.edu.tr/stakeholders_group/public_docs-/ehealth-stakeholders-interoperability122005.pdf.

[10] "Smart Open Services for European Patients (epSOS)," http://-www.epsos.eu/about.html.

[11] K. Y. et.al., "The Use of HL7 CDA in the National Health Information System (NHIS) of Turkey," in *9th International HL7 Interoperability Conference (IHIC)*, October 2008.

[12] "IHE Exchange of Personal Health Record Content Profile," http://-www.ihe.net/Profiles/index.cfm#pcc.

[13] "HITSP Biosurveillance Interoperability Specification," http://www.hitsp.org/ConstructSet_Details.aspx?&PrefixAlpha=1-&PrefixNumeric=02.

[14] "National Health Information Network (NHIN)," http://www.hhs.gov-/healthit/healthnetwork/background/.

[15] "Canada Health Infoway," http://www.infoway-inforoute.ca/.

[16] "NHS Connecting for Health," http://www.connectingforhealth.nhs.uk/.

[17] "Dossier Medical Personnel (DMP)," http://www.d-m-p.org/.

[18] "UN/CEFACT Core Components Technical Specification," http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf (2008).

[19] "XSL Transformations (XSLT)," http://www.w3.org/TR/xslt.

[20] "Logical Observation Identifiers Names and Codes (LOINC)," http://-www.regenstrief.org/loinc/.

[21] "The Systematized Nomenclature of Medicine (SNOMED)," http://-www.ihtsdo.org/snomed-ct/.

[22] "Unified Medical Language System (UMLS) Knowledge Source Server (KSS)," http://umlsks.nlm.nih.gov/.

[23] "HL7 Reference Information Model," http://www.hl7.org/v3ballot/html-/infrastructure/rim/rim.htm.

[24] "Open Health Tooling, Instance Editor," http://hl7book.net-/index.php?title=Eclipse_Instance_Editor.

[25] R. J. Spronk and G. Grieve, "Common issues found in implementations of the HL7 Clinical Document Architecture (CDA)," in *9th International HL7 Interoperability Conference (IHIC)*, October 2008.

[26] "CEN/TC Published Standards," http://www.cen.eu/CENORM-/Sectors/TechnicalCommitteesWorkshops/CENTechnicalCommittees-/Standards.asp?param=6232&title=CEN%2FTC+251.

[27] "Unified Medical Language System (UMLS)," http://www.nlm.nih.gov-/research/umls/.

[28] "International Statistical Classification of Diseases and Related Health Problems, 10th Revision (ICD-10), Second Edition," World Health Organization, Geneva, Switzerland, Tech. Rep., 2005, http://www.who.int-/whosis/icd10/.

[29] "XSL Transformations (XSLT) 2.0," http://www.w3.org/TR/xslt20.

[30] "The National Health Data Dictionary (NHDD) of Turkey," http://www.sagliknet.saglik.gov.tr/portal_pages/notlogin/bilisimciler-/docs/usvs_sozluk_1.1.rar.

[31] "HL7 Data Types - Abstract Specification," http://www.hl7.org/v3ballot-/html/infrastructure/datatypes/datatypes.htm.

[32] Dipak Kalra and David Lloyd, "EHR Context Examples," Tech. Rep., http://www.chime.ucl.ac.uk/~rmhidsl/EHRcomDownloads/EHR-Context_Examplesv4.ppt.

[33] CEN EN 13606-3, "Health informatics – Electronic healthcare record communication – Part 3: Reference Archetypes and Term Lists," European Committee for Standardization, Brussels, Belgium, Tech. Rep. EN 13606-3, 2008.

[34] "CEN EN 13606 Reference Model XML Schema by LinkEHR Project," http://pangea.upv.es/linkehr/wp-content/uploads/2008/05-/en13606schema.zip.

[35] "Resources for NHIS, Turkey Transformation Environment," http://-www.srdc.com.tr/~mustafa/nhistt.

[36] DRAFT prEN ISO 21090, "Health Informatics – Harmonized data types for information interchange," Tech. Rep.

[37] CEN/ISSS eHealth Standardization Focus Group, "Current and future standardization issues in the eHealth domain: Achieving interoperability," CEN/ISSS, Tech. Rep., 2005, ftp://ftp.cenorm.be-/PUBLIC/Reports/eHealth/eHealthStandardizationReport-Part2-Final version 2005-03-01.pdf.

[38] "The Health Coding Reference Server 2," http://sbu.saglik.gov.tr-/SKRS2%5FListesi/.

[39] "HL7 Common Terminology Services (CTS), Release 1," http://-www.hl7.org/v3ballot/html/infrastructure/cts/cts.htm.

[40] C. A. Goble, P. Crowther, and D. Solomon, "A medical terminology server," in *Database and Expert Systems Applications*, 1994, pp. 661–670.

[41] "International Classification of Primary Care, Second edition (ICPC-2)," http://www.who.int/classifications/icd/adaptations/icpc2/en/index.html.

[42] "The XML Path Language (XPath)," http://www.w3.org/TR/xpath.

[43] "SNOMED-CT Usage in NHS CFH HL7 Templates," http://www.connectingforhealth.nhs.uk/systemsandservices/data-/standards/news/tusage.ppt.

[44] S. R. Renly, S. E. Knoop, J. H. Kaufman, and R. Ram, "Creating CDA R2 Laboratory Reports to Meet Public Health Surveillance Requirements," in *9th International HL7 Interoperability Conference (IHIC)*, October 2008.

[45] "Hl7 version 3 standard: Transport specification - web services profile, release 2," http://www.hl7.org/v3ballot/html/infrastructure/transport-/transport-wsprofiles.htm.

[46] "HL7 Implementation Guide for 13606," http://www.hl7.org.uk-/repository/uploads/402/1/HL7%20Implementation%20Guide%20for-%2013606%20scopev0r6.doc.

[47] "HL7 Domain Message Information Model (D-MIM)," http://-www.hl7.org/v3ballot/html/help/v3guide/v3guide.htm#v3dmim.

[48] "HL7 Refined Message Information Model (R-MIM)," http://-www.hl7.org/v3ballot/html/help/v3guide/v3guide.htm#v3grmim.

[49] "Saxon, The XSLT and XQuery Processor," http://saxon.sourceforge.net-/.

[50] "Apache Axis Web Service Library for Java," http://ws.apache.org/axis/.