Unlocking the Value of Clinical Information: What you Need to do Now to Enjoy the Benefits in the Future

Hooman Katirai[1,2], Ulrich Sax[2],

[1] Clinical Decision Making Group
MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar Street, Area 250, Cambridge, MA 02139, USA
hooman@MIT.edu

[2] Children's Hospital Informatics Program (CHIP)
Harvard-MIT Division of Health Sciences & Technology
320 Longwood Ave, Boston, MA 02125, USA
ulrich.sax@childrens.harvard.edu

Abstract. Electronic communication and connectivity are among the core functionalities of an Electronic Health Record (EHR) infrastructure with comprehensive life-long records. But clinical data is still mostly locked up in disjoint data silos, and often stored in non-standard formats. This paper elucidates the opportunities and drawbacks of using the HL7 Clinical Document Architecture (CDA) as a standard for storing clinical information for computer decision support and knowledge management. As EHR document standards are continually being improved, we argue that the ease in upgrading to new standards should be a significant factor in the design of an EHR infrastructure. To achieve this upgradeability, we need to decouple a document's data from the standards used to represent the data. We argue that this separation between data and form can be achieved using a technique called *compilable templates,* which was recently developed by the authors in a Personal Health Record (PHR) project. Web services offer a compelling means of implementing such templates owing to their language and platform neutrality. Finally we show that some software that one would typically expect to find in a knowledge management context can be automatically generated from a compilable template, saving time and money while reducing the possibility of error.

1 Introduction

Communication and interoperability are the main goals in creating a useful electronic infrastructure in the healthcare system[1]. Patient data should be available both to the patient and all subsequent institutions dealing with the patient. This is not the case today. In the current situation, a patient's data will typically be scattered across numerous databases housed in different points of care. These data silos will each utilize a proprietary data format – often incompatible with the data standards of other organizations. As a result clinical data can rarely be exchanged between different organizations[2].

Personal Health Records (PHR) offer the possibility of medical records that can be easily accessed and annotated both within and without organizations[3, 4]. Post-genomic clinical studies and the application of data mining methods imply standardized patient data and standardized terminology being accessible to knowledge discovery tools[5]. Early attempts on standardizing medical vocabulary show the complexity of this task[6-8]. The lion's share of clinical phenotype data is not encoded for automated processing.

The main goals of enabling clinical data for decision support systems (DSS) and knowledge management (KM) are (a) minimizing the number of fields that express similar concepts in the federated databases and where it makes sense (b) codifying the data to obtain the best tradeoff between specificity and practicality.

An examination of the data models of a number of standards shows that the HL7 Reference Information Model (RIM) and the HL7 CDA architecture are well positioned to fulfill the goal of communication and interoperability of health information systems in general and EHRs in particular[9, 10].

The compilable template technologies introduced in this document were developed while integrating data from hospitals into the Personal Internetworked Notary and Guardian (PING) – a multi-institutional Personal Health Record designed to integrate health information from multiple sources[4, 11].

2 The HL7 CDA

The HL7 Clinical Document Architecture (CDA) is a document markup standard, which specifies the structure and the semantics of clinical documents in extensible Markup Language (XML). Persistence, stewardship, potential for authentication, wholeness and human readability are the main characteristics of the CDA[10, 12].

CDA defines three levels; in level one only the document header is fixed with a certain set of document and object identifiers. Level two additionally defines a structured document body, which has to be fully RIM-derived in level three[10]. The document type in the header as well as the universal observation identifiers have to be defined according to Logical Observation Identifier Names and Codes (LOINC)[13] document codes. European projects like PICNIC[14] and SCIPHOX[15] have successfully made use of CDA level one documents.

3 Compilable Templates[1]

To date, only a few CDA document types have been defined[9, 17] . Because changes to the HL7 CDA standard are likely, the effort required to upgrade to new standards should be a consideration in the design of a PHR infrastructure. One way to achieve upgradeability is to decouple the code that generates CDA documents from the code

---

[1] Our use of the word 'template' should not be conflated with the HL7 usage of the term 'template' in the sense of Archetypes[16].

that queries a database via a functional abstraction layer. The code that generates CDA documents can be treated as a black box function that writes CDA documents given a tuple whose elements represent the values of individual fields. We term this function the *forward function*; it produces a CDA document given a tuple of values. Fig. 1 shows a simplified example of a compilable template.

(a) A SNIPPET FROM A COMPILABLE TEMPLATE:

```
<!--#INCLUDE: "header.inc" -->
<person_name>
<nm>
  <v3dt:GIV V="___firstName___"/>
  <v3dt:MID V="___midInitial___"/>
  <v3dt:FAM V="___lastName___ "/>
</nm>
</person_name>
<!-- #INCLUDE: "footer.inc" -->
```

(b) PRODUCES THE FOLLOWING FORWARD FUNCTION:

```
String Toy_example(String first_name, String middle_initial,
       String last_name)
```

(c) CALLING THIS FUNCTION WITH A SET OF VALUES GIVES:

```
<levelone xmlns="urn:hl7-org:v3/cda"
 xmlns:v3dt="urn:hl7-org:v3/v3dt"
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation=
 "urn:hl7-org:v3/cda levelone_1.0.attachments.xsd">
<clinical document_header>
<person_name>
 <nm>
    <v3dt:GIV V="George"/>
    <v3dt:MID V="F"/>
    <v3dt:FAM V="Carson"/>
 </nm>
</person name>
</clinical document_header>
</levelone>
```

Fig. 1. This snippet from a compilable template shows (a) a simplified compilable template, (b) the automatically generated forward function and (c) the CDA document produced after calling the forward function with a set of values. Full examples of templates are available online[20].

Data mining algorithms generally do not accept XML documents as input; rather, they require their training sets to be in the form a table (where each row in the table represents a tuple). The act of transforming a CDA document into a tuple implies the existence of what we call an "*inverse function*". An inverse function is the opposite operation as the forward function; that is, given a CDA document, it returns a tuple of values. Figure 2 shows a simple example of an inverse function.

Inverse functions have to deal with the fundamental difference between XML documents and relational databases. An XML document is a hierarchical data structure, whereas a database table represents a flat data structure. When one converts an XML file into a database table, all implicit hierarchic information in the CDA document is lost. This emphasizes the need for a hybrid approach: "convert the content into database tables for data mining, but store the original CDA document as well".

Both the forward function and the inverse function can be automatically generated from a file we term a "*compilable template*". The file is a *template* insofar as it is similar to a CDA document but it contains variables in place of data values. The file is also termed *compilable* because it contains commands that be processed by a compiler[21] to generate outputs such as the forward and inverse functions.

```
(a) SNIPPET FROM A CDA Document:

<levelone xmlns="urn:hl7-org:v3/cda"
    xmlns:v3dt="urn:hl7-org:v3/v3dt"
    xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
    xsi:schemaLocation=
    "urn:hl7-org:v3/cda levelone_1.0.attachments.xsd">
  <clinical document_header>
    <person_name>
      <nm>
        <v3dt:GIV V="George"/>
        <v3dt:MID V="F"/>
        <v3dt:FAM V="Carson"/>
      </nm>
    </person name>
  </clinical_document_header>
</levelone>

(b) USING THE FOLLOWING  INVERSE FUNCTION:

Toy_example( String Toy_Example_CDA_Document)

(c) CALLING THIS FUNCTION WITH A CDA DOCUMENT GIVES
    (assumed a table person does not exist yet):

CREATE TABLE Persons (firstName,  midInitial, lastName);
INSERT INTO Persons values ('George', 'F', 'Carson');
```

Fig. 2. This snippet shows (a) a CDA document, (b) the automatically generated inverse function and (c) the resulting SQL-query to write the content in a database table.

Compilable Template	Resultant Function

`<!--#INSERT:`
`for (int i=0; i<first_name.length;i++){ -->`
`<person_name>`
`<nm>`
`<v3dt:GIV V="___firstName [i]___"/>`
`<!-- #INSERT: if (mid_Initial[i]!=null) -->`
`<v3dt:MID V="___midInitial[i]___"/>`
`<v3dt:FAM V="___lastName[i]___ "/>`
`</nm>`
`</person name>`
`<!--#INSERT: } // closing brace for staatement -->`
	`String Toy_example`
	`(`
	`String firstName[],`
	`String midInitial[],`
	`String lastName[]`
	`)`

Fig. 3. The 'for' statement above shows how variable length arguments can be incorporated into compilable templates (resulting in arrayed inputs in the resultant function) while the 'if' statement shows how optional arguments can be encoded (i.e the line following the "if-statement" will not be output for patients that have no middle initial).

Web services are functions that execute on a remote server. Given an XML file called a Web Service Description Language (WSDL)[18] file, one can automatically generate the code required to access a web service in just about any programming language of significance. Owing to the language and platform neutrality of web services, we believe that web services represent an ideal means of implementing the forward function – particularly in the context of a PHR where one can expect the cooperating institutions to utilize a variety of platforms and programming languages. Web services are elegant. In our demonstration system the invocation of the function to produce a CDA document takes only two lines of source code in the C# programming language and three lines in Java.

The use of compilable templates can automate many common tasks. To give the reader an idea of the scope of automation possible, consider the automation already achieved in the PING project. In addition to automating the production of the code for the forward and inverse functions, our template compiler also (1) creates web services for the forward and inverse functions (2) installs the web services on a web server (3) generates WSDL files for each web service (4) uses the WSDL file to generate the functions needed to access these web services corresponding to the forward functions in two languages (C#.Net and Java), and (5) installs on a web server appropriate documentation to allow authorized programmers to access the web services.

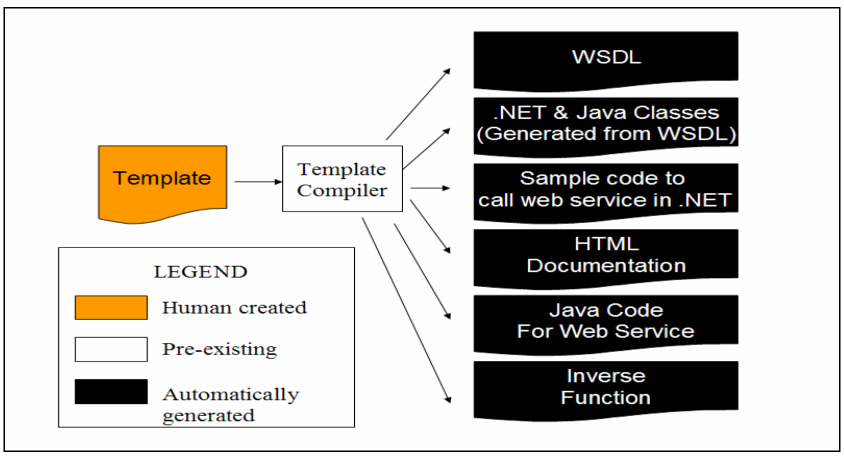

Fig. 4. A compilable template represents a single point of maintenance from which one can generate software and elements whose production would normally require human effort.

It is in this vein that we refer to compilable templates as a *single point of maintenance,* because all of the above functionality is automatically generated from a single compilable template file. When compared to the alternative approach of writing the CDA document manually (in which each step would represent a point of maintenance) the time and cost savings become compelling.

4 Opportunities and Challenges

The opportunity of our approach is it reduces the cost of moving existing documents to new standards, while immediately reducing the work required to produce software commonly needed for KM applications.

Electronically signed documents present special challenges since the alteration of a single bit is sufficient to void the signature. As a result, signed documents cannot be upgraded to a new standard per se without abandoning the validity of their signature.

Therefore we recommend the following hybrid approach. A copy of a signed document should be stored for archival purposes, while an unsigned version of the document can be upgraded to the newer standard, with a URL pointing to the old document.

An even bigger obstacle yet is the lack of ontological compatibility on the clinical domain. Using terminologies like LOINC or SNOMED CT etc. and the appropriate mapping to the Unified Medical Language System (UMLS)[19] could be a first step to approach the problem, but first attempts show the complexity of this task[6-8].

5 Conclusion

CDA documents offer many opportunities for computer decision support and knowledge management. In order to keep the infrastructure maintainable and upgradeable, additional services like web services and compilable templates are necessary.

Compilable template enable a single point of maintenance as the forward function, inverse function and associated documentation can be automatically generated from the template. To achieve ontological compatibility in an extremely heterogeneous healthcare environment still more work is necessary.

6 Acknowledgements

The authors are grateful to (1) Peter Szolovits at MIT for reviewing this manuscript and for his sage advice and criticism and to (2) Isaac Kohane at Children's Hospital, for helpful discussions and insights and for suggesting that web services be used in this project.

This work was supported by Deutsche Forschungsgemeinschaft (DFG, SA1009/1-1) and by the National Institutes of Health through contract N01-LM-3-3515 from the National Library of Medicine.

References

1. (IOM) IoM. Key Capabilities of an Electronic Health Record System - Letter Report. July 31, 2003. Available at: http://books.nap.edu/html/ehr/NI000427.pdf. August 16, 2004.

2. Waegemann CP. Status Report 2002: Electronic Health Records. *MRI*. Available at: http://www.medrecinst.com/resources/ehr2002/index.shtml. August 16, 2004.

3. Simons WW, Mandl KD, Kohane IS. The PING Personally Controlled Electronic Medical Record System: Technical Architecture. *J Am Med Inform Assoc.* 10 18 2004.

4. Riva A, Mandl KD, Oh DH, et al. The personal internetworked notary and guardian. *Int J Med Inf.* Jun 2001;62(1):27-40.

5. Kohane I, Glaser J. Informatics for Integrating Biology and the Bedside (I2B2). Available at: http://www.i2b2.org/index2.html. October 18, 2004.
6. Holloway E. Meeting Review: From Genotype to Phenotype: Linking Bioinformatics and Medical Informatics Ontologies. *Comp Funct Genom.* 2002;2002(3):447-450.
7. Verschelde J-L, Dos Santos MC, Deray T, Smith B, Ceusters W. Ontology-assisted database integration to support natural language processing and biomedical data-mining. *Journal of Integrative Bioinformatics.* 15.01.2004 2004(0001, 2004).
8. HL7 Clinical Genomics SIG. HL7 Clinical Genomics SIG San Diego Meeting Minutes Jan 21-22, 2004. *HL7*. Available at: http://www.hl7.org/library/committees/clingenomics/minutes/HL7%20CG%20SIG%20Meeting%20Minutes%202004%2001%2021%20v2.doc. October 18, 2004.
9. Lovis C, Lamb A, Baud R, Rassinoux AM, Fabry P, Geissbuhler A. Clinical Documents: Attribute-values Entity Representation,Context, Page Layout And Communication. *Proc AMIA Symp.* 2003:396-400.
10. Dolin RH, Altschuler L, Boyer S, Beebe C, Behlen FM, Biron PV. HL7 Clinical Document Architecture (Release 2.0). *Health Level Seven, Inc.* Available at: http://www.hl7.org/library/committees/structure/CDA.REleaseTwo.CommitteeBallot02.Dec.2003.zip. October 18, 2004.
11. PING - Personal Internetworked Notary and Guardian. *Children's Hospital Informatics Program (CHIP)*. Available at: http://www.ping.chip.org/IntroductionToPing.html. September 1, 2004.
12. Dolin RH, Alschuler L, Beebe C, et al. The HL7 Clinical Document Architecture. *J Am Med Inform Assoc.* Nov-Dec 2001;8(6):552-569.
13. McDonald CJ, Huff SM, Suico JG, et al. LOINC, a universal standard for identifying laboratory observations: a 5-year update. *Clin Chem.* Apr 2003;49(4):624-633.
14. Bray B. The PICNIC approach to regional care networks. *Stud Health Technol Inform.* 2003;96:80-87.
15. Heitmann KU, Schweiger R, Dudeck J. Discharge and referral data exchange using global standards--the SCIPHOX project in Germany. *Int J Med Inf.* Jul 2003;70(2-3):195-203.
16. Kernberg M, Elkin PL. HL7 Template and Archetype Architecture. *HL7*. Available at: http://www.hl7.org/Library/Committees/template/HL7_San_Diego_v2.3.zip. August 27, 2004.
17. Health_Level_Seven. HL7 Additional Information Specification Implementation Guide (Release 2.0 Ballot, August 2003). *Health Level Seven, Inc.* Available at: http://www.hl7.org/Special/committees/claims/HL7ClmAttIG.PDF. 09.02.2004.
18. W3C WWWC. Web Services Description Language (WSDL) 1.1. *W3C*. Available at: http://www.w3.org/TR/wsdl. September 3, 2004.
19. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* Jan 1 2004;32 Database issue:D267-270.
20. More complete examples of compilable templates and our template compiler. Available at: http://members.rogers.com/hoomank/templates.
21. Apache Foundation. Apache Web Services Project - an open source Java web services framework. Available at: http://ws.apache.org/axis. November 22, 2004.