Design and Implementation of an Extensible Annotation Scheme for Health Informatics Data

Mallikarjun Shankar
Oak Ridge National Laboratory
1 Bethel Valley Rd
Oak Ridge, TN 37831
shankarm@ornl.gov

Christopher Tomkins-Tinch
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, NY 14623
tomkinsc@gmail.com

ABSTRACT

We address the data-interoperability challenge in health informatics brought about by explosive data growth and the proliferation of diverse data silos. Bringing together the functional areas of medical encounters – patient and provider interaction, medical diagnosis, patient history, clinical follow up, and the claims lifecycle, requires an extensible and flexible data management infrastructure. The electronic health record (EHR) is a key artifact that threads the overall workflow from inception to resolution (treatment and claims disposition). The EHR must link to a variety of information sources which can vary and evolve over time, creating natural hurdles of extensible data representation, storage, retrieval, and information mutation. We have designed and developed an annotation system that addresses the central challenge of such data evolution – thus enabling interoperability of data modules in the health informatics life-cycle. Based on meta-analysis of prior records, the system provides context-relevant fields for new records creating a basis for automatic annotation. While point solutions that build custom tool-sets, databases, and sharing repositories can address fixed or clearly-defined data needs, we present an annotation system that explicitly allows extensibility over time.

Categories and Subject Descriptors

H.2 [Information Systems]: Data Structures, Data Representations, Naming Schemes

General Terms

Data Management, Data Experimentation

Keywords

Annotation, Tagging, Data Representation, Schema, NoSql, JSON, Lucene

1. INTRODUCTION

Managing the proliferation of diverse data - from medical encounters between providers and patients to the clinical care and claim history - is a critical data challenge for health informatics. The electronic health record (EHR) aims to provide a longitudinal view into a patient's interaction with care providers within an institution to enable better care and streamline workflows. Fields in an EHR are populated with data from related electronic medical records, and underlying data systems. In this paper, we focus our attention on enabling the underlying data systems to store information in a flexible manner to allow EHR and other systems to insert and extract data without rigid constraints.

The absence of rigid constraints requires a systematic extensible methodology. We present an approach that uses annotations to connect and represent information. The annotations represented as key-value pairs capture contents of the data-record, and their explicit simplified format allows extensibility.

1.1 From Disparate Health Informatics Data to Disparate EHR Standards

Data-silos within a health information system cause users – practitioners and administrators - to visit multiple systems for a patient's aggregated information. A natural evolution to an integrated conceptual representation - shown in Figure 1- suggests how the EHR concept integrates and pulls together disparate information in a unified manner.

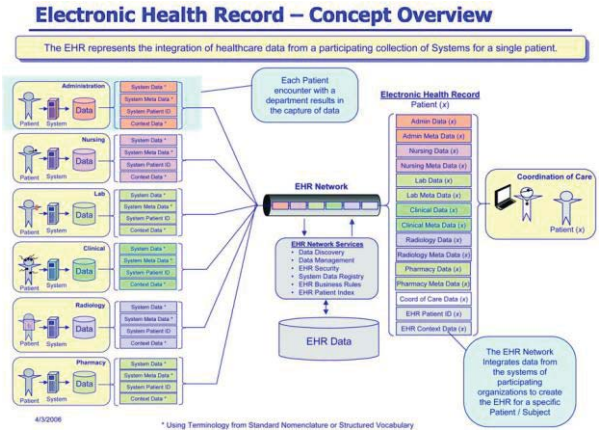

Figure 1: EHR Integrates Health Information [1]

Key challenges remain, though, in connecting the constituent datasets into one virtual view of the patient's information. Specifically, each database will need to provide its data via a database-specific adapter to the EHR interface. Such an approach may not allow the EHR system to recall new information easily from the constituent databases as they evolve over time.

Efforts at standardizing EHR data and communication are ongoing across various communities [2,3,4,5]. This social phenomenon suggests that we will see a proliferation of standards – a somewhat counterproductive development to meeting the original goal of uniformity and interoperability.

1.2 Modern Data Management for Health Informatics

Traditional data systems created structures that (i) preserved referential integrity (correctness on a data field mutation) and (ii) allowed erstwhile file systems hardware to support table-oriented files. The importance of the above two criteria for data-store design have diminished in the context of emerging modern approaches to handling data stores. In part, techniques in the areas of search, machine learning, and distributed commodity computing (i.e., cloud and cluster computing) have shown that structuring data differently offers dividends in supporting flexible and scalable storage. In addition, properties such as referential

integrity, serializability can be maintained with the new schemes even if they encounter some performance cost as a result. We identify the following two critical design requirements for health informatics data stores that can be addressed by modern informatics advances.

1) Data-store Extensibility: Since it is critical that health information stores absorb new modalities of data and new sources of analysis (especially in the era of the personal genome and lifetime-long data-stores), a data storage approach must first and foremost address the data-diversity and data-interoperability requirement by allowing extensibility. We contend that the ultimate benefits from flexibility outweigh traditional database design criteria for health informatics systems.

2) Data-store Flexibility: A common requirement in bioinformatics, biomedical, and health informatics systems is the annotation of records with new information. This suggests that the system design must easily allow re-annotation of existing information. In the context of EHRs, new medical information must be incorporated seamlessly without having to redesign the underlying infrastructure. Such a requirement also points to linked-data approaches [7] (though less semantics focused and more structural) to adding and annotating new information.

We address these two requirements by creating an annotation scheme which we implement using commodity open source components. In Section 2, we discuss the design constituents of this scheme. In Section 3, we outline the technology and software we have used to realize this annotation and storage scheme. We also outline how our ReST-ful services [10] expose underlying data to tools that can construct EHR content on the fly.

2. EXTENSIBLE DATA

We discuss our system design in terms of our chosen approaches to data representation and storage, and extensibility and (re)annotation.

2.1 Data Representation and Storage

Unlike conventional data stores that rely on rigidly-defined relational databases for information persistence and retrieval, we employ an approach that makes use of the increasingly popular "NoSQL" method of storing information with only rudimentary structures. The explicitly rudimentary (or atomic) structure allows extensibility – when new information needs to be added, it is stored in the same well-defined rudimentary framework. The most general form of the storage for such information is a document. In many of these systems data fields are not modified as much as they are added to (thus reducing the requirements on normalization [7]) and the document provides a good framework to store the information. (An implementation detail to foreshadow here is that as the document size grows, after a certain pre-defined size, the additional information is stored in a new document which is linked with explicit named pointers - backwards and forwards - between the documents.) Individual clinical reports, diagnostic data, and administrative notes may each be represented in this way.

2.2 Extensibility and Annotation

The document represents the underlying data element for information in the existing stores shown in Figure 1. We also refer to such an entity as a "data-packet". It can hold the EHR-view constructed statically or dynamically at a given point in time. The structure of the document itself is a series of key-value pairs. (This can be thought of as two-dimensional tagging.) The extensibility of the storage model derives from the unrestricted nature of the keys and values. A given document may contain any number of values, identified by associated keys. Values may store content or may function as references to other documents. In addition, new entries may be added to the document by simply appending key-value pairs. (The relational equivalent would either need placeholder columns for new entries and parameters, or a labored pointer-based traversal scheme to connect new and external data. This may lead to sparse tables, as contributors may not populate every column or correctly index each relation.) A schema-less approach offers us greater storage flexibly: analogous to a binder holding the artifacts of a physical health record relating to an individual, it is reasonable to conceive of electronic health records as collections of linked electronic documents.

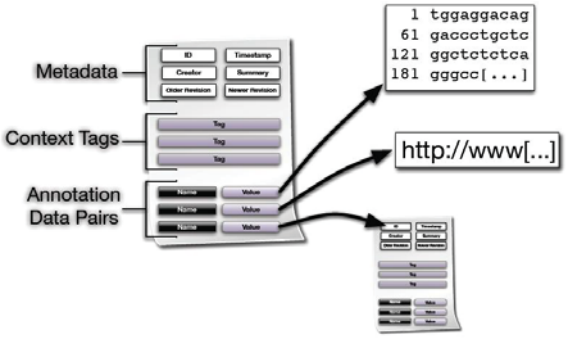

Figure 2: Key-Value Pairs Scheme

Figure 2 illustrates a document holding meta-data, context tags (which provide syntactic sugar), and key-value pairs as annotations. These key-value pairs can retain all the information that a traditional table structure may store. In addition, the un-typed nature of the value fields allow "links" to other documents (including, e.g., images from examinations), related documents, etc.


```
{
    "_id" : "a8b1413c8d3eec6bb089740bc0f085fb",
    "summary" : "Annual Physical Examination",
    "timestamp": "2010-07-29T17:20:14.818547",
    "context": {
  {"tag":"regular care",  "creator":"Dan Smith, MD", "timestamp":"2010-07-29T17:20:19.000000"},
  {"tag":"physical exam", "creator":"Dan Smith, MD", "timestamp":"2010-07-29T17:20:19.000000"},
  {"tag":"new patient", "creator":"Office manager for Dr. Dan Smith", "timestamp":"2010-07-29T17:20:14.818547"}
    },
 "annotation": {
   "manual_input":[
     {
        "creator":"Office manager for Dr. Dan Smith",
        "timestamp":"2010-07-29T17:20:14.818547",
        "key":"address_street",
        "value":"809 Elm Street"
     },
     {
        "creator":"Office manager for Dr. Dan Smith",
        "timestamp":"2010-07-29T17:20:14.818547",
        "key":"address_zip",
        "value":"37920"
     },
     {
        "creator":"Dr. Dan Smith",
        "timestamp":"2010-07-29T17:20:19.000000",
        "key":"Symptoms reported",
        "value":"None"
     },
     {
        "creator":"Dr. Dan Smith",
        "timestamp":"2010-07-29T17:20:19.000000",
        "key":"Bood pressure",
        "value":"125/80"
     },
     {
        "creator":"LabCare Diagnostics",
        "timestamp":"2010-08-01T10:10:14.000000",
        "key":"Albumins",
        "value":"4.2, g/dl"
     }
   ],
   "automatic_input":[
     {
        "creator":"DICOM_Feed",
        "status":"processed",
            "key":"ImageLink",
            "value":"http://hstore.us/imglink/0923r79o98X23/rad0923234098.ng",
            "associated_file":"http://hstore.us/imglink/0923r79o98X23/rad0923234098.dstore",
        "timestamp":"2010-07-30T12:05:13.513937"
     }
     {
        "creator":"Health Insurer"
        "status":"processed"
        "timestamp":"2010-07-30T12:05:13.513937"
     }
   ],
 }
}
```


Figure 3: Simplified EHR Contents in Name-Value Format

Figure 3's basic set of information illustrates the document representation, the key-value pair storage of information, and links (in the lower part of the record) to an external image and accompanying information. The format shown in this figure is the JSON format [9]– a human as well as machine readable format (not unlike XML) but with closer ties to current programming models (such as C#, Java, and Javascript).

2.3 Data Linking

Rather than associate EHR data using database operations such as joins, in our system data is associated through link references among documents. A master document within the EHR may define an individual, and link to documents which describe, for example, separate instances of care or all entries related to the chronic disease of an individual. These sub documents can, in turn, link to diagnostic data including laboratory measurements. In this sense, an EHR is not a discrete and rigidly prescribed package of information, but rather the network of information that relates to an individual at the root level. A complete copy of one's record may be virtually constructed by traversing the links among data.

Any mandatory requirements of a given key or value in a document is not an engineering constraint of the data storage backend, but falls within administrative consideration of users of the system. While data preserved within a schema-less document store may be isomorphic with data stored in a relational representation, the former finds advantage in the ease and flexibility of extending a record. Key-value pairs may be added or removed from any document. Historical evolution of a document is tracked by storing a list of doubly-linked copies of the document, each with specific revisions and a corresponding timestamp.

Taken together, the simplified representation, the ability to annotate flexibly, and the linking capability gives us the extensible framework that can support growing volumes and kinds of health informatics data. We have implemented these design constructs by bringing together and modifying open source components which we discuss next.

3. SYSTEM IMPLEMENTATION

We have implemented the system with a web-based user-interface front-end supported by a well-accepted model-view-controller framework underneath. We also support automated input-output mechanisms through a ReST API.

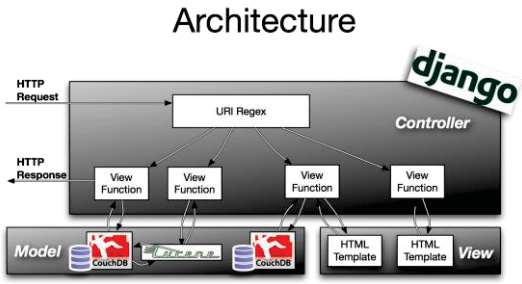

Figure 4: Flexible Annotation Store Architecture

3.1 Storage Infrastructure

Figure 4 shows the main components of our implementation. The data store is implemented using the document store NoSQL database CouchDB[11]. The interface to the document database is provided by the web application framework, Django[12]. Django provides the main GUI for data input, and exposes web-accessible addresses for searching and performing operations on the document database. Stored document data is made available in the lightweight human-readable JSON.

The storage layer itself could be replaced by other document databases and may be set up to be centralized or distributed. The critical requirement is the link-ability and the extensible addition of annotations. In the distributed case, document references must contain a primary data store.

3.2 EHR Information Retrieval

As clinical notation and interpretation are largely human activities, it is vital that an EHR system be equipped to store and retrieve free-text entries. Prose and shorthand notes may be inserted within any field of an EHR document. Again, these are stored as annotations to the document record. Indexing and query of these fields is performed by an adaptation of Apache Lucene[13]. The Lucene retrieval engine permits a query syntax with flexibility to account for any inconsistencies or incompleteness of the textual content.

Here we note another advantage of the approach we are taking to store health informatics information. We can perform powerful distributed and scalable searches across all the fields of the data (structured and unstructured). We also support an ability to auto-suggest key-value pairs, based on the occurrence of similar context tag sets within the database. Although traditional databases may be tuned to operate on SQL queries efficiently, being able to distribute the data and use modern data analysis frameworks (such as map-reduce[14]) is an advantage our system possesses. The scalable search and analysis capability offers a powerful counterweight to the apparent loss of finely tuned structure in our explicitly simplified representation system.

3.3 ReST-ful Services and Dynamic EHR Construction

The implementation exposes ReST API's to create new data-packets, retrieve them by identifiers, history, and query (using Lucene syntax) – these are listed in Figure 5.

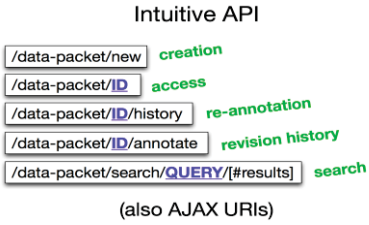

Figure 5: Data-Packet APIs

EHRs derived directly and indirectly from data-packets will be exchanged in the medical enterprise's workflows. The data-packets themselves will be uniquely identified, but alterations and annotations will be reinserted in the storage infrastructure. We anticipate that with new data added to EHRs, the viewer presentation of the EHR will be dynamically created from its constituents. While such an approach is similar to XML packet

formats, our implementation systematizes the storage components for those source data-packets.

4. CONCLUSIONS

We have presented an approach that enables extensible and flexible storage of health informatics data. Focusing on the needs of EHR interoperability, we show how modern informatics approaches for data representation can address changes due to the inevitable growth (both in diversity and volume) of data-stores. We discussed the main components of our implementation architecture which we build with open source components. Our implementation demonstrates that members of this community can assemble a data storage and retrieval structure for EHRs that allow long-term data storage, (re)annotation, free-text search, and dynamic record assembly.

5. ACKNOWLEDGMENTS

This manuscript is authored by employees of UTBattelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. We also gratefully acknowledge support from the DTRA/CBDx program.

6. REFERENCES

[1] Electronic Health Records Overview, April 2006, National Institutes of Health, NCRR, http://www.ncrr.nih.gov/publications/informatics/EHR.pdf

[2] Health Level Seven International, http://www.hl7.org/

[3] Integrating the healthcare enterprise initiative, Templates for Communication, http://www.ihe.net/

[4] ISO-13606-1:2008 and others in the Series, Health informatics -- Electronic health record communication -- Part 1: Reference model.

[5] Eichelberg, M., Aden, T., Riesmeier, J., Dogac, A., Laleci, G.B., Electronic Health Record Standards – A Brief Overview, International Conference on Information and Communications Technology, 2006 [http://www.srdc.metu.edu.tr/webpage/projects/ride/publications/icict06_20060810.pdf]

[6] Meijer, E., Bierman, G., A Co-Relational Model of Data for Large Shared Data Banks, CACM, 2011 No. 4

[7] Christian Bizer, Tom Heath and Tim Berners-Lee. Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, Special Issue on Linked Data., 2009

[8] Helland, P., If You Have Too Much Data, then “Good Enough” is “Good Enough”, CACM, Database, Vol 54, No. 6, pp. 40-47

[9] Javascript Object Notation, http://www.json.org

[10] Fielding, R., Representational State Transfer, Ph.D. Thesis.UCI, 2000

[11] CouchDB, http://couchdb.apache.org

[12] Django Framework, The web framework for perfectionists with deadlines, http://www.djangoproject.com

[13] Apache Lucene. http://lucene.apache.org/

[14] Dean, Jeffrey & Ghemawat, Sanjay (2004). "MapReduce: Simplified Data Processing on Large Clusters"