Patient-Like-Mine

A Real Time, Visual Analytics Tool for Clinical Decision Support

Peter Li[1], Ph.D., Simon N. Yates[2], Jenna K. Lovely[3], Pharm.D. R.Ph. BCPS, David W. Larson[4], M.D. MBA
Department of Surgery, Mayo Clinic, Rochester, MN USA
[1]jenli.peter@gmail.com, [2]yates.simon@mayo.edu, [3]lovely.jenna@mayo.edu, [4]larson.david2@mayo.edu

Abstract— **We developed a real-time, visual analytics tool for clinical decision support. The system expands the "recall of past experience" approach that a provider (physician) uses to formulate a course of action for a given patient. By utilizing Big-Data techniques, we enable the provider to recall all similar patients from an institution's electronic medical record (EMR) repository, to explore "what-if" scenarios, and to collect these evidence-based cohorts for future statistical validation and pattern mining.**

Keywords- electronic medical record, clinical decision support, real-time analytics, visual analytics, data mining.

I. Introduction

When determining the course of action for a given patient, the physician has to integrate clinical knowledge, state of the patient, and his/her own personal experience. Clinical knowledge is composed of models of diseases and their progression and interventions. However, many patients have comorbidities and medical histories that cause the disease progression and the optimal treatment plan to deviate from the known clinical models. As a result, the physician (or, in general, the provider team) has to recall from their own experience of similar past cases to develop the care plan. Unfortunately, this approach is often subjective and limited to the team's ability to recall all relevant cases. We developed an interactive, visual analytics tool, "Patient-Like-Mine", in the Mayo Clinic's Enhanced Analytics to Surgical Excellence (EASE) program that focuses on improving Clinical Pathway compliance [1] as well as recognizing patterns early to assist this clinical decision making process. We remove the limits of subjectivity by accessing the institutional Electronic Medical Record (EMR) database so that the collective experience from *ALL* the past and present patients with a similar background could be utilized in real-time care planning.

Operational requirements of "Patient-Like-Mine" includes:

- Perform search on a large (>1 billion facts) and complex (>1 thousand properties) with up-to-date (>1 thousand data points/second) dataset in real-time.
- Align the resulting patients' (i.e. the cohort's) medical histories to provider-chosen "landmark" events, e.g. time of surgery or date of admission.
- Restrict the value and the relative time of *ANY* clinically relevant parameter based on the provider's judgment, e.g. "systolic blood pressure between 50 and 90, 3 days post-operation".
- Provide an interactive, graphical user interface to compare the given patient to "similar patients" and to explore "what-if" scenarios in a transparent fashion.

II. Approach

The architecture for this project is based on the following four major components:

- Automated deployment for Azure public (and private) cloud nodes with Health Insurance Portability and Accountability Act (HIPAA)-compliant disk encryption, strict firewall configurations, and secure login. The cloud is necessary for scalable performance, but we also need to secure the protected health information (PHI) that will now be placed onto the cloud nodes.
- A scalable search engine that can expand and shrink based on existing load. For this project, we chose ElasticSearch (http://elastic.co) for model simplicity (JSON-based), query expressiveness (structured and text), inherent performance (subsecond response), and secure transport protocol (SSL/TLS). We developed data transform and importing tools for Fast Healthcare Interoperability Resources (FHIR, http://hl7.org/fhir), Reference Information Model (RIM, http://hl7.org/implement/standards/rim.cfm), and SQL-based sources that captured EMR at Mayo Clinic.
- A schema-driven abstraction layer for ElasticSearch query building and data export. This generates consistent queries from declarative API calls and converts the return results from nested JSON to a tabular format for UI ingestion and downstream statistical analysis.
- An intuitive, graphical UI for exploring "Patient-Like-Mine". A single patient's structured clinical parameters can be plotted via a "strip-chart" graphic with a zoomable and scrollable interface. By extension, a set of "similar patients" in a cohort would paint a distribution contour over the same chart. By changing the constraints over clinical outcome and intervention parameters, we can analyze "what-if" scenarios to facilitate clinical decision making.

The project is divided into 3 phases. The first phase is architectural and implementation feasibility. Second phase is clinical practice validation. Third phase is distribution and expansion into additional clinical practices.

III. Results

We present the results from the recently completed first phase, demonstrating architectural and implementation feasibility.

A. Cloud Deployment

We developed an Azure-cloud deployment tool based on Linux shell scripts for ease of portability and extensibility. A given cloud deployment is defined by a set of configuration files that describe the system (IT) and application parameters (Figure 1). System configuration includes disk encryption via LUKS (for encryption-at-rest), firewall settings, and logins via public-key based ssh only. Passwords, certificates, and keys are generated on-the-fly so that each deployment will has its own set of security tokens.

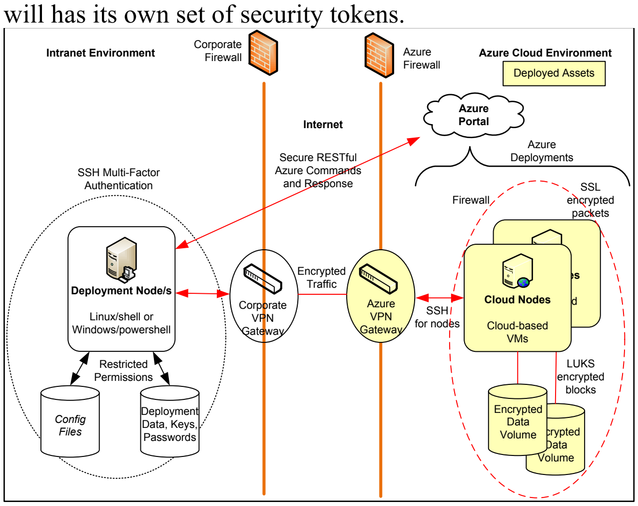

Figure 1. System architecture for cloud deployment.

Current applications supported are ElasticSearch (ES) and nodejs (Figure 2). ES can be deployed with SHIELD module for secure internode and client communication (for encryption-in-flight). While other ES modules, such as Kibana, are also deployed, but they are not considered to be secure at this time, and is limited to non-production environments. Nodejs is deployed on intranet/on-premise VM that has credentials to connect to local LDAP server for authentication and authorization. ES will only allow connections from this nodejs server via https.

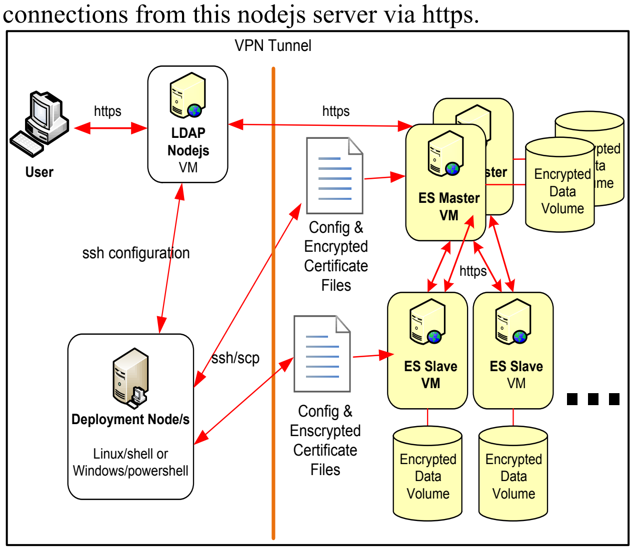

Figure 2. Application architecture for cloud deployment.

B. ElasticSearch Engine

ES is an open source, commercial product that serves as the search engine for many well-known internet sites (see http://elastic.co). It is built on top of Apache Lucene indexing engine (http://lucene.apache.org) processing JSON-based documents, but with distributed, multi-node scalability as a core feature. It has a very large set of querying commands, flexible aggregations, and parent-child relationship. In addition, it has a large ecosystem of tools and language support.

Typical EMR captures clinical data from a messaging format (HL7 v2 messages). However, with the industry acceptance of FHIR and RIM messaging objects, we are seeing more complex and polymorphic schema. These complex datasets require transformations to an analytics-friendly fact or event schema. We also took advantage of ES supported parent-child relationships and JSON-based nested array elements to provide additional properties for landmark event alignment and relative times (Figure 3).

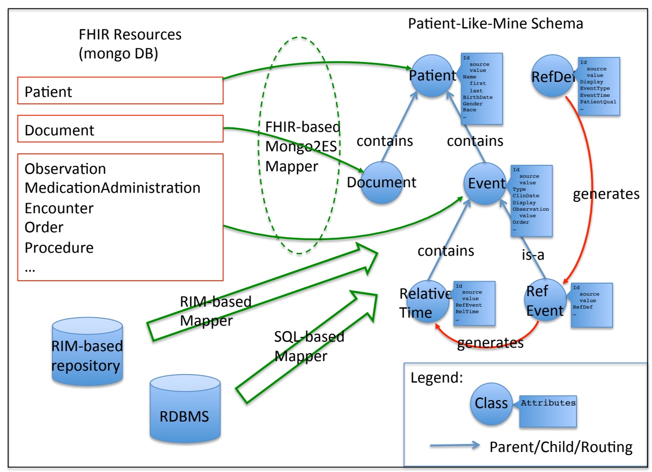

Figure 3. Mapping of HL7 schema to Event-based schema.

C. Schema-based Abstraction Module

While ElasticSearch query language is very expressive and powerful, it is also very easy to create queries that return non-intuitive answers because of the different contexts involving nested elements, parent-child relationships, and aggregates. Constraint clauses associated with different parent-child/nested elements need to be repeated at different places of the query construct to ensure proper object selection and projection (Figure 4). We built an abstraction module that takes a set of schema-based constraints and generates a consistent query for execution.

To simplify downstream processing of returned results from an ElasticSearch query, this abstraction module also includes a schema-based specification for transforming JSON-based data into a flatter table structure for use. This also simplifies exports to any statistical analysis that may need to be performed. This module is implemented in Javascript for use in both client browser and nodejs web server.

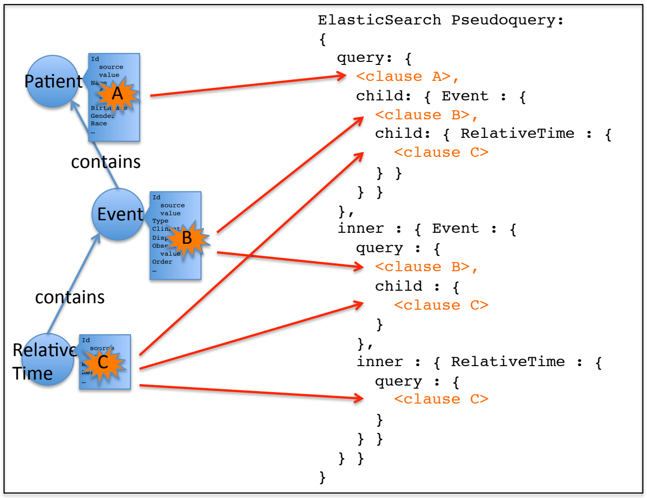

Figure 4. Constraint duplication for consistent results under ES.

D. Intuitive UI

Data overload is often a complaint about today's EMR. In addition to presenting the patient data, we will also be presenting data from 10's to 1000's of similar patients. The cohort data must be summarized so that a visual comparison can be made readily between the patient of interest and the cohort. We extend the box-and-whisker statistical plots into continuous contours to support continuous variables (Figure 5). This graphic allows the provider to quickly determine if the trajectory of a patient's clinical parameter is within nominal bounds, based on a cohort of similar patients.

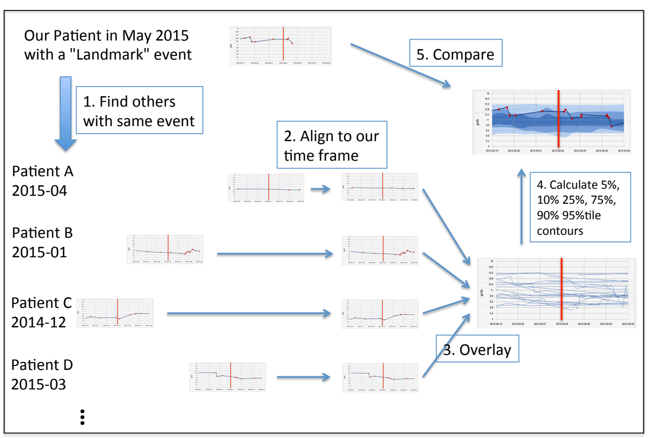

Figure 5. Construction of cohort contour graphs.

To build a specific real-time cohort, we allow the user to create, move, and size constraint "boxes" over the graph of the patient data. Each constraint box acts as a filter for matching patients whose data intersect the "bounding box" (Figure 6). A physician can decide when and where to place such constraints to create a specific cohort based on his/her clinical assessment, i.e. a customized, ad-hoc "patient-like-mine" cohort to predict patient outcome and to select best treatment options.

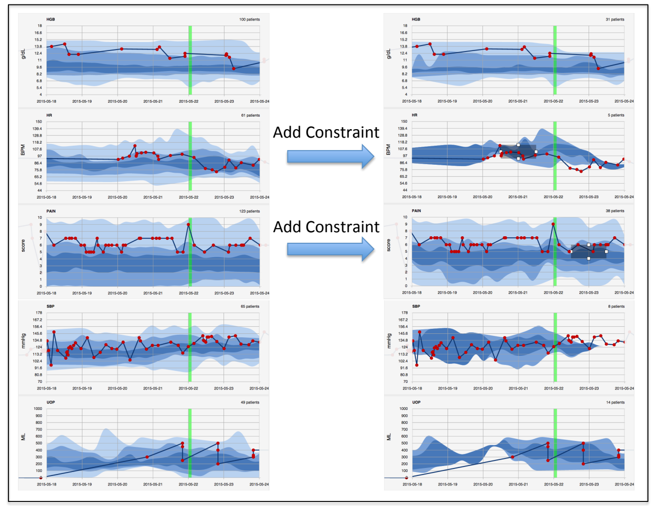

Figure 6. Impact of constraint creation on contours.

IV. Conclusion

The objective of the EASE program is to improve patient care and outcome. Our big-data approach creates a transparent, interactive environment that enables a provider to formulate more specific plan for a given patient using real-world evidence found in an institutional EMR. By using a very flexible UI, the provider or team can also explore "what-if" scenarios that would have previously taken a statistical/database team and considerable time to develop.

However, the number of "similar" patients in any given repository can be limited, reducing the robustness of this recall-based paradigm. By building this system on a cloud-based architecture, it can solve this problem simply by including more patients from other institutions. Thereby increasing the chance of finding patients that match a set of complex or rare clinical features.

Our next phase is to demonstrate clinical utility and relevance through deployment in a controlled practice setting that will provide systematic feedback for improvements and define novel use cases.

Acknowledgment

This project is funded by Mayo Clinic Clinical Practice Committee and the Office of Information and Knowledge Management. We also thank the efforts of the many people on the core EASE team and Mayo Clinic IT support staff.

References

[1] DW Larson, JK Lovely, RR Cima, EJ Dozois, H Chua, BG Wolff, JH Pemberton, RR Devine, M Huebner., "Outcomes after implementation of a multimodal standard care pathway for laparoscopic colorectal surgery.," Br J Surg. 2014 Jul;101(8):1023-30. doi: 10.1002/bjs.9534. Epub 2014 May 15.