ZDNet article – Semantic Data Lake Architecture in Healthcare and Beyond

Data lakes stink. That’s because lots of them turn to data swamps, and swamps stink. What’s the difference between a data lake and a data swamp?

A data lake is built on top of cost efficient infrastructure. More often than not these days this is Hadoop, leveraging two of it most alluring properties: Lots of storage for cheap and schema-on-read. That means you can store all your data and more now and worry about it later.

And that’s exactly what many organizations end up doing, resulting in a data swamp. A data swamp is a data lake where data goes to die: Without descriptive metadata and a mechanism to maintain it, you get a big pile of data that is effectively unusable.

A part of this has to do with Hadoop, as support for metadata and data governance has been one of its sore points. The situation there is improving, but there still are a couple of issues.

The first one is obvious: Even the greatest tools are no use if you don’t put them to use. So the fact that there is the option to add metadata to your data does not mean that everyone does it.

The second one is that not all metadata are created equal. When we start talking about descriptive metadata, the need for semantics quickly becomes pronounced. So what do we get when we add semantics on top of data lakes? Semantic data lakes.

HEALTHCARE CHALLENGES, SEMANTIC DATA LAKE SOLUTIONS

Montefiore Health System has implemented a semantic data lake (SDL), and we discuss with Franz Inc., the provider of the semantic element, about overall architecture and the role of semantics.

Located in the Bronx, Montefiore Health System serves one of the most ethnically and socioeconomically diverse populations in the US. The complex includes, but is not limited to, Montefiore Medical Center, Albert Einstein College of Medicine, and a research facility.

Like all healthcare organizations, Montefiore faces many data-related challenges. As Dr. Andrew D. Racine, system senior vice president and chief medical officer at Montefiore puts it:

“The challenge where you’ve got hundreds of thousands of patients impacting the institution at any given point is to have the appropriate information about each one of those patients at the fingertips of the therapist who’s interacting with them at the time of that interaction.”

Montefiore is using its varied and vast amounts of raw data for deeper analysis to flag patients who are at risk or help clinicians identify optimal treatment plans. In order to be able to build such advanced analytics solutions, Montefiore has deployed a Semantic Data Lake (SDL) utilizing an array of technologies and components.

The SDL solution provides capabilities that include:

- Predictive analytics at scale to anticipate and account for various patient outcomes in timeframes in which treatment can be administered to affect care.
- Machine learning algorithms to integrate the results of

previous outcomes that significantly impact the analysis and effects of future patient objectives.
- Disposable data marts to quickly provision project-specific environments to manipulate data and analytics results without duplication or redundancy.
- Ontological pipeline to rapidly integrate new data sources and requirements into existing models, and validate the clinical process for highly targeted patient subsets.

AN ONTOLOGICAL DATA PIPELINE

An ontological data pipeline sounds fancy, but what is it exactly and why should you care? It's a data pipeline in which incoming data is annotated with metadata using an ontology. An ontology is arguably the most advanced form of schema around in terms of its ability to capture semantics, hence the semantic aspect of the data lake.

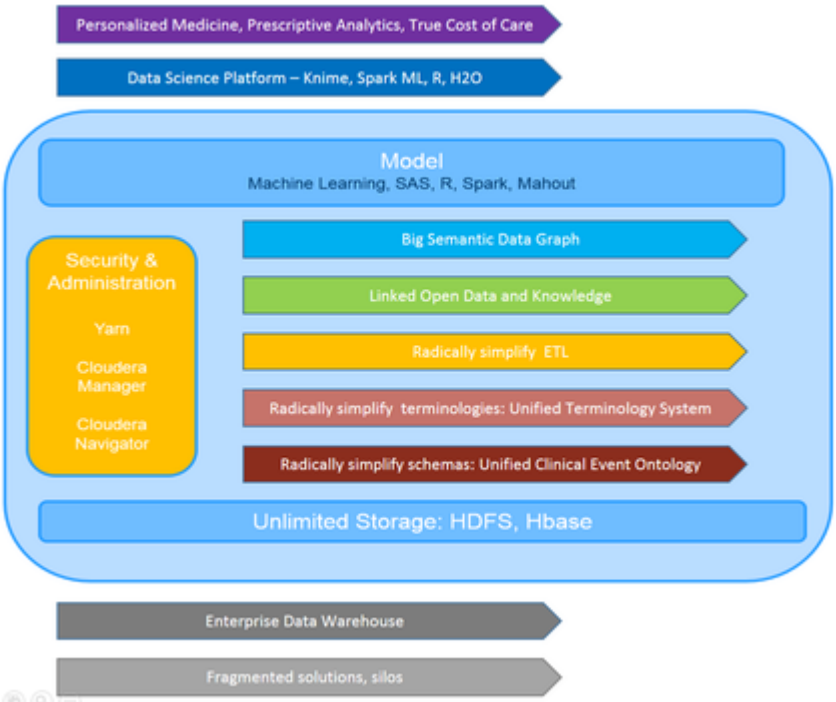
We discussed the approach and architecture with Dr. Jans Aasman, CEO of Franz, Inc. Franz Inc. is the vendor behind AllegroGraph, the RDF graph database that handles the descriptive metadata/ontological pipeline aspect of the solution.

Aasman explains that the SDL supports both fast real time input (for example HL7 streams) and large, batch oriented bulk inserts from ETL (Extract Transform Load) processes.

But the million-dollar question is how does the semantic annotation happen. Are all data that enter the lake already annotated upon ingestion, or is there further annotation required? How is it performed – automatically, semi-automatically, manually? Are there tools for this?

Aasman says they use a visual ETL tool to draw a mapping between data in the EDW or HL7 streams to a healthcare ontology that covers everything that could ever happen to a

patient in the hospital life cycle:



Semantic data lake architecture. (Image: Franz Inc.)

“This creates a declarative mapping that is read in by a Java program that automatically transforms (mostly) relational data into a graph representation (aka triples). Every element in the graph is annotated by the table and column it came from and the ETL date.

“In addition, we annotate every triple with what we call ‘triple attributes’ that enable us to selectively make data available for users in their different roles. This is a spectacular new feature in AllegroGraph that we will be publicly announcing soon.

“In this setting, vocabulary management is extremely important. Healthcare has more than 180 vocabularies, taxonomies and terminology systems, such as Mesh, Snomed, UMLS, LOINC, RxNorm, etc.”

Data integration is one of the strong points of ontological modeling, and Aasman says that these taxonomies are all interconnected and linked to important ‘real life’ concepts like ICD9 and ICD10, procedure codes and NDC for medications:

“This combined and integrated terminology system (the healthcare ontology) is at the heart of the ETL process, and is incredibly important for queries and analytics,” he says.

SPARQL OVER SPARK

Ontologies and graph databases sound great and all, but there’s more to the SDL solution. Where and how exactly does ontological modeling and AllegroGraph fit in the big picture?

Aasman explains: “We run distributed AllegroGraph on a Cloudera cluster. We can read/write from HDFS and we can run Spark on top and use MLlib for our analytics. Distributed AllegroGraph, the database underneath the SDL architecture, provides all the features of a Lambda architecture.”

That’s an unusual choice, one which means for example that instead of SQL, SPARQL is used as the query language. Why go for it? And how well does it perform compared to more conventional solutions?

“Relational databases do great when your data fits in relatively simple schema, there is no network in your data and you do big aggregate queries. Graph databases do better when you do graph algorithms where it is unpredictable how deep your graph algorithm will go.

“In addition, graph databases perform far better when you have a lot of ad hoc queries or when your data is ridiculously complex or if your application will benefit from reasoning,” Aasman says.

What about query complexity? Aasman says that as a vendor they see queries ranging from one line to 1,500 lines of code, and provided a typical SPARQL query from the Montefiore project for good measure:

A real-world SPARQL query from the Montefiore use case. (Image: Franz Inc.)

“This query finds the top 100 patients that are most similar to one particular patient from a set of 2.7 million patients. The first subquery finds for a particular patient his or her gender and race and all the icd9 codes.

“Because these icd9 codes are very specific, we link the icd9 codes to concepts in our knowledge base and we go up the terminology ladder recursive way and then down again to find all family members of that icd9 code.

“Once we have those we find all the other patients that have the highest overlap in icd9 codes (well, the super members) with our start patient. This is another example of the compactness of SPARQL.

“We can also use Spark to do a SPARQL query against distributed AllegroGraph. We use Spark for analytics and then we can save the results of analytics back into AllegroGraph as newly learned information,” he says.

The SDL supports both fast real time input and large, batch oriented bulk inserts from ETL processes. AllegroGraph is an append only graph database, explains Aasman, so new data are appended to the existing indices:

“There are continuous background optimization processes that merge all the chunks of data into one linearly sorted index space, but the reality is that if data is streaming 24/7 the indices are never perfectly sorted so the query engine has to look both in the existing indices and appended new chunks.”

GRAPH BROWSERS, TIME MACHINES AND MACHINE LEARNING

Aasman adds that Gruff, AllegroGraph's graph browser, allows users to visually create a query and then generate SPARQL (or Prolog) query code. Franz Inc just released a new version of Gruff, adding what they call "Time Machine" capabilities to it.

Many use cases for graph databases involve temporal events. Events are modeled as objects that have a start time, end time, a type, some actors and a geospatial location.

Aasman says Gruff v7.0's new time slider feature enables users to visually demonstrate how graphs comprised of temporal events are constructed over time, allowing time machine like exploration of your data.

Last but not least, the Machine Learning part. This is not something graph databases typically offer, so how does it work for AllegroGraph?

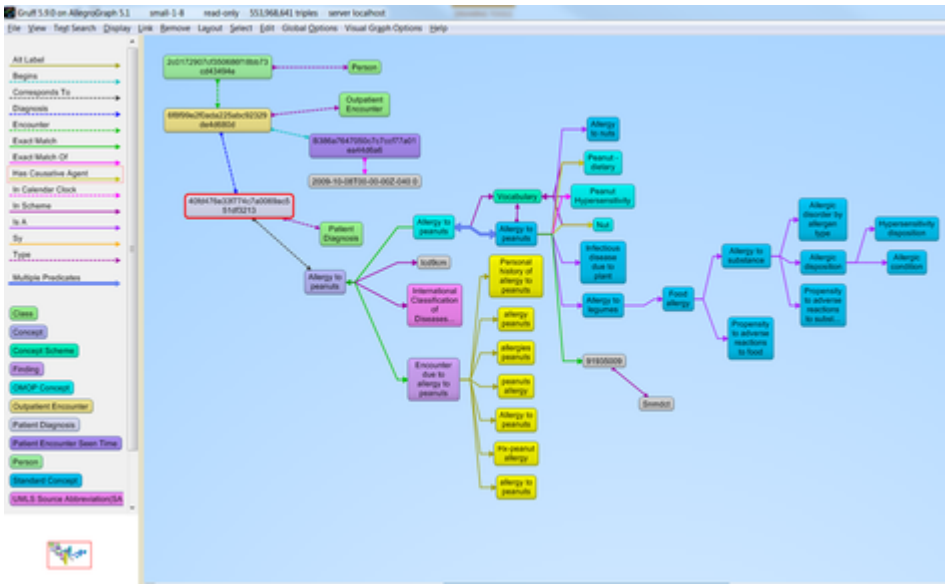
Data scientists don't really care what they do their analytics against, claims Aasman, as long as they can get their feature sets from the underlying data store as a csv file, or even better, as a (panda) data frame.

"To make life more simple for data scientists that want to work with AllegroGraph we currently have an open source R interface and an open source AllegroGraph – Python interface that is directly installable via Anaconda.

"However, we have an even better integration point and that is that we put all the results of analytics back in AllegroGraph as triples and then make that navigable via Gruff.

"See an example below. Not only do we store all the results, but also the metadata about the results, such as: who did the analysis, when, what scripts were used, what data sets were

used, etc," he says.



Rich metadata is one of the benefits of semantic data lakes. (Image: Franz Inc.)

SEMANTIC DATA LAKES IN THE CLOUD?

This looks like a good way to play on each individual system's strengths in an SDL solution, although the range of technologies uses make it a rather complex one. Would it not help if organizations had access to such productized solutions in the cloud?

Franz Inc provides off the shelf tools like AllegroGraph as part of the implementation along with bespoke tools and programming for a complete solution. For Montefiore, the solution is deployed on a local cluster of machines in their data center.

Aasman says that most hospitals are not used to putting their data in the cloud yet, but with HIPAA compliance by Amazon, Azure and Google Cloud the future will be in the cloud, also for Montefiore. Still Aasman feels that local clusters are better for the time being, for 2 reasons.

The first one is convenience: "It is really convenient to have a local cluster for development that you can deploy directly

to a similar production cluster. We can easily reinstall kernels, fix security issues, and minimize deployment time."

The second one is price: "All graph databases are more performant with high performance SSDs and lots of RAM – if the data is much bigger than memory. We find that large memory machines with SSDs in the cloud are still very expensive."

Aasman adds that they see a lot of demand for AllegroGraph in the cloud, primarily on AWS and they are currently exploring AWS for the US Intelligence Community. Franz Inc used to offer a managed service in the cloud, but Aasman believes it was ahead of its time as most of their customers wanted to maintain control.

Aasman however sees opportunities in developing managed taxonomies and ontologies that are domain specific and plan to revisit this offering next year. It would probably make sense for many organizations interested in SDLs to be able to offload as much of the know-how and workload to the cloud as possible.