Using Connected Data to Empower a Financial Services Organization: Project Helix at UBS

Gilad Geron[1], Tony Hammond[2], and Ilya Venger[2]

[1] UBS Business Solutions AG, Badenerstrasse 574, Zurich, Switzerland
[2] UBS Business Solutions AG, 5 Broadgate, London, EC2M 2QS, United Kingdom
{gilad.geron, tony-za.hammond, ilya.venger}@ubs.com

1 Motivation and current situation

UBS is the world's largest wealth manager with a diverse financial portfolio. Its technology landscape supports core financial activities as well as enterprise management and other enabling capabilities. UBS Group Technology recently introduced four 'big bets' – a series of initiatives aimed at accelerating developer productivity while streamlining and increasing controls. Big Bet #3 is aimed at making a step change in the understanding of our technology landscape.

As is the case with many enterprises, data is often managed on an application level. A big picture, holistic view requires harmonization of data models, identifiers and adding semantics to connect diverse knowledge domains across the organization.

Helix is a project under Big Bet #3 to build out an enterprise knowledge graph (EKG). The unique capability of an EKG is the ability to gain insights over disparate datasets. Strong management buy-in from the start helped establish RDF as a critical technology for achieving large-scale data integration. To deliver on the vision of an enterprise-wide knowledge graph we partnered with Cambridge Semantics and their product Anzo. We selected this tool for its end-to-end capability from data ingest to analytics and distribution. One of the unique features of Anzo is Anzograph – a scalable in-memory graph store providing for fast query lookups. Another powerful feature is its very strong support for executing chains of SPARQL queries which allows us to enrich our models incrementally.

The ability to reason about technology and the representation of data (physical, logical, conceptual) is a necessary foundation to realizing a consistent and scalable knowledge graph. To achieve this we have started by building out a set of OWL-based domain ontologies describing technology and data assets. Focusing on relational databases first (where most of our data resides), we wanted to leverage the existing metadata repositories, schemas and the interrelationships between data points to derive ontologies consistent with the current world models. We have developed a mapping model and a methodology based on RDF

shapes to express relationships and link from physical columns to ontologies capturing the expert domain knowledge.

We present two case studies onboarding legacy models and connecting instance data.

2 Exposing existing models

Our main configuration management database (CMDB) holds extremely rich information about technology assets and has been our primary source to start building out a knowledge graph in the technology domain. As well as instance data, the CMDB includes a semantic metadata repository which holds definitions for all classes of assets (component and system types) and a catalog of all the parameters. However, relationships between objects are not explicitly stored and the graph-like structure of the data is not explicitly used.

To create an application-based ontology on this two hundred million triples dataset, we implemented the following steps:

1. ingested and RDF'ized the contents of the database and the metadata
2. ran a set of SPARQL queries on top of the CMDB metamodel to create an ontology with classes and properties
3. instantiated the contents of the database and assigned to classes
4. queried the instance data to augment the ontology with object/datatype property distinctions and domain–range relationships
5. connected the resulting graph (expressed in our model) with other domain ontologies by running queries on instance data across domains and manual enrichment

3 Mapping with shapes

For rapid expansion of the graph to multiple domains we leveraged a central data warehouse which had information from the core CMDB alongside multiple other sources. It was simple to generate a number of CSVs 'walking' across the data landscape. The Anzo ingestion engine onboards tabular data into an RDF dataset with an auto-generated model (based on source tables and column names). However, to supplement the data with semantics we needed to 'lift' this RDF dataset into our own semantically-rich domain models.

We created a set of RDF shapes based on our model for each of the ingest classes overlaid with type and annotation property mappings. At runtime, a single SPARQL query runs over the shapes, datasets and our own models materializing triples into a named graph. The query also generated names for the new data instances using typed namespaces from our ontologies. This has proven to be highly effective with a pilot small dataset size of tens of millions of triples and a couple of dozen shapes.