Semantic Ontologies and Financial Reporting: An Application of the FIBO

Oliver BROWNE [a,1], Nenad KRDZAVAC [a], Philip O'REILLY [a], Mark HUTCHINSON [a]
[a] *Accounting, Finance and Information Systems, University College Cork, Ireland*

Abstract. This paper illustrates the application of a developed global fund reporting ontology (GFRO) for efficient financial reporting. The GFRO extends Financial Industry Business Ontology (FIBO). Existing reporting financial information systems lack the ability to integrate data from heterogenic sources and provide unified and consistent financial reports that will comply with regulations. This study reveals that by integrating the power of XSLT and Semantic Web technologies, operationalised through the development of a scalable working prototype, allows financial services industry experts to build more flexible and consistent reports. Our research shows that the consistency of financial reports can be dramatically improved by using an appropriate inference engine.

Keywords. Ontologies, FIBO, financial reporting, information systems, reasoning

Introduction

Data required for reporting to regulatory bodies is open to interpretation, from teams of legal experts to data analysts and senior management, interpretation of legal documents can leave experts arguing ad infinitum about minor nuances of language and terminology. These challenges have spurred the need for the development of a standardised language across financial instruments and institutions, a terms sets such that there is little room for interpretation and the regulator receives transparent and comparable data from all institutions for aggregation, and to be able to prove that the report is consistent with regulations which are constantly changing and developing [1].

Towards addressing this shortcoming, we have developed GFRO and implemented a framework that uses GFRO to provide consistent and unified financial reporting across heterogenic data from different sources. Our research aims to advance research on ontologies by illustrating their application for improved reporting capabilities over a broad subset of financial instruments; specifically, bonds and equities, and to perform reasoning over source data to infer new observations.

Bonds are a debt investment instrument in which an investor loans money to an entity, the entity borrows the money for a fixed period of time and repays moneys to the investor as incremental interest payments with a bullet payment of the principle at a set date. Many variations of bonds coupon and principle payments also exist. Equities are a piece of ownership and generally control of an entity, generally referred to as shares or stocks.

[1] Corresponding Author

1. Main Ontology-based Features of the Framework for Financial Reporting

The most important task in adapting the FIBO [1] standard to serve as a core in financial reporting, is to bridge the gap between domain specific databases and FIBO ontologies. Table 1 contains a list of extensions that were made to FIBO in this research study to support GFRO.

Table 1. Sample list of extensions to FIBO

FIBO: extended features	Description
Full Service Fund	Service level provided by fund.
Mixed Fund Classification	The investment strategy for the asset allocation of the fund.
Real Estate Investment Trust	Type of equity instrument not represented in conceptual taxonomy.
Bond Lot Number	Lot number of the holding of the bond instrument, necessary in case purchases of same instrument made on different dates and aggregated in data.
Gain or Loss	Unrealized and realized gains and losses are paramount in the ongoing valuation of funds.
Accrued Interest Money Amount	Accrued interest represented as a monetary value as opposed to a percentage of par.
MMIF Yield	Indicator of implementation of specific regulator required formula for reporting of data.
European Market Infrastructure Reporting (EMIR) Indicator	Indicates that an asset must be included as part of an EMIR report.

Figure 1 presents an overview of the POC architecture. It reveals the process of generating the resulting RDF graph starting from raw data. The first step is to extract data necessary for reporting into csv files. Each csv file is converted into appropriate XML and then converted by XSLT transformations to appropriate RDF representation.

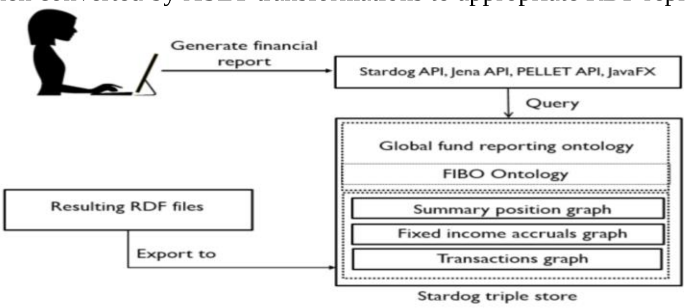

Figure 1. Architecture of consistent reporting using FIBO

We generate three types of rdf graphs. The first one covers point in time representations of the financial instruments i.e. month end data. The second transformation covers transactional data of financial instruments i.e. buying and selling of positions. The last one covers accrued interest to certain financial instruments such as bonds.

The second component of the implementation establishes communication between the end user and the triple store. It uses parametrized SPARQL queries to deliver data needed to generate financial reports, or conversely, return exceptions in case of inconsistencies. On top of this, is the GUI layer, which implements a graphical user interface that allows subject matter experts to create custom intelligent reports over financial data including regulatory reports, automatically populating regulatory reporting templates. The end user can produce reports across many funds or fund types over features of that fund. We use the *JavaFX* [2] library to generate different type of charts.

Figure 2 illustrates a UML class diagram of the implementation of all layers. The central class is *GenerateReport*. Method *getTransformation* runs the XSLT transformation of financial data stored in a folder and generates appropriate RDF file, and exports that file into the triple store, also loading GFRO into the triple store. The system runs SPARQL queries necessary for reporting of financial instruments. Some of these queries are parametrized, and some of them are not. For example, parametrized queries for bond reporting are reusable for any type of bond. All queries are stored in one folder.

When an end user submits a request for generating a report, then method *runQueries()* in Query class runs all queries over Stardog [3] triple store, and generates results that are stored in the output folder as a csv file as well as remembered results as properties in bean classes such as *FinancialInstrumentBean* class. The *Query* class allows a user to query more than one graph in the triple store. Methods in *GenerateReport* class such as *getPieChart()* uses csv files and methods in bean classes to generate and visualize report to end user.

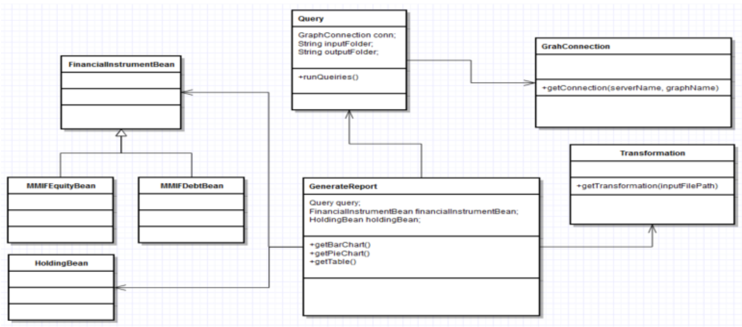

Figure 2. (B): UML class diagram of financial reporting service

2. POC Output: Illustrative Example

Figure 3 provides a view of a worked sample of a dynamic fund report. This report provides a summary view of the fund at a point in time represented in an easily digestible manner by finance subject matter experts.

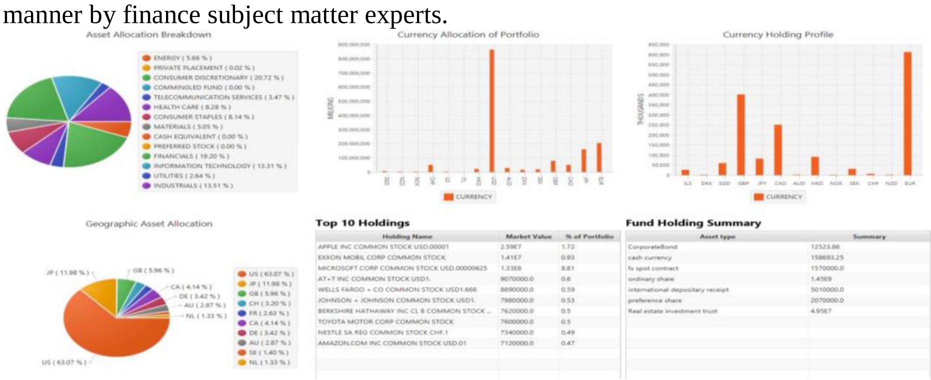

Figure 3. Sample representation of a fund at point in time.

Clockwise from bottom left; we represent the equity holdings of the fund, by country, and next by sectoral allocation of the equities as pie charts. The bar charts are used to represent currency allocation of the equities within the fund and the physical cash holding of the fund. Data in the tables' represent overall summary data. The representation of the data is standard in its presentation; however, the method of retrieval and querying is unique in its flexibility.

Databases find it difficult to query from a data end point and must be queried more generally and data collated and validated [3], this approach allows us to be flexible over querying by using shared characteristic over reasoning, removing the need for collation of spreadsheets and manual processes.

3. Conclusions and Future Work

The paper demonstrates a framework for consistent financial reporting that complies with regulations by adapting and extending FIBO to meet the requirements of the data. This a flexible approach to extending current reporting over legacy database systems rather than design new databases.

Although we tested the process described in the paper over a large amount of complex and varied data representative of fund level data, a limitation of our research is that we did not test beyond the scope of bonds and equities. The next stage of our research is to explore the utility of our framework in a bid data setting.

This application of regulatory and risk reporting over a complex set of data and the ability to automatically verify results over reasoning should allow further research into the benefits of ontologies. The adoption of a shared ontology could potentially dramatically reduce the timeframe for transaction processing. The financial ontology standards (FIBO) are at early stages of development, these standards are conceptually strong but lack the testing to allow for robust implementation. All findings were reported to the EDM Council and Object Management Group to help to improve the FIBO standard wherever possible [5].

4. Acknowledgements

The proof of concept outlined in this paper was funded by the State Street Corporation. In particular we would like to attribute the ongoing success and funding of this project to David Saul, Dáire Lawlor, Daragh McGetrick

References

[1] M. Bennett, The financial industry business ontology: Best practice for big data, *J. Bank. Regul.* **14** (2013) 255–268. doi:10.1057/jbr.2013.13.

[2] Java FX, http://docs.oracle.com/javase/8/javase-clienttechnologies.htm, Accessed on September, 2017.

[3] Stardog, 2017, http://stardog.com/ Accessed on September, 2017.

[4] G.W. Dickson, R.L. Leitheser, J.C. Wetherbe, and M. Nechis, Key Information Systems Issues for the 1980's, MIS Q. 8 (1984) 135–159. https://doi.org/10.2307/248662

[5] FIBO, https://www.edmcouncil.org/financialbusiness