Welcome!

FIBO relationship to other Financial Industry Standards

The XBRL Bank Call Report (FFIEC 031) in FIBO
Jurgen Ziemer
Ontologist at Jayzed Data Models Inc.
FIBO is key to better Data Management.
And so are Industry Standards like ISO20020 FpML, FIX, Mismo.

Getting started with FIBO and integrating industry standards can be a challenge.

This presentation shows how to integrate the XBRL industry standard.
Leveraging FIBO includes “Proof & Trust” and “Managing Complexity” use cases. For the business side, Finance Officers & Regulators Proof and Trust in the data is critical. Proof & trust are the upper layers of the semantic web.

- How the data was derived
- On which data
- By whom

For program managers and architects the goals is to better manage complexity.

The slides have a heading of
Key points (finance)
Explanation
Details
Roughly corresponding to the roles.
Has anybody seen this picture?

A typical 1960 open-plan office.

I saw this 15 years ago on my first Basel 2 assignment. Saw this literally again with CCAR. EDMC helps to manage complexity DCAM and FIBO can reduce the complexity.
The bright future!

Industry standards and compliance are excellent drivers for FIBO implementation. Because they have good definitions and clean data.

Semantic compliance with FIBO reduces data management complexity. Because data, mapping, lineage – everything is a triple.
We have done Extract, Transform and Load in conventional IT. The semantic data migration model is no different. **The critical point** is that the Staging Area is in Ontology Web Language. Thus uniform Semantic rules address the critical business **transformation** rules.

Step 1: Extract the XBRL into Ontology Staging
Step 2: Transform Staging and load into FIBO

Not just for XBRL. **This applies to all data sources:** database, messages, files.

- The Federal Deposit Insurance Corporation is author and auditor.
- The Federal Financial Institution Examination Council is an interagency body serving FDIC, Federal Reserve and other agencies. FFIEC prescribes standards and forms, processes filings and disseminates reports to the public.

The Call report format is XBRL.

We reverse engineer the XBRL schema into Ontology classes.
The Open Source XBRL Ontology is a complete 1-to-1 representation of the XBRL schema. Every element in the XBRL schema is represented in the ontology. All XBRL files have a corresponding ontology.

With XBRL based sources, we won’t be impacted with structural changes.
Each namespaces in the XML schema files becomes an ontology file.

The XBRL ontology has all extensions, some 40 files for Solvency. But for the Call Report we only need the 3 base modules.

We use TopQuadrant's Topbraid Composer as the ontology development environment. But the approach, methodology and architecture is independent of the tooling.
The diagram shows the XSD schema for Monetary Items with its attributes for Precision and number of Decimals.
The reverse engineered OWL class has restrictions that reflect the XSD cardinality.
(optional).
The sxml:tag point to the original XSD definition.
The XBRL instance import processing a Monetary Item will find the tag, locate the OQL class and create a resource.
Likewise, the export will write an XML record with S value.
XBRL is not specific to Financial Services.

The taxonomy differentiates Bank Call Report from Insurance Solvency.

Thus, we map taxonomies to FIBO – not the XBRL framework.
FTIEC provides schema and Bank call reports

The taxonomy download is a ZIP with all required files. The individual institution report is available in PDF, SDF (open with Excel), and XBRL.
The Bank ontology has an OWL version of the FFIEC taxonomy

- The XBRL instance ontology
- FFIEC extensions to XBRL Item Types
- FFIEC Micro Data Reference Manual (MDRM) concepts that define instance items. >4,200 subclasses of instance:Item
- The reverse engineered XBRL instance (JPM Chase 09/30/2016) that hold the data values.
The FFIEC instance file is a simple list of reporting items values.

```xml
<cc:RSSD9017
certainty="95"
contextRef="CI_852218_2016-09-30">
JPMorgan Chase Bank, National Association</cc:RSSD9017>
```

The import generates instances of the FFIEC concept class - here RSSD9017 (Legal Title of bank).

The instance file is just a header and some 2000 lines of lines like this. This also makes it very easy to generate XBRL filing out of FIBO. Just put the values in lines following the syntax.
A query on the populated ontology matches the FFIEC individual institution download.

<table>
<thead>
<tr>
<th>Short_Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting date</td>
<td>20160930</td>
</tr>
<tr>
<td>FDIC certificate number</td>
<td>628</td>
</tr>
<tr>
<td>Legal title of bank</td>
<td>JPMorgan Chase Bank, National Association</td>
</tr>
<tr>
<td>City</td>
<td>Columbus</td>
</tr>
<tr>
<td>State abbreviation</td>
<td>OH</td>
</tr>
<tr>
<td>Zip code</td>
<td>43240</td>
</tr>
</tbody>
</table>

FFIEC download.  
Ontology query results

The left is a copy of the PDF page.
We reverse engineer the FFIEC taxonomy into OWL

The taxonomy ontologies import XBRL-linkbase: edit checks, calculations, presentation, definitions, captions, instructions, and edit check messages.

The taxonomy files are needed to query and report out of OWL staging. We only require ffiec-cep (concepts) for data import.
We validate the import reproducing the FFIEC report in SPARQL, the ontology query language.

```sparql
SELECT ?Call_Date ?Bank_RSSD_Identifier ?NDRM ?Value
WHERE {
  ?end_date_inst a instance:ContextPeriodType .
  ?end_date_inst instant ?instant .
  BIND (xs:string(?instant) as ?Call_Date) .
  ?ident a instance:ContextEntityType-identifier .
  BIND ( ?n:enوسطرین (91b_label_1or_2) as ?NDRM) .
  ?91b_label_1or_2 rdf:type rdfs:label ?NDRM .
  ?office_concept rdfs:subClassOf* instance:ItemAttrro .
  ?concept_inst a Office_concept .
}
ORDER BY ?Call_Schedule /?Line_number
```

Similar to SQL, the `SELECT` specifies the result set columns.

The `WHERE` clause joins the triplets.

ORDER BY provides the sorting.

The biggest challenge for SQL folks is that the `SELECT` doesn’t separate with comma.
The FFIEC excel (SDF on top) and SPARQL result set.

https://cdr.ffiec.gov/public/ManageFacsimiles.aspx

http://bankontology.com/br/query/Call%20Report%20PMC%2020160930%20Query%20Results.xlsx

The complete query and also a SELECT * is on the website.
Step 2 defines the complex business logic. It is an OWL to OWL ontology transformation. The inference engine (reasoner) examines the mapping and executes SPARQL constructs to move the data.

All within the ontology – all semantic web technology.
We graphically map FFIEC XBRL MDRM into FIBO classes.

RCON9224 is the MDRM element for the Legal Entity Identifier. We map the element to FIBO LegalEntityIdentifier class. We also map the value to the FIBO data property. The inference engine will
1. Construct an instance of LegalEntityIdentifier
2. Construct a data property, UniqueIdentifier
The Legal Entity ID provides the URI local name

The argument is the value of the Legal Entity Identifier in reporting item RSSD9017.

The template builds the URI as
FIBO-prefix : FIBO-class _ “LEI” argument

http://www.omg.org/spec/EDMC/FIBO/BE/LegalEntities/LEIEntities/LegalEntityIdentifier.LEI.7H6GLXDRUGQFU57RNE97

We anchor our target URIs around the LEI. FIBO Corporation, Depository Institution, Address and MonetaryAmount URIs are also based in LEI.
The data property mapping assigns the MDRM value to the FIBO unique identifier.

The mapping context from Staging class to FIBO class

Source and target data property

(left blank)
SPARQL rules specify complex data transformations.

The hasSourceInstance object property provides data lineage. For every FIBO instance, we have a link to its original instance in the Call Report.

```
spinrule

CONSTRUCT { 
    ?targetLEI ?ro-rethasSourceInstance ?this .
}
WHERE { 
    BIND (spinmap:targetResource(?this, ffiec-031-spin-fiborCCONS224-LegalEntityIdentifier) AS ?targetLEI) .
}
```

The **BIND** function uses the mapping context to determine the FIBO target instance.
We run the inference engine out of Topbraid. It take quite a while because in development we include the full FIBO ontology.
FIBO has a detailed taxonomy of Organization (48 subclasses) and Financial Service Provider (46 subclasses). Unfortunately, they are all primitive classes. We must assert that our FFIEC import is a Stock Corporation and Depository Institution.

It would be better to have more Defined Classes. The inference engine rather than the mapping architect should determine the type of bank.
• FIBO differentiate between the Agent (Stock Corporation) and the Role (Depository Institution) it plays. The two are connected via the hasIdentity object property.
• The LegalEntityIdentifier identifies the Corporation.
• The Corporation has a registered Address.
• Address links to state and country, already in FIBO.
• FIBO stores Capital (and others) as Monetary Amount. The number plus semantics, like currency and date.
• The Depository Institution (not the corporation!) has an FDIC Certificate number. We use this number to create the URI.
• The certificate ties up to FDIC and FDIC directory, already in FIBO.

Nice design. But FIBO need more support for Accounts and Balances!
The data query traverses the joins and shows data properties

```sql
WHERE {
  ?corporation fibo-hd-aap-aqt:hasName ?institution .
  ?corporation fibo-be-le-fbo:hasRegisteredAddress ?registered_address .
}
```

<table>
<thead>
<tr>
<th>institution_name</th>
<th>JPMorgan Chase Bank, National Association</th>
</tr>
</thead>
<tbody>
<tr>
<td>lei</td>
<td>7H6GLXDRUGQFU57RNE97</td>
</tr>
<tr>
<td>country_name</td>
<td>United States</td>
</tr>
<tr>
<td>state_name</td>
<td>Ohio</td>
</tr>
<tr>
<td>issued_capital</td>
<td>$176,083,000,000</td>
</tr>
<tr>
<td>fdic_certificate</td>
<td>628</td>
</tr>
<tr>
<td>regulator</td>
<td>FDIC</td>
</tr>
<tr>
<td>registration_directory</td>
<td>FDIC Institution Directory</td>
</tr>
</tbody>
</table>
This is the great value ontology provides for data management. Besides class mapping we can also dive down to query property mappings and rules.
Again, the value that ontology adds to data management. **Data Lineage is inherent**
The lineage query results trace the FIBO instance back to its XBRL source record.

<table>
<thead>
<tr>
<th>fibo_class</th>
<th>fibo_inst</th>
<th>call_cert_inst</th>
<th>ffier_class</th>
<th>xml_tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>StockCorporation</td>
<td>LXDRUGQFUS7RNE97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fibo-be-la-lei:</td>
<td>LegalEntityIdentifier_1 7I7H61</td>
<td></td>
<td></td>
<td><a href="http://www.ffiec.gov/xbrl/call/conceptsRCC0974">http://www.ffiec.gov/xbrl/call/conceptsRCC0974</a></td>
</tr>
<tr>
<td>LegalEntityIdentifier</td>
<td>G1YXANEGQ5US78890E7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fibo-fbic-fct-fec:</td>
<td>DepositoryInstitution_LEI_7</td>
<td></td>
<td></td>
<td><a href="http://www.ffiec.gov/xbrl/call/conceptsRSS017">http://www.ffiec.gov/xbrl/call/conceptsRSS017</a></td>
</tr>
<tr>
<td>DepositoryInstitution</td>
<td>H6GCDKDRUGQFUS7RNE97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fibo-fbc-fct-ucjrg:</td>
<td>FIDCCom.LifeteNumber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIDCCom.LifeteNumber</td>
<td>628</td>
<td></td>
<td></td>
<td><a href="http://www.ffiec.gov/xbrl/call/conceptsRSS017">http://www.ffiec.gov/xbrl/call/conceptsRSS017</a></td>
</tr>
</tbody>
</table>

FIBO | Ontology Staging | XBRL Source

(left blank)
The complex transformation and business rules are defined OWL-to-OWL, within the ontology. The export, conversion from OWL to XBRL is just mechanical, tooling.
Map FIBO into FFIEC 031 staging classes.

Our example, the FIBO registration address maps to 3 FFIEC MDRMs.

RSSD9130 – city
RSSD9200 – state
RSSD9220 – zip code
The mapping is a reversal of the Staging to FIBO mapping context.

Unfortunately there is no “reverse mapping” ETL button.

We have to specify
• How to build (match) the Staging URI
• Transform FIBO formats into FFIEC formats.
• The UPDATE operation is still a challenge.

The Staging to FIBO mapping context query, provides our to-do list.

The XBRL import assigns system generated URIs. We must match the correct instance URI in order to update. Or should we just create a completely new instance ontology?

The answer also depends on, whether we look at a single institution filing every 3 months or a consolidator / regulator holding multiple institutions in OWL Staging.
Test case: JPM Chase moves to Vermont

We change the bank’s city, subdivision and postal code.
We run the inference engine and examine results.

The resource form for city is an instance of MDRM RSSD9130.

The text node has changed to Montpelier (capital of Vermont).

We would also examine the other changed resources.
Export the staging into FFIEC compliant XBRL.

Line 1968 shows the updated city. The XBRL is valid.

We use XMLSpy as an XBRL tool. Topbraid only accepts XML files, so we have to modify the header and save as XML and XBRL.
We export the modified staging OWL into XML.

We invoke the export dialog for the staging OWL file.

The root instance “#r” will export all semantic XML triples.

The JPMC and some other bank XBRL files are on the bankontology.com website in the data directory. You are welcome to test with your filing software.
You can follow the links to the base namespaces. The FFIEC Call Report imports XBRL.
As the upper/core ontology FIBO is on top of the owl imports. Financial Regulation Ontology has entities common for Banks, Funds, Hedge funds and Insurance. So it is the domain to hold the XBRL ontology. FinRegOnt imports FIBO and a Legal Ontology. Bank Ontology holds the reverse engineered FFIEC taxonomy. It imports the XBRL classes from FinRegOnt. To try it out, just create a “My Bank” set of ontologies, importing the bank ontology.
To recap:
We started in the bleak 1960s.
In the bright future everything is a triple.
- Standards and Regulations drive FIBO.
- FIBO makes compliance data management less complex.
Lessons learned:

- FIBO has good support for compliance. We can load XBRL data into FIBO and generate the Call Report out of FIBO.
- XBRL import and transformations are slow.
- FIBO is too assertive. Have more defined classes!

XBRL, Staging FIBO - two simple steps in either direction.

The appendix has links to the ontologies, documentation and tutorials.
References

   Chapter one has an intro to OWL, FIBO, and the Legal reference ontology. There is also a getting started with Protégé section.
2. XBRL Ontology: http://finregont.com/xbrl/
   Includes links to the ontology files and documentation.
4. XBRL consortium, US website: https://xbrl.us/

(left blank)