Triplificating and linking XBRL financial data

Roberto García
Universitat de Lleida
Jaume II, 69
25001 Lleida, Spain
+34 973 702 742
rgarcia@diei.udl.cat

Rosa Gil
Universitat de Lleida
Jaume II, 69
25001 Lleida, Spain
+34 973 702 742
rgil@diei.udl.cat

ABSTRACT

One of the main ways of populating the Web of Data is by triplifying existing data sources. One interesting candidate for this approach is data based on the XML Business Reporting Language (XBRL), a standard for business and financial reporting. Many institutions are making available or requiring data in this format, e.g. the US SEC through the EDGAR program. However, XBRL data is loosely interconnected and it is difficult to mix and query it. Our contribution is a translation from XBRL filings to linked data, which we have applied to more than 1000 filings obtaining 3 million triples. The resulting semantic data is easier to integrate and cross query. Moreover, it can be interconnected with the rest of the Web of Data in order to extract its full potential.

Categories and Subject Descriptors

F.4 Mathematical Logic and Formal Languages
H.3 Information Storage and Retrieval
J.1 Administrative Data Processing

General Terms

Management, Economics, Languages.

Keywords

Business, Semantic Web, Linked Data, accounting, finance, interoperability.

1. INTRODUCTION

The main way to populate the Web of Data is by triplifying existing data sources. The motivation to do so is that usually this data is not offering its full potential because it is isolated, i.e. not connected to other external pieces of data that enrich them. It might even be the case that the data is loosely interconnected internally. Most of the time this is due to the fact that the technological solutions used to publish that data do not make it easy to interconnect it internally and to other external data sources.

Business reporting is a domain where the need for a common data format for reports has already been identified. XBRL (eXtensible Business Reporting Language) is an XML language intended for modelling, exchanging and automatically processing business and financial information. XBRL is being deployed in many different scenarios [1], especially thanks to the support of some regulators and government agencies. For instance, there is the EDGAR[1] program promoted by the U.S. Securities and Exchange Commission (SEC). It performs automated collection, validation, indexing, acceptance and forwarding of submissions by companies and others who are required by law to file forms with the SEC.

It has evolved from a voluntary program and now there is a mandate for a three years phase-in schedule starting 2009 with companies with public float over $5 billion (approximately 500 companies) and ending 2011 with all companies filing to the SEC doing so using XBRL. Moreover, the Government Information Transparency Act will require federal agencies to collect their data in a uniform, searchable format using XBRL thereby simplifying mandatory financial reporting for companies that receive federal funds.

However, despite the great success in the adoption of XBRL, we have observed some limitations in its support for cross analysis of financial information in XBRL tools and applications, as it is detailed in Section 2, that might threaten its usefulness. These limitations are not just among data based on different accounting principles, which are represented in XBRL using taxonomies. It even happens when comparing filings for different companies based on the same taxonomies or filings for the same company based on different versions of the taxonomies. And of course, this limitation is strongest when trying to link it to non-XBRL data.

We argue that this limitation is inherited from the technologies underlying XBRL, especially XML. XML takes a document-oriented approach, where each document presents a tree structure. This makes it difficult for XML-based tools to provide functionalities that blur this separation into documents and that overcome the limitations of a tree structure when mashing-up data from different sources. Moreover, XBRL does not provide formal semantics that might help to integrate different taxonomies using logic reasoners.

In any case, the integration of XBRL data into comparable information is a strong requirement for the analysis of business and financial information at a global scale. This might increase the efficiency and effectiveness of the decision-making processes relying on this kind of information. For instance, bankruptcy prediction and other tasks related to the assessment

I-SEMANTICS 2010, September 1-3, 2010 Graz, Austria

[1] Electronic Data Gathering, Analysis, and Retrieval system, http://www.sec.gov/edgar.shtml

of the solvency of a firm, a business sector or set of interrelated companies. Many have already pointed out this issue and proposed Semantic Web technologies as a natural choice for XBRL data integration, cf. Section 2.

Despite these potential benefits, currently, financial and business data is being produced using XBRL and it seems that more and more XBRL data is going to be available in the future. XBRL is been promoted by regulators and government agencies like the SEC, as it has been shown before, but also other bodies like the European Union or the Spanish Securities Commission [2].

Consequently, our opinion is that the best short term approach in order to get financial and business data to the Semantic Web is not to propose an alternative language based on Semantic Web technologies, but to apply methods to map existing XBRL to semantic metadata. This also seems the best option in the short and midterm to populate the Web of Data with business information.

The rest of this paper is organised as follows. The next subsection introduces the structure of XBRL and Section 2 presents the related work. Then, in Section 3, we present our approach. It is based on a transformation from XML data to RDF using the XBRL to RDF tool, which is described in Section 3.1. Then, the second step is to translate the XML Schemas that structure XBRL data to OWL ontologies using the XML Schema to OWL tool detailed in Section 3.2.

The results of the previous transformations, as detailed in Section 4, are a set of OWL ontologies for the main XBRL taxonomies used in the EDGAR program. Based on these ontologies, it has been possible to translate all the EDGAR instance documents from XML based on these taxonomies to RDF based on the resulting ontologies.

This dataset has been then linked to existing ones in order to integrate it into the Web of Data and has been tested using a linked data publishing and management platform. Finally, Section 5 presents the conclusions and the future work.

1.1 XBRL

XBRL is based on two kinds of documents, instance documents and taxonomies. Instance documents report business facts and point to a set of taxonomies, which define the meaning of these facts, e.g. under what accounting principles they hold, what other facts they are related to or what kind of things do they refer to.

1.1.1 Instances

More concretely, an XBRL instance document contains business Facts. An example of a Fact could be "sales in the last quarter". If the Fact is simple valued, like "the long-term debt is 350,000" whose value is just a number, it is called Item. If the Fact has a more complex value, like "for the *preferred stock*, the *preferred stock par value per share* is 0 and the *preferred stock shares authorized* is 2000", it is called Tuple.

Items are represented in XBRL as a single XML element with the value as its content while Tuples are represented by XML elements containing nested Items or Tuples, i.e. subelements.

However, facts are not isolated entities and it is not enough to provide their values, it is also necessary to contextualize them. Consequently, three more entities are introduced in the XBRL model:

- **Context**: it defines the *entity* (e.g. company or individual) to which the fact applies, the *period* of time the fact is relevant and an optional *scenario*. Scenarios provide further contextual information about the facts. Contexts are referenced from Facts using the "contextRef" attribute.
- **Unit**: it defines a unit of measure, such as "USD" or "shares". They are referenced from Facts using the "unitRef" attribute. Complex units can also be defined, like "USD per share".
- **Reference**: The kinds of facts under consideration are defined by taxonomies, which specify their meaning in the context of some accounting principles or purpose, e.g. Facts relevant for banking and savings institutions.

1.1.2 Taxonomies

Taxonomies are the other kind of XBRL document. A taxonomy defines a hierarchy of concepts, basically kinds of Facts, and captures part of their intended meaning. In XBRL there is a set of base taxonomies that define the core concepts and other ones that extend them in order to particularize these concepts for concrete accounting principles, application domains, etc. Additionally, it is possible to extend existing taxonomies and accommodate them to particular needs.

Taxonomies are based on XML Schemas, which provide the taxonomy building primitives and the extension mechanisms. Moreover, there are also "linkbases", which allow establishing links beyond the tree structure of a taxonomy by virtue of their use of XLink.

2. RELATED WORK

The U.S Securities and Exchange Commission (SEC) offers some online tools that allow interacting with the data available in XBRL form. There is a tool called Interactive Financial Reports[2] that allows viewing and charting companies financial information. It also provides some functionality that allows comparing different filings and different companies, though it is hard to use and prone to even the slightest differences between the compared filing facts, even when there is just a name change for facts from filings of the same company.

There is also the Financial Explorer, which presents company financial data through very informative diagrams. In this case, it is just possible to show data from one company at a time. Finally, there is the Executive Compensation tool, which allows comparing just two facts, Public Market Capitalization and Revenue, across all filed companies.

Apart from the SEC tools, there are some other XBRL tools, most of them proprietary and with quite high licensing cost. Among them, the Fujitsu XBRL Tools[3] should be highlighted because they are one of the most popular tool sets and it is available for XBRL Consortium members and academic users. The tools comprise taxonomy and instance editors, viewers and validators.

The most powerful tool in this set, though still in beta and with many usability problems, is the Instance Dashboard. This

[2] SEC's Interactive Financial Report Viewer, http://www.viewerprototype1.com/viewer

[3] Fujitsu XBRL Tools, http://www.fujitsu.com/global/services/software/interstage/xbrltools/

application can consume multiple instance documents and, by specifying a base taxonomy, users can perform some comparison analysis, though limited to facts in that taxonomy that appear in all the filings.

As it can be noted from the previous analysis, the main limitation of XBRL tools is their limited support for cross analysis of financial information, not just among data based on different taxonomies, even when comparing filings for different companies based on the same taxonomies.

This limitation is inherited from the technologies underlying XBRL, especially from XML. XML takes a document-oriented approach, where each document presents a tree structure. This makes it difficult for XML-based tools to provide functionalities that blur this separation into documents and that overcome the limitations of a tree structure when mashing-up data from different sources.

Consequently, Semantic Web tools are being considered by people like Charles Hoffman, the father of XBRL: "*This field [W3C semantic standards] is rich with possibilities and stands as the next logical step in the natural progression of information technology to seek a higher value proposition*" [3].

This interest is materializing, and the combination of XBRL and the Semantic Web has been receiving some attention in different blogs[4,5], mailing lists and web groups[6]. However, it is difficult to find concrete results that put into practice Semantic Web technologies in the XBRL field.

Moreover, most of these results are specific for some parts of XBRL. For instance, there is an ontology about financial information based on XBRL that is specific for investment funds [[4]] and, though it is generated using a generic XBRL taxonomy to OWL ontology algorithm, there is not an equivalent tool that translates generic XBRL instance data. There is also another tool that translates quarterly and semester accounting information submitted to the Spanish securities commission (CNMV) to Semantic Web technologies [2].

Moreover, both approaches are based on procedural code specially developed in order to extract specific patterns from the XBRL data. Consequently, they are difficult to scale to the whole XBRL specification and sensitive to minimal changes in it. We propose an approach that, instead of directly processing XBRL data, takes profit from the fact that it is expressed using XML and specified using XML Schemas. OpenLink XBRL Sponger is the only tool to our knowledge that transforms generic XBRL instance data to RDF [5]. However, in this case, there is not an associated mapping from the taxonomies instance data is based on to ontology languages.

3. TRIPLIFICATION PROCESS

The proposed approach is based on the transfer of existing XBRL taxonomies and instance data to Semantic Web technologies. This transfer is based on the XML Semantics Reuse methodology [6] and the XML Schema to OWL and XML to RDF tools implemented in the ReDeFer project[7].

This methodology combines an XML Schema to web ontology transformation, called XSD2OWL, with a transparent translation from XML to RDF, XML2RDF. The ontologies generated by XSD2OWL are used during the XML to RDF step in order to generate semantic metadata that takes into account the XML Schema intended meaning.

This approach differs from other attempts to move metadata from the XML domain to the Semantic Web. Some of them just model the XML tree using the RDF primitives [7]. Others concentrate on modelling the knowledge implicit in XML languages definitions, i.e. DTDs or the XML Schemas, using web ontology languages [8,9]. Finally, there are attempts to encode XML semantics integrating RDF into XML documents [10,11].

However, none of them facilitate an extensive transfer of XML metadata to the Semantic Web in a general and transparent way. Their main problem is that the XML Schema implicit semantics are not made explicit when XML metadata instantiating this schemas is translated. This is so because the RDF data produced from XML instance data looses its links to the XML Schemas that structure them and model the relations among different XML entities.

These relations among different XML entities are what carry the XML Schema implicit semantics. They capture part of the meaning intended by the schema developer that, though XML Schema does not provide a way to encode semantics, is recorded in the way XML Schema constructs are used. For instance, by modeling that element "father" is a *subtitutionGroup* for element "parent", it is possible to interpret that "parent" is more general than "father" and that "father" can appear where "parent" appears. More details about the implicit semantics of XML Schema constructs as compared to OWL ones are provided in Section 3.2.

Therefore, the previous transformations from XML to RDF do not take profit from the meaning encoded in XML Schemas and produce RDF metadata almost as semantics-blind as the original XML. Or, on the other hand, they capture this semantics but they use additional ad-hoc semantic constructs that produce less transparent metadata.

3.1 XML2RDF

The XML to RDF transformation follows a structure-mapping approach [7] and tries to represent the XML metadata structure, i.e. a tree, using RDF. The RDF model is based on the graph so it is easy to model a tree using it. Moreover, we do not need to worry about the loss of semantics produced by structure-mapping. We formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF metadata using the instantiation relation *rdf:type*.

The structure-mapping is based on translating XML metadata instances to RDF that instantiates the corresponding constructs in OWL. The more basic translation is from *xsd:elements* and *xsd:attributes* to *rdf:Properties* (*owl:ObjectProperties* for node to node and *owl:DatatypeProperties* for node to value relations).

[4] Raggett, D. XBRL and RDF, 2008. http://people.w3.org/~dsr/blog/?p=8

[5] DuCharme, B. Changing my mind about XBRL again, 2008. http://www.snee.com/bobdc.blog/2008/08/changing_my_mind_about_xbrl_ag.html

[6] XBRL Ontology Group, http://groups.google.com/group/xbrl-ontology-specification-group

[7] ReDeFer project, http://rhizomik.net/redefer

Values are kept during the translation as simple types and RDF blank nodes are introduced in the RDF model in order to serve as the source and destination for properties. They will remain blank for the moment until they are enriched with semantic information.

The resulting RDF graph model contains all that we can obtain from the XML tree. It is already semantically enriched thanks to the *rdf:type* relation that connects each RDF property to the *owl:ObjectProperty* or *owl:DatatypeProperty* it instantiates. It can be enriched further if the blank nodes are related to the *owl:Class* that defines the package of properties and associated restrictions they contain, i.e. the corresponding *xsd:complexType*. This semantic decoration of the graph is formalised using *rdf:type* relations from blank nodes to the corresponding OWL classes.

At this point we have obtained a semantically enabled representation of the input metadata, a representation that makes the meaning intended by the XML and XML Schema modelers explicit from a computer point of view. The instantiation relations can now be used to apply OWL semantics to metadata. Therefore, the semantics derived from further enrichments of the ontologies, e.g. integration links between different ontologies or semantic rules, are automatically propagated to instance metadata thanks to inference.

Focusing on XBRL data, what we get by applying this triplification process of the corresponding XML data is summarised in Fig. 1. This figure shows the XBRL core concepts as they are modeled in the resulting RDF data. The report is modelled as an instance of the class “ReportType” and XBRL facts are modelled as instances of the class “FactType”.

In fact, if a direct modelling of the underlying XML tree was performed, facts should be modelled as RDF Properties because they correspond to XML elements. However, in order to make the resulting RDF data more usable as it is more intuitive to view a fact as class instance than as a relation one, we have introduce a modification in the basic XML2RDF algorithm as it is detailed in the next subsection.

Then, continuing from the “FactType” instance, there are relations to the actual value of the financial fact modelled using rdf:value and two properties stating the decimals and unit used for that value. There is also a property linking the fact to its context, which details the involved entity, the time period and the scenario.

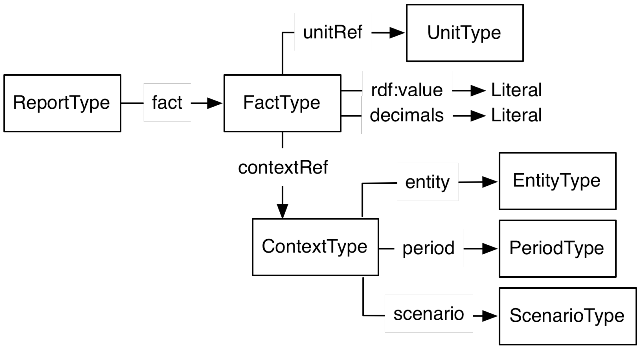

Fig. 1. RDF model for the core XBRL concepts generated using XML2RDF and XSD2OWL (boxes correspond to classes and arrows to properties having them as domain/ranges)

3.1.1 Algorithm

Table 1 shows part of the algorithm that implements the XML to RDF translation. Basically, starting from the root element, it traverses the XML tree and produces triples for all attributes and elements recursively using the “transResProps” method. All the references to the traversed elements and their attributes are mapped to their equivalent in the OWL ontologies corresponding to the original XML Schemas. This is done by the “map” function.

There is an additional method, “transFact”, which is responsible for making all the processed facts class instances instead of property instances, and also introducing the necessary connections to the rest of the generated triples that make the resulting data follow the core model shown in Fig. 1.

Table 1. XBRL to RDF Algorithm.

```
Model XBRL2RDF(Document doc, Mapping map)
{
 Model rdf;
 Resource r = rdf.createResource(doc.url);
 Element e = doc.getDocumentElement();
 Resource p = map(e.nsURI())+e.localName();
 if (p.subClassOf(XBRLI.FactType)
 {
  r = transFact(r, p, rdf);
 }
 Class range = map.getRange(null, Property(p));
 transResProps(r, e, range, rdf, map);
}

Resource transFact(Resource r, Resource p,
          Model rdf)
{
 Resource f = rdf.createResource();
 f.addProperty(RDF.type, Class(p));
 r.addProperty(XBRLI.item, f);
 return f;
}

transResProps(Resource r, Element e,
         Class domain, Model rdf, Mapping map)
{
 foreach (a in e.attributes())
 {
  Property p = map(a.nsURI())+a.localName();
  r.addProperty(p, a.getValue());
 }
 foreach (c in e.childNodes())
 {
  if (c.isTextNode())
  {
   Property p = map(c.nsURI()+c.localName());
   r.addProperty(p, c.getValue());
  }
  else
  {
   Resource rC = rdf.createResource();
   Property p = map(c.nsURI()+c.localName());
   r.addProperty(p, rC);
   Class range = map.getRange(domain, p);
   rC.addProperty(RDF.type, range);
   transResProps(rC, c, range, rdf, map);
  }
 }
}
```

3.2 XSD2OWL

The XML Schema to OWL transformation is responsible for capturing the schema implicit semantics, which is determined by the combination of XML Schema constructs. The transformation is based on translating these constructs to the OWL ones that best capture their intended meaning. These translations are detailed in Table 2.

The XML Schema to OWL transformation is quite transparent and captures a great part XML Schema semantics. The same names used for XML constructs are used for OWL ones, although in the new namespace defined for the ontology. XSD and OWL constructs names are identical; this usually produces uppercase-named OWL properties because the corresponding element name is uppercase, although this is not the usual convention in OWL.

Table 2. XBRL Schema to OWL translations for the XML Schema constructs

XML Schema	OWL	Mapping motivation
element[@substitutionGroup= "xbrli:item"]	owl:Class	Facts, though elements, are mapped to classes
element | attribute	rdf:Property owl:DatatypeProperty owl:ObjectProperty	Named relation between nodes or nodes and values
element@substitutionGroup ="xbrli:item"	rdfs:subClassOf	The corresponding element is mapped to a owl:Class rdfs:subClassOf xbrli:item
element@substitutionGroup	rdfs:subPropertyOf	Relation can appear in place of a more general one
element@type	rdfs:range	The relation range kind
complexType|group |attributeGroup	owl:Class	Relations and contextual restrictions package
complexType//element	owl:Restriction	Contextualised restriction of a relation
extension@base | restriction@base	rdfs:subClassOf	Package concretises the base package
@maxOccurs @minOccurs	owl:maxCardinality owl:minCardinality	Restrict the number of occurrences of a relation
sequence choice	owl:intersectionOf owl:unionOf	Combination of relations in a context

4. LINKED DATA

As a result of the previous triplification process, we have generated an ontological infrastructure for the XBRL core, currently XBRL 2.1. It is composed of the ontologies resulting from mapping the XBRL core XML Schemas using the XSD2OWL transformation: XBRL Instance, XBRL Linkbase, XBRL XL and XBRL XLink. These ontologies have been adapted to accommodate the changes introduced by XBRL to RDF that make the output semantic data more usable, basically by making facts classes and no longer properties.

Apart from the previous schemas, EDGAR XBRL data is also based on the schemas shown in Table 3, which have been also translated. These schemas are part of the EDGAR Standard Taxonomies. The US Financial Reporting - February 28, 2005 taxonomies have been considered as they are used by the input data currently submitted to this program. These include the US GAAP (Generally Accepted Accounting Principles) and also some non-GAAP schemas.

Table 3. US GAAP and Non-GAAP taxonomies transformed

- **US GAAP** (Generally Accepted Accounting Principles):
  - ○ Primary Terms Elements (USFR-PTE)
  - ○ Primary Terms Relationships (USFR-PTR)
  - ○ Financial Services Terms Elements (USFR-FSTE)
  - ○ Financial Services Terms Relationships (USFR-FSTR)
  - ○ Investment Management Terms Relationships (USFR-IME)
  - ○ Industry:
    - ■ Banking and Savings Institutions (US-GAAP-BASI)
    - ■ Commercial and Industrial (US-GAAP-CI)
    - ■ Insurance (US-GAAP-INS)
    - ■ Investment Management (US-GAAP-IM)
- **Non-GAAP**:
  - ○ Accountants Report (USFR-AR)
  - ○ Management Discussion and Analysis (USFR-MDA)
  - ○ Management Report (USFR-MR)
  - ○ SEC Certifications (USFR-SECCERT)

Each filing for the companies participating in the EDGAR program contains an XBRL XML file representing the actual financial data and also a specific XML Schema extending the XBRL core. This schema provides specific guides for the corresponding financial data. Both files are translated using XML2RDF and XSD2OWL respectively. Table 4 shows and example of a XBRL fragment and the below the RDF one obtained from the XML2RDF transformation, which is also enriched with rdf:type links to the OWL ontologies obtained from transforming the involved XBRL XML Schemas.

For instance, for Adobe Systems Inc filing on 2008-07-03, there are the adbe-20080616.xml file containing the instance data and the adbe-20080530.xsd schema for data structures specific for this filing. They are mapped, respectively, to the RDF file for instance data adbe-20080616.rdf and the OWL ontology adbe-20080530.owl for the schema.

All the previous ontologies are available from the BizOntos Business Ontologies web page[8] and the semantic data for all the processed filings can be queried and browsed from the Semantic XBRL site[9]. Currently, more than 1 thousand filings have been processed from EDGAR. The combination of all these filings once mapped to RDF amounts almost 3 million triples. At this step, it is possible to take advantage of semantic web technologies in order to improve the interconnectedness of the dataset by means of semantics-enabled data integration.

[8] BizOntos, http://rhizomik.net/ontologies/bizontos

[9] SemanticXBRL, http://rhizomik.net/semanticxbrl

Table 4. XBRL fragment (first row) and the corresponding RDF one (second row)

XBRL XML Fragment

```
<context id="AsOf20061201_Consolidated_Unaudited">
  <entity>
    <identifier scheme="http://www.sec.gov/CIK">796343</identifier>
    <segment><adbe:Consolidated /></segment>
  </entity>
  <period>
    <instant>2006-12-01</instant>
  </period>
  <scenario><adbe:Unaudited /></scenario>
</context>
…
<usfr-pte:CashCashEquivalents decimals="-3"
contextRef="AsOf20061201_Consolidated_Unaudited"
unitRef="USD">772500000</usfr-pte:CashCashEquivalents>
```

XBRL RDF Fragment

```
<xbrli:context>
  <xbrli:contextType
rdf:about="AsOf20061201_Consolidated_Unaudited">
    <xbrli:entity>
      <xbrli:contextEntityType rdf:about="&semxbrl;CIK/796343">
        <xbrli:segment>
          <xbrli:segmentType>
            <adbe20080530:Consolidated rdf:parseType="Resource"/>
          </xbrli:segmentType>
        </xbrli:segment>
      </xbrli:contextEntityType>
    </xbrli:entity>
    <xbrli:period>
      <xbrli:contextPeriodType>
        <xbrli:instant>2006-12-01</xbrli:instant>
      </xbrli:contextPeriodType>
    </xbrli:period>
    <xbrli:scenario>
      <xbrli:contextScenarioType>
        <adbe20080530:Unaudited rdf:parseType="Resource"/>
      </xbrli:contextScenarioType>
    </xbrli:scenario>
  </xbrli:contextType>
<xbrli:context>
…
<xbrli:item>
  <usfr-pte:CashCashEquivalents>
    <rdf:type rdf:resource="&xbrli;monetaryItemType"/>
    <xbrli:unitRef rdf:resource="http://dbpedia.org/resource/USD"/>
    <xbrli:decimals>-3</xbrli:decimals>
    <xbrli:contextRef
rdf:resource="#AsOf20061201_Consolidated_Unaudited"/>
    <rdf:value>772500000</rdf:value>
  </usfr-pte:CashCashEquivalents>
</xbrli:item>
…
```

4.1 Linking to other Data Sets

In order to connect the XBRL RDF dataset with other ones in the Web of Linked Data, the entities in the XBRL model have been analyzed in order to detect those also described in other datasets. The more prominent ones are companies, a kind of EntityType present in most EDGAR filings. XBRL data provides an identifier for these entities, the Central Index Key (CIK) number. It is a number given to an individual or company by the U.S. SEC and used to identify the filings of a company, person, or entity in several online databases, including EDGAR. However, there are some EDGAR filings that do not use this identifier and use the "CompanyName" one instead. For most of them it is possible to get the corresponding CIK using EDGAR's CIK Lookup service[10]. Unfortunately, as the filings are directly submitted by the participant companies, there are some discrepancies between the names in the filings and those in the lookup service.

Even when a CIK identifier is available in the EDGAR dataset, it might be impossible to directly connect it to company descriptions available in DBPedia because just 23 of them have the "secCik" property that links them to the company CIK. Actually, we have been able to map just 5 companies to DBPedia using the DBPedia secCik property as just some of them are currently using XBRL filings. Consequently, we have explored some alternative ways to connect companies to DBPedia. We have conducted this exploration with the help of the Silk framework [12], a tool for discovering relationships between data items within different Linked Data sources.

Using the Silk - Link Specification Language (Silk-LSL), we have specified which links should be generated between data sources as well as which conditions data items must fulfill in order to be interlinked. These link conditions combine various similarity metrics and can take the graph around a data item into account, which is addressed using an RDF path language.

The simplest case is to define the link specification using the CIK property. In this case, it is just specified to look for pairs of resources, one from the semantic XBRL dataset and the other from DBPedia one, that have the same value for the dbpprop:secCik property. Note that we have used this property during the triplification process in order to model the ID in the input XBRL. As mentioned before, from this link specification we are able to get just 5 owl:sameAs links between both datasets.

The next possibility we have explored is to link resources with almost identical company names. We have used a combination of the Jaro and Q-Gram similarity measures implemented by Silk. We have been forced to use a quite high threshold for accepted links because the presence of quite common words in company names, like "Inc.", "Corp.", "Co.", "Ltd.", etc., and their many variants makes it very difficult to get reliable links based on the company name.

After a review of the links generated using the previous approach we have been able to generate 27 new owl:sameAs relations between the datasets. This is also a quite scarce amount given that we currently have 543 companies in our dataset. Our last attempt to date to generate links to DBPedia is to take profit from the fact that for 398 companies in our dataset we have the ticker.

The obvious approach is to use the dbpprop:ticker property to generate links to the corresponding DBPedia resources. However, just 4 of them have this property. Fortunately, we have observed that many DBPedia companies have alternative URI based on their ticker. In this case, the approach to specify the links has been to explore the dbpprop:redirect links pointing to DBPedia public companies and strip the URI in order to get potential tickers. Eg., dbpedia:Microsoft is dbpprop:redirect of

[1010] Search EDGAR: CIK Lookup, http://sec.gov/edgar/searchedgar/cik.htm dbpedia:MSFT. Using this approach we have been able to generate 64 owl:sameAs links to DBPedia.

This continues to be a quite limited amount so we continue to explore other ways to generate links to dbpedia. Meanwhile, we have also explored other datasets we can link to. A really interesting candidate is the "U.S. Securities and Exchange Commission Corporate Ownership RDF Data"[11] generated by Joshua Tauberer from SEC and CorpWatch[12] data.

This is a very interesting dataset because it provides information about who is in the board of many of these companies and also the subsidiary relation among companies. We can use this data in order to generate complex queries that aggregate the financial data we are triplifying from SEC taking into account groups of companies that hold different kinds of ownership relations, e.g. are all subsidiaries of the same company or share board members.

In this case it has been easy to generate the links to this dataset because all companies are identified using their CIK. Not all of them are providing XBRL filings so from a total amount of 543 companies in our dataset and 12589 companies in the ownership dataset, we have obtained 398 links.

Finally, the other kind of entities that might be connected to external datasets is units. The easiest kind of entities is currencies because most of the filings use the ISO 4217 code in order to identify them. The rest of the units are specific to the filings, for instance there is the "shares" or "pure" units that do not have equivalents in other datasets. Consequently, we are just linking currencies to their descriptions in DBPedia.

4.2 Use Case

As a result of how the original XML tree is semantically enriched when it is mapped to RDF and how different XML trees are interconnected when mapped to RDF graphs, it is possible to query and traverse the mix of many XBRL filings in novel and more productive ways.

All this functionality has been put into practice for the semantic dataset resulting from mapping the EDGAR XBRL filings to RDF. The 3 million triples resulting from the mapping have been published online using the Rhizomer tool [13]. Data can be queried, traversed and edited online[13] through a web user interface for human users. Moreover, through HTTP and content negotiation, Rhizomer also makes data available for machine consumption. The overall architecture of the resulting system is shown in Fig. 2.

For human users, this tool makes it possible to interact with Semantic Web data by posing semantic queries through dynamic forms or by browsing the RDF graph interactively. The entry page provides some sample queries that return an HTML rendering of the selected parts of the graph, which can be then used as the starting point for the browsing steps.

These sample queries illustrate how semantic queries take profit from the hierarchical relations in the original XML Schemas, i.e. hierarchies of elements and complex types that are translated to property and class hierarchies respectively. Moreover, there is also a query that exploits the fact that some of the Adobe Systems Inc. ontologies have been integrated and returns data from different filings for equivalent facts with different names.

Finally, there are additional views dynamically plugged in depending on the kind of resource being browsed. Many of them are the same available from Exhibit [14] (timeline, map, facets,…). In addition to visualization plugins, it is also possible to integrate other kinds of services that manipulate data. The whole system is built on top of a OpenLink Virtuoso[14] repository that provides scalability to more than tens of millions of triples and provides RDF Schema inferencing and support for OWL equivalence constructs.

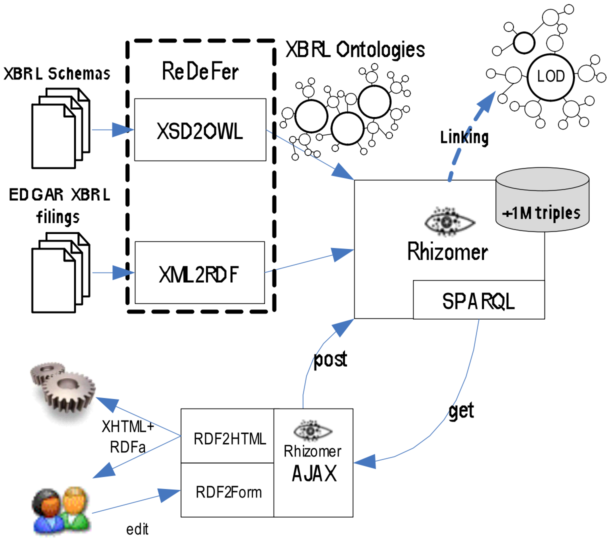

Fig. 2. Architecture of the proposed solution for semantic XBRL generation, linking and publishing

5. CONCLUSIONS AND FUTURE WORK

As it has been shown in this paper, and previously by others as detailed in the related work section, it is possible to triplify XBRL data. The contribution of our approach is that the RDF semantic data keeps all the original information and structure and that it also includes the involved XML Schemas that structure the XBRL data. These schemas are mapped to Web ontologies, which make all the semantics implicit in the original XML Schemas explicit and available when semantically querying RDF data.

Moreover, it is also possible to take profit from Web ontology primitives in order to semantically integrate different filings following different XML Schemas, i.e. XBRL taxonomies. Once mapped to ontology concepts and relations, the XBRL contexts, facts and other resources defined for different filings can be related as more specific, more general or equivalent. It is also possible to link them to other datasets like DBPedia.

[11] http://www.rdfabout.com/demo/sec/

[12] http://api.corpwatch.org/

[13] SemanticXBRL, http://rhizomik.net/semanticxbrl

[14] OpenLink Virtuoso open-source edition, http://virtuoso.openlinksw.com/wiki/main/

This approach has been put into practice in the context of the SEC's EDGAR program that promotes XBRL filings for USA companies. It has been possible to apply the previous XML to RDF and XML Schema to Web ontology mappings to all the EDGAR filings and more than 3 million triples have been obtained.

We have also have made all this semantic information generated from the EDGAR program available online, so it can be queried and browsed using a Web user interface. The proposed semantic queries illustrate the benefits of the semantic integration available once XBRL data is translated to semantic data.

We have tried to keep the semantic XBRL data as tied as possible to the original XML data because we do not see our proposal as an alternative to XBRL. Semantic Web technologies have some limitations that currently do not make them a clear alternative to XBRL.

For instance, OWL does not provide the primitives to easily model features available in XBRL like the calculation facilities provided by calculation linkbases. Moreover, the characteristics of the logic formalisms underlying OWL might not be the more intuitive choice in some XBRL use scenarios. For instance, a great part of OWL relies on the Open World Assumption and it is based on restrictions instead of on constraints [4].

On the contrary, we see XBRL and the Semantic Web as clearly complementary. XBRL can be used for business and financial data representation and validation, while its translation to Semantic Web technologies can be the way to make all this data publicly available enabling cross analysis of this data thanks to semantic integration and a graph base model.

This vision must be more deeply tested and validated. In order to do that, we are currently working on integrating ontology alignment tools into the mapping process. This way it is going to be possible to extensively put semantic integration into practice. For instance, through semantic queries that relate data coming from different filings, accounting principles and even other datasets, like the ownership relations among companies or news streams semantically annotated using services like OpenCalais[15].

Another future plan is to exploit XBRL semantic data beyond querying and browsing. In this respect, our idea is to take profit from the Rhizomer human-Semantic Web interaction platform in order to implement additional ways to interact with this data. For instance, we are currently evaluating an interactive mechanism for plotting numeric values as graphs reusing Freebase Parallax [15]. This would allow performing semantic queries for specific facts across different filings and then plotting their values.

6. ACKNOWLEDGMENTS

The work described in this paper has been partially supported by Spanish Ministry of Science and Innovation through the Open Platform for Multichannel Content Distribution Management (OMediaDis) research project (TIN2008-06228).

7. REFERENCES

[1] Bonson, E., Cortijo, V., Escobar, T. 2009. Towards the global adoption of XBRL using International Financial Reporting Standards (IFRS). International Journal of Accounting Information Systems, 10(1), pp. 46-60, 2009

[2] Núñez, S., de Andrés, J., Gayo, J. E., Ordoñez, P. A. 2008. Semantic Based Collaborative System for the Interoperability of XBRL Accounting Information. Technologies and Information Systems for the Knowledge Society. LNCS Vol. 5288, Springer, Berlin, pp. 593-599.

[3] Hoffman, C. 2006. Financial Reporting Using XBRL: IFRS and US GAAP Edition. Lulu.com.

[4] Lara, R., Cantador, I., Castells, P. 2008. Semantic Web Technologies for The Financial Domain. The Semantic Web: Real-World Applications from Industry. Springer, Berlin, pp. 41-74.

[5] Erling, O., Mikhailov, I. 2009. RDF Support in the Virtuoso DBMS. Networked Knowledge - Networked Media, Springer, 2009, pp. 7-24.

[6] García, R. 2006. XML Semantics Reuse. Chapter 7 in A Semantic Web Approach to Digital Rights Management, PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain. http://rhizomik.net/~roberto/thesis

[7] Klein, M.C.A. 2002. Interpreting XML Documents via an RDF Schema Ontology.: Proceedings of the 13th Int. Workshop on Database and Expert Systems Applications, DEXA'02, IEEE Computer Society, pp. 889-894.

[8] Amann, B., Beeri, C., Fundulaki, I., Scholl, M. 2002. Ontology-Based Integration of XML Web Resources. Proceedings of the 1st International Semantic Web Conference, ISWC'02. LNCS, Vol. 2342, Springer, Berlin, pp. 117-131.

[9] Cruz, I., Xiao, H., Hsu, F. 2004. An Ontology-based Framework for XML Semantic Integration. Proceedings of the 8th Int. Database Engineering and Applications Symposium, IEEE Computer Society, pp. 217- 226.

[10] Lakshmanan, L., Sadri, F. 2003. Interoperability on XML Data. Proceedings of the 2nd International Semantic Web Conference, ISWC'03, LNCS Vol. 2870, Springer, Berlin, pp. 146-163.

[11] Patel-Schneider, P.F., Simeon, J. 2002. The Yin/Yang web: XML syntax and RDF semantics. Proceedings of the 11th World Wide Web Conference. ACM Press, pp. 443-453.

[12] Volz, J., Bizer, C., Gaedke, M., Kobilarov, G. 2009. Silk – A Link Discovery Framework for the Web of Data. 2nd Workshop about Linked Data on the Web (LDOW2009), Madrid, Spain.

[13] García, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M. 2008. A Platform for Object-Action Semantic Web Interaction. Proceedings of the 16th Int. Conf. on Knowledge Engineering and Knowledge Management Patterns, EKAW'08. LNCS Vol. 5268, Springer, Berlin, pp. 404-418.

[14] Huynh, D. 2007. User Interfaces Supporting Casual Data-Centric Interactions on the Web. Doctoral Thesis at MIT EECS / CSAIL. http://davidhuynh.net/media/thesis/thesis.php

[15] Huynh, D., Karger, D. 2009. Parallax and Companion: Set-based Browsing for the Data Web. http://davidhuynh.net/media/papers/2009/www2009-parallax.pdf

[15] http://www.opencalais.com