Linking XBRL Financial Data

Roberto García, Rosa Gil

Abstract. One of the main ways of populating the Web of Data is by translating existing data sources. One interesting candidate for this approach is data based on the eXtensible 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 U.S. Securities and Exchange Commission (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 Semantic Web technologies, which we have applied to more than 1000 filings obtaining more than 2 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.

1. Introduction

The main way to populate the Web of Data is by translating 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, 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

R. García
Universitat de Lleida. Jaume II, 69. 25001 Lleida, Spain
e-mail: rgarcia@diei.udl.cat

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

(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, we have observed limited support for cross analysis of financial information in XBRL tools and applications, as it is detailed in Section 1.2. This is 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.

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 data contained in XBRL into comparable information is a strong requirement for the analysis of business and financial information at the global level. 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 of the solvency of a firm, a business sector or set of interrelated companies.

Many have already pointed to this issue and propose Semantic Web technologies as a natural choice for data integration [1] and, in this concrete case, for XBRL data integration, cf. the related work in Section 1.2 or the W3C Workshop on Improving Access to Financial Data on the Web[2]. However, this is not enough, the Semantic Web provides the technologies for data integration but some principles are required that facilitate Web-wide deployment of highly interlinked XBRL data. Linked Data [2] provides these principles to publish data in the World Wide Web in a way that helps making it easily discoverable through the links that connect it to other pieces of data.

Despite these 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 [3].

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 and alternative language based on Semantic Web technologies, but to apply methods to map existing

[2] Program of the W3C Workshop on Improving Access to Financial Data on the Web, http://www.w3.org/2009/03/xbrl/program.html

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 chapter is organised as follows. The next subsections introduce the structure of XBRL, then the related work is presented followed by the description of the our contribution in Section 2. It is based o the XML Semantics Reuse Methodology. The first step is to map the XML Schemas that structure XML data to OWL ontologies using the XSD2OWL mapping. Then, the second step is to map XML data to RDF using the XML2RDF mapping.

Once our approach has been presented, the results of the previous mappings are shown in Section 3. They 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 map all the EDGAR instance documents from XML based on these taxonomies to RDF based on the resulting ontologies.

From these ontologies and semantic data, it has been possible to establish some mechanisms, facilitated by Semantic Web technologies, that enrich the dataset with additional links. First, some links to external datasets of the Web of Linked Data. Second, internal links that integrate the different filings by aligning the ontologies they use.

We are currently starting to evaluate this semantic dataset, as it is detailed in Section 4. It is compared to similar undergoing initiatives and it has been made publicly available for querying and browsing through a Web user interface. Finally, in Section 5, the conclusions and future work are presented. The main conclusion is that though RDF data through semantic queries and integration primitives offers a new range of possibilities; it already lacks enough expressive power to substitute XBRL, as it is explained in the conclusion.

We think that the best approach, for the moment, is to combine both approaches and transform XBRL data to semantic form in order to facilitate cross-querying and semantic integration, while keeping the original data in order to benefit from specific XBRL services. Consequently, we concentrate now our future work in completing the mapping from XBRL to Semantic Web, to provide integration facilities at the taxonomy level and to enrich the links of the resulting semantic dataset to other ones in the Web of Linked Data.

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 related to or what kind of things do they refer to.

1.1.1. Instances

More concretely, a 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, four 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. The period of time can have zero length for instance and its value is based on ISO 8601 for date and time values. Scenarios provide further contextual information about the facts, such as whether the business values reported are actual, projected or budgeted. Contexts are referenced from Facts using the “contextRef” attribute, which specifies that the given Fact is valid for the entity, period and scenario defined in the Context.
- **Unit**: it defines a unit of measure, such as “USD” or “shares”. They are referenced from Facts using the “unitRef” attribute, which specifies that the numeric or fractional value of the Fact is based on that unit of measure. Complex units can also be defined, like “USD per share”. Currency units are based on ISO 4217.
- **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. These kinds of facts are then used in instance documents in order to specify actual values for them. However, they are linked to their definition in the taxonomies, typically through schema references, in order to be able to retrieve their meaning.
- **Footnote**: it contains some additional support content and it is associated to Fact using XLink[3].

Table 1 shows part of an instance document from the EDGAR program that contains a Context element which defines a company, a time period and the scenario “unaudited”. Then, there is a Fact that holds in that context. The Fact references the Context and the value unit, while their content is the actual numeric value for that fact.

1.1.2. Taxonomies

The other kind of XBRL document represents taxonomies. 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.

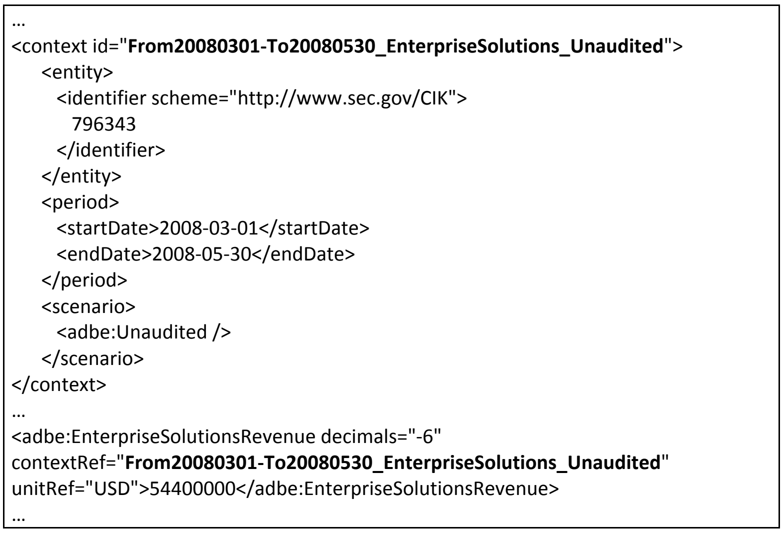
Taxonomies are based on XML Schemas, which provide the taxonomy building primitives and the extension mechanisms. Moreover, there are also “linkbases”, which

[3] XLink, http://www.w3.org/TR/xlink/

allow establishing links beyond the tree structure of a taxonomy by virtue of their use of XLink:

- **Schemas** define concepts that are instantiated as Items or Tuples, depending on their complexity, in the instance documents. They are based on XML Schema elements (xsd:element). A concept definition provides the fact name, whether it is a tuple or an item and its value data type (such as monetary, numeric, fractional or textual).
- **Linkbases** define links from concepts in a taxonomy to labels, pieces of content or to other concepts. The XBRL 2.1 specification defines five different kinds of linkbases.
  - **Label Linkbase**: set of links that provides human readable strings for concepts, potentially in multiple languages.
  - **Reference Linkbase**: these links associate concepts with citations of some body of authoritative literature.
  - **Calculation Linkbase**: these are links that associate a set of values of concepts in taxonomies with a mathematical calculation that must be checked for consistency, for instance that a set of concepts with percentage values sum up 100%.
  - **Definition Linkbase**: it provides semantic relations between concepts like is-a, whole-part, etc.
  - **Presentation Linkbase**: This linkbase associates concepts with other concepts so that the resulting relations can guide the creation of a user interface, rendering, or visualisation.

Table 1. Context and facts examples from an EDGAR filing



```
…
<context id="From20080301-To20080530_EnterpriseSolutions_Unaudited">
    <entity>
      <identifier scheme="http://www.sec.gov/CIK">
        796343
      </identifier>
    </entity>
    <period>
      <startDate>2008-03-01</startDate>
      <endDate>2008-05-30</endDate>
    </period>
    <scenario>
      <adbe:Unaudited />
    </scenario>
</context>
…
<adbe:EnterpriseSolutionsRevenue decimals="-6"
contextRef="From20080301-To20080530_EnterpriseSolutions_Unaudited"
unitRef="USD">54400000</adbe:EnterpriseSolutionsRevenue>
…
```

1.2.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 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[4], 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[5] 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 application can consume multiple instance documents and, by specifying a base taxonomy, users can perform some comparison analysis, though limited to facts in the taxonomy that appears 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*" [4].

This interest is materializing, and the combination of XBRL and the Semantic Web has been receiving some attention in different blogs[6,7], mailing lists and web groups[8].

[4] SEC's Financial Explorer, http://209.234.225.154/viewer/home/

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

[6] DuCharme, B. Changing my mind about XBRL again, in: Bob DuCharme's weblog, bobdc.blog, 2008.
http://www.snee.com/bobdc.blog/2008/08/changing_my_mind_about_xbrl_ag.html

[7] Raggett, D. XBRL and RDF, in: Dave Raggett's Blog, 2008.
http://people.w3.org/~dsr/blog/?p=8

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 [5] and, though it is generated using a generic XBRL taxonomy to OWL ontology algorithm, there is not and equivalent tool that maps generic XBRL instance data. There is also another tool that maps quarterly and semester accounting information submitted to the Spanish securities commission (CNMV) to Semantic Web technologies [3].

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 sensible 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 maps generic XBRL instance data to RDF [6]. However, in this case, there is not and associated mapping from the taxonomies instance data is based on to ontology languages.

2. Approach

There are many 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 mapped. 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 2.1.

Therefore, the previous mappings from XML to RDF do not take profit from the meaning encoded in XML Schemas and produce RDF metadata almost as semantics-

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

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.

Therefore, we have chosen the XML Semantics Reuse methodology [12] and the XML Schema to OWL and XML to RDF tools implemented in the ReDeFer project[9]. This methodology combines an XML Schema to web ontology mapping, called XSD2OWL, with a transparent mapping 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 has already shown its usefulness with other quite big XML Schemas in the Digital Rights Management domain, such as MPEG-21 and ODRL [13], and also in the E-Business [14] and multimedia metadata domains [15], where it produced the more complete MPEG-7 ontology to date [16].

2.1.XSD2OWL Mapping

The XML Schema to OWL mapping is responsible for capturing the schema implicit semantics, which is determined by the combination of XML Schema constructs. The mapping is based on translating these constructs to the OWL ones that best capture their intended meaning. These translations are detailed in Table 2 and Table 3 shows an example mapping.

The XSD2OWL mapping 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. XSD2OWL translations for the XML Schema constructs

XML Schema	OWL	Mapping motivation
element | attribute	rdf:Property owl:DatatypeProperty owl:ObjectProperty	Named relation between nodes or nodes and values
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

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

Therefore, XSD2OWL produces OWL ontologies that make explicit the semantics of the corresponding XML Schemas. Table 3 shows a piece of an XML Schema and the OWL that is generated following this approach.

The only caveats are the implicit order conveyed by *xsd:sequence* and the exclusivity of *xsd:choice*. For the first problem, *owl:intersectionOf* does not retain its operands order, there is no clear solution that retains the great level of transparency that has been achieved. The use of RDF Lists might impose order but introduces ad-hoc constructs not present in the original metadata.

Table 3. XML Schema to OWL mapping example (namespaces omitted for readability)

XML Schema	**OWL (Abstract Syntax)**
`<complexType`	
`name="`**`contextOrganisationType`**`">`	
`<complexContent>`	
`<extension base=`	`Class (`**`contextOrganisationType`**
`"`**`contextEntityType`**`">`	`complete`
`<sequence>`	**`contextEntityType`**
`<element name="`**`Country`**`" type=`	`restriction(`**`Country`**
`"`**`CountryType`**`"/>`	`allValuesFrom(`**`CountryType`**`)`
`</sequence>`	`cardinality(1)))`
`</extension>`	
`</complexContent>`	
`</complexType>`	

Moreover, as it has been demonstrated in the Semantic Web community, the element ordering does not contribute much from a semantic and knowledge representation point of view [17] in most cases and when it is a requirement it is more convenient to explicitly represent it using some sort of order attribute or property. For the second problem, *owl:unionOf* is an inclusive union, the solution is to use the disjointness OWL construct, *owl:disjointWith*, between all union operands in order to make it exclusive.

2.2.XML2RDF Mapping

Once all the metadata XML Schemas are available as mapped OWL ontologies, it is time to map the XML metadata that instantiates them. The intention is to produce RDF metadata as transparently as possible. Therefore, a structure-mapping approach has been selected [7] instead of a model-mapping one [18].

XML model-mapping is based on representing the XML information set using semantic tools. This approach is better when XML metadata is semantically exploited for specific purposes. However, when the objective is to obtain semantic metadata from different kinds of input XML data, it is better to follow a more transparent approach.

Transparency is achieved in structure-mapping models because they only try 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 semantics loose produced by structure-mapping. We have formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF metadata using the instantiation relation *rdf:type* later.

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

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.

2.3.Algorithm

Table 4 shows part of the algorithm that implements the XML to RDF mapping. Basically, starting from the root element, it traverses the XML tree and produces triples for all attributes and elements recursively using the “mapResProps” 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.

3. Results

First of all, we have generated an ontological infrastructure for the XBRL core, currently XBRL 2.1. It is composed by the ontologies resulting from mapping the XBRL core XML Schemas using the XSD2OWL mapping: XBRL Instance, XBRL Linkbase, XBRL XL and XBRL XLink.

Apart from the previous generic schemas, the schemas in Table 5 have been also mapped in order to be able to map the XBRL data submitted to the SEC’s EDGAR program. 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.

Table 4. XML2RDF Algorithm

```
Model XML2RDF(Document d)
{
 Model rdf;
 Resource r = rdf.createResource(doc.url);
 Element e = doc.getDocumentElement();
 Property p = map(e.nsURI())+e.localName();
 Class range = map.getPropertyRange(null, p);
 r.addProperty(RDF.type, range);
 mapResProps(r, e, range, rdf);
}

mapResProps(Resource r, Element e, Class domain, Model rdf)
{
 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.getPropertyRange(domain, p);
   rC.addProperty(RDF.type, range);
   mapResProps(rC, c, range, rdf);
  }
 }
}
```

From US GAAP (Generally Accepted Accounting Principles) the schemas, and corresponding ontologies, are: Primary Terms Elements (USFR-PTE),Primary Terms Relationships (USFR-PTR), Financial Services Terms Elements (USFR-FSTE), Financial Services Terms Relationships (USFR-FSTR) and Investment Management Terms Relationships (USFR-IME). For specific industries: Banking and Savings Institutions (US-GAAP-BASI), Commercial and Industrial (US-GAAP-CI), Insurance (US-GAAP-INS) and Investment Management (US-GAAP-IM).

There are also some non-GAAP schemas that have been also mapped to OWL ontologies: Accountants Report (USFR-AR), Management Discussion and Analysis (USFR-MDA), Management Report (USFR-MR) and SEC Certifications (USFR-SECCERT).

Linking XBRL Financial Data

Table 5. US GAAP and Non-GAAP taxonomies mapped to OWL

- **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 and 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 mapped using XML2RDF and XSD2OWL respectively.

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[10] and the semantic data for all the processed filings can be queried and browsed from the Semantic XBRL site[11]. Currently, 489 filings have been processed from EDGAR. The combination of all these filings once mapped to RDF amounts slightly more than 1 million triples, concretely 1,023,929 triples. A triple is the minimal component of an RDF graph and corresponds to one of its edges connecting two of its nodes.

Table 7 in the Evaluation section shows the RDF metadata resulting from applying the XML2RDF mapping to the XBRL context and fact shown in Table 1. The RDF metadata references classes and properties from the OWL ontologies resulting from mapping the XML Schemas used in the XML instance. This includes the XBRL schemas and also those specific for the concrete filing being processes.

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

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

For a more general view of the resulting semantic dataset, Fig. 1 shows a diagram of the resulting RDF model. At this step, it is possible to take profit from semantic web technologies in order to facilitate connecting the resulting data to other datasets, but also to improve the interconnectedness of the dataset. Both processes are detailed in the next subsections.

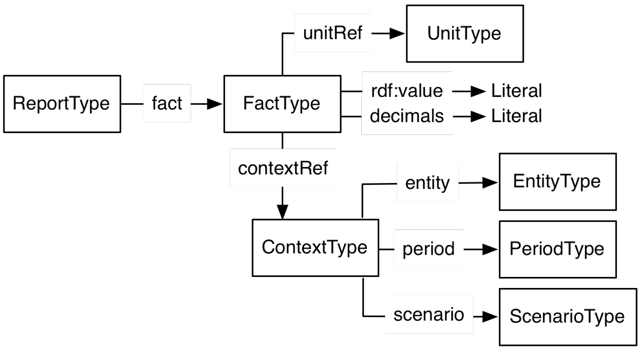


Fig. 1. RDF Model of the semantic XBRL dataset (squares: classes, arrows: properties)

3.1. Links to External Data

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[12]. 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 conduced this exploration with the help of the Silk framework [19], 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

[12] Search EDGAR: CIK Lookup, http://sec.gov/edgar/searchedgar/cik.htm

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 observer 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 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"[13] generated by Joshua Tauberer from SEC and CorpWatch[14] 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

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

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

from a total amount of 543 companies in our dataset and 12589 companies in the ownership dataset, we have obtained 398 links. Table 6 shows a summary of the number of links to external datasets and the method employed to generate them.

Table 6. Summary of the number of links to external datasets

Linking Method	# links to DBPedia	# of links to Corporate Ownership Data
SEC's identifier CIK	5	398
Company name	27	
Company ticker	64	

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.

3.2. Semantic Integration

Apart from the links to other datasets, the EDGAR dataset resulting from the transformation to Semantic Web technologies can be also enriched with internal links. As it has been mentioned earlier, each XBRL EDGAR filing consists of an XML instance file accompanied by a XML Schema taxonomy. This taxonomy is specific for the filing, it changes from filing to filing. The taxonomy defines a set of facts specific for the filing. New facts are introduced, other used in previous filings are removed and some of them suffer minimal modifications.

For instance, the 2008-07-03 filing from Adobe Systems Inc. refers to the fact "InvestmentLeaseReceivable" defined in the adbe-20080530.xsd taxonomy while the 2008-09-16 filing refers to "Investment*In*LeaseReceivable" defined in the adbe-20080829.xsd taxonomy. Apart from these slight differences, many facts appearing in the earlier filing do not appear in the later and the reverse. This happens even if both filing are of the same kind, in this case they are both "Form 10-Q -- Quarterly report [Sections 13 or 15(d)]" filings.

These differences among filings, even when they are of the same type and from the same company, make it really difficult to integrate them and to perform queries crossing individual filing boundaries. Consequently, we have taken profit from the semantic integration tools provided by Semantic Web technologies. The Web Ontology Language (OWL) provides a set of primitives that allow stating that two classes, two properties or two instances are the same. It is also possible to state that something is a subclass or subproperty of another class or property respectively, that two classes are disjoint, etc.

These semantic integration statements are then used by inference reasoners, which are capable of dealing with their semantics while making their implications totally transparent for the users or applications using them. For instance, it is possible to state that, continuing with the previous example, “InvestmentLeaseReceivable” and “InvestmentInLeaseReceivable” are equivalent. Consequently, when the user queries for any of them, the other will be automatically included in the results.

Unfortunately, the process of detecting equivalent or similar concepts and relations from different ontologies, called ontology alignment, is a very time consuming one. Moreover, in the case of the EDGAR XBRL filings, there are a lot of ontologies to align because, as has been already mentioned, each filing has its own one. Consequently, automatic or semiautomatic alignment tools are required in order to get a scalable solution.

Currently, we have just performed some alignments among the ontologies for the Adobe Systems Inc. filings. This alignment process has not been integrated into the whole XBRL to Semantic Web application yet. For instance, we have applied the alignment implementation provided by the Falcons tool [20] for the two Adobe Inc. ontologies commented in this section getting a equivalence matching quality of “0.988” for the “InvestmentLeaseReceivable” and “InvestmentInLeaseReceivable” properties.

The maximum quality value is “1.0”, which is has been obtained for the 26 properties with identical names in both ontologies. Overall, more than 70 have been obtained with a minimum matching quality of “0.741”. The amount of concepts and properties with the same or very similar labels seems to indicate that it is possible to achieve a great degree of semantic integration among the ontologies for the filings coming from the same companies. We are currently evaluating the quality of the alignments generated by different tools and for filings from different companies.

In any case, we can currently take profit from the fact that most facts in the filings are not from these filing specific taxonomies. Most of them come from the standard XBRL taxonomies. Consequently, we have focused on integrating and cross querying filings from the point of view of the facts from these standard taxonomies, as it is shown in the Evaluation section, where the results of the previous process of moving XBRL data to the semantic space have been put into practice.

4. Evaluation

The XSD2OWL and XML2RDF mappings have been validated in different ways. First, we have used OWL validators in order to check the logical consistency of the resulting ontologies. Once all the ontologies were validated, which also includes checking that all the dependencies among them are met, we proceeded to put them into practice, together with the semantic metadata generated by the XML2RDF mapping.

In parallel with our efforts, the ontologies we have generated for XBRL using the XSD2OWL mapping are being used by OpenLink Software[15], who has also tested them independently. These ontologies have been chosen by OpenLink as the ontological framework for their software component responsible for translating XBRL data to semantic data based on RDF, which they call the XBRL Sponger.

[15] OpenLink Software, http://www.openlinksw.com

This parallel effort provides us an independent evaluation of the generated ontologies, which they have found as appropriate in order to structure the RDF data they generate from the XBRL filings. Moreover, they also generate RDF data from XBRL so we have also evaluated our XML2RDF mapping in comparison to their mapping. As it is shown below, they have implemented their own mapping for this step thought their and our instance level mappings are based on the same ontologies.

First of all, there is a significant difference in the number of triples generated by the OpenLink XBRL Sponger and XML2RDF. For instance, for the same EDGAR XBRL filing[16], the XBRL Sponger produces 900 triples while XML2RDF produces 4739 triples. One possible reason for this difference is that we have followed quite different approaches relative to how the original XML tree structure is captured in the RDF graph. However, there is also a significant difference in the amount of instance data captured in the output RDF. While XBRL to RDF captures all the data in the original XBRL instance, the XBRL Sponger captures just a small part of it in comparison.

For instance,

Table 7 shows in the first row a portion of XBRL XML instance data from the previous filing. This XBRL corresponds to a context and to a fact that references de previous context. The second row contains the RDF generated from the previous XBRL XML by the OpenLink Sponger. As it can be shown, the result is a "sioc:Contanier" object for the context object that contains just some of the properties of the original container plus the fact and its value. Some of the information for the context and most of it for the fact is not captured. Moreover, the whole structure is flattened.

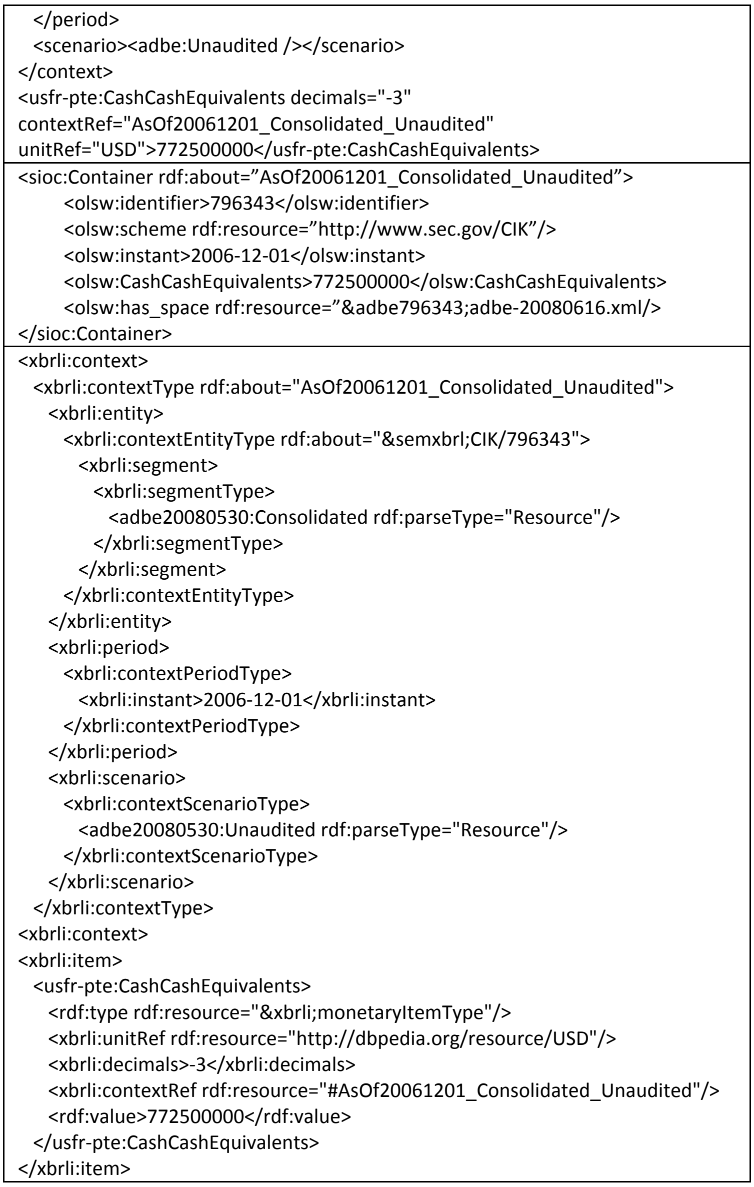
On the other hand, the third row in

Table 7 shows the mapping for the same XBRL XML as generated by our XML2RDF mapping. As it can be seen, the result is much move verbose, even more than the original XBRL. However, it does capture all the original information and keeps the original structure. Even more, the original XBRL does not explicitly refer to the XML Schema *complexTypes* defined in the schemas and used in the instance data. This information is available in the XML2RDF semantic data and can be used, together with the hierarchical realtions among complex types, when resolving semantic queries against this data.

Table 7. XBRL XML instance data example (first row), OpenLink XBRL Sponger mapping (second row) and XML2RDF XBRL mapping (third row) for the previous example

```
<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>
```

[16] Adobe Systems Inc. EDGAR filing 2008-07-03, XBRL file: http://www.sec.gov/Archives/edgar/data/796343/000079634308000005/adbe-20080616.xml



```
  </period>
  <scenario><adbe:Unaudited /></scenario>
</context>
<usfr-pte:CashCashEquivalents decimals="-3"
contextRef="AsOf20061201_Consolidated_Unaudited"
unitRef="USD">772500000</usfr-pte:CashCashEquivalents>
```

```
<sioc:Container rdf:about=”AsOf20061201_Consolidated_Unaudited”>
      <olsw:identifier>796343</olsw:identifier>
      <olsw:scheme rdf:resource=”http://www.sec.gov/CIK”/>
      <olsw:instant>2006-12-01</olsw:instant>
      <olsw:CashCashEquivalents>772500000</olsw:CashCashEquivalents>
      <olsw:has_space rdf:resource=”&adbe796343;adbe-20080616.xml/>
</sioc:Container>
```

```
<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>
```

Apart from instance data, it is also possible to compare the OWL ontologies generated following the proposed approach to those available from the two other initiatives introduced in the related work section. It has not been possible to compare the instance data generated by these initiatives because it not publicly available nor documented in the corresponding publications or associated documents.

In relation with [5], which focuses on investment funds taxonomies and their corresponding ontologies, they also perform an automatic mapping from XBRL taxonomies to OWL ontologies. However, the mapping is not as complete as the proposed one, especially in relation with cardinalities. The cardinalities in the input XBRL taxonomies do not seem to be taken into account and thus the output ontologies define all properties as FunctionalProperties or cardinalities equal to one.

Finally, comparing with the results reported in [3], they focus on just one taxonomy, the IPP-XBRL taxonomy that was promoted by the Spanish Securities Commission (CNMV) then, and just instance data based on this taxonomy can be generated.

4.1.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 more than 2 million triples resulting from the mapping have been published online using the Rhizomer tool [21]. Data can be queried, traversed and edited online[17] through a web user interface for human users. Moreover, through HTTP and content negotiation, Rhizomer also makes data available for machine consumption and makes it possible to integrate it into the Web of Linked Data. The overall architecture of this solution 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.

This 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.

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

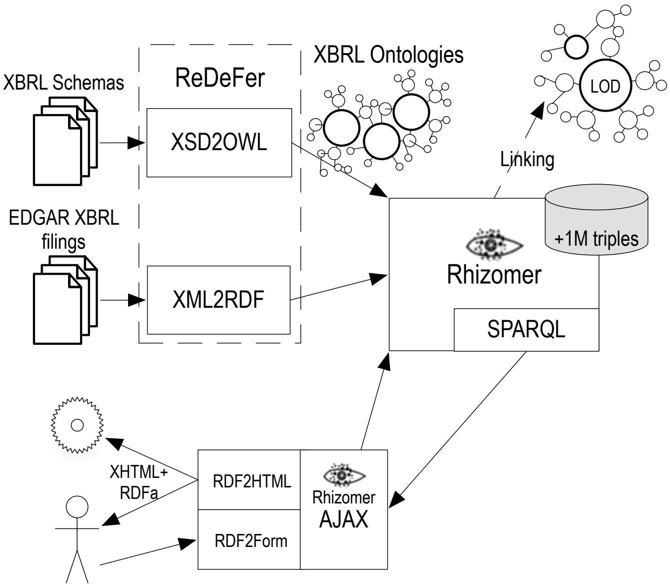


Fig. 2. Architecture of the proposed solution for semantic XBRL generation, linking and publishing

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 [22] (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[18] repository that provides scalability to more than tens of millions of triples and provides RDF Schema inferencing and support for OWL equivalence constructs.

5. Conclusions and Future Work

As it has been shown, it is possible to map the XML data for XBRL filings in order to generate RDF semantic data that keeps all the original information and structure. This mapping also includes the involved XML Schemas that structure the XML 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 con-

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

texts, facts and other resources defined for different filings can be related as more specific, more general or equivalent.

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 2 million triples have been obtained.

Our approach has been independently validated by the OpenLink Software, a company that is currently using the resulting XBRL ontologies in its XBRL to RDF mapping product. However, OpenLink does not follow the same XML to RDF mapping approach. Their approach has been compared to ours showing that our proposal retains much more of the original XBRL information and structure.

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.

However, it is important to note that 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 [5].

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 and test the benefits of cross-filings semantic queries and browsing.

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 available through the Parallax interface to Freebase [23]. This would allow performing semantic queries for specific facts across different filings and then plotting their values.

Acknowledgements

The work described in this chapter 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).

References

[1] Lytras, M., García, R. Semantic Web Applications: A framework for industry and business exploitation - What is needed for the adoption of the Semantic Web from the market and industry, International Journal of Knowledge and Learning 4(1) (2008) 93-108.

[2] Bizer, C., Heath, T., Idehen, K., Berners-Lee, T. Linked data on the web (LDOW2008), in: Proceeding of the 17th international conference on World Wide Web, ACM, 2008, pp. 1265-1266.

[3] Núñez, S., de Andrés, J., Gayo, J. E., and Ordoñez, P. A Semantic Based Collaborative System for the Interoperability of XBRL Accounting Information, in: Emerging Technologies and Information Systems for the Knowledge Society. Lecture Notes in Computer Science Vol. 5288, Springer, Berlin, 2008, pp. 593-599.

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

[5] Lara, R., Cantador, I., and Castells, P. Semantic Web Technologies for The Financial Domain, in: J. Cardoso and M. Lytras (Eds.), The Semantic Web: Real-World Applications from Industry. Springer, Berlin, 2008, pp. 41-74.

[6] Erling, O., Mikhailov, I. RDF Support in the Virtuoso DBMS, in: Pellegrini, T., Auer, S., Tochtermann, K., and Schaffert, S. (eds.) Networked Knowledge - Networked Media, Springer, 2009, pp. 7-24.

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

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

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

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

[11] Patel-Schneider, P.F., Simeon, J. The Yin/Yang web: XML syntax and RDF semantics, in: Proceedings of the 11th World Wide Web Conference, WWW'02. ACM Press, 2002, pp. 443-453.

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

[13] García, R., Gil, R., Delgado, J. A Web Ontologies Framework for Digital Rights Management, Journal of Artificial Intelligence and Law 15, 2 (2007) 137-154.

[14] García, R., Gil, R. Facilitating Business Interoperability from the Semantic Web, in: Proceedings of the 10th International Conference on Business Information Systems, BIS'07, Lecture Notes in Computer Science Vol. 4439, Springer, Berlin, 2007, pp. 220-232.

[15] García, R., Perdrix, F., Gil, R., Oliva, M. The Semantic Web as a Newspaper Media Convergence Facilitator, Journal of Web Semantics 6, 2 (2008) 151-161.

[16] García, R., Tsinaraki, C., Celma, O., Christodoulakis, S. Multimedia Content Description using Semantic Web Languages, in: Semantic Multimedia and Ontologies: Theory and Applications, Y. Kompatsiaris and P. Hobson Eds. Springer, Berlin, 2008, pp. 17-54.

[17] Berners-Lee, T. Why RDF model is different from the XML model. W3C Dessign Issues, 1998. http://www.w3.org/DesignIssues/RDF-XML.html

[18] Tous, R., García, R., Rodríguez, E., and Delgado, J. Arquitecture of a Semantic XPath Processor, in: Proceedings of 6th Int. Conference on E-Commerce and Web Technologies, K. Bauknecht, B. Pröll and H. Werthner Eds., EC-Web'05, Lecture Notes in Computer Science Vol. 3590, Springer, Berlin, 2005, pp. 1-10.

[19] 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.

[20] Hu, W., Qu, Y. Falcon-AO: A practical ontology matching system. Journal of Web Semantics, 6, 3 (2008) 237-239.

[21] García, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M. 2008. A Platform for Object-Action Semantic Web Interaction, in: Proceedings of the 16th Int. Conf. on Knowledge Engineering and Knowledge Management Patterns, A. Gangemi, J. Euzenat Eds., EKAW'08. Lecture Notes in Computer Science Vol. 5268, Springer, Berlin, pp. 404-418.

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

[23] Huynh, D., Karger, D. Parallax and Companion: Set-based Browsing for the Data Web. Submitted to the World Wide Web Conference, 2009. Available from http://davidhuynh.net/media/papers/2009/www2009-parallax.pdf