Representing Financial Reports on the Semantic Web

- A Faithful Translation from XBRL to OWL

Jie Bao[1], Graham Rong[2], Xian Li[1], and Li Ding[1]

[1]Tetherless World Constellation
Rensselaer Polytechnic Institute
Troy, NY, 12180-3590, USA
{baoj,lix15,dingl}@rpi.edu

[2]Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA, 02142, USA
grong@sloan.mit.edu

Abstract. We discuss a translation of financial reports from the XBRL format into Semantic Web language OWL. Different from existing approaches that do mechanic translation from XBRL's XML schema into OWL, our approach can faithfully preserve the implicit semantics in XBRL and enable the logic model of financial reports. We show that such a translation reduces the risk of redundancy and inconsistency, and enables the quick and useful inference on XBRL based financial reports for business decisions.

1 Introduction

XBRL (eXtensible Bussiness Reporting Language) is an XML-based standard for exchanging business information, e.g., public company financial reports. XBRL provides considerable benefits in the preparation, analysis and communication of business information. In recent years there has been rapid growth in international adoption of XBRL (cf. a survey as of Apr 2010 [12]). U.S. Securities and Exchange Commission (SEC) has mandated all companies to file financial reports in XBRL by October 31, 2014.

However, despite its broad acceptance, XBRL per se remains largely a structural model of financial reports, without addressing the *logic model* of these reports. For example, while we can declare the equivalency of two concepts in XBRL using arc roles, there is no means in XBRL to infer new relations from the equivalency relation. For another example, we can not specify in XBRL that an item should not be listed both as credit and as debit items. Furthermore, the document-oriented nature of XBRL makes it difficult to process, browse and query data from a large set of XBRL files.

Recently Semantic Web [1] has been argued as a natural choice for complementing XBRL with a logic or semantic data model [13, 11, 10]. This is due to the fact that Semantic Web languages, e.g., RDF and OWL, are inherently built with a graph-based open data model and naturally support integration from different data sources and applications. Furthermore, these languages are based on formal knowledge representation formalisms thus enable automatic processing and inference about data.

Garcia and Gil [13] have provided a mapping from XBRL to RDF and OWL (additional details are discussed in related work). This mapping is based on a structural transformation from XML Schema to OWL and thousands of XBRL reports have been published as linked data using this approach. rdfabout.com[1] provides corporate ownership information derived from SEC filings; however, that data is only a partial mapping from financial report data covering individual ownership and subsidiary information for selective companies.

In this paper, we provide an improved semantic data model for XBRL by translating it into OWL. This is done by making explicit the implicit semantic assumptions and constraints in XBRL. Compared with previous work, our contributions are:

- Our model is based on the intended semantic model of XBRL, which is currently provided informally as human-readable description in the XBRL specification [8] and is only partial captured by the current XML schema. By encoding these implicit semantics using OWL, we obtained a more accurate data model for XBRL that incorporates domain knowledge.
- To correctly capture semantics of XBRL, we are required to model both ontological constraints and rule constraints. To ensure desirable computational properties of the result, we transform rules into OWL 2 DL axioms which is known to be decidable (i.e., can ask any queries in finite time) and have mature tool support. This further enables automatic processing and reasoning of financial data represented using our model.
- Leveraged by the inference capability of OWL, the semantic data model is significantly simplified from the XBRL structural model (as given in the XML schema) without losing meaningful information. This reduces redundancy in the data as well as the risk of data inconsistency.

2 Preliminary: XBRL

The XBRL Specification[2] offers a framework for definition and extension of semantics in business reporting and production and validation of data from entities that need to communicate business performance. XBRL employs XML Schema and XLink technologies to describe different *taxonomies* for specific domains so that each XBRL document is an instance of an specific XBRL taxonomy.

[1] http://rdfabout.com/demo/sec/
[2] http://www.xbrl.org

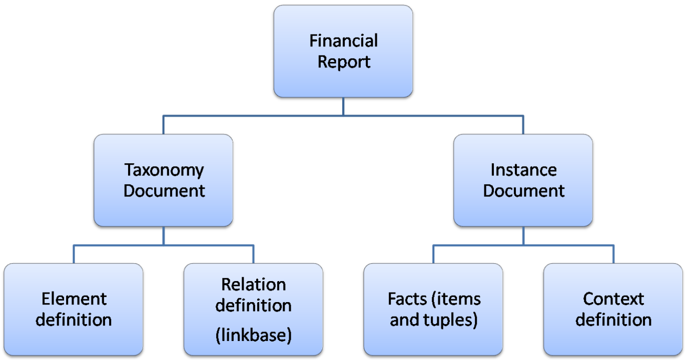

Fig. 1. Structure of XBRL financial reports

Different XBRL taxonomies have been developed in different jurisdictions, for instance, U.S. GAAP Financial Statement Taxonomy and China Listed Company Information Disclosure Taxonomy. These taxonomies are essentially different XBRL data standards. A taxonomy consists of a taxonomy schema and a set of *linkbases*. A taxonomy schema defines the reporting *concepts* as XML *elements*. Each element is given a name and a type. XBRL *instances* contain the *facts* as well as the descriptions of their *contexts* using the context and unit elements. The XBRL specification requires that an element should be associated with a context element using the contextRef attribute. If the element is of a numeric type (or a type derived from a numeric type), it must be associated with a unit element using the unitRef attribute. The structure of a typical XBRL reports is shown in Figure 1.

XBRL relies heavily on Xlink for everything, e.g. taxonomies, versioning and formulae. The set of xlinkes (called arcs) in a XBRL taxonomy forms its linkbase. There are five types of arcs defined in the XBRL standard:

- A *definition* arc specifies the conceptual relationships between elements, such as parent-child relationship and equivalency relations.
- A *calculation* arc defines the numeric relationships between elements.
- A *presentation* arc specifies the hierarchical grouping and the order of the elements when they are presented in a report for viewing purposes.
- A *label* arc provides the human-readable documentation for the elements defined in the taxonomy schema. An element may have multiple labels in different natural languages, or to be used for different documentation purposes and in different scenarios.
- A *reference* arc, similar to the label arc, provides further explanations to the elements by linking them to authoritative references (e.g., SEC regulations or FASB (Financial Accounting Standards Board) standards) that define the meaning of the elements.

Each US SEC XBRL filing consists of: 1) instance files with reported facts, including numeric and textual facts, dates and periods, footnotes and reporting dimensions; 2) taxonomies (schemas) for the instance file, including the definitions of markup elements, references to the reporting taxonomy, etc.; 3) taxonomy extensions, including labels for reporting concepts, currencies, additional reporting concepts and relationships between these concepts.

3 Semantic Web and XBRL: A Motivating Example

In this section, we describe a hypothetical motivating application about investment research and show that how a Semantic Web representation of XBRL reports will enable better consumption of the data and leverage investment decision making.

An investor wants to pick up stocks in the smart phone industry to diversify his portfolio. He needs information from different perspectives to access the risk levels of each company within the industry. Using conventional tools, he may look at current and historical market data to understand the volatility of company's equity through the market information from service providers, such as Bloomberg or Thomson Reuters, or from free Web services, such as Yahoo! Finance. In addition, the investor may conduct equity fundamental analysis using financial statements data, e.g., the company's performance, debt level, unhedged risk, etc. Although XBRL filings enable the interactive access to financial reports, it still requires tremendous human cognitive efforts to draw comparisons across different filings of a given company, or certain financial measurements across different companies. For instance, the investor may need to look at both 10-Q (quarterly report) and 10-K (annual report) from several companies in the past a few years to understand the financial landscape of the given industry.

Semantic Web technologies may help the investor from the following aspects:

– *Semantic Web technologies provide an expressive and flexible data model* (with respect to tree- (XML) or relational data models) that makes the querying, filtering and visualization of data easier. Comparisons across different documents become simpler with data being represented as a graph model. The investor can select a certain risk measurement, e.g., cash flow, and compare the cash flow reports both across companies within an industry and within a single company in different time periods. Because the underlying logical relationships among the data is represented, relevant information such as competitors, subsidiaries, industry categories can be easily be connected and compared.
– *Semantic Web simplifies the integration with external data sources and domain knowledge bases.* For example, the investor may want to find the risk positions of all companies that make smart phones within Northeastern America. For this goal, in addition to financial information from SEC interactive data reports, he may use company profile knowledge and smart phone model and manufacturer knowledge from DBPedia[3] He may also utilize social information from the Web. For example, he just wants to focus on

[3] http://dbpedia.org. DBPedia is a Semantic Web knowledge base with facts extracted from Wikipedia articles.

companies whose board members are more concentrated instead of being a board member of many companies (from the rdfabout data set). By linking the data in financial information with other social information, e.g., current events, political instability, customers' market shares, one can evaluate a company not only based on its financial reports but all the entities in its entire ecosystem.
- *Semantic Web languages provide inference abilities for discovering insights from data.* For instance, a phone is very likely to be a smart phone if it provides wifi connection, even if it is not explicitly listed as a "smartphone". For another example, if "cash" is part of "current assets", and "current assets" are part of "assets", then we can infer that "cash" is part of "assets".

In summary, with Semantic Web, financial information translated from XBRL-based reports can be used together with ontologies developed based on different domain expertise and analysis interests to allow the different insights and views for knowledge that otherwise can not easily be available.

4 Representing XBRL Data Model for the Semantic Web

In this section, we describe a translation of the XBRL data model into Semantic Web representations using OWL (Web Ontology Language). More specifically, we use OWL 2 DL [9] to achieve both the semantic faithfulness of the translation and desirable computation properties (e.g., inference and query complexity) of the resulting knowledge bases (KB). For the sake of readability, we use the OWL 2 Functional-Style Syntax.

The translation results in several different types of KBs:

- An XBRL ontology that captures some of the structural constraints defined in the XRBL XML Schema specification, and implicit semantic requirements which are only informally given by the specification or by default assumptions. This ontology will be shared by all translated KBs and its components are identified by the ★ sign in the paper.
- Taxonomy ontologies that correspond to XBRL taxonomy documents.
- Instance ontologies that correspond to XBRL instance documents.

The URL prefixes we used in the translation are given in Table 1.

Our translation is based on XBRL 2.0 [8]. The result can be easily extended to XBRL 2.1 as it has only a set of small changes to XBRL 2.0[4].

4.1 Representing XBRL Elements

We first describe the translation of XBRL taxonomies into OWL.

[4] Note that some XBRL 2.0 features, e.g., The CWA attribute on the numericContext element, has been eliminated in XBRL 2.1. Those features should be removed from the translated KB of an XBRL 2.1 document accordingly

Table 1. Prefixes used in the translation

Prefix	URL	Note
owl	http://www.w3.org/2002/07/owl#	OWL
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF
rdfs	http://www.w3.org/2000/01/rdf-schema#	RDF Schema
xsd	http://www.w3.org/2001/XMLSchema	XML Schema datatypes
xbrli	http://www.xbrl.org/2003/instance	XBRL instance
xbrll	http://www.xbrl.org/2003/linkbase	XBRL linkbase
xbrlo	http://tw.rpi.edu/2010/xbrl	XBRL ontology
time	http://www.w3.org/TR/owl-time/	OWL time ontology
ex	http://www.example.com#	A fictional website

Elements The <element> tag defines an XBRL concept which corresponds an OWL class. For example, the following is an XBRL element of monetaryItemType and its OWL translation:

XBRL

```
<element id="currentAssets"
   name="currentAssets"
   type=xbrli:monetaryItemType
   xbrli:balance="credit"
   substitutionGroup = "xbrli:item">
</element>
```

OWL

```
Declaration( Class(ex:currentAssets))
SubClassOf(ex:currentAssets xbrlo:monetaryItemType)
SubClassOf(ex:currentAssets xbrlo:credit)
```

Note that the `substitutionGroup` attribute is translated since it only has value `xbrli:item` or `xbrli:tuple`, which can already been inferred from the `type` information. The optional `id` attribute is also not translated as it's almost always the same as `name` (which is required to be uniquely identifiable).

If an element is the root of a taxonomy, it is a subclass of `xbrlo:root`.

Elements Types (★) The basic element types in XBRL form a class hierarchy (prefix `xbrlo:` omitted):

- `itemType` has 6 subclasses: `monetaryItemType, sharesItemType, decimalItemType, stringItemType, uriItemType` and `dateTimeItemType`
- `elementType` has 2 subclasses: `itemType` and `tupleType`
- `balanceType` has 2 subclasses: `credit` and `debit`

Each instance of an item type has one and only one value of a particular datatype, e.g., `stringItemType` should have a value and the value is of the `xsd:string` type:

```
SubClassOf(xbrlo:itemType
       DataExactCardinality( 1 xbrlo:value))
SubClassOf(xbrlo:stringItemType
       DataAllValuesFrom(xbrlo:value xsd:string))
```

The type constraint of item types is summarized in the following table:

`monetaryItemType, sharesItemType, decimalItemType`	`xsd:double`
`stringItemType`	`xsd:string`
`uriItemType`	`xsd:anyURI`
`dateTimeItemType`	`xsd:dateTime`

The tuple type will be discussed in the next section (instance document).

4.2 Representing XBRL Relations

Locators XBRL uses locators to identify a concept (element) in a taxonomy document, e.g.,

```
<loc xlink:type="locator" xlink:href="balanaceSheet.xsd#
    currentAssets"
xlink:label="loc_currentAssets">
```

Since in OWL we can directly identify a class using its IRI, it's not necessary to use locators. Therefore, the locator "loc_curentAssets can be replaced by the class "balancesheet.owl#currrentAssets". Thus, given a locator "L", we use C(L) to denote the class denoted by the locator.

Arcs Arc-type elements join the resources referenced in their from and to attributes, for instance:

```
<definitionArc xlink:type="arc"
  xlink:from="loc_assets"
  xlink:to  ="loc_currentAssets"
  xlink:show = "replace"
  xlink:acuate = "onRequest"
  xlink:title = "From Assets to Current Assets"
  xlink:arcrole = "http://www.xbrl.org/linkprops/arc/parent-
      child"/>
```

For conciseness, let C(loc_assets)=A and C(loc_currentAssets) = C. An arc is represented as a property in OWL. A naive approach is to relate the two resources (elements) using property assertions, such as

```
ObjectPropertyAssertion(ex:arc1 A C )
```

Where `ex:arc1` is a new property name for the arc. However, such an approach will result in a OWL 2 Full ontology (hence violate inference termination requirement in OWL 2) since classes A,C are also used as individuals. A better OWL 2 DL translation is:

```
EquivalentClasses(  A  ObjectHasSelf( ex:pA  ) )
EquivalentClasses(  C  ObjectHasSelf( ex:pC  ) )
SubObjectPropertyOf( ObjectPropertyChain( ex:PA
    owl:topObjectProperty ex:hasParent ) ex:arc1)

AnnotationAssertion(rdfs:label ex:arc1 "From Assets to
    Current Assets"^^xsd:string))
```

```
SubObjectPropertyOf(ex:arc1 xbrlo:definitionArc)
SubObjectPropertyOf(ex:arc1 xbrlo:parent-child)
```

The first three axioms encode the rule ex:arc1(x,y) $<-$ A(x), C(y), where ex:pA and ex:pC are two helper properties. This correctly captures the semantics that *instances* of A and C have the relation `xbrll:parent-child`.

Attributes `xlink:actuate` (always has value "onRequest") and `xlink:show` (has value "embed" if the resources linked are in different files, otherwise "replace") are not translated as they can be trivially inferred.

Arc Types There are 5 arc types which are all subclasses of `xbrlo:arc`:

- xbrlo:calculationArc[5]: it has an attribute "weight". To obtain an OWL 2 DL translation, we may introduce a helper individual for the arc and associate the weight to that individual (so that an XBRL processor can find such information), for example:

  ```
  EquivalentClasses( ObjectOneOf(ex:i1) ObjectHasSelf(
      ex:arc1 ) )
  DataPropertyAssertion (xbrl:weight exi1 "1"^^xsd:decimal)
  ```

- xbrlo:presentationArc: it has an **order** attribute which can be modeled similarly to calculationArc.
- xbrlo:definitionArc
- xbrlo:labelArc and xbrlo:referenceArc are non-semantic types and are omitted in this paper.

Arc Roles (★) Arc roles have intended semantics. For example, if `loc_assets` has a child-parent relation to `loc_currentAssets`, then it is expected that `loc_currentAssets` has a parent-child relation to `loc_assets`. However, such semantics are left implicit in the XBRL specification, which lead to both redundancy and the risk of data inconsistency. Thus, we have to also add to the XBRL taxonomy document that

```
<definitionArc xlink:type=arc
  xlink:from  ="loc_currentAssets"
  xlink:to  ="loc_assets"
  xlink:show = "replace"
  xlink:acuate = "onRequest"
  xlink:title = "From Current Assets to Assets"
  xlink:arcrole = "http://www.xbrl.org/linkprops/arc/child-
     parent"/>
```

Leveraging OWL's inference ability, we can eliminate such redundancy by defining properties of arc role. For example, the following OWL axioms declare that `parent-child` and `child-parent` are inverse to each other, and that a definition arc is symmetric:

[5] Note that while OWL itself does not provide numeric calculation, there are extensions of OWL that are able to, cf. Manchester OWL Arithmetics http://www.cs.man.ac.uk/ iannonel/owlcalculations/syntax.html

```
InverseObjectProperties(xbrlo:parent-child xbrlo:child-parent
    )
SymmetricObjectProperty(xbrlo:definitionArc)
```

Similar declarations are added for other arc roles (e.g., `xbrlo:dimension-element` is inverse of `xbrlo:element-dimension`) and arc types. As `xbrlo:dimension-element` indicates equivalency, we require it to be reflexive, transitive and symmetric, i.e.,

```
ReflexiveObjectProperty(xbrlo:dimension-element)
TransitiveObjectProperty(xbrlo:dimension-element)
SymmetricObjectProperty(xbrlo:dimension-element)
```

Fig 2 shows an example of calculating assets using different dimentions[6]. In XBRL, 18 arcs are required whereas in the OWL translation only 9 arcs are needed; in addition, we can infer that "Total Assets by Geography" and "Total Assets by Produce Line" must have the same value without calculating the value of "Assets".

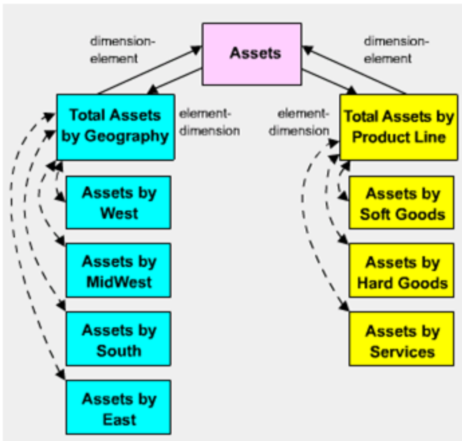

Fig. 2. Equivalency relations and child-parent relations

4.3 Representing XBRL Instances

Items Items are actual facts in the report and are translated into OWL fact assertions. For example, the XBRL fragment:

```
<assets numericContext="c1">300</assets>
```

is translated into OWL assertions

[6] The example is originally from http://us.kpmg.com/microsite/xbrl/train/86/86.htm

```
ClassAssertion(ex:assets _:x1)
ObjectPropertyAssertion(_:x1 xbrlo:hasContext ex:c1)
DataPropertyAssertion(_:x1 xbrlo:value "300"^^xsd:double)
```

where _:x1 is a newly introduced anonymous individual.

Tuples Tuples are concepts that are used to contain other concepts. The structural relation of a tuple with is component concepts is represented using the `xbrlo:tupleValue` property, e.g.,

```
<address>
  <street>8th St</street>
  ...
</address>
```

is translated to

```
ClassAssertion(ex:address _:x1)
ClassAssertion(ex:street _:x2)
ObjectPropertyAssertion(_:x1 xbrlo:tupleValue _x2)
DataPropertyAssertion(_:x2 xbrlo:value "8th St"^^xsd:string)
```

Contexts (★) Contexts are used to provide additional information related to the items (facts). A context is an instance of the class `xbrlo:numericConcept` or `xbrlo:nonNumericContext`, which are both subclasses of `xbrlo:context`. For example:

```
<numericContext id="c1" precision="12" cwa="true">
  <period><instant>2001-12-31</instant></period>
  ...
<numericContext>
```

will be translated into OWL

```
ClassAssetion(xbrlo:numericContext ex:c1)
DataPropertyAssertion(xbrlo:precision ex:c1 "12"^^xsd:integer
    )
DataPropertyAssertion(xbrlo:cwa ex:c1 "true"^^xsd:boolean)
ObjectPropertyAssrtion(xbrlo:period ex:c1 _:p1)
ClassAssertion(time:Instant _:p1)
DataPropertyAssertion(time:inXSDDateTime _:p1 "2001-12-31'')
```

Here we reuse the OWL Time ontology[7] to represent period data.

The `xbrlo:context` class contains optional components `entity`, `period`, `unit` and `scenario`. The `xbrlo:numericConcept` class has additional required attributes `precision` and `cwa` (closed world assumption)[8]. This requirement can be represented as cardinality constraints in OWL (partially shown)

```
SubClassOf(xbrlo:contex ObjectMinCardinality("0"^^xsd:integer
     xbrlo:entity))
```

[7] http://www.w3.org/TR/owl-time/

[8] Note that CWA in XBRL is different from CWA in OWL [14] which models integrity constraints.

```
SubClassOf(xbrlo:numericConcept ObjectMinCardinality("1"^^
    xsd:integer xbrlo:precision))
```

We summarize the correspondence of key XBRL and OWL notions in Table 4.3. Translation of some non-semantic features of XBRL, e.g., annotations, are given in the appendix.

Table 2. Correspondence of XBRL and OWL notions

XBRL	**OWL**
Taxonomy Document	Axiom set
Instance Document	Assertion set
Element	Class
Datatype	Datatype
Locator	directly identified by the resources IRI
Arc	Property
Item	Instance
Context	Instance (of `Context` class)
"`type`" attribute	"`SubClassOf`" axiom
"`name`" attribute	local name of the IRI of the resource
"`id`" attribute	not translated

5 Related Work

The importance of making financial data open has been widely recognized. Garcia and Gil [13] explored transforming financial data to linked data so as to make them accessible on the Web and linkable with other linked open data. They adopted a direct approach of mapping public financial data which are in the format of XBRL to RDF format, and then publishing them on the Web as Linked Open Data. The actual data in the SEC filings were converted to RDF by modeling XML tree using triples, and the schema of the SEC Edgar database was restructured as OWL. Although the resources in XBRL were typed with classes in ontologies corresponding to the schema as well as ontology alignment to integrate ontologies for different filings, the drawback of this translation method is that it fails to preserve the relationships among the classes. For example, the relations among debt, equity, and financial leverage ratio. This is partly due to that this approach relying on a structural transformation from XML Schema to RDF and OWL without addressing the semantic modeling of XBRL. On the other hand, our approach translates XBRL into OWL based on the intended semantic model of XBRL, not its structural representation.

As analyzed in their most recent talk[6], Garcia summarized that lessons from their existing work in publishing open financial data as linked data, and pointed out further issues that need to be addressed, such as extend current direct mappings acrossing filings for the same company, across companies or even across accounting principles. The difficulties of modeling mathematical calculations among reporting values using OWL are also recognized as a major challenge.

Declerck and Krieger [4] translated XBRL into description logic which is the logic foundation of OWL. This work represents the XBRL base taxonomy using OWL. However, [4] does not specify how to translate linkbase.

Several other authors explored the integration of Semantic Web technologies and the XBRL technology in general. The work from [2] elaborated the requirements of efficient data organization and searching the perspective of industry, and highlighted the importance of specific taxonomies in presenting data for analysis in the the process of high speed trading, moving through data supply chain, screening it and finding comparable in a timely way at decision-making level. They also pointed out the potential benefits of taxonomy extension for for filing companies which do different kinds of extensions.

The agility of Semantic Web tools in integrating financial data of XBRL and other formats are expressed in [3] by stating that XBRL takes what was unstructured and adds structured data, while semantic standards add meaning to that data so that it becomes very consumable, usable and agile, providing real time insight into integrated data. The idea was demonstrated with commercial products, both an plugin for Excel which creates ontologies from the spreadsheet data, and Web-based one where ontology is exposed and data are pulled into a view with faceted browsing, sorting, reformating, filtering and further graphical views would be generated. The goal of these products are to make use of aggregated data and share them in real time.

Opportunities of Semantic Web knowledge representation in helping XBRL was explored by [7]. The overall benefit that Semantic Web can offer XBRL was summarized in terms of knowledge representation with expressiveness, interoperability and performance optimization, knowledge acquisition targeting business users collaboration, related domain ontologies and knowledge base, and variability to more applications and domains.

None of the above work provides a formal model for XBRL that faithfully captures the implicit semantics of XBRL and enables automatic inference using XBRL data, which are the focus of our work.

6 Conclusions

In this paper we provide a semantic data model for XBRL by using OWL to express the implicit semantics currently described informally in XBRL specifications. We show that such a semantic model is able to better capture the domain knowledge related to financial reports, and reduces redundancy in the original XBRL model. We also expect such a representation will enhance transparency in reporting of financial data, as well as the integration of financial reports and other domain knowledge bases.

Our ongoing work includes the modeling of the US GAAP (Generally Accepted Accounting Principles) and the IFRS (International Financial Reporting Standards) taxonomies using Semantic Web languages. Another future work is to publish XRBL data in the SEC EDGAR database as a part of the semantic government data cloud (http://data-gov.tw.rpi.edu) [5] and link it to other government data sets (e.g., bankruptcy data and macroeconomic data).

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
2. A. Bhatnagar. Xbrl taxonomy extension comparability issues and potential semantic web solutions. http://www.w3.org/2009/03/xbrl/talks/Ashu-Bhatnagar.pdf, 2009.
3. M. Cataldo. The agility of semantic web tools for xbrl and other financial data. http://www.w3.org/2009/03/xbrl/talks/Michael-Cataldo.pdf, 2009.
4. T. Declerck and H.-U. Krieger. Translating xbrl into description logic. an approach using protege, sesame & owl. In *BIS*, pages 455–467, 2006.
5. L. Ding, D. DiFranzo, A. Graves, J. R. Michaelis, X. Li, D. L. McGuinness, and J. Hendler. Data-gov wiki: Towards linking government data. In *Proceedings of the 2010 AAAI Spring Symposium on Linked Data Meets Artificial Intelligence*, 2010.
6. R. Garca. Ontological challenges for financial information: some lessons from the rhizomik initiative. http://www.w3.org/2009/03/xbrl/talks/Roberto-Garcia.pdf, 2009.
7. B. Grosof. Opportunities for semantic web knowledge representation to help xbrl. http://www.w3.org/2009/03/xbrl/talks/Benjamin-Grosof.pdf, 2009.
8. L. Hampton and D. vun Kannon. Extensible Business Reporting Language (XBRL) 2.0 Specification. http://www.xbrl.org/tr/2001/xbrl-2001-12-14.pdf, Dec 2001.
9. B. Motik, P. F. Patel-Schneider, and B. Parsia. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. World Wide Web Consortium (W3C) Recommendation, 2009.
10. S. M. Núñez, J. de Andrés Suárez, J. E. L. Gayo, and P. O. de Pablos. A semantic based collaborative system for the interoperability of xbrl accounting information. In *WSKS (1)*, pages 593–599, 2008.
11. S. M. Núñez, J. E. L. Gayo, J. de Andrés, and P. O. de Pablos. Analysis of xbrl documents containing accounting information of listed firms using semantic web technologies. In *MTSR*, pages 375–381, 2007.
12. C. O'Kelly. XBRL World Wide Adoption Survey April 2010. http://www.slideshare.net/xbrlplanet/xbrl-world-wide-adoption-survey-april-2010, 2010.
13. R. G. Roberto Garcia. Publishing xbrl as linked open data. In *CEUR Workshop Proceedings*, volume 538, 2009.
14. J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity Constraints in OWL. In *AAAI*, page In Press, 2010.

Appendix

Translation of non-semantic elements in XBRL

XML Fragment	OWL Syntax
<annotation> <documentation>abc</documentation> <annotation>	Annotation(rdfs:comment "abc")
<import namespace="http://foo.com" schemaLocation="instance.xsd">	Ontology(Import(<http://foo.com/instance.owl>)
Extended links e.g., <defintionLink xlink:type="extended" xlink:role="....balanceSheet" [locators and arcs] </defintionLink>	Not translated
<xlink:footnoteArc ... xlink:from = loc_A xlink:to = B xlink:arcrole = "http://...footnote-fact"> <xlink:footnote ... xlink:label=B xml:lang="en">abc </xlink:footnote>	DataPropertyAssertion (xbrlo:value _:b "abc@en") Annotation(xbrlo:footnote A _:b) SubAnnotationPropertyOf (xbrlo:footnote rdfs:comment)