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 the Semantic Web language OWL. Different from existing approaches that do structural 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 better 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 [5]).

However, despite its broad acceptance, XBRL remains largely to be 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. Furthermore, its inherited document-oriented nature makes it difficult to process, browse and query data from a large set of XBRL files.

Recently Semantic Web has been argued as a natural choice for complementing XBRL with a logic or semantic data model [6]. 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. In addition, these languages are based on formal knowledge representation formalisms thus enable the automatic processing and inference about data.

Garcia and Gil [6] have provided a mapping from XBRL to RDF and OWL. This mapping is based on a structural transformation from XML Schema to OWL. Tthousands of XBRL reports have been published as linked data using this approach. rdfabout.com[1] provides the 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. Declerck and

[1] http://rdfabout.com/demo/sec/

Krieger [2] translated the XBRL base taxonomy into description logic which is the logic foundation of OWL (2) DL. However, they did not specify how to translate the linkbases in XBRL. None of the above work provides a logic model that faithfully captures the implicit semantics of financial reports in XBRL and enables the automatic inference on XBRL data.

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 include:

– Our model is based on the intended semantic model of XBRL which is currently provided informally as human-readable description in the XBRL specification [3] and is only partially captured in the current XML schema. By encoding these implicit semantics using OWL, we have obtained a more accurate data model for XBRL, that also incorporates domain knowledge.
– To correctly capture the semantics in XBRL, we need to model both ontological constraints and rule constraints. To ensure desirable computational properties of the result, we transform some rules into OWL 2 DL axioms which are known to be decidable, i.e., being able to answer any query 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 information. This reduces both the redundancy in the data model and the risk of data inconsistency.

It is noteworthy, while OWL 2 DL covers a fairly large subset of XBRL's expressivity, there are semantic constraints of XBRL that can only be modeled by other Semantic Web languages, e.g., RIF (Rule Interchange Format) and integrity constraints [7]. These are left as future work.

This paper is accompanied by a technical report [1] with more detailed motivating examples and related work, and additional details of the translation from XBRL to OWL.

2 Representing XBRL Data Model for the Semantic Web: General Issues

In this section, we describe the general issues of the translation of the XBRL data model into Semantic Web representations using OWL (Web Ontology Language). More specifically, we use OWL 2 DL [4] 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 XBRL Specification[2] offers a framework for the definitions of the semantics in business reporting and the production and validation of data from entities that need

[2] http://www.xbrl.org

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. A taxonomy consists of a *taxonomy schema* and a set of *linkbases*. A taxonomy schema defines the reporting *concepts* as XML *elements*. XBRL *instances* contain the *facts* as well as the descriptions of their *contexts* (such as the reporting date of the fact and the currency unit used).

Correspondingly, 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. We assume its URL base name is `xbrlo`.
– Taxonomy ontologies that correspond to XBRL taxonomy documents.
– Instance ontologies that correspond to XBRL instance documents.

For the naming convention and the URL prefixes used here, please refer to the accompanying technical report [1].

Our translation is based on XBRL 2.0 [3]. The result can be easily extended to XBRL 2.1 as it is an extension to XBRL 2.0.

3 Representing XBRL Concepts

We first describe the translation of XBRL taxonomies into OWL.

Concepts The `<element>` tag defines an XBRL *concept* which corresponds to 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 not translated since it can only have 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 usually the same as `name`.

Elements Types (★) The basic element types in XBRL form a class hierarchy (prefix `xbrlo:` omitted); every class is disjoint with its siblings (except `elementType` and `balanceType`):

```
elementType                      balanceType
  itemType                           credit
    numericItemType                  debit
      monetaryItemType
      sharesItemType
```

```
        decimalItemType
      stringItemType
      uriItemType
      dateTimeItemType
    tupleType
```

Each instance of an `itemType` has one and only one value of a particular datatype. For example, an instance of `stringItemType` should have exactly one value of the `xsd:string` type:

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

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

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

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

4 Representing XBRL Relations

XBRL relies on Xlink for relating entities defined in the schema, e.g., taxonomies and formulae. The set of xlinkes (called arcs) in an XBRL taxonomy forms its *linkbase*. There are five types of linkbases defined in the XBRL standard: *definition* linkbase, *calculation* linkbase, *presentation* linkbase, *label* linkbase and *reference* linkbase.

Locators XBRL uses locators to identify a concept (element) in a taxonomy document, e.g., the following defines a locator to the concept `"balanaceSheet.xsd #currentAssets"`.

```
<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`. Given a locator "L", we use C(L) to denote the class it points to (i.e., the class corresponds to its `xlink:href` property value).

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. Since an arc has no name in XBRL, a new property name is introduced for it.

A naive translation approach is to relate the two concepts (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 an OWL 2 Full ontology, hence violates inference termination requirement in OWL 2, since classes A and 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:pC ) 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 the *instances* of A and C have the relation parent-child.

Attributes xlink:actuate which always has value "onRequest", and xlink:show which has value "embed" if the resources linked are in different files and 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[3]: 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 ex:i1 "1"^^xsd:decimal)
  ```

- xbrlo:presentationArc: it has an order attribute which can be modeled similarly to calculationArc.
- xbrlo:definitionArc: discussed above.
- xbrlo:labelArc and xbrlo:referenceArc are non-semantic types and their translations are given in [1].

Arc Roles (★) Arc roles have intended semantics. For example, if loc_assets has a child-parent relation to loc_currentAssets, 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 leads to both the redundancy and the risk of data inconsistency. Thus, for the example above, we have to also add the following content to the XBRL taxonomy document:

```
<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"/>
```

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

Leveraging OWL's inference ability, we can eliminate such redundancy by defining the properties of the 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:

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

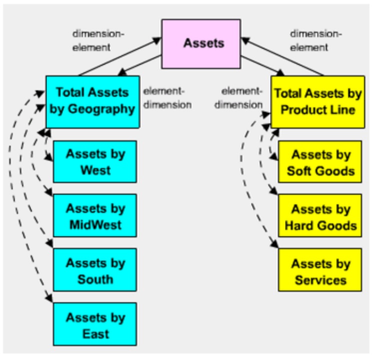

Fig. 1. Equivalency relations and child-parent relations

Note that `child-parent` (and similarly `parent-child`) relations in XBRL have different semantic meanings from OWL subclass relations. In XBRL, a `child-parent` relation describes how the *value* of *instances* of the related concepts are related, whereas in OWL a subclass relation means *subset relations between instance sets* of the related concepts.

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 1 shows an example of calculating assets using different dimensions[4]. 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".

5 Representing XBRL Instances

Items Items are actual facts in the report thus are translated into OWL fact assertions. Since they have no name in the XBRL document, they will be mapped to anonymous individuals in OWL. For example, the XBRL fragment:

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

is translated into OWL assertions

```
ClassAssertion(ex:assets _:x1)
ObjectPropertyAssertion(xbrlo:hasContext _:x1 ex:c1)
DataPropertyAssertion(xbrlo:value _:x1 "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 its component concepts is represented using the `xbrlo:tupleValue` property, e.g.,

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

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

is translated to

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

When the innermost block of a tuple has literal value, we can also use properties to model. For instance, the above example may also be modeled as

```
DataPropertyAssertion(ex:hasAddress _:x1 "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:numericContext` 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)
ObjectPropertyAssertion(xbrlo:period ex:c1 _:x)
ClassAssertion(time:Instant _:x)
DataPropertyAssertion(time:inXSDDateTime _:x "2001-12-31"^^xsd:dateTime)
```

Here we reuse the OWL Time ontology[5] to represent period data.

The `xbrlo:context` class contains optional components `entity, period, unit` and `scenario`. The `xbrlo:numericContext` class has additional required attributes `precision` and `cwa` (closed world assumption)[6]. This requirement can be represented as cardinality constraints in OWL (only two such constraints are shown here)[7]:

```
SubClassOf(xbrlo:context ObjectMinCardinality("0"^^xsd:integer
    xbrlo:entity))
SubClassOf(xbrlo:numericContext ObjectMinCardinality("1"^^xsd:integer
    xbrlo:precision))
```

Only an instance of `numericItemType` (`monetaryItemType`, `sharesItemType`, or `decimalItemType`) can have an instance of `numericContext` as its context, therefore we have the constraint:

```
SubClassOf(xbrlo:numericContext ObjectAllValuesFrom(
  ObjectInverseOf(xbrlo:hasContext) xbrlo:numericItemType)
```

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

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

[7] All cardinality constraints in our translation should be understood as integrity constraints using the semantics described in [7], i.e., they will be used for data validation but not inference of new knowledge.

We summarize the correspondence of key XBRL and OWL notions in Table 1. Translation of some non-semantic features of XBRL, e.g., annotations, are given in [1].

6 Conclusions

Table 1: Correspondence of Key XBRL and OWL Notions

XBRL	OWL
Taxonomy Document	Axioms
Instance Document	Assertions (facts)
Element	Named class
Datatype	Datatype
Locator	directly identified by the resource's IRI
Arc	Named property
Item	Anonymous individual
Context	Instance (of `Context` class)
"`type`" attribute	"`SubClassOf`" axiom
"`name`" attribute	local name of the IRI of the resource
"`id`" attribute	not translated
"`title`" attribute	`rdfs:label` annotation

In this paper we provide a semantic data model of XBRL-based financial data by using OWL so as to express the semantics currently described implicitly in XBRL specifications. We show that such a semantic model is able to better capture the domain knowledge related to financial reports, and reduces the redundancy, e.g. relation definition, in the current XBRL models. We also believe such a representation will enhance transparency in financial report filing, 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) and link it to other government data sets (e.g., bankruptcy data and macroeconomic data).

References

1. J. Bao, G. Rong, X. Li, and L. Ding. Representing financial reports on the semantic web (extended version). Technical report, TW-2010-17, Tetherless World Constellation, RPI, Troy, NY, USA.
2. T. Declerck and H.-U. Krieger. Translating XBRL Into Description Logic. An Approach Using Protege, Sesame & OWL. In *BIS*, pages 455–467, 2006.
3. 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.
4. 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.
5. C. O'Kelly. XBRL World Wide Adoption Survey April 2010. http://www.slideshare.net/xbrlplanet/xbrl-world-wide-adoption-survey-april-2010, 2010.
6. R. G. Roberto Garcia. Publishing xbrl as linked open data. In *CEUR Workshop Proceedings*, volume 538, 2009.
7. J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity Constraints in OWL. In *AAAI*, page In Press, 2010.