MFO—The Federated Financial Ontology for the MONNET Project

Hans-Ulrich Krieger, Thierry Declerck and Ashok Kumar Nedunchezhian
Language Technology Lab, German Research Center for AI (DFKI GmbH)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany

Keywords: Business Ontologies, Merging, Alignment & Integration, Representation of Temporally-changing Information.

Abstract: This paper describes work carried out in the European project **MONNET** which deals in part with the extraction of company data from stock exchange pages and its representation in a semantic repository on which inferences and queries are carried out. The special focus of the paper lies on the construction of an integrated ontology **MFO**—the MONNET Financial Ontology—that has been constructed from several independent ontologies which are brought together by an interface specification, expressed in OWL.

1 INTRODUCTION

Company data in MONNET is obtained by harvesting stock exchange Web sites, such as DAX or Euronext. This task is realized by "harvesters", standalone Java programs that produce ABox data, compliant with the integrated ontology. The output of a harvester is a set of *quintuples*, RDF triples that are "annotated" by two further temporal arguments, expressing the temporal extent in which an atemporal fact holds.

This paper shed some light on the company data and the construction of the integrated ontology that has been assembled from several independent ontologies which are brought together by an interface specification, expressed in OWL (McGuinness and van Harmelen, 2004). We will also sneak a peek on the temporal entailment rules (Krieger, 2012) that are built into the semantic repository hosting the data and which can be used to derive useful explicit information. This includes identifying companies from different times, monitoring data for unusual events, etc.

2 COMPANY SNAPSHOTS

We have mainly focused on the Xetra pages that are operated by *Deutsche Börse* and which include stock market indices, such as DAX, MDAX, SDAX, and TecDAX. The good thing with the representation of titles in the different indices is that their HTML pages have an identical layout, so that the Xetra harvester that was originally implemented for DAX perfectly works for the other Xetra indices as well.

Recently, indices from the NYSE Euronext have also been investigated, e.g., from Euronext 100 or Next 150. Again, the layout for these different indices in Euronext is identical, but since Xetra and NYSE Euronext are operated by different marketplace organizer which, in addition, offer differing information about companies, the Euronext harvester had to be reimplemented. Contrary to Xetra, Euronext seems to provide much more information for a company, e.g., a finer industry sector classification, but also fiscal/monetary data for the past three years.

Harvesting company Web pages not only means to collect captions and their corresponding values, but also to transform this data into a meaningful representation that is compliant with existing standards, such as XSD, RDF, and OWL, but also compatible with the ontology schema (see next section). This includes (i) the proper use of class and property names, (ii) the introduction of fresh URIs, (iii) the syntactical transformation of values, and (iv) the combination of values from different places.

Let us give an example to highlight the four tasks of the harvester. We focus here on data for the sport-business company *adidas* that were obtained on the 6th of January, 2012:

```
dax:DE000A1EWWW0_1325848842055
dax:endOfBusinessYear
"--12-31"^^xsd:gMonthDay
"2012-01-06T12:20:42"^^xsd:dateTime
"2012-01-06T12:20:42"^^xsd:dateTime .

dax:DE000A1EWWW0_1325848842055
dax:totalCapitalStock
"209216186EUR"^^xsd:monetary
```

Krieger H., Declerck T. and Nedunchezhian A..
MFO—The Federated Financial Ontology for the MONNET Project.
DOI: 10.5220/0004107903270330
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 327-330
ISBN: 978-989-8565-30-3

```
  "2012-01-06T12:20:42"^^xsd:dateTime
  "2012-01-06T12:20:42"^^xsd:dateTime .
```

For better readability, we have depicted the two quintuples using five consecutive lines each. `dax:DE000A1EWWW0_1325848842055` is a brand new URI for *adidas* at the moment the harvester was invoked, generated from the so-called ISIN number `dax:DE000A1EWWW0` for *adidas* and the Unix epoch time. `dax:endOfBusinessYear` and `dax:totalCapitalStock` are exactly the properties that are associated with the captions *Fiscal year end* and *Total stock* from *adidas*' Web page. The end of business year was originally given by `31/12`, whereas we make use of the XSD data type `gMonthDay`, yielding the XSD atom `"--12-31"`. The total stock value results from a combination of an absolute value, a multiplication factor of 1,000, and a currency abbreviation, resulting in `"209216186EUR"`, an atom of our own XSD data type `monetary`. Finally, the starting and ending points of these two relational fluents make use of the XSD data type `dateTime`. Since the snapshot was obtained in a moment of time, the starting and ending time is the same here.

3 ONTOLOGIES

Even though the ABox data, i.e., the company snapshots, do come with a temporal extent, the ontologies, more exactly, the TBoxes and RBoxes, which provide class and property axioms are **not** equipped with temporal information, thus still being represented as triples. For instance, we do **not** state that an URI is a class at a certain time and a property at a different time. Or that a class is a subclass of another class for only some amount of time. Thus TBox and RBox of the integrated ontology represent universal knowledge that is true at any time, so there is no need to equip them with a fourth and fifth temporal argument. This quality gives rise to the use of ontology editors such as Protégé for manually constructing the TBoxes and RBoxes of some of our ontologies.

Originally, we have started with company snapshots from the DAX index, resulting in the *DAX* ontology which has turned out to be universal for the other Xetra indices. Later then, this ontology was extended in two ways: *firstly*, we incorporated axioms to store parts of the annual XBRL company reports, and *secondly*, we imported our own OWL version of the NACE taxonomy for characterizing companies against a classification of industry sectors.

At a later stage, further information came in, so that we opted to separate independent information from one another, not only by introducing different namespaces, but also by locating this information in distinct ontologies, so that they can be *reused* by other projects and applications.

It is worth noting that the classification of companies against industry sectors is of utmost importance in our domain. We have thus decided to view industry sectors as *subclasses* of the class `Company`, both in the *DAX* and *Euronext* ontology. Overall, we now provide three, partly overlapping industry sector ontologies:

1. *DAX* comes up with a coarse and flat string-based characterization of industry sectors. We have manually turned these 11 values into direct subclasses of class `dax:Company`. In early 2012, *Deutsche Börse* restructured their pages, adding more sectors and a further layer of subsectors.
2. In an older project, we have worked with the four level deep NACE classification which covers about 1,000 industry sectors. We have automatically constructed our own OWL ontology *NACE* that is hooked up to `dax:Company` in the interface ontology *IF* (see below).
3. *Euronext* makes use of the four-layered ICB industry sector classification. We have automatically transformed the latest ICB specification of approx. 200 sectors into an OWL ontology *ICB* which is interfaced as a subclass hierarchy for `en:Company` in *IF*.

Overall, **MFO** consists of **nine**, truly independent ontologies which do **not** have knowledge of one another (see Figure 1). **Two** further ontologies, called *IF* and *XEBR2XBRL* bring them together through the use of interface axioms, using axiom constructors, such as `rdfs:subClassOf` and `owl:equivalentProperty`.

Let us give a few examples to see how this works. As we said above, we have interfaced *NACE* with *DAX*. This is realized through the following obvious axiom (`IndustrySector` is the synthetic superclass of all NACE sectors). We use Description Logics syntax below to state the interface axioms:

$$\mathsf{dax{:}Company} \equiv \mathsf{nace{:}IndustrySector}$$

Let us present further examples. E.g., an XEBR report is a special form of a Dublin Core resource:

$$\mathsf{xebr{:}Report} \sqsubseteq \mathsf{dc{:}Resource}$$

The free-text company information in *DAX* corresponds to something similar in *Euronext*:

$$\mathsf{dax{:}portrait} \equiv \mathsf{en{:}activity}$$

XEBR reports are linked to DAX companies through property `if:hasReport`:

$$\top \sqsubseteq \forall \mathsf{if{:}hasReport}^{-} . \mathsf{dax{:}Company}$$

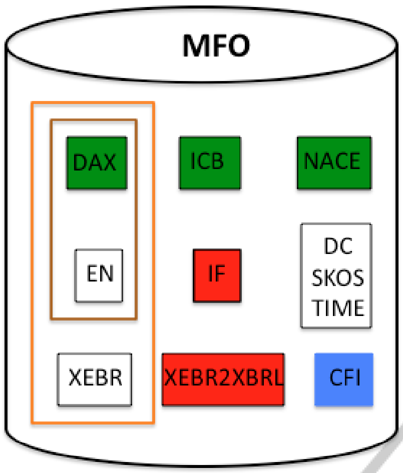

Figure 1: The MONNET Financial ontology MFO consists of 11 sub-ontologies overall. The color encoding refers to ontologies focussing on models of *industry sector classification* (**green**), *stock exchange* (**brown**), *reporting* (**orange**), *financial instruments* (**blue**), and *interface* (**red**). As can be seen from the picture, some of the ontologies even model several aspects of our domain; e.g., *DAX* alone deals with industry sector classification, reporting, and the description of stock exchange-listed information.

$$\top \sqsubseteq \forall \text{if:hasReport} \,.\, \text{xebr:Report}$$

It is worth noting that across the ontologies, each property has been cross-classified as being either *synchronic* (i.e., property instances staying constant) or *diachronic* (changing over time). This property characteristic can be used, amongst other things, to check the consistency of a temporal ABox or as a distinguishing mark in an entailment rule (cf. section 5).

4 SEMANTIC REPOSITORY

The integrated ontology as well as ABox data from company snapshots, XEBR reports (annual financial reporting documents) and other sources are uploaded to OpenLink's Virtuoso, hosted by one of our partners in this projects. No inference rules are applied here at the moment, thus only explicit knowledge can be obtained through SPARQL (Prud'hommeaux and Seaborne, 2008) queries. This can often be sufficient, since SPARQL is a quite expressive query language.

Attentive readers of this paper, however, will ask themselves how this goes together with the representation of company snapshot data, as explained in section 2. We demonstrated that snapshot data is encoded via quintuples, whereas ordinary semantic repositories (such as Virtuoso or OWLIM) always assume a triple-based representation.

In order to make the snapshots accessible in Virtuoso, we perform a quintuple-to-triple conversion which is compatible with W3C's *N-ary Relations* Best Practice proposal (Hayes and Welty, 2006). A description of further possible representation schemes can be found in (Krieger et al., 2008).

The idea behind the reduction is quite simple: all arguments lying in the range of a relation instance are hidden in a "container" object. The hidden arguments, in our case the actual value of the atemporal binary fact, the starting and the ending time can be obtained through pre-defined properties. Thus a quintuple

```
subj pred obj start end .
```

might equivalently be represented through 5 RDF triples:

```
subj pred cont .
cont rdf:type nary:RangePlusTime .
cont nary:value obj .
cont nary:starts start .
cont nary:ends end .
```

Note that `cont`, the container object, is a brand-new individual, usually an RDF blank node, that needs to be introduced for each quintuple.

Given such a representation, we can now query useful information, say, the evolution of the *total capital stock* for *adidas*, the company on which we focussed in section 2:

```
SELECT ?v ?s
WHERE {
  ?c dax:isin ?i .
  ?i nary:value "DE000A1EWWW0" .
  ?c dax:totalCapitalStock ?t .
  ?t nary:value ?v .
  ?t nary:starts ?s .
}
```

5 ENTAILMENT RULES

Due to space requirements, we can not focus here on the temporal extension of the set of domain-independent entailment rules for RDFS (Hayes, 2004) and OWL (ter Horst, 2005); see (Krieger, 2012) for the complete picture. The extension becomes necessary, since the harvested ABox data presented in section 2 is equipped with a temporal extent. Not only do such domain-independent rules uncover inconsistencies, they also make implicit consistent knowledge explicit.

Before we go on, we like to mention that the temporal entailment rules in (Krieger, 2012) as well as the custom rule and the query below have been fully implemented within our own semantic repository *HFC*, an extended rule-based forward chainer that we have developed over the last years and which is comparable to other popular engines. Rules in *HFC* consist of a left-hand side and a right-hand side of clauses, separated by `->`. A sequence of clauses is interpreted conjunctively. Rules can make use of LHS tests (`@test`) which need to be fulfilled to successfully instantiate a RHS. Rules might also be equipped with an action section (`@action`) that binds RHS-only variables to values returned by functions. These two last points distinguish *HFC* from other forward engines, and exactly further lightweight tests and actions are needed to address temporal entailment properly (see section 5.2). *HFC* is used in the working examples below. Contrary to Jena, OWLIM, or Virtuoso, *HFC* is furthermore able to operate directly over arbitrary (flat) tuples, i.e., *n*-ary relations, without sticking to container individuals (see last section).

5.1 Domain-independent Rule

The rule below implements the temporal extension of Pat Hayes' version of universal instantiation in RDFS, called rdfs9 (Hayes, 2004):

```
?i rdf:type ?c ?start ?end
?c rdfs:subClassOf ?d
->
?i rdf:type ?d ?start ?end
```

Names starting with `?` in the above rule indicate logic variables. Recall that the TBox axiom pattern (second line) must **not** be equipped with time, since it is used to express a universal truth in our ontology. Thus, if an instance `?i` is of class `?c` within the temporal interval given by `?start` and `?end`, then `?i` is also of class `?d` for [`?start`, `?end`], since `?c` is a subclass of `?d`. However, *before* and *after* [`?start`, `?end`], it is **no** longer guaranteed that `?i` is also of class `?d`.

5.2 Domain-dependent Rule

This rule turns two quintuples which coincide in subject, predicate, and object position and which share a *non-empty* temporal intersection into a larger unit. Since only diachronic properties (see section 3) are supposed to change over time, we add a further typing constraint here. Note that the below *HFC* rule even *quantifies* over the property position `?p` and uses a lightweight LHS test and two lightweight RHS actions:

```
?p rdf:type time:DiachronicProperty
?c ?p ?v ?s1 ?e1
?c ?p ?v ?s2 ?e2
->
?c ?p ?v ?s ?e
@test
IntersectionNotEmpty ?s1 ?e1 ?s2 ?e2
@action
?s = Min2 ?s1 ?s2
?e = Max2 ?e1 ?e2
```

5.3 Custom Query

The mapping of industry sectors from different sub-ontologies, as described in section 3, allows us to find "rivals" of a company across stock exchanges. Assuming that `dax:Banks`, `icb:ICB8300`, and `nace:nace_64.1` all denote the same set of individuals (viz., banks/financial institutions), the rivals of *Deutsche Bank* can be easily obtained:

```
SELECT DISTINCT ?rival
WHERE ?db dax:name "Deutsche Bank" ?s ?e &
      ?db rdf:type ?type ?s ?e &
      ?rival rdf:type ?type ?s2 ?e2
FILTER ?db != ?rival
```

We note here that the query language in *HFC* slightly differs from SPARQL and can make direct use of quintuples, instead of applying the "triplification" step, as described in section 4.

ACKNOWLEDGEMENTS

The research described here has been financed by the European Integrated project **Monnet** (www.monnet-project.eu) under contract number FP7 ICT 248458.

REFERENCES

Hayes, P. (2004). RDF semantics. Technical report, W3C.

Hayes, P. and Welty, C. (2006). Defining N-ary relations on the semantic web. Technical report, W3C.

Krieger, H.-U. (2012). A temporal extension of the Hayes/ter Horst entailment rules and an alternative to W3C's n-ary relations. In *Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS 2012)*.

Krieger, H.-U., Kiefer, B., and Declerck, T. (2008). A framework for temporal representation and reasoning in business intelligence applications. In *AAAI 2008 Spring Symposium on* AI Meets Business Rules and Process Management, pages 59–70. AAAI.

McGuinness, D. L. and van Harmelen, F. (2004). OWL Web Ontology Language Overview. Technical report, W3C.

Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL query language for RDF. Technical report, W3C. W3C Recommendation.

ter Horst, H. J. (2005). Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *Journal of Web Semantics*, 3:79–115.