

Context Mediation in the Semantic Web: Handling OWL Ontology and Data Disparity through Context Interchange

Philip Tan¹, Stuart Madnick^{1,2}, Kian-Lee Tan^{1,3}

¹Singapore-MIT Alliance, ²MIT Sloan School of Management,
³Department of Computer Science, National University of Singapore

Abstract The COntext INterchange (COIN) strategy is an approach to solving the problem of interoperability of semantically heterogeneous data sources through context mediation. COIN has used its own notation and syntax for representing ontologies. More recently, the OWL Web Ontology Language is becoming established as the W3C recommended ontology language. We propose the use of the COIN strategy to solve context disparity and ontology interoperability problems in the emerging Semantic Web – both at the ontology level and at the data level. In conjunction with this, we propose a version of the COIN ontology model that uses OWL and the emerging rules interchange language, RuleML.

Index Terms — Context Interchange, Interoperable, OWL Web Ontology Language, RuleML, Semantic Web.

I. INTRODUCTION

The COntext INterchange (COIN) strategy [10] is a mediator-based approach for achieving semantic interoperability among heterogeneous data sources and receivers. As realizations of the strategy, COIN [7] and eCOIN, a recent extension, [6] are two working prototypes that implement the Context Interchange strategy. eCOIN uses FOL/Prolog as the representation and implementation language for the application ontology in the context mediation process. Various sample applications have since been implemented to illustrate its ability to solve semantic interoperability problems in areas such as financial services, weather information, and airfare aggregation and comparison.

One of the core ideas of the Semantic Web is the ability to associate machine understandable meanings to information. A taxonomy, or ontology, is used to enhance the quality of data and information available on the Web, subsequently enhance the functioning of the Web in improving Web searches, relating information by inference rules and complicated query answering [1].

With various active independent ontology development activities around the world, the age-old problem of heterogeneous data interoperability also manifests itself in

the ontology area. One way to minimize the extent of ontology heterogeneity and disparity is to create a controlled and centralized ontology collection, with the goal to minimize duplication and incompatibility of ontology. However, with decentralized knowledge engineering and ontology development widely implemented in the industry and academic, the problem of ontology disparity is unavoidable. The full potential of ontology and language standardization using OWL will only be realized if they are used in combination with other ontologies in the future to enable data sharing [8].

In fact, W3C recognizes the existence of such problem – “We want simple assertions about class membership to have broad and useful implications. ...It will be challenging to merge a collection of ontologies.” [11].

OWL provides a number of standard languages construct that aims at solving a subset of this problem. Ontology mapping constructs such as `equivalentClass`, `equivalentProperty`, `sameAs`, `differentFrom` and `AllDifferent` only allow ontology context consolidation at a very limited level. These language constructs are only useful if the consolidation effort requires only disambiguation between ontology. In other words, we can use these facilities to tell that a human in ontology A is the same as person in ontology B, but if they are different, we will not be able to tell how different these two classes are; needless to say that limits interoperability between the two ontologies.

A Our Contributions

Our goal in this paper is to illustrate the novel features of the Context Interchange mediation strategy in solving ontology disparity problem in Semantic Web. Even though this research originated from a long-standing research in the data integration area, the use of this strategy in handling ontology interoperability presented in this paper is new with respect to our previous works and other relevant work in this area. In conjunction with this, we present a new COIN ontology representation model using OWL and RuleML, in alignment to the new and emerging W3C standards.

B Related Work

One relevant effort in the Semantic Web/OWL space is Context OWL (C-OWL) [3], a language whose syntax and semantics have been obtained by extending the OWL syntax and semantics to allow for the representation of contextual ontologies. However, the extension focused on limited context mapping using a set of bridge rules that specify the relationship between contexts as one of the following: equivalent, onto (superset), into (subset), compatible, incompatible. The limited expressiveness of the language fails to address the contextual differences such as those possible with COIN.

On standardization of the COIN ontology representation, Lee [9] has presented a XML-based metadata representation for the COIN framework. The essence of that work lies in modeling and storing of the metadata in RDF format as the base format. A number of intermediate representations of were proposed: RDF, RuleML, RFML and the native Prolog representation used in COIN. The core ontological model of COIN in RDF format is transformed into the aforementioned intermediate representation by applying Extensible Stylesheet Language Transformation (XSLT) on the fly. Context mediation for heterogeneous data is then executed using the ontological model encoded in the COIN language. It is worth noting that the approach proposed in this work primarily deals with a single representation at a time. The intermediate ontological model is represented in RDF, RuleML or RFML individually, but not as a combination of the different formats, which is the approach taken in our approach.

The rest of the paper is organized as follows. After this introduction, we present a motivational example to highlight the Context Interchange strategy in handling ontology disparity problem. Section III describes the building blocks of the Context Interchange strategy. Section IV details the COIN-OWL ontology model design, design considerations and limitation. The final section presents a summary of our contributions and describes some ongoing research and future research directions.

II. CONTEXT INTERCHANGE IN ACTION

One of the easiest ways to understand the Context Interchange framework is via a concrete example. Consider two financial data sources: Worldscope (worldscope) and Disclosure Corporate Snapshot (disclosure) as shown in Figure 1.

Worldscope provides basic financial information on public companies worldwide, while Disclosure is an information directory on companies publicly traded on U.S. exchanges. Worldscope reports all the financial data information in US dollars, and on the scale factor of 1000, while disclosure reports the financial data information in the local currency of the companies, and on the scale factor of 1.

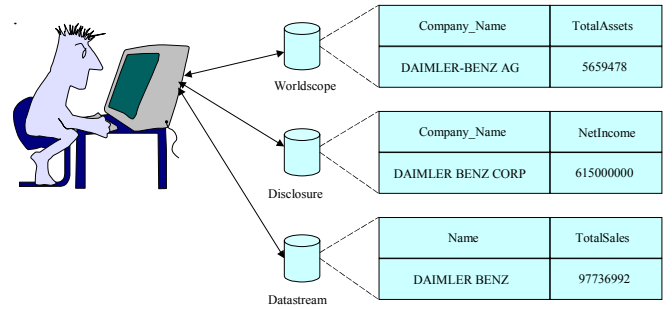


Fig. 1. Multiple databases with similar data, but differing contexts

Using these financial data sources, users are able to post queries on the public companies of interest. For example, to retrieve the asset data of Daimler-Benz AG from the worldscope database, the user may issue the following SQL query:

```
select Worldscope.TotalAssets
from Worldscope
where Worldscope.Company_Name =
"DAIMLER-BENZ AG";
```

On the other hand, to retrieve net income data from disclosure, the following SQL query can be used:

```
select
Disclosure.Date,Disclosure.NetIncome
from Disclosure
where Disclosure.Company_Name =
"DAIMLER BENZ CORP"
and Disclosure.Date = "12/31/93";
```

Although the data can be retrieved from both sources, there are important discrepancies in the data due to the difference in context of the data sources, both in the currencies and the scale factors used (as well as company naming conventions and date formats). Thus, if one wanted to retrieve the TotalAssets from Worldscope and the NetIncome from Disclosure, the results could be confusing since the results would be provided with these context differences.

In a conventional database system, to perform a join table query between Worldscope and Disclosure, these context disparities would have to be resolved manually and encoded in the SQL query. Using COIN, these context discrepancies (different company name format, date format, financial data currency type and scale factor) are mediated automatically and queries such as the following can be used without the user having to know anything about the actual contexts of the sources (the results will be returned to the user in the context defined for the user, independent of the contexts of the sources):

```
select Disclosure.Date,
Worldscope.TotalAssets,
Disclosure.NetIncome
from Disclosure, Worldscope
where Disclosure.Company_Name =
"DAIMLER BENZ CORP"
and Disclosure.Company_name =
Worldscope.Company_Name
and Disclosure.Date = "12/31/93";
```

This automated context reasoning and mediation capability is the essence of the Context Interchange strategy. Using the same context reasoning and mediation engine, ontology interoperability is achieved by defining meta-ontology that describes the disparate ontologies. This is discussed in the subsequent section.

III. CONTEXT INTERCHANGE STRATEGY ESSENTIALS

The Context Interchange framework employs a hybrid of the loosely- and tightly-coupled approaches in data integration in heterogeneous data environment. The COIN framework was first formalized by Goh et. al in [7] and further realized by Firat [6]. The Framework comprises three major components:

- The domain model, which is a collection of rich types, called semantic types. The domain model provides a lexicon of types, attributes and modifiers to each semantic type. These semantic types together define the application domain corresponding to the data sources which are to be integrated.
- The elevation theory, made up of elevation axioms which define the mapping between the data types of the data source and the semantic types in the domain model. Essentially, this maps the primitive types from the data source to the rich semantic types in the application domain.
- The context theory comprising declarative statements which either provide for the assignment of a value to a context modifier, or identify a conversion function which can be used as the basis for converting the values of objects across different contexts.

These three components form the complete description of the application domain, required for the context mediation procedure as described in [5].

Due to space constraints, we limit the details of the COIN strategy. For detailed theoretic formalism and implementation details, the readers are referred to the literatures [4, 5, 6, 7, 10].

A Context Interchange and Ontology Interoperability

One major perspective the Context Interchange strategy employs is the relational view of the data. Semi-structured data, including information from HTML and XML web pages can be used in the prototype via the Cameleon web wrapper engine [6]. This aspect of the strategy is one distinct area that sets itself apart from the common usage of OWL, where ontology and data are often maintained together in the semi-structured format of OWL.

Intuitively, the use of OWL in COIN can be viewed as the meta-ontology layer on top of OWL, providing an extension to OWL to support context-aware ontology to the current context-oblivious ontology in OWL.

Our approach in solving the ontology interoperability problem is by applying the COntext Interchange strategy at

the ontology level, treating disparate ontologies as the subjects to be mediated. This can be done by creating an application meta-ontology describing the contexts of the ontologies. Using this application meta-ontology, the contextual difference can be solved in the same way that the semantic interoperability problem of heterogeneous databases is solved using COIN. With this, we can integrate and interoperate among the disparate ontologies, and subsequently integrate the underlying data represented by these ontologies.

Additionally, the same approach can be used to mediate not only data sources in the Semantic Web, but also traditional relational databases. This is important since it is expected that relational databases will co-exist with the new Semantic Web paradigm. This extension of the Context Interchange strategy will be capable of handling data interoperability within Semantic Web data sources, traditional database sources, as well as interoperability between the Semantic Web data sources and traditional databases.

IV. COIN-OWL ONTOLOGY MODEL

Prior to describing the COIN-OWL ontology model design, we explain in brief the OWL Web Ontology Language and Rule Markup Language (RuleML)

A OWL Web Ontology Language

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full:

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies.
- OWL DL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with the field of Description Logic.
- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It would be difficult for reasoning

software to support complete reasoning for every feature of OWL Full.

A more detailed comparison of the sublanguages is available from [11].

B Rule Markup Language (RuleML)

The RuleML Initiative is a collaboration with the objective of providing a basis for an integrated rule-markup approach. This is achieved by having all participants collaborate in establishing translations between existing tag sets and in converging on a shared rule-markup language. The main goal for the RuleML kernel language is to be utilized as a specification for immediate rule interchange.

Rules can be stated (1) in natural language, (2) in some formal notation, or (3) in a combination of both. Being in the third, 'semiformal' category, the RuleML Initiative is working towards an XML-based markup language that permits Web-based rule storage, interchange, retrieval, and firing/application.

The XML schema definition of RuleML can be viewed as syntactically characterizing certain semantic expressiveness subclasses of the language. As eCOIN represents the ontological model in Prolog, which is in the horn-logic family, our use of RuleML is focused on the datalog and hornlog sublanguage. These two sublanguages provide a comprehensive language facility in describing rules encoded in Prolog. As the application ontologies in COIN may involve complex rules, our design and implementation uses both the datalog and hornlog sublanguages.

C Context Interchange Ontology Model in OWL (COIN-OWL)

Approach. In eCOIN, the FOL/Prolog program formed by the collection of domain model definitions, elevation theories and context theories is used to detect and mediate context disparity and heterogeneity in a query using an abductive procedure defined in [12]. One important principle of our work is to preserve this constraint programming engine in the COIN framework.

We adopt a layered architecture, as shown in Figure 2, in the use of OWL in context interchange framework: (1) the domain ontology will be modeled in OWL (and its extension or relevant technology), (2) the ontology will be transformed to eCOIN FOL/Prolog as the native representation of the domain, and finally, (3) the native program will be taken as input to the abductive engine for context mediation.

The OWL ontology model can be viewed as the front-end of the system, where it is the main interfacing layer to the ontology administrator of the eCOIN system. In the intermediate layer, the transformation from OWL to the native FOL/Prolog program will be transparent and automatic. The transformation process is detailed in the later section of the thesis. With the derived program in its native

FOL/Prolog format, the existing mediation engine can be reused in its entirety.

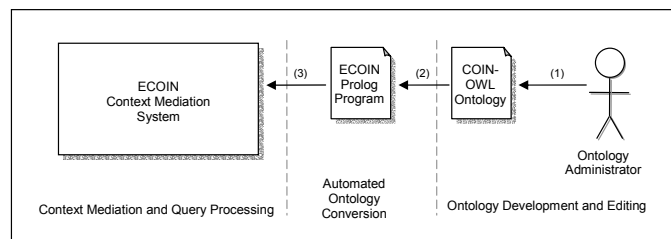


Fig. 2. Three-tier approach for Context Interchange ontology modeling using OWL

The big win of this approach is that it minimizes re-work: there is little value in reinventing the wheel, especially when the current functionality of the system provides the total capability currently required. At the same time, the abstraction provided by the middle tier of the architecture shields the ontology administrator from the actual implementation of the COIN context mediator. This componentization fulfills our aim of adoption of OWL in the framework, yet ensuring minimal impact to the existing COIN system.

The conversion from the OWL version of a COIN domain model to its Prolog version is done using Protégé OWL API, while conversion from RuleML to Prolog is done by using XSL Transformation technology, through the use of eXtensible Stylesheet.

OWL and Rule-based Ontology. One major challenge of the adoption of OWL in the ontology model is that the COIN ontology model encompasses a number of constructs that are not directly available in OWL. Constructs such as Domain Model and Elevation Axioms can be represented in OWL rather easily – conceptually, these constructs describe the relationships among the data types, and can be modeled accordingly using corresponding constructs in OWL that express relationships among classes.

The problem, however, lies in the modeling of context theory, which is the pivotal component in the COIN framework. The collection of context axioms in a context theory is used either to provide for the assignment of a value to a modifier, or identify a conversion function, which can be used as the basis for converting the values of objects across different contexts. Often, the expressiveness of rules is required to define the conversion of a semantic type in the source context to a different context.

In our proposed design, axioms requiring such flexibility are encoded in RuleML. RuleML allows rule-based facts and queries to be expressed in the manner similar to conventional rule language such as Prolog. The concrete representation of RuleML is XML, which fits seamlessly in our effort to standardize the ontology representation in eCOIN.

We chose to use RuleML because it has received significant support and participation from academia and industry in the RuleML working group and it is likely that

RuleML may eventually be accepted as part of the W3C standard for Rule-based ontology in Semantic Web. The early adoption of such emerging W3C standard promotes standardization of our effort and allows our work to be re-used by other interested parties in the Semantic Web and data/context integration space.

D COIN-OWL Ontology Model Design

In this section, we examine the modeling of the COIN ontology in OWL with respect to domain model, elevation theory and context theory. The COIN ontology (expressed in OWL) can be used as a base OWL ontology to model disparate data sources for the purpose of data integration by means of context mediation. Where appropriate, the concrete XML presentation of the model is presented to illustrate the proposed implementation of the model.

Domain Model. By definition, the domain model defines the taxonomy of the domain in terms of the available semantic types and modifiers to each semantic types. In addition, the notion of primitive type is used to represent the data types that are native to the source or receiver context.

OWL uses the facilities of XML Schema Datatypes and a subset of the XML Schema datatypes as its standard datatypes (or equivalently, its primitive datatypes). On the other hand, the primitive types in the COIN language consist of string and number. Trivially, the COIN datatypes can be represented using its counterparts in OWL, namely xsd:string and xsd:int, xsd:float or xsd:double.

Source Sets. This COIN concept, the intensional description of the data sources, is not directly available in OWL, as OWL is used as the descriptive language only for semi-structured data on the Web. COIN, on the other hand, is designed to deal with a wide range of data sources, which makes the declarative description of the data sources indispensable for data integration and context mediation.

Context Axioms. A core concept in COIN is the notion of context differences and the ability to interoperate among contexts through context mediation. The fundamental component to context axioms is the definition of context itself.

Context definition: The interpretation of a semantic object value that is decorated by modifiers may vary according to the values taken by the modifier (e.g., the semantic object “TotalAssets” – or the more generic “monetary unit” - might be in US dollars or Euros). The value of the modifier is determined by prior domain knowledge, dependent on the context of the domain. This value can either be static (e.g., monetary units are always US dollar in Worldscope context), or dynamically obtained from other attributes (e.g., monetary units are in the currency of their country). This hierarchical structure translates to the need of modeling a parent ModifierValue

class, with two subclasses ModifierStaticValue and ModifierDynamicValue.

Conversion function: A more complex construct available in COIN is the conversion function. In essence, conversion functions enable interoperability of semantic objects across different contexts. This is achieved by defining generic conversion rules for each semantic type that may yield different value under different contexts.

This requirement calls for a language facility that is both flexible and supports rule-based data. However, OWL lacks the ability to model rules in an extensible manner. Therefore, we used RuleML for conversion function modeling. As an example, consider the simple conversion function in eCOIN’s Prolog representation, that converts the month expressed as a 3-letter abbreviation into its corresponding numeric value (and vice versa):

```
rule(month("Jan", 01), (true)).
```

This rule can be represented using RuleML as follows:

```
<fact>
  <_head>
    <atom>
      <cterm>

<_opc><ctor>rule</ctor></_opc>
  <cterm>

<_opc><ctor>month</ctor></_opc>
  <ind>Jan</ind>
  <ind>01</ind>
  </cterm>
  <ind>true</ind>
  </cterm>
</atom>
</_head>
</fact>
```

Elevation Axioms. Elevation axioms are used to describe the functional relationship between data sources and domain model. Intuitively, the elevation axioms can be viewed as the mapping of the primitive relation to its semantic relation. At the lower level, each column and data cell are mapped to their semantic counter part via skolemization.

Complete Ontology Model. Combining the previous individual elements of the ontology model, we present the complete COIN-OWL ontology model in the form of the UML class diagram in Figure 3. Each of the major ontology elements are shaded in gray grouping for clarity.

E. Design Considerations

One of the objectives of our design is to adopt emerging W3C standards as the data exchange standard in the Context Interchange project while reusing the established context mediation strategy and implementation in the project. This means that the proposed COIN model in OWL must be able to be translated to FOL/Prolog for actual context mediation and query execution process. This guiding principal is crucial in ensuring the practicality of the proposed model.

Choice of OWL Sublanguage. As introduced in the earlier section, OWL is classified into three language family: OWL Lite, OWL DL and OWL Full. The OWL sub-language used in our design is OWL Lite, as this family of language is sufficiently expressive to represent the COIN ontology model.

With our three-tier architecture, the actual reasoning and context mediation is performed at the backend (see Figure 2). This essentially means that the computation guarantee of OWL Lite and OWL DL is not required. In other words, we have the liberty to use any of these three classes of OWL sublanguages.

However, OWL Lite contains the language constructs that are rich enough for this purpose. One reason for not pushing to use the upper language family of OWL DL and OWL Full is to preserve the computability of the ontology for future. This allows the reasoning and context mediation, should there be a need in the future, to be performed directly at the OWL level without having to first translate the OWL ontology to the native ECOIN Prolog application.

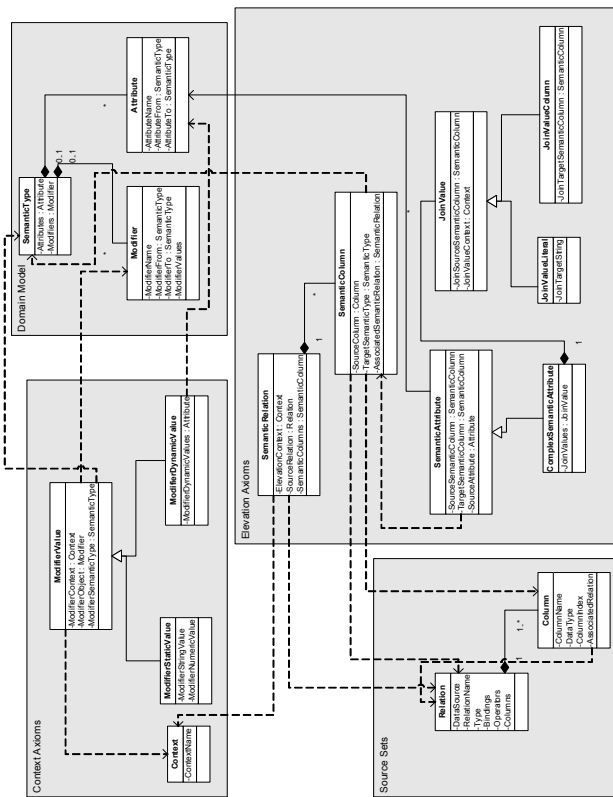


Fig. 3. UML class diagram of the complete COIN-OWL model

OWL Ontology and Data. As part of the design and operation of COIN, we have a slightly different usage adoption of OWL. In the standard usage of OWL for ontology modeling, the ontology and data are both stored in OWL. Depending on the generality of the taxonomy definition, the ontology and data may co-exist on the same OWL document. In other cases, the ontology is defined and stored in a central OWL ontology library, and referenced in

the OWL data document using external namespace reference. An example of such usage is the OWL Wine ontology (at <http://www.w3.org/TR/2002/WD-owl-guide-20021104/wine.owl>), where both the ontology definition and the individual instantiation (i.e. actual data) are stored in the same OWL document. On the other hand, COIN utilizes the application ontology in a different manner. The COIN-OWL ontology model describes the context semantics of the data sources. Modeled in OWL, this ontology is then used by the context mediation engine to resolve context disparities among the data sources.

While the COIN ontology is modeled in OWL, the actual data may not necessarily be stored in OWL. This is because by design, COIN is architected to solve the heterogeneous data source interoperability problem. This means that the data to be reconciled by COIN will be from disparate data sources, comprising traditional relational databases or traditional semi-structured data sources on the World Wide Web (in XML or HTML) or even OWL.

Static Type Checking. One of the biggest differences between modeling the ontology in eCOIN and COIN-OWL is the strongly enforced typing facility in OWL. In OWL, all ObjectProperty and DataProperty requires the formal definition of the range of the property, i.e. the type of object that can be specified in property.

As an example, in eCOIN, we model semantic types and modifiers using the following constructs:

```
rule(semanticType(companyName), (true)).
rule(semanticType(companyFinancials), (true)).
rule(modifiers(companyFinancials, [scaleFactor, currency]), (true)).
```

Here, it is possible for someone to accidentally put `companyName` as the modifier for `companyFinancials`:

```
rule(semanticType(companyName), (true)).
rule(semanticType(companyFinancials), (true)).
rule(modifiers(companyFinancials, [companyName]), (true)).
```

However, as all classes are strongly typed in OWL, the following ontology will yield an error when validated against the COIN ontology:

```
<coin:SemanticType
rdf:ID="companyName" />
<coin:SemanticType
rdf:ID="companyFinancials">
  <coin:Modifiers
rdf:resource="#companyName">
</coin:SemanticType>
```

Functional Property. In all flavors of OWL (OWL Lite, OWL DL and OWL Full), a property P of object X can be tagged as functional such that for objects Y and Z, X.P=Y and X.P=Z implies Y=Z. property P of object X is denoted as X.P.

In other words, object X can functionally determine Y in $X.P=Y$. Using this language feature, we can enforce a many-to-one relationship between classes. Given the wide array of language features in OWL, this is particularly useful in enforcing syntactically and semantically correct COIN ontology.

As an additional note, such requirements can also be enforced using the owl:cardinality construct. However, it is worth noting that the use of this construct depends on the sublanguage family of OWL. Cardinality expressions with values limited to 0 or 1 are part of OWL Lite. This permits the user to indicate 'at least one', 'no more than one', and 'exactly one'. Positive integer values other than 0 and 1 are permitted in OWL DL. owl:maxCardinality can be used to specify an upper bound. owl:minCardinality can be used to specify a lower bound. In combination, the two can be used to limit the property's cardinality to a numeric interval.

RuleML for Rules Modeling. In the previous work in [9], RDF was used to model the COIN ontology model. However, the work was unable to address the need for a more extensible framework in rules representation. In particulars, conversion rules were encoded as raw string in the RDF document:

```
<coin:Ont_ModifierConversionFunction>
convfunc|rule(cvt(companyFinancials,
O, currency, Ctxt, Mvs, Vs, Mvt, Vt),
(attr(O, fyEnding, FyDate),
value(FyDate, Ctxt, DateValue),
olsen_p(Fc, Tc, Rate, TxnDate),
value(Fc, Ctxt, Mvs), value(Tc, Ctxt,
Mvt), value(TxnDate, Ctxt, DateValue),
value(Rate, Ctxt, Rv), Vt is Vs * Rv)).
...
rule(month("Oct", 10),
(true)).
rule(month("Dec", 12),
true)). |companyFinancials|currency
</coin:Ont_ModifierConversionFunction>
```

These rules were then extracted programmatically from the RDF document and used in context mediation. In comparison, the adoption of RuleML for rules modeling provided a cleaner method for this purpose. In COIN-OWL, these rules are stored as RuleML:

```
<rulebase>
<!-- rule(month("Apr", 04), (true)).
-->
<fact>
<_head>
<atom>
<cterm>
<_opc><ctor>rule</ctor></_opc>
<cterm>
```

```
<_opc><ctor>month</ctor></_opc>
<ind>Apr</ind>
<ind>04</ind>
</cterm>
<ind>>true</ind>
</cterm>
</atom>
</_head>
</fact>
</rulebase>
```

While this format may look lengthier, this mode of representation adheres to the publicly accepted RuleML language constructs, and thus allow re-use and interchange of rules easily.

Semantic Web Rule Language (SWRL). During the course of our work, a number of relevant emerging standards have branched from RuleML, including RuleML Lite and Semantic Web Rule Language (SWRL). RuleML Lite adopts an integrated concrete syntax of XML and RDF, expanding the language construct available in modeling rules. This opens up the possibility of a tighter integration between the conversion rules in RuleML and the core ontology in OWL. One possibility is to refer to the entities modeled in the OWL ontology using rdf:resource or href attributes, instead of treating the same entity in both documents as individual and disjoint entities in each of the document.

SWRL has been considered but not implemented in this project as the modeling language is still in its very early stage. SWRL is the result of an effort to integrate RuleML into OWL, and hence holds a more holistic view of rules and ontology in the Semantic Web, compared to the use of OWL and RuleML separately.

From the following example, we note that the OWL ontology and RuleML rules are all modeled in one cohesive SWRL document. The rules fragment that expresses $x3$ hasSex male refers to the OWL class male seamlessly using the owl:Individual construct:

```
<swrlx:individualPropertyAtom
swrlx:property="hasSex">
<ruleml:var>x3</ruleml:var>
<owlx:Individual owlx:name="#male"
/>
</swrlx:individualPropertyAtom>
```

In RuleML 0.8, the RuleML version used in the current COIN-OWL ontology model, such language facility is not available. To refer to an individual defined in the OWL ontology, there is no other ways but to initialize a new individual in the RuleML rules document, hence creating a slight gap between the OWL ontology and RuleML rules.

V. CONCLUSION

In summary, we have presented an ontology interoperability framework based on the Context Interchange strategy. In conjunction with that, we proposed an ontology modeling approach using OWL and RuleML in

conjunction with the Context Interchange strategy. The COIN-OWL ontology model design is built on the building blocks of the OWL Lite sublanguage family and the Rule Markup Language, which are used to model the core ontology and the rule-based metadata in COIN, respectively. In relation to the ontology model, we have highlighted the design considerations, strengths and some of the limitations of the design.

With the growing adoption of OWL and the gradual realization of the Semantic Web vision, this work is instrumental in bridging the gap between COIN and Semantic Web. With this COIN-OWL model, it is hopeful that COIN will be able to reach a larger spectrum of audiences, and hence bringing even more contribution to the database/Semantic Web community in the area of heterogeneous data interoperability and ontology interoperability.

As part of the conclusion of our work, we would like to highlight some of the interesting and promising research areas. The use of the Context Interchange strategy in ontology interoperability and data sharing is an ongoing research work of our group. We are currently working on creating a fully working prototype of the OWL ontology interoperability framework discussed in the paper.

We also noted that in parallel with the development of RuleML, a number of relevant emerging standards have been proposed in the rules interchange community, including RuleML Lite and Semantic Web Rule Language (SWRL). As these standards mature, in particular SWRL, which combines OWL and RuleML, we see that such standards promise a more cohesive rule-based ontology model. One reservation on SWRL, however, is that it is based on the RuleML datalog sublanguage, where as the minimum requirement for our current implementation requires the hornlog sublanguage family for total compatibility with Prolog. These are issues that need further study.

VI. ACKNOWLEDGEMENTS

The authors acknowledge the help of Aykut Firat and Hongwei Zhu in reviewing drafts of this paper. The research reported herein has been supported, in part, by the Singapore-MIT Alliance (SMA).

REFERENCES

- [1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," in *Scientific American*, vol. 5, 2001, pp. 34-43.
- [2] H. Boley, "The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations," In Proceedings of the 14th International Conference of Applications of Prolog, 2001.
- [3] P. Bouquet, F. Giunchiglia, F. v. Harmelen, L. Serafini, and H. Stuckenschmidt, "C-OWL: Contextualizing Ontologies," In Proceedings of the Second International Semantic Web Conference, 2003.
- [4] S. Bressan, K. Fynn, C. H. Goh, S. E. Madnick, T. Pena, and M. D. Siegel, "Overview of a Prolog Implementation of the COnText INterchange Mediator," In Proceedings of the 5th International Conference and Exhibition on The Practical Applications of Prolog., 1997.
- [5] S. Bressan, C. H. Goh, T. Lee, S. E. Madnick, and M. Siegel, "A Procedure for Mediation of Queries to Sources in Disparate Contexts," In Proceedings of the International Logic Programming Symposium, Port Jefferson, N.Y., 1997.
- [6] A. Firat, "Information Integration Using Contextual Knowledge and Ontology Merging," Ph.D. Thesis, Massachusetts Institute of Technology, Sloan School of Management, 2003.
- [7] C. H. Goh, S. Bressan, S. Madnick, and M. Siegel, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," *ACM Transactions on Information Systems*, vol. 17, pp. 270-293, 1999.
- [8] H. Kim, "Predicting How Ontologies for the semantic Web Will Evolve," *Communications of the ACM*, vol. 45, pp. 48-54, 2002.
- [9] P. W. Lee, "Metadata Representation and Management for Context Mediation," Master Thesis, Massachusetts Institute of Technology, Sloan School of Management, 2003.
- [10] M. Siegel and S. Madnick, "A Metadata Approach to Resolving Semantic Conflicts," In Proceedings of the 17th Conference on Very Large Data Bases, 1991.
- [11] M. K. Smith, C. Welty, and D. L. McGuinness, "OWL Web Ontology Language Guide," 2003. <http://www.w3.org/TR/2003/PR-owl-guide-20031215>.