A Practical Exploration of Ontology interoperability with Conceptual Graphs for added expressivity in the Semantic Web

Ashima Shah , Richard Hill and Simon Polovina

Abstract. In the pursuit of adding expressivity to ontologies by the use of ISO Common Logic, we explore how Conceptual Graphs (CG) can enhance ontologies modeled in the Semantic Web (SW). Various strengths and weaknesses of SW's RDF/OWL are identified and considered how they be addressed by CG. We put these strengths and weaknesses to the test through a rich and topical business example in the form of a financial trading case study. We continue this practical vein by generating its ontology using the Protégé OWL tool and then use the CG CoGui Tool to elicit the comparative expressiveness as realised by these contemporary, representative software tools. In particular, the case study's OWL ontology is subjected to its business rules by means of the CG operations in CoGui. The work demonstrates how an OWL ontology can interoperate and in turn be enriched by CG tools in a practical, implementable way and from this experience stimulate industry interest in adding meaningful semantics to the SW to make it a mainstream business proposition.

1 Introduction and Motivation

Interoperability is one of the major challenges which needs to be addressed for successful implementation of the Semantic Web. Semantic Web includes languages like RDF and OWL, which are very well supported by W3C. However it is yet to be proved that the RDF and OWL can universally meet the requirements of complex business scenario. Semantic Web has strong resemblance to Artificial Intelligence (AI) but it is not artificial intelligence, however it can benefit from knowledge representation languages of AI. Conceptual Graphs (CG) is a formalism which has classes, relation, Individual and qualifiers. Meaning expressed by Conceptual Graphs is easily understood by humans, tractable by machines (agents, software or applications). CG is one of the key technologies in ISO Common Logic (CL) standard (ISO/IEC 24707:2007). ISO standard has helped technologies like CG to be used with other standards according to their strengths. In summary CL provides new opportunities to SW to achieve its vision of automation, integration and machine understandability.

Cook, Polovina and Loke (2009) have also asserted importance of Common Logic in the Semantic Web. They have demonstrated interoperability of ontologies for small scale experiments using CG tools like Amine and RDF/OWL tools like Protégé. They could demonstrate interoperability between CL and SW. We have taken their research forward , as We observed that most of the experiments performed for evaluating the mapping of RDF/OWL to CG were done in the earlier stages and were done using small scale examples. We took up this challenge of mapping OWL ontology to CG ontology for topical business example in the form of a financial trading case study.

We have evaluated CL standards against the strengths and weaknesses identified by Reynolds et al (2005). In order to demonstrate practical applicability of CG and its advantages in the business environment we made use of tools like CoGui which provide interface to import OWL ontology.

2 OWL and CG

2.1 Strengths and Weaknesses

Reynolds et al (2005) conducted an assessment of RDF and OWL modelling through experiment. In this experiment they identified various strength and weaknesses. Some of the important ones are listed here below.

2.1.1 Strengths

"Support for information integration and reuse of shared vocabularies"	Data merger is easily automated when sources share compatible vocabularies and resource identification mechanism. This is possible because • RDF uses URI to identify resources , relationships and concepts • OWL provides a mechanism to publish shared ontology • OWL's equality inference mechanism allow merger of data from overlapping views from common resources. OWL's provides mechanism to map classes and property between different vocabularies. This mapping process is not an automated process and is dependent on ontology alignment technologies.
"Handling of semi-structured data"	This is definitely a strong point of RDF/OWL because in a heterogeneous environment where all the users are allowed to create data/information, it is very much

	possible that data can be semi-structured.
"Separation of syntax from data modeling"	Tools are available which allow the developer to model data without having to worry about the syntax.
"Web embedding"	Since the Semantic Web is aimed at the extension of WWW, this is probably the strongest point of RDF/OWL. Ability to identify resources, concepts and relationships through URI enables users to publish vocabularies on the web; secondly it allows users to reuse existing concepts from other vocabularies.
"Extensibility and resilience to change"	This very strong point of RDF/OWL because RDF can handle semi-structure data; this increases the flexibility and loose coupling. RDF processor can ignore property which are present in the request and can continue to process the requests. Secondly change is inevitable and it is virtually impossible to keep upgrading all the systems as soon as something is changed, hence if an Old Processor can continue to process the request even if it does not have the definition of the new properties or does not have access to new definitions. More importantly RDF is the underlying language to extend the Ontologies; hence URI can be used to dynamically load the definitions.

Table 1.

2.1.2 Weaknesses

"Weak ability to validate documents"	In order to support semi-structured data and resilience to change in data, RDF/OWL had to sacrifice strict validation of the document. This is a genuine trade off to accommodate extendibility. However such issues will exist in almost all languages.
"Expressivity limitations"	• Cross-Slot constraints and operation: - It is difficult to define relation between two properties in OWL. For example it is not possible to define that value of property a1 should be greater than the value of property2. • Identity Criteria: - OWL language do not allow concept of composite key like ER modelling. This limitation makes it difficult to model correct domain model , however workaround exist which

	include adding unique identity constraints , such work around have impact on performance and business logic. • Property Composition: - OWL does not allow property definitions where one property is combination of other two properties. Such limitation make domain model more complicated. • Defaults: -OWL does not allow default values for a property.
“No built-in representation of processes and change”	RDF/OWL does not support the notion of time, state. While this is limitation in OWL/RDF other languages will also suffer from the same weaknesses

Table 2.

Reynolds et al (2005) paper was based on OWL, since 2005 various shortcomings were analyzed and OWL has been reengineered. In April 2008 Working Group decided to call OWL 1.1 as OWL 2. The OWL 2 was designed to address various shortcomings of OWL 1. The later version has (Grau et al 2008):

1. Increased language expressivity
2. Increased relational expressivity
3. Increased data type expressivity
4. More restrictive variant of keys which are relatively easy to implement.

OWL 2 has the new XML syntax which is easy to parse and process. These advancements of OWL 2 were aimed to resolve the expressivity issues so that it can be robust and effective for practical use.

2.2 Standard requirements of CL

Standardization of CL is an important step towards the making technologies like CG useful for SW. Brief summary of the standard is given below

ISO Standards	Evaluation against RDF/OWL and SW vision
First-order logic with equality.	CL is first-order logic with syntactic limitations removed and network use in mind. This lends lot of strengths to CL in comparison to RDF/OWL
Syntax for logical expressions	This improves expressivity
Single XML syntax	This improves interoperability and enhances data exchange.

The language should be able to express various commonly used logical forms for commonly used patterns of logical sentences.	This improves expressivity
It should be capable of rendering any content expressible in RDF, RDFS, or OWL.	This improves interoperability options for CL and RDF
It should support human-readable syntax	This meets one of the major requirements of SW vision.
It should be easy and natural for use on the Web	This is an important requirement for any internet based application or agents
The XML syntax should be compatible with the published specifications for XML, URI syntax, XML Schema, Unicode, and other conventions relevant to transmission of information on the Web.	This reduces the lexical and compatibility issues
It should support use of URIs and URI references.	This makes CL Ontologies as powerful as RDF/OWL ontology
It should be able to use URIs to give names to expressions and sets of expressions to support cross referencing.	This matches key strengths of RDF/OWL
It should support open networks	This is an important requirement for any internet based application or agents
No agent should be able to limit the ability of another agent to refer to any entity or to make assertions about any entity.	This meets one of the major requirements of SW vision.
The language should support ways to refer to a local universe of discourse and be able to relate it to other such universes.	This meets one of the major requirements of SW vision.
It should support new names and	This improves interoperability

use them in published Common Logic content.	
It should not make arbitrary assumptions about semantics.	This is an improvement over RDF/OWL weaknesses.

Table 3.

The above evaluation theoretically proves the advantages of CL (CG) in SW. As next step we evaluate the interoperability of OWL ontology and CG ontology using tools like Protégé and CoGui.

We put these strengths and weaknesses to the test in the form of a financial trading case study. We generated ontology using the Protégé OWL tool and then used the CG CoGui Tool to elicit the comparative expressiveness as realised by these contemporary, representative software tools.

3 Case study

In order to evaluate various strengths and weaknesses and test the interoperability of OWL ontology and CG Ontology, the below mentioned process was followed for the Case Study

Protégé

1. An ontology for the case study was generated in Protégé
2. Classes , Properties , Sub properties were constructed in Protégé
3. DL queries were run to check the rules

Cogui

4. RDF/OWL file generated using Protégé were imported in Cogui
5. Query facts were run to check the rules
6. All the results of the queries were recorded and analysed
7. The RDF/OWL file generated through Protégé and COGXML file generated through Cogui were compared and analysed.

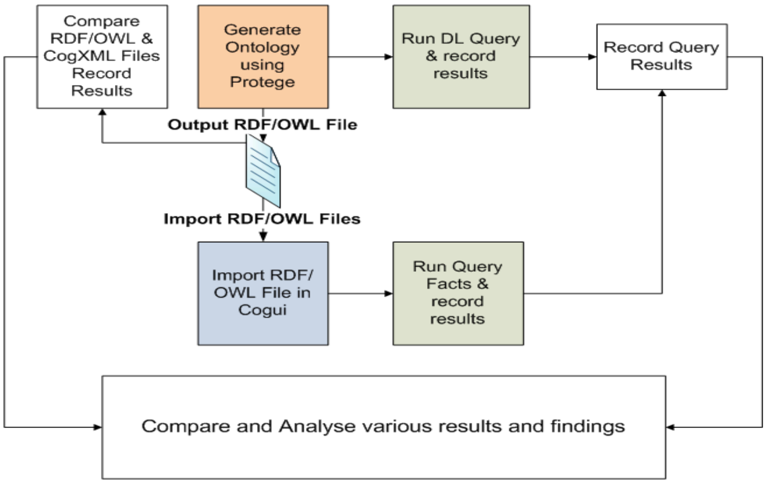

Fig. 1.

Brief Summary of the Case study

1. TechRules Advisor is a fictitious asset management company.
2. The firm buys sells and manages client's asset.
3. A portfolio manager of the firm manages the assets also called 'Portfolio' which has value and creation date.
4. A portfolio manager can buy and sell assets for the customers.
5. An order (buying and selling assets) can be placed by a portfolio manager or Trader.

The domain model of FT case study was inspired by Wegner OCL diagram. The Domain model adopted might not be perfect , but the aim of the study was to evaluate the RDF/OWL ontology to Conceptual Graphs ontology so that interoperability can be proved.

Various classes were created in Protégé and following rules were applied

1. A portfolio is rated platinum, if the total asset value is greater than 1000000 $. It is rated gold, if total asset value is less than 1000000 $ and greater than 100000 $. It is rated regular, if total asset value is less than 100000$.
2. An order cannot be placed both by the trader and by the portfolio manager.

3.1 Tools

OWL Protégé – (For RDF and OWL) Protégé is a tool used to generate OWL based ontology. Protégé implements a rich set of knowledge-modelling structures and actions that support the creation, visualization, and manipulation of Ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. (http://protege.stanford.edu/overview/index.html)

CoGui – (For Conceptual Graphs and import of RDF) Cogui is a tool used to generate Conceptual Graph based ontology. It has the capability to import RDF/OWL files. Cogui is a visual tool for building Conceptual Graph knowledge bases (KB). It allows creating a KB, to edit its structure and content, and to control it. The KB can be serialized in the XML format called COGXML. (http://www.lirmm.fr/cogui/nutshell.php)

4 Analysis

The ontology created in protégé consists of

1. Class hierarchy
2. Object properties
3. Data properties.
4. Individuals

4.1 Comparison of Classes

The diagram shown below provides a details a cut down view of the class hierarchy.

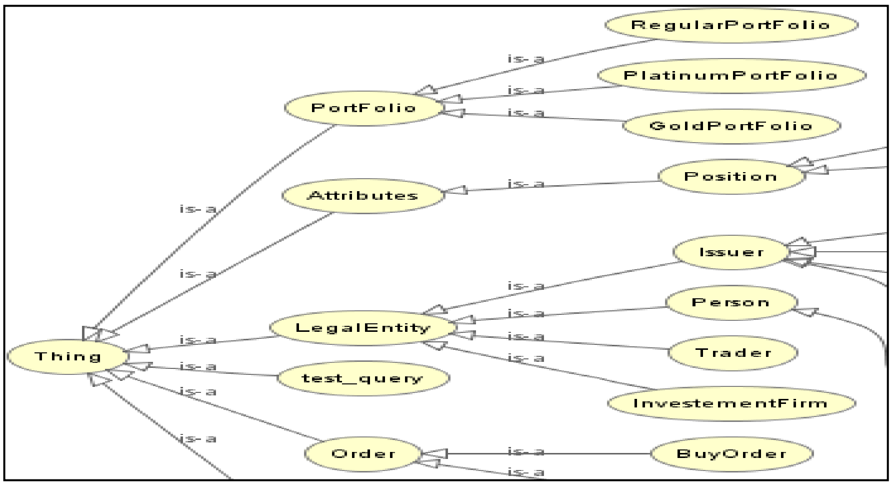

Fig. 2.

When the Protégé file was imported in CoGui tool , the entire ontology was imported 'as is' without loss of any classes. The testimony to this fact is shown in the diagram below which is a cut down version of the class diagram.

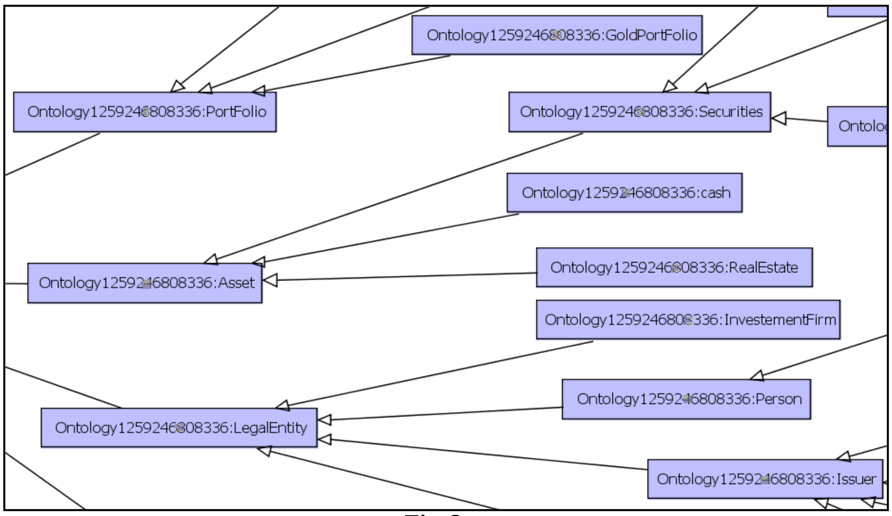

Fig. 3.

The table below elaborates various classes created in OWL ontology and compares it with the Ontology created in CoGui after importing RDF/OWL files. (Only a small number of classes are shown in the table below)

Protégé		**Cogui**	
Super Class	**Sub Class**	**Super Class**	**Sub Class**
Thing		TopCG	
		Class.	
		Thing	
	Asset		Asset
	1. Real Estate		1. Real Estate
	2. Securities		2. Securities
	a. Bonds		a. Bonds
	b. Options		b. Options
	c. Stock		c. Stock
	3. Cash		3. Cash
	Portfolio		Portfolio
	1. RegularPortfolio		1. RegularPortfolio
	2. GoldPortfolio		2. GoldPortfolio
	3. PlatinumPortfolio		3. PlatinumPortfolio

Table 4.

4.2 Comparison of Object and Data Properties

The table below elaborates various Object properties created in OWL ontology and compares it with the Ontology created in CoGui after importing RDF/OWL files.

<table>
<tr><th colspan="2">Protégé</th><th colspan="2">Cogui</th></tr>
<tr><td>Object Property</td><td>Data Type Property</td><td>Object property</td><td>Data Type Property</td></tr>
<tr><td>has Asset</td><td>hasTotalAssetValue</td><td>has Asset</td><td>hasTotalAssetValue</td></tr>
<tr><td>hasEmployer</td><td>canBeSold</td><td>hasEmployer</td><td>canBeSold</td></tr>
<tr><td>hasIssuer</td><td>hasDayCreated</td><td>hasIssuer</td><td>hasDayCreated</td></tr>
<tr><td>hasOwner</td><td>hasDaysHeld</td><td>hasOwner</td><td>hasDaysHeld</td></tr>
<tr><td>isManagedBy</td><td>hasFirstName</td><td>isManagedBy</td><td>hasFirstName</td></tr>
<tr><td>isPlacedBy</td><td>hasLasName</td><td>isPlacedBy</td><td>hasLasName</td></tr>
<tr><td>isPlacedFor</td><td>hasName</td><td>isPlacedFor</td><td>hasName</td></tr>
<tr><td colspan="2" rowspan="5">No instances found in Protégé</td><td colspan="2">Other types created in Cogui are</td></tr>
<tr><td>firstCG</td><td></td></tr>
<tr><td>restCG</td><td></td></tr>
<tr><td>rdfType</td><td></td></tr>
<tr><td>valueCG</td><td></td></tr>
<tr><td colspan="2" rowspan="2">RDF/OWL provides default properties and mechanism to define range , restriction , disjoint classes and member of the class.</td><td>Xsd:MaxInclusive</td><td></td></tr>
<tr><td>Xsd:minInclusive</td><td></td></tr>
<tr><td colspan="2" rowspan="8">Cogui/ CGs do not have default values , every property definition and restriction is a created as a relationship type entity.</td><td>Xsd:MaxExclusive</td><td></td></tr>
<tr><td>XSD:MinExclusive</td><td></td></tr>
<tr><td>onDataRange</td><td></td></tr>
<tr><td>onDataType</td><td></td></tr>
<tr><td>OnProperty</td><td></td></tr>
<tr><td>Qualifiedcardinality</td><td></td></tr>
<tr><td>Somevaluesfrom</td><td></td></tr>
<tr><td>withRestriction</td><td></td></tr>
</table>

Table 5.

4.3 Visual Comparison of Files

In order to investigate the human readable aspect of the Semantic Web the XML files generated by Protégé and CoGui were visually compared. Since the files were large only one class representation has been shown here below.

Protégé

```
<owl:Class rdf:about="#GoldPortFolio">
    <rdfs:subClassOf rdf:resource="#PortFolio"/>
</owl:Class>
```

Cogui

```
<ctype id="http://www.semanticweb.org/ontologies/Ontology1259246808336.owl#PortFolio" label="Ontology1259246808336:PortFolio" x="610" y="299">
<translation descr="" label="Ontology1259246808336:PortFolio" lang="en"/>
<ctype id="http://www.semanticweb.org/ontologies/Ontology1259246808336.owl#GoldPortFolio" label="Ontology1259246808336:GoldPortFolio" x="899" y="240">
<translation descr="" label="Ontology1259246808336:GoldPortFolio" lang="en"/>
<order id1="http://www.semanticweb.org/ontologies/Ontology1259246808336.owl#GoldPortFolio" id2="http://www.semanticweb.org/ontologies/Ontology1259246808336.owl#PortFolio"/>
```

4.4 Rules comparison

To evaluate the implementation of the rules various restrictions were applied for the following classes

Portfolio (Super Class – Generalised)

a. Regular (Sub Class – Specialised)
b. Gold (Sub Class – Specialised)
c. Platinum (Sub Class – Specialised)

Rule 1

In order to implement the class restriction a data property called 'hasTotalAssetValue' was created. This property was created for the Portfolio class. The sub classes of portfolio were specialised by implementing the restriction as defined in the Rule 1.

Sub Classes of Portfolio	Restriction
Regular Portfolio	***"hasTotalAssetValue some double[<100000]"***
Gold Portfolio	***"hasTotalAssetValue some double[>= 100000] and hasTotalAssetValue some double[<= 1000000]"***
Platinum Portfolio	***"hasTotalAssetValue some double[> 1000000]"***

Table 6.

These rules were validated by creating following individuals

1. Portfolio1_Regular having asset value of 1000
2. Portfolio2_Gold having asset value of 100000
3. Portfolio3_Platinum having asset value of 9999999999

Protégé's DL query engine was used to verify the rules.

DL Query Command	Protégé Results
"hasTotalAssetValue some double [> "1000"^^double]"	Following individuals were highlighted by Protégé DL Query 1. Portfolio2_Gold 2. Portfolio3_Platinum
"hasTotalAssetValue some double [< "1000"^^double]"	Following individuals were highlighted by Protégé DL Query 1. Portfolio1_Regular

Table 7.

Rule comparisons in Cogui

When the RDF/OWL file was imported into Cogui, it imported the individuals also. For example the diagram given below shows the 'Portfolio3_Gold' individual imported as a fact in Cogui

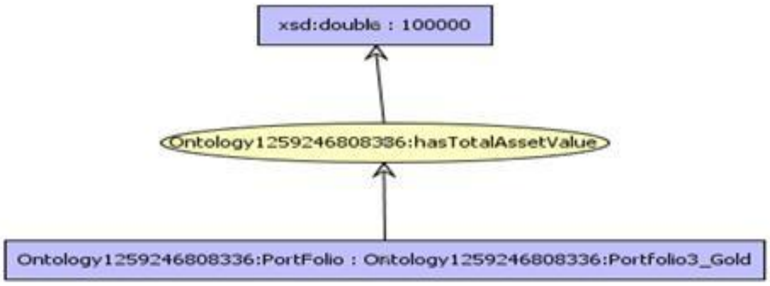

Fig. 4.

A new rule was created in Cogui. After running the query fact following results were observed

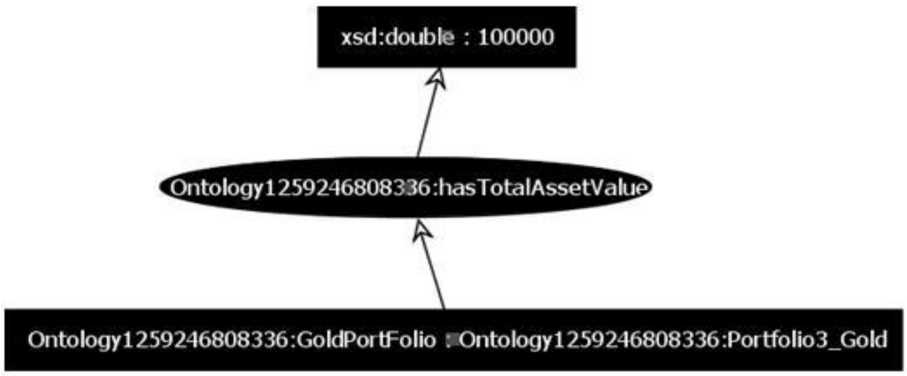

Fig. 5.

This is one of the most important aspects of the case study, because the results show two important concepts.

1. The individual created in Protégé "Portfolio3_Gold" was automatically taken by the query wizard. This individual was imported as a fact.
2. The rule created in CoGui was used for the projection.

It can be seen that the Class GoldPortfolio and Individual Portfolio3_Gold both are projected in the result graph. This proves that the 'facts' imported from Protégé and the rules written in CoGui are integrated and used by the reasoning engine, hence it can be said that RDF/OWL is interoperable with CL.

4.5 Outcomes of the Analysis

Based on the above data, it can be confirmed that the RDF/OWL files can be mapped to Conceptual graphs. Details of the analysis are given below:

Interoperability confirmations

1. RDF/OWL ontology is successfully imported and mapped to Conceptual Graphs
2. RDF/OWL classes were mapped to Concept Types in CoGui.
3. Object and Data Property were mapped to Relationship types
4. The domain and ranges in Protégé were converted as signature of the relation in Cogui.
5. The URIs are imported and retained in the Conceptual Graphs
6. All the relationship types were imported.
7. All the individuals created in Protégé were automatically created in CoGui. The relationship defined for individuals were successfully imported.
8. The rules created in CoGui were successfully used by the imported Individuals

Unconfirmed facts

1. It is not known how to express data property related restrictions in CoGui.
2. The data type and object type properties were successfully imported, but the rules were not automatically created.
3. RDF/OWL ontology have the ability to use pre-defined primitives like double , string , integers etc. Whereas these primitive data types are imported as Concept Types.

Mapping of Ontologies was done using tools and visual comparison of the output files. No RDF/OWL ontology modelling tool was found which could import CG Ontologies.

5 Conclusion

This study has successfully confirmed that the RDF/OWL can be mapped to Conceptual Graphs. Conceptual Graphs have the capability to support all the strengths identified for RDF/OWL. Case Study has provided evidences that RDF/OWL can be successfully mapped to CL . This hypothesis was also proposed by Polovina, Cooke and Loke (2009).

The table below summarizes the support to interoperability requirements

Requirement	**RDF/OWL**	**Conceptual Graphs**
Application and Ontologies should be able to reference resources and concepts defined elsewhere on web	Supports	Supports
Resources and concepts should be sharable	Supports	Supports

over web.		
The underlying language and data exchange mechanism should be understandable by all for example XML	Supports	Supports
Capability to merge several Ontologies is essential.	Not known	Supports (Proved by case study analysis)
Capability to understand different Ontologies	Not known	Supports (Proved by case study analysis)
Languages and Ontologies should be expressive enough for serious use	Supports (OWL 2)	Supports
Languages should be able to support kinds of logical reasoning that are found to be needed to conduct the business of Semantic Web	Supports (OWL 2)	Supports
To resolve incompatible Conceptualization and different modelling styles some form of standardization is required	W3C	ISO

Table 8.

Some of the other conclusions are listed below

1. RDF/OWL has also progressed since it was officially recommended by W3C in 2004. OWL 2 has provided greater expressivity capability as it was very easy to express data restrictions using Protégé.
2. Protégé is very good tool which can handle complex business logic.
3. CoGui can import RDF/OWL files, but in absence of separately installed Cogitant reasoning engine it has limited capability. It only provides Query fact Wizard.

Conceptual Graph tools need to be developed further, to be able to compete with Protégé. Protégé's DL query engine is easy to understand and provides results by listing individuals and classes. This is very important from a user's point of view because it helps the user to confirm the business rules. Similarly CoGui provides graphical representation of the query projection showing the path taken by the reasoning engine and the rules applied by it.

It is easy to define data properties in Protégé thereby in RDF/OWL as it supports predefined primitive types like integer , double , string etc. Whereas Conceptual Graphs do not support these , it can be recommended that CL specifications should be enhanced to accept these data types as standard concept types and when the tools are developed or improved they should have the capabilities to use these data types in rule definitions.

Since there are no tools available, which can map Common Logic to RDF/OWL? it can be concluded that all ontologies should be mapped to CL as suggested by Polovina et al (2009) as it has been observed that CoGui can import RDF/OWL.

Future research can be taken up with Cogitant reasoning engine. The current research has shown that the default Cogitant reasoning engine available with CoGui is very powerful but some limitations were observed. It is important to repeat the case study with full capability Cogitant reasoning engine , this will provide further insight into "Why all rules were not imported or displayed in the graphs ?"

A transaction in the FT Case study can be modelled using TRAM as identified by Hill and Polovina (2009) , based on this a future research project can be undertaken to create transaction model in Protégé , which can then be imported in CG tools like CoGui to further confirm the interoperability between RDF/OWL and Common Logic.

Investigation is required to identify tools which can map Common Logic to RDF/OWL then only the interoperability concerns can be satisfied and resolved.

6 References

1. Cogui Nutshell [Online], www.lirmm.fr/cogui/nutshell.php.
2. Conceptual Graphs [Online], http://conceptualgraphs.org/
3. DELUGACH , H:ISO/IEC WD24707 Information technology — Common Logic (CL): A Framework for a Family of logic based languages , Pacific Northwest National Laboratory,Chantilly , Volume 7(2007).
4. DIENG-KUNTZ ,Rose,CORBY, Olivier. Conceptual Graph for Semantic Web Applciation. Lecture notes in computer science, 1611-3349, Pages 19-50, Springer Berning Heidelberg Publisher (2005).
5. GRAU ,B, HORROCKS , I, MOTIK,B, PARSIA ,B,PATEL-SCHNEIDER, P,SATTLER, U:OWL 2: The next step for OWL,Web Semantics.Web Semantics: Science, Services and Agents on the World Wide Web, Volume 6, Issue 4,Pages 309-322(2008)
6. POLOVINA, Simon:A Transactions Pattern for Structuring Unstructured Corporate Information in Enterprise Applications International. Journal of Intelligent Information Technologies, 5(2), Pages 67-81(2009)
7. HORRIDGE,, Matthew :A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools, The university of Manchester, Edition 1.2(2009)
8. POLOVINA, Simon, COOKE, James and LOKE, Jeremy:A Practical Exploration of Ontology Interoperability .Lecture Notes in Computer Science.Springer Berlin / Heidelberg, Volume 5662/2009 ,0302-9743 (Print) 1611-3349, Pages 247-256. (2009) ,
9. POLOVINA, Simon, MIZZI, Doreen, BROWA ,Wolfgang and LOKE, Jeremy: A Commentary on Standardization in the Semantic Web, Common Logic and Multi Agent Systems. Program Committee (Alphabetic Order) [Online].Pages 33-41(2009)

10. Protégé [Online]. http://protege.stanford.edu/overview/index.html
11. REYNOLDS, Dave,THOMPSON, Carol, MUKERJI ,Jishnu,COLEMAN,Derek:An assessment of RDF/OWL modelling , HP Technical Report , HP Laboratories Bristol , Hewlett Packard , HPL-2005 -189,2005,Tech.Rep (2005)
12. Semantic Web [Online]. http://www.w3.org/2001/sw/