IJRCM

INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION AND MANAGEMENT

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing, Human Resource Management, General Management, Banking, Insurance, Corporate Governance and emerging paradigms in allied subjects like Accounting Education; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Monetary Policy; Portfolio & Security Analysis; Public Policy Economics; Real Estate; Regional Economics; Tax Accounting; Advertising & Promotion Management; Business Education; Business Information Systems (MIS); Business Law, Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labor Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; Public Administration; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism, Hospitality & Leisure; Transportation/Physical Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Digital Logic; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Multimedia; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design. The above mentioned tracks are only indicative, and not exhaustive.

Anybody can submit the soft copy of his/her manuscript **anytime** in M.S. Word format after preparing the same as per our submission guidelines duly available on our website under the heading guidelines for submission, at the email addresses, infoijrcm@gmail.com or info@ijrcm.org.in.

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR

IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF ____.**

(e.g. Computer/IT/Finance/Marketing/HRM/General Management/other, please specify).

DEAR SIR/MADAM

Please find my submission of manuscript titled '____' for possible publication in your journal.

I hereby affirm that the contents of this manuscript are original. Furthermore it has neither been published elsewhere in any language fully or partly, nor is it under review for publication anywhere.

I affirm that all author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if our/my manuscript is accepted, I/We agree to comply with the formalities as given on the website of journal & you are free to publish our contribution to any of your journals.

NAME OF CORRESPONDING AUTHOR:

Designation:

Affiliation with full address & Pin Code:

Residential address with Pin Code:

Mobile Number (s):

Landline Number (s):

E-mail Address:

Alternate E-mail Address:

2. **INTRODUCTION**: Manuscript must be in British English prepared on a standard A4 size paper setting. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of the every page.
3. **MANUSCRIPT TITLE**: The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.
4. **AUTHOR NAME(S) & AFFILIATIONS**: The author (s) full name, designation, affiliation (s), address, mobile/landline numbers, and email/alternate email address should be in italic & 11-point Calibri Font. It must be centered underneath the title.
5. **ABSTRACT**: Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para.
6. **KEYWORDS**: Abstract must be followed by list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.
7. **HEADINGS**: All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.
8. **SUB-HEADINGS**: All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.
9. **MAIN TEXT**: The main text should be in a 8 point Calibri Font, single spaced and justified.
10. **FIGURES &TABLES**: These should be simple, centered, separately numbered & self explained, and titles must be above the tables/figures. Sources of data should be mentioned below the table/figure. It should be ensured that the tables/figures are referred to from the main text.
11. **EQUATIONS**: These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.
12. **REFERENCES**: The list of all references should be alphabetically arranged. It must be single spaced, and at the end of the manuscript. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio," Ohio State University.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITE

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Economic and Political Weekly, Viewed on July 05, 2011 http://epw.in/user/viewabstract.jsp

XMOWL MODEL: SUPERVISED APPROACH TO TRANSFORM SYNTACTIC MODEL TO SEMANTIC MODEL

SHIKHA SINGH
RESEARCH SCHOLAR
SINGHANIA UNIVERSITY
PACHERI BERI, JHUNJHUNU

DR. U. S. PANDEY
ASSOCIATE PROFESSOR
SCHOOL OF OPEN LEARNING
UNIVERSITY OF DELHI
DELHI

ABSTRACT

Ontologies are used in the assimilation of information resources by describing machine readable terms and definitions in semantic manner of the information sources. But, creating an ontology is a difficult and time-consuming process, especially in the early stage of extracting key concepts and relations. This paper proposes a method for domain ontology building by extracting ontological knowledge from UML models of existing systems. We compare the OUP based XML model elements with the OWL ones and derive transformation rules between the corresponding model elements. Based on these rules, we define an XSLT document which implements the transformation processes. XSLT document will be based on DOM and XPath..We expect that the proposed method reduce the cost and time for building domain ontologies with the reuse of existing UML models.

KEYWORDS
OWL, Semantic model, UML, XML, XSLT.

INTRODUCTION

The Semantic Web allows data to be shared and reused across application, enterprise, and community, boundaries by providing a common framework". (Cayzer, 2004)

Both intelligent agents and human users can view Semantic Web as an expressive, collaborative, and open, information system in the Web. Hence it comes with the following features:

Simple but expressive data model. URI (Uniform Resource Identifier)- based vocabulary and RDF triples are being used in RDF data model to describe the world.

Collaborative publishing. Agents publish data independently using common meta-ontology (i.e. RDFS and OWL) and same RDF graph model.

Open system in web context. Semantic Web data is published throughout the Web. Both URI-based vocabulary and RDF triples are distributed extensible. (Ladner and Petry, 2005)

Syntactic Web emphasize on the current, mostly HTML-based World Wide Web, in order to distinguish it from the Semantic Web, a concept in which web pages carry information that can be read and understood by machines in a systematic way. The term stems from the contrast between syntax, which is the mechanics of a language used to convey information, and semantics, which is the actual meaning of that information. On a syntactic web page, which is any document on the web that does not contain special tagging to convey meaning, meaning is difficult to parse by a computer program.

An example is a site that gives the weather for any city in the world, in HTML form. Even though the site offers dynamic, database-driven information, it is presented in a purely syntactic way. One could imagine a computer program that tried to retrieve this weather information through text parsing or "web scraping". Though it would be possible to do, if the creators of the site ever decide to change around the layout or HTML of the site, the computer program would most likely need to be rewritten in some way. In contrast, if the weather site published its data semantically, the program could retrieve that semantic data, and the site's creators could change the look and feel of the site without affecting that retrieval ability.

Applications of Semantic Web would be more than just browsing of Web pages. For example, consider the task of constructing a list of the computer science courses available in Indian Universities at particular location. This is a complex search task for any user of the current search engine technology but rather simple to complete with a database of courses available in Universities of India. To perform it using the existing search engines, we would have to retrieve all the universities providing these courses in entire India and then search for particular location, and then you will find only the universities not the desired course. In a database, this would be a simple query using a combination of select, join and sort operations.

The solution lies in part with the creation of the Semantic Web (SW). The Semantic Web will act as a Web of machine understandable data for the purpose of easily automating user and computer tasks. Metadata would be used defensibly for this purpose [6]. Metadata is data that describes attributes and properties about other data, such as Web pages. Derived meta-metamodels include Entity Relationship Diagrams, Formal languages, EBNF, Ontology languages, XML Schema, and MOF. The strengths of these languages tend to be in the familiarity and standardization of the original language.

In Semantic Web software agents will be used by computers to scan and interpret information on Web pages. These software agents are the programs which crawl through the Web, searching for relevant information. The semantic web has collection of information called ontologies.

The Unified Modeling Language (UML) has gained widespread acceptance for the development of software systems [2]. UML is used to visualize, specify, construct and document a software system from multiple perspectives. In a distributed development scenario it is important to allow software engineers and customers to exchange, access, review, query and browse UML models.

Due to its wide acceptance in object oriented software developments, researchers find a new way for reusing the already developed UML models to create new generation Web i.e to develop ontologies. Various research have been going on ontology development, our focus is on how ontology can be developed using already existed UML model. So in this paper we are discussing XMOWL model which emphasize on XMI to OWL transformation.

XML-based Metadata Interchange (XMI) is an interchange format for metadata defined in terms of the MOF standard. In addition to supporting the exchange of complete models, XMI supports the exchange of models in differential form. A generic transformation can be configured to generate (via XSLT) a specialized transformation that will be used to transform a UML model. The approach promotes model reuse, speeds up the modeling process and can be used to assure that only predefined semantics (as specialized by an agent) is included in the transformed model [3].

This paper is based on semi-automated approach for transformation of UML model to OWL.

A general usage scenario for such transformations looks like this: A UML class model ci stored in a UML repository. Now a software agent wants to transform ci, to obtain a model ci+1 which expresses the new semantic model. However, suppose completely manual model transformation by an agent is unacceptable - we would like to assure that the semantics contained in ci+1 is understood by a UML model-driven compiler (e.g.a model-based software generator [4]). As a solution to this problem, we half-fabricate a model part (the difference between ci+1 and ci) and represent it as a generic transformation. By configuring a generic transformation, the agent produces an XMI document describing the transformation that has to be applied to obtain the model ci+1.

First section will be on Literature Review and the next section will be on OUP and tools for generating the XMI document from UML model , on XML and XML schemas and third will be based on XSLT document to generate new XML document from XML. And the last section will be based on Ontology OWL creation from XML us

LITERATURE REVIEW

A variety of different research projects and commercial initiatives have been applying UML for ontology representation. UML can be used directly as an ontology representation or as a graphical front-end for another ontology representation language (e.g., DAML, OWL). UML has been used with a variety of implementations (e.g., Java objects, Open Knowledge Base Connectivity (OKBC)[9]. UML has been applied to a variety of ontology related tasks (e.g., ontology mapping, consistency checking).

Cranefield and Purvis (1999, 2000) have investigated the use of UML class diagrams for representing ontologies and UML object diagrams for representing instance knowledge [9]. A "UML Data Binding" tool for Java (Cranefield, 2001) has been developed to generate Java classes and RDF schemas from a class diagram encoded in the XMI format. In this work, UML is used directly, not as a graphical syntax for another knowledge representation language.

Bergenti and Poggi (Bergenti, 2000) have proposed an approach to agent-oriented software engineering based on the use of UML to model various aspects of a multi-agent system. One of their proposed diagrams is an "ontology diagram", which depicts classes representing agents and domain entity types, and associations representing domain predicates that can be encoded as KIF or FIPA-SL agent message content.

As developing the ontology for semantic web from scratch is time consuming task, so reusability of UML diagrams reduced the cost of ontology development. In this paper I had tried to transform UML class model of already existed class in java application to OWL ontology class.

OUP (ONTOLOGY UML PROFILE)

UML profile is a concept used for adapting the basic UML constructs to a specific purpose. Essentially, this means introducing new kinds of modeling elements by extending the basic ones, and adding the new elements to the modeler's repertoire of tools. Also, free-form information can be attached to the new modeling elements. The Ontology UML Profile extends UML in a standard way to enable ontology modeling in the widely used UML modeling tools.

The Class is one of the most fundamental concepts in the ODM and the Ontology UML Profile. As we noted in the discussion about the essential concepts of the ODM, there are some differences between the traditional UML Class or the concept of a Class in object-oriented programming languages and an ontology class as it is defined in OWL (owl:Class). Fortunately, we are not trying to adopt UML as a stand-alone ontology language, since that might require changes to the basic concepts of UML (Class and others). We only need to customize UML as a support to the ODM.

XML

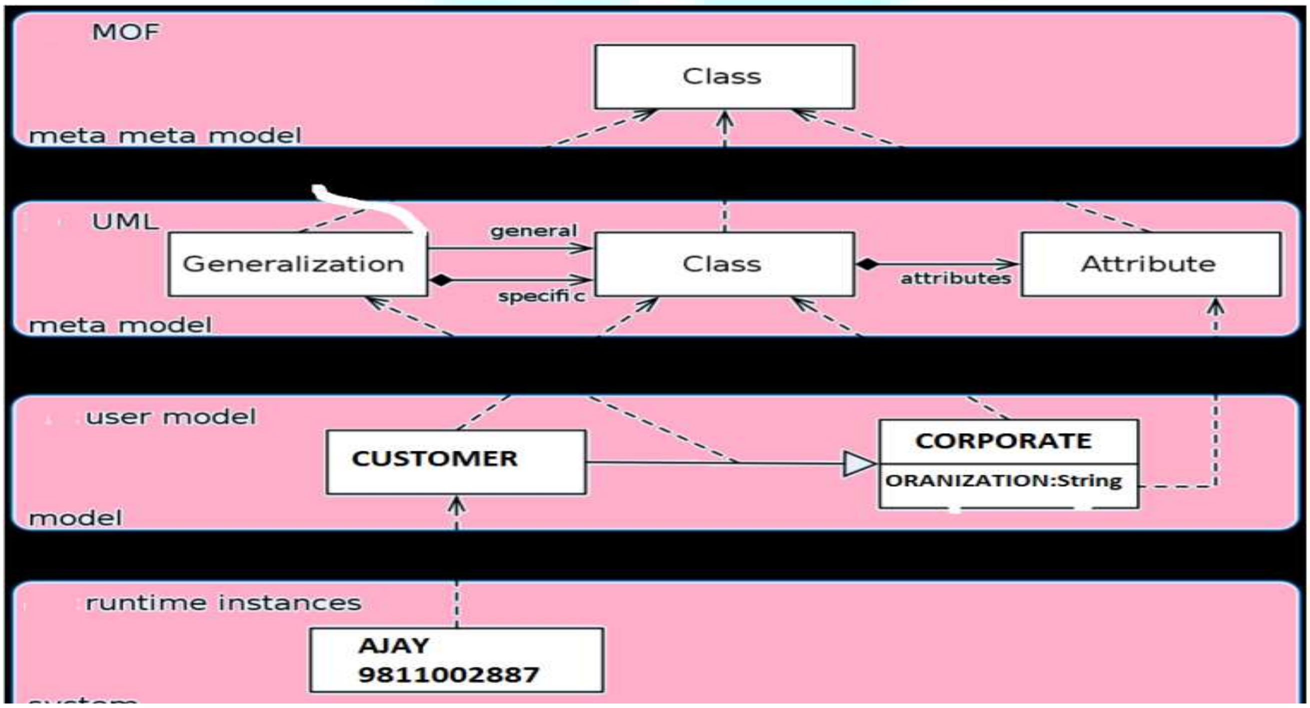
XML has gained ground in becoming a key enabler of these systems in terms of transport of information and commands. XML schemas, which are used to define and constrain the nature of XML exchanged, have consequently come into the limelight. XML has become the de-facto standard for sharing and exchanging information via the web. The prevalence of OMG specifications has motivated analysts to generate UML models for such applications to serve as a PIM[5]. A UML class diagram can be constructed to represent the elements, relationships, and constraints of an XML application visually and then transformed into one or more XML schemas. XML Metadata Interchange (XMI) was designed to enable easy interchange of metadata between modeling tools and can be further used to generate XML DTDs from UML diagrams using a set of conversion

A. XMI (XML Metadata Interchange)

XMI is a standard which defines how to serialise object instances. Although XML is a very good way to store information in a tree structured way, it is not object oriented. XMI extends XML to make it object oriented.

More recent work by the Object Management Group (OMG) has produced the Meta-Object Facility (MOF) and its variants, which promise to unify UML, OWL, and any other modeling language or paradigm (OMG's MOF[8]). UML was born in the software engineering domain. OWL and CL were born in the domain of formal models and logical inference and classification. It is instructive to compare their meta-models. MOF, first and foremost the meta-model of UML, captures four main concepts: 1) classes, 2) associations, 3) datatypes, and 4) packages. OWL models are composed of 1) classes, 2) properties, and 3) instances.

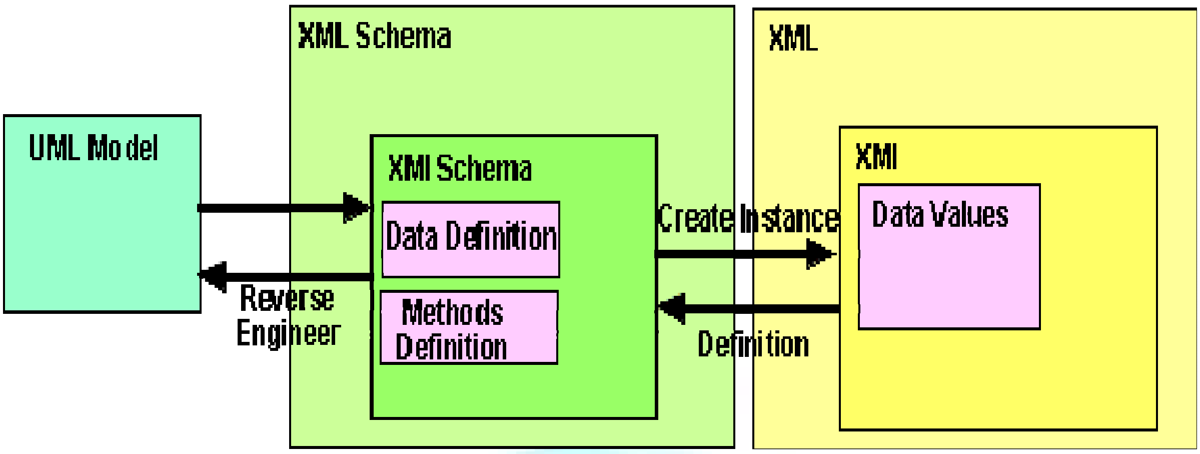
FIGURE 1: FOUR LAYER MOF ARCHITECTURE OF UML CLASS DIAGRAM



B. Using XMI to store UML Models

We can use XMI to store UML models. XMI is defined for storing object instances; however we can create instances of a meta model (possibly based on the Meta Object Facility (MOF) specification from the Object Management Group (OMG).

FIGURE 2: ARCHITECTURE OF UXML MODEL

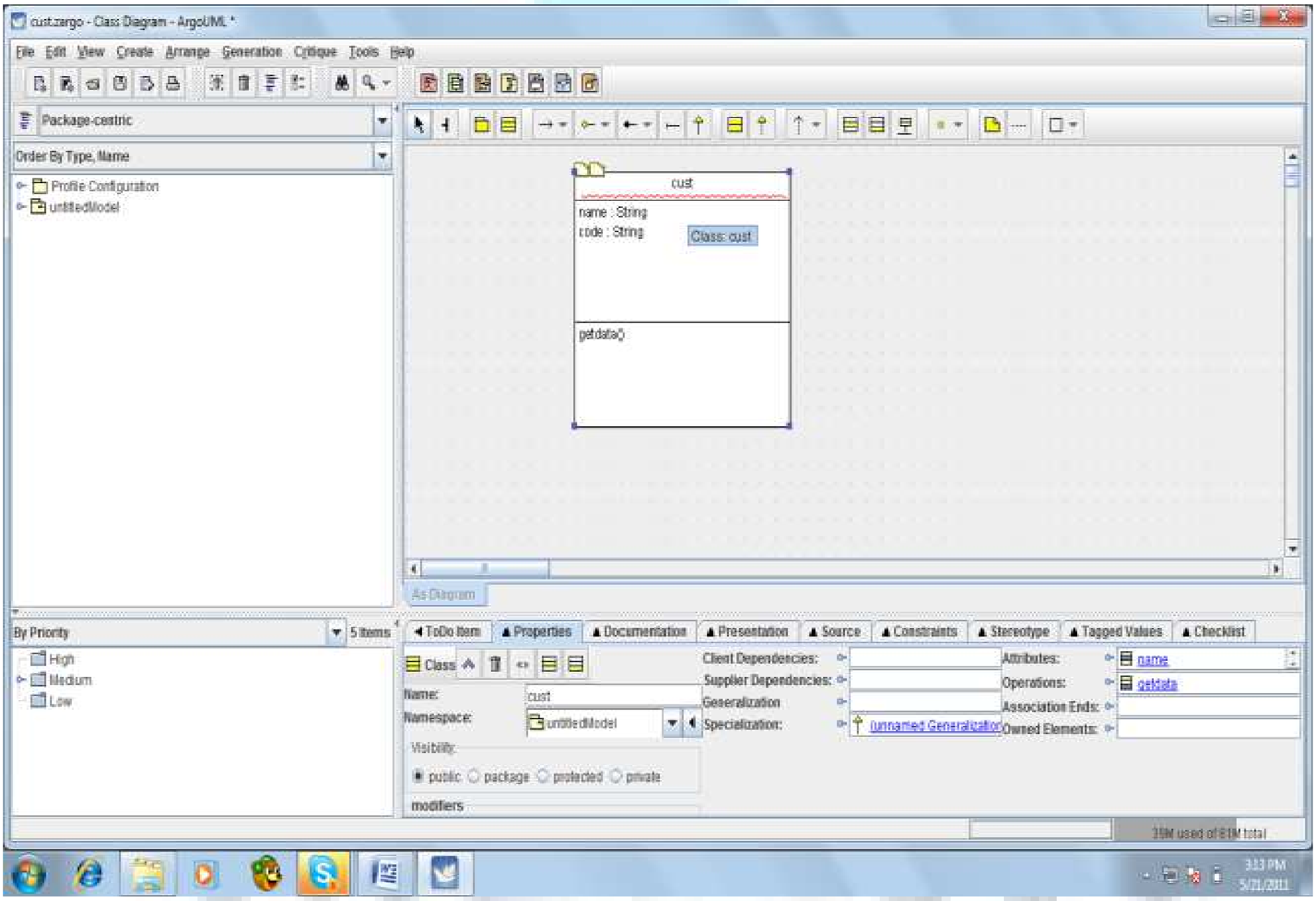


The XMI schema also extends the XML schema so that definitions of objects can be stored. This provides a way to hold a UML model.

We focus only on the class diagrams, since, the XML applications we are addressing are the generation of XML documents from their corresponding schemas and they involve only structure of the application.

I have used open source UML tool for creating UML model diagram.I had created a UML class diagram for Customer class i.e cust class and then saved as .xmi file for the same.

FIGURE 3: CUST CLASS USING OPEN SOURCE TOOL ARGOUML



XML dump of the above file is as given below:

XMI xmi.version = '1.2' xmlns: UML = 'org.omg.xmi.namespace.UML' timestamp = 'Sat May 21 15:27:15 GMT+05:30 2011'>
<XMI.header> <XMI.documentation>
<XMI.exporter>ArgoUML (using Netbeans XMI Writer version 1.0)</XMI.exporter>
<XMI.exporterVersion>0.32.2(6) revised on $Date: 2010-01-11 22:20:14 +0100 (Mon, 11 Jan 2010) $ </XMI.exporterVersion>
</XMI.documentation>
<XMI.metamodel xmi.name="UML" xmi.version="1.4"/></XMI.header>
<XMI.content>
<UML: Model xmi.id = '127-0-0-1-5fc8e80a:12f66f0ebfe:-8000:0000000000000865'
name = 'untitledModel' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'

XML transformed file for the above XMI is given below:

<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="cust" type="cust" />

```
- <xs:complexType name="cust">
- <xs:sequence>
<xs:element name="code" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="name" type="xs:string" minOccurs="1" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

C. XML Schema

With XML, the implementation will be an XML schema. The industry generally takes two views on the relationship between the models and the XML schema. Some authors draw a clear line between the design models, typically UML models or entity-relationship models which are supposed to be abstract, and the XML schemas which include lots of implementation details. This distinction promotes a clean separation between the modeling activity and the implementation activity. Modeling is typically done by business analysts, while implementation is the responsibility of technicians.

D. Ontology

Ontologies play an important role in defining the terminology that agents use in the exchange of knowledge-level messages and therefore the choice of an ontology representation language is a significant issue when designing a multi-agent system[10]. An ontology defines a common set of concepts and terms that are used to describe and represent a domain of knowledge

The Artificial-Intelligence literature contains many contradictory definitions of Ontology. For the purposes of this project an ontology is a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (called role restrictions)). Together with the classes instances ontology forms the knowledge base (Gruber, 1995). In real terms there is a very fine between where knowledge base starts and Ontology begin. The domain concepts are described by the classes. Most of the Ontologies focus on classes (Noy and McGuinness). For example, a class of persons represents all persons.

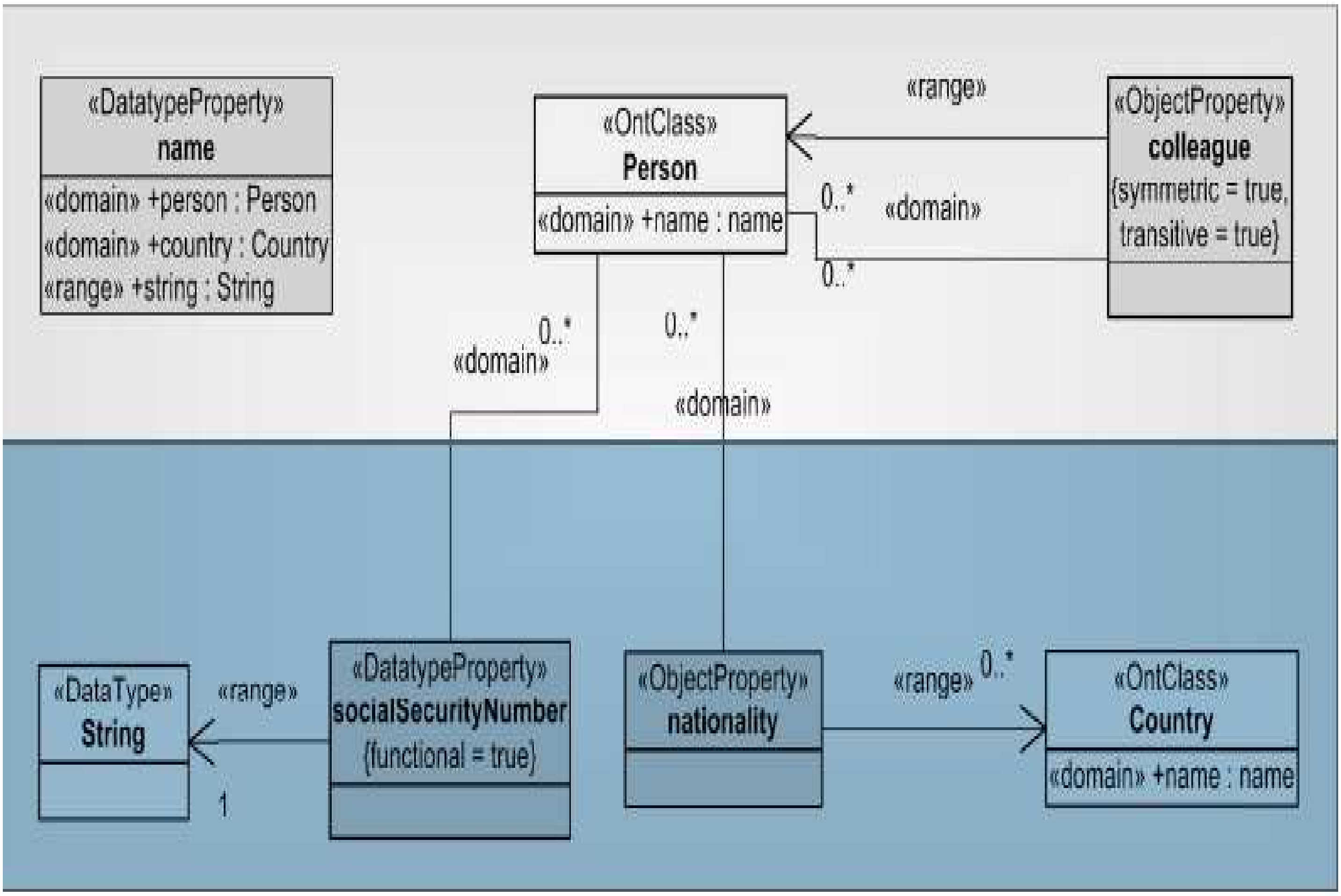
Description logic based ontology languages such as OWL, however, are usually defined in terms of an abstract (text-based) syntax and most care is spent on the formal semantics. As I had discussed earlier that OWL structure is composed of class, properties and instances.

The building blocks of OWL are as follows:

TABLE 1: OWL BUILDING BLOCKS

Element	Description	Example
Individuals	Representation objects in the real world	Ajay, ora1
Classes	Sets of Individuals. Classes may be atomic or complex.	Person, Customer, Order
Object Properties	Represent relationships between two individuals	hasSister,hasParent,workFor
Data Properties	Link individuals to concrete values	has Age,hasName

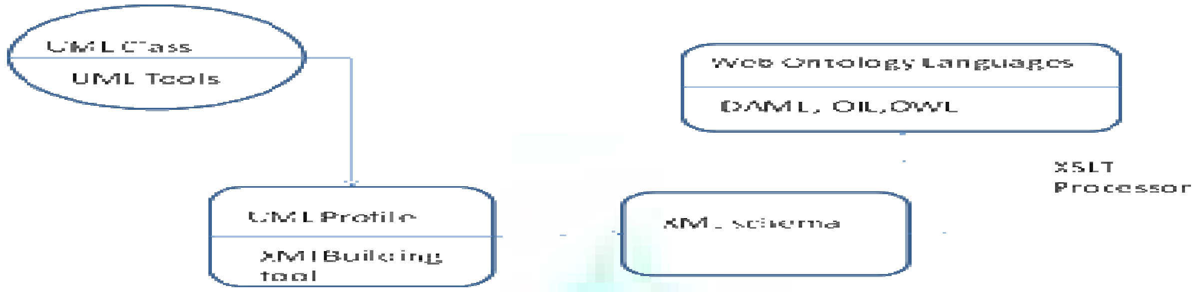
FIGURE 4: OWL ONTOLOGY PROPERTIES



WWWC and DARPA had progressed on the development of tractable ontology language i.e. DAML with the enhancement of UML[1]The researcher had introduced a new model to enhance the domain property of DAML using UML. But we have focused on transformation of UML to OWL.

CONVERSION OF SYNTACTIC MODEL TO SEMANTIC MODEL

FIGURE 5: XMOWL MODEL FOR SYNAPTIC TO SEMANTIC MODEL

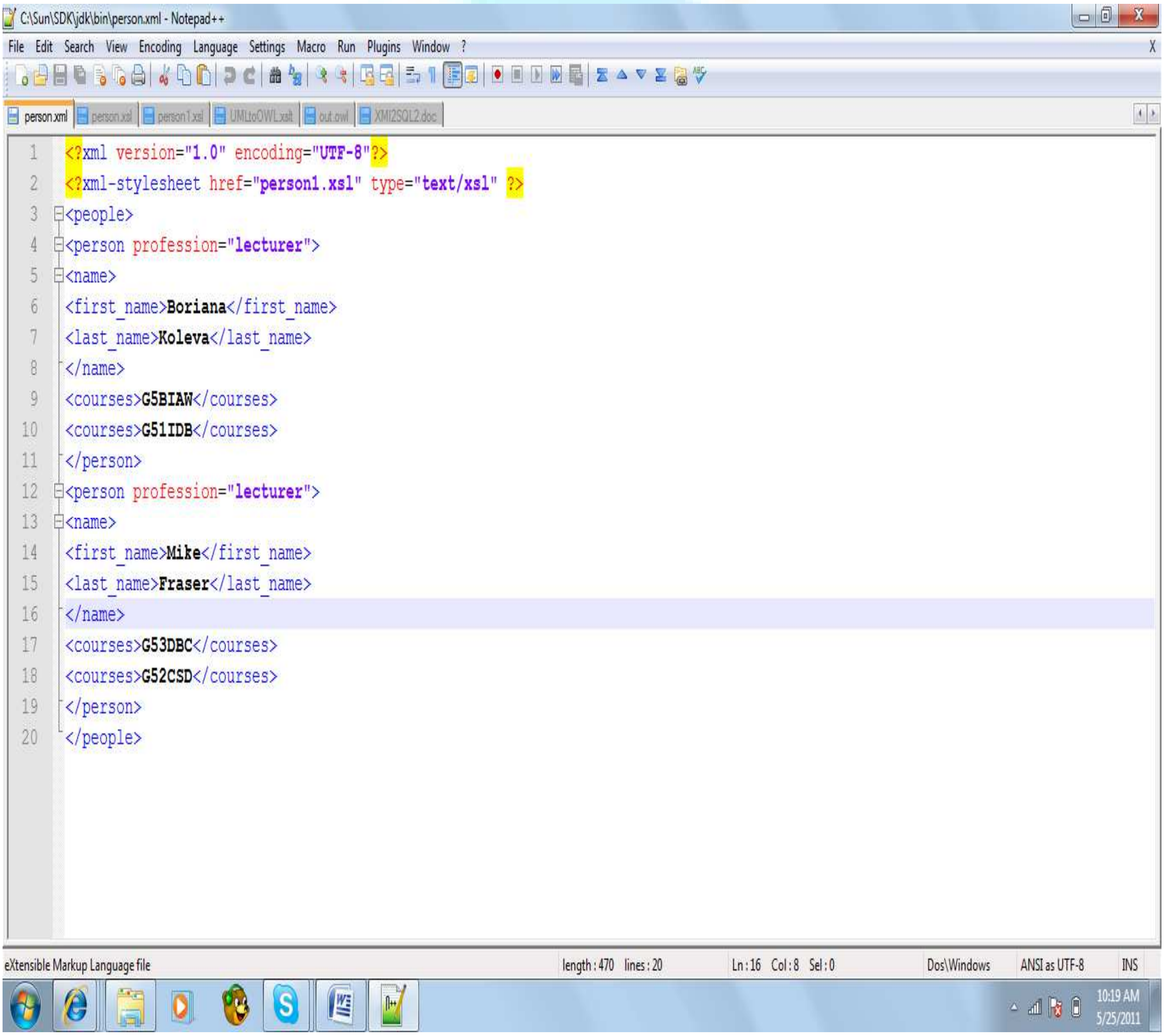


An XSL stylesheet processor accepts a document or data in XML and an XSL stylesheet and produces the presentation of that XML source content that was intended by the designer of that stylesheet. There are two aspects of this presentation process: first, constructing a result tree from the XML source tree and second, interpreting the result tree to produce formatted results suitable for presentation on a display, on paper, in speech, or onto other media. The first aspect is called tree transformation and the second is called formatting. The process of formatting is performed by the formatter. This formatter may simply be a rendering engine inside a browser.

Researchers are working on many projects regarding conversion of UML to OWL and working with different techniques of using XSLT parsers. I had worked on saxon parser to convert XML file to OWL file. XSLT document contains rules to convert XML to other XML form, as ontology is also one form of XML as change to RDF/XML , thus providing the rules for conversion into ontology OWL.

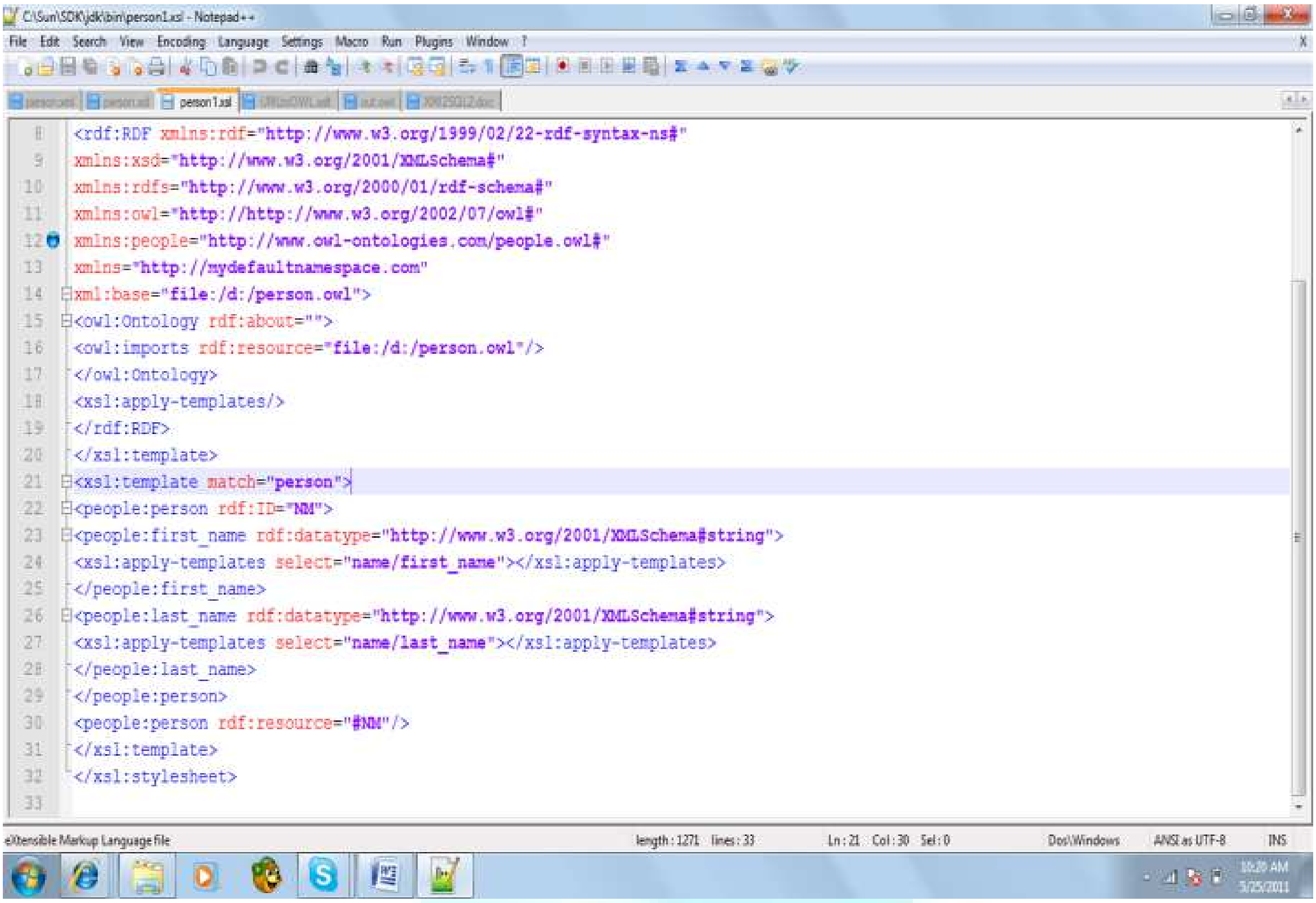
Here is an example of XML file and XSLT document to convert into another XML i.e OWL form.

FIGURE 6: Person.xml



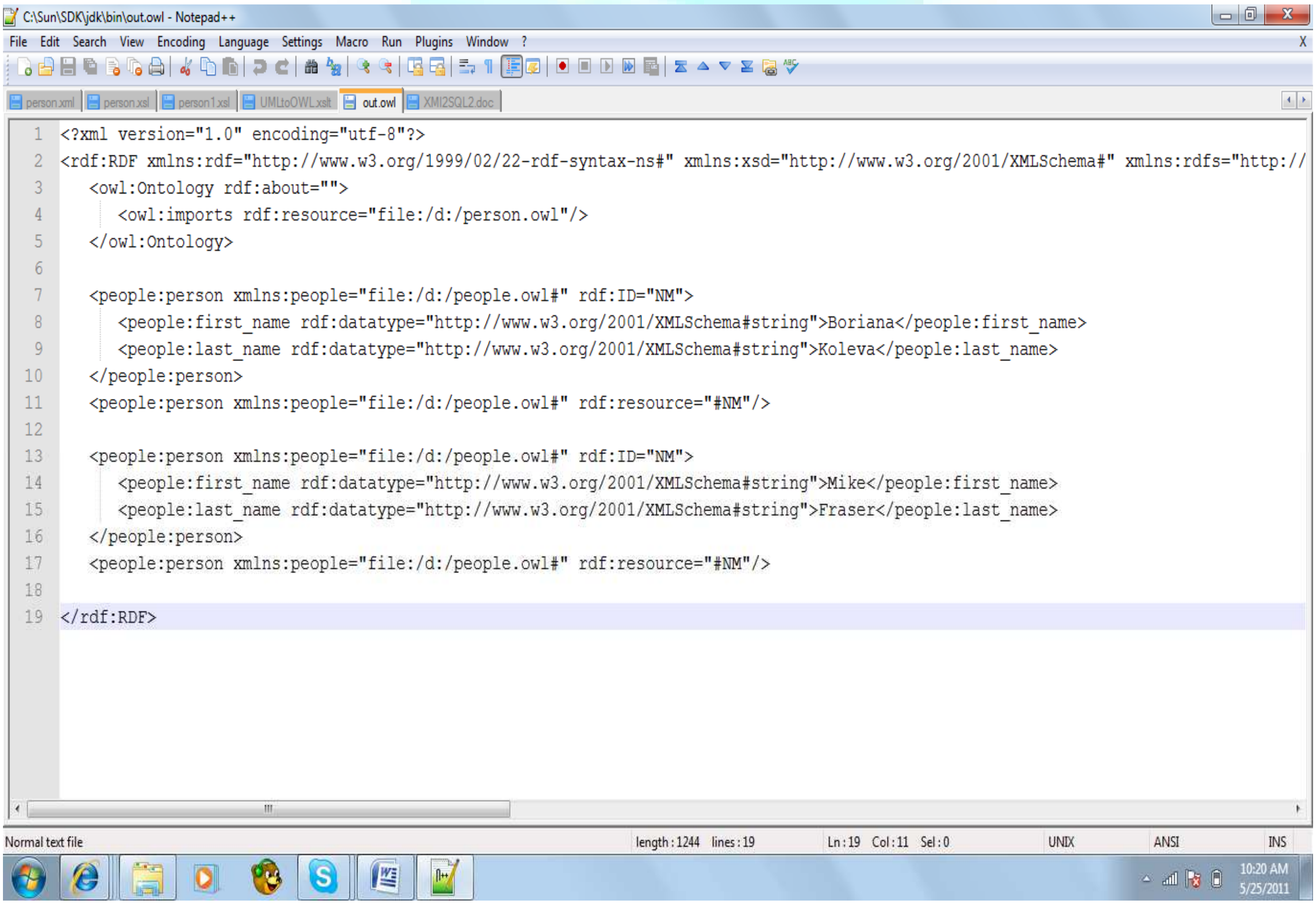
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="person1.xsl" type="text/xsl" ?>
<people>
<person profession="lecturer">
<name>
<first_name>Boriana</first_name>
<last_name>Koleva</last_name>
</name>
<courses>G5BIAW</courses>
<courses>G51IDB</courses>
</person>
<person profession="lecturer">
<name>
<first_name>Mike</first_name>
<last_name>Fraser</last_name>
</name>
<courses>G53DBC</courses>
<courses>G52CSD</courses>
</person>
</people>
```

FIGURE7: Person1.xsl



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://http://www.w3.org/2002/07/owl#"
xmlns:people="http://www.owl-ontologies.com/people.owl#"
xmlns="http://mydefaultnamespace.com"
xml:base="file:/d:/person.owl">
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="file:/d:/person.owl"/>
</owl:Ontology>
<xsl:apply-templates/>
</rdf:RDF>
</xsl:template>
<xsl:template match="person">
<people:person rdf:ID="NM">
<people:first_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
<xsl:apply-templates select="name/first_name"></xsl:apply-templates>
</people:first_name>
<people:last_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
<xsl:apply-templates select="name/last_name"></xsl:apply-templates>
</people:last_name>
</people:person>
<people:person rdf:resource="#NM"/>
</xsl:template>
</xsl:stylesheet>
```

FIGURE 8: RESULTANT OWL DOCUMENT



```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="file:/d:/person.owl"/>
  </owl:Ontology>

  <people:person xmlns:people="file:/d:/people.owl#" rdf:ID="NM">
    <people:first_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Boriana</people:first_name>
    <people:last_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Koleva</people:last_name>
  </people:person>
  <people:person xmlns:people="file:/d:/people.owl#" rdf:resource="#NM"/>

  <people:person xmlns:people="file:/d:/people.owl#" rdf:ID="NM">
    <people:first_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Mike</people:first_name>
    <people:last_name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Fraser</people:last_name>
  </people:person>
  <people:person xmlns:people="file:/d:/people.owl#" rdf:resource="#NM"/>

</rdf:RDF>
```

CONCLUSION

The real world business-to-business integrations are now built on top of XML technologies. In business Applications web search is based on syntactic data. But in the near future, when companies start shifting to a semantic context, there will be a great need for a way to bridge the gap between the syntactic and the semantic paradigms.

Having this problem in mind, we presented a solution for transforming UML/XML data from a widely used standard to a semantic data model. We did it as a proof of concepts and therefore we only propose a solution for common class person. There are many elements that can be used to describe a person as customer also but that are not very common.

Analyzing the results obtained from the mappings becomes obvious that the transformation from UML to OWL is simple and straight-forward but is semi automated. Future work will be based on OCL constraints and also on many other concepts of UML i.e generalization, specialization etc. for conversion and also on mapping of ontologies for reusability.

REFERENCES

1. Baclawski, Kokar, Kogut, Hart, Smith, Holmes, Letkowski, and Aronson. "Extending UML to Support Ontology Engineering for the Semantic Web." UML2001, October 2001b
2. Christian Nentwich1, Wolfgang Emmerich1, Anthony Finkelstein1 and Andrea Zisman “BOX: Browsing objects in XML”
3. Jernej Kovse, Theo Härder “Generic XMI-Based UML Model Transformations”
4. Czarnecki, K., Eisenecker, U.W.:” Generative Programming: Methods, Tools and Applications”, Addison-Wesley, 2000.
5. Krish Narayanan, Shreya Ramaswamy,” Specifications for Mapping UML Models to XML Schemas”
6. Berners-Lee, Hendler, and Lassila. "The Semantic Web," Scientific American, May, 2001
7. Cranefield, S., and Purvis, M. "UML as an Ontology Modeling Language," Proc. of the Workshop on Intelligent Information Integration, 16th Int. Joint Conference on AI (IJCAI-99). Germany, 1999
8. Cranefield, S. "UML and the Semantic Web," Proc. of the International Semantic Web Working Symposium, Palo Alto, 2001
9. Stephen Canefield_, Stefan Hausteiny and Martin Purvis” UML-Based Ontology Modeling for Software Agents”

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application and Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mails i.e. **infoijrcm@gmail.com** or **info@ijrcm.org.in** for further improvements in the interest of research.

If you have any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-

Co-ordinator