Ontology Based Business Rules Extraction Model and Algorithm (OBBREMA)

Maha Osrof
Raid Zaghal
Faculty of Higher Studies, Computer Science Department, Al-Quds University, Palestine

Abstract: Extracting business rules from legacy systems is a difficult process, since business rules are hidden in the code. And legacy systems keep changing all the time. We suggest in this paper to use Ontology, as a conceptual model that represent business rules expressively, for extracting business rules to solve extraction problems. First of all, we did a mapping using analysis and comparison between business rules categories and Ontology concepts to determine what exactly to extract. The case studies show how Ontology represents expressive and real world business rules and they help us in determining relationships between Ontology concepts. We implement the mapping in our own case study. Then we propose the Ontology Based Business Rules Extraction Model (OBBREM) that extracts business rules from Ontology depending on our one to one mapping and the case studies. Finally we propose a translation for our model into an extraction algorithm Ontology Based Business Rules Extraction Algorithm (OBBREA) using backtracking analysis for the case studies. This algorithm helps in extracting business rules from Ontology in expressible way to help software engineers and analysts in the analysis process.

Keywords: Mapping, business rules categories, Ontology concepts, backtracking

1. Introduction

Our motivation in this paper is to help software engineers and system analysts in developing and extracting relevant Business Rules for new applications that belong to a certain domain using the Ontology as a tool. Since Ontology is a conceptual model that expresses a certain domain in terms of a set of well-defined rules, it would be easier and faster to discover the BR related to the target domain and reuse whatever is useful for the new application instead of beginning the analysis process from scratch. Extracting BR has many problems, since it needs many steps and stages, it needs business analysts to be involved all the time which is not worthy in small systems. Extracting BR from legacy systems is difficult because legacy systems keep changing all the time, need to integrate with other systems, and BR are hidden in the code[1]. We need extracted BR to determine requirements for certain applications. “To understand the functions (services) of the current system expressed in the organization’s terms, to maintain business rules in plain English by the organization’s business analysts, to integrate legacy vocabulary and rules with new vocabulary and rules, to reuse in other applications and departments that use the same vocabulary and rules (from the same domain or that use the same data dictionary, to train new personnel on the vocabulary and rules, to support audits for regulatory compliance, to develop system requirements, design validation, and acceptance test specifications for systems based on vocabularies and rules”[1]. When BR are not known very well, the developers can code them and return them back to business analysts for verification. The business analysts most of the times are not satisfied with the results so they spent a long time reviewing them with the developers before accepting the result they got. So extracting the right BR will save time and effort. BR can be used in documentation. They can make maintenance cost of the application lower [20]. Having explicit BR prevents loss of knowledge when employees leave the organization.

In this paper, we produce a solution to the problems associated with extracting BR by suggesting that Ontology as a conceptual model can be used as a tool to extract BR for certain domain by introducing a model and algorithm for extracting BR from Ontology.

We stated the previous work related to extracting BR and Ontology in section 2 then we illustrated the mapping between BR categories and Ontology concepts in section 3. We implement the mapping in qualified teacher case study in section 4. In section 5 we introduce the extraction model and algorithm. Finally we present conclusion and future work in section 6.

2. State of Art

Extracting BR can be a difficult process, extracting such rules from legacy systems have many problems because legacy systems keep changing all the time as described by [1] “legacy systems need ongoing enhancement, desired functionality changes, must integrate with other systems”. Besides, in such systems, BR are imbedded (hidden) inside the code so they become much difficult to extract. Also extracting BR depends on complex and poorly documented software, needs an interactive process with business analysts. Extracting BR is done in three steps” first get business vocabulary, then build rules using vocabulary, interleave activities in practice”[1]. Also analysts need specific tools to extract clues from the code like compilers with level detail across application language then they apply business judgment to formulate business vocabulary in
English and record this vocabulary and code connections. Business analysts need analysis-based code browser or annotation tools to identify BR in English. Extracting BR using some engines such as Business Rules Management Systems (BRMS) needs business analysts to be involved in the whole process which is not worthy in small applications [19].

In [26] the author explained how extracting BR needs several steps and a lot of effort done like interviews with business analysts, reviewing inputs and outputs, examining documentation to map business processes to various program clusters, also transitioning from extracted business logic to BR involves an extracting, filtering and packaging process. Moreover extracting BR from code is not simple because BR have different forms so to extract them we have to classify them as input data, conditions, loops, access rights [3].

In [22] the authors presented a method to extract BR using data as the key that identify the rules since knowing outputs of a given BR makes it possible to determine how those results were calculated and which arguments were used. So their method is backtracking from results till reaching BR. The purpose is to determine for each application what statements change it and affect it and where these statements are located and under what conditions do they work. That is why they considered BR composed of four elements: results, arguments, assignments and conditions.

In [29] the authors produced a new framework to extract BR from large complex legacy systems. This framework consists of five steps: slicing program, identifying domain variables, data analysis, presenting BR and business validation. And they applied their framework on a large financial legacy system.

In [25] the authors developed a tool to extract BR from process specifications written in natural language using a set of linguistic patterns and keywords in addition to grammatical heuristics. This tool includes a natural language parser for working out the grammatical structure of sentences in a documented specification.

In [15] the authors produce a multilingual solution to improve agility in business application by enabling the domain experts to specify BR to the business application directly in many natural languages depending on Elixir MDA Framework and business model.

In [8] the authors invented a new approach for modelling BR called ORM-ML that represents ORM models textually and the syntax of the resulting model is marked-up by XML tags syntax.

To know if we can use Ontology to extract BR we studied what others has done with Ontology. In prior work [4] the authors explain that it is not enough to build an Ontology relying on specialized knowledge engineers only, because this will not reflect the real-world settings, that is why they introduce an Ontology maturing methodology which takes advantage of expert users in addition to specialized engineers.

In [13] the authors suggested the Ontology as a methodology for measuring the correlation between the amount, types and quality of systems engineering efforts used during a program and the success of the program, so this measurement will yield more specific relationships between systems engineering activities, such as requirements management effort, and the cost/schedule compliance of the program.

In [7] the authors created an Ontology to represent user domain in a deposit system for a banking application so their results show that it can construct user-driven software for defending frequent requirement changes.

In [5] authors used an Ontology to provide means to identify and document certain and possibly conflicting interpretations of regulatory requirements.

Authors such [21] introduce Ontology as a tool for extracting information for decision makers by collecting data from different applications in the same domain and summarizing it after making sure it is for the same entity. This can help decision makers in business investment but it was a real problem for them to make sure that the related data is for the same entity.

In [23] the authors introduced Ontology as domain rules contain the semantics of concepts and conceptual relationships of a particular application domain on contrary to data model which represents the structure and integrity of the data elements of a specific application. So Ontology is a general Concept.

In [6] the authors explained that Ontology represent a domain of real world concepts. So this domain must include BR as a part of it. Their analysis of mathematical models of ontology and BR shows that these models are compatible. Therefore, domain ontology can be used to elicit a set of BR. They propose a framework, which can be used for the domain ontology axioms transformation into the Event-Condition-Action ECA rules and then into the active DBMS triggers.

In [28] the authors suggested that Ontology represents BR since it represents real world domain. Their analysis showed that structural assertions are captured by Ontology terms and relationships and the other complex rules are represented by Ontology axioms. They showed this is true by Ontology mathematical definition.

In [12] they represented a mapping between Object Role Model ORM which is a conceptual modelling language used in Ontology engineering. It contains group of constraints can represent an Ontology using rich graphical notation. And OWL 2 DL Web Ontology Language to utilize benefits of both ORM and OWL 2 DL.

A lot of work has been done on Ontology but each one has focused on a separate idea like sharing user information, or using Ontology to know user requirements, or using Ontology to eliminate the ambiguity in interpretation of regulations or extracting information for business intelligence domain, but it was not actually employed the Ontology as a conceptual model to extract BR for specific domain to help analysts and software engineers in analysis process, and to solve problems of extracting BR. This is what we will explain and develop in this study.

3. Mapping Business Rules Categories to Ontology Concepts

Business rules must be expressive enough to capture business complexity and they should be easy and suitable for business analysts to update and maintain [8]. Ontology can serve as a knowledge base that defines BR in an expressive way. In this paper, we did mapping between Ontology concepts and BR categories to see how Ontology expresses each kind of BR. This mapping is depending on our comparison and analysis for the definition of each
business rule and the definition of each Ontology concept. Also this mapping helped us in determining what exactly to extract from Ontology to help software engineers in analysis process.

Business rules fall into four categories: definitions of business terms, facts relating to terms to each other, constraints or action assertions and derivations. The first category describes how people think and talk about things. They are structured BR, and called structural assertions. The terms are of two types: business terms that have a specific meaning for the business in some designated context. For example, business terms in rental car context are booking, reservation, and rental request. The other type, common terms are considered as parts of basic vocabulary, such as, car, city etc. [24]. Terms are represented as entities in Entity Relational Model ERM, but how are they represented in Ontology? Ontology concepts that are represented as classes can be used to express business terms since classes are interpreted as sets that contain individuals which describes entity. They are built up of descriptions that specify the conditions that must be satisfied by an individual for it to be a member of the class. Also classes define or express specific concepts and concepts identify the way of people think and talk about things.

Facts relating to terms to each other can be represented as relationships that assert an association between two –binary relationships- or more terms –N-ary relationships- or as attributes [24]. In Ontology binary relationships that link individuals from one entity (class) to individuals from another entity (class). Ontology object properties used to represent binary relationships. Also Ontology object properties have some characteristics to enrich the meaning of object properties such as functional, transitive, reflexive, irreflexive, symmetric, antisymmetric, inverse property. Also some facts are expressed by compound associations with more than two components that are called N-ary relationships. For example in rental car system “a customer may request a model of car from a rental branch on a date”, this fact includes four terms: customer, car model, rental branch and date [24]. Ontology represents N-ary relationships in different ways depending on the case. For example we may create an additional attribute that describing a relation instance itself, with links from the subject of the relation to this instance and with links from this instance to all participants that represent additional information about this instance[11].

Another case of N-ary relationship links individuals that play different roles in a structure without any single individual standing out as the subject or the “owner” of the relation[11]. We break N-ary relationship to multiple binary relationships.

Another case of N-ary relationship is represented in business rule (Ahmed has temperature which is high but falling) here the relation temperature-observation has two properties temperature-value and temperature-trend.

Another case of N-ary relationship listing relationship when the relationship represented by a sequence of ordered list. Since the order is important we represent it in Ontology using an ordering relation (nextSegment) between instances[11].

Another kind of Ontology properties is datatype properties (attributes that are the other type of business facts) that link individual to a data value of different types as (integer, string, boolean etc.). For example to represent the business fact’ Ahmad’s age is 25’ in the Ontology we used individual Ahmad hasAge 25. Also object properties have hasValue property restriction which “describes the set of individuals that have at least one relationship a long a specified property to a specific individual” [14] and this is mapped to attributes that describe classes.

Constraints on BR are made to prevent some actions from taking place or to prevent a record from being created and they are called action assertions [24]. An action assertion can be either a condition, an integrity constraint or an authorization. First action assertion identified a condition which would be depicted graphically on Entity Relational Diagram ERM, as constraints that constrain a relationship to applying ‘at least one’ or ‘no more than’ one occurrence of an entity. In Ontology these constraints are expressed as property restrictions. A property restriction describes a class of individuals based on relationships of which members of the class participate in [14]. Ontology restrictions are of three categories: quantifier restrictions, cardinality restrictions and hasValue restrictions. First category quantifier restrictions has two types existential restrictions and universal restrictions [14].

Also datatype properties at class level has datatype restriction that is used to specify restrictions on possible values such as specifying a range of values for a number [14]. We can map conditional constraints to datatype restrictions.

Ontology necessary and sufficient conditions on a class that is made when we did properties at superclass level to make equivalent class. These conditions are used to make sure that any random individual that satisfies these conditions belongs to that class. In addition to that all individuals of that class must satisfy these conditions. That means these conditions are constraints to determine what are the constructed individuals in each class. So it is mapped to action assertion integrity constraint. Since an integrity constraint is an assertion that must always be true, it is considered to have immediate enforcement power because it prohibits any actions which would result in a false value [24].

For conditional constraint “if statement” it can be represented in the OWL-based Web Service Ontology OWL-S Language which is extension of the Web Ontology Language OWL. “OWL-S is considered as supporting tools and agent technology to enable automation of services on the semantic web” [2].

Terms are base facts that are given in the world and stored in the system. A derived fact is created by inference or a mathematical calculation from terms, facts, other derivations, or action assertion [24]. For example when we calculate the salary of an employee this is a derived fact from base salary and allowances plus deductions. Derived facts or derivations can be mapped to Ontology axioms. When we define a closure axiom in Ontology we restrict the value for the relationship (object property) to belong to specific domains [14]. So we always can infer and derive that this object property has these specific domains. Also when we make covering axiom in Ontology we can derive new information. Since the definition of covering axiom as follows, having three classes A,B, and C with B and C being subclasses of A. Which means any member of B or C is also a member of A. Making class A covered by class B and C makes us derive that any member of A is also a member of classes B union C. Also class A would be a
super class for classes (B U C). A covering axiom marked itself as a class that is the union of the classes being covered [14].

In addition to that, the characteristics of Ontology object properties such as symmetric, antisymmetric, reflexive, transitive, functional, inverse, irreflexive can help in deriving new information from it as their definitions explained. Table 3.1 explains the summary table for our mapping between business rules categories and Ontology concepts.

<table>
<thead>
<tr>
<th>Table 3.1 Mapping business rules concepts into Ontology concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Rules Concept</td>
</tr>
<tr>
<td>Business Rules Terms and SubTerms</td>
</tr>
<tr>
<td>Business Rules Facts</td>
</tr>
<tr>
<td>Business Attributes</td>
</tr>
<tr>
<td>Conditional Constraints</td>
</tr>
<tr>
<td>Action Assertion Integrity Constraints</td>
</tr>
<tr>
<td>If Statement Constraints</td>
</tr>
<tr>
<td>Derivations</td>
</tr>
</tbody>
</table>

4. Case Study

In this paper, we explain how different types of BR are represented in an Ontology using Protégé tool. We illustrate different case studies to help us in concluding our business rules extraction model by determining relationships between Ontology concepts. But in this paper we focused on our own case study we created it from the teachers domain. This Ontology domain is built to determine if the teacher is qualified or not qualified. We did not focus on results but on the way BR are represented to explain, how easy and clear it is to represent BR in Ontology and to implement the mapping between BR categories and Ontology concepts. Other case studies are already implemented, and we studied how BR represented in them to help us in generalizing our extraction model. These case studies are explained in [18].

In our own case study we chose a specific domain to build Ontology which is called Qualified Teacher. Ontology of Qualified Teacher determines if a specific teacher is qualified or not depending on “Teacher Education Strategy in Palestine” so each teacher considered as a member of our Ontology classes will be a qualified teacher because that means this teacher satisfies the necessary conditions to be qualified and any teacher does not fit to be a member of our Ontology classes will not be qualified.

4.1. Qualified Teacher Main Classes and Subclasses & Necessary and Sufficient Conditions

Our Qualified Teacher Ontology has six main classes which are “Teacher” class, “Grade” class, “University” class, “Certificate” class one of important classes for our Ontology since it holds information about certificates that teachers had and is considered a main factor to determine if a teacher is qualified or not, “Subject” class, and finally “Level” class. These classes are made disjoint from each other which means they don’t have multiple inheritance.

One of main classes is: “Teacher” class, which has five subclasses to express different categories of teachers. These subclasses are determined upon qualification categories for a teacher based on “Teacher Education Strategy in Palestine”. These subclasses are “SecondaryTeacher” that teaches secondary level, i.e. grades 11 and 12 and must have BA degree in science or arts specializing in a subject taught at schools and a diploma in education for secondary level specializing in teaching a specific subject or (MA degree in education). Another subclass “KindergartenTeacher” who teaches pre-school level and must have a BA degree in education with a major in pre-school education. Notice that as interim stage a two-year diploma would be accepted as a certificate for kindergarten teacher. Another subclass “LowBasicLevelTeacher” which must have a BA in education with a major in teaching lower basic level. Also “LowBasicLevelTeacher” is supposed to teach all subjects for grades (1-4). Another subclass, “HighBasicLevelTeacher”, that teaches grades from 5 to 10 and is classified for two programs. “HighBasicLevelTeacherProgramA” must have a BA in education with a major in high basic level and teaching a specific subject such as (Arabic, Math, English, Physics etc…). “HighBasicLevelTeacherProgramB” must have a BA degree in sciences or arts or other (such as engineering for TVE schools), in a subject taught at schools and a diploma in education for high basic level: teaching a specific subject as previous (or an MA).

The last subclass is “AfterSchoolTeacher”. This class defines the teachers who are responsible for training educators and trainers for tertiary level and these teachers must have at least diploma in teacher education specialization in higher or adult education.

“AfterSchoolTeacher” program is to qualify university teachers and adult educators and we made it EquivalentClass to let any random member that satisfies the conditions that is a teacher and has diploma in teacher education specializing in higher or adult education be a member of “AfterSchoolTeacher” class. i.e. change necessary conditions to necessary and sufficient conditions.

In addition to that each member of the “AfterSchoolTeacher” satisfies these conditions. And these BR are determined as explained in the Figure 4.1

![Figure 4.1 Business Rules for AfterSchoolTeacher Class](image)

And here are the BR that inherited from the super class “Teacher” as described in Figure 4.2

![Figure 4.2 Inherited Object Properties for Subclasses of Teacher Class](image)
4.2. Object Properties & Quantifier Restrictions

“AfterSchoolTeacher” and other subclasses of class “Teacher” inherit some relationships (object properties) from “Teacher” super class such as: “specialist in only subject”. Only is “quantifier universal restriction. We restrict the teacher to have specialist relationship with only one subject. Another object property (relationship) “teachesgrades some Grade” which means teacher can teach one or more grades that is why we used “some” which means the relationship existence i.e. the teacher can teach at least one grade since some is quantifier existential restriction.

And other object properties as explained in Figure 4.2. object properties in Ontology are considered the same binary relationships link between entities.

4.3. Cardinality Restrictions

In Figure 4.3 we explain the cardinality restriction (exactly). We define “KindergartenTeacher” class to be restricted to teach exactly one kindergarten grade. Since each kindergarten teacher must be responsible for one grade, in addition to basic qualifications required for kindergarten teacher. Since kindergarten teacher must have BA in education with major in pre-school education or temporary a two year diploma.

4.4. Value Partitions

Also BR could be limitation for some values as “PromotionValue” class describes some values for subclass “ExpertTeacher” regarding years of experience. Expert teacher must have at least five years of experience as it is described in Figure 4.4. and novice teacher has years of experience less than five years. This can be done by creating value partitions. Value partitions can be created to refine classes descriptions [14]. In our case study we wanted to describe the experience period for teacher. We did this by creating a “ValuePartition” class as a sub class of Thing. Then we created a “PromotionValue” as a subclass of “ValuePartition” to restrict the range of possible values for teachers regarding their experience. Teachers were classified for two categories regarding experience years, expert teachers and novice teachers, we did this using cardinality restrictions as its shown in Figure 4.4.

4.5. Data Properties

In Ontology attributes (data properties) are represented as relationships at the level of individuals (members) of class. For example teacher (Ahmad hasname “Ahmad ali Othman”) and his age is 25 as its clear in the data property (hasage 25).

To generalize this data property we make it at class level. For example the data property (hasname some string) and the data property (hasage some integer).

Also we can use data properties restrictions to define ranges for classes. For example we have another datatype or data property called hasyearsofexperience. We restrict expert teacher to have years of experience more than five years and we restrict novice teacher to have years of experience less than five years. It is important to know that Data properties are supposed to be functional. For example teacher cannot have two or more ages, or two or more names etc.

4.6. A Closure Axiom and Covering Axiom

We use closure axiom on property to restrict the property with universal restriction to say that it can only be filled by the specified fillers [14]. For example in our Ontology, a teacher can teach different levels. In “Level11-12” the only grades must be taught are eleven and twelve grades. To restrict these grades only we make a closure axiom.

covering axiom can derive new information. In our case study we covered class “PromotionValue” by the classes “ExpertTeacher” and “NoviceTeacher”. We can derive that to be a member of “PromotionValue” class it must be a member of “ExpertTeacher” class or “NoviceTeacher” class

4.7. Has Value Restrictions

Has value restriction which is used to describe the set of individuals that have at least one relationship along a specified object property to a specific individual [14] we have example for it in[18] in Pizza Ontology. other ready implemented case studies are explained in [18].
5. Extraction Model and Algorithm

5.1. Extraction Model

Figure 5.1 shows our suggested model for extraction of business rules. Ontology Based Business Rules Extraction Model (OBBREM) represented a road for parsing Ontology code to extract business rules. We draw this model depending on our mapping process between business rules categories and Ontology concepts, which helped us in determining what we need to extract from Ontology to get business rules. Also our extraction model depend on our own case study and other ready implemented case studies that explained in[18] which helped us in determining the relations between different Ontology concepts, as a consequence we determine how business rules are related to each other. We used Entity Relational Model (ERM) conventions in modelling OBBREM with few modifications such as we do not need cardinality or ordinary relationships between entities, we eliminate some concepts such as primary key and foreign key. We used ERM to link concepts to each other. We inferred these concepts and relations between them depending on deep analysis for how Ontology represents BR and how they are mapped to Ontology concepts.

OBBREM suggested extraction for classes of Ontology. Classes represent one of business rules structural assertions which is terms. As shown in Figure 5.1 class can have subclasses and we must extract subclasses to know the hierarchy in terms. Classes and subclasses can be primitive or defined. Primitive classes only have necessary conditions. That means members of these classes must satisfy these conditions. But defined classes have in addition to necessary conditions sufficient conditions. This guaranteed any random member satisfies these necessary and sufficient conditions can be a member of the class[14]. And this information helped in determining what are the exact conditions and constraints to define business rules terms and sub terms. A class can have disjoint property. It determines the relation between sibling classes, and it means that the member of a class cannot be a member of another class. And from this information we can determine how terms have multiple inheritance.

As shown in Figure 5.1 classes have object properties which represent the second category of BR structural assertions which is facts. facts related terms to each other. Facts expressed binary and N-ary relationships between terms. And as its shown in the OBBREM object properties (relationships) either binary or N-ary has property restrictions. These restrictions define the relationship as existential, universal, the cardinality of the relationship and hasValue restrictions. Existential restriction is represented by “some” key word and it can express the words “should” be” in BR. The other property restriction is universal restriction. universal restriction is represented by “only” key word and it can express the words “must be” in BR. Another property restriction cardinality restrictions that specified the relationship cardinality that explained numbers of individuals of each class participated in the relationship. So we can extract relationships and their cardinality, in other way relationship constraints from object properties and their cardinality restrictions. The last kind of property restriction is hasValue restriction which “describes the set of individuals that have at least one relationship a long a specified property to a specific individual” [14] and this helps in extracting attributes that describe classes. BR like X <connecting verb> Y, X contains Y could be extracted from object properties.

Also Its shown in Figure 5.1 object properties have some characteristics such as symmetric, reflexive, transitive, functional, inverse, irreflexive or antisymmetric. And we can extract business rules derivations from these information. For example if property P is reflexive we can derive information from that P can be related to itself. The same for other characteristics of object properties we can derive new information depending on property characteristic definition.
We can extract another derived information i.e. business rules derivations from object properties. Since object properties have closure axioms that help in deriving new information. "a closure axiom on a property consists of a universal restriction that acts along the property to say that it can only be filled by the specified fillers" [14]. We can derive that the restriction has a class (filler) that is the union of the classes that occur in the existential restrictions for the property.

Also its shown in Figure 5.1 the class can have a covering axiom and we can derive new information from that depending on the definition of covering axiom. And this helps in knowing how business terms affected by each other.

Also as its shown in OBBREM classes have datatype properties(attributes). Datatype properties represented as attributes in BR. attributes that describe each class can be extracted from datatype properties. Any business rule like X is A type of Y could be extracted from datatype properties. Datatype properties could be defined at class level for all individuals. Or it can be defined at individual level as explained in Qualified Teacher case study.

Also datatype properties at class level has datatype restriction that is used to specify restrictions on possible values such as specifying a range of values for a number [14]. We can extract constraints from datatype restrictions.

### 5.2. Extraction Algorithm

Our OBBREM model can be translated into algorithm that is used for extracting BR from Ontology. This algorithm is inferred from the case studies and the mapping between business rules categories and Ontology concepts. Since we know how to represent each business rule in Ontology so using backtracking analysis we can infer how to extract BR into algorithm called Ontology Based Business Rules Extraction Algorithm OBBREA.

```plaintext
Begin of algorithm
Parse OWL Code
Loop
  Look for classes and check for disjoint property to determine multiple inheritance
  Begin
    If defined class then
      Begin
        Print class name
        Print "this class has necessary and sufficient conditions that means any random individual satisfy these conditions would be a member of the class "
      End
    End
  End
Elseif Primitive Class then
  Begin
    Print class name
    Print "this class has necessary conditions only which means all members of the class must satisfy these conditions"
  End
Elseif N-ary Object Property
  Begin
    Look for Object Property that related to the class
    If binary Object Property then
      Begin
        Look for characteristics of Object Property
        Print derived business rules depending on characteristics
        Definitions
        Look for Property Restriction
        If Property Restriction is Existential then
          Begin
            Look for related filler class
            Print class name " " Object Property "" 'some' "" filler class
          End
        Elseif Property Restriction is Universal then
          Begin
            Look for related filler class
            Print class name " " Object Property "" 'Only' "" filler class
          End
        Else if Property Restriction is Cardinality
          Look for related filler class
          Print class name " " Object Property "" Cardinality Value "" filler class
          End
        Else if Property Restriction is hasValue
          Look for related filler individual
          Print class name "" Object Property "" 'Value' "" filler individual
          Print attribute (filler individual) related to the class
          End
      End
    End
End
End
```

### 6. Conclusion and Future Work

We proved that Ontology can be used as a tool to extract business rules easily. We suggested a model for extraction BR that Ontology Based Business Rules Extraction Model (OBBREM), we concluded our model with a mapping between business rules categories and Ontology concepts and the case studies. Our model shows that entities in Figure 5.1 can be considered as key words or clues for parsing Ontology to extract BR without interfere of business experts. Case studies showed that business rules are not hidden in Ontology code. Ontology separates
business rules from the data and presentation layers which helps in extracting these business rules. Business rules are represented in Ontology in conceptual way. they are expressed in Ontology as they are expressed in real world. So extracting them from Ontology ease the process of maintaining and modifying them by analysts, business rules experts and software engineers. Then using backtracking analysis we proposed an algorithm for extraction BR that is Ontology Based Business Rules Extraction Algorithm (OBBREA). In future we intend to provide a tool that could parse Ontology depending on our (OBBREA) to extract BR. To help in the analysis process for any new application belongs to certain domain. So a new application will not be built from scratch. It will use the domain business rules.

References


[20] Saggion Horacio, Funk Adam, Maynard Diana, Bontcheva Kalina, “Ontology-based Information Extraction for Business Intelligence”, Department of Computer Science, University of Sheffield, United Kingdom, 2007


