ISSN: 0000-0000 (print), 0000-0000 (online)

AIMS Journal

Journal homepage: www.jdconline.net/aims

Regular Paper

Automated Translation of SBVR Queries to SQL using Case Based Reasoning

Tooba Nadeem[a], Judith Symonds[b]

[a]*Queens Academic Group, Auckland, New Zealand*

ARTICLE INFO

Article history:
Received 11 December, 2016
Revised 21 March, 2017
Accepted 11 April, 2017

Keywords:
SBVR Queries
Automatic Query Generation
Natural Language Processing

ABSTRACT

Software The proposed framework describes that as SBVR (Semantics of Business Vocabulary and Business Rules) presents the Natural Languages with formal presentation and specification. A lot of work already been done in transforming business designs into SBVR, as well as transforming UML models into SBVR specification. It leads towards enhanced approach of transforming SBVR specification and presentation to generation of databases. The adopted approach in this research activity aims to reduce ambiguities and uncertainty from natural languages in describing business requirements. Utilization of Controlled Natural Languages (CNLs) that are subset of natural languages is enhanced methodology limiting vocabulary and grammar in order to eradicate ambiguities and intricacy. The business analyst or people can state business rules with formal syntax and semantics. An automated mechanism supports the designed system to convert the natural language query into SQL command which will work as interface for database. This all work will be done by using SBVR standard for presentation of business semantics and business rules.

1. Introduction

In business perspective, considerate endeavours are put to forth to capture the organizational requirements. To establish the integrity and consistency among business processes, goals, resources and constraints it has become indispensable for origination of "Entrepreneur modelling" or "business modelling". Although beyond business modelling it is better approach for business policy makers to formulate the enterprise rules and other standard observance.

In order to automate the transactional activities of enterprises directs towards implementation of database management system. Database system supports the core businesses of organization and helps in decision making and strategic planning and day to day operations. Moreover, with the presentation of all business data within databases and maintenance of database management system it evolves the necessity of presenting business language into some formal manner. Natural languages format in business designs has been transformed into more formal presentation. For accomplishing this significant task, a novel standard SBVR (Semantics of Business Vocabulary and Business Rules) [1] is employed for developing semantic models of business vocabularies and business rules. The contemporary approach also assists in building up of enterprise modelling and automation of business processes as well. Apart from this, the presentation of Natural Language into formal SBVR standard enables to make business rules accessible to software tools. It also stimulates the business experts in managing business rules ultimately translating these rules into generation of automated system.

The prime motivation of following research activity is to write the business specifications and complex entities into formal and detailed natural language. Furthermore, development of automated system that helps to generate database queries from underlying database used. The standard provided for writing of business specifications is SBVR Semantics of Business Vocabulary and Business Rules. SBVR depicts a formal structure of business semantics and rules. Formerly, in business application modelling it was common practice that business requirements were identified and business analyst

* *Corresponding author.* Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: *toobadotcom@hotmail.com*

http://dx.doi.org/......

captured these requirements in plain English [2], [3]. Whereas, the rules presented in plain English cannot be run on machine. Because the rules in plain English having informal presentation and syntactically inconsistent. To avoid inconsistencies and contradiction SBVR provides recent approach for writing business rules into formal English. They are easily machine processed yielding business object models, databases and software components.

The adopted approach in this research activity aims to reduce ambiguities and uncertainty from natural languages in describing business requirements. Utilization of Controlled Natural Languages CNLs which are subset of natural languages is enhanced methodology limiting vocabulary and grammar in order to eradicate ambiguities and intricacy. The business analyst or people can state business rules with formal syntax and semantics [4]. OMG 2008 facilitates the business owners to classify their business requirements and business rules into most recent approach i.e. SBVR which is syntactically consistent and semantically formal. Defining business rule in SBVR yields toward development of automated software tools in future. In order to retrieve information from automated system, it provides facility of querying by using some structured language with combination of underlying database. Moreover, the automated software enables the novel user to generate SQL queries that ultimately process on machine. The information retrieval includes inferences from database management system of automated system by applying few AI Artificial Intelligence techniques.

The Semantics of Business vocabulary and Business rules specifications is applicable to the domain of all kinds of businesses as well as all types of business activities in all organizations. SBVR gives a detailed description of meanings of business words and business statements. The foundations of SBVR are formal logic and mathematical logics. This standard is recommended specification of OMG and provides base for formal and natural languages commonly used in businesses. SBVR formalize the operational rules of businesses. It also formalizes the business rules and business vocabularies, further these rules and vocabularies are interpreted by computer systems. These rules may be operational rules and security policies of business. SBVR covers two specification concerning business, these are Business vocabulary and Behavioural guidance. The Business vocabulary comprises of "concepts" which are taken as "nouns". Furthermore, the "names", "definitions" are also considered under this category. In addition to this the "Behavioural guidance" includes operational rules of business. These rules are policies and permission that governs the action of organization.

The core intention behind this research work is to transform the business rules and specification into generation of database query. Structured Query Language (SQL) is best recommended match with databases in order to get access to stored business data. The proposed research methodology works out and yields good quality results in decision making environment. The adoption of Case Based Reasoning CBR approach enables clients and benefactor to get access to confirmed solution of targeted problem among different similar cases.

1.1. SBVR Vocabulary

SBVR vocabulary is like content of natural language but different in many ways. By applying the SBVR specifications on business, we can develop business software tool. The tool enables to specifying, managing and interchanging the business vocabulary and behavioural guidance as a business asset. The terms and rules of business are documented in online database of organization. The basic elements of SBVR are Terms, Names and Fact types. Terms are nouns or group of words which represents the business entities. Name represents the Instances of Terms. Fact type gives the relationship between terms and names. It has template: term-verb-term.

SBVR vocabulary composed of few Meta model parts. Business specifications written in SBVR formal format based on Meaning and Representation vocabulary, logical formation of Semantic variable and mathematical and formal logics. Meaning consists of concept, noun concept and object type. Whereas, concept type is related to concept by being an instance of concept. It is necessary that the set of characteristics are incorporated by an object type is not the set of characteristics that is incorporated by another object type. The concept type can appear as an instance of concept type, fact type, and role.

1.2. Logical Formulation of Semantic Vocabulary

Logical formulation denotes formal semantic structure of business vocabulary including semantic structure of underlying business specifications. SBVR provides the structure of meaning of rules written in natural language. The meaning of statement in SBVR is called facts. The Semantic formulation has two types. First kind is logical formulation which is simple as well as complex depending upon its definitions or statements. Furthermore, different logical operations are carried out on its propositions these operations include quantifications, atomic formulations based on fact types and other formulations named objectifications and nominalizations.

Semantic formulation uses recursive approach. Several semantic formulations are enclosed with other semantic formulations. Quantifications and objectifications are represented as logical variables so that they may also refer to instances of concepts. Only the closed formulation has the "meaning". The hierarchal structure of embedded formulation in a simple business rule is given by an example as shown in Table 1.

Table 1: Extracting information from SBVR rule.

Example 1	
Plain English	A customer has to place at least one order.
SBVR English	It is obligatory that each customer has to place at least one order.

The semantic formulations represent the rules as sentences that give the full structure of meaning. The different semantic formulation may have same meanings. In the above given example (see Table 1) the "customer" and "order" both are concepts. So identifying one concept with another concept doesn't change the semantic formulations.

SBVR Rules: Rules play a very important role in defining business semantics: they can influence or guide behaviours and support policies. This means that rules represent the primary means by which an organization can direct its business, defining the operative way to reach its objectives.

The rule based approach states rules which are built on Facts and facts are built into concepts expressed in terms.

The Structural Business rules state:

```
It is necessary that
It is possible that
```

The operative Business rules are stated:

```
It is obligatory that
It is permitted that
```

The SBVR specification and presentations are based on “Rules” and these rules are based on fact types. The business rules are based on fact types and concepts are associated with fact types gathering these elements we develop a business vocabulary to represent them. The “Rules” defined in SBVR are distinct from one another. We attach a distinct rule with every business policy. It is said automating a “Rule”. In automating the Rule a policy is enforced with it.

1.3. Case Based Reasoning and Decision Making

Decision making is part of cognitive process that enables a user to select best possible result for the targeted problem and Case Based Reasoning (CBR) is a typical approach that is used for decision making in computer science.

Case Based Reasoning (CBR) is an intelligent system that enables the IT professionals to automate the problem solutions like diagnosis, scheduling and designs. The newly launched problems are solved by matching of cases from underlying databases and adapting worthwhile solutions to the problems. In paradigm of problem solving, to find out the solutions of new problems on different domains requires distinctive approach. Case Based Reasoning is an approach for finding the solution of target problems by mapping similar problems already stored in case library or repository. Case based reasoning method is employed with different decision making domains to attain effective strategic planning. CBR method also cooperates to evaluate and select confirmed solution among numerous similar cases. The CBR concept is perceived as prototype theory which enhances the human problem solving and deals in cognitive approach. In addition to this the K-nearest neighbour K-NN approach is applied to find the similarity of already stored cases to target problem. CBR is continuous process of learning as new experience is retained as new problem is solved. The new experience is saved as case into case library.

The significance of Case Based Reasoning approach in field of software engineering field is determined with the emergence of Computer Aided Software Engineering tools [2]. The document introduces the adoption of CBR approach in CASE tools yield UML diagrams from natural language text.

The rest of the paper is structured as follows. Section 2 describes the methodology for transformation of SBVR rules into generation of SQL query. This transformation will be accomplished by imparting another approach Case Base Reasoning (CBR). All the transformation rules are illustrated by examples. Section 3 describes a prototype transformation tool AUTO_SQL_Gen. whereas section 4 describes experiments, results and analysis. Section 5 presents the related work to our research and the paper is closed with the conclusion section.

2. SBVR Rules to SQL Query Generation

The process of transformation of SBVR rules into SQL query will Transformation of SBVR rules into SQL query generation will be performed in two steps. In first step, the input string written into SBVR formal English text will undergo lexical and semantic parsing to yield SBVR keywords. After extracting SBVR elements like Terms and Fact types are indexed into CBR case library as sample case. The first step is further divided into few numbers of modules. Descriptions of both steps are given in subsequent sub sections.

In first step the input SBVR statements will passes through different modules. The string of English text in Structured English SBVR will be treated as input. Generation of SQL query from SBVR specifications and requirements, against underlying database will be workable under linguistic analysis. As our concern is with SBVR structured English formal natural language. Syntax and Semantics is related to our research subject.

The mapping process starts with the syntax analysis of SBVR rules to extract various elements of the SBVR rule i.e. noun concepts, verb concepts, fact types, etc. Following section describes the process of mapping classes and their respective associations with a common SBVR rule.

2.1. Parsing of SBVR Query

The NL parser behaves just like an intelligent system, natural language engine performs task of translating formal English sentences into SQL query with respect to specific case already keep in knowledge base. The system takes input in the form of structured English SBVR and forwarded it to NL engine for parsing. Consequently, each word in sentence will be labelled with tag, furthermore the tagged sentence assumed for grammatical class. In next step, sentence is supplied to syntactic parser which produce possible parse tree from the sentence. The obtained precise parse trees are then transferred to semantic analyzer of CBR, where it will be transformed into SQL query. The output produced in the form of SQL query is based on input structured English SBVR text and it will be stored with other case solution for future requirements to particular cases.

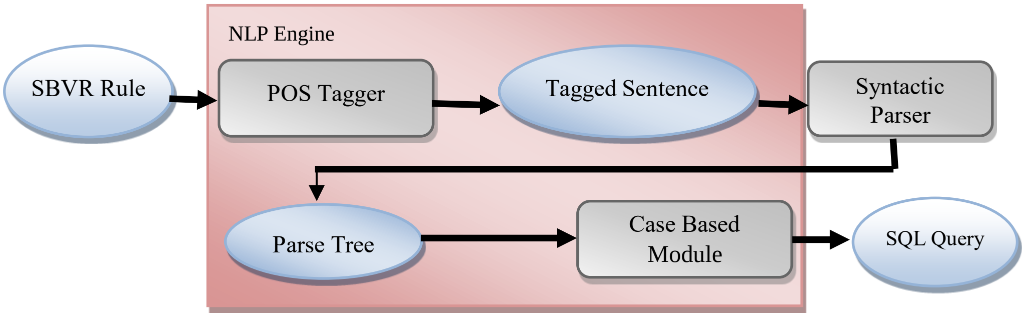

Fig 1: Architecture of prototype tool: AUTO_SQL_GEN

Figure 1 shows modular description consisting of NLP engine of NL-Parser. Figure 1 also illustrates the process of translating input sentence. It requires sentence as input to system i.e. structured SBVR English text. The sentences written in SBVR structured English are forwarded in arranged manner to NLP engine for processing. In next step, words in sentences are tagged with Label. These labels represent the grammatical class of that word. Tagged sentence will be further provided for syntactic Parsing. This module extracts the all possible parse trees from tagged sentence. Among variety of parse trees, select the desired parse tree and pass on to case base semantic analyzer. Consequently, this module generates SQL query from the input structured SBVR English. This is partial diagram which will be combined to original one.

2.2. Lexical Analysis

Lexical analysis is the first phase in processing of natural language based SBVR queries.

Tokenization

A process of identifying token in a NL sentence is called tokenization. During tokenization the information such like total words in NL sentence and token's position in the text.

Parts of Speech Tagging

During POS tagging it may come across some unknown or inflected words. Inflected words refer to grammatical variants, as well as modification of words corresponding different grammatical categories. The sentence will go through Morphological processing for identification of inflected words. After removal of inflected words from tagger lexicon, the sentence will be processed again under NTAG. In case of incorrect tagging of sentence, the unknown or inflected words will be tagged manually. We are using Stanford POS tagger assigning correct parts of speech written in formal SBVR text i.e. noun, verb to words. Let us have a sentence in SBVR formal English text the lexicon will be tagged in following manner. For example, the query will be entered in SBVR formal English text such as "It is necessary that each student is registered for at most seven courses". The Stanford POS tagger will assign a tagger each word in input SBVR sentence.

2.3. Syntactic Parsing

The next phase of transformation of SBVR text into SQL query is Syntactic parsing. The Syntactic parser determine maximum parse tree for given sentence. Basically, syntax specifies rules to arrange words in a way to build a sentence. The grammar represents the specification of a sentence. The syntactic parser works together with context free grammar for English language. Furthermore, the parsing of SBVR formal English text will be accomplished through the use of parser. The parser employed in our used methodology is Stanford Parser. For example, we will input query into SBVR formal English text the Stanford parser will generate parse tree for given query such as "It is necessary that each student is registered for at most seven courses".

2.4. Case Based Reasoning

The CBR is contemporary approach for designing of Knowledge –based systems in the subject of AI Artificial Intelligence. CBR comes with reusability feature, once a case is resolved with specific methodology it can be revived for another similar case.

By adopting CBR methodology in our newly proposed system i.e. automatic generation of SQL query from underlying database. The designed system uses CBR approach to semantically analyze the input text. After structuring of lexical from input string, it will be parsed into grammatical words.

2.5. Extracting SQL syntax by Adopting CBR Approach

By adopting CBR methodology in our newly proposed system i.e. automatic generation of SQL query from underlying database. The designed system uses CBR approach to semantically analyze the input text. After structuring of lexical from input string, it will be parsed into grammatical words.

For acquiring SQL query from SBVR structured English we devise a prototype tool named AUTO_SQLGEN. The AUTO_SQLGEN behaves like an intelligent system depending upon underlying knowledge base for automatic generation of SQL query.

The proposed approach consists of a set of four main modules SQL Editor, Knowledge Base manager, Knowledge Base and CBR Engine as shown in Figure 2.

2.6. SQL Editor

The SQL Editor will be responsible of creating SQL query while knowledge Base manager acts like as Database management system. The key role of KB administrator is to keep and maintain Knowledge base updated and consistent. The SQL editor is the interface between AUTO_SQLGEN and database. The KB manager is interface between KB administrator and our proposed system.

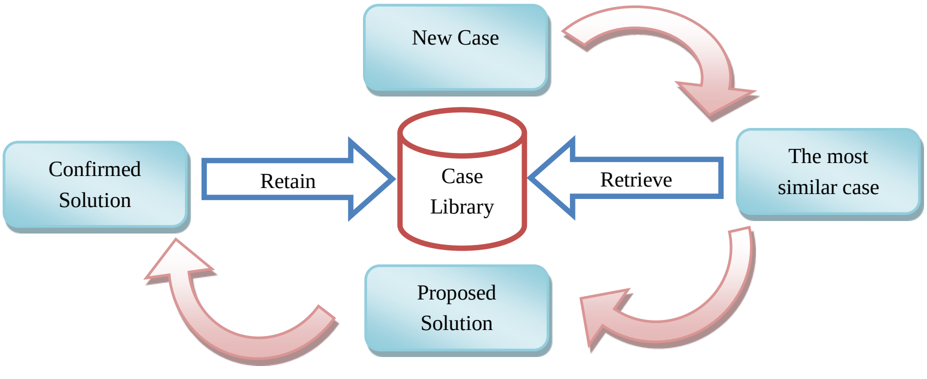

Fig 2. Storing and retrieval of cases in Case Library.

The integrated development environment of SQL editor lets the user to write queries in formal English which can consequently make amendments in retrieving query from Knowledge base. The knowledge base of our system will be capable of storing the results of previous/frequent/recent cases. These cases are input text strings already converted into SQL query by the predefined algorithm. Indexed memory is used for retrieving the case effectively and accurately.

The adopted CBR approach will generate the SQL query using historic knowledge stored in KB. The used approach consists of seven sub module which retrieval, design composition, design patterns, Analogy, verification learning. The retrieval sub module will access the case with respect to similarity with problem. Design solution gathers different pieces of cases to accumulate into construction of new solution. The Analogy sub module will match the problem and selected cases, the case verification module capable of checking the consistency and of system. The Last module Learner is trained to be taught new cases.

All necessary requirements (for example input string in formal SBVR text) are kept and indexed as cases into case library. The NL_ parser cases comprises of parse tree containing lexical of sentence in its leaves, as well as relevant SQL query and underlying database. These cases are then indexed according to the verbs present in the sentence. Each verb is attributed a subset from DBNET, then an index representing the case is created and attached to the corresponding subset node in DBNET. The use of verbs for indexing is due to the fact that the verb provides a relational and semantic framework for its sentence. Basically, the verb occupies a focal position in the sentence, and no valid sentence may exist without a verb.

The pre-requisites for attaining the solution of problem through CBR approach is problem description, which evaluates the current problem with subsequent problems stored earlier in Case Base with their known solutions ((R. Lopez., Et Al 2005). The retrieval and reuse of the problem will be modified according to variation in current problem. After necessary evaluation, the proposed solution of current problem will be finally revised. Consequently, the problem and its proposed solution will be retained as new case.

The retrieval of already stored cases by using CBR approach based on problem solving cycle [6]. The retrieval cycle consists of “REs” that is retrieve, reuse, revise and retain. While adopting this approach, the retrieval and similarity assessment will be centrally focused. The diagram shown below depicts the working of CBR cycle and also presents function of similarity assessment and adaptation in retrieval process.

An important aspect of Case Based Reasoning CBR approach in natural language processing is recalling of problem and its solution to resolve other similar problems. The main problem would be resolved through retrieval of previous problem cases.

3. Tool Support

For acquiring SQL query from SBVR structured English we devise a prototype tool named AUTO_SQLGEN. The AUTO_SQLGEN behaves like an intelligent system depending upon underlying knowledge base for automatic generation of SQL query. Our proposed approach consists of four main modules SQL Editor, Knowledge Base manager, Knowledge Base and CBR Engine. The SQL Editor will be responsible of creating SQL query while knowledge Base manager acts like as Database management system. The key role of KB administrator is to keep and maintain Knowledge base updated and consistent. The SQL editor is the interface between AUTO_SQLGEN and database. The KB manager is interface between KB administrator and our proposed system.

The integrated development environment of SQL editor lets the user to write queries in formal English which can consequently make amendments in retrieving query from Knowledge base. The knowledge base of our system will be capable of storing the results of previous/frequent/recent cases. These cases are input text strings already converted into SQL query by the predefined algorithm. Indexed memory is used for retrieving the case effectively and accurately.

4. Experiments and Results

The description of problem statement of College Registration System shows the performance of AUTO_SQLGEN. This problem statement is adhered to the issues that have been resolved during testing of prototype tool.

“The system offers the benefactor that may be a student to register his/her self for maximum seven courses. The student on trial is those who are on probation or being tested. Student on trial is allowed to register only for five courses not more than this. During registration process regular student specifies his/her name, section and section code also. Registered students are directed to join the section after registration process. The system does not allow the student to select courses without registration. If a student wants to retrieve information about courses for which he/she being enrolled must specify student name, section code and course title. The system will respond consequently into formation of SQL query and returning the resulted dataset. Each SBVR rules will be transformed into SQL query and underlying schema will construct.”

Table 2: Extracting information from SBVR rule.

SBVR rules	Fact type	Terms
It is necessary that each student is registered for at most seven courses	Student is register for course	Students
	Student has joined Section	Section
It is necessary that each student on trial register for at most five courses	Section is accessible for course	Course
It is obligatory that each student has exactly one name	Student on trial	Name
It is obligatory that each course has exactly one title	Student has name	(concept -type) varchar 255
	Course has title	Code concept – type
		Varchar 255
It is obligatory that each section has exactly one code	Section has code	Title concept – type
		Varchar 255

4.1. Formation of SQL query and its retrieval through CBR approach

After extracting the terms and fact types from SBVR formal English, the outcomes mainly consists of SBVR elements are indexed into case library as sample case. According to our prototype tool requirement, the terms extracted from SBVR rules are further mapped to SQL Keywords. The relation between Student and Name and Title and Course has one-to-one cardinality. The AUTO_SQLGEN will generate SQL query of SELECT keyword.

The NTAG() will match the target problem with already stored cases in case library. After mapping to the extracted SBVR elements are utilized as constructs of SQL query. The formation of SQL query is presented below.

```
CREATE TABLE STUDENT (ID INT NOT NULL AUTO_INCREAMENT NAME
        VARCHAR(255)  NOT NULL   IS ON TRIAL  BOOL PRIMARY KEY (ID));

CREATE TABLE COURSE (ID INT NOT NULL AUTO_INCREAMENT NAME
        VARCHAR(255)  NOT NULL   IS ON TRIAL  BOOL PRIMARY KEY (ID));

CREATE TABLE SECTION (ID INT NOT NULL AUTO_INCREAMENT TITLE
        VARCHAR(255)  NOT NULL PRIMARY KEY (ID));

CREATE TABLE STUDENT_JOINED_SECTION (STUDENT_ID NT SECTION_ID INT
         PRIMARY KEY (STUDENT_ID, SECTION_ID)
         FORGEIN KEY (STUDENT_ID) references  STUDENT(ID)
        FORGEIN KEY (SECTION_ID references SECTION(ID));

CREATE TABLE STUDENT_IS_REGISTERED_FOR_COURSE (STUDENT_ID INT, COURSE_ID    INT
        PRIMARY KEY (STUDENT_ID, COURSE_ID),
        FORGEIN KEY (STUDENT_ID) references STUDENT (ID),
        FORGEIN KEY (COURSE_ID) references COURSE (ID));
```

```
CREATE TABLE COURSE_IS_ ACCESIBLE_FOR_SECTION (SECTION_ID INT, COURSE_ID INT
        PRIMARY KEY (COURSE_ID), SECTION (ID)
        FORGEIN KEY (COURSE_ID) references COURSE (ID)
        FORGEIN KEY (SECTION_ID) reference SECTION (ID))
```

The above listing of SQL DDL shows the Schema of resulting Schema.

4.2. SELECT query

In order to manipulate the SBVR formal text this has been broken down into SBVR elements are further mapped to SQL Keywords by employing SQL commands like SELECT command. It may acquire the following formation:

```
SELECT STUDENT_NAME  AS [STUDENT NAME],
COUNT (COURSE_ID) AS NUMBER.OF.COURSE
FROM STUDENT, COURSE,  STUDENT_IS_REGISTERED_FOR_COURSE
WHERE STUDENT_ID = STIDENT_IS_REGISTERED_FOR_COURSE
AND  STUDENT_IS_REGEISTERED_FOR_COURSE.COURSE_ID =
          COURSE_ID
GROUP BY STUDENT_ID
HAVING COUNT(COURSE_ID)>7
```

5. Related Work

The main objective of Object Management Group is to eliminate the differences between domain experts and business software. . The aim to construct business models is to answer the interrogations related to business problems. Business process is steps of activities to achieve particular output for domain experts.

The information presented in the paper [3], demonstrates the integration of SBVR into Model Driven Architecture MDA. This paper discusses the main concepts of SBVR and their effects on MDA. He also discusses the behaviours of facts utilized in conceptual schema. The Modelling languages implemented in designing of information system, enforces different concepts and facts. The transformation of SBVR into MDA is step by step mappings and shown in schema. Conceptual schema of information system exactly defines database schema, object model. It also leads how to implement business rules. The distinctive approach for translating natural languages into SBVR business rules requirements direct to complex semantic analysis of English Language [4]. The business rules directs the business process and its hard task to write business rules into natural language and then further translating them into OCL or SBVR specification. The Semantic formulation of SBVR, written in structured language is transformed into Unified Modelling Language UML diagrams which are Activity diagram, Sequence diagram and Class diagram [5]. Furthermore, this paper describes how the SBVR designs are transformed into business modal architecture like Platform Independent modal PIM. During the transformation of SBVR vocabulary into PIM, main focus should be given to automation of rules. The automation of rules means enforcement of business rules.

The "Generative Information System" research based subject [6] focuses on relationship between SQL and SBVR models. Describing the entire architecture of system that executes SBVR model needs to represent how SBVR rules are automatically transformed into SQL Data Manipulation Language DML that are automatically verified against database containing SQL query. The need of development of Intelligent Database System IDS helps invoice to put query from relational database in natural language [7]. It leads towards the development and designing of intelligent system referred as "Natural language interface to Database". This paper describes in detail the architecture of NLIDB and introduction to its sub components. Furthermore, the adopted methodology and approaches is also presented.

6. Conclusion

This research paper presents a framework that automatically transforms the SBVR specification and requirements into generation of database query using Case Based Reasoning approach. For this, our developed AUTO_SQL_GEN is trained enough to take user requirements into SBVR and consequently performs semantic analysis on input formal English text. Finally the SQL query will be generated through CBR approach. The maturity of tool is demonstrated with evidence of case studies. A mechanism will automatically generate the database query by processing the natural language in terms of business semantics and business rules. The resulting database have interface with SQL, so SQL command can easily manipulate the database. In future algorithms are needed to improve the effectiveness and efficiency of system.

REFERENCES

[1] OMG: Semantics of Business vocabulary and Rules (SBVR), OMG Standard, v. 1.0, (2008)

[2] Seco. N., Gomes. P., Pereira. F.C., (2004), Using CBR for Semantic Analysis of Software Specifications, Lecture Notes in Computer Science Volume. 3155, pages: 778-792

[3] Hendryx. S., (2005), Model Driven Architecture and Semantics of Business Vocabulary and Business Rules.

[4] Bajwa. I.S., Lee. M.G., Bordbar. B., (2011), SBVR Business Rules Generation from Natural Language Specification. AAAI Spring Symposium Series.

[5] Raj A., Prabharkar T., Hendryx S.: Transformation of SBVR Business Design to UML Models. In ACM Conference on India software engineering, pp.29-38 (2008)

[6] Moschoyiannis. S., Marinos. A., Krause. P., (2010), Generating SQL Queries from SBVR Rules. International conference on semantic Web rules, Pages 128-143.

[7] Nihalani. N., Motwani. M., Silakari. S., (2011), Natural Language Interface to Database using Semantic matching. International journal of computer applications 31(11) pages 29-34.

[8] Cabot. J., Pau. R., Raventos. R., (2010), From UML/OCL to SBVR Specifications: a Challenging Transformation. Information System archive, Volume 35, Issues 4, Pages 417- 440

[9] Nicolae. O., Wagner. G., (2008), Verbalising R2ML rules into SBVR. SYNASC 10th International symposium on Symbolic and Numeric Algorithms for scientific computing, Pages 265-272.

[10] Aqueo. K., Governatori. G., Sadiq. S., (2010), SBVR based Business Contract and Business Rule IDE. RuleML-2010 Challenge, at the 4th International Web Rule Symposium. CEUR Workshop Proceedings 649.

[11] Karanade. N.D., Patil. G.A. (2009), Natural Language Database Interface for Selection of Data Using Grammar and Parsing. World Academy of Science, Engineering and Technology 59.

[12] Nihalani. N., Motwani. M., Silakari. S., (2011), Natural Language Interface to Database using Semantic matching. International journal of computer applications 31(11) pages 29-34.

[13] Tang. L.R., (2008), Using a Machine Learning Approach for Building Natural Language Interfaces for Databases: Application of Advanced Techniques in Inductive Logic Programming. Journal of computer science, Informatics and Electronic Engineering, Volume 2 Issues 1.

[14] Brennan. S.E. (1988), The Multimedia Articulation of Answers in a Natural Language Database Query System. Stanford University and Hewlett Packard Labs.

[15] Minock. M., Olofsson. P., Naslund. A., (2008), Towards Building Robust Natural Language Interfaces to Databases. NLDB '08 Proceedings og 13th international conference on Natural Language and Information System: Application of Natural Language to Information System, Pages 187-198.

[16] Thompson. C., Moonay. R.J., Tang. L.R., (1997), Learning to Parse Natural Language Database Queries into Logical Form. ICML-97 Workshop on Automata Induction, grammatical Inference, and Language Aquicision, Nashville.

[17] Li. Y., Yang. H., Jagadish. H.V., (2006), Constructing a Generic Natural Language Interface for an XML Database. EDBT proceeding of 10th International Conference on Advances in Databases, Pages 737-754.

[18] Popescu. A.M., Etzoini. O., Kautz. H., (2003), Towards a Theory of Natural Language Interfaces to Databases. IUI proceedings of 8th International conference on Intelligent User Interfaces, Pages 140-157

[19] Selveter. S.C., (1984), Natural Language Database Updates. Proceedings of 3rd ACM SIGACT- SIGMOD symposium on principles of Database system pages: 239-247

[20] Ran. A., Lencevicius. R., (2007), Natural Language Query System for RDF Repositories. Proceedings of 7th International Symposium on Natural Language Processing.

[21] Meng. F., Chu. W.W, (1999), Database Query Formation from Natural Language Using Semantic Modeling and Statistical Keyword Meaning Disambiguation. Computer science department, University of California, Los Angles.

[22] Androutsopoulos. I., Ritchie. G.D., Thanisch. P., (1994), Natural Language Interfaces to Databases – An Introduction. Journal of Natural Language Engineering. Research Paper no. 709, Department of Artificial Intelligence, University of Edinburgh, 1994.

[23] Asif. N.M, Ullah. S., Bajwa. I. S., (2012), Interacting with Data Warehouse by Using a Natural Language Interface. Natural Language Processing and Information Systems, Lecture notes in Computer Science, volume 7337, pp 372-377

[24] Arora. N., Sahiwal. J. K., Sharma. S.K., (2012), Creating Data Warehouse For Natural Language Processing. An international Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2012, Vol. 6.

[25] Kuchmann-Beauger. N., Aufaure. M. A. (2011), A Natural Language Interface for Data Warehouse Question Answering. 16th International conference on Applications of Natural Languages to Information System, pages 201-208.

[26] Prasad. K.S.N., Ramakrishna. S., (2010), Text Analytics to Data Warehousing. International Journal of Computer Science and Engineering Vol. 02, Pages 2201-2207.

[27] Bajwa, I. S., Naeem, M. A., & Riaz-Ul-Amin, M. A. C. (2006, February). Speech Language Processing Interface for Object-Oriented Application Design using a Rule-based Framework. In 4th International Conference on Computer Applications.

[28] Shannon. K.J., (2011), Implementing Natural Language to Structured Query Language Translator.

[29] Kolodner. J.K., (1991), Improving Human Decision Making through Case-Based Decision Aiding. AI Magzine, Volume No. 2, Issue No. 2, pages 52-68

[30] Seco. N., Gomes. P., Pereira. F.C., (2004), Using CBR for Semantic Analysis of Software Specifications, Lecture Notes in Computer Science Volume. 3155, pages: 778-792

[31] Lopez. R.M., Mcsherry. D., et.al. (2005), Retrieval, reuse, revision, and retention in case based

[32] Reasoning. The Knowledge Engineering Review, Vol. 00:0, 1–2. 2005, Cambridge University Press

[33] Seyedhosseini. S.M., Ghadami. M.M., (2011), Applying Case-Based Reasoning in Supporting Strategy Decisions. World Academy of Science, Engineering and Technology 50.