Semantic Search and Query Over SBVR-based Business Rules using SMT based Approach and Information Retrieval Method

Kritika Anand, Sayandeep Mitra and Pavan Kumar Chittimalli
TCS Innovation Labs, Pune, India

Keywords: Business Rules, First Order Logic, SBVR, SMT Solvers, Information Retrieval.

Abstract: Presently, business organizations are regulating their activities with the aid of Business Rules (BR's). A single rule set of an organization contains large and diverse categories of BR's , thereby making it difficult for Business Analysts and end users to analyze and extract relevant BR's. Rule Search with natural language terms fail due to their inability to capture logical semantics present in BR's. In this paper, we present a novel approach to give correct and complete sets of `SBVR` (Semantics of Business Vocabulary and Business Rules) based BR's based on a specified query. We integrate conventional Information Retrieval Approach of text based searches over the rule base and corresponding meta-data with a `SMT` (Satisfiability Modulo Theory) based approach capturing the higher first order logic of the rules. The major applications of this approach are change impact analysis when rules are added, deleted or modified from a rule set, identifying the candidate set of rules affected due to change in the rule set and during match and gap analysis where we compare two sets of BR's identifying similarity and difference in business functionality between them. We show the implementation of our tool along with its performance on industry level datasets.

1 INTRODUCTION

There is a major paradigm shift in business-system design and development by the introduction of BR approach (Ross, 2003). BR's are the logical constraints imposed by enterprise business organizations to regulate their business activities. BR's are imperative for flexibility and efficacy of business systems. A BR's Approach is a methodology to mine, represent, automate, and change rules from a strategic business perspective (Von Halle, 2001).

BR's are subject to redesign depending upon external market conditions, government policies or with attempt to maximize revenue. Also, due to the IT transformation which include business practices like mergers and acquisitions, upgrades, incorporation of a new application, etc, the Business Analysts need to revisit their BR's. BR's and the ability to change them effectively are fundamental to improving business adaptability (Ross, 2003). BR's are rarely independent, the removal or changing of facts in BR repository may impact other rules in the system.

There is a need of system that can minimize the impact of IT transformation and the external factors on the BR. BR engine is required along with business modeling for planned agility of business systems (McCoy and Sinur, 2006). In order to maintain agility at a competitive level, there is a need to focus on 2 main problems:

i To study the change impact when a rule is added, modified or deleted from a rule repository.
ii Semantic search and query over the large repository of BR's.

Typically, a BR set consists of extensive and wide variety of rules. For example in a car rental company, rule knowledge base may consist of rules concerned with Rental Reservation Acceptance, Car Allocation for Advance Reservations, Walk-in Rentals, Handover, No Shows, Early Return, etc. As a result, Business Analysts and end users finds it difficult to analyze and extract relevant BR's. Therefore, we need a mechanism to query and retrieve relevant rules from rule repository or knowledge base keeping in mind the user's intent.

Most of the operational Information Retrieval (IR) systems in existence today use Boolean logic during search (Frants et al., 1999). In a search query, Boolean logic helps us to define the logical relationship between multiple search terms. The operators used to express the relationship are *AND*, *OR* and *NOT*. The IR search engines can be keyword based (*Google*, *Yahoo*), semantic based (*Hakia*) or hybrid of

Anand, K., Mitra, S. and Chittimalli, P.
Semantic Search and Query Over SBVR-based Business Rules using SMT based Approach and Information Retrieval Method.
DOI: 10.5220/0007710700470058
In *Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2019)*, pages 47-58
ISBN: 978-989-758-375-9

two approaches (Tümer et al., 2009). The keyword-based search is based on the occurrence of words in a document. Semantic search in information retrieval engines seek to improve search accuracy by understanding searcher's intent and the contextual meaning of terms in the query (Malve and Chawan, 2015). The logical nature of BR's makes the rules inter-connected and dependent on one another. Therefore, query and search on BR's must take into consideration the logical relationships among BR's.

Our current approach focuses on the rules expressed in SBVR (Team et al., 2006), an OMG standard that works as a bridge between business and IT. The main reason for choosing SBVR for the rule modeling is due to its declarative nature, natural language representation and support for the first order logic. The SBVR meta-model is used to represent business knowledge as (1) Specifying business vocabularies, (2) Specifying BR's. In our literature review, we could not identify much work done in the field of searching and querying over the BR's. The work (Sukys et al., 2012) describes the transformation framework from questions in SBVR into SPARQL (Harris et al., 2013) queries over the ontologies defined in Web Ontology Language (OWL2) (Hitzler et al., 2009) and SWRL (Karpovic and Nemuraite, 2011) analyzes the subset of Semantics of Business Vocabulary and BR's (SBVR) for a comprehensive representation of ontological knowledge defined using OWL2. The paper (Hassanpour et al., 2010) captures logical relationship between SWRL and the OWL ontology language to form a dependency graph. Based on the graph, (Hassanpour et al., 2010) cluster nodes within a layer if they have similar dependencies. The paper (Anand et al., 2018) detects inconsistencies in BR's based on model checking that exploits the First-order logic (FOL) (Smullyan, 2012) basis of SBVR specification. In this paper, we propose that to analyze the change impact on BR's, we need to query the business repository taking into account all the 3 aspects of BR's : *semantic*, *logical* and *keyword* based.

We also propound one of the other major application of search and query is the area of Match and Gap Analysis. The technique (Mitra et al., 2018) compares a set of BR's of a particular organization with the rules of a reference model, to get a measure of similarity among the business functionality of the two. The tool also measures the functionality gap that is present between embedded logic in the reference BR set and the other rule set.

The rest of the paper is organized as follows. Section 2 presents the list of challenges that motivated the need of the work. Section 3 provides a detailed classification of queries. Also, the section discusses in profundity how the approach of conventional IR method and SMT-Libv2 is used in Semantic search and query in SBVR based BR's. Section 4 and 5 discusses the application of our approach in *Change Impact Analysis* and *Match and Gap Analysis* respectively. Finally, experimental studies and discussions are provided in Section 6.

2 MOTIVATION

Enterprises remove, add and modify the rules without analyzing the impact on the existing rules. This may result in inconsistency and inclusion of unfirable rules. Consider the set of rules $\{R_1, R_2, R_3\}$:

R_1: It is necessary that if purchase price *of* customer *is greater than* 2500 and *less than* 5000 then customer *is* Bronze Customer.
R_2: It is necessary that if purchase price *of* customer *is greater than* 5000 and *less than* 10000 then customer *is* Silver Customer.
R_3: It is necessary that if purchase price *of* customer *is greater than* 10000 then customer *is* Gold Customer.

Now, a Business Analyst plans to introduce a new category i.e "*Platinum customer*" for the customers whose purchase price is greater than 15000 to already existing categories (*Bronze*, *Silver* and *Gold*). The addition of rule R_4 causes rule R_3 to get changed to R_5.

R_4: It is necessary that if purchase price *of* customer *is greater than* 15000 then customer *is* Platinum Customer.
R_5: It is necessary that if purchase price *of* customer *is greater than* 10000 and *less than* 15000 then customer *is* Gold Customer.

Clearly, there is a need of BR engine that can automate the change impact for planned agility of business system.

Let us take queries from *Insurance* domain:

Q_1: *'Retrieve me all the rules related to vehicle purchase price is* 40000.*'*

Q_2: *'Retrieve me all the rules for which the product Group code is Motor Home Plus.'*

The following rule is relevant and should be retrieved from the given *Insurance* rule base:

```
“‘For other than EAE states during
a ‘New Business’, ‘NLOB’ and ‘First
Auto In Umbrella transaction’,
‘Cancel’ the coverage for the vehicle
with reason code ‘Decline based on
Application information without further
investigation’ when one of the below
condition is met:
 • Vehicle type is ‘Motor Home’
 • Product group code is other than
   ‘Facility Tier’ and ‘Bureau Rating’
 • vehicle purchase price is greater
   than 35000 and less than 60000.
```

As we can see query Q_1 maps to rule's third condition i.e *'vehicle purchase price is greater than* 35000 *and less than* 60000*'* and Q_2 falls into the category of the second condition *'Product group code is other than 'Facility Tier' and 'Bureau Rating'*. The rule set retrieved for a query include the computational analysis and the domain-based semantics. Therefore, the keyword-based IR techniques will not present the correct result.

The General Financial Rules (GFRs) are a compilation of rules and orders of Government of India to be followed by all while dealing with matters involving public finances. It consists of 324 rules from 'Defalcation and Losses', 'Submission of Records and Information', 'Control of expenditure against budget', 'Classification of transaction in Government accounts'. If an end user wants to retrieve the rules related to 'Local Fund' or 'service is agent of private body' or 'cost of project is greater than Rs 100 crore or above', the conventional information retrieval techniques will not be able to reclusively retrieve the results.

To tackle this limitation, we propose that to analyze the change impact on BR's, we need to query the business repository taking into account all the 3 aspects of BR's : *semantic*, *logical* and *keyword* based. Our paper aims to retrieve relevant rules from a set of SBVR based BR's when certain rules belonging that particular set is targeted for a change.

3 CLASSIFICATION OF QUERIES

We identify 6 classes of queries: taxonomic, metadata-based, logical, complex query involving quantitative analysis and domain based semantics, data-rule compliant and question-answering based query. A query can fall into any one of the category or the combination of two or more categories. We discuss each type of query in detail in this section.

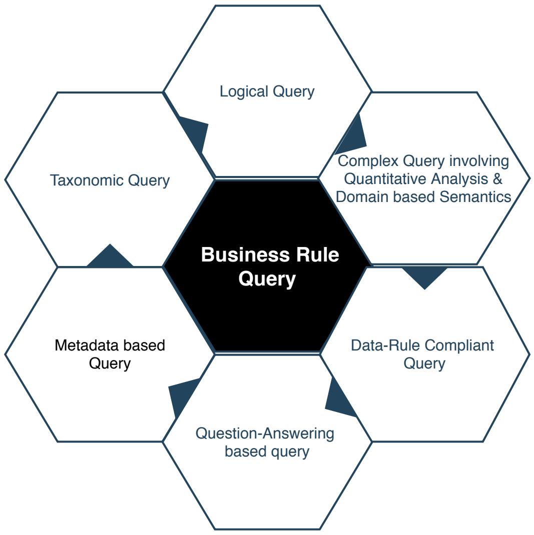

Figure 1: Classification of Queries on BR's.

Examples accompanying most of these queries for illustration and better understanding have been taken from real expert system applications.

1. *Taxonomic Query :*
   Such type of queries take into consideration the hierarchical aspects of concept types involved in the SBVR vocabulary. The concepts can be individual noun concepts (e.g., *Germany to Poland car movement*) or general noun concept (e.g., *international car movement*) as seen from the block diagram in Figure 2. The Figure 2 is a sub-set of the block diagram used in EURent (KDM Analytics, 2016), a car rental company data. Figure 3 consists a small sub-set of rules taken from the EURent dataset.

   We keep a dictionary of SBVR terms used in the SBVR vocabulary. Then for each term, we have a list that records the rules in which the term occurs in. Such a list is conventionally known as a *posting list* and all the *postings lists* taken together are referred to as the postings (Christopher et al., 2008). The dictionary with pointer to *posting list* is kept in memory and the *posting list* is stored in disk. The idea is very much similar to the indexing the vocabulary used in information retieval for accessing the relevant documents.

   The lexicons obtained after tokenization and stemming have been replaced by SBVR vocabulary and the SBVR rules are used instead of documents. Figure 4 represents the *posting list* for sub-set of terms used in rules in Figure 3.

   First, let us consider the query term only consists of single SBVR term. The retrieved set of rules *relevant_rule*, as an output to the query consists of i Rules present in the *posting list* of query term q

ii Rules present in the *posting list* of terms present in upward hierarchy of q (q_u) = $\{q_{u1}, q_{u2},...., q_{um}\}$.

iii Rules present in the *posting list* of terms present in downward hierarchy of q (q_d) = $\{q_{d1}, q_{d2},...., q_{dn}\}$.

iv Rules present in the *posting list* of synonym of q, q_u *and* q_d

$$\begin{aligned} relevant\_rule\ (Q) = posting\_list(q) \cup posting\_list(q_u) \\ \cup posting\_list(q_d) \cup posting\_list(synonym(q)) \\ \cup posting\_list(synonym(q_u)) \\ \cup posting\_list(synonym(q_d)) \end{aligned}$$

Consider the query when the user is interested in retrieving the rules associated with **'one way car movement'** from set of rules shown in Figure 3. The users should be presented with all the rules relating to *'one way car movement'*, the upward hierarchy i.e *car movement*, the downward hierarchy i.e *local car movement*, *in-country car movement* and *international car-movement* and its specialization *Germany to Poland car movement*. The conventional keyword based or entity based search will not give the desired rules as it does not consider the hierarchy important to the rule set semantics.

In General, Q can consist of the query terms $\{q_1, q_2, q_3,....q_n\}$, Q can be the conjunction or disjunction of q_i or its negated form $\neg q_i$.

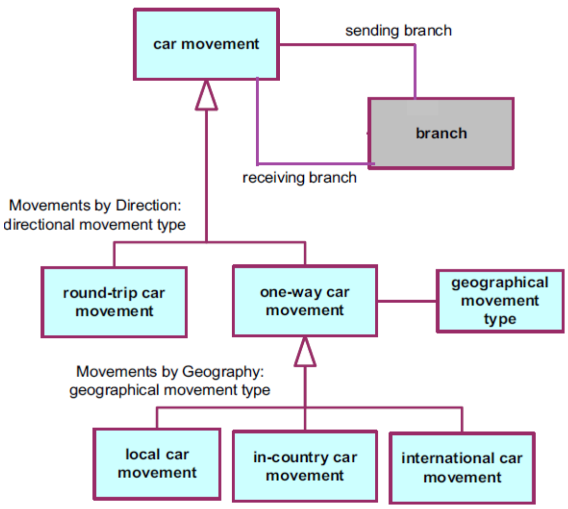

Figure 2: Building block showing taxonomic relation of SBVR concept 'car movement' (planned movement of a rental car of a specified car group from a sending branch to a receiving branch).

For Conjunctive Query $Q = \{q_1\ and\ q_2\ and\ ...\ q_n\}$,

$$\begin{aligned} relevant\_rule\ (Q) = relevant\_rule\ (q_1) \wedge \\ relevant\_rule(q_2) \wedge ...... \\ ..... \wedge relevant\_rule\ (q_n) \end{aligned}$$

For Disjunctive Query $Q = \{q_1\ or\ q_2\ or\ ...\ q_n\}$,

$$\begin{aligned} relevant\_rule\ (Q) = relevant\_rule\ (q_1) \vee \\ relevant\_rule(q_2) \vee ...... \\ ..... \vee relevant\_rule\ (q_n) \end{aligned}$$

For Query $Q = \{q_1\ and\ not\ q_2\}$, the relevant rule set will consists of rules which are present in *posting list* of q_1 or present in *posting list* of synonym of q_1 but are not present in the downward hierarchy of *posting list* of q_2 (q_{2d}). For Q = 'round trip car movement and not one way car movement', rule R_2, R_3, R_7 and R_8 will be retrieved from a set of rules in Figure 3. Rule R_3 is retrieved in the result set as the *Noun Concept rental movement* is synonym of the *Noun Concept car movement*.

2. *Logical Query :* Sometimes, the execution of one or more rules in business system can trigger other rules. This logical nature makes the rules interconnected and dependent on one another. It is very important for business analysts to analyze and comprehend such interconnected rules. So we propose the idea of construction of *Rule Dependency Graph* to map the logical relations taking into consideration the semantic and syntactic properties of rules. The approach will explore the interactions between SBVR rules and check how a rule is affected by execution of other rules.

SBVR is based on first-order logic and can also adopt higher-order logic restricted to Henkin semantics (e.g., for dealing with categorization types). In general, standard higher-order logic allows quantification over uncountably many possible predicates (or functions)(SBVR, 2008). The rule R_4 in Figure 5 encompasses an obligation formulation which ranges over an implication logical formulation . The implication logical formulation scopes over 2 logical operands. The first logical operand is conjunction and the second logical operand ranges over an universal quantification. This universal quantification introduces a variable 'rental' and ranges over another universal quantification. The latter universal quantification introduces a second variable that ranges over concept *'car exchange'* and scopes over an atomic formulation *'rental incurs car exchange'*.

Let's take a SBVR rule of the form : *'if* fa_1 *and* fa_2 *and.....*fa_n *then* fc_1 *and* fc_2 *and....*fc_m*'*

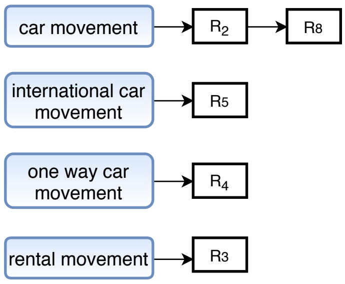

Figure 4: *Posting List* for sub-set of terms used in rules in Figure 3.

where fa_1 ,fa_2 ,.... fa_n are the atomic facts associated with antecedent and fc_1, fc_2, ... fc_m are atomic facts associated with consequent. We construct the the *fact tree* of atomic facts involved in the rule. For example, the forest of *Fact tree* associated with rule R_2 (of Figure 5) corresponding to atomic facts : fa_1 : *'driver has driving license'* and fc_1 : *'age of driver is greater than 21'* is shown in Figure 6.

As we can see in Figure 5, rule R_4 is clearly dependent on Rule R_6 as the atomic fact of consequent of R_6 ($R_6 fc_1$) i.e *'car is in need of service'* matches with the atomic fact of antecedent of R_4. Therefore, there is an edge from R_6 to R_4 in Figure 7. Rule R_4 does not initially seem to be dependent on either rule but if the underlying SBVR vocabulary declares that *rental* has specialization as *luxury rental*, then a dependency can be inferred. As the atomic fact of R_2 i.e $R_2 fc_1$ (age of driver is greater than 18) subsumes atomic fact of R_3 i.e $R_3 fa_1$ (age of driver is greater than 21), there is a dependency from R_2 to R_3. The complete *Rule Dependency Graph* for set of SBVR rules(Figure 5) is represented in Figure 7. We detect the logical dependencies in SBVR rules based on model checking that exploits the *FOL* basis of SBVR specification.

Test for checking dependency between rules:

We use the work described in (Anand et al., 2018) to convert the SBVR rule to SMTLibv2 (Barrett et al., 2010) mappings. The mappings are constructed based on *Many-sorted logic* and *graph reachability algorithm*. *Many-sorted logic* is the generalization of *FOL* in which the domain of universe is classified into disjunct sorts (or types). We present a SMT based method to detect the the presence of edge between two SBVR rules. Let's first simplify the notation of the negation of the formula obtained by smt($atomFact_1$) →smt($atomFact_2$).

$$\neg(smt(atomFact_1) \rightarrow smt(atomFact_2))$$
$$\equiv \neg(\neg smt(atomFact_1) \vee smt(atomFact_2))$$
$$.........[MaterialImplication]$$
$$\equiv (smt(atomFact_1) \wedge \neg smt(atomFact_2))$$
$$.........[DeMorgan'sLaw]$$

A formula F subsumes another formula F' ($F \succ F'$) if for each interpretation I, $I \models F'$ *implies* $I \models F$ (Lukichev, 2010). We use this definition to detect logical dependency present in SBVR rules. To detect an edge between two rules, let us consider 2 cases.

- Case I: *Rules involving Numeric Range Quantification*:
  SBVR rules and definitions can consist of aspects in which a thing or property associated with a Noun Concept is measurable in terms of *'greater than'* or *' less than'* or *'equal to'* (as shown in Rule R_2 in Figure 5). Also, sometimes the Noun Concepts are scoped by numeric range quantification like *exactly-n-quantification*, *atmost-n-quantification* having minimum or maximum cardinality as shown in Rule R_2 in Figure 3. Figure 5 depicts rules relating to car rental company. The SBVR vocabulary and rules used in Figure 5 is an extension of *EURent* data, to provide clear insights of our

R_1: It is necessary that rental *is open* if estimated rental charge *is provisionally charged to* credit card *of* renter.
R_2: It is necessary that each car movement *has* exactly 1 receiving branch and sending branch.
R_3: It is necessary that each rental movement *specifies* exactly 1 car group.
R_4: It is necessary that car transfer *has* a transfer drop off branch if transfer drop off branch *is* receiving branch *of* one way car movement that *is included in* car transfer.
R_5: It is necessary that rental *is open* if rental *has* international car movement then rented car *satisfies* legal requirements and emission requirements *of* visiting country.
R_6: It is permitted that rental *is open* if each driver *of* rental *is* not barred driver.
R_7: Round trip car movement *has* pick up branch that *is same as* return branch *of* car rental.
R_8: It is necessary that each car movement *has* exactly 1 movement-id.

Figure 3: SBVR rules related to EURent car rental company.

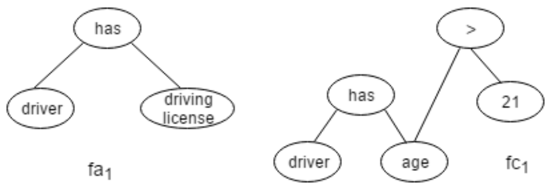

Figure 6: Forest of fact trees $\{f_{a1}, f_{c1}\}$ corresponding to the Rule R_2

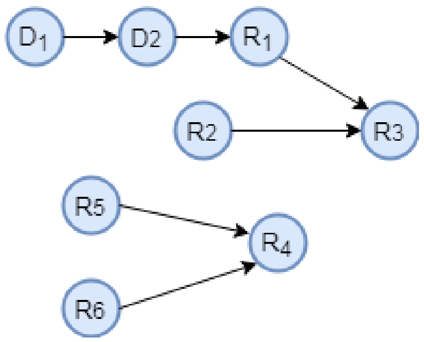

Figure 7: Rule Dependency Graph for the set of rules in Figure 5.

approach.

We construct the *FOL* and `SMTLibv2` for each atomic fact involved in a rule. The *FOL* corresponding to atomic fact $R_2 f_{c1}$ i.e first consequent of R_2 is

> $R_2 f_{c1}$: age of driver is greater than 21.
> ***fol***($R_2 f_{c1}$): $\forall$x driver(?x) $\wedge$ ageInt(?x) > 21
> ***smtlib*** :
> (assert (forall ((x Cluster_Driver) (y Int))
> ($\Longrightarrow$ (driver_domain x)(> (ageInt x) 21))))

The *FOL* corresponding to atomic fact $R_3 f_{c1}$ i.e first antecedent of R_3 is

> $R_3 f_{a1}$: age of driver is greater than 18.
> ***fol***($R_3 f_{11}$) : $\forall$x driver(?x) $\wedge$ ageInt(?x) > 18
> ***smtlib***:
> (assert (forall ((x Cluster_Driver) (y Int))
> ($\Longrightarrow$ (driver_domain x) (> (ageInt x) 18))))

We then check the satisfiability of negation of ($R_2 f_{c1} \Longrightarrow R_3 f_{a1}$) i.e., *forml* = (smt($R_2 f_{c1}$)) $\wedge$ ¬(smt($R_3 f_{a1}$)). An instance of `driver that has age greater than 21` is bound to have an `age greater than 18`. `SMT` solver will fail to find a solution that satisfies $R_2 f_{c1}$ but not satisfy $R_3 f_{a1}$. When we assert the *forml* to be true and run the `SMT` solver, solution will be *UNSAT*. The presence of unsatisfiable core (*UNSAT*) will tell there is an edge from R_2 to R_3.

- Case II: *Rules not involving Numeric Range Quantification*:
  Rules R_4 and R_5 in Figure 5 falls under this category. We construct the `SMTLibv2` coressponding to atomic fact $R_4 f_{a1}$ as shown below:

> $R_4 f_{a1}$: rental is open.
> ***fol***($R_4 f_{a1}$) : $\forall$x rental(?x) $\wedge$ isOpen(?x)
> ***smtlib*** :
> (assert ($\forall$ ((x Cluster_Rental))
> ($\Longrightarrow$ (rental_domain x)(isOpen x)))).

The *FOL* corresponding to atomic fact $R_5 f_{c1}$ i.e first consequent of R_5 is

> $R_5 f_{c1}$: luxury_rental is open.
> ***fol***($R_5 f_{a1}$) :
> $\forall$x luxury_rental(?x) $\wedge$ isOpen(?x)
> ***smtlib*** :
> (assert ($\forall$ ((x Cluster_Rental))
> ($\Longrightarrow$ (luxury_rental_domain x)(isOpen x))).

> Germany to Poland car movement
> D_1 Definition: international car movement whose source country *is* Germany and destination country *is* Poland
> Germany to Poland car rental
> D_2 Definition: car rental that *includes* Germany to Poland car movement
> R_1: It is necessary that if rental *includes* international car movement then rented car *satisfies* legal requirements *and* emission requirements visting country.
> R_2: It is necessary that if driver *has* driving license then age *of* driver *is greater than* 21 years.
> R_3: It is necessary that driver *is authorized in* visting country if age *of* driver *is greater than* 18 years and issuing date *of* driver license *is before* scheduled pick up date *of* rental *by atleast* 1 year *and* rented car *satisfies* legal requirements *and* emission requirements visting country.
> R_4: It is necessary that if rental *is open* and car *is in need of* service then rental *incurs* car exchange.
> R_5: It is necessary that if driver *of* rental *is* not barred driver then luxury_rental *is open*.
> R_6: It is necessary that if service reading *of* car *is greater than* 5500 miles then car *is in need of* service.

Figure 5: `SBVR` rules related to car rental company.

We then check the satisfiability of negation of ($R_4 f_{a1} \implies R_5 f_{c1}$) i.e. *forml* = (smt($R_4 f_{a1}$)) $\wedge$ $\neg$(smt($R_5 f_{c1}$)). As the *luxury car rental* is a specialization of *rental*, it is not possible that all *rentals* are open but the *luxury rental* is not open. SMT solver will fail to find a solution that satisfies $R_4 f_{a1}$ but not satisfy $R_5 f_{c1}$. The presence of unsatisfiable core (*UNSAT*) will tell there is an edge from R_5 to R_4.

We will check the dependency between the rules by performing the test between with SMTLibv2 corresponding to each *fact tree* ($R_j f_{ak}$) of antecedent of rule R_j with SMTLibv2 associated with each *fact tree* ($R_i f_{cl}$) of consequent of rule R_i. If the solver returns *UNSAT*, then there is an edge present between rules R_i and R_j.

Figure 7 depicts the complete *Rule Dependency graph* for the set of rules shown in Figure 5. As seen from the graph, definition 'D_2' is triggering the rule R_1 as *'Germany to Poland rental'* is specialization of rental as inferred from D_2 and *'Germany to Poland car movement'* is an instance of *'international car movement'* as deduced from D_1. As a result, there is an edge present from D_1 to D_2 and in turn from D_2 t o R_1.

3. *Metadata based Query*
BR's are often associated with the contextual information known as meta-data. The information is usually structured that describes the data like date-timestamp when the rule was created or modified, the expiry date of rules, states or products on which the rule is applicable or the user ID of the person who created the rules. Queries on meta-data are normally database or *SQL* (Date and Darwen, 1989) queries like *'List the rules that are applicable for state 'AX'* or *'List the rules formulated in last 1 year'*.

4. *Complex Query involving Quantitative Analysis and Domain based Semantics*
The rule set retrieved for a query may include the computational analysis and the domain-based semantics. Let us take query from *Insurance* domain which we consider in Section 2.

Q_1 : *'Retrieve me all the rules related to vehicle purchase price is* 40000.*'*

Q_2 : *'Retrieve me all the rules for which the product Group code is Motor Home Plus.'*

We devise SMT based approach to retrieve the rules for such complex queries. We formulate SMT formula for the the query q i.e *smt(q)* and for the atomic facts associated with antecedents and consequents : *smt(R) = smt(fa_1)*, smt(fa_2),......smt(fc$_1$), smt(fc_2).... Then,

$\forall$ *forml* $\in$ *smt(R)*
assert *conjunction* of *smt(q)* and negation of *smt(forml)*
i.e (assert (*smt(q)* $\wedge$ $\neg$ *smt(forml)*))
If the SMT solver returns *UNSAT* then *R* is a part of result set.

5. *Data-Rule Compliant Query*
It is very important for data to abide by both industry regulations and government legislations. Legislation laws and business policies are changed regularly by the government, as a result of which businesses continually have to respond to changes in the legal framework e.g. the "Sarbanes-Oxley Act of 2002" is a United States federal law that was passed in response to a number of corporate accounting scandals that occurred between 2000-02 (Sarbanes Oxley Act, 2002).

Business Analysts are also interested to check the validity of the records or data present in database with business rules. The data compliance ensures that the data is in accord with established guidelines or specifications. Such type of queries are useful to ensure the data is consistent with demand of dynamically changing regulatory environment and the operational policies. These queries checks the user's given data with the rules and gives the answer in Boolean format. For instance, for a query q : *"Check the validity of 'customer has age that is equal to* 65 *years and that customer holds a SBI credit card"'*. The query will return false as it contradicts with the rule r: *'To avail SBI credit card, customer must lie within age bracket of* 21 *to* 60 *years'*. We convert the query q to following SMT formula :
fol : $\exists x$, customer(?x) $\wedge$
holds(?x, SBI_Credit_Card) $\implies$ ageInt(?x) = 65
smt(q) :

```
(assert(exists(
(x Cluster_Customer)(y Cluster_Card))
(and (customer_domain x)
(implies
  (and (customer_domain x) (holds x y)
       (= y SBI_Credit_Card) )
   (= (ageInt x) 65))))
```

The SMTLibv2 formula has been formulated based on *Many Sorted Logic* and *Graph Reachability* approach as described in (Anand et al., 2018).
smt(r) :

```
(assert(forall(
(x Cluster_Customer)(y Cluster_Card))
(and (customer_domain x)
(implies
  (and (customer_domain x) (holds x y))
       (= y SBI_Credit_Card) )
```

```
(or (> (ageInt x) 21)
    (< (ageInt x) 60))))))
```

The SMT solver will produce an unsatisfiable core as the SMT-Libv2 formula corresponding to the query i.e. *smt(q)* contradicts with SMT-Libv2 formula corresponding to the rule i.e. *smtlib(r)*.

6. *Question-Answering based Query*
The Question-Answering queries demands an answer to be given with a short string snippets expressing named entities, temporal expression, location or a numerical value (Aunimo et al., 2007). The queries can be
q1 : 'What is the rental duration of rental ?'
q2 : "What is the maximum limit of service reading of a rental car ?'
The result of the query can retrieved from a single rule or have to be extracted from the combination of two or more rules.
(Šukys et al., 2017) defined 9 model transformation rules to transform SBVR questions to *SPARQL* queries. The work (Šukys et al., 2017) handles different SBVR questions that are transformed in the solution: i. questions to find individuals of certain type (e.g., Find persons) ii. simple questions with roles bound to variables or individuals (e.g., What states that border Illinois?) iii. counting questions (e.g.,How many states border Illinois?) iv. questions with cardinality restriction (e.g., Find states that border at least 3 states.) v. questions with numerical comparison (e.g., Find cities that have population greater than 100000.) vi. questions to find minimum or maximum values (e.g., Find state that has largest population.).

4 APPLICATION OF SEARCH AND QUERY OVER SBVR-BASED RULES

4.1 Change Impact Analysis

Change Impact Analysis provides accurate understanding of the implications of a proposed change on a rule set. Sometimes, the rules are inter-dependent or interleaved with each other. Modification in one rule can impact other rules in business system. Therefore, it is necessary to identify the rules and vocabulary that might have impacted if business analysts performs a desired change. The technique of semantic and logical search and query can be adapted to use in *Change Impact Analysis*. It is imperative to detect candidate set of rules which can be triggered or affected when a business rule is targeted for a change.

The following instances highlight the importance of querying in *Change Impact Analysis*:

1. **Instance I:** Let's suppose business owner of car rental company wants to remove the term *'international car movement'* from vocabulary as shown in Figure 2. From Figure 2, it can be depicted all the rules related to *'international car movement'* and its specialization (like *'Germany to Poland International car movement'*) should be removed from the system. To achieve this, we fire a taxonomic query i.e. *Q* = *'international car movement and not one way car movement'*. The query *Q* will retrieve all the rules (*relevant_rule (Q)*) related to international car movement and its specialization. The *relevant_rule (Q)* will not cover the rules related to one way car movement, local car movement, in-country car movement, round-trip car movement or car movement. Therefore, to remove the term 'international-car movement', all the rules present in (*relevant_rule (Q)* should be removed from the system.
2. **Instance II:** Consider set of rules R_1, R_2 and R_3 introduced in section 2, an introduction of rule R_4 causes the rule R_3 to changed to R_5. To find set of impacted rules by the addition of rule R_4, let' s first find the atomic facts involved in the rules. By parsing the SBVR XMI, we obtain atomic fact $R_2 f_{a1}$: 'purchase price of customer is greater than 15000' and $R_4 f_{c1}$: 'customer is Platinum customer'. The rule R_3 to be changed can be retrieved if we fire a fact based query on atomic fact $R_4 f_{a1}$. The atomic fact of R_3 i.e $R_3 f_{a1}$ 'purchase price of customer is greater than 10000' subsumes atomic fact $R_4 f_{a1}$.
3. **Instance III:** Consider a scenario when we want to add a rule R_q to a set of rules in Figure 5. Suppose R_q :'if *age* of *driver* is greater than 16 then *driver* has *driving license*'. As we can see from the set of SBVR rules, the newly added rule contradicts with the rule R_2. This type of anomaly can be achieved with the help of logical query. We assert and append the rule R_q to the existing SBVR vocabulary and rules involved in Figure 5. Then, we construct a the *Rule Dependency Graph*. The graph will help to analyze the set of dependencies present between the rules. There will be an edge present from R_2 to R_q as the atomic fact *'age of driver is greater than 21'* is subsumed by the atomic fact *'age of driver is greater than 16'*. By analyzing and comprehending the incoming and outgoing edges of R_q, we can know the candidate set of rules that can be affected by the addition of R_q.

4.2 Match & Gap Analysis

In modern business model, which undergoes continuous change comparing two sets of rules to find matches and gaps among them is a very important operation. The work done by (Mitra et al., 2018) presents an intelligent and automated way to perform Match & Gap Analysis on BR's mentioned in `SBVR`. We believe that application of Search and Query techniques will aid their approach to perform more efficiently.

The work done in (Mitra et al., 2018) initially uses NLP techniques along with domain knowledge to initially match the `SBVR` vocabulary of both rule sets, followed by selecting a set of possible matches between the rule sets. The set of possible matches are then matched using logical equivalence by converting the rules to `SMT-LIBv2` which gives us the matches and gaps (if present). Presently the approach tries to generate the set of possible matches by considering linguistic similarity of facts only, which causes some possible matches involving quantification range and stricter conditions to miss out. To ensure better precision, we propose the use of search and query to enhance the candidate set for logical equivalence.

For every rule r consisting of facts $\mathcal{F} = \{f_1, f_2, \cdots f_n\}$ in rule set $\mathcal{R}_1$, we run a *Complex Query* involving facts $\mathcal{F}$ on the other rule set $\mathcal{R}_2$. The query generates the set of rules $\mathcal{R}_2(r) \subseteq \mathcal{R}_2$, which contain rules having similar facts to $\mathcal{F}$. We find that replacing the existing method of using fact similarity with this approach generates richer candidate sets.

R_1 (1): It is necessary that if purchase price *of* customer *is greater than* 2500$ then customer *is* Bronze Customer and customer *is eligible for* credit_card.
R_1 (2): It is necessary that if purchase price *of* customer *is greater than* 3500$ then customer *is* Silver Customer and customer *is eligible for* credit_card.

R_2 (1): It is necessary that if purchase price *of* customer *is equal to* 4000$ then customer *is eligible for* credit_card.

Figure 8: Example Rule Sets R_1 and R_2.

Figure 8 shows two rule sets R_1 & R_2. As per the present approach mentioned in (Mitra et al., 2018), neither of $R_1(1)$ or $R_1(2)$ will be shown as possible matches to $R_2(1)$, even though they being possible matches. On the other hand, running a complex based query which is a disjunction of two facts involved in rule $R_2(1)$, i.e., f_1 : purchase price *of* customer *is equal to* 4000$ or f_2 : customer *is eligible for* credit_card .

Another application of search and query on Match and Gap Analysis is to identify the extent of gaps present. Most of the times, two rules from different rule sets will not be a perfect match. At present, the approach described in (Mitra et al., 2018) does not highlight the exact nature of gap present in approximate matches. Figure 9 shows two rules from two different rule sets which will show as a match as per (Mitra et al., 2018), but the match is partial with Mercedes and car are similar conceptually but not linguistically.

Vocab$_1$: car
driver
rented_car
RS_1 (1): It is necessary that if car *has* driver then car *is* rented_car

Vocab$_2$: four_wheeler
Mercedes
General Concept: four_wheeler
driver
rented_car
RS_2 (1): It is necessary that if Mercedes *has* driver then Mercedes *is* rented_car.

Figure 9: Example Rule Sets RS_1 and RS_2.

Using search and query we can find out the term hierarchy of *Mercedes*, and identify the term which has the closest linguistic similarity to *car*, which in this case is *four_wheeler*. This allows us to propose to the business analyst that a rule containing *four_wheeler* instead of *Mercedes* will be a better match.

5 EXPERIMENTAL STUDIES AND DISCUSSIONS

To evaluate our approach, we built '***Logic and Semantic Rule Searcher***' on top of `SBVR` rule editor in our *BURRITO* tool. The `SBVR` rule editor facilitates an easy way to business analysts for specifying the `SBVR` vocabularies and rules. The tool is allowed to generate the `SBVR XMI` based on the `SBVR` 1.2 meta-model, that can be provided as an input to the *Rule Searcher*. The *Rule Searcher* presents the user with the following options :

- Term based Search: This option clusters the rules on the basis of term specified by user. The input to term based search can be the conjunction/disjunction of terms or their negation. For

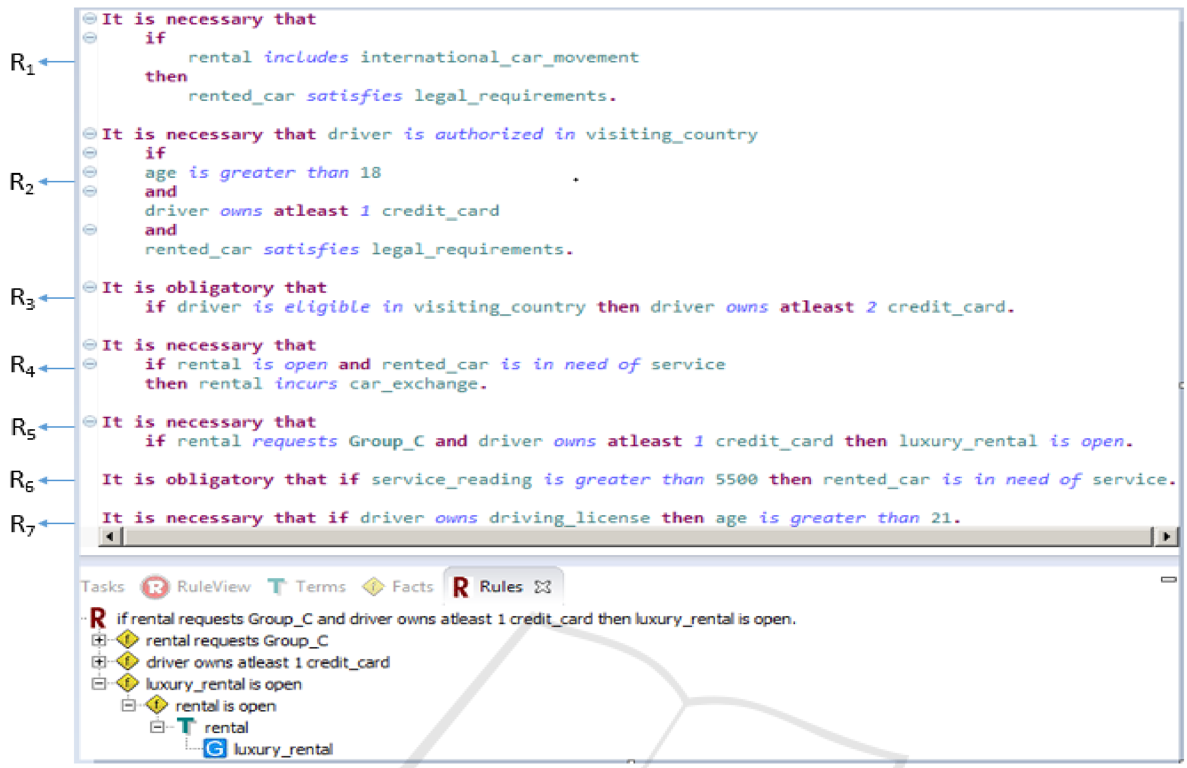

Figure 10: Snapshot of business rule editor of BuRRiTo tool.

instance, in a car rental company the user can be interested the rules with '*international car movement* and not *one way car movement*'

- Fact based Search: In this option, we retrieve the rules based on the fact that the user has given as input. The results are retrieved taking into account the *keyword*, *semantic* and *logical* aspects of rules and query.

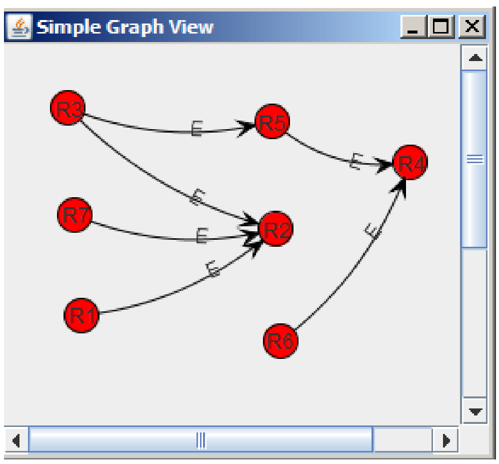

Figure 11: Snapshot of Rule Dependency Graph created from the BR's depicted in Figure 10.

- Analyze Rule Knowledge Base: It constructs a rule dependency graph that assists the user to explore the interactions between SBVR rules and check how a rule is affected by or affects the execution of other rules.
- Analyze the Change Impact: It detects the *Candidate Rules* impacted by *addition/ deletion/ modification* of rule/term.

Figure 10 depicts the screenshot of SBVR editor in our BuRRiTo tool. The editor checks for the duplicate facts and terms and also assists user to write the rules correctly based on the defined SBVR vocabulary. The *Rule Dependency Graph* constructed from the rules is shown in Figure 11.

To evaluate the usefulness and efficacy of our techniques, we used the tool to perform a case-study using two real-world applications

1. **EU Rent Car Rental (KDM Analytics, 2016):** It consists of 64 rules from a car rental company concerned with Rental Reservation Acceptance, Car Allocation for Advance Reservations, Walk-in Rentals, Handover, No Shows, Early Return, etc.
2. **Rules from Industrial Insurance Application:** We obtained a set of 110 rules from the Industrial case-study belonging to insurance domain. The

Table 1: Comparison of work on **EURent** and **Insurance Data**.

	EURent		Insurance	
	Tool	Manual	Tool	Manual
Term-based Search	10	8	5	4
Fact-based Search	4	2	3	2
Addition of rule/ term	1	0	2	1
Deletion of rule/ term	2	1	4	2
Modification of rule	2	2	2	1

rules were complex containing the data related to liability and package policy.

As per our knowledge, there is no study that has been conducted on searching in SBVR based rules. We provided a researcher working in the field of Natural Language Processing with a query belonging to the EU-Rental domain and another query to an experienced business analyst from the Insurance Domain. In both the cases we asked them to manually retrieve the results from the rule set based on the queries. When compared with the results from our tool, our tool retrieved a richer set for both the queries than manual searches in considerable lesser time. The most promising observation was that the tool gave no false positives in the results. We represent the results in Table 1 for reference.

6 CONCLUSION AND FUTURE WORK

We have presented a novel approach to give correct sets of *SBVR* business rules for a user's specified query. We have integrated the conventional informational retrieval approach to perform text-based query over knowledge base and meta-data and *SMT*-based-approach to capture the higher first order logic of *SBVR*. The paper aims to retrieve the set of SBVR rules for a user's specified query taking into consideration the *logical*, *keyword* and *semantic*. We build a rule graph to analyze and visualize logical dependencies present in SBVR rules. The method leverages the transformation frameworks from *SBVR* to *SMTLibv2* to incorporate the

- quantifications (*universal*, *existential*, *at-most-n*, *at-least-n* and *exactly-n*),
- logical operators (*logical negation*, *conjunction*, *disjunction* and *implication*)
- synonyms, synonymous forms and specialization or taxonomic relations relations in SBVR.

The paper also discusses how the interaction of different types of query can be adapted to be used to give the potential candidate rules rules when a business rule is targeted for a change (*addition*, *deletion* or *modification*). An intuitively appealing approach therefore seems to be to enhance the flexibility and resilience of systems to cope with impact of changes in the business rules. We also depict the application of searching and querying in *Match and Gap Analysis* to compare a set of Business Rules of a particular organization with the rules of a reference model.

The generic framework sketched in the above approach supports decision making in the organization. In terms of analysis, an important task is to better detect subsuming rules, rules involving circularity, unfirable rules and the duplicate or redundant rules. The information can then help in designing the anticipatory strategies that will soon be needed to detect such kinds of rules. The future research in the field of searching and querying should also take into consideration the approach for *Question-Answering* queries, which have not been considered in the current paper. We want to provide stronger experimental results to showcase the efficiency of our tool. As mentioned in (Mitra and Chittimalli, 2017), there is a lack of strong datasets of SBVR Vocabulary and Rules. We are presently working on generating an empirical survey covering different spectrum of SBVR analysis which shall also contain precision and recall for this tool. Therefore, there is a need of a standardized and universally accepted case study which captures all the complexities in business rules and can serve as benchmark data for all the future works.

REFERENCES

Anand, K., Chittimalli, P. K., and Naik, R. (2018). An automated detection of inconsistencies in sbvr-based business rules using many-sorted logic. In *International Symposium on Practical Aspects of Declarative Languages*, pages 80–96. Springer.

Aunimo, L. et al. (2007). Methods for answer extraction in textual question answering.

Barrett, C., Stump, A., Tinelli, C., et al. (2010). The smt-lib standard: Version 2.0. In *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories (Edinburgh, England)*, volume 13, page 14.

Christopher, D. M., Prabhakar, R., and Hinrich, S. (2008). Introduction to information retrieval. *An Introduction To Information Retrieval*, 151(177):5.

Date, C. J. and Darwen, H. (1989). *A guide to the SQL Standard: a user's guide to the standard relational language SQL*. Addison-Wesley.

Frants, V. I., Shapiro, J., Taksa, I., and Voiskunskii, V. G. (1999). Boolean search: Current state and perspectives. *Journal of the American Society for Information Science*, 50(1):86–95.

Harris, S., Seaborne, A., and Prud'hommeaux, E. (2013). Sparql 1.1 query language. *W3C recommendation*, 21(10).

Hassanpour, S., O'Connor, M. J., and Das, A. K. (2010). Visualizing logical dependencies in swrl rule bases. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 259–272. Springer.

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S. (2009). Owl 2 web ontology language primer. *W3C recommendation*, 27(1):123.

Karpovic, J. and Nemuraite, L. (2011). Transforming sbvr business semantics into web ontology language owl2: main concepts. *Information Technologies*, pages 27–29.

KDM Analytics (2016). EU-Rent Car Rental case study. http://www.kdmanalytics.com/sbvr/EU-Rent.html.

Lukichev, S. (2010). Improving the quality of rule-based applications using the declarative verification approach. *International Journal of Knowledge Engineering and Data Mining*, 1(3):254–272.

Malve, A. and Chawan, P. (2015). A comparative study of keyword and semantic based search engine.

McCoy, D. W. and Sinur, J. (2006). Achieving agility: The agile power of business rules. *Gartner, Special report on Driving Enterprise Agility*, 20.

Mitra, S. and Chittimalli, P. K. (2017). A systematic review of methods for consistency checking in sbvr-based business rules. In *Joint Proceedings of the 3rd Modelling Symposium (ModSym), Developmental Aspects of Intelligent Adaptive Systems (DIAS), and Educational Data Mining Practices in Indian Academia (EDUDM) co-located with 10th Innovations in Software Engineering (ISEC 2017), Jaipur, India, February 5, 2017.*

Mitra, S., Prakash, C., Chakraborty, S., and Chittimalli, P. K. (2018). Matgap: A systematic approach to perform match and gap analysis among sbvr-based domain specific business rules. In *Asia Pacific Software Engineering Conference (APSEC)*.

Ross, R. G. (2003). *Principles of the business rule approach*. Addison-Wesley Professional.

Sarbanes Oxley Act (2002). Sarbanes Oxley Act. https://www.thebalance.com/sarbanes-oxley-act-and-the-enron-scandal-393497.

SBVR, O. (2008). Semantics of business vocabulary and business rules (sbvr), version 1.0.

Smullyan, R. R. (2012). *First-order logic*, volume 43. Springer Science & Business Media.

Šukys, A., Nemuraitė, L., and Butkienė, R. (2017). Sbvr based natural language interface to ontologies. *Information Technology And Control*, 46(1):118–137.

Sukys, A., Nemuraite, L., Paradauskas, B., and Sinkevicius, E. (2012). Transformation framework for sbvr based semantic queries in business information systems. *Proc. BUSTECH*, 2012:19–24.

Team, S. et al. (2006). Semantics of business vocabulary and rules (sbvr). Technical report, Technical Report dtc/06–03–02, Object Management Group, Needham, Massachusetts.

Tümer, D., Shah, M. A., and Bitirim, Y. (2009). An empirical evaluation on semantic search performance of keyword-based and semantic search engines: Google, yahoo, msn and hakia. In *Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on*, pages 51–55. IEEE.

Von Halle, B. (2001). *Business rules applied: building better systems using the business rules approach*. Wiley Publishing.