	<b>OPAALS PROJECT</b>  Contract n° IST-034824
---	---

## **WP2**

### **Automatic Code Structure and Workflow Generation from Models**

#### **Deliverable 10.14**

#### **Graphical Editor for Knowledge Representation based on SBVR**



Project funded by the European Community  
under the "Information Society Technology"  
Programme

**Contract Number:** IST-034824

**Project Acronym:** OPAALS

**Deliverable N°:** 10.14

**Due date:** November 2008

**Delivery Date:** February 2009

**Short Description:** The report is on Visual Syntax of SBVR and graphical Visual Editor of SBVR based on Visual Syntax.

**Authors:** IITK (Ashish Agrawal, Sharad Singh)

**Partners contributed:** IITK

**Made available to:** OPAALS Consortium and European Commission

Versioning		
Version	Date	Name, organization
0.1	27/12/2008	IITK
0.2	27/01/2009	IITK
0.3	11/02/2009	IITK
0.4	25/02/2009	IITK

#### Quality check

**Internal Reviewers:** Antonio Margarito (NAICA), Alexandros Marinos(UNIS)

**Dependences:**

<b>Achievements</b>	We have conceived a new graphical representation, Visual Syntax for the SBVR meta-model and developed an editor, SBVR Visual Editor, which facilitate users to develop and manage knowledge models in Visual Syntax of SBVR. SBVR Visual Editor provides good user interface with drag and drop features and better visualization for documents.
<b>Work Packages</b>	WP2
<b>Partners</b>	UNIS, NAICA
<b>Domains</b>	Computer Science Domain: This report covers knowledge representation using SBVR meta-model, an OMG standard. Introduction of SBVR is not covered in this report but methodology for creating semantic models is discussed through out the report.
<b>Targets</b>	Business Analysts, Business Knowledge Modellers, SMEs.
<b>Publications</b>	1. Visual SBVR, Prakash Musam, Sharad Singh, Prabhakar TV, Third International conference on Digital Information Management (ICDIM 2008) , London, November 2008.
<b>PhD Students</b>	Masters Students: 1. Amit Raj (IITK, 2007): Worked for foundation of SBVR Visual Syntax including its requirements. 2. Prakash Musam (IITK, 2008): Completed the representation of SBVR Visual Syntax. 3. Ashish Agrawal (IITK, current): Worked in development and testing of SBVR Visual Editor.
<b>Outstanding features*</b>	SBVR Visual Editor is the first open source editor which facilitate users to develop and create SBVR models in a graphical syntax. The source code of the project has been hosted on Source Forge and made available to all.
<b>Disciplinary domains of authors*</b>	Ashish Agrawal: Computer Science & Engineering Domain  Sharad Singh: Computer Science & Engineering Domain



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit : <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

## Contents

1. Introduction .....	6
2. Visual Syntax of SBVR Structured English .....	7
2.1 SBVR Meta-Model in Visual Syntax.....	7
2.2 Graphical Representation of SBVR Meta-Model .....	11
2.2.1 Graphical Representation of SBVR Vocabulary.....	12
2.2.2 Graphical Representation of SBVR Rules .....	13
3. SBVR Visual Editor (SBVR-VE).....	23
4. SBVR-VE Example .....	28
5. Source Code & Conclusions.....	31
6. References.....	32
Appendix A   SBVR-VE User Manual .....	33
Vocabulary Editor.....	33
Rules Editor .....	34

## List of Figures

Figure 1: Concept Map for the SBVR Structured English .....	10
Figure 2: Mapping between Logical Formulation models and SBVR keywords.....	11
Figure 3: Snapshot of Vocabulary file in SBVR-VE.....	25
Figure 4: Snapshot of SBVR rules file in SBVR-VE .....	26
Figure 5: SBVR-VE methodology for vocabulary and rules .....	27
Figure 6: Structured English representation of E2.2.1.5 from SBVR specification[1].....	29
Figure 7: SBVR Visual syntax representation of E2.2.1.5 of SBVR specification. ....	30
Figure 8: SBVR Vocabulary Editor .....	34
Figure 9: SBVR Rules Editor .....	36

## 1. Introduction

---

Semantic of Business Vocabulary and Business Rules (SBVR) [1] is an OMG standard meta-model to create semantic models of Business Vocabulary and Rules using Natural Language of business analyst from different organizations and communities. It is a representational-free meta-model which works as a bridge between business people and IT people and it is more close to business people rather than IT which contrasts it from earlier models. Target of SBVR is to provide a platform to the business people through which they can communicate their business knowledge to the IT people unambiguously using their natural language and over come the huge software failures due to miscommunication.

The existing representations of SBVR are *SBVR Structured English (SBVR SE)* [1], *RuleSpeak Business Rule Notation* [2] and *ORM Notation* [3].

In this work, we have conceived a new graphical representation, **Visual Syntax** of SBVR. The mantra of Visual syntax is to make vocabulary and rules easy to write and easy to understand. We have categorized the SBVR meta-model into tokens and keywords. We have designed graphical object for each token or keyword. A token works as a placeholder for the text associated with Name, Term or Verb used in concepts, facts and rules. Keywords are used to create rules from facts. Thus, Visual syntax provides better visualization for large documents compare to plain text and at the same time, it preserves full freedom given by SBVR meta-model.

We have implemented a graphical SBVR Visual Editor, **SBVR-VE** based on visual syntax of SBVR. SBVR-VE facilitate user to create and maintain Vocabulary files and Rules file. Vocabulary file contains Concepts (Names, Terms and Verbs) and Facts. Rule file contains Quantification, Modal and Logical Rules. SBVR-VE editor is based on eclipse platform and provides simple & interactive user interface with drag & drop functionality.

*Section 2* of this report describes Visual Syntax of SBVR. *Section 3* describes features of SBVR Visual Editor. In *Section 4* we have shown a part of the EU-Rent vocabulary of SBVR specification in Visual Syntax and compared it with Structured English representation. *Appendix A* is a short user manual of SBVR Visual Editor.

---

## **2. Visual Syntax of SBVR Structured English**

---

Representation of business knowledge using SBVR meta-model consists of two parts, 1) Business Vocabulary and 2) Business Rules. Business vocabulary is a cohesive set of interconnected concepts which organizations or communities use in their talking or writing in the course of doing business. These concepts are entities which are represented through Names/Terms and Facts (which are relation between these concepts). Business Rules define the structure of business and provide element of guidance on the actions.

Existing representations of SBVR are based on textual representation. Although these representations give freedom to write in plain English, size of documents grows very large in the case of large organization and it increases complexity in understanding these documents. SBVR Visual Syntax is based on graphical representation of objects in SBVR document. In Visual Syntax, the SBVR meta-model has been categorized into tokens & keywords. A token is a placeholder for text associated with Names, Terms and Verbs in the SBVR meta-model. The text associated with each token is represented in English. Keywords are added to facts in order to create rules. In Visual Syntax a graphical representation is assigned to each token/keyword. Visual syntax provides different kind of syntax to rules on the basis of its type. Thus, visual syntax keeps the strength of existing representations by using plain English and provides better visualization by graphical objects.

Section 2.1 describes keywords/tokens in SBVR Meta-model and Section 2.2 describes Structured English representation and Visual Syntax representation for these tokens. Intend of SBVR-VS is to provide better visualization for knowledge represented in SBVR. The basis for building SBVR-VS representation is SBVR-SE representation but aim of SBVR-VS is not to generate SBVE-SE representation. However, using the mapping between SBVR-VS and SBVR-SE (described in Section 2.2), tools can be developed to convert SBVR SE models to SBVR VS models and vice versa. These tools can help in sharing knowledge represented in different representations. Other models of SBVR like Logical Formulation can directly be generated from SBVR-VS models in the same way, as generated from SBVR-SE. Figure [2] describes the mapping of SBVR-VS keywords/tokens categories to logical formulation models of SBVR on abstract level. We have not worked on full methodology of extracting Logical formulation from models in SBVR-VS.

### **2.1 SBVR Meta-Model in Visual Syntax**

In visual syntax, SBVR meta-model has been categorized into tokens and keywords. Following is the list of tokens & keywords in SBVR Structured English meta-model. These are also explained through concept map In Figure [1]. Section 2.2 describes Structured English representations and Graphical Representations corresponding to these tokens.

1. Business Vocabulary Tokens
  - a. Name
  - b. Term
  - c. Verb
2. Business Rules Keywords
  - a. Quantification keywords
    - i. each
    - ii. at least 1
    - iii. at least n
    - iv. at most 1
    - v. at most n
    - vi. at least n and at most m
    - vii. exactly 1
    - viii. exactly n
    - ix. more than 1
    - x. some
  - b. Modal keywords
    - i. Obligation
      1. It is obligatory that
      2. must
    - ii. Permission
      1. It is permitted that
      2. may
    - iii. Prohibition
      1. It is prohibited that
      2. must not
    - iv. Necessity
      1. It is necessary that
      2. always
    - v. Possibility
      1. It is possible that
      2. sometimes
      3. can
    - vi. Impossibility
      1. It is impossible that
      2. never
  - c. Logical keywords
    - i. It is not the case that P
    - ii. P and Q
    - iii. P or Q



- iv. if P then Q
- v. if P
- vi. P if and only if Q
- vii. not both P and Q
- viii. neither P nor Q
- ix. P whether or not Q
- x. P or Q but not both
- d. Other keywords
  - i. a, an, the, another, etc.

The tokens & keywords of the SBVR structured English are shown in the following Concept Map.

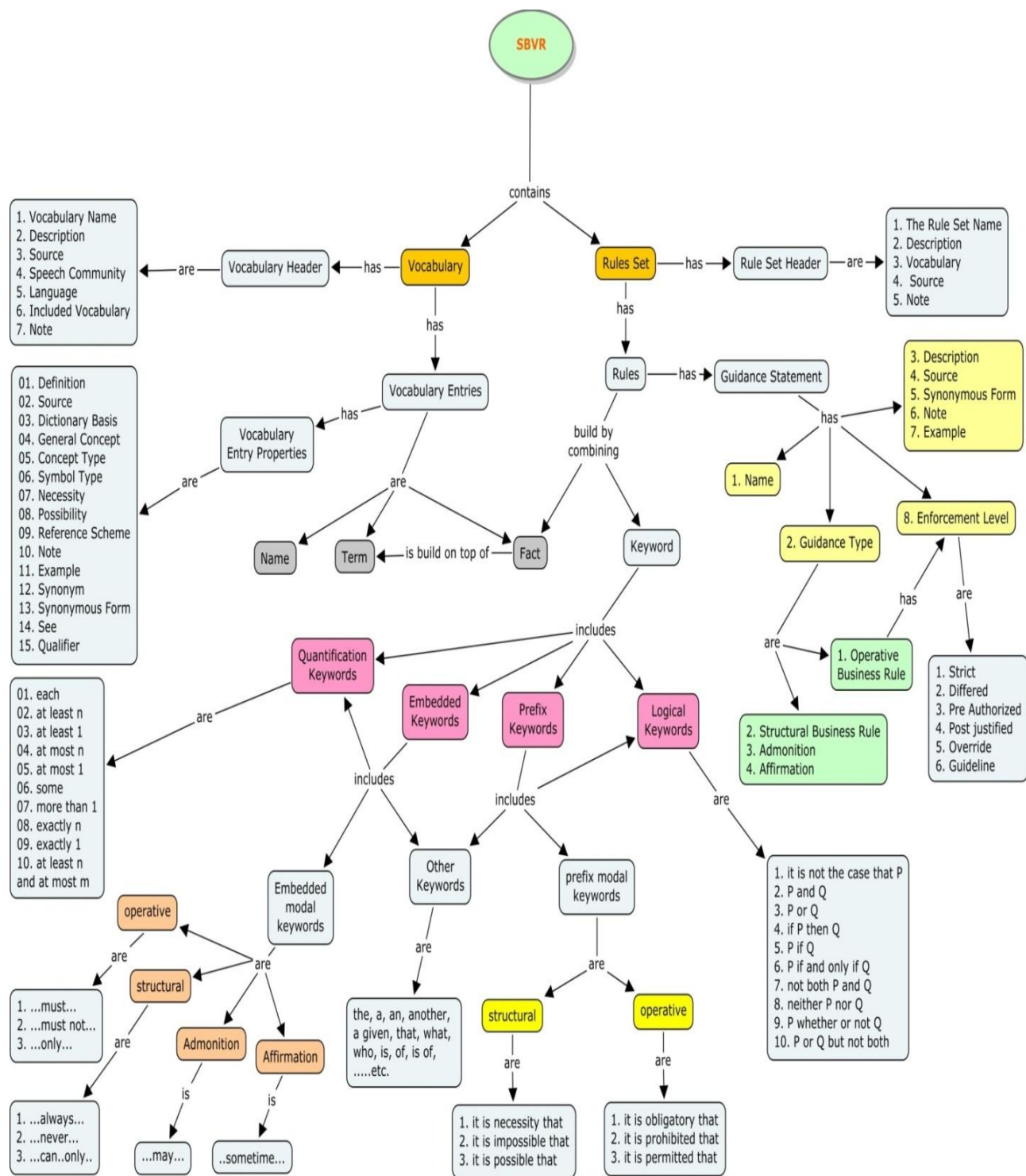


Figure 1: Concept Map for the SBVR Structured English

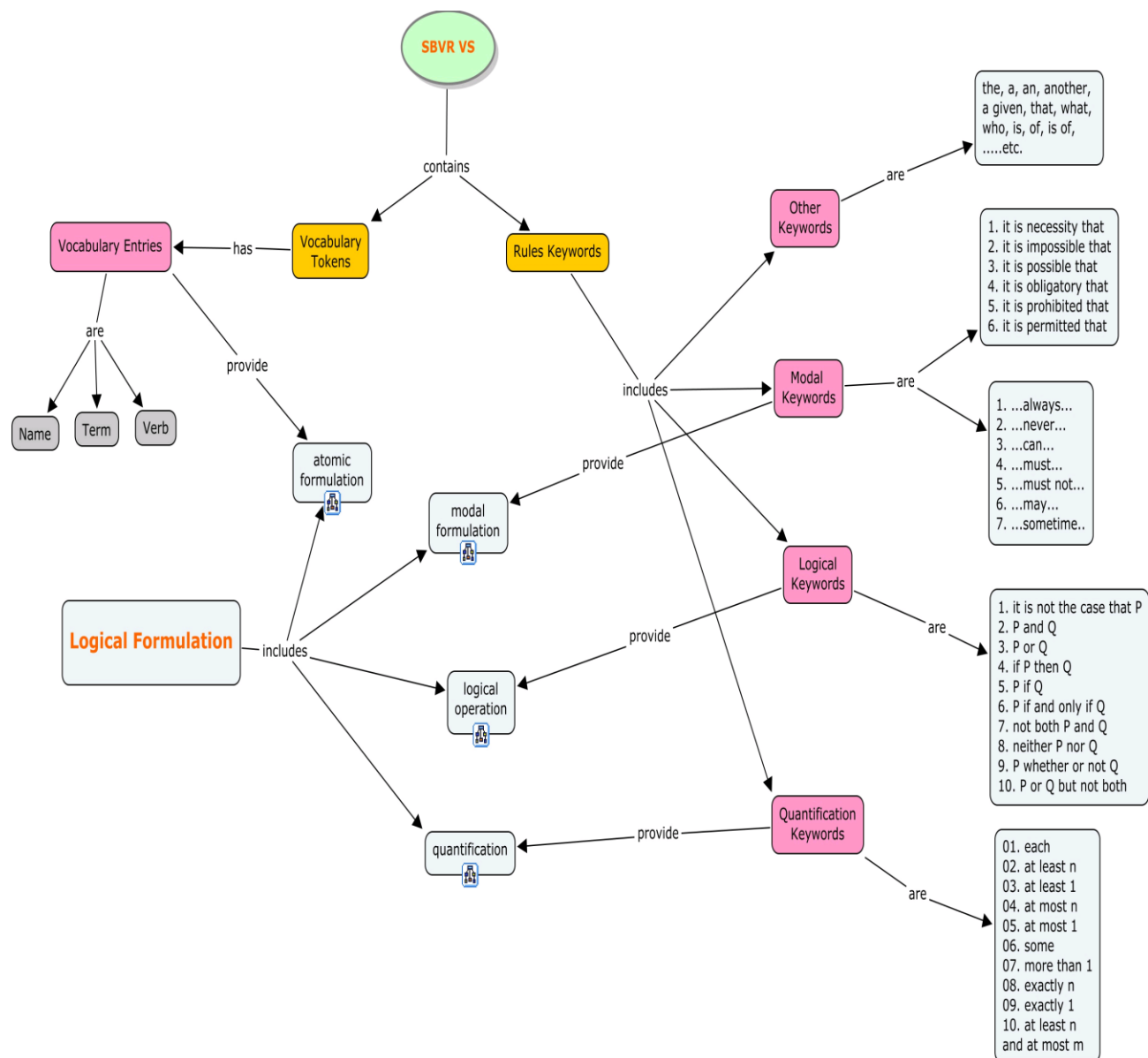


Figure 2: Mapping between Logical Formulation models and SBVR keywords

## 2.2 Graphical Representation of SBVR Meta-Model

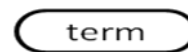
### 2.2.1 Graphical Representation of SBVR Vocabulary

#### 1. Tokens of SBVR Vocabulary

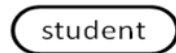
SBVR Vocabulary consists of business concepts (Terms, Names and Verbs) and facts.

- **term**: In SBVR Structured English [1], the ‘term’ font is used for a designation for a noun concept. Terms are usually defined using lower case letters unless they include a proper noun. Terms are defined in singular form. Plural forms are implicitly available for use.

In Visual Syntax, Term is represented with this symbol



E .g.

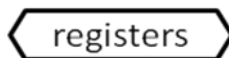


- **verb**: In SBVR Structured English [1], the ‘verb’ font is used for designations for fact types — usually a verb, preposition or combination thereof. Such a designation is defined in the context of a form of expression.

In Visual Syntax, Verb is represented with this symbol

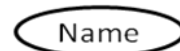


E .g.

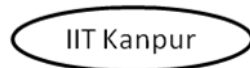


- **Name**: In SBVR Structured English [1], the ‘Name’ font is used for a designation of an individual concept — a name. Names tend to be proper nouns (e.g., [IIT Kanpur](#)). This style is applied to a name where it is defined and wherever it is used.

In Visual Syntax, Name is represented by an ellipse.



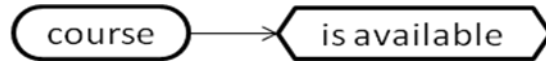
E .g.



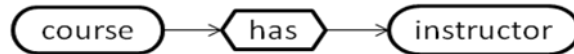
#### 2. Fact formation

We use these tokens to form facts. These facts may be unary or binary facts which are further used to form tertiary or n-ary fact. A fact is formed by connecting a term/name, verb with a term/name sequentially.

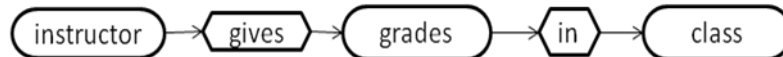
- “[course](#) is available”



- “course has instructor” is a binary fact which can be graphically represented as:



- “instructor gives grades in class” is a tertiary fact which can be graphically represented as:



## 2.2.2 Graphical Representation of SBVR Rules

### 1. Tokens of SBVR Rules

It consists of several keywords.

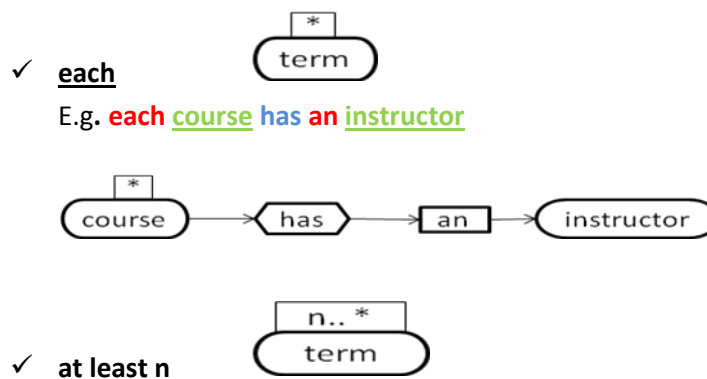
- **Keywords:** SBVR Structured English [1], the ‘keyword’ font is used for linguistic symbols used to construct statements. There are some in built keywords in SBVR which are of great interest for constructing the rules. These keywords are combined with terms and fact types to create the rules. The categorization of keywords is shown below:

The categories of keywords are:

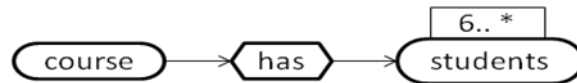
1. Quantification Keywords
2. Modal Keywords
3. Logical Keywords
4. Other Keywords

#### a) Quantification Keywords

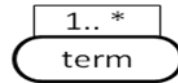
The Visual Syntax representation of these keywords is as follows.



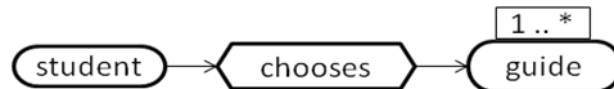
E.g. course has at least six students



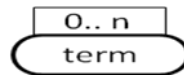
✓ at least 1



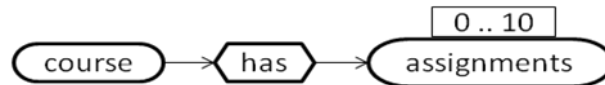
E.g. student chooses at least one guide



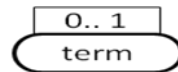
✓ at most n



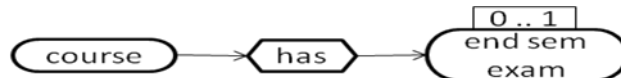
E.g. course has at most ten assignments



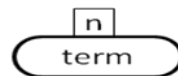
✓ at most 1



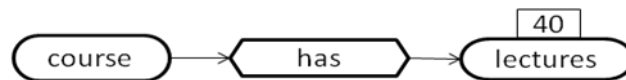
E.g. course has at most 1 end sem exam



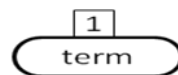
✓ exactly n



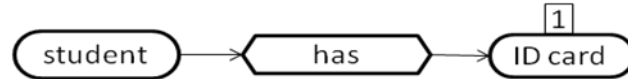
E.g. course has exactly forty lectures



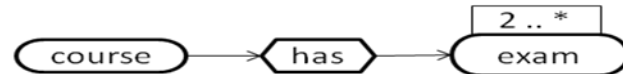
✓ exactly 1



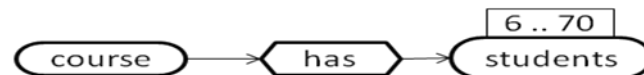
E.g. student has exactly one ID card



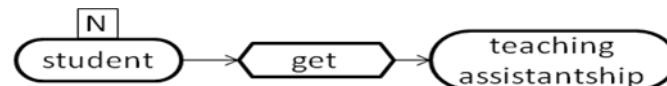
E.g. course has more than one exam



E.g. course has at least 6 and at most 70 students



E.g. some students get teaching assistantship



## b) Modal Keywords

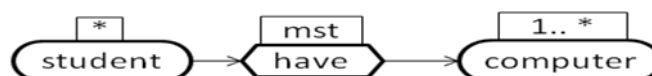
These are sub-divided into two categories

- i) **Embedded Modal Keywords:** These are further sub – divided into two categories.

- ✓ **Operative:** The representation of these keywords is as follows

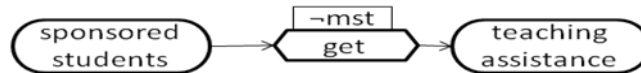


E.g. each student must have at least one computer

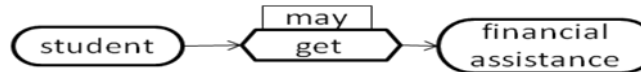




E.g. sponsored students must not get teaching assistantship



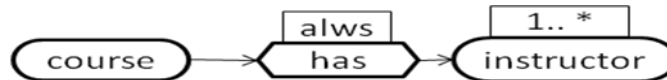
E.g. student may get financial assistantship



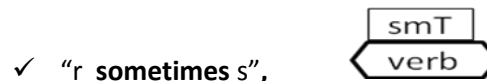
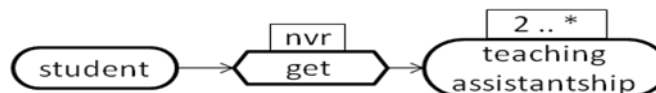
✓ **Structural:** The representation of these keywords is as follows.



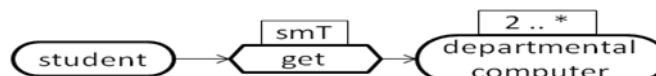
E.g. course always has at least one instructor



E.g. student never get more than one teaching assistantships

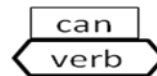


E.g. student sometimes get more than one departmental computer

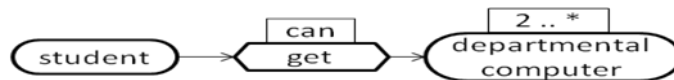




✓ "r can s",



E.g. student can get more than one departmental computer

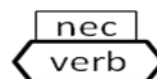


## ii) Prefix Modal Keywords

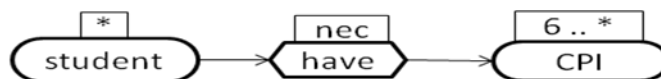
They are divided into two categories:

✓ **Structural** : : The representation of these keywords is as follows

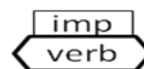
✓ it is necessary that



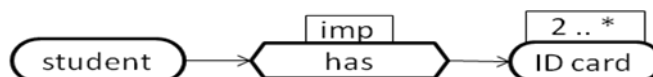
E.g. it is necessary that each student have more than 6 CPI



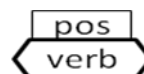
✓ it is impossible that



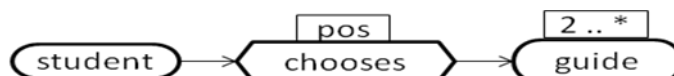
E.g. it is impossible that student has more than one ID card



✓ it is possible that

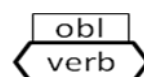


E.g. it is possible that student chooses more than one guide

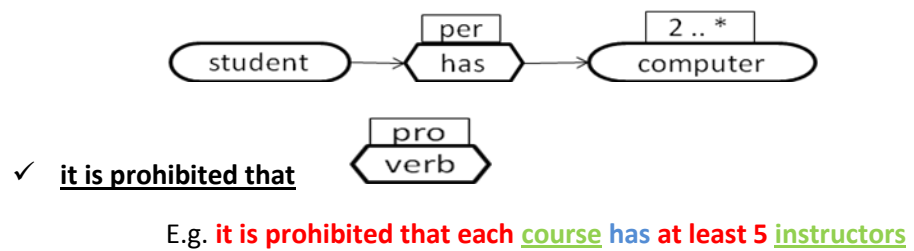
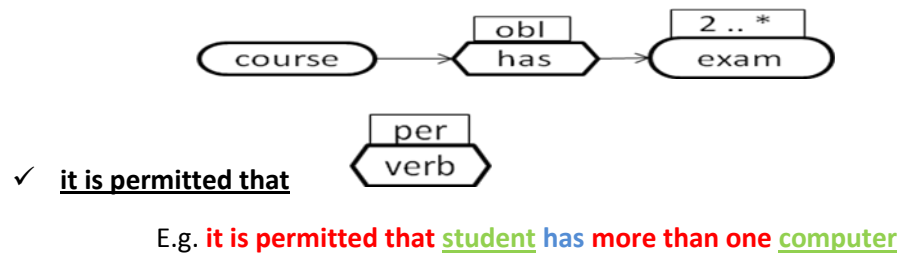


✓ **Operative** : : The representation of these keywords is as follows

✓ it is obligatory that

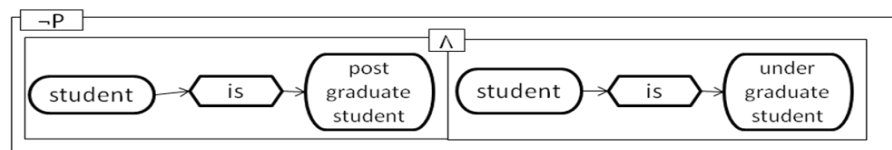
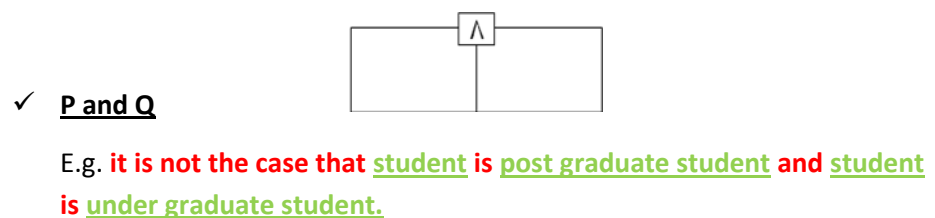
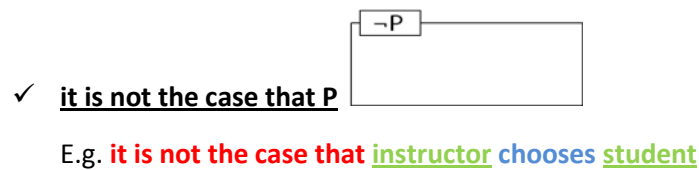


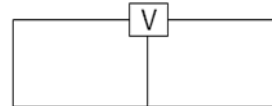
E.g. it is obligatory that each course more than one exam



### c) Logical Keywords

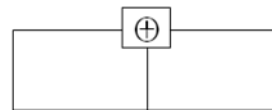
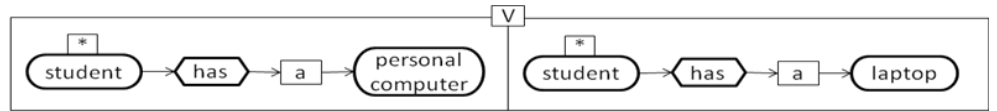
The representation of these keywords is as follows.





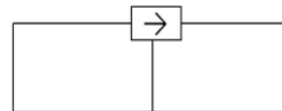
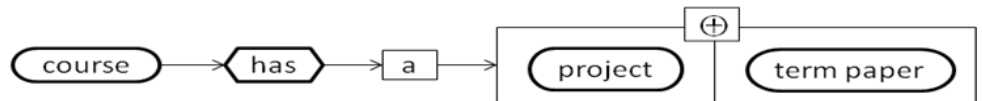
✓ **P or Q**

E.g. **each** student **has** **a** personal computer **or** **each** student **has** **a** laptop



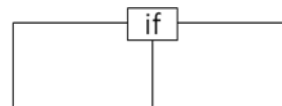
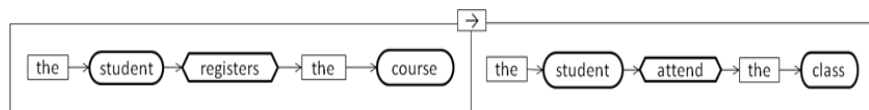
✓ **P or Q but not both**

E.g. course **has** **a** project **or** **a** term paper **but not both.**



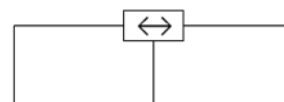
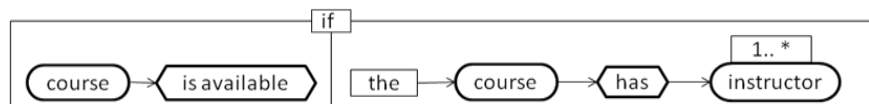
✓ **if P then Q**

E.g. **if** **the** student **registers** **the** course **then** **the** student **attends** **the** class



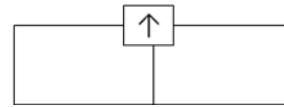
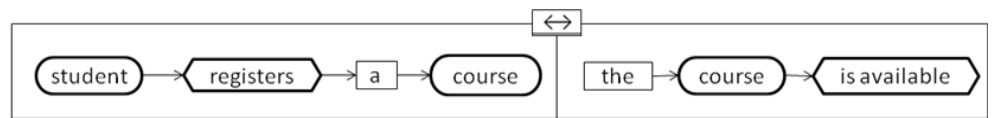
✓ **Q if P**

E.g. course **is available** **if** **the** course **has** **at least one** instructor



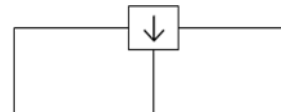
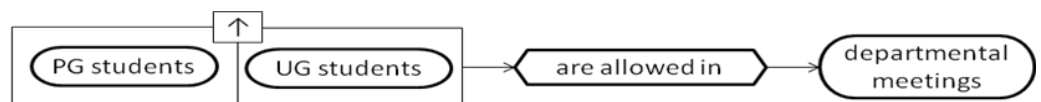
✓ **P if and only if Q**

E.g. student registers a course if and only if the course is available



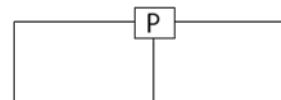
✓ **not both P and Q**

E.g. **not both** PG students and UG students are allowed in departmental meetings



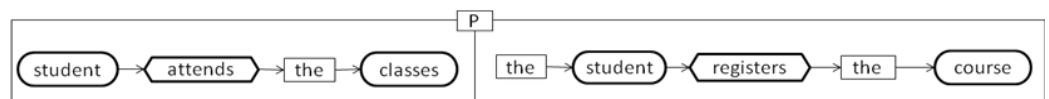
✓ **neither P nor Q**

E.g. departmental laptops are given to **neither** PG students **nor** UG students



✓ **P whether or not Q**

E.g. student attends the classes **whether or not** the student registers **the** course



#### d) **Other Keywords**

There are few keywords which are not included in any of the above categories. They can be added to the fact types while creating the rules. Some of the other keywords are “a”, “an”, “the”, “another”, “a given”, “that”, “who”, “what” etc.

✓ It is represented as

keyword

E.g. that

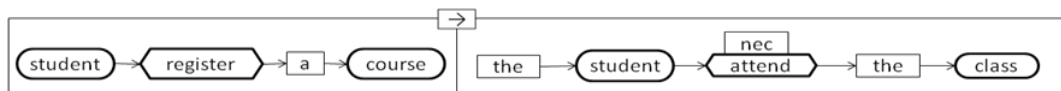
These keywords are used to make the sentence more readable. SBVR-VS targets to provide better visualization to the document and makes it easy to understand. These optional keywords facilitate users to provide more readability to the document and can be useful in the case of converting SBVR VS to other representations.

## 2. Rule Formation

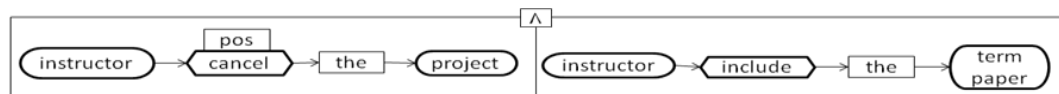
The SBVR rules are categorized as follows:

- a) **Structural Rules:** These are the rules which can represent how the business is organized. These can be formed by adding the structural modal keywords like “it is necessary that”, “it is possible that”, “it is impossible that”.

E.g. if student registers a course then it is necessary that the student attends the class



E.g. it is possible that the instructor cancel the project and the instructor include a term paper



E.g. it is impossible that more than one department has the same course

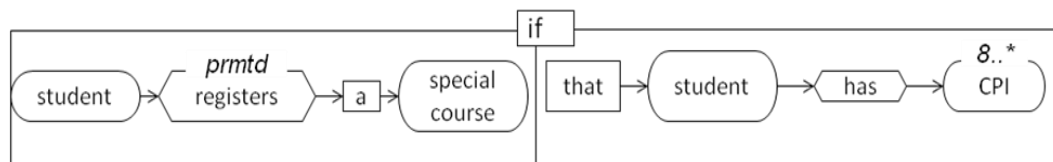


- b) **Operative Rules:** These are the rules which guide the behavior of the business. These can be formed by adding the operative modal keywords like “it is obligatory that”, “it is permitted that”, “it is prohibited that”.

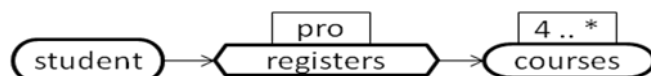
E.g. it is obligatory that each student of a course attends at least 20 classes



E.g. **it is permitted that** student **registers a** special course **if that** student **has more than** **8** CPI



E.g. **it is prohibited that** student **registers more than 4** courses



### 3. SBVR Visual Editor (SBVR-VE)

---

We have implemented a graphical editor, **SBVR Visual Editor (SBVR-VE)** based on visual syntax of SBVR. SBVR-VE is based on eclipse platform. It provides simple and user-friendly interface with drag and drop features.

SBVR-VE has following parts:

1. **Vocabulary Editor:** This is an editor for creating business vocabulary including Names, Terms and Facts.
2. **Rule Editor:** This editor is a set of editors for creating different kind of rules from business vocabulary. Parts of Rule Editor are
  - a. **Embedded Editor:** Editor for adding quantification keywords in facts or intermediate rules.
  - b. **Modal Editor:** Editor to add modal keywords in facts or intermediate rules.
  - c. **Logical Editor:** Facilitate user to create logical rules from facts/rules.

Different parts of Vocabulary and Rule editor are explained in Appendix A (SBVR VE User Manual).

Workflow for creating semantic model in SBVR-VE follows the business rules mantra [1] “Rules are based on facts and facts are based on concepts”. Procedure for creating vocabulary and rules are explained in Figure [5]. So for creating semantic models, first step is to specify concepts and create facts by connecting them. Rules are created by adding different kinds of keywords (explained in section 2.2.2) to the facts. According to the type of the keyword, rules are categorized into three types, 1) Embedded Rules (from quantification keywords and embedded modal keywords), 2) Prefix Modal Rules (from prefix modal keywords) and 3) Logical Rules (from logical keywords). A final rule can have multiple keywords of different types. So the process for final rule creation starts from facts, iterates through intermediate rules (of above 3 types, by adding keywords one by one) and reaches to its final state once all the keywords have been added to it. During the creation of Rules, any change in the form of a concept (singular to plural or plural to singular) has to be done manually by the user.

Some Features of SBVR-VE are:

#### 1. SBVR Visual Syntax

The biggest strength of SBVR-VE is that it is based on SBVR Visual Syntax which provides better visualization than Structured English. Visual syntax makes understanding the business knowledge very easy and the text associated with graphical objects is still represented in natural language. Visual Syntax helps in visualizing vocabulary and rules in tree structure.

#### 2. Easy modeling using Drag & Drop options

In SBVR Visual syntax, SBVR meta-model is categorized in tokens/keywords as in Figure [1]. In SBVR-VE, we have graphical object for each token and it shows these tokens on the palette view according to type of editor. To use any graphical object, User can drag and drop the desired token from palette to the main view. This makes creating model in SBVR-VE very easy.

### **3. Validation to SBVR Meta-model**

SBVR-VE checks validation of any operation done by user and adhere to SBVR Meta-Model. For example if a user tries to create any fact without any verb concepts SBVR-VE will show error. SBVR-VE provides tokens only available in the SBVR model. So at the time of giving freedom to user, it also restricts them to be in SBVR domain.

### **4. Outline View**

For better user interface, SBVR-VE also provides outline view for each type of editor in which data in the file is represented in tree structure. In vocabulary file, in outline view it represents tree of Names, Terms, Verb and textual representation of facts. In Rules editors, it shows tree structure of textual representation of rules. Outline view make the document more comprehensive and helps user in navigation.

### **5. Properties View:**

SBVR meta-model captions are handled as properties of elements in SBVR-VE. SBVR-VE separates these properties associated with any concept, fact or rule from its graphical representation. It shows the properties attributes in the properties view panel where user can directly put the values of attributes. By default it shows the available attributes corresponding to the type of object (i.e. concept, fact and rule). This way SBVR-VE reduces size of main view of vocabulary or text.

### **6. Easy Editing**

Graphical editor makes future editing in the document very easy. Changes in any concept will automatically reflect in all facts and rules built on that concept.

### **7. Separation of Vocabulary and Rules**

SBVR-VE separates vocabulary and rules file. In rules file it provides different kind of editors based on type of rule user want to make. Although vocabulary and rules files are in separate files, SBVR-VE keeps synchronization between rules and its corresponding file. At the time of opening a new Rule file for an existing vocabulary file, SBVR-VE automatically fetches all the facts available for rule creation. Changes in vocabulary file will also be reflected in rules file.

A user manual for working in SBVR-VE is added in Appendix A. Figure [3] is a snapshot of vocabulary editor in SBVR-VE. Figure [4] is a snapshot of rules editor in SBVR-VE. Figure [5] shows SBVR-VE methodology of how to create facts and different kind of rules.



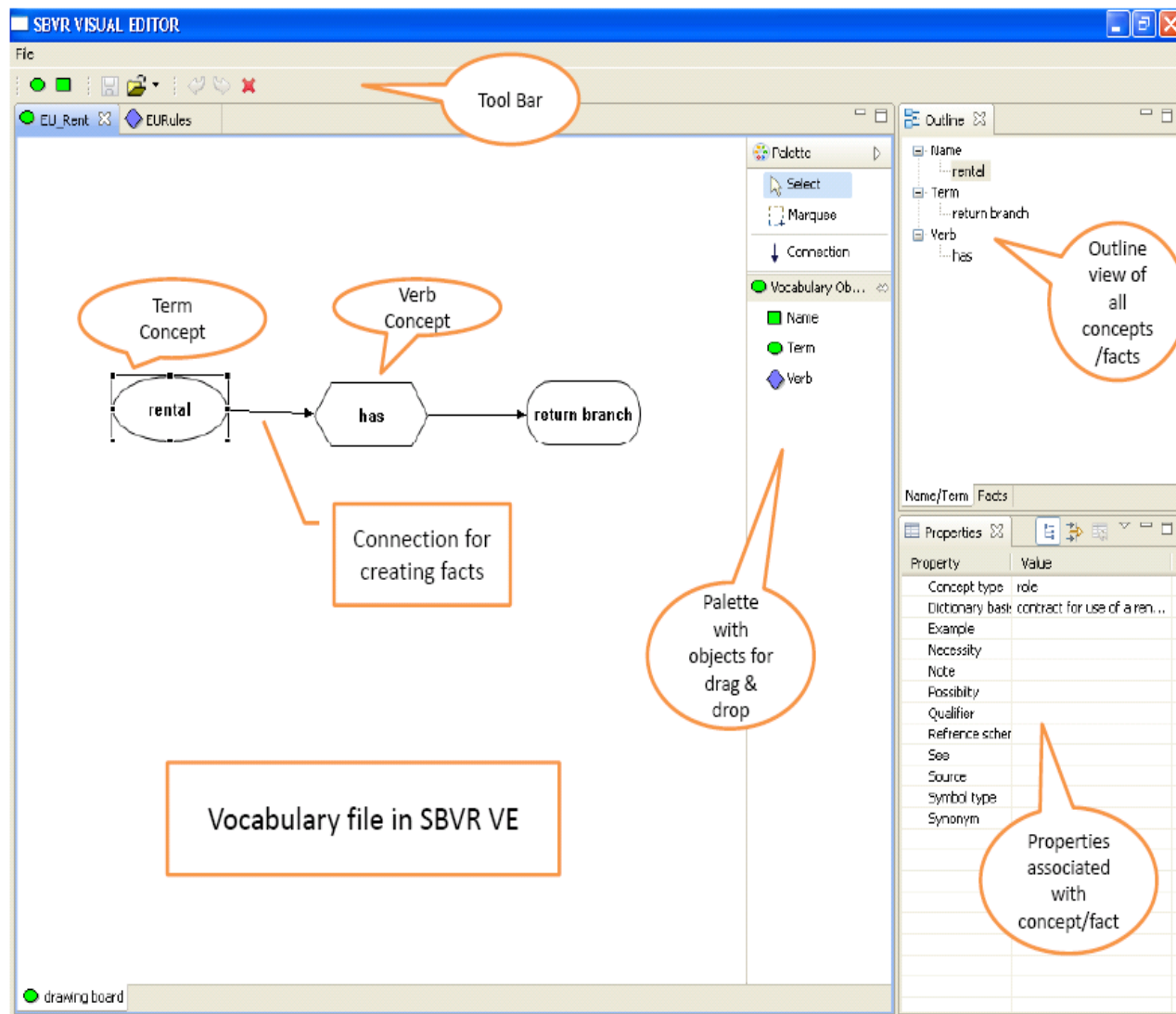
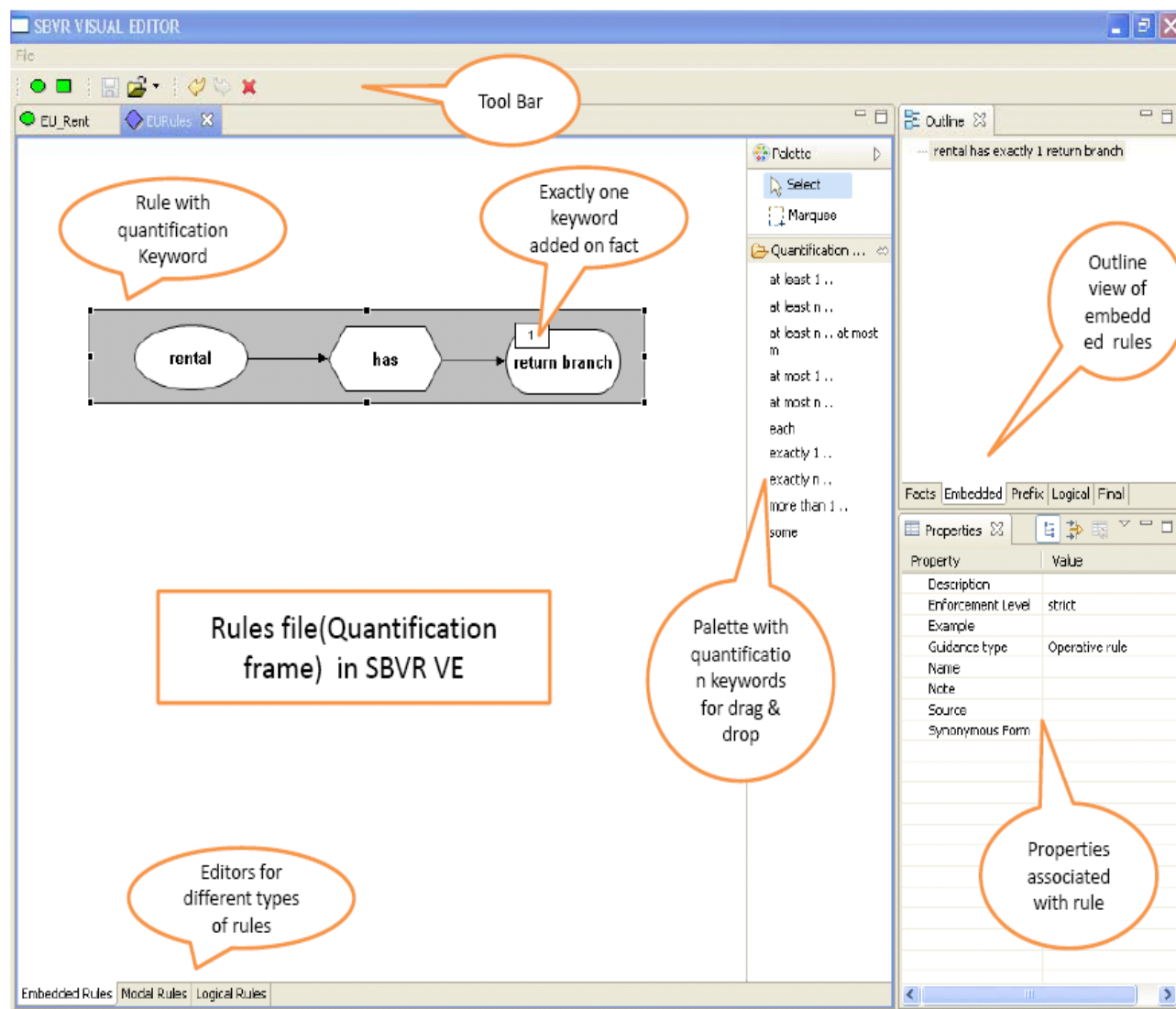


Figure 3: Snapshot of Vocabulary file in SBVR-VE



**Figure 4: Snapshot of SBVR rules file in SBVR-VE**

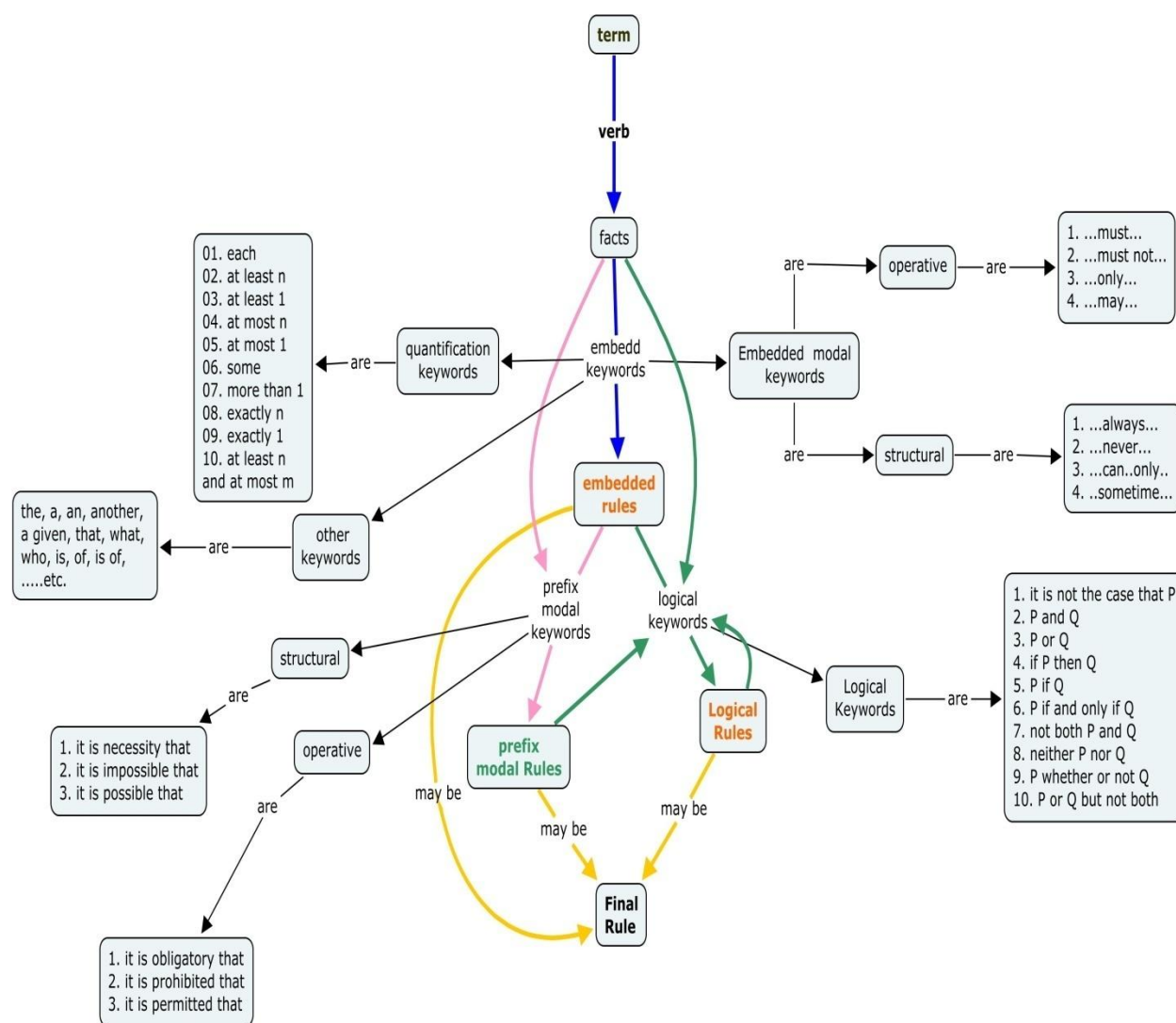


Figure 5: SBVR-VE methodology for vocabulary and rules

## 4. SBVR-VE Example

---

In this section, example of Visual syntax is shown for a small vocabulary. We have taken a part of vocabulary of EU-Rent organization from example in SBVR specification [1, section E2.2.1.5]. Figure [6] is a snapshot of SBVR Structured English representation of E2.2.1.5 section of EU-Rent example in SBVR specification. Figure [7] is the main view of SBVR Visual Editor for the same vocabulary.

Few points are worthy to be noted here,

1. In Visual Editor, attributes related to any concept or fact (e.g. Definition, Synonyms etc.) are removed from the main view and stored in Properties view (as shown in Figure [3]).
2. Main view contains concepts and connections between concepts. Boundary of a fact (whether it is n-ary fact or binary fact) is not shown in Main view but can be extracted from Outline view.

This example is presented to show how a vocabulary will look like in SBVR Visual Editor. Although Figure [7] does not contain full view of vocabulary (outline & properties view), but few benefits of document structure in SBVR Visual Editor can be noted here.

1. It is obvious from the figure that visualizing number of concepts in a vocabulary is very easy in the case of Visual Editor. In Visual Editor a concept is written only once. This also gives reusability of concepts.
2. It is very easy to identify whether a concept is term, name or verb due to different graphical shapes for them. Similarly for rules, it is easy to locate what kinds of restrictions (quantification, modal etc.) are applied on a rule.
3. Due to graphical notation and connections, Visual editor helps in visualizing concepts and relations between them (direct or indirect) in a tree/graph structure. For example in Figure [7], facts are shown in a structure, centered at concept “rental”. Here, figuring out the exact relations for the concept will require help of outline view.

Necessity:	The <u>renter provides current contact details</u> .
Necessity:	Each <u>driver of the rental</u> has a <u>valid driver license</u> .
<b>actual pick-up date/time</b>	
Concept Type:	<u>role</u>
General Concept:	<u>dateTime</u>
Definition:	<u>dateTime</u> when the <u>rented car of a rental</u> is picked up by the <u>renter</u>
<b>actual return dateTime</b>	
Concept Type:	<u>role</u>
General Concept:	<u>dateTime</u>
Definition:	<u>dateTime</u> when a <u>rented car</u> is returned to EU-Rent
<b>current contact details</b>	
Concept Type:	<u>role</u>
Definition:	<u>contact details</u> that have been confirmed as up-to-date by the <u>renter</u>
<b>optional extra</b>	
Definition:	Item that may be added to a <u>rental</u> at extra charge if the <u>renter</u> so chooses
Example:	<u>One-way rental, fuel one-payment, additional insurances, fittings (child seats, satellite navigation system, etc. etc.)</u>
Source:	CRISG [optional extra]
<b>pick-up branch</b>	
Concept Type:	<u>role</u>
General Concept:	<u>branch</u>
Necessity:	The <u>pick-up branch of a rental</u> is not changed.
Note:	If the <u>renter</u> wishes to change the <u>pick-up branch</u> of a <u>rental</u> , EU-Rent regards it as a cancellation and a new <u>rental</u> .
<b>rejected request for pick-up</b>	
Definition:	<u>request for pick-up</u> that is rejected by EU-Rent
<b>rental</b>	
Definition:	contract with a <u>renter</u> specifying use of <u>some car of a car group</u> for a <u>rental period</u> and a <u>car movement</u>
Dictionary Basic:	contract for use of a rental car by a renter for an agreed period under the rental company's terms and conditions for rental. [CRISG]
<b>rental contract document</b>	
Definition:	information artifact that is the manifestation of a <u>rental</u>
<b>rental duration</b>	
Concept Type:	<u>role</u>
Definition:	<u>duration</u> used to calculate a <u>rental charge</u>
<b>rental includes rental period</b>	
Concept Type:	<u>partitive fact type</u>
Synonymous Form:	<u>rental period is included in rental</u>
Necessity:	Each <u>rental includes exactly one rental period</u> .
Note:	The <u>rental period</u> may be changed by rescheduling at the renter's request, by early or late arrival for rental, and by late return from rental.
<b>rental is manifested in rental contract document</b>	
Concept Type:	<u>associative fact type</u>
<b>rental period</b>	
Concept Type:	<u>role</u>
Definition:	<u>variable period</u> that is included in a <u>rental</u>
<b>rented car</b>	
Concept Type:	<u>role</u>
Definition:	<u>rental car</u> that is assigned a <u>rental</u>
<b>rented car is assigned to rental</b>	
Synonymous Form:	<u>rental has rented car</u>
Necessity:	A <u>rented car is assigned to a rental</u> if and only if the <u>rented car is the rental car that is assigned to the car movement that is included in the rental</u>
<b>renter has possession of rented car</b>	
Definition:	the <u>renter</u> has the <u>rented car</u> for use on rental
Synonymous Form:	<u>rented car is in the possession of renter</u>
<b>request for pick-up</b>	
Definition:	request from a <u>renter</u> to pick up the <u>rental car of a rental</u> that has been reserved by him
<b>request for pick-up matches rental</b>	
Necessity:	The <u>rental is assigned</u> (has a rental car assigned to it).
Necessity:	The <u>pick-up branch of the rental</u> is the <u>branch</u> at which the request is made
Necessity:	The <u>renter of the rental</u> is the person making the request for pick-up.
Necessity:	The <u>scheduled start date of the rental</u> is the day of the request for pick-up.
Note:	This entry is partly informal in order to limit the case study size.
<b>requested car group</b>	
Concept Type:	<u>role</u>
Definition:	<u>car group</u> that is requested for a rental
Necessity:	At a given <u>dateTime</u> each <u>rental has exactly one requested car group</u> .

<b>rental duration is measured in rental time unit</b>	
Necessity:	Each <u>rental duration</u> is measured in a <u>whole number of rental time units</u> .
Example:	A rental with an end date/time that was 11 days and 7 hours after its start date/time would have a rental duration of 1 x 1-week RTU plus 5 x 1-day RTU. If EU-Rent were to introduce a 3-day RTU, this would change to 1 x 1-week RTU plus 1 x 3-day RTU plus 2 x 1-day RTU.
<b>rental has actual pick-up date/time</b>	
Definition:	<u>rental has actual pick-up date/time</u> .
Necessity:	A <u>rental has an actual pick-up date/time</u> if and only if the <u>actual pick-up date/time</u> is the <u>start date/time of the rental period that is included in the rental</u> .
<b>rental has actual return date/time</b>	
Necessity:	A <u>rental has an actual return date/time</u> if and only if the <u>actual return date/time</u> is the <u>end date/time of the rental period that is included in the rental</u> .
<b>rental has pick-up branch</b>	
Necessity:	A <u>rental has a pick-up branch</u> if and only if the <u>pick-up branch</u> is the <u>pending branch of the car movement that is included in the rental</u> .
<b>rental has rental duration</b>	
Necessity:	A <u>rental has a rental duration</u> if and only if the <u>rental duration</u> is the <u>duration of the period that is the rental period that is included in the rental</u> .
<b>rental has requested car group</b>	
Necessity:	A <u>rental has a requested car group</u> if and only if the <u>requested car group</u> is the <u>car group that is specified in the car movement that is included in the rental</u> .
Possibility:	The <u>requested car group of an advance rental</u> is changed before the <u>actual pick-up date/time of the advance rental</u> .
Necessity:	The <u>requested car group of an advance rental</u> is not changed after the <u>actual pick-up date/time of the advance rental</u> .
<b>rental has return branch</b>	
Necessity:	A <u>rental has a return branch</u> if and only if the <u>return branch</u> is the <u>receiving branch of the car movement that is included in the rental</u> .
<b>rental has scheduled pick-up date/time</b>	
Necessity:	A <u>rental has a scheduled pick-up date/time</u> if and only if the <u>scheduled pick-up date/time</u> is the <u>scheduled start date/time of the rental period that is included in the rental</u> .
<b>rental has scheduled return date/time</b>	
Necessity:	A <u>rental has a scheduled return date/time</u> if and only if the <u>scheduled return date/time</u> is the <u>scheduled end date/time of the rental period that is included in the rental</u> .
<b>rental includes car movement</b>	
Concept Type:	<u>partitive fact type</u>
Synonymous Form:	<u>car movement is included in rental</u>
<b>return branch</b>	
Concept Type:	<u>role</u>
Definition:	<u>branch</u> stipulated in the <u>rental contract</u> for return of the <u>rented car</u>
Note:	If the renter does not return the car to the location of this branch, a penalty charge will be levied.
Necessity:	Each <u>rental has exactly one return branch at a given dateTime</u> .
Possibility:	The <u>return branch of a rental</u> is changed before the <u>actual return date/time of the rental</u> .
<b>scheduled pick-up date/time</b>	
Definition:	<u>date/time</u> at which a <u>rented car</u> is scheduled to be picked up from EU-Rent
Note:	The <u>possibilities and necessities for changing the start date/time of variable period</u> apply to <u>scheduled pick-up date/time of rental</u> .
<b>scheduled return date/time</b>	
Definition:	<u>date/time</u> at which a <u>rented car</u> is scheduled to be returned to EU-Rent
Note:	The <u>possibilities and necessities for changing the end date/time of variable period</u> apply to <u>scheduled return date/time</u> .
<b>valid credit card</b>	
Concept Type:	<u>role</u>
Definition:	<u>credit card</u> that is acceptable for payment of the <u>rental charges of the rental</u> for which it is presented
Necessity:	The card is of a type that EU-Rent accepts.
Necessity:	"Expiry date" on the <u>valid credit card</u> is after the <u>scheduled end date/time of the rental</u> .
Note:	"Cardholder" is the person presenting the card. This entry is informally defined in order to limit the case study size.
<b>valid driver license</b>	
Concept Type:	<u>role</u>
Definition:	<u>driver license</u> that is acceptable for the <u>rental</u> for which it is presented
Necessity:	"Expiry date" on the <u>valid driver license</u> is after the <u>scheduled end date/time of the rental</u> .
Necessity:	"Driver" is the person presenting the license.
Necessity:	The <u>rented car</u> falls within "vehicle types."
Note:	The license is legally acceptable in the country of the pick-up branch. This entry is informally defined in order to limit the case study size.
<b>E.2.2.1.6 Rental Categorization</b>	
This subclause defines some categorizations of rental, which enable some subsequent simplification of formulations and representations.	
It also introduces the use of characteristic type for defining states - see <u>rental state</u> and <u>advance rental state</u> .	

Figure 6: Structured English representation of E2.2.1.5 from SBVR specification[1].

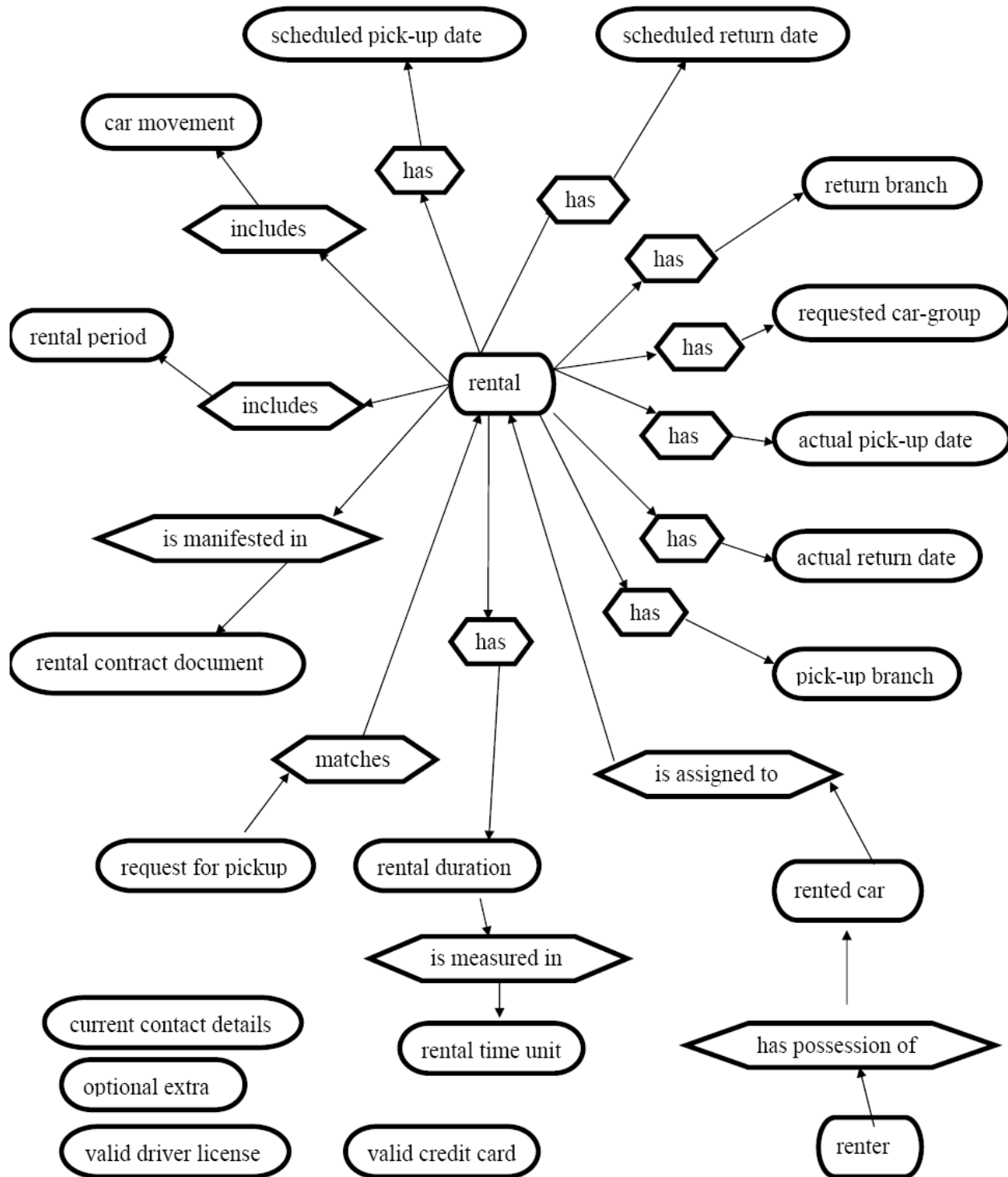


Figure 7: SBVR Visual syntax representation of E2.2.1.5 of SBVR specification.

## ***5. Source Code & Conclusions***

---

In this report we have described graphical knowledge representation based on SBVR. This representation, SBVR Visual Syntax provides better visualization of SBVR documents. We have also implemented a graphical editor, SBVR Visual Editor (SBVR-VE) based on Visual Syntax. Current editor provides features for creating and maintaining vocabulary and rules file. In future, it could have features to convert its graphical representation into interchangeable xml format and vice versa. This will facilitate both users and developers to integrate visual editor with other SBVR based tools.

The pre-alpha release of source code has been hosted on SourceForge.net. Project home URL is ([www.sourceforge.net/projects/sbvrve/](http://www.sourceforge.net/projects/sbvrve/)).

## 6. References

---

- [1]. Semantics of Business Vocabulary and Business Rules (SBVR), Object Management Group, <http://www.omg.org/spec/SBVR/1.0/PDF/>.
- [2]. RuleSpeak Business Rule Notation, Business Rules Solutions [Ross2003], ([www.brsolutions.com](http://www.brsolutions.com))
- [3]. Object-Role Modeling (ORM) notation, Halp[1998,2001].
- [4]. Visual SBVR, Prakash Musam, Sharad Singh, Prabhakar TV, Third International conference on Digital Information Management (ICDIM 2008) , London, November 2008.
- [5]. Business Rules Community, ([www.brcommunity.com](http://www.brcommunity.com) )



## Appendix A SBVR-VE User Manual

---

To start working with the SBVR Visual Editor (SBVR VE) the user will have to create a vocabulary. On top of that vocabulary he can create a rule. This walkthrough about using SBVR VE is divided in two parts. The first part deals about how to work with Vocabulary editor and the second part deals with how to work with rules editor.

### Vocabulary Editor

#### *Parts of vocabulary editor:*

1. **Editor:** editor is the area where the graphical objects will be created
2. **Outline view:** shows all the objects created on the editor
  - a. **Name/Term** tab: Display all Names/Terms/Verbs created in file
  - b. **Facts** tab: Display all facts created in file.
3. **Property view:** shows and let the user edit several properties of the graphical objects which are captions in SBVR meta-model.
4. **Palette:** this contains all the graphical objects which can be dragged and dropped on the editor.

#### *How to:*

1. Open a new/existing Vocabulary File:
  - a. Options for opening file are in file menu as well as on Tool bar.
2. Create a Name/Term/Verb.
  - a. Select Name/Term/Verb object from palette and drop to the editor. Text associated with it can be edited by clicking on it. These objects are movable inside editor area.
3. Create a connection:
  - a. Select Connection (Arrow sign) from palette and connect any two objects.
4. Create a Fact:
  - a. Select all objects in the fact in order of start to end concept, one by one (using ctrl). Right click and select "Create Fact" from context menu of the selected verb on which fact should be created. If the same fact is already created then it will show error message. If selected sequence is not valid then also error message will be displayed. Note that objects in the fact should be selected in the same sequence as they appear in the fact.
5. Change Properties of Name/Verb/Fact:
  - a. Select Name/Verb/Fact from editor view or outline view. In the properties view, all default properties attributes will be displayed. Fill the value of available attributes.
6. Save a File
  - a. Use ctrl+s or Save option in tool bar.
7. View created facts:
  - a. Select Facts tab in outline view. All facts with their textual representation will be shown.

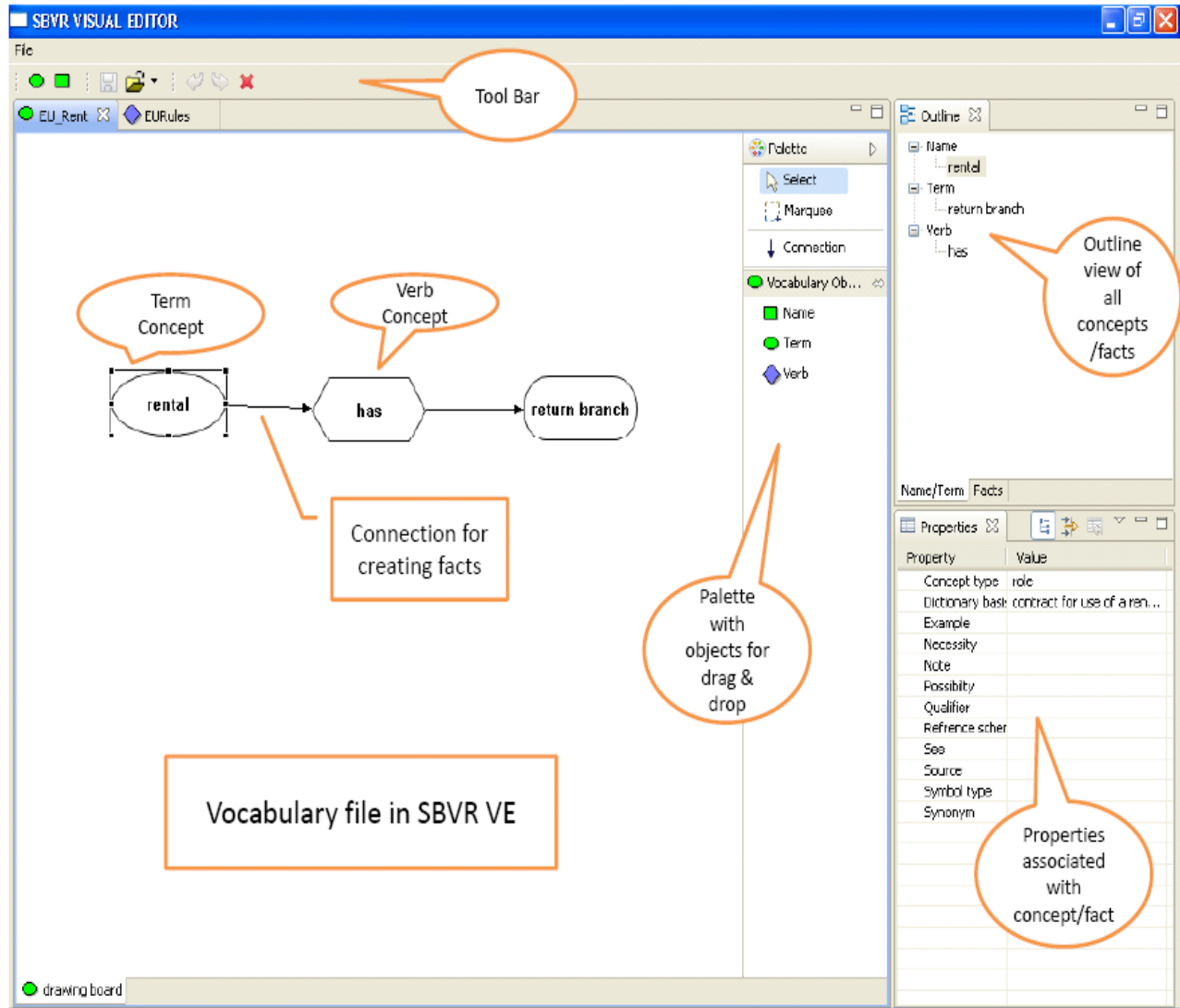


Figure 8: SBVR Vocabulary Editor

## Rules Editor

The rules editor is been divided into three sub editors and the rules outline view is also sub divided in five outline views

### Parts of Rules Editor:

1. **Editor:** sub divided in three parts as Embedded editor, Modal Editor and Logical Editor
  - a. **Embedded editor:** To insert embedded quantification keywords in facts/existing rules.
  - b. **Modal Editor:** To insert Modal keywords in facts/existing rules.
  - c. **Logical Editor:** To insert Logical keywords in facts/existing rules.
2. **Outline View:** sub divided in five parts as facts, embedded, prefix, logical and final.
  - a. **Facts tab:** Display all facts from vocabulary file.

- b. **Embedded** tab: Display all rules with their textual representation created using embedded editor.
  - c. **Prefix** tab: Display all rules with their textual representation created using Modal Editor.
  - d. **Logical** tab: Display all rules with their textual representation created using Logical Editor.
  - e. **Final tab**: Display all rules moved to final stage (No need of editing them).
3. **Property view**: shows and let the user edit several properties of the graphical objects which are captions in SBVR meta-model.
4. **Palette**: this contains all the graphical objects which can be dragged and dropped on the editor

#### How to:

1. Open a new/existing Rule File:
  - a. Options for opening a rule file are available in File Menu and at tool bar. In opening rule file, it will ask to select an already opened vocabulary file in dialog box which will be used to fetch available facts.
2. Create any rule, which editor to use:
  - a. According to SBVR methodology, Rules are created by adding keywords to facts. Different types of available keywords are described in the concept map of SBVR visual syntax. So in order to create a rule with multiple keywords of different type, start with fact and add keywords one by one to fact and then to intermediate rule. Figure [5] explains visual syntax methodology for rules creation.
3. Add quantification keyword to fact/Intermediate rule.
  - a. Select embedded editor in main frame.
  - b. Drop fact/intermediate rule from outline view to editor main frame.
  - c. Select the required keyword from palette and drop on the desired verb/term.
  - d. Other keywords can be added by clicking on the connection.
4. Add Modal keyword
  - a. Select Modal Editor in main frame. You can select your required modal editor by changing rule type in properties view.
  - b. Select the required keyword from palette and drop on the desired verb/term.
5. Add logical keyword
  - a. Select keyword from palette and object from outline view.
6. Save rule file
  - a. Use option at Task bar/File menu or ctrl+s

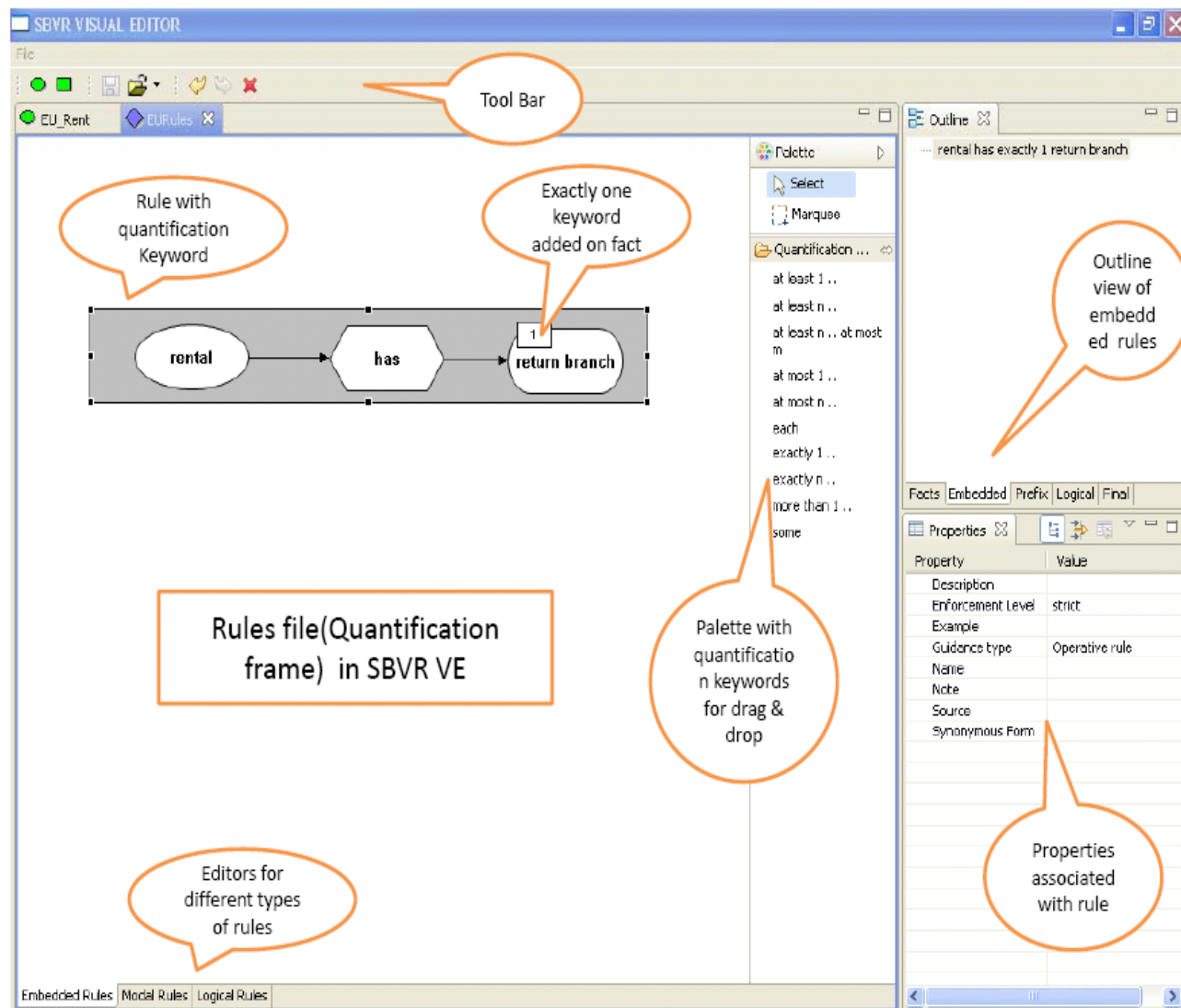


Figure 9: SBVR Rules Editor