SBVR - Semantics of Business Vocabulary and Business Rules

Knut Hinkelmann
What does SBVR do?

SBVR realizes the ‘Business Rules Mantra’:

“Rules are built on Facts. Facts are built on Terms.”

... to describe the business language of the activities of organizations
... in a way that is easily understandable by business people  

(Chapin et al. 2008)
SBVR does not prescribe a specific language for terms, facts and rules.

Terms and facts can be represented, for example, in:
- SBVR Structured English
- Unified Modelling Language (UML)
- Object-Role Modeling (ORM)

The SBVR specification itself uses SBVR Structured English to describe its vocabularies.
Meaning and Representation Vocabulary

- Meaning
- Concept
- Question
- Proposition
- Fact

Noun Concept

- Individual Concept
- Object Type (also: General Concept)
- Concept Type

Fact Type Role (also: Unary Fact Type)

Verb Concept

- Role
- Characteristic (also: Unary Fact Type)
- Binary Fact Type
Noun Concepts: General and Individual

Noun Concepts
- General Concept: Car
- General Concept: Country
- Individual Concept: Switzerland

Things in the real world
- Countries
  - France
  - Germany
  - UK
  - Switzerland
  - Netherlands

Cars
- VIN# 12345
- VIN# 13872
- VIN# 13991
- VIN# 16277
- VIN# 17002
- VIN# 17456
- VIN# 19334
- VIN# 20113

Pre-defined population – represented in vocabulary
General population – represented in database

(Chapin & Hall 2006)
The Conceptual Schema as a Semantic Net

- Two types of concepts:
  - general concept (class)
  - individual concept (instance)

- From this distinction we have at least three kinds of relations (binary fact types)
  - structural relations:
    - Relation between individual and general concepts (also called instance-of)
      - John Smith specializes employee
    - Relation between general concepts (also called is-a or SubClass-Of)
      - employee specializes person
  - non-structural relations
    - arbitrary relations, e.g.
      - John Smith works-for sales department

Unfortunately, SBVR uses the same name for both kinds of structural relations.
Fact Types – Examples

Fact Types

♦ Unary (characteristic): **flight is full**
  • 1 placeholder, filled by ‘flight’

♦ Binary: **aircraft is assigned to flight**
  • two placeholders, filled by ‘aircraft’ and ‘flight’

♦ N-ary: **reassigned flight replaces missed flight after late arrival**
  • three placeholders representing roles, filled by ‘flight’, ‘flight’ and ‘late arrival’

(Chapin et al 2008)
Kinds of Business Rules

Two kinds of business rule:

- **Operational:** govern what the business does
  - “It is obligatory that …”
  - “It is permitted that …” (and its negation, “It is forbidden that …”)

- **Structural:** true by definition
  - “It is necessary that …”
  - “It is possible that …” (and its negation, “It is impossible that …”)

(Chapin & Hall 2006)
Statements of Guidance

■ There is a distinction between
  ♦ rules
  ♦ rule statements

■ There are three kinds of statement to express operative business rules
  ♦ obligation statement
  ♦ prohibition statement
  ♦ restricted permission statement

■ There are also three kinds of statement to express structural business rules
  ♦ necessity statement
  ♦ impossibility statement
  ♦ restricted possibility statement
Defining a Business Rule

■ Start with a fact type, e.g.
  rental has driver

■ Apply an obligation or necessity to it, e.g.
  it is obligatory that rental has driver.

■ Add qualifications, quantifications and conditions, if necessary, e.g.
  it is obligatory that rental has at most 4 drivers
Levels of Enforcement

- Levels of Enforcement are separated from rules

**level of enforcement**

- **Definition:** a position in a graded or ordered scale of values that specifies the severity of action imposed in order to put or keep an **operative business rule** in force
- **Dictionary Basis:** a position on a real or imaginary scale of amount, quantity, extent, or quality [NODE 'level']
- **Dictionary Basis:** compel observance of or compliance with [NODE 'enforcement']

- Only operative rules have levels of enforcements

**operative business rule has level of enforcement**

- SBVR does not prescribe any enforcement levels. It only gives examples (the ones also mentioned in BMM):
  - strict, deferred, pre-authorized, post-justified, override, guidelines
Static Constraints

- **Constraints** are used to define bounds, borders, or limits on fact populations, and may be static or dynamic.

- A *static constraint* imposes a restriction on what fact populations are possible or permitted, for each fact population taken individually.

- Example:
  - \( \forall x \exists y \text{Employee}(x) \land \text{Date}(y) \land \text{born_on}(x,y) \)
  - **Each** Employee has a date of birth
Derivation Rules / Inference Rules

- Derivation rules indicate how the population of a fact type may be derived from the populations of one or more fact types or how a type of an individual may be defined in terms of other types of individuals and fact types.

- Example 1:
  ♦ Person1 is an uncle of Person2 if Person1 is a brother of some Person3 who is a parent of Person2,

  \[ \forall x,y,z \text{ Brother}(x,y) \land \text{Parent}(y,z) \rightarrow \text{Uncle}(x,z) \]

- Example 2:
  ♦ Each person is a employee if the person works for a company

  \[ \forall x,y \text{ Person}(x) \land \text{Company}(y) \land \text{Works_for}(x,y) \rightarrow \text{Employee}(x) \]
Open/Closed World Semantics

Dealing with missing Information

- The closed world assumption (CWA) is the presumption that what is not currently known to be true is false.
  - Under the CWA, if a proposition cannot be proved true, it is false.

- The open world assumption (OWA) states that lack of knowledge does not imply falsity.
  - Under the OWA, if a proposition cannot be proved true and its negation cannot be proved true, the truth of the proposition is unknown
Database Example for Open/Closed World

Suppose we have the following sample database with the employee number and name of each employee, as well as the cars they drive (if any):

<table>
<thead>
<tr>
<th>Employee</th>
<th>Drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>empNr</td>
<td>carRegNr</td>
</tr>
<tr>
<td>1</td>
<td>ABC123</td>
</tr>
<tr>
<td>2</td>
<td>AAA246</td>
</tr>
<tr>
<td>3</td>
<td>DEF001</td>
</tr>
</tbody>
</table>

- Users typically adopt the closed world assumption when interpreting data in databases.
  - If a data is not in the database, it is assume to be false
- Example: Select employee number of each employee who does not drive a car
  ```
  select empNr from Employee where empNr not in (select empNr from Drives).
  ```
- Correctness of the result depends on whether all employees with their car registry are in the database.

Does it include all employees?

Does it include all drivers?
Open/Closed World and Negation

- The open or closed world semantics is important for negation
  - Closed World (CWA): a failure to find a fact implies its negation
    → negation as failure
  - Open World (OWA): lack of knowledge does not imply falsity. A proposition is false only if its negation can be proved.
    → full negation

- Example:
  - If the customer did not pay his goods he is reminded.
Open or closed world?

- A business might have complete knowledge about some parts and incomplete knowledge about other parts.
- Thus, in practice a mixture of open and close world assumption may be applied.
- To cope with this situation, one might, for example,
  - assume open world semantic by default and
  - apply local closure to specific parts (or vice versa).
- Local closure means that for some parts of the overall DB schema the closed world assumption applies.
- Local closure can be asserted explicitly for individual and fact types, e.g.
  - employee is closed (all employees are known)
  - has-name is closed (all names of employees are known).
Constraints and Changing Fact Populations

Static Constraints and Derivation Rules are applied at a single state

♦ If the fact model changes there is a new state, for which the constraints and derivation rules are applied again without regard of previous states.

♦ Example:

Assume that customers get a discount if their shopping exceeds 1'000 Fr. within 12 months. The calculation of the discount changes as soon as a shopping is made such that 1'000 Fr. are reached and may be reduced again if the customer does not buy enough within 12 months.

A dynamic constraint imposes a restriction on transitions between states of fact populations.

♦ Dynamic constraints compare one state to another state.

♦ Example:

A person's marital status may change from single to married, but not from divorced to single.

(The semantics of dynamic constraints is not defined in SBVR 1.0)
Adding and Deleting Facts

- If we allow deletion or changes of knowledge, it may occur that previous inferences become invalid.
  - We might delete a fact
    - because we revise our decision on whether it is true (correcting a failure)
    - or because we decide that a fact is no longer of interest

- With Negation as Failure (Closed World), also adding facts may make previous inferences invalid
  - Example: Every employee has a date of birth
    \[
    \forall x \forall y \text{Employee}(x) \land \text{Date}(y) \land \neg \text{born}_\text{on}(x,y) \Rightarrow \text{forbidden}
    \]
    If for an employee there is no date of birth in the database the constraint is violated (i.e. „forbidden“ is derived). If the date of birth is added, the rule should not not be applicable anymore.

- Making previous derivations invalid is called non-monotonic. To avoid this problem, static constraints are applied to individual state of the fact model.