SBVR Meanings and Representations Metaschemas

(April 2008)

Ruth Raventós

raventos@lsi.upc.edu

Departament Llenguatges i Sistemes Informàtics

Universitat Politècnica de Catalunya

1. Introduction

The *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0* document (Object Management Group 2008) is an *Available Specification*, published by the Object Management Group (OMG), that defines the vocabulary and rules for documenting the semantics of business vocabulary, business facts and business rules. The specification is applicable to the domain of business vocabularies and business rules of all kinds of business activities of all kinds of organizations. It is conceptualized optimally for business people rather than automated rules processing, and is designed to be used for business purposes, independent of information system designs.

SBVR was developed, initially, by the Business Rules Group (Object Management Group 2004), who have been working exclusively in the area since the late 1980s. Key notions of the SBVR approach are presented succinctly by the BRG's Business Rule Manifesto (Business Rule Group 2003). The core idea from the Manifesto is "Rules build on facts, and fact build on concepts as expressed by terms. Terms express business concepts; facts make assertions about these concepts; rules constraint and support these facts". Therefore, a business vocabulary should contain all the specialized terms and definitions of concepts that a given organization or community uses in their course of business communication. Rules in SBVR are always constructed by applying modal operators (necessity, obligation, possibility and permissibility) to fact types. For example, the necessity "It is necessary that each edited book has at least one editor" is based on the fact type "Edited book has editor".

The SBVR specification includes the description of the different types of meaning and their representation. It also includes an XMI schema for the interchange of business vocabularies and business rules among organizations and between software tools. Moreover, the specification defines an English vocabulary, called SBVR Structured English, as one of the possibly many notations that can map to the SBVR representation of the metamodel. SBVR Structured English uses a small number of English structures and common words to elaborate vocabularies and rules.

This report describes two metamodels as instances of MOF: the first one is the subset of the SBVR metamodel necessary for describing conceptual schemas as a combination of concepts and facts as defined in SBVR. The second one describes the subset of the SBVR metamodel concerning representations of the meanings.

The representation of both metaschemas, as instances of MOF, facilitates the translation the translation of SBVR vocabularies to and from UML or any other language, also represented as instance of MOF.

2. SBVR Meanings Metaschema

This section describes the subset of the SBVR metamodel necessary for describing conceptual schemas as a combination of concepts and facts as defined in SBVR. Note that, in order to translate UML to SBVR, only the subset of SBVR that describes meanings (concepts and propositions) is necessary. In this section, the SBVR meanings metamodel is defined, in UML, as an instance of MOF. The rest of this section is structured as follows: Section 2.1 shows the figures that form the abstract syntax of the subset of SBVR considered to translate to UML, and Section 2.2 describes, briefly, each concept included in the abstract syntax and their constraints.

2.1. Abstract Syntax

From Figure 2.1 to Figure 2.11 show the fragment of the SBVR metamodel (Object Management Group 2008) considered for the translation from UML and OCL.

The metaclasses included are those whose instances may be translated to form an UML conceptual schema. Therefore, all the elements concerning to representations and business statements, instead of meanings, have not been included Moreover, elements concerning to meanings that are not described in UML conceptual schemas have neither been included:

- Questions: meanings that are interrogatories;
- URI (Uniform Resource Identifiers Vocabulary);
- Modal logics different than necessities modal formulations;

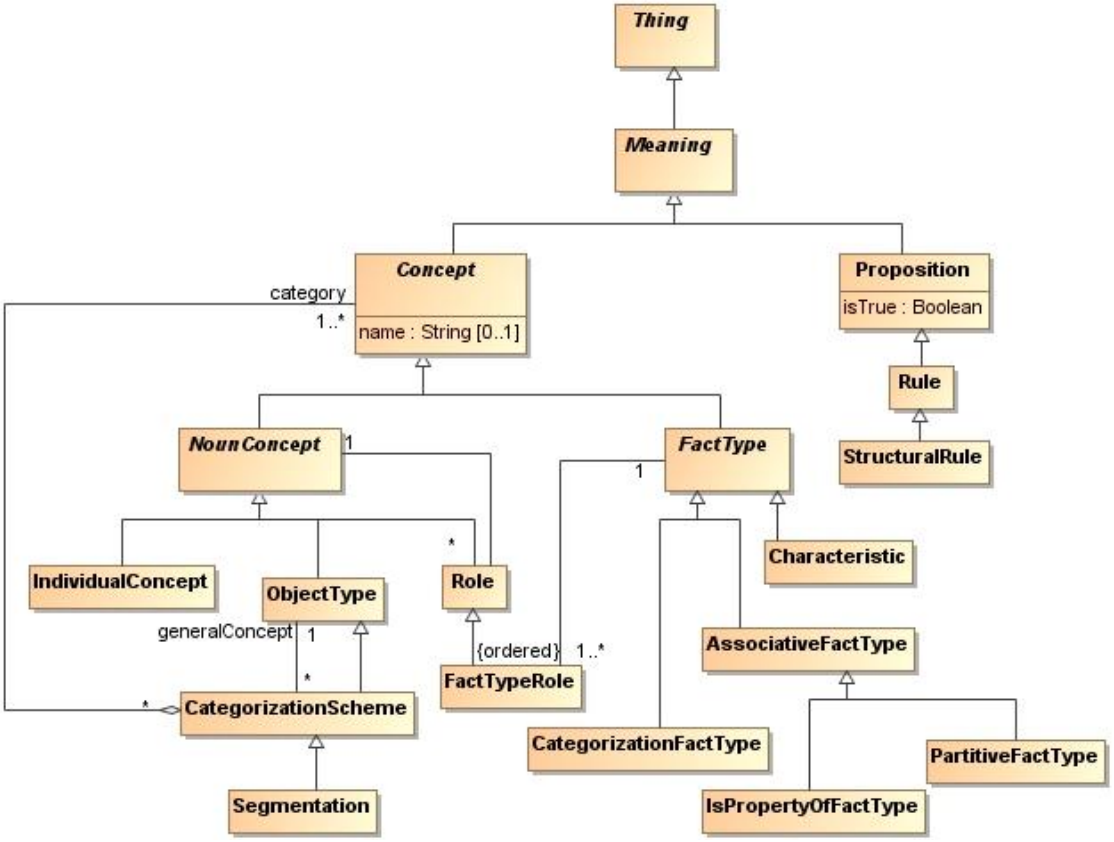

Figure 2.1 Meanings: Concepts and Rules

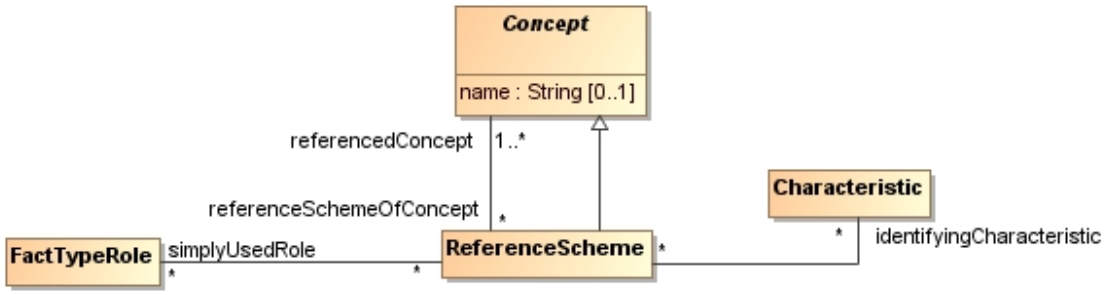

Figure 2.2 Reference Schemes.

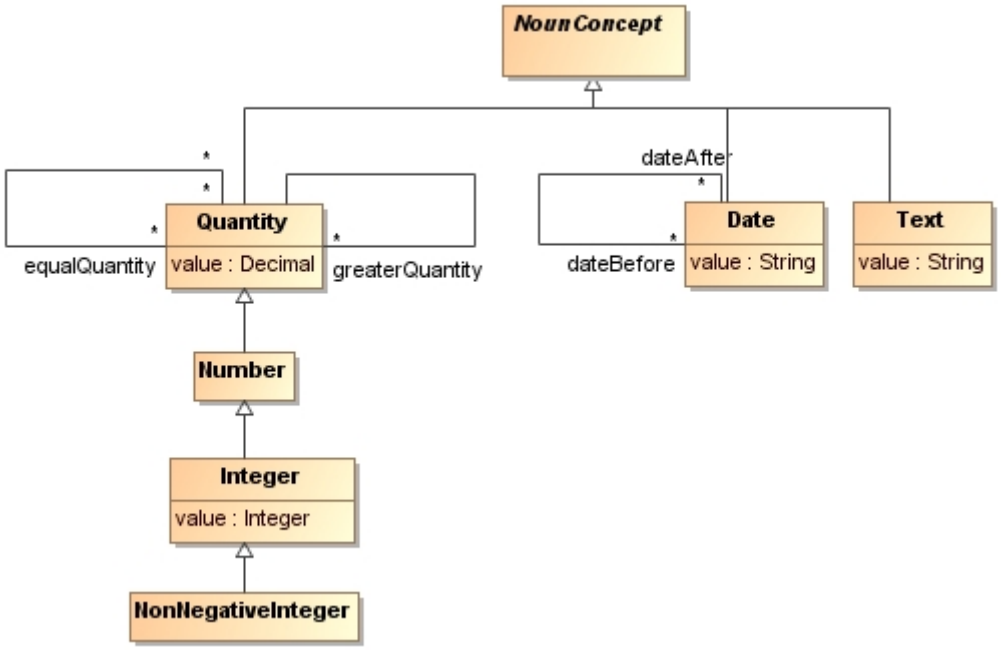

Figure 2.3 Elementary Concepts.

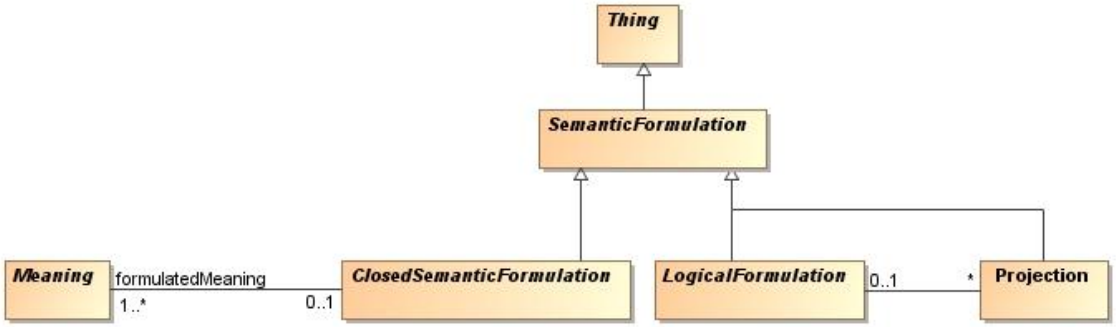

Figure 2.4 Semantic Formulations.

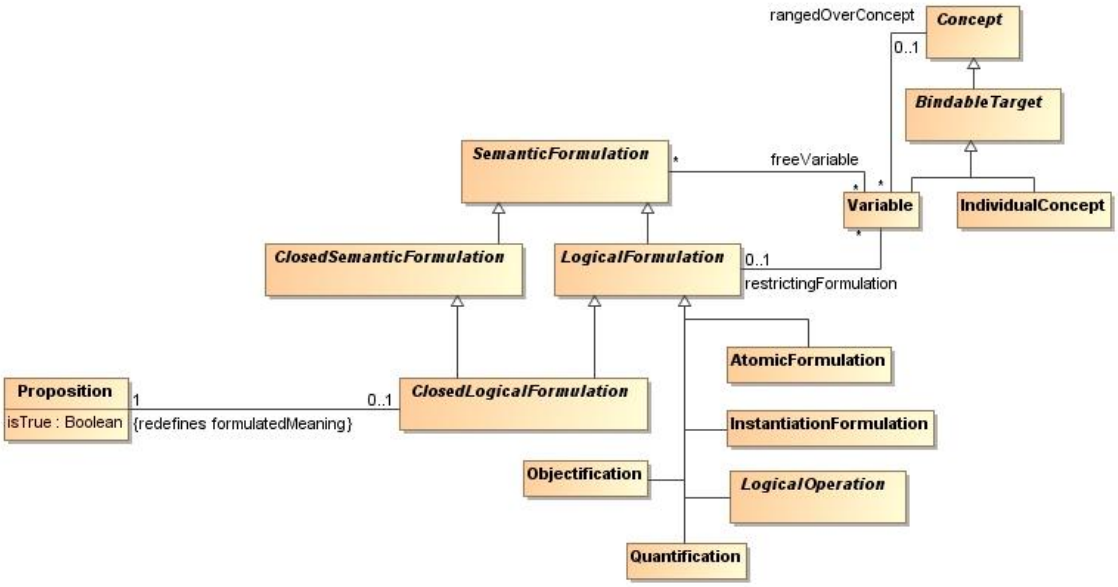

Figure 2.5 Logical Formulations, Variables and Bindings.

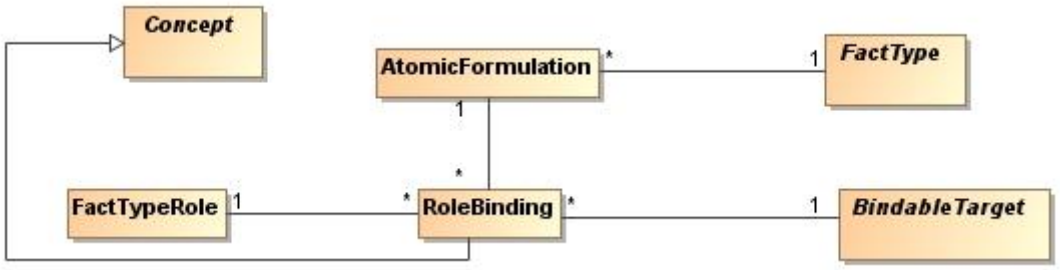

Figure 2.6 Atomic Formulations.

Figure 2.7 Instantiation Formulations.

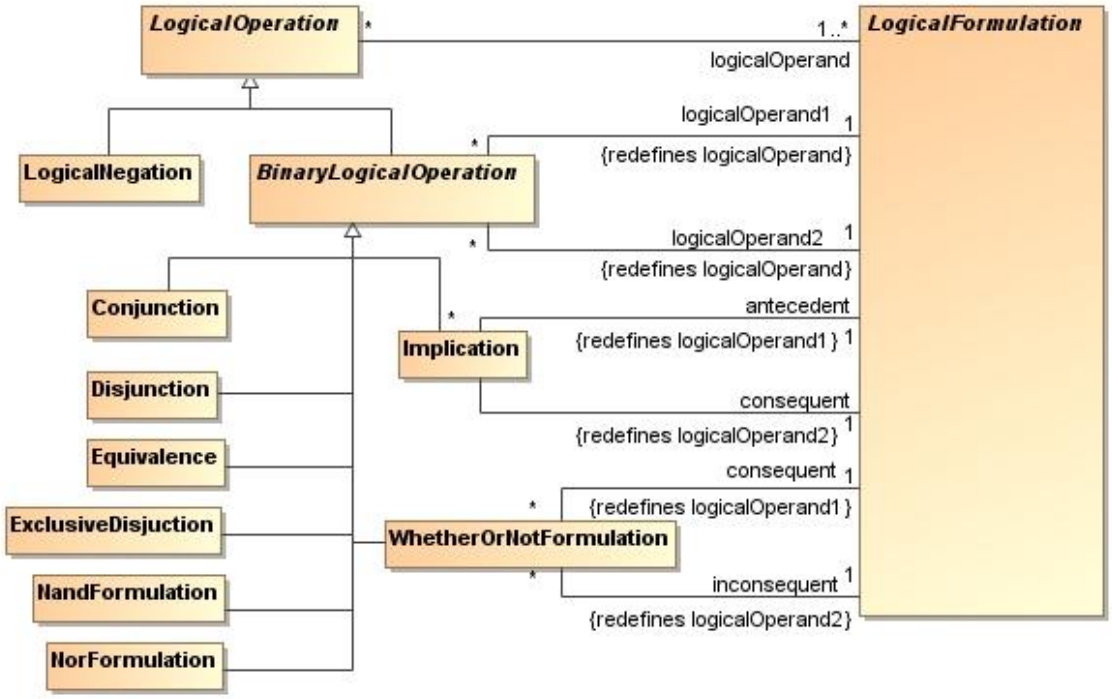

Figure 2.8 Logical Operations.

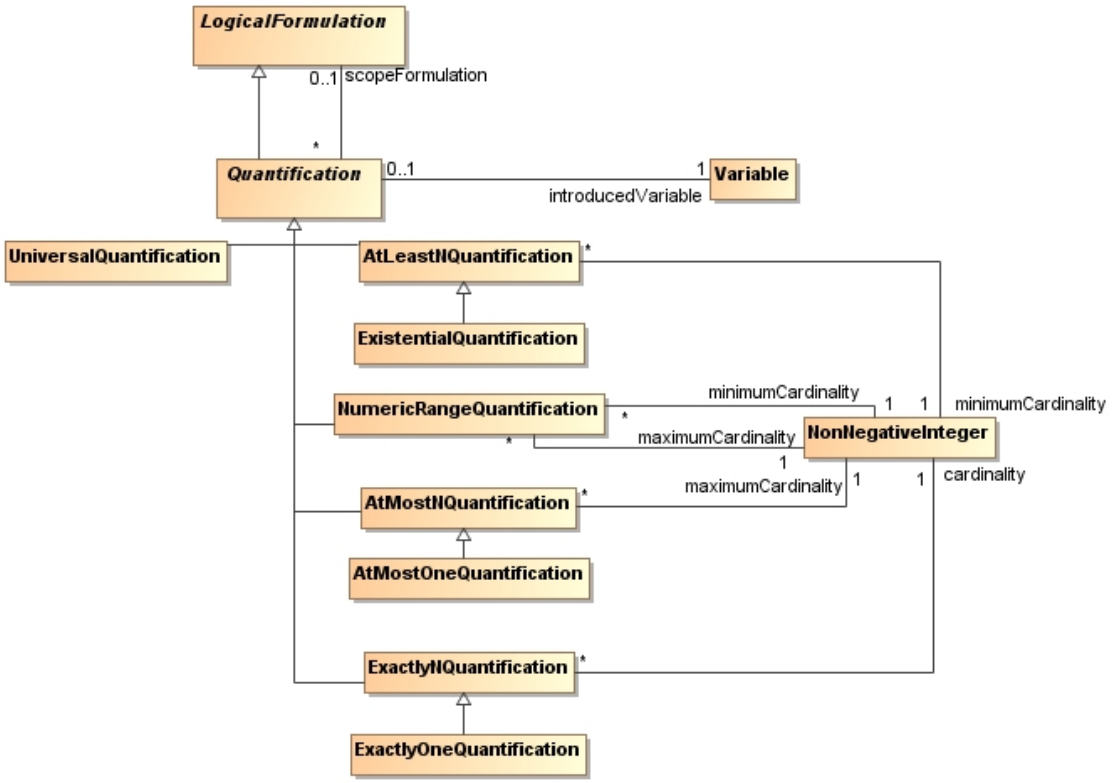

Figure 2.9 Quantifications.

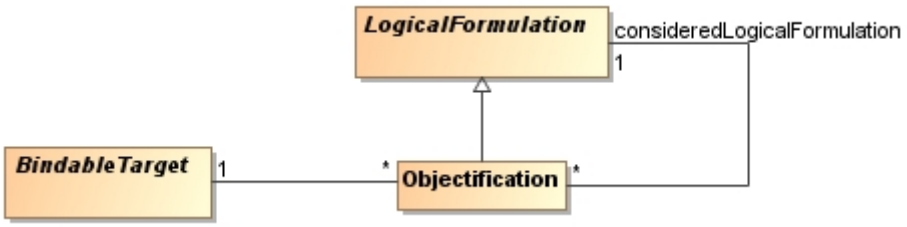

Figure 2.10 Objectifications.

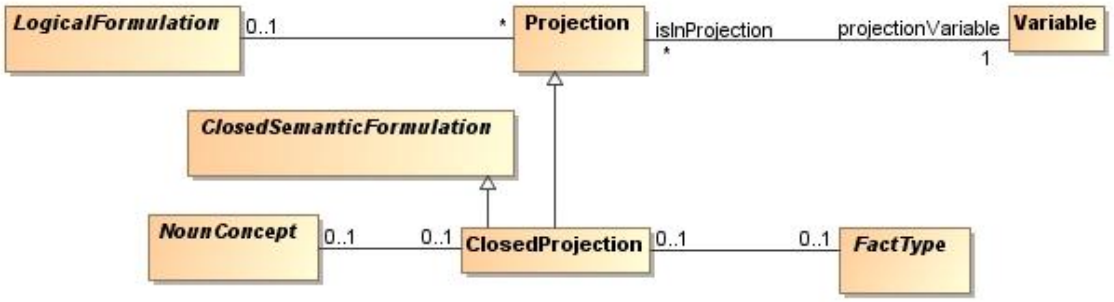

Figure 2.11 Projections.

2.2. Concepts Definition

This section includes a brief description of the concepts illustrated in the figures of above. The definitions are taken from the SBVR specification document (Object Management Group 2008).

Each concept is specified following the same conventions as in the OMG UML Superstructure document. The following clauses may be included for each concept:

- *Heading* of the concept (formal name).
- *Generalizations*: the direct generalization of the concept.
- *Description:* informal description of the purpose, nature and usage of the concept.
- *Attributes:* contains a list of the attributes that are defined for the metaclass.
- *Associations*: describes all the association ends owned by the concept.
- *Constraints:* a numerical list of the constraints that define additional well-formedness rules that applies to this concept.
- *Note:* gives additional clarifications to the concept.

The complete specification in the USE tool of the SBVR metaschema can be found in Appendix D, including the constraints formally expressed in OCL.

2.2.1. AssociativeFactType

Generalizations

- FactType.

Description

Fact type that has more than one role and that has a non-hierarchical subject-oriented connection drawn from experience, based on practical rather than theoretical considerations.

Constraints

[1] An associative fact type must have at least 2 fact type roles.

2.2.2. AtLeastNQuantification

Generalizations

- Quantification.

Description

Quantification that applies the ‘at least n’ quantification operation ($\exists \geq n$), where ‘n’ represents a minimum cardinality.

Association

- minimumCardinality : NonNegativeInteger[1] - References the minimum cardinality or greater that the at-least-n-quantification satisfies.

2.2.3. AtMostNQuantification

Generalizations

- Quantification.

Description

Quantification that applies the ‘at most n’ quantification operation ($\exists^{\leq n}$), where ‘n’ represents a maximum cardinality.

Association

- maximumCardinality : NonNegativeInteger[1] - References the maximum cardinality or less that the at-most-n-quantification satisfies.

2.2.4. AtMostOneQuantification

Generalizations

- AtMostNQuantification.

Description

At-most-n quantification that applies the ‘at most one’ quantification operation ($\exists^{0..1}$).

Constraint

[1] The attribute value of the maximum cardinality has value 1.

2.2.5. AtomicFormulation

Generalizations

- LogicalFormulation.

Descriptions

Logical formulation that is based on a fact type and that has a role binding for each role of the fact type and that formulates the meaning: there is an actuality that involves in each fact type role the thing to which the variable of the corresponding role binding refers. The meaning invoked by an atomic formulation puts each role's referent binding in its respective role.

Associations

- roleBinding : RoleBinding[*] - References the role binding(s) that occur in this atomic formulation.
- factType : FactType[1] – References the fact type that underlies this atomic formulation.

Example

The statement "Antoni Olivé is author of Conceptual Modeling of Information Systems" is formulated by an atomic formulation based on the fact type "person is author of authored publication". The atomic formulation has two role bindings. The first role binding is of the role 'person' of the fact type and binds to the individual concept 'Antoni Olivé'. The second role binding is of the role 'authored publication' of the fact type and binds to the individual concept 'Conceptual Modeling of Information Systems'.

2.2.6. BinaryLogicalOperation

Generalizations

- LogicalOperation

Description

Logical operation that operates on two logical operands.

Association

- logicalOperand1: LogicalFormulation[1] - References the first logical operand that the binary logical operation operates on.
- LogicalOperand2 : LogicalFormulation[1] - References the second logical operand that the binary logical operation operates on.

2.2.7. BindableTarget

Generalizations

- Concept.

Descriptions

Variable or individual concept.

Association

- roleBinding : RoleBinding[*] - References the role binding(s) that is related to this bindable target.
- instantiationFormulation : InstantiationFormulation[*] – References the instantiation formulations that classifies this bindable target.
- objectification : Objectification [*] – References the objectification formulation that binds to this bindable target.

2.2.8. CategorizationScheme

Generalizations

- ObjectType.

Description

Scheme for partitioning things in the extension of a given general concept into the extensions of categories of the general concept.

Association

- generalConcept : ObjectType[1] - References the general concept that is divided into categories by this categorization scheme.
- category : Concept[*] - references the categories included in this categorization scheme.

2.2.9. CategorizationFactType

Generalizations

- FactType.

Description

Fact type that is defined with respect to a given concept and another concept that is a category of that concept such that each instance of the fact type is an actuality that a particular instance of the concept is also an instance of the category.

2.2.10. Characteristic

Generalizations

- FactType.

Description

Fact type that has exactly one role.

Association

- referenceScheme : ReferenceScheme[*] – References the reference schemes that uses this characteristic to identify or partially identify the instances of a given concept.

2.2.11. ClosedLogicalFormulation

Generalizations

- ClosedSemanticFormulation.
- LogicalFormulation .

Description

Logical formulation that is a closed semantic formulation, i.e., that includes no variable without binding.

Association

- Proposition : Proposition[1] - References the proposition that is meant by this closed logical formulation.

2.2.12. ClosedProjection

Generalizations

- Projection.
- ClosedSemanticFormulation.

Description

Projection that is a closed semantic formulation.

Associations

- nounConcept : NounConcept[0..1] - References the concept that defines this closed projection.
- factType : FactType[0..1] – References the fact type that defines this closed projection.

Constraints

[1] Each variable that is in a closed projection maps to exactly one role of each fact type that is defined by the closed projection.

[1] If a closed projection defines a fact type then each role of the fact type is mapped from exactly one variable that is in the closed projection.

[2] A closed projection that defines a noun concept is on at most one variable.

2.2.13. ClosedSemanticFormulation

Generalizations

- SemanticFormulation

Description

A closed semantic formulation is a semantic formulation that is closed, i.e., includes no variable without binding.

Association

- formulatedMeaning : Meaning[1::*] - References the meaning(s) structured by this closed semantic formulation.

2.2.14. Concept

Generalizations

- Meaning

Description

A concept is a unit of knowledge created by a unique combination of characteristics.

Concept is mapped to Formal Logic to *type*. *Type* is defined in SBVR as the named set of possible instances, where for any given state of the (business) domain, exactly one subset of the *type* is the population of the *type* in that state. At any given time, the population of a *type* is the set of instances of that *type* that exists in the (business) domain at that time. It follows that if two *type(s)* are equal, then for each state of the domain they must have the same population.

Attributes

- name : String[0..1] – Name of the concept.

Association

- concept : Concept[*] – References the concepts that are general concepts of this concept.
- categorizationScheme : CategorizationScheme[*] – References the categorization scheme which has this concept as a category.
- instantiationFormulation : InstantiationFormulation[*] – References the instantiation formulation that classifies things to be an instance of this concept.

2.2.15. Conjunction

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of each of its logical operands is true.

2.2.16. Date

Generalization

- NounConcept

Description

A particular day of the month or year.

Attributes

- value: String – The actual value of date.

Associations

- dateBefore : Date[*] – References the dates before this date.
- dateAfter : Date[*] – References the dades after this date.

2.2.17. Disjunction

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of at least one of its logical operands is true.

2.2.18. Equivalence

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of its logical operands is either all true or all false.

2.2.19. ExactlyNQuantification

Generalizations

- Quantification.

Description

Quantification that applies the ‘exactly n’ quantification operation ($\exists^{=n}$), where ‘n’ represents a cardinality.

Association

- cardinality : NonNegativeInteger[1] - References the cardinality that the exactly-n-quantification satisfies.

2.2.20. ExactlyOneQuantification

Generalizations

- ExactlyNQuantification.

Description

Exactly-n-quantification that applies the ‘exactly 1’ quantification operation ($\exists^{=1}$).

Constraint

[1] The attribute value of the cardinality has value 1.

2.2.21. ExclusiveDisjunction

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of one logical operand is true and the meaning of the other logical operand is false.

2.2.22. ExistentialQuantification

Generalizations

- AtLeastNQuantification.

Description

At-least-n-quantification that applies the ‘at least 1’ quantification operation ($\exists^{\geq 1}$).

Constraint

[1] The attribute minimum cardinality has value 1.

2.2.23. FactType

Generalizations

- Concept.

Description

Concept whose instances are all actualities and that is a basis for atomic formulation, having at least one role. An actuality is defined as a state of affairs (event, activity, situation or circumstance) that occurs in the actual world. A FactType corresponds to a *predicate* in predicate logic. But while there is no further distinction between different kinds of predicates in standard logic, SBVR has different classifications of fact types.

Association

- factTypeRole : FactTypeRole[1..*] ordered - References the ordered fact type role(s) that participates in this fact type. The role is an abstraction of a thing playing a part in instances of the fact type.
- atomicFormulation : AtomicFormulation[*] – References the atomic formulations that are based on this fact type.
- closedProjection : ClosedProjection[*] – References the closed projections that define this fact type.

2.2.24. FactTypeRole

Generalizations

- Role.

Description

Role that specifically characterizes its instances by their involvement in an actuality that is an instance of a given fact type.

Association

- factType : FactType[1] - References the fact type in which is this fact type role.
- referenceScheme : ReferenceScheme[*] – References the reference schemes that use this fact type role as identification or partical identification of an instance of a concept.
- roleBinding : RoleBinding[*] – References the role bindings that are bindings of this fact type role, which is of the fact type that underlies an atomic formulation.

2.2.25. Implication

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that operates on an antecedent and a consequent and that formulates that the meaning of the consequent is true if the meaning of the antecedent is true.

Associations

- antecedent: LogicalFormulation[1] - References the logical operand that is the condition considered by this implication.
- consequent : LogicalFormulation[1] - References the logical operand that is the implied or result operand to this implication.

2.2.26. IndividualConcept

Generalizations

- NounConcept, BindableTarget.

Description

An individual concept is a noun concept that corresponds to only one object.

2.2.27. Integer

Generalizations

- Number.

Description

Number that has no fractional part.

Attributes

- value : Integer – value of the integer.

2.2.28. InstantiationFormulation

Generalizations

- LogicalFormulation.

Description

Logical formulation that considers a concept and binds to a bindable target and that formulates the meaning: the thing to which the bindable target refers is an instance of the concept.

Associations

- conceptConsidered : Concept[1] – Refers to the concept considered as instance in this instantiation formulation.
- bindableTarget : BindableTarget[1] – Refers to the thing that is being classified by this intantiation formulation.

2.2.29. IsPropertyOfFactType

Generalizations

- AssociativeFactType.

Description

Associative fact type that is defined with respect to a first given concept and a second given concept so that each instance of the fact type is an actuality that an instance of the first concept constitutes an essential quality of an instance of the second concept.

Constraints

[1] An is-property-of-fact-type has exactly two fact type roles.

2.2.30. LogicalFormulation

Generalizations

- SemanticFormulation.

Description

A logical formulation is a semantic formulation that formulates a proposition, this is, it is the conceptual structure of meaning that formulates a proposition.

Association

- logicalOperation : LogicalOperation[*] - References the logical operation(s) that operate on this logical formulation.
- quantification : Quantification[*] - References the quantification(s) that scopes over this logical formulation.
- objectification : Objectification[*] – References the objectifications that are considered by this logical formulation. The objectification is of the state or event that corresponds to the meaning of the logical formulation.
- projection : Projection[*] - References the projection(s) that are constrained by this logical formulation.
- variable : Variable[*] – References the variables that are restricted by this logical formulation.

2.2.31. LogicalNegation

Generalizations

- LogicalOperation.

Description

Closed logical operation that has exactly one logical operand and that formulates that the meaning of the logical operand is false.

Constraint

[1] The number of logical operands of a logical negation must be one.

2.2.32. LogicalOperation

Generalizations

- LogicalFormulation.

Description

A logical operation is a logical formulation that operates on logical operands..

Association

- logicalOperand : LogicalFormulation[1..*] - References the logical formulation(s) that this logical operation operates on.

2.2.33. Meaning

Generalizations

- Thing.

Description

Meaning is what is meant by a word, sign, statement, or description; what someone intends to express or what someone understands.

Association

- closedSemanticFormulation : ClosedSemanticFormulation[0..1] - References the closed semantic formulation that structures the meaning.

2.2.34. NandFormulation

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of at least one of its logical operands is false.

2.2.35. NorFormulation

Generalizations

- BinaryLogicalOperation.

Description

Binary logical operation that formulates that the meaning of each of its logical operands is false.

2.2.36. NounConcept

Generalizations

- Concept.

Description

A noun concept is defined in SBVR as a concept that is not a fact type.

Associations

- closedProjection : ClosedProjection – References the closed projection(s) that defines this noun concept.

2.2.37. NonNegativeInteger

Generalizations

- Integer.

Description

Integer that is greater than or equal to zero.

2.2.38. Number

Generalizations

- Quantity.

Description

Quantity belonging to an abstract mathematical system and subject to laws of succession, addition, and multiplication.

2.2.39. NumericRangeQuantification

Generalizations

- Quantification.

Description

Quantification that applies the 'NUMERIC RANGE' quantification operation ($\exists_{n..m}$), where '*n*' represents a minimum cardinality and '*m*' represents a maximum cardinality.

Associations

- minimumCardinality : NonNegativeInteger[1] - References the minimum cardinality that the numeric-range-quantification satisfies.
- maximumCardinality : NonNegativeInteger[1] - References the maximum cardinality that the numeric-range-quantification satisfies.

2.2.40. Objectification

Generalizations

- LogicalFormulation.

Description

Logical formulation that involves a bindable target and a considered logical formulation and that formulates the meaning: the thing to which the bindable target refers is a state of affairs that corresponds to the meaning of the considered logical formulation.

Association

- consideredLogicalFormulation : LogicalFormulation[1] - References the logical formulation that considers this objectification.
- bindableTarget : BindableTarget[1] - References the bindable target that involves this objectification.

2.2.41. ObjectType

Generalizations

- NounConcept.

Description

Noun concept that classifies things on the basis of their common properties.

Association

- role : Role[*] – References the roles that ranges over this object type. This is, each characteristic that is incorporated by the object type is incorporated by the role.
- categorizationScheme : CategorizationScheme[*] – References the categorization scheme that partitions things in the extension of this object type.

2.2.42. PartitiveFactType

Generalizations

- AssociativeFactType.

Description

Fact type that has two roles and where each instance *is* an actuality that a given part is in the composition of a given whole.

Constraints

[1] A PartitiveFactType must have exactly 2 roles.

2.2.43. PositiveInteger

Generalizations

- NonNegativeInteger.

Description

Non negative integer that is not zero.

2.2.44. Projection

Generalizations

- SemanticFormulation.

Description

Semantic formulation that operates over one or more variables corresponding to involvements in actualities and that is possibly contrained by a logical formulation and projects one or more of those variables.

Association

- variable : Variable[1] - References the variable that is introduced by this projection such that satisfying referents of the variable are in the result of the projection.
- logicalFormulation : LogicalFormulation[0..1] - References the logical formulation that constrains this Projection.

Constraints

[1] No variable that is in a projection is introduced by a quantification.

2.2.45. Proposition

Generalizations

- Meaning.

Description

Meaning that that is true or false.

Attributes

- isTrue: Boolean – indicates that the meaning of the proposition is true.

Association

- closedLogicalFormulation : closedLogicalFormulation[*] - References the closed logical formulation(s) that means this proposition.

2.2.46. Quantification

Generalizations

- LogicalFormulation.

Description

Logical formulation that applies a logical quantification operation to a variable and that has either the meaning: all referents of the variable satisfy a scope formulation; or the meaning: a bounded number of referents of the variable exists and satisfy a scope formulation, is there is one.

Association

- scopeFormulation : LogicalFormulation[0..1] - References the logical formulation such that each referent of the variable introduced by the quantification satisfies the logical formulation if the meaning formulated by the scope formulation is true with every occurrence of the variable interpreted as referring tto the referent.
- introducedVariable:Variable[1] – References the variable that introduces this quantification.

2.2.47. Quantity

Generalizations

- NounConcept.

Description

The aspect in which a a this is measurable in terms of greater, less or equal.

Association

- equalQuantity : Quantity[*] – References the quantities that are mathematically equivalent to this quantity.
- greaterQuantity : Quantity[*] – References the quantities that are mathematically greater than this quantity.

2.2.48. ReferenceScheme

Generalizations

- Concept.

Description

Chosen way of identifying instances of a given Concept.

Association

- referencedConcept : Concept[1..*] – References the concepts that can be identified using the reference scheme.
- simplyUsedRole : FactTypeRole[*] – References any fact type role that serves as identification or partial identification of an instance of the concept.
- identifyingCharacteristic : Characteristic[*] – References any characteristic that serves as identification or partical identification of an instance of the concept.

2.2.49. Role

Generalizations

- NounConcept.

Description

NounConcept that corresponds to things based on their playing a part, assuming a function or being used in some situation.

Association

- nounConcept : NounConcept [1] - References the noun concept that the role ranges over.

Constraints

[1] Roles range over concepts that are not individual concepts.

2.2.50. RoleBinding

Generalizations

- Concept.

Description

Connection of an atomic formulation to a bindable target to a role of the fact type that underlines the atomic formulation.

Association

- atomicFormulation : AtomicFormulation[1] - References the atomic formulation in which occurs this role binding.
- bindableTarget : BindableTarget[1] – References the bindable target that is referenced by this role binding.

- factTypeRole : FactTypeRole[1] - references the fact type role that is a binding of the atomic formulation of the role binding.

2.2.51. Rule

Generalizations

- Proposition.

Description

Proposition that is a claim of obligation or of necessity

2.2.52. Segmentation

Generalizations

- CategorizationScheme.

Description

Categorization scheme whose contained category(es) are complete (total) and disjoint with respect to the general concept that has the categorization scheme.

2.2.53. SemanticFormulation

Generalizations

- Thing.

Description

Conceptual structure of meaning.

2.2.54. StructuraRule

Generalizations

- Rule.

Description

Rule that is a claim of necessity. It is also called definitional rule.

2.2.55. Text

Generalizations

- NounConcept

Description

Character sequence.

Attributes

- value: String – The actual value of text.

2.2.56. Thing

Description

Anything perceivable or conceivable.

2.2.57. UniversalQuantification

Generalizations

- Quantification.

Description

Quantification that applies the universal quantification operation (∀) scoping over a logical formulation.

Constraint

[1] Each universal quantification scopes over a logical formulation.

2.2.58. Variable

Generalizations

- BindableTarget.

Description

A Variable references to an element of a set, whose referent may vary or is unknown.

Association

- rangedOverConcept : Concept[0..1] - References the concept that ranges over this variable.
- restrictingFormulation : LogicalFormulation[0..1] – References the logical formulation that restricts this variable.
- semanticFormulation : SemanticFormulation[*] – References the semantic formulations that includes this variable without binding.
- quantification : Quantification[0..1] – References the quantification that introduces this variable.
- isInProjection : Projection[*] – References the projection that is in this variable.
- projection : Quantification[0..1] – References the projection that uses this variable as an auxiliary variable.

Note

The set of referents of a variable is limited to instances of the concept, if given. If the variable is restricted by a logical formulation, the set is further limited to those things for which the meaning formulated by that logical formulation is true when the this is substituted for each occurrence of the variable in the formulation. If there is no concept and no restricting logical formulation the set includes every thing.

2.2.59. WhetherOrNotFormulation

Generalizations

- BinaryLogicalOperation" on page 9.

Description

Binary logical operation that has a consequent and an inconsequent and that formulates that the meaning of the consequent is true regardless of the meaning the inconsequent.

Associations

- antecedent: LogicalFormulation[1] - References the logical operand1 of this whether-or-not formulation.

- consequent : LogicalFormulation[1] - References the logical operand2 of this whether-or-not formulation.

3. SBVR Structured English Metaschema

The *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0* document (Object Management Group 2008), as described in Chapter 5, defines the vocabulary and rules for documenting the semantics of business vocabulary, business facts and business rules. The specification is applicable to the domain of business vocabularies and business rules of all kinds of business activities of all kinds of organizations. It is conceptualized optimally for business people rather than automated rules processing, and it is designed to be used for business purposes, independent of information system designs.

The SBVR specification includes the description of the different types of meaning and their representation. The specification defines an English vocabulary, called SBVR Structured English, as one of the possibly many notations that can map to the SBVR representation of the metamodel. SBVR Structured English uses a small number of English structures and common words to elaborate vocabularies and rules.

This section overviews the SBVR Structured English notation and describes the subset of the SBVR metamodel concerning representations of the meanings. Note that, some additional elements have been added to the SBVR Representations metaschema to have a straightforward SBVR Structured English notation. The subset, as in the previous chapter, is defined, in UML, as an instance of MOF. The DBLP example introduced in Section1 is represented as an SBVR Structured English vocabulary.

The rest of this section is structured as follows: Section 3.1 overviews SBVR Structured English as one of the possible notations of the SBVR representations. Section 3.2 shows the figures that form the abstract syntax of the subset of SBVR used to represent meanings in SBVR Structured English and describes, briefly, each concept included in the abstract syntax.

3.1. Overview of SBVR Structured English

SBVR Structured English is a proposed notation to express meanings. This section, reviews the main characteristics of the notation, to describe a vocabulary, which includes necessities of SBVR.

3.1.1. Expressions in SBVR Structured English

Any expression, in SBVR may be written in one of the four font styles:

term	The ‘term’ font is used for a designation for a type, one that is part of a vocabulary being used or defined (e.g., **person**, **paper**).

Name	The ‘name’ font is used for a designation of an individual concept (instances) — a name. Names tend to be proper nouns (e.g., Antoni).
verb	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.
keyword	The ‘keyword’ font is used for linguistic symbols used to construct statements – the words that can be combined with other designations to form statements and definitions (e.g., ‘**each**’ and ‘**it is obligatory that**’)

The SBVR Structured English uses designations and forms of expressions exactly as they are defined in a vocabulary. Plural forms are not used to avoid linguistic difficulties. For example, a formal statement would say "each concept" rather than "all concepts". Both the active form and the passive form of a verb need to be defined in a vocabulary if both are used.

3.1.1.1. *Key words and phrases for logical formulations*

Key words and phrases are shown below for expressing each kind of logical formulation. The letters ‘*n*’ and ‘*m*’ represent use of a literal whole number. The letters ‘*p*’ and ‘*q*’ represent expressions of propositions.

Quantification

each	**universal quantification**
some	**existential quantification**
at least one	**existential quantification**
at least *n*	**at-least-n quantification**
at most one	**at-most-one quantification**
at most *n*	**at-most-n quantification**
exactly one	**exactly-one quantification**
exactly *n*	**exactly-n quantification**
at least *n* **and at most** *m*	**numeric range quantification**
more than one	**at-least-n quantification** with *n* = 2

Logical Operations

it is not the case that *p*	**logical negation**
p **and** *q*	**conjunction**
p **or** *q*	**disjunction**
p **or** *q* **but not both**	**exclusive disjunction**
if *p* **then** *q*	**implication**
q **if** *p*	**implication**
p **if and only if** *q*	**equivalence**
not both *p* **and** *q*	**nand formulation**
neither *p* **nor** *q*	**nor formulation**
p **whether or not** *q*	**whether-or-not formulation**

Where a subject is repeated when using '**and**' or '**or**', the repeated subject can be elided.

3.1.1.2. *Modal Operations*

A possible style of SBVR Structured English for modal operations is the Prefix Style that introduces rules by prefixing a statement with keywords that convey a modality. An structural rule uses the keyword: **It is necessary that**

3.1.1.3. *Other Keywords*

the	Used with a designation to make a pronominal reference to a previous use of the same designation. This is formally a binding to a variable of a quantification.
a, **an**	Universal or existential quantification, depending on context based on English rules.
another	(Used with a term that has been previously used in the same statement) existential quantification plus a condition that the referent thing is not the same thing as the referent of the previous use of the term.
a given	Universal quantification pushed outside of a demonstrative expression where 'a given' is used such that it represents one thing at a time – this is used to avoid ambiguity where the 'a' by itself could otherwise be interpreted as an existential quantification.

3.1.2. Describing a Vocabulary

In SBVR Structured English, a vocabulary is described in a document section having glossary-like entries for concepts having representations in the vocabulary.

3.1.2.1. *Vocabulary Entries*

Each entry is for a single concept, which is called the entry concept. It starts with a representation of the concept, either a designation or a form of expression.

Any of several kinds of captioned details can be listed under the representation. A skeleton of a vocabulary entry is shown below followed by an explanation of the use of each caption. Only those entries considered for the mapping between UML and SBVR are showed.

<primary representation>
Definition:
General Concept:
Concept type:
Necessity:
Reference Scheme:

Primary Representation: Designation or Form of Expression

The designation or form of expression, called the ‘primary representation’ with respect to each entry, can be for any concept type. The primary representation for a fact type is a form of expression. Three examples are given below:

person

person *has* name

Catalunya

Definition

A definition is shown as an expression that can be logically substituted for the primary representation. A definition is fully formal if all of it is styled as described above.

General concept

The 'General Concept' caption can be used to indicate a concept that generalizes the entry concept.

Concept Type

The 'Concept Type' caption is used to specify a type of the entry concept. This is typically not used if the concept has no particular type other than what is obvious from the primary representation. A name is implicitly for an individual concept. Any term is implicitly for a noun concept. A form of expression is implicitly for a fact type.

Necessity

A ‘Necessity’ caption is used to state something that is necessarily true. A necessity is an element of guidance expressed as a structural business rule statement. A guidance statement can be expressed formally or informally. A statement that is formal uses only formally styled text — all necessary vocabulary is available (by definition or adoption) so that no external concepts are required. Such a statement can be represented as a logical formulation. For example:

It is necessary that each authored publication *has* at least one author.

The above example includes three key words or phrases ('it is necessary that', 'each' and 'at least one'), two designations for types and one for a fact type (from a form of expression).

The key phrase “it is necessary that” can be omitted from a statement of a structural rule captioned “Necessity” because it is implied in the caption.

Reference Scheme

The 'Reference Scheme' caption is used to state how things denonted by a term can be distinguished from each other based on one or more facts about the things. A reference scheme is expressed by referring to at least one role of a binary fact type.

3.2. SBVR Representations

Figure 3.1 shows the fragment of the SBVR metamodel (Object Management Group 2008) that describes the representations in SBVR. Note that, in order to have a straightforward notation in

SBVR Structured English the *StructuredEnglishText* metaclass has been added; it has two attributes *value* that constrain the expression of the representation and the attribute *font* which represents any of the four font styles used in SBVR Structured English. A representation, in SBVR Structured English, is composed by a set of ordered instances of *StructuredEnglishText*. Moreover, there are three additional metaclasses: *GeneralConceptCaption*, *ConceptTypeCaption* and *ReferenceSchemeCaption*. The first one represents the designation of a general concept, the second one represents the type of concept and the third one represents the reference scheme of a concept.

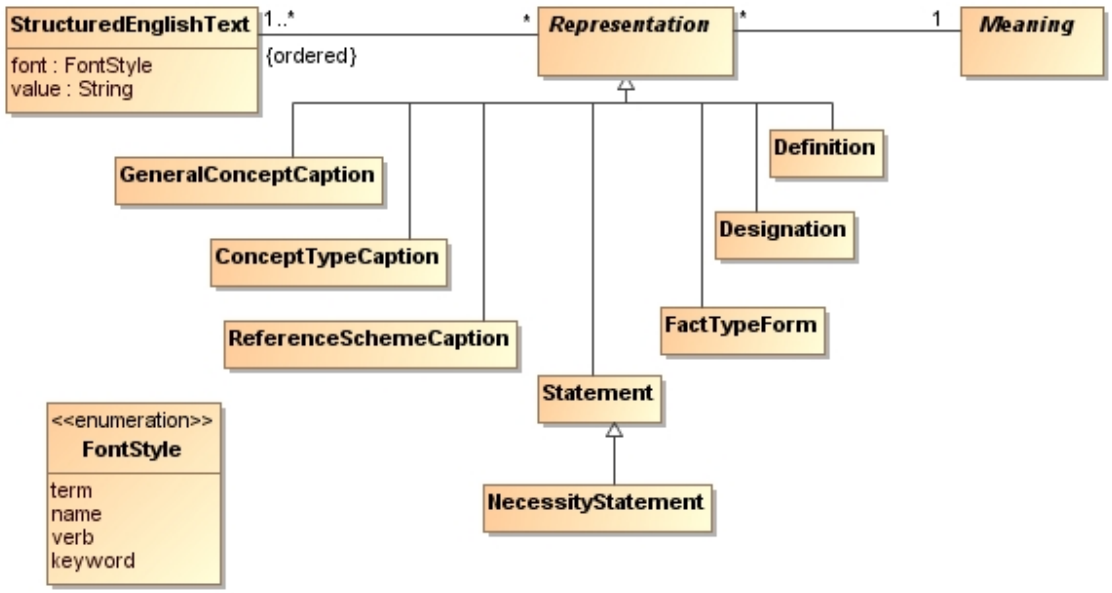

Figure 3.1 SBVR Representations.

In the following, there is a brief description of the concepts illustrated in the Figure 3.1.

The following clauses may be included for each concept:

- *Heading* of the concept (formal name).
- *Generalizations*: the direct generalization of the concept.
- *Description:* informal description of the purpose, nature and usage of the concept.
- *Attributes:* contains a list of the attributes that are defined for the metaclass.
- *Associations*: describes all the association ends owned by the concept.
- *Note:* gives additional clarifications to the concept.

3.2.1. StructuredEnglishText

Description

A text represented in a SBVR Structured English font style (see section 3.2).

Attributes

- value : String – The value of this text.

- font : FontStyle – The font style used in the representation of the value.

Association

- representation : Representation [1] – References the representation of meaning that uses this expression.

3.2.2. Representation

Description

Actuality that a given expression represents a given meaning.

Associations

- structuredEnglishText : StructuredEnglishText [1..*] – References the ordered Structured English expressions used to represent a meaning.
- meaning : Meaning[1] – References the meaning that express this representation.

Note

In the specification of SBVR, a meaning associated to a representation relates the representation of such a meaning. Here, representations are only related to noun concepts or fact types, and the interpretation is the following: a representation is associated to a concept or a fact type if the representation is part of the vocabulary entry that describes the concept or the fact type.

3.2.3. Definition

Generalizations

- Representation.

Description

Representation of a concept by a descriptive statement (expression) which serves to differentiation it from related concepts.

Associations

- concept : Concept [1] – References the concept that defines this definition.

3.2.4. Designation

Generalizations

- Representation.

Description

Representation of a concept by a sign which denotes it.

3.2.5. FactTypeForm

Generalizations

- Representation.

Description

Representation of a fact type by a pattern or template of expressions based on the fact type.

3.2.6. GeneralConceptCaption

Generalizations

- Representation.

Description

Representation to indicate a concept that generalizes another concept.

Note

GeneralConceptCaption, is not included in SBVR Representations metamodel, but in SBVR Structured English. It has been included to have an straightforward notation from the metamodel to SBVR Structured English.

3.2.7. ConceptTypeCaption

Generalizations

- Representation.

Description

Representation to indicate the type of the concept, i.e., role, is-property-of fact type and so on.

Note

ConceptTypeCaption, is not included in SBVR Representations metamodel, but in SBVR Structured English. It has been included to have an straightforward notation from the metamodel to SBVR Structured English.

3.2.8. ReferenceSchemeCaption

Generalizations

- Representation.

Description

Representation to state how things denoted by the term can be distinguished from each other based on one or more facts about things.

Note

ReferenceSchemeCaption, is not included in SBVR Representations metamodel, but in SBVR Structured English. It has been included to have an straightforward notation from the metamodel to SBVR Structured English.

3.2.9. Statement

Generalizations

- Representation.

Description

Representation of a proposition by an expression of the proposition.

3.2.10. NecessityStatement

Generalizations

- Statement.

Description

Statement that express a structural rule and that is expressed positively in terms of necessity.

4. DBLP Example

This section describes an example of schema, represented in UML, SBVR and SBVR Structured English. The example is based on the case study developed by Planas and Olivé (Planas, Olivé 2006) with two additional association classes (IsEditorOf and IsAuthorOf) and an additional attribute which type is an enumeration (Gender).

The DBLP Case Study is a document that contains parts of the conceptual schema of the DBLP systems, written in UML. DBLP, a computer science bibliography website (http://www.informatik.uni-trier.de/~ley/db/), was originally a database and logic programming bibliography site, homed at Universität Trier, in Germany. The DBLP server provides bibliographic information on major computer science journals and proceedings. Initially the server was focused on DataBase systems and Logic Programming (DBLP), now it is gradually being expanded toward other fields of computer science. Nowadays, it's suggested that DBLP stands for "Digital Bibliography & Library Project". The server, mirrored at five sites across the Internet, indexes more than 985000 articles and contains several thousand links to home pages of computer scientists (January 2008).

The structural schema presented in the case study deals with persons (authors and editors) and their publications, which may be edited books or authored publications such as authored books, book chapters and journal papers. Book chapters and journal papers may or may not be conference papers.

Figure 4.1 shows the structural schema of DBLP.

The following constraints have been included in the example:

[1] Person: name

```
context Person inv nameIsKey:
Person.allInstances() -> isUnique(name)
```

[2] Book: isbn

```
context Book inv isbnIsKey:
Book.allInstances() -> isUnique(isbn)
```

[3] BookSeries: id

```
context BookSeries inv idIsKey:
BookSeries.allInstances() -> isUnique(id)
```

[4] Journal: issn

```
context Journal inv issnIsKey:
Journal.allInstances() -> isUnique(issn)
```

[5] Journal: title

```
context Journal inv titleIsKey:
Journal.allInstances() -> isUnique(title)
```

[6] ConferenceSeries: name

```
context ConferenceSeries inv nameIsKey:
ConferenceSeries.allInstances() -> isUnique(name)
```

[7] ConferenceEdition: title

```
context ConferenceEdition inv titleIsKey:
ConferenceEdition.allInstances() -> isUnique(title)
```

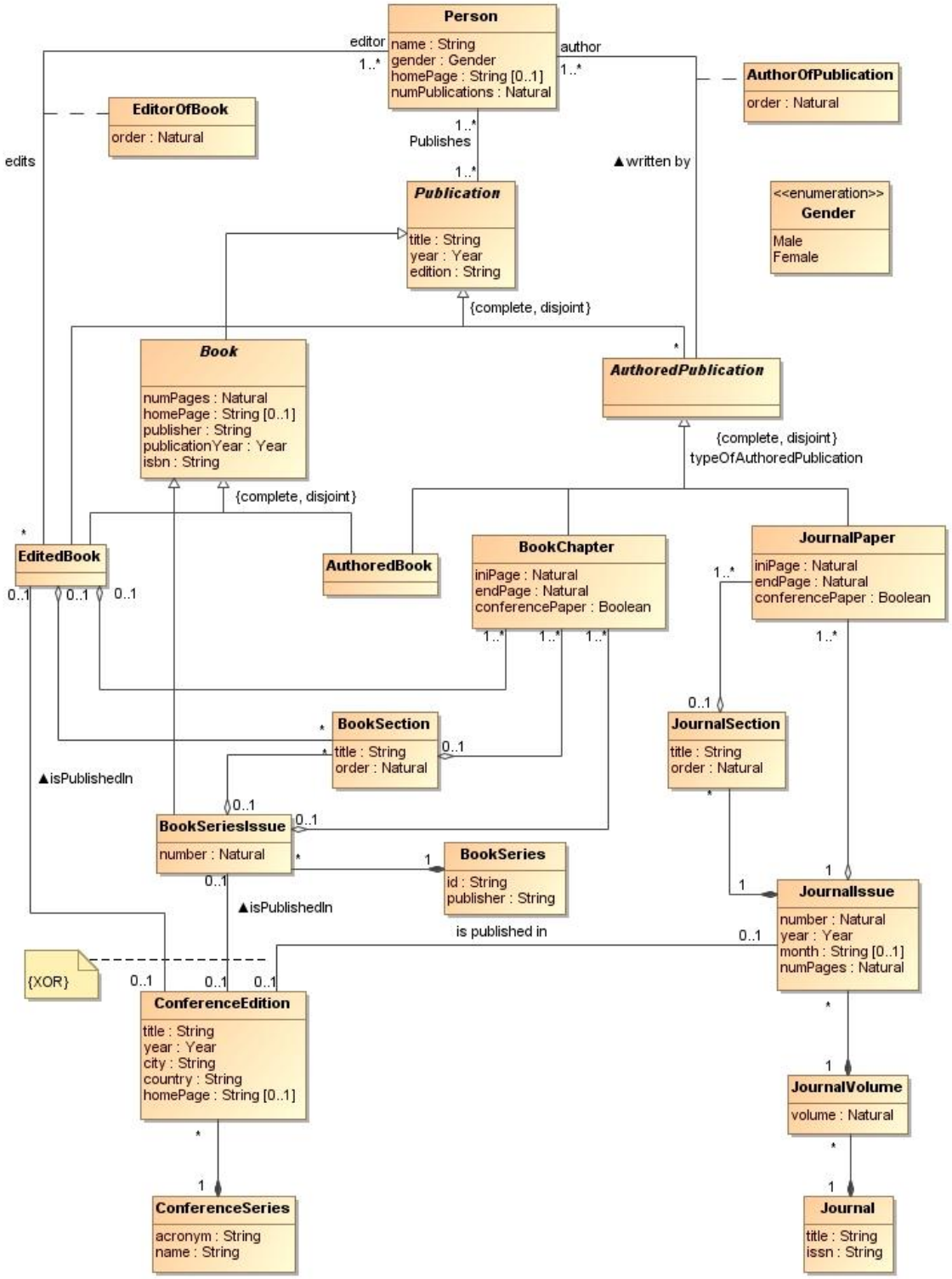

Figure 4.1 Structural schema of DBLP

4.1. DBLP Schema: Example of instance of the SBVR Metaschema

This section describes two small fragments of the DBLP schema as an instance of the SBVR metaschema proposed in Section 2.

Given the following small portion of vocabulary expressed in SBVR Structured English:

authoredBook
GeneralConcept: **book**

editedBook
GeneralConcept: **book**

book
Definition: **editedBook** or **authoredBook**

Figure 4.2 shows the UML instance diagram corresponding to such portion.

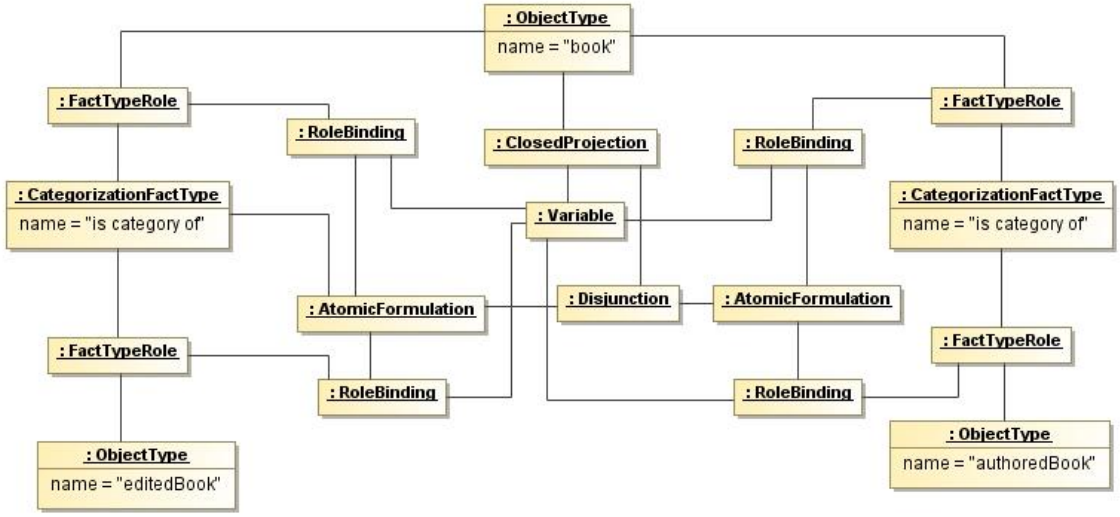

Figure 4.2 Fragment of UML Instance Diagram of SBVR Metaschema (1).

Given the following small portion of vocabulary expressed in SBVR Structured English:

numPages
Concept type: **Natural**
Concept type: **role**

book ***has*** **numPages**
Concept type: **is-property-of fact type**
Necessity: **each book *has* exactly one numPages**.

Figure 4.3 shows the UML instance diagram corresponding to such portion.

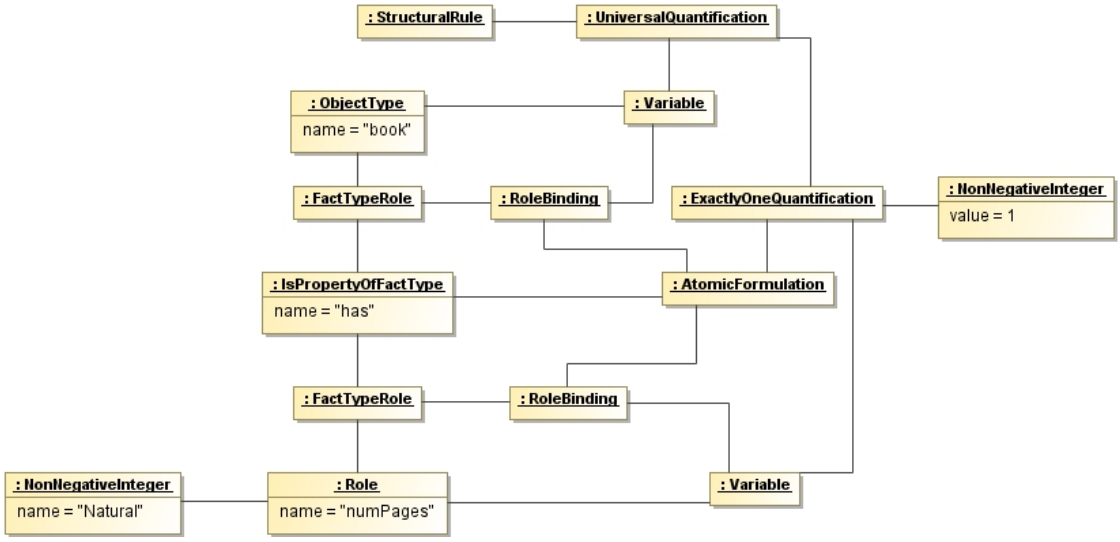

Figure 4.3 Fragment of UML Instance Diagram of SBVR Metaschema (4).

4.2. DBLP Vocabulary in SBVR Structured English Notation

The result of the querying the instances of the DBLP example as instances of the SBVR Representations schema, and after reformatting the font style is the following:

Female

Male

Gender
Definition: Female or Male

Natural

String

Year

acronym
Concept type: String
Concept type: role

authored book
General concept: authored publication
General concept: book

author of publication
Definition: each author of publication that *is of* an author and that is of an authored publication *is* an actuality that the author *is author of* the authored publication

authored publication
Definition: authored book or book chapter or journal paper

General concept: publication
Necessity: each book *is* a authored book or *is* a book chapter or *is* a journal paper
Necessity: each authored publication that *is* a authored book neither *is* a book chapter nor *is* a journal paper
Necessity: each authored publication that *is* a book chapter neither *is* a authored book nor *is* a journal paper

book
Definition: edited book or authored book
General concept: publication
Necessity: each book *is* a edited book or *is* a authored book but not both
Reference scheme: isbn

book chapter
General concept: authored publication

book section

book series
Reference scheme: id

book series issue
General concept: book

city
Concept type: String
Concept type: role

conference edition
Necessity: each conference edition that *is published in* a book series issue neither *is published in* a edited book nor *is published in* a journal issue
Necessity: each conference edition that *is published in* a edited book neither *is published in* a book series issue nor *is published in* a journal issue
Necessity: each conference edition that *is published in* a journal issue neither *is published in* a edited book nor *is published in* a book series issue
Reference scheme: title

conference series
Reference scheme: name

country
Concept type: String
Concept type: role

edited book
General concept: book
General concept: publication

edition
Concept type: String
Concept type: role

editor of book
Definition: each editor of book that *is of* an editor and that is of a book *is* an actuality

that the editor *edits* the book

end page
Concept type: Natural
Concept type: role

gender
Concept type: Gender
Concept type: role

home page
Concept type: String
Concept type: role

id
Concept type: String
Concept type: role

ini page
Concept type: Natural
Concept type: role

isbn
Concept type: String
Concept type: role

issn
Concept type: String
Concept type: role

journal
Reference scheme: issn
Reference scheme: title

journal issue

journal paper
General concept: authored publication

journal section

journal volume

month
Concept type: String
Concept type: role

name
Concept type: String
Concept type: role

num pages
Concept type: Natural
Concept type: role

num publications
Concept type: Natural

Concept type: role

number
Concept type: Natural
Concept type: role

order
Concept type: Natural
Concept type: role

person
Reference scheme: name

publication
Definition: edited book or authored publication
Necessity: each publication *is* a edited book or *is* a authored publication but not both

publication year
Concept type: Year
Concept type: role

publisher
Concept type: String
Concept type: role

title
Concept type: String
Concept type: role

type of authored publication
Definition: categorization scheme that *is for* authored publication
Necessity: type of authored publication *contains* the categories journal paper, authored book and book chapter

volume
Concept type: Natural
Concept type: role

year
Concept type: Year
Concept type: role

author *is author of* authored publication
Concept type: associative fact type
Necessity: each authored publication *has* at least one author

author of publication *has* order
Concept type: is-property-of fact type
Necessity: each is author of *has* exactly one order

book *has* home page
Concept type: is-property-of fact type
Necessity: each book *has* at most one home page

book *has* isbn
Concept type: is-property-of fact type
Necessity: each book *has* exactly one isbn

book ***has*** **num pages**
Concept type: **is-property-of fact type**
Necessity: **each book *has* exactly one num pages**

book ***has*** **publication year**
Concept type: **is-property-of fact type**
Necessity: **each book *has* exactly one publication year**

book ***has*** **publisher**
Concept type: **is-property-of fact type**
Necessity: **each book *has* exactly one publisher**

book chapter ***being conference paper***
Concept type: **characteristic**

book chapter ***has*** **end page**
Concept type: **is-property-of fact type**
Necessity: **each book chapter *has* exactly one end page**

book chapter ***has*** **ini page**
Concept type: **is-property-of fact type**
Necessity: **each book chapter *has* exactly one ini page**

book chapter ***is part of*** **book section**
Concept type: **associative fact type**
Necessity: **each book chapter *is part of* at most one book section**
Necessity: **each book section *has* at least one book chapter**

book chapter ***is part of*** **book series issue**
Concept type: **associative fact type**
Necessity: **each book chapter *is part of* at most one book series issue**
Necessity: **each book series issue *has* at least one book chapter**

book chapter ***is part of*** **edited book**
Concept type: **associative fact type**
Necessity: **each book chapter *is part of* at most one edited book**
Necessity: **each edited book *has* at least one book chapter**

book section ***has*** **order**
Concept type: **is-property-of fact type**
Necessity: **each book section *has* exactly one order**

book section ***has*** **title**
Concept type: **is-property-of fact type**
Necessity: **each book section *has* exactly one title**

book section ***is part of*** **edited book**
Concept type: **associative fact type**
Necessity: **each book section *is part of* at most one edited book**

book series ***has*** **id**
Concept type: **is-property-of fact type**
Necessity: **each book series *has* exactly one id**

book series ***has*** **publisher**
Concept type: **is-property-of fact type**
Necessity: **each book series *has* exactly one publisher**

book series ***includes*** **book series issue**
Concept type: **partitive fact type**
Necessity: **each book series issue *has* exactly one book series**

book series issue ***has*** **number**
Concept type: **is-property-of fact type**
Necessity: **each book series issue *has* exactly one number**

conference edition ***has*** **city**
Concept type: **is-property-of fact type**
Necessity: **each conference edition *has* exactly one city**

conference edition ***has*** **country**
Concept type: **is-property-of fact type**
Necessity: **each conference edition *has* exactly one country**

conference edition ***has*** **home page**
Concept type: **is-property-of fact type**
Necessity: **each conference edition *has* at most one home page**

conference edition ***has*** **title**
Concept type: **is-property-of fact type**
Necessity: **each conference edition *has* exactly one title**

conference edition ***has*** **year**
Concept type: **is-property-of fact type**
Necessity: **each conference edition *has* exactly one year**

conference edition ***is published in*** **book series issue**
Concept type: **associative fact type**
Necessity: **each book series issue *has* at most one conference edition**
Necessity: **each conference edition *is published in* at most one book series issue**

conference edition ***is published in*** **edited book**
Concept type: **associative fact type**
Necessity: **each conference edition *is published in* at most one edited book**
Necessity: **each edited book *has* at most one conference edition**

conference edition ***is published in*** **journal issue**
Concept type: **associative fact type**
Necessity: **each conference edition *is published in* at most one at least one journal issue**
Necessity: **each journal issue *has* at most one conference edition**

conference series ***has*** **acronym**
Concept type: **is-property-of fact type**
Necessity: **each conference series *has* exactly one acronym**

conference series ***has*** **name**
Concept type: **is-property-of fact type**
Necessity: **each conference series *has* exactly one name**

conference series ***includes*** **conference edition**
Concept type: **partitive fact type**
Necessity: **each conference edition *has* exactly one conference series**

editor ***is editor of*** **edited book**

Concept type: associative fact type
Necessity: each edited book *has* at least one editor

editor of book *has* order
Concept type: is-property-of fact type
Necessity: each editor of book *has* exactly one order

journal *has* issn
Concept type: is-property-of fact type
Necessity: each journal *has* exactly one issn

journal *has* title
Concept type: is-property-of fact type
Necessity: each journal *has* exactly one title

journal *includes* journal volume
Concept type: partitive fact type
Necessity: each journal volume *has* exactly one journal

journal issue *has* month
Concept type: is-property-of fact type
Necessity: each journal issue *has* at most one month

journal issue *has* num pages
Concept type: is-property-of fact type
Necessity: each journal issue *has* exactly one num pages

journal issue *has* number
Concept type: is-property-of fact type
Necessity: each journal issue *has* exactly one number

journal issue *has* year
Concept type: is-property-of fact type
Necessity: each journal issue *has* exactly one year

journal issue *includes* journal section
Concept type: partitive fact type
Necessity: each journal section *has* exactly one journal issue

journal paper *being conference paper*
Concept type: characteristic

journal paper *has* end page
Concept type: is-property-of fact type
Necessity: each journal paper *has* exactly one end page

journal paper *has* ini page
Concept type: is-property-of fact type
Necessity: each journal paper *has* exactly one ini page

journal paper *is part of* journal issue
Concept type: associative fact type
Necessity: each journal issue *has* at least one journal paper
Necessity: each journal paper *is part of* exactly one journal issue

journal paper *is part of* journal section
Concept type: associative fact type
Necessity: each journal paper *is part of* at most one journal section

Necessity: each journal section *has* at least one journal paper

journal section *has* order
Concept type: is-property-of fact type
Necessity: each journal section *has* exactly one order

journal section *has* title
Concept type: is-property-of fact type
Necessity: each journal section *has* exactly one title

journal volume *has* volume
Concept type: is-property-of fact type
Necessity: each journal volume *has* exactly one volume

journal volume *includes* journal issue
Concept type: partitive fact type
Necessity: each journal issue *has* exactly one journal volume

person *has* gender
Concept type: is-property-of fact type
Necessity: each person *has* exactly one gender

person *has* home page
Concept type: is-property-of fact type
Necessity: each person *has* at most one home page

person *has* name
Concept type: is-property-of fact type
Necessity: each person *has* exactly one name

person *has* num publications
Concept type: is-property-of fact type
Necessity: each person *has* exactly one num publications

person *publishes* publication
Concept type: associative fact type
Necessity: each person *publishes* at least one publication
Necessity: each publication *has* at least one person

publication *has* edition
Concept type: is-property-of fact type
Necessity: each publication *has* exactly one edition

publication *has* title
Concept type: is-property-of fact type
Necessity: each publication *has* exactly one title

publication *has* year
Concept type: is-property-of fact type
Necessity: each publication *has* exactly one year

References

Business Rule Group 2003, *The Business Rule Manifesto. v.2.0*, Ronald G. Ross, http://www.businessrulesgroup.org/brmanifesto.htm.

Object Management Group 2008, *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. OMG Available specification. (formal/2008-01-02)*.

Object Management Group 2004, *Business Rule Team, Business Semantics of Business Rules (BSBR)*.

Planas, E. & Olivé, A. 2006, *The DBLP Case Study*. http://guifre.lsi.upc.edu/DBLP.pdf.