RC25363 REV (WAT1303-040) March 17, 2014
Computer Science

IBM Research Report

Mapping SBVR to OWL2

Elisa Kendall
Thematix Partners

Mark H. Linehan
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

Mapping SBVR to OWL2

March 28, 2014

Elisa Kendall — ekendall@thematix.com — Thematix Partners

Mark H. Linehan — mhl@mlinehan.us — IBM Research

This paper proposes a mapping of SBVR[1] vocabularies to a combination of OWL2[2] elements and annotations. The goal of this work is to define a reversible mapping: an SBVR vocabulary can be mapped to OWL2 and back again without loss of semantic information. That is, given a transformation chain *V1* --> *O1* --> *V2*, where *V1* and *V2* are SBVR vocabularies, and *O1* is an OWL2 ontology, the semantic meaning of *V1* and *V2* are equivalent. This does not mean that *V1* and *V2* are textually identical because their format (e.g. Structured English versus graphical), syntactic structure, and lexical details (e.g. white space) may differ. It does imply that in a transformation chain *V1* --> *O1* --> *V2* --> *O2*, the two SBVR vocabularies have the same meaning, and the two OWL2 ontologies have the same meaning. This also means that the OWL2-based transformation offers a viable format for exchanging SBVR vocabularies between two SBVR tools, as an alternative to the XMI-based format specified in SBVR clause 13.

The scope of this work is SBVR vocabularies, excluding behavioral rules.

The mapping described here depends upon two machine readable files, both of which are available as open source:

- purl.org/sbvr2owl/SBVRTaggedText.xsd – the XML schema that is described in section 4.1 of this document. Referenced in this document by the prefix `sbvr-tt`.
- purl.org/sbvr2owl/VocabularyDescription.owl – an OWL ontology that implements the OWL annotations and entities that are described in sections 4.2 and 4.3 of this document. Referenced in this document by the prefix `sbvr-vd`.

This mapping has been applied to the Date-Time Vocabulary (DTV) to generate a set of DTV ontologies in OWL2.

---

[1] SBVR is the *Semantics of Business Vocabulary and Rules* specification from the Object Management Group (OMG), available at http://www.omg.org/spec/SBVR.

[2] The *OWL2 Web Ontology Language* standard of the World Wide Web Consortium (W3C), available at http://www.w3.org/TR/owl2-overview/.

Table of Contents

Table of Tables

1 Mapping SBVR Names and Namespaces to OWL2

SBVR has a fairly complex structure for concept naming and organization. Mapping this structure to OWL2 requires some care.

.1.1 Representations

In SBVR, the term "representation" is the relationship between an expression (e.g. the character sequence "c-a-r") and a meaning (e.g. the concept "car"). Representations include definitions, examples, and other forms, as well as names for concepts. Representation types that refer to concepts include "name" (a representation of an individual concept, what OWL calls an 'individual'), "term" (a representation of a general concept, what OWL calls a 'concept'), and "verb concept wording" (a representation of a verb concept, what OWL calls a 'property').

SBVR distinguishes between "primary" representations and alternative representations, which are called "synonyms" (for general and individual concepts) or "synonymous forms" (for verb concepts). In this SBVR→OWL mapping, the primary representations are mapped to (partial) URLs in OWL2 using camelCase name conversion: blanks are dropped, and the initial characters of the second and subsequent words are upper-cased. For example "driver drives car" becomes "driverDrivesCar". When camelCase conversion causes a lexical change in a representation, the original expression of the representation is captured in one of the following annotations:

Table 1: SBVR Primary Representations and OWL Annotations

SBVR Representation Type	OWL Annotation
primary term for a general concept	`sbvr-vd:term`
name for an individual concept	`sbvr-vd:name`
verb concept wording	`sbvr-vd:wordConceptWording`

Synonyms and synonymous forms are captured as `sbvr-vd:synonym` and `sbvr-vd:synonymousForm` annotations.

.1.2 Vocabularies and Terminological Dictionaries

SBVR concepts are grouped in "vocabularies" (which contain only the representations for concepts) and "terminological dictionaries" (which add the definitions and other semantics). Each vocabulary is associated with a language (e.g. "English", "Chinese"), and a terminological dictionary may "present" (incorporate) multiple vocabularies. Different vocabularies provide representations in different languages for the single set of concepts defined in a terminological dictionary. The SBVR `See:` caption may be used to link multiple representations of the same concept.

Each SBVR terminological dictionary and each vocabulary is mapped to its own OWL ontology. In some usage scenarios, the IRI of the OWL ontology, or just the domain name portion of the IRI, may be specified externally. In other scenarios, it may be convenient to form the OWL ontology IRI, by default, possibly by concatenating the characters "-owl" to the SBVR terminological dictionary or vocabulary IRI. The `rdfs:label` annotation is used to capture the original name of the vocabulary or terminological dictionary.

In order to maintain namespace separation between ontologies, the OWL name (IRI) of each concept in an ontology is formed by concatenating the ontology name (IRI) to the camelCase form of the concept name, separated by the right slash ('/') character. For convenience, the ontology name and slash are associated with an XML entity. For example, a terminological dictionary with an IRI of http://www.example.org/example might be mapped to an ontology named http://www.example.org/example-owl. An XML entity named `ex` might expand to the IRI with a trailing slash: http://www.example.org/example/. A general concept 'car' in this terminological dictionary would be mapped as `&ex;car`, which expands to http://www.example.org/example/car. Similarly, a verb concept 'driver drives car' would be referenced as `&ex;driverDrivesCar`.

SBVR vocabularies can import other vocabularies via the SBVR Structured English "Included Vocabulary:" caption. Imports are mapped to `owl:imports` statements and `sbvr-vd:includedVocabulary` annotations. Concept references are significantly affected by vocabulary imports, since the IRIs of imported concepts incorporate their ontology's IRIs. For example, a definition or structural rule in vocabulary `example` might reference a concept named `car` in vocabulary `other`. In order to convert the reference to the form `&other;car`, the mapping tracks the relationship of concepts to vocabularies for imported vocabularies that themselves import other vocabularies, to any depth.

.1.3 Synonyms

SBVR synonyms are alternative terms for noun concepts. For example, the concept "car" may have a synonym "auto". Structural rules may use synonyms to reference the concepts, so mapping Necessities to OWL (section

3) requires that the synonyms also be mapped to OWL. Synonyms mean the same thing as their primary terms, so synonyms of general concepts are mapped to OWL classes that are marked as `owl:EquivalentClass` of the OWL classes generated for the primary terms. Each synonym is also captured in an `sbvr-vd:synonym` annotation of the "primary term" class.

Synonyms of individual concepts are mapped to `owl:NamedIndividuals` that are marked owl:sameAs the `NamedIndividuals` generated for their primary names.

.1.4 Synonymous Forms

Synonymous forms are alternative wordings for verb concepts. For example, the verb concept "driver *drives* car" might have a synonymous form "driver *steers* car". Synonymous forms are mapped to OWL to support structural rules (Necessities) that use the synonymous forms. For example, the OWL equivalent of the rule "Each driver *steers* at most one car" depends upon the OWL equivalent of "driver *steers* car".

Mapping synonymous forms to OWL depends upon the number of roles of the verb concept, and, for binary verb concepts, the order of the roles of the verb. For all synonymous forms, the property generated for the primary verb concept wording includes an `sbvr-vd:synonymousForm` annotation containing the text of the synonymous form.

.1.4.1 Synonymous Forms of Unary Verb Concepts (Characteristics)

Synonymous forms of verb concepts that have just one role are generated as `owl:DatatypeProperty`, and as `owl:equivalentProperty` of the data property generated for the primary wording of the verb concept.

.1.4.2 Synonymous Forms of Binary Verb Concepts

Synonymous forms of verb concepts that have two roles are generated as `owl:ObjectProperty` or `owl:DatatypeProperty` according to the type of property generated for the primary verb concept wording, with `owl:equivalentProperty` to capture the relationship to the "primary" property.

When the roles of the synonymous form are reversed, compared to the roles of the primary form, the synonymous form is understood as an inverse of the primary form. For example, "car *is driven by* driver" is an inverse of "driver *drives* car". The OWL mapping uses `owl:inverseOf` instead of `owl:equivalentProperty`.

If a verb concept participates in any of the OWL logical properties, per Table 6, then any synonymous forms are marked with the same OWL logical properties.

When a synonymous form uses verb symbols for property association ("*has*", "*of*", and "*is of*") or partitive ("*includes*" or "*included by*") verbs, the primary verb concept is automatically marked as a property or partitive verb concept, even if the primary verb concept wording does not use the verb symbol. For example, a verb concept "driver *owns* car" that has a synonymous form "driver *of* car" is treated as a property association verb concept as described in section 2.3.3.

.1.4.3 Synonymous Forms of *n*-ary Verb Concepts

Synonymous forms of *n*-ary verb concepts are captured only as `sbvr-vd:synonymousForm` annotations because structural rules that use *n*-ary verb concepts are not mapped to OWL.

.1.4.4 Character Sets

SBVR does not limit the character set used in representations, and some vocabularies use characters such as '<' and '≤' in verb concepts. XML accepts a much more limited character set for identifiers, so characters are translated as shown in this table:

Table 2: Mapping Characters in XML Identifiers

SBVR Character	Mapped As
<	_lt_
=	_eq_
>	_gt_
≤	_le_
≥	_ge_
+	_plus_
other characters not in the sets a-z, A-Z, _, -, and .	_

2 Mapping SBVR Domain Vocabularies to OWL2

Many elements of SBVR vocabularies are mapped directly to equivalent OWL2 elements. Where necessary, an OWL annotation is introduced to capture any SBVR meaning that is not expressible in OWL2.

.2.1 SBVR General Concepts

SBVR general concepts map to `owl:class`, but (as discussed below) verb concepts may map to either `owl:ObjectProperty` or `owl:DataTypeProperty` depending upon the supertype (more general concept) of each concept . In SBVR Structured English[3], the supertypes may be specified in any of several ways:

[3] See Annexes A and B of the SBVR specification.

- As the first term mentioned in a Definition that starts with "<supertype> that ...."
- As the two or more terms mentioned in a Definition of the form "<supertype1> that *is* a <supertype2> ...."
- Mentioned in a "General Concept:" caption.
- As the first term mentioned in a Definition that starts with "<supertype> …."[4]

The Structured English techniques listed above can be mixed-and-matched within a single glossary entry. Each entry may have multiple "General Concept:" captions, and each caption can have multiple supertypes (implying multiple inheritance). OWL2 `subClassOf` assertions capture the generalization relationships of SBVR concepts that have just one general concept. SBVR general concepts that have multiple general concepts, are mapped as shown in Table 5.

.2.2 SBVR Roles

An SBVR role is a noun concept that corresponds to things that play a part in instances of verb concepts. SBVR defines four concepts related to roles, which are distinguished here for clarity:

Table 3: OWL Equivalents of SBVR Roles

Type of Role	OWL2 Equivalent	Discussion
verb concept role	included as the placeholders in `sbvr-vd:verbConceptWording` and `sbvr-vd:synonymousForm` annotations	In the simple case, a verb concept role simply names a general concept to which the verb applies. For example, the verb concept 'time interval *has* duration' has two verb concept roles: 'time interval' and 'duration', indicating the domain and range of the verb concept. Alternatively, a verb concept role may be a situational role (see below).

[4] This style is not formally documented in SBVR Structured English, but is found in example vocabularies.

Type of Role	OWL2 Equivalent	Discussion
placeholder	included in `sbvr-vd:verbConceptWording` and `sbvr-vd:synonymousForm` annotations	A placeholder is a designation for a verb concept role. A verb concept may have multiple synonymous forms that use different placeholders, but the placeholders designate the same verb concept roles. For example, in the synonymous forms 'time interval *has* particular duration' and 'time interval *has* duration', the placeholders 'duration' and 'particular duration' designate the same verb concept role.
verb concept role designation	incorporated in `sbvr-vd:definition`, `sbvr-vd:necessity`, and `sbvr-vd:possibility` annotations	A designation of a verb concept role, used in the 'invocation' of a verb concept. For example, 'duration' in 'the duration *of* some time interval'. Note that the same expression may designate a placeholder, a verb concept role designation, and a situational role but they are always logically different concepts.
situational role	owl:class with `sbvr-vd:hasConceptType` annotation of `SituationalRole`	A general concept that is also a role. For example, the glossary entry: student Concept Type: role General Concept: person ... defines the situation role 'student' as a role of a 'person'.

Verb concept roles are represented by placeholders, which appear in verb concept wordings. These are captured as `sbvr-vd:verbConceptWording` or `sbvr-vd:synonymousForm` annotations. The placeholders can be distinguished by the XML tags used to identify the parts of speech within these annotations, as described in section 4.1, below.

Verb concept role designations are indicated by their usage within definitions and rules. They are identified as nouns by the XML tags in the annotations for the definitions and rules.

Situational roles are general concepts that are also roles. A typical use is defining the range of a role. For example, a vocabulary that has the verb concept 'car *has* driver' might define the situation role 'driver' with general concept 'person'. This means that 'driver' ranges over 'person'. Situational roles map to owl:class with an `sbvr-vd:hasConceptType` annotation of `sbvr-vd:situationalRole`.

.2.3 SBVR Verb Concepts

SBVR verb concepts map to OWL object properties or data properties, depending upon the number and type of the verb concept roles.

The primary verb concept wording of each verb concept is captured in the `sbvr-vd:verbConceptWording` annotation. Any synonymous forms are captured in `sbvr-vd:synonymousForm` annotations.

.2.3.1 Characteristics (Unary Verb Concepts)

Characteristics are verb concepts with one verb concept role: unary verb concepts, such as 'car *is red*'. These map to OWL data properties of type `xsd:boolean`. The data property name is derived from the verb concept wording of the characteristic. The example becomes an `owl:datatypeProperty` named 'carIsRed' whose domain is 'car' and range is `xsd:boolean`.

.2.3.2 Binary Verb Concepts

Binary verb concepts are verb concepts with two roles. Two important types of binary verb concepts are partitive verb concepts and property association verb concepts, which are both described below.

Binary verb concepts that are not property associations are mapped to `owl:ObjectProperty` or `owlDataTypeProperty`, with appropriate domain and range axioms. The domain is the general concept that the first verb concept role ranges over. The range is the general concept that the second verb concept ranges over, or one of the built-in types inventoried in the following table. Some of this list is taken from SBVR clause 8.6; the remaining come from the Date-Time Vocabulary. Binary verb concepts whose range is any of these SBVR types are mapped to `owl:DataTypeProperty` according to the following table. For clarity, the table shows both OWL and XSD datatypes because OWL2 does not support all the XSD datatypes. In those cases, OWL implementations should use `xsd:string`, but interpret the string as in the corresponding XML schema datatype and use an `sbvr-vd:hasConceptType` annotation to indicate the SBVR type.

Table 4: Mapping SBVR Concept Types to OWL Datatypes

SBVR Concept	OWL Datatype	XSD Datatype	Comment or Example
expression	xsd:stringError: Reference source not found	xsd:string	
text	xsd:string[1]	xsd:string	
number	xsd:decimal[2]	xsd:decimal	
integer	xsd:integer	xsd:integer	
nonnegative integer	xsd: nonNegativeInteger	xsd: nonNegativeInteger	
positive integer	xsd: positiveInteger	xsd: positiveInteger	
calendar date	xsd:string[3,4,6]	xsd:date	Year, or month & year, or day & month & year
time of day	xsd:string[3,4,6]	xsd:time	Hour, or minute & hour, or second & minute & hour
date time	xsd:dateTime[4]	xsd:dateTime	Combination of a calendar date and a time of day.
Gregorian year coordinate	xsd:string[3,4]	xsd:gYear	2012
Gregorian month coordinate	xsd:string[3,4]	xsd:gMonth	January
Gregorian day of year coordinate	xsd:string[3,4]	(none)	day of year 45 (i.e. the 45th day of some calendar year)
Gregorian day of month coordinate	xsd:string[3,4]	(none)	day of month 12 (i.e. the 12th day of some calendar month)

SBVR Concept	OWL Datatype	XSD Datatype	Comment or Example
Gregorian year month coordinate	xsd:string[3,4]	xsd:gYearMonth	July 2010
Gregorian year month day coordinate	xsd:string[3,4]	xsd:date	9 July 2010
Gregorian year day coordinate	xsd:string[3,4]	(none)	2010 day 33 (i.e. the 33rd calendar day of 2010)
Gregorian month day coordinate	xsd:string[3,4]	xsd:gMonthDay	9 July
hour coordinate	xsd:string[3,4]	(none)	11 p.m.
minute coordinate	xsd:string[3,4]	(none)	minute 23 (i.e. minute of hour 23 of some hour of day)
second coordinate	xsd:string[3,4]	(none)	second 45 (i.e. second of minute 45 of some minute of hour)
hour minute coordinate	xsd:string[3,4]	(none)	11:23
hour minute second coordinate	xsd:string[3,4]	xsd:time	11:23:45
day of week coordinate	xsd:string[3,5]	(none)	Tuesday, day of week 2
week of year coordinate	xsd:string[3,5]	(none)	week 33 (i.e. the 33rd calendar week of a calendar year)
week day coordinate	xsd:string[3,5]	(none)	Tuesday week 33

SBVR Concept	OWL Datatype	XSD Datatype	Comment or Example
year week coordinate	xsd:string[3,5]	(none)	2010 week 33
year week day coordinate	xsd:string[3,5]	(none)	Tuesday 2010 week 33

Notes:

1. We map SBVR texts and expressions to `xsd:string`, rather than `rdf:PlainLiteral`, because the SBVR metamodel provides no way to indicate the language of a text. (Entire SBVR vocabularies can be associated with languages, but not individual texts.)
2. We map SBVR numbers to `xsd:decimal` on the assumption that business people think of decimal numbers. Basic SBVR has no way to specify real numbers.
3. OWL2 does not support the equivalent `xsd` datatype, so we use xsd:string instead. Use the `sbvr-vd:hasConceptType` annotation to preserve the SBVR type.
4. The Date-Time Vocabulary clause 13 interchange format for durations, dates, and times is a superset of the ISO 8601 and the XML Schema lexical syntax for each of these.
5. XML Schema does not support dates that involve weeks. Such dates should be interchanged as described in the Date-Time Vocabulary, clause 18.3, which is consistent with ISO 8601.
6. Alternatively, depending upon the application requirements, these types could be mapped to xsd:dateTime.

.2.3.3 Property Association Verb Concepts

SBVR clause 11.1.5 categorizes some binary verb concepts (verb concepts with exactly two roles) as “property association” verb concepts. These are verb concepts that define attributes of classes, a "quality or trait actually belonging to a thing itself". Examples include "person *has* heart" and "board of directors *of* company".

In SBVR Structured English, property association verb concepts are conventionally defined using the verb symbols '*has*', '*is of*', or '*of*', as in 'car *has* engine', either in the primary verb concept wording or in a synonymous form of the verb concept. For example, the verb concept 'driver *owns* car' with a synonymous form 'driver *has* car' is a property association because of the verb symbol '*has*' in the synonymous form. Tools are free to provide other ways to identify these verb concepts. One possibility is to use 'General Concept: property association'.

The meaning of all property association verb concepts is that one role of the verb concept is a "subject" class whose instances have the property (attribute) specified by the other role. For example, the 'car' subject class has the property 'engine'. Since all property association verb concepts have the same meaning, we predefine OWL object (`sbvr-vd:has`) and data (`sbvr-vd:hasData`) properties. We map each property association verb concept to an OWL object or data property that is a subproperty of one of these predefined properties.

The `owlObjectProperty` (or `owl:DataTypeProperty`) relates the `owl:class` that is the domain of the property to the subject class (or data type) that the property ranges over. For example, an SBVR property association verb concept 'train *has* engine', maps to an OWL `ObjectPropertyAssertion (trainHasEngine train engine)` and `SubObjectPropertyOf(trainHasEngine sbvr-vd:has)`.

In SBVR Structured English, property association verb concepts implicitly have inverse forms. For example, "driver *of* car" has an implicit inverse "car *has* driver". For property association verbs that map to object properties, we automatically create the corresponding inverse object properties to support structural rules that use the inverses. OWL does not support inverse data properties, so we do not generate inverses for property association verbs whose domain is a data type. This is not a limitation because there can be no structural rules that constrain the cardinality of the relationship of a data type to an object property.

.2.3.4 Partitive Verb Concepts

SBVR clause 11.1.5 categorizes some binary verb concepts (verb concepts with exactly two roles) as "partitive" (whole-part) verb concepts. These are verb concepts that capture the semantic that a part is in the composition of a whole; i.e. mereology. In Structured English, these verb concepts use the verb symbol "*include*", or a variation such as "*includes*" or "*included*" or "*included by*". Alternatively, these concepts could use the caption "General Concept: partitive verb concept".

We map partitive verb concept using the same pattern as property association verb concepts (section 2.3.3), but make them sub properties of `sbvr-vd:isPartitive` (when the range of the verb concept is an owl class) or `sbvr-vd:isPartitiveData` (when the range is a data type). For object properties, we also automatically generate the inverse form.

.2.3.5 *N*-ary Verb Concepts

Verb concepts that have more than 2 verb concept roles are called *n*-ary verb concepts. These are relationships among more than two concepts. OWL2 has no equivalent, but there are several well-known patterns for handling such relationships. This paper adopts "Pattern 1" as described in Defining N-ary Relations on the Semantic Web (W3C, 2006) by Noys, Rector, Hayes, and Welty. In this pattern, each *n*-ary verb concept is mapped to an OWL2 class, and each verb concept role is mapped to an object or data property of the class, according to whether the role's range is listed in Table 4. Some writers name this technique "reification".

For example, an SBVR verb concept 'driver *applies for* car insurance *from* insurance company' has three verb concept roles. Assuming that the roles are already defined in OWL, and all three roles are classes rather than data types, the OWL2 mapping of this verb concept includes these assertions:

1. `Class(driverAppliesForCarInsuranceFromInsuranceCompany)`
2. `ObjectPropertyAssertion(driverAppliesForCarInsuranceFromInsuranceCompany_driver driverAppliesForCarInsuranceFromInsuranceCompany driver)`
3. `ObjectPropertyAssertion(driverAppliesForCarInsuranceFromInsuranceCompany_carInsurance driverAppliesForCarInsuranceFromInsuranceCompany carInsurance)`
4. `ObjectPropertyAssertion(driverAppliesForCarInsuranceFromInsuranceCompany_insuranceCompany insuranceCompany)`

Line 1 of this example defines the reified class. Lines 2 through 4 define three object properties, one for each of the roles. Note that the object properties are uniquely named by combining the class name with the role names. If a role is a datatype (per Table 4Error: Reference source not found), then a `DataPropertyAssertion` is used instead of an `ObjectPropertyAssertion`. Each role is restricted to participate just once in the corresponding `Object` or `DataPropertyAssertion`, by defining the class shown in line 1, above:

```
<owl:equivalentClass>
         <owl:Restriction>
            <owl:onProperty rdf:resource=
"driverAppliesForCarInsuranceFromInsuranceCompany_driver"/>
            <owl:onClass rdf:resource="driver"/>
            <owl:qualifiedCardinality
                  rdf:datatype="xsd:nonNegativeInteger">1
            </owl:qualifiedCardinality>
         </owl:Restriction>

      </owl:equivalentClass>
```

The OWL class is defined as a subclass of `sbvr-vd:verbConcept` to capture the SBVR meaning of the class. The object or data properties are sub-properties of `sbvr-vd:verbConceptRole` or `sbvr-vd:verbConceptDataRole`, respectively, to distinguish them from properties that are mapped from SBVR binary verb concepts.

.2.4 SBVR Individual and Unitary Concepts

SBVR individual and unitary concepts are concepts whose extension is limited to at most one individual. Individual concepts correspond to the same individual at all times, whereas the extension of unitary concepts may change over time. For example, the concept "Washington, D.C." is an individual concept because its extension always refers to the same thing, whereas "Air Force One" is a unitary concept because its extension (the airplane that is carrying the President of the U.S.A.) varies over time.

SBVR uses the term "individual concept", rather than "individual", because it restricts its attention to what OWL2 calls "T-box" features. The "individual concept" idea permits SBVR to support rules that refer to individuals without formally adopting the equivalent of an "A-box".

This creates an awkward choice when mapping individual concepts to OWL2. We could map individual concepts to OWL2 classes, or to OWL2 individuals, or both. The choice is somewhat arbitrary. We feel that the best "semantic match" of SBVR individual concepts is to OWL2 named individuals, so we make that choice. For example, the SBVR individual concept "Washington, D.C." is mapped to OWL2 `NamedIndividual(WashingtonD.C.)`, with an annotation `sbvr-vd:hasConceptType` of `sbvr-vd:individualConcept`. If the individual concept is defined in SBVR as a subtype of "city", then we add `ClassAssertion(city WashingtonD.C.)`.

In SBVR Structured English, the supertype of individual concept are defined either using General Concept: captions, or via definitions using the pattern "the <supertype> …."

SBVR unitary concepts are true concepts and map to OWL2 classes as described in section 2.1. The annotation `sbvr-vd:hasConceptType` is used to indicate that the SBVR concept is `sbvr-vd:unitaryConcept`. OWL2 (apparently) has no way to limit the size of the extension of a concept, so the semantic that a unitary concept has at most one instance is lost in OWL2.

.2.5 SBVR Categorization Types

An SBVR categorization type is a concept whose instances are categories of some concept. A categorization type is similar to a UML power type. For example, in the Date-Time Vocabulary, one instance of the categorization type 'quantity kind' is the concept 'duration', which itself has instances such as "1 day", "5 hours 15 minutes", etc. Other instances of 'quantity kind' might include 'length', 'mass', etc.

Categorization types are mapped to OWL as described in section 2.1 for any general concept. Add the annotation `sbvr-vd:hasConceptType` with value `sbvr-vd:categorizationType`.

Concepts that are instances of categorization types are also mapped as in section 2.1. No special annotation identifies these concepts, since they can be recognized by their `subClassOf` relationship to the categorization type.

These concepts may be referenced as individuals in rules, but the references do not cause difficulties in this mapping because we do not map most rules.

.2.6 SBVR Characteristic Types

SBVR characteristic types are categorization types whose instances are characteristics. The canonical example is the concept 'color' whose instances are the characteristics (unary verb concepts) 'thing *is red*', 'thing *is blue*', etc.

Characteristic types are mapped as described in section 2.1, like other general concepts. The annotation `sbvr-vd:hasConceptType`, with value `sbvr-vd:characteristicType`, identifies classes that are SBVR characteristic types.

The instances of characteristic types are characteristics that are mapped to OWL data properties of type `xsd:Boolean`, as described in section 2.3.1. The relationships of the characteristics to the characteristic types is captured with `sbvr-vd:characteristicOf` annotations.

.2.7 SBVR Reference Schemes

SBVR reference schemes specify how individuals may be identified by the value of one or more of their properties. For example, nations may be identified by their name or the design of their flag, or by their ISO country codes. Reference schemes are equivalent to OWL2 keys. Each role mentioned in a reference scheme maps to a property in an OWL2 `HasKey` assertion. The Structured English text of each SBVR reference scheme is preserved in an `sbvr-vd:referenceScheme` annotation.

For example, an SBVR reference scheme 'name *of* country' maps to `HasKey(country () name)`, with an `sbvr-vd:referenceScheme` annotation that has 'name *of* country' using `sbvr-tt:TaggedText` tagging.

3 Mapping SBVR Definitions and Structural Rules to OWL2

Using Structured English, general, individual, and unitary concepts may be defined formally (using only keywords and defined concepts) or informally (using one or more undefined words). The meaning of informal definitions can be interpreted by machine only to the extent that they start with "<concept> that …", which means that the defined concept is a specialization of <concept>. Formal definitions have formal meaning, but the expressive power of Structured English is much larger than OWL2, so most have no OWL2 equivalent.

Some SBVR definitions are specified according to patterns that can be mapped to OWL2. The following table lists these patterns and their OWL2 equivalents.

Table 5: Patterns for Concept Definitions

Patterns for Definitions of <concept$_1$>		
SBVR Pattern	OWL Equivalent	Description
<concept$_2$> that …	subClassOf(<concept$_1$> <concept$_2$>)	single inheritance
<concept$_2$> that *is* a <concept$_3$> (… that *is* a <concept$_n$>)	equivalentClass(owl:Class (intersectionOf(<concept$_2$>, <concept$_3$>, … <concept$_n$>)))	multiple inheritance
<concept$_2$> or <concept$_3$> (or … <concept$_n$>)	equivalentClass(owl:Class (unionOf(<concept$_2$>, <concept$_3$>, … <concept$_n$>)))	extensional definition
<concept$_2$> …	subClassOf(<concept$_1$> <concept$_2$>)	single inheritance

SBVR necessity rules constrain concepts in some way. As with definitions, some necessities use standard patterns that have OWL2 equivalents.

Table 6: Patterns for Necessity Rules

Patterns for Necessity Rules		
SBVR Pattern	OWL Equivalent	Description
Each <concept$_1$> <verb> exactly *n* <concept$_2$>.	ObjectExactCardinality(*n*, concept1VerbConcept2 concept1) or DataExactCardinality(*n*, concept1VerbConcept2 concept1)	exact cardinality
Each <concept$_1$> <verb> has *n of* <concept$_2$>.	ObjectExactCardinality(*n*, concept1VerbConcept2 concept1) or DataExactCardinality(*n*, concept1VerbConcept2 concept1)	exact cardinality
<concept$_1$> <verb> has *n* of <concept$_2$>.	ObjectExactCardinality(*n*, concept1VerbConcept2 concept1) or DataExactCardinality(*n*, concept1VerbConcept2 concept1)	exact cardinality

Patterns for Necessity Rules		
SBVR Pattern	OWL Equivalent	Description
Each <concept$_1$> <verb> at least *n* <concept$_2$>.	ObjectMinCardinality(*n*, concept1VerbConcept2 concept1) or DataMinCardinality(*n*, concept1VerbConcept2 concept1)	minimum cardinality
Each <concept$_1$> <verb> at most *n* <concept$_2$>.	ObjectMaxCardinality(*n*, concept1VerbConcept2 concept1) or DataMaxCardinality(*n*, concept1VerbConcept2 concept1)	maximum cardinality
Each <concept$_1$> <verb> at least *n* <concept$_2$> and at most *m* <concept$_2$>.	ObjectIntersectionOf(ObjectMinCardinality(*n*, concept1VerbConcept2 concept1) ObjectMaxCardinality(*m*, concept1VerbConcept2 concept1)) or ObjectIntersectionOf(DataMinCardinality(*n*, concept1VerbConcept2 concept1) DataMaxCardinality(*n*, concept1VerbConcept2 concept1))	combination of minimum and maximum cardinality
No <concept$_1$> *is* a <concept$_2$>.	DisjointClasses(concept1 concept2)	The intersection of the extensions of the two concepts is empty
Each <concept> <verb> the <concept>.	owl:ReflexiveProperty	reflexive binary concept
Each <concept> <verb$_1$> not <verb$_2$> the <concept>.	owl:IrreflexiveProperty	irreflexive binary concept
If the <concept$_1$> <verb> some <concept$_2$> and the <concept$_2$> <verb> some <concept$_3$> then the <concept$_1$> <verb> <concept$_3$>.	owl:TransitiveProperty	transitive binary concept

In the cardinality patterns listed above, the keyword "A" may be used instead of "Each", and the keyword "one" may be used instead of the cardinality value, *n*.

SBVR Possibility rules remind users that instances may form unexpected relationships or take on unexpected values. For example, "Each situation kind *has* zero or more occurrences", to make it clear that it is permissible for situation kinds to have no occurrences. OWL2 has no way to express such permissions, so Possibility statements are captured in OWL2 only as annotations.

4 Mapping SBVR captions and representations to OWL Annotations

This is a list of the various SBVR Structured English captions, the corresponding SBVR representation types and how we map them to OWL2 annotations. It assumes that the portions of the 'meaning' side of the SBVR metamodel is mapped to an OWL class structure that can be referenced as the domain and/or range of these annotations.

Where the definition of an SBVR representation matches that of an existing RDFS, OWL2, SKOS or Dublin Core annotation, the latter is proposed if the domain and the range are the same. If the domain or range are different but the concept matches an existing RDFS, OWL2, SKOS, or Dublin Core annotation, a sub-annotation is defined.

SBVR annotations are listed in SBVR Annex C, "Structured English". Some annotations correspond to one of the SBVR "representation types" given in SBVR figure 8.4, figure 11.6, or 11.7. The table below gives the correspondence between captions and representation types because the correspondence is not documented in the SBVR specification.

Table 7: Mapping Structured English Captions to OWL Annotations

Structured English Caption	SBVR Representation Type	SBVR Informal Meaning	OWL2 Equivalent
(primary term for a general concept)	signifier	Identifies a general concept	Convert to camelCase, eliminating blanks. Add a `sbvr-vd:term` annotation with the original name if the converted name is different.
(primary term for an individual concept)	name	Identifies an individual concept	Same as a name (signifier) for a general concept, but use `sbvr-vd:name` annotation if necessary.

Structured English Caption	SBVR Representation Type	SBVR Informal Meaning	OWL2 Equivalent
(primary verb concept wording for a verb concept)	verb concept wording	Identifies a verb concept	Convert to camelCase as above. Capture the original verb concept wording as `sbvr-vd:verbConceptWording`. See section 2.3.
(primary term for a glossary entry of a role)	signifier	Identifies a role with a placeholder	A role appears in an OWL ontology only if it is a situation role, in which case it has an `sbvr-vd:situationalRole` annotation. See section 2.2.
(primary term for a glossary entry for a vocabulary)		Identifies a vocabulary with a name	Convert to camelCase, as for general concepts. Use either an externally-specified string or the camelCase result as the last part of the ontology IRI, possibly suffixed with `-owl`. Capture the original vocabulary name as `rdfs:label`.
Concept Type		Used when a glossary entry is other than type general concept, verb concept, or individual concept to specify the metamodel type of the entry	Map as `sbvr-vd:hasConceptType` and map the glossary entry to OWL per the specified concept type.
Definition	definition	expression that defines a concept	`sbvr-vd:definition`

Structured English Caption	SBVR Representation Type	SBVR Informal Meaning	OWL2 Equivalent
Description	description	expression that gives a "verbal portrait" of a concept	Process to recognize any supertypes, as described in sections 2.1 and 2.4. Capture the text of the definition in an `sbvr-vd:description` annotation.
Dictionary Basis		citation of a dictionary source	`sbvr-vd:dictionaryBasis`
Example	descriptive example	descriptive material that is a sample of a concept	`sbvr-vd:example`
General Concept		Specifies the supertype of a concept	`owl:subClassOf`
	icon	nonverbal designation whose signifier is a picture	`sbvr-vd:icon`
Included Vocabulary		Imports one vocabulary into another	Map to `owl:imports`; capture as `sbvr-vd:includedVocabulary`
Language		Specifies the human language used for representations	Capture as `sbvr-vd:vocabularyLanguage`.
Namespace URI		Specifies the URI or a vocabulary or a concept that has properties	Map to ontology or IRI, as discussed in section 1.2, and capture as `sbvr-vd:namespaceURI`.
Necessity		Specifies a structural rule	Capture as `sbvr-vd:necessity`; also map to OWL restriction if the Necessity statement uses any of the patterns given in section 3.

Structured English Caption	SBVR Representation Type	SBVR Informal Meaning	OWL2 Equivalent
	nonverbal designation	designation that is not expressed as words of a language	`sbvr-vd:symbol`
Note	note (comment, remark)	comment about a concept	`sbvr-vd:note`
Possibility		Specifies structural advice	Capture as `sbvr-vd:possibility`.
Reference Scheme		Defines a reference key	Map to `owl:HasKey`, as described in section 2.7. Capture the Structured English text as `sbvr-vd:referenceScheme`.
See		cross-reference to another glossary entry	`sbvr-vd:synonym` or `sbvr-vd:synonymousForm` and `sbvr-vd:see`, referencing the target concept
Source	reference	citation of the source of a concept or a representation	`sbvr-vd:source`
Speech Community		Identifies the speech community that defines a vocabulary.	Capture as `sbvr-vd:speechCommunity`.
Subject Field		Specifies a qualifier for a primary term	Use to extend the URI of the ontology; capture as `sbvr-vd:subjectField`
Synonym		alternative expression of a general or individual concept	Capture as `sbvr-vd:synonym`

Structured English Caption	SBVR Representation Type	SBVR Informal Meaning	OWL2 Equivalent
Synonymous Form		alternative expression of a verb concept wording	sbvr-vd:synonymousForm

.4.1 SBVR Tagged Text Format

SBVR Structured English text often contains a mixture of plain text (usually shown in black normal font), nouns (underlined teal), verbs (*italic blue*), keywords (red), and names (double-underlined teal). Multiple uses of the same noun may be distinguished by subscripts. For example, a Necessity rule given as "If a time interval$_1$ *overlaps* a time interval$_2$, then the time interval$_1$ *is* not *before* the time interval$_2$." This *styled text* is always applied for terms, verb concept wordings, Definitions, Necessities, and Possibilities. Styled text may also be used with Notes, Examples, Descriptions, Source and Dictionary References, and other text.

The styled text carries useful meaning. Consider these examples: "each boat sails ... " and "each boat *sails* ...." In the first example, the quantifier "each" applies to a noun "boat sails", whereas in the second example, the quantifier applies to a noun "boat" and "*sails*" is a verb.

When we transform an SBVR vocabulary to OWL2, we capture such text statements as annotations such as `sbvr-vd:definition`. To enable a fully-reversible transformation, we must identify the parts of speech in the text. We do this by defining the range of these annotations as a special form of text literal that may contain these internal XML tags:

Table 8: sbvr-tt:TaggedText

Tag Name	Purpose
sbvr-tt:keyword	Structured English keyword, such as "and"
sbvr-tt:name	Name for an individual concept, such as "Mark"
sbvr-tt:noun	Term for a general concept, such as "time interval"
sbvr-tt:subscript	Number that distinguishes among multiple references to the same noun, such as "time interval$_1$"
sbvr-tt:verb	Symbol for a verb, such as "*is before*" in "time interval$_1$ *is before* time interval$_2$".

To encode these XML tags, we use an XML schema that we call "SBVRTaggedText". This schema supports a mixture of plain text and the XML tags shown above. This example shows how the XML tags are used to delimit the various parts of speech. The original text styles are shown here to illustrate the example, but are not present in the OWL2 annotation.

```
"<sbvr-tt:keyword>If a</sbvr-tt:keyword> <sbvr-sbvr-
tt:noun>time interval</sbvr-sbvr-tt:noun><sbvr-
tt:subscript>1</sbvr-tt:subscript> <sbvr-
tt:verb>overlaps</sbvr-tt:verb> <sbvr-tt:keyword>a</sbvr-
tt:keyword> <sbvr-sbvr-tt:noun>time interval</sbvr-sbvr-
tt:noun><sbvr-tt:subscript>2</sbvr-tt:subscript> <sbvr-
tt:keyword>, then the</sbvr-tt:keyword> <sbvr-sbvr-
tt:noun>time interval</sbvr-sbvr-tt:noun><sbvr-
tt:subscript>1</sbvr-tt:subscript>  <sbvr-tt:verb>is</sbvr-
tt:verb> <sbvr-tt:keyword> not</sbvr-tt:keyword> <sbvr-
tt:verb>before</sbvr-tt:verb> <sbvr-tt:keyword>the</sbvr-
tt:keyword> <sbvr-sbvr-tt:noun>time interval</sbvr-sbvr-
tt:noun><sbvr-tt:subscript>2</sbvr-tt:subscript> <sbvr-
tt:keyword>.</sbvr-tt:keyword>"
```

Several points are worth noting:

- As with any XML text, the less-than ("<") and greater-than characters must be encoded as "<" and "&gt".
- White space that is adjacent to the start or end of tagged text may arbitrarily be included or excluded from the tagged text.
- A subscript always appears outside a <sbvr-tt:noun> tag.

<sbvr-tt:TaggedText> is defined as this XML Schema, which is available at purl.org/sbvr2owl/SBVRTaggedText.xsd:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://purl.org/sbvr2owl/SBVRTaggedText.xsd"
      xmlns:tns="http://purl.org/sbvr2owl/SBVRTaggedText.xsd"
      elementFormDefault="qualified">
      <element name="TaggedText" type="tns:TaggedTextType"
id="TaggedText"/>
      <complexType name="TaggedTextType" mixed="true">
            <choice maxOccurs="unbounded">
                  <element name="keyword" type="string" />
                  <element name="term" type="string" />
                  <element name="verb" type="string" />
                  <element name="name" type="string" />
                  <element name="subscript" type="string" />
            </choice>
      </complexType>
       </schema>
```

When the content of an annotation, such as `sbvr-vd:definition`, is SBVRTaggedText, the annotation must include `rdf:datatype="&sbvr-tt:TaggedText"` to indicate how the annotation should be interpreted.

Many of the annotations listed in this document are motivated by and derived from the Dublin Core, RDFS, SKOS, and OMG Specification Metadata[5] annotations. The fact that their range is `<sbvr-tt:TaggedText>` means that they must be derived from those annotations rather than be direct use of them.

.4.2 Inventory of OWL Annotations

These prefixes are used in the following table:

dct: annotations defined by Dublin Core terms

rdfs: RDF Schema

sbvr-vd: annotations and entities defined to capture aspects of SBVR; defined in purl.org/sbvr2owl/VocabularyDescription.owl

sbvr-tt: SBVR tagged-text format, as described in section 4.1 and available at purl.org/sbvr2owl/SBVRTaggedText.xsd

skos: annotations defined by the Simple Knowledge Organization System

sm: OMG Specification Metadata

An "(any)" domain or range means that any OWL class or property may participate as the domain or range of the annotation.

---

[5] The OMG Specification Metadata annotation set is proposed at the OMG for capturing metadata associated with a formal specification. Example annotations include document reference number, publication date, and copyright information.

Table 9: Inventory of OWL Annotations

Annotation Name	Domain	Range	Description
rfs:label	rdfs: resource	rdfs:literal	Captures the SBVR name of an SBVR vocabulary or terminological dictionary
sbvr-vd:characteristicOf	OWL data property	OWL class	Captures the relationship of a characteristic to a characteristic type.
sbvr-vd:definition	(any)	sbvr-tt:Tagged Text	Text of an SBVR definition; sub-annotation of skos:definition
sbvr-vd:description	(any)	sbvr-tt:Tagged Text	Text of an SBVR description; sub-annotation of dct:description
sbvr-vd:dictionaryBasis	(any)	sbvr-tt:TaggedText	Citation of a dictionary source; sub-annotation of sbvr-vd:source
sbvr-vd:example	(any)	sbvr-tt:TaggedText	Text of an SBVR example; sub-annotation of skos:example.
sbvr-vd:hasConceptType	(any)	sbvr-vd:concept	Used to specify the type of an SBVR concept if it is not obvious or if the SBVR glossary entry has a "Concept Type:" caption
sbvr-vd:hasGeneralConcept	(any)	sbvr-vd:concept	Used when the SBVR vocabulary has a “General Concept” caption
sbvr-vd:icon	(any)	xsd:anyURI	Icon for a concept

Annotation Name	Domain	Range	Description
sbvr-vd:includedVocabulary	(any)	sm:bibliographicCitation	Identifies a vocabulary that is included within another vocabulary; sub-annotation of sm:dependsOn
sbvr-vd:name	owl:namedIndividual	xsd:string	Name of an SBVR individual concept. Used only if the camelCase form o the name differs from the SBVR form. Sub-annotation of skos:prefLabel.
sbvr-vd:namespaceURI	(any)	xsd:anyURI	Identifies the original URI of a vocabulary or attributive namespace
sbvr-vd:necessity	(any)	sbvr-tt:TaggedText	Text of an SBVR Necessity
sbvr-vd:note	(any)	sbvr-tt:TaggedText	Text of an SBVR note; sub-annotation of skos:note.
sbvr-vd:possibility	(any)	sbvr-tt:TaggedText	Text of an SBVR Possibility
sbvr-vd:referenceScheme	(any)	sbvr-tt:TaggedText	Text of an SBVR reference scheme.
sbvr-vd:see	(any)	(any)	Synonym provided with a cross-reference to another concept; sub-annotation of rdfs:isDefinedBy

Annotation Name	Domain	Range	Description
sbvr-vd:source	(any)	sbvr-tt:Tagged Text	citation of a source reference; sub-annotation of sm:directSource
sbvr-vd:speechCommunity	(any)	sm:organization	Associates a speech community (by name) with a vocabulary. Example: the "EU-Rent English Community" specifies the "EU-Rent English Vocabulary".
sbvr-vd:subjectField	(any)	xsd:string	Identifies a "subject field" that uses specialized terminology. Example: medicine.
sbvr-vd:symbol	(any)	xsd:anyURI	Symbol for a concept
sbvr-vd:synonym	(any)	xsd:string	Synonym of a concept; sub-annotation of skos:altLabel
sbvr-vd:synonymousForm	(any)	sbvr-tt:tagged-text	Synonym for a verb concept; sub-annotation of skos:altLabel
sbvr-vd:term	owl:class	xsd:string	Name of an SBVR general concept. Used only if the camelCase form o the term differs from the SBVR form. Sub-annotation of skos:prefLabel.

Annotation Name	Domain	Range	Description
sbvr-tt:verbConceptWording	(any)	sbvr-tt:TaggedText	Primary wording of a verb concept; sub-property of skos:prefLabel
sbvr-vd:vocabularyDescription	(any)	xsd:string	Vocabulary name, if not the same as the ontology name. Sub-annotation of sm:specificationTitle
sbvr-vd:vocabularyLanguage	(any)	sbvr-vd:language	Identifies the language of a vocabulary; sub-annotation of dct:language
skos:hiddenLabel	(any)	(any)	Identifies a prohibited term or verb concept wording.

.4.3 Inventory of SBVR Concepts

The following table lists the SBVR concepts that have corresponding OWL classes that are used in the range of the sbvr-vd:hasConceptType annotation.

Table 10: Inventory of sbvr-vd Concepts

SBVR Concept	OWL Class or Property	Meaning
categorization type	sbvr-vd:categorizationType	OWL class that is an SBVR categorization type.
characteristic types	sbvr-vd:characteristicType	OWL class that is an SBVR characteristic type.
concept	sbvr-vd:concept	SBVR abstract type for any kind of concept; not normally used.
concept type	sbvr-vd:conceptType	SBVR general concept that is a specialization of SBVR 'Concept Type'

SBVR Concept	OWL Class or Property	Meaning
general concept	`sbvr-vd:generalConcept`	Not often used because we map SBVR general concepts to OWL classes.
meaning	`sbvr-vd:meaning`	SBVR abstract type for any kind of meaning, not normally used.
individual concept	`sbvr-vd:individualConcept`	OWL individual that is an SBVR individual concept.
partitive verb concept for object properties	`sbvr-vd:isPartitive`	Mereological object property in which the concept that is the range is "included" in the class that is the domain, where the range is not a data type. Example: time interval$_2$ *includes* time interval$_1$.
partitive verb concept for data types	`sbvr-vd:isPartitiveData`	Mereological data type property in which the concept that is the range is "included" in the class that is the domain, where the range is a data type.
inverse partitive verb concept	`sbvr-vd:isPartitiveReversed`	Inverse of mereological object property. Example: time interval$_1$ *is included by* time interval$_2$.
property association for object properties	`sbvr-vd:has`	Object property in which the concept that is the range is a "quality or trait actually belonging to [the domain class]", and the range is not a data type. Typically identified by the verb *has* or *of*. Example: time interval *has* particular duration.

SBVR Concept	OWL Class or Property	Meaning
property association for data types	`sbvr-vd:hasData`	Object property in which the concept that is the range is a "quality or trait actually belonging to [the domain class]", and the range is a data type. Typically identified by the verb *has* or *of*. Example: time point *has* index, where index is an integer.
inverse property association	`sbvr-vd:has`	Inverse of property association object property Example: particular duration *of* time interval.
set	`sbvr-vd:set`	OWL class that is an SBVR set.
situational role	`sbvr-vd:situationalRole`	General concept that is also a role
unitary concept	`sbvr-vd:unitaryConcept`	OWL class that is an SBVR unitary concept.
verb concept	`sbvr-vd:verbConcept`	OWL class or object or data property that is mapped from an SBVR verb concept. Normally used only to indicate classes formed from *n*-ary verb concepts because the others can be recognized without using an annotation.
verb concept role that is an object property	`sbvr-vd:verbConceptRole`	Identifies OWL object properties that are generated to represent the roles of *n*-ary verb concepts.
verb concept role that is a data type	`sbvr-vd:verbConceptDataRole`	Identifies OWL data type properties that are generated to represent the roles of *n*-ary verb concepts.