ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 3

The Comprehensive Mapping of Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 Ontologies

Jaroslav Karpovič, Gintarė Kriščiūnienė, Linas Ablonskis, Lina Nemuraitė

Kaunas University of Technology, Department of Information Systems
Studentų st. 50-308, LT-51368 Kaunas, Lithuania
e-mail: jaroslav.karpovic@ktu.lt, gintare.kriksciuniene@ktu.lt, linas.ablonskis@ktu.lt, lina.nemuraite@ktu.lt

crossref http://dx.doi.org/10.5755/j01.itc.43.3.6651

Abstract. The goal of the paper is to analyse the subset of Semantics of Business Vocabulary and Business Rules (SBVR) for a comprehensive representation of ontological knowledge defined using the Web Ontology Language OWL 2. SBVR is the OMG metamodel, which separates the representation and meaning of business concepts and business rules, and makes them understandable for business experts as well as for software systems. The SBVR can act as an interface between business participants and semantic technologies, such as OWL 2 that has developed means for describing ontological data and reasoning with them. SBVR provides the richer model for knowledge representation than OWL 2. Though there are a few proposals that have shown that it is possible to transform the significant subset of SBVR concepts into OWL 2 ontology, the suitability of SBVR to represent OWL 2 ontologies has not been studied in detail. The paper addresses the mentioned issue with regards to the transformation from SBVR into OWL 2.

Keywords: Ontology; business concepts; business rules; representation; meaning; SBVR; OWL 2.

1. Introduction

The abundance and complexity of today information systems cause to look for new ways to cope with them. One of the well-known problems remains the miscommunication between business participants and information technology staff, as business experts usually cannot represent their requirements using software models. Moreover, they are not capable to verify business policies and rules implemented in software artefacts. Semantics of Business Vocabulary and Business Rules (SBVR) [1] (and its newest version 1.2 [2], which will be referenced in the rest of the paper) is intended for defining semantics of business vocabulary and business rules in a structured natural language for the exchange of them among organizations and between software tools. Besides other purposes, the SBVR was designed to serve for "transformation of the meanings of concepts and business rules as expressed by humans into forms that are suitable to be processed by tools, and vice versa" [2]. One of such forms is the Web Ontology Language OWL 2 [3], which has developed means for describing ontological data and reasoning with them.

OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. The subset of OWL 2, called OWL 2 Description Logics (DL), is based on the fragment of the first order logic. As a result of this, the description logic's theory and implementation algorithms can be exploited by OWL 2 tools (e.g., editors and reasoners). OWL 2 ontologies are widely used for developing Semantic Web portals, Semantic Web Services, search engines, and enterprise applications. This language is considered the appropriate candidate for checking consistency of SBVR business vocabularies and business rules, which still do not have tools for that purpose.

SBVR, which is based on the first order logic with some extensions to the restricted higher-order logic and modal logic, provides the richer model for knowledge representation than OWL 2, and can be applied for describing various structural and behavioural models, e.g., [4], [5], [6]. SBVR, in contrast with other knowledge modelling approaches, makes a clear distinction between the meaning and representation, providing concepts for expressing the each of them. In SBVR, the concepts (noun concepts and verb concepts), propositions and questions are designed for expressing the meaning; the designations, placeholders, definitions, verb concept wordings and statements are designed for expressing the representations. The meaning can have many representations, and each representation can have different meanings in different contexts.

There are several proposals that have shown that it is possible to transform the significant subset of

SBVR concepts into the OWL 2 ontology [7], [8], [9], [10] (some correspondences between SBVR and OWL are presented in SBVR specification [2]). However, the opposite question (how well OWL 2 features can be described in SBVR) has not been studied in detail despite there are a few on-going works for transforming OWL 2 ontologies into SBVR [11], [12], [13]. We will consider the suitability of SBVR to represent OWL 2 ontologies from the perspective of the SBVR editor (the 1st version of the editor was presented in [4]) and SBVRtoOWL2 transformation (the 1st version of the transformation was presented in [7], [8]).

The paper is structured as follows. Section 2 analyzes SBVR concepts and related works concentrating on SBVR and OWL 2 mutual matching problems. Section 3 presents an illustrative. Section 4 presents specifications of OWL 2 concepts in SBVR Structured Language and several minor extensions to achieve the compliance between the two knowledge models. Finally, Section 5 presents conclusions and future works.

2. SBVR concepts and related works

The SBVR noun concepts can be general concepts (e.g., persons, organizations, products, etc.), roles (e.g., the birth date or name of the general concept "person"), and verb concept roles, played by general concepts (e.g., the aggregate and component are two roles played by the product). The SBVR verb concepts mean relations (fact types) among noun concepts; the individual noun concepts and individual verb concepts allow expressing the facts (propositions) about the real problem domain. The propositions are formulated by closed semantic formulations (i.e., semantic formulations, which have no free variables) – structures of meaning, constructed from logical formulations (modal formulations, logical operations, projections, etc. [2]). As logical formulations may be used as constraints and logical operands in other logical formulations, the SBVR is the most expressive model for expressing knowledge about the domain under consideration.

The terminological knowledge in SBVR is presented by synonyms and synonymous forms, from which the primary (preferred) representations are defined for each meaning. Synonyms usually are modelled in lexical ontologies, representing them as classes related by synonym relation (e.g., synsets in WordNet ontology). Linguistic knowledge in SBVR is expressed by representations: designations and placeholders for noun concepts; verb concept wordings for verb concepts; definitions for concepts, and statements for propositions. Statements have the most complex structure, which allows representing phrases and sentences. But the SBVR does not specify morphological and syntactic characteristics appropriate for complete lexical semantics.

Currently, the complete and explicitly specified SBVR transformation into OWL 2 does not exist, but there are several works devoted for that purpose. The correspondences between SBVR and the previous version of the OWL concepts are presented in the SBVR specification [2] but such mapping is insufficient for OWL 2. SBVR metamodel was created on the base of Object Role Modelling (ORM2), authored by Halpin [14], and works of Business Rule Group. There are ORM2 mappings to Description Logics (DL) [15], [16], and even direct mapping to OWL 2 [17], but the latter is not quite correct. The most exhaustive ORM2 mapping to First Order Logic (FOL) is described by Franconi and Mosca in [18] (generalized in [19]) where the essential theoretical backgrounds are presented that are needed in order to provide a formally consistent translation of ORM2 conceptual scheme into the OWL 2 ontologies. However, the research does not present explicit SBVR to OWL 2 transformations. Besides, the ORM2 and SBVR mappings to OWL 2 are not exactly the same. SBVR has the surface semantics beyond deep semantics [20], which may be formalized in First Order Logic (FOL) or DL. This surface semantics involves both terminological and linguistic concepts. It is desirable to involve that semantics into OWL 2 ontologies for applications of Semantic Web.

The most recent works for transforming SBVR to OWL 2 are presented in Kendall and Linehan [9] and Reynares et al. [10]. The first of them proposes a mapping of SBVR vocabularies to a combination of OWL 2 elements and annotations. The authors state that "the goal of this work is to define a reversible mapping: an SBVR vocabulary can be mapped to OWL 2 and back again without loss of semantic information" though format, syntactic structure, and lexical details "may differ". Kendall and Linehan devote a lot of attention to mapping SBVR Date Time concepts [21] (extension of SBVR) to OWL 2 data types; vocabulary captions; representations (including synonyms and synonymous forms) to annotations. Though they state that the "scope of their work is SBVR vocabularies, excluding behavioural rules", they do not discuss such vocabulary concepts as categorization schemes and segmentations, and consider only a small subset of semantic formulations for business rules. By our opinion, they propose the vague solution for mapping verb concepts to inverse object properties (it remains unclear, how the OWL 2 InverseObjectProperty expression can be related with some of synonymous forms; moreover, the requirement to have the certain inverse object property may be absent in SBVR (i.e., it is not necessary to have the inverse of each object property). There are more inaccuracies, e.g., SBVR exactly-one-quantification does not mean the OWL 2 functional property (see definition in [3]); the verb symbol "has" does not necessary mean the SBVR property association [1], [2]; binary association should not map to the OWL 2 data property.

In contrast, Reynares et al. [10] do not pay any attention to SBVR representations but try to cover the greater part of SBVR concepts and logical formulations. However, their transformations seem immature and weakly illustrated by the presented transformation example, whose results are incorrect as some ontology classes fail into several subsumption hierarchies (e.g., the InternationalCarMovement is the subClassOf of the One-Way_Car_Movement and Thing, etc.). The example has only one individual and no assertions for demonstrating the consistency of the ontology. The concepts of the SBVR categorization scheme and segmentation, roles as well as the SBVR characteristic are misunderstood (the latter is treated as is_property_of fact type in SBVR 1.0 terms though it really is the unary fact type).

The SBVRtoOWL2 transformation, the 1st version of which was proposed in [7], [8], also is focused on the SBVR meaning, though we capture the primary representation of every concept in ontology labels, and one synonymous form for each verb concept wording, which is needed for representing the inverse object property in the OWL 2. Annotation labels are conventional means for expressing naming of classes, object and data properties, and individuals. However, they are insufficient for presenting linguistic knowledge extracted from SBVR vocabularies and rules. There are several possibilities to map SBVR terminological and linguistic information to ontology concepts using simplified [22] or complex [23] linguistic models. The most relevant approach for expressing linguistic, terminological and ontological information is presented by Cimiano et al. [24] where it is stated that linguistic and ontological levels should be clearly separated aiming at clearly associating linguistic information to ontological entities. Such a separation should allow to associate different lexicons with one and the same ontology, to specify the meaning of linguistic constructions with respect to ontologies of various domains, and to support representation of complex lexical entries for multiple languages. The approach overlaps with SBVR in many aspects though SBVR lacks complete linguistic representations. Therefore, we will regard the principle of separation in defining SBVR transformations into OWL 2 by creating separate ontologies for domain concepts (meanings) and representations (the creation of lexical ontologies from SBVR is a special study, which is beyond the scope of the current paper).

For transforming SBVR into OWL 2, we have chosen the OWL 2 metamodel based on its direct semantics [25] for its conceptual clarity. OWL 2 also has RDF based semantics [26] but RDF based metamodel of OWL 2 still does not exist. A mapping from OWL 2 to RDF graphs is defined in [27], and it is possible to convert OWL 2 Functional style ontologies into RDF documents [28] but the reverse is not always true. The RDF format is more flexible, capable representing the OWL 2 FULL, whereas OWL 2 Functional style ontologies are limited to Description Logics (DL). Consequently, our transformations are limited to Description Logics, but we do not consider this as a shortcoming as current ontology reasoners are able to work with OWL 2 DL compatible ontologies only. Therefore, SBVR vocabularies under transformation should conform to Description Logics as well as to regard particularities of OWL 2 ontologies allowing efficient reasoning, maintenance and evolution.

3. Illustrative example

The example of the SBVR business vocabulary of the photo equipment and photography domain is presented as UML class diagram in Fig. 1. The presented vocabulary does not cover the complete domain model but rather serves as the representative example for analysing potentiality of SBVR to represent OWL 2 ontologies. Therefore, some concepts may seem artificially introduced.

In the SBVR metamodel specification [2], UML models are used in two different interpretations: they represent the SBVR vocabulary and the SBVR MOF metamodel at the same time. In Fig. 1, UML classes represent SBVR general concepts; UML associations – SBVR associations; UML aggregations – SBVR partitive verb concepts; UML generalizations – SBVR categorizations; UML incomplete, disjoint generalization sets – SBVR categorization schemes; UML complete, disjoint generalization sets – SBVR segmentations. The generalization hierarchy without specified generalization set means the SBVR categorization (e.g., product is categorized as electronics product, creative product, or other product; such a categorization usually means the permanent hierarchy of primitive, non-derivable concepts). The verb "has" is implied as the verb symbol for verb concepts, represented by unnamed relations, e.g., "product has accessory".

Roles are represented in two ways: as properties (attributes and association ends); or as classes with stereotype <<role>>. Such roles are related with general concepts via the UML generalization symbol with stereotype <<is role of>>. Unnamed association ends (e.g., "product") imply situational roles played by general concepts, e.g., "component is contained in product". The synonymous forms of verb concepts (required for OWL 2 inverse object properties) are represented as separate relations without direction arrows (e.g., "replacing product is for replaceable product" is the synonymous form of the primary form "replaceable product has replacing product". The latter representation is the minor deviation from the OMG recommendation for using UML in the SBVR models [2]. The second deviation is the representation of generalizations between associations as subset constraints, e.g., the association "photo camera contains photo element" subsets the association "product contains component" (the latter was done for better visibility).

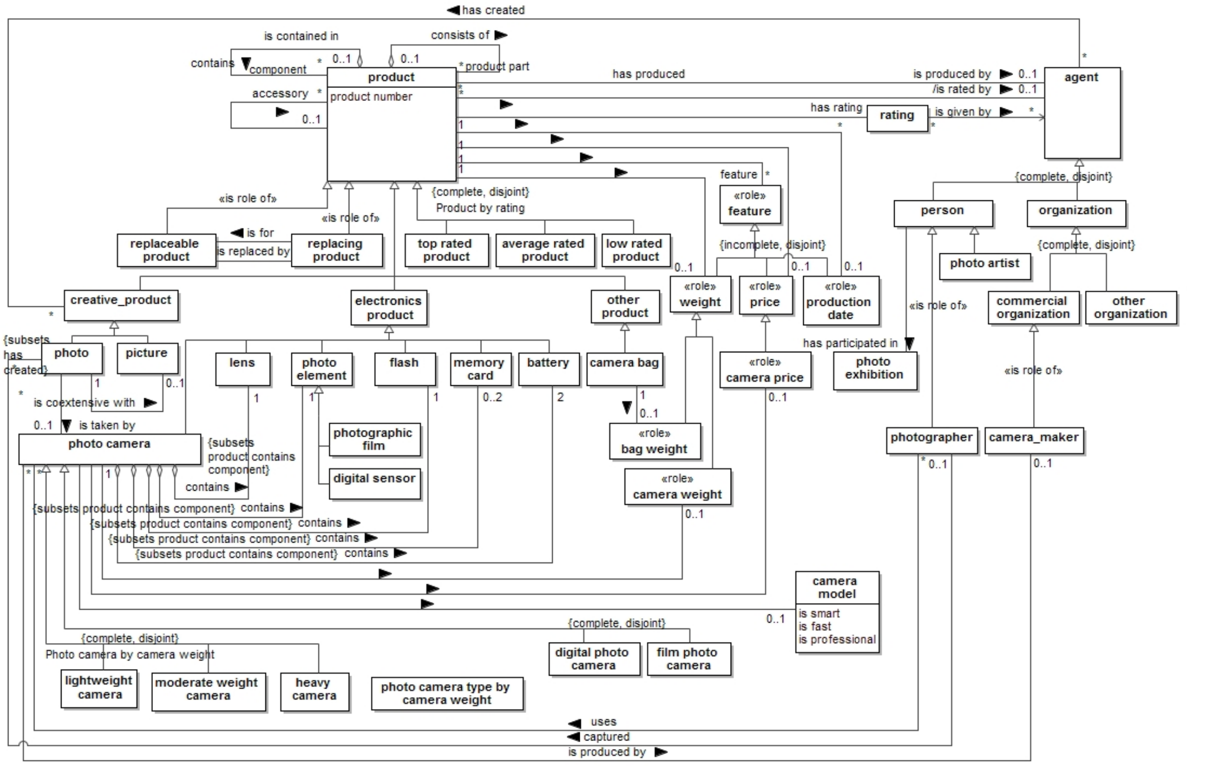

Figure 1. The SBVR business vocabulary for photo equipment and photography presented as UML class diagram

4. Ontology concepts and their representation in SBVR

For SBVR Structured Language specifications, we use the SBVR style of terms, *verbs*, Names and keywords [2], where terms represent noun concepts (general concepts, roles and verb concept roles); verbs represent symbols used in verb concept wordings, meaning verb concepts, and in statements, meaning propositions (facts). Vocabulary entries introduce the primary forms (preferred representations) of SBVR concepts, and can have captioned details, e.g., General concept, Concept type, Synonym or Synonymous form, etc. (all concepts will be explained in the following text).

For OWL 2, we use its abstract syntax [3]. Main concepts of the OWL 2 are axioms and entities (classes, object properties, data properties, data types and named individuals). The ontology concept "Ontology" must be created before all.

OWL 2 Ontology is an instance of the Ontology class from the OWL 2 metamodel. It is the main concept of the ontology and includes all ontology axioms and annotations (Fig. 2).

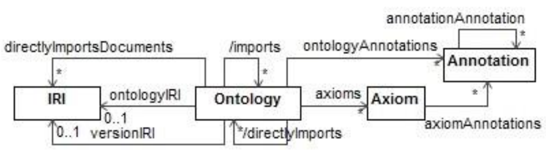

Figure 2. The structure of OWL 2 Ontologies (based on [3])

OWL 2 ontology can be obtained from the SBVR vocabulary, defined as the individual concept of the general concept "vocabulary"; its name, namespace and language can be obtained from the vocabulary name, namespace and language:

SBVR: Vocabulary → OWL: Ontology
SBVR: namespace → OWL: Namespace
SBVR: Namespace URI → OWL: IRI

Example 1:

SBVR: Photo equipment
  General concept: vocabulary
  Language: EN
  Namespace URI:
  http://isd.ktu.lt/SBVR/Photo_equipment
OWL: Ontology
  (http://isd.ktu.lt/OWL/Photo_equipment)

Any ontology is uniquely identified by the ontology IRI (internationalized resource identifier) and each ontology concept is uniquely identified in its namespace. Ontology IRI can be obtained from the SBVR namespace URIs (universal resource identifiers) – the subset of IRIs, which can contain non-Latin alphabet characters. However, in the current practise it is yet desirable to restrict resource identifiers to URIs. Otherwise, it is possible to expand URIs to IRIs.

Ontology can include (import) other ontologies in order to gain access to their entities. SBVR metamodel has a corresponding verb concept 'vocabulary *incorporates* vocabulary'; however, all incorporated vocabularies may comprise a complex hierarchy, which must be accessible during transformation. Therefore, imported ontologies and incorporated vocabularies are left out of scope of the current paper.

OWL 2 Annotations can be used to specify additional information in ontology, e.g., comments. The annotation consists of an annotation property and an annotation value (a literal or Entity IRI). It is possible to transform some of SBVR vocabulary entries such as informal definition, source, note, and example into OWL 2 standard or custom annotations. We use standard annotation property "label" for human readability of entity names in a vocabulary language (e.g., English, as in the example above, or Lithuanian), and the additional annotation property "label_sbvr" dedicated for specifying original entity names in SBVR style that can be useful for the reverse transformation of OWL 2 ontology into SBVR.

SBVR: designation → OWL:Annotation

SBVR: verb concept wording → OWL:Annotation

Example 2:

SBVR: photo_camera
OWL: Declaration(AnnotationProperty(label_sbvr))

AnnotationAssertion(label_sbvr photo_camera "photo_camera"@en)

AnnotationAssertion(rdfs:label photo_camera "photo camera"@en)

Commonly, OWL 2 ontologies use language tagging offered by RDF literals (rdfs:label@en). The SBVR does not define the standard designation of a vocabulary language. For transforming SBVR representations into ontology labels, mapping of language tags should be introduced into the SBVR to OWL 2 transformation. It is possible to assume, e.g., that SBVR uses the same language tags as OWL 2.

4.1. SBVR representation of OWL 2 Entities

OWL 2 Entities define named elements of the OWL 2 ontology. All entities are uniquely identified by their IRIs and must be declared by the Declaration axiom (Fig. 3).

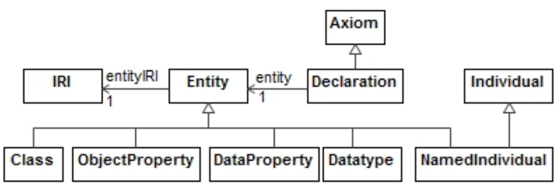

Figure 3. OWL 2 Entities and their Declarations [3]

The OWL 2 Entity IRI can be obtained from the SBVR attributive namespace URI, if it is specified for the concept, or constructed from vocabulary namespace URI:

SBVR: Namespace URI → OWL: EntityIRI

OWL 2 Classes and Individuals. The OWL 2 class C is treated as the set of individuals with the same set of properties. SBVR equivalent to the OWL 2 class is the SBVR 1.2 [2] general concept (object type in the SBVR 1.0 [1]). By default, a single noun or noun phrase, specified in the term style in the entry of the SBVR vocabulary, has the meaning of the general concept if another concept type (e.g., role) is not assigned to it. We use the underscore for constructing names of OWL 2 entities for the consistency with the SBVR style:

SBVR: general concept → OWL: Class

Example 3:

SBVR: photo camera
OWL: Declaration(Class(photo_camera))

OWL 2 Named Individuals (Fig. 3) together with ClassAssertions, assigning these individuals to corresponding classes, can be represented by SBVR individual concepts and their classifications – propositions, meaning that the individual noun concepts are instances of the general noun concepts:

SBVR: individual concept → OWL: NamedIndividual

SBVR: classification → OWL: ClassAssertion

Example 4:

SBVR: Nikon D5100
General concept: photo camera
OWL: Declaration(NamedIndividual(Nikon_D5100))
ClassAssertion(photo_camera Nikon_D5100)

OWL 2 Data Types. OWL 2 has a rich set of data types including RDFS Literals, RDF DataTypes, XSD DataTypes and Plain Literals [3]. SBVR has just a few elementary concepts (text, URI, number, integer, nonnegative integer, positive integer) that can be used for representing the corresponding OWL 2 data types (string, IRI, decimal, integer, nonNegativeInteger, positiveInteger). However, the SBVR allows extensions. SBVR extension for Data and Time [21] defines various extensions of SBVR elementary concepts for representing dates and time durations. In the SBVR2toOWL 2 transformation, we have introduced boolean and date time as the most necessary elementary concepts. Further extensions can be added as necessary.

OWL 2 Object Property defines a particular relationship between two individuals. OWL 2 object properties can be obtained from SBVR binary associations or partitive verb concepts (previously association and partitive fact types [1]).

SBVR: association → OWL: ObjectProperty

Example 5:

```
SBVR: photo is_taken_by photo_camera
OWL: Declaration(ObjectProperty(
  is_taken_by_ photo_camera))
  ObjectPropertyDomain
  (is_taken_by_ photo_camera_photo)
  ObjectPropertyRange(
  is_taken_by_ photo_camera_photo_camera)
  AnnotationAssertion(label_sbvr
  is_taken_by_ photo_camera
  "photo is_taken_by photo_camera"@en)
  AnnotationAssertion(rdfs:label
  is_taken_by_ photo_camera
  "photo is taken by photo camera"@en)
```

For ensuring the uniqueness of object property names, the latters are constructed by concatenating verbs and role names and using two underscores as a separator. Moreover, we preserve the entire primary forms of verb concept wordings in object property labels (otherwise the information about the first verb concept role may be lost). Such a construction allows creating the clear links between SBVR vocabularies and ontologies, which can be useful in several practical cases.

OWL 2 does not include semantics of part-whole relations, thus the latters can be presented in the same way as associations. In order to preserve the meaning of partitive verb concepts, we borrow an idea from Kendall and Linehan [9] and map the SBVR partitive verb concept to the OWL 2 object property, making it the subproperty of the partitive_object_property – the OWL 2 irreflexive object property introduced for preserving the meaning of partitive verb concepts (this rule applies only for partitive verb concepts that do not have more general concept):

```
SBVR: partitive verb concept which does not
have more general concept →
OWL: ObjectProperty, SubObjectPropertyOf
(ObjectProperty Partitive_object_property)
```

OWL 2 Data Properties. Similarly as object properties define properties between two individuals, data properties define relations between individuals and data types. OWL 2 Data Property is a binary relation and could be obtained from the SBVR property association, which is defined by connecting a term, representing the general concept, with the term, representing the role, by the verb symbol "has" and specifying the concept type as property association (otherwise, the verb concept would be interpreted as association). The difference of the DataProperties from the ObjectProperties is that DataPropertyRanges are represented by the SBVR elementary concepts, which are specified as general concepts for roles:

```
SBVR: property association | characteristic →
OWL: DataProperty, DataPropertyDomain,
DataPropertyRange
```

Example 6:

```
SBVR: camera_weight
        General concept: integer
      photo camera has camera weight
        Concept type: property association
OWL: Declaration(DataProperty(camera_weight))
  DataPropertyDomain(camera_weight
  photo_camera)
  DataPropertyRange(camera_weight xsd:integer)
```

Other possibility is to obtain OWL 2 data properties from SBVR characteristics (unary verb concepts), e.g.:

Example 7:

```
SBVR: camera_model is_professional
OWL: Declaration(DataProperty(is_professional))
  DataPropertyDomain(is_professional
    camera_model)
  DataPropertyRange(is_professional
    xsd:boolean)
```

4.2. SBVR representation of OWL 2 Class Axioms and Class Expressions

OWL 2 allows defining relationships between class expressions using class axioms (Fig. 4) [3].

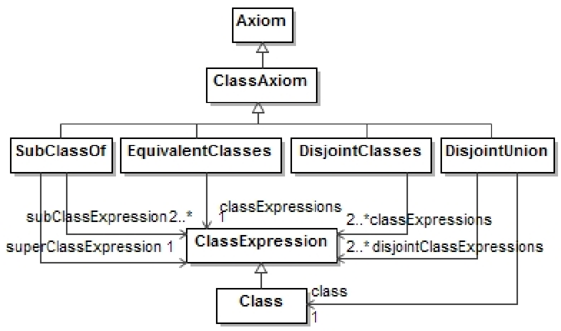

Figure 4. OWL 2 Class Expressions and Class Axioms [3]

SubClassOf class axiom provides possibility to create class specialization hierarchies by defining the subsumption dependency between classes, in which a class CE_1 is the subclass of another, more general, class (or class expression) CE: SubClassOf(CE_1, CE). It means that all individuals of class CE_1 are also the individuals of the class CE. Such an axiom can be constructed using SBVR categorization – proposition that a certain general concept (category) specializes another (more general) concept:

```
SBVR: categorization → OWL: SubClassOf
```

Example 8:

```
SBVR: photo_camera
      digital_photo_camera
        General concept: photo_camera
OWL:SubClassOf(digital_photo_camera
      photo_camera)
```

During transformation of SBVR vocabulary and business rules into OWL 2 ontologies, OWL 2 SubclassOf axioms are formulated along with many other OWL 2 axioms and restrictions (Fig. 5): `AllValuesFrom`, `SomeValuesFrom`, `Object-HasSelf`, `ObjectHasValue`, cardinality restrictions, etc.

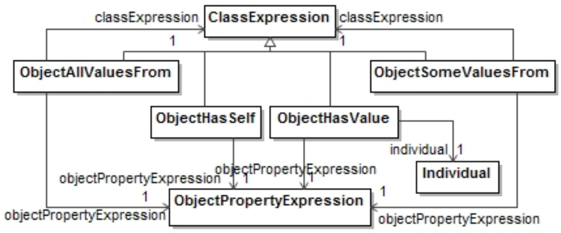

Figure 5. OWL 2 class expressions for restriction of object properties [3]

Class expression **AllValuesFrom,** defining universal quantifications on object properties or data properties, follows from SBVR necessity statements, which use more general verb concepts for defining specific general concepts as players of more general roles:

SBVR: necessity statement → OWL: SubClassOf, ObjectAllValuesFrom

Example 9:

SBVR: It is necessary that digital photo camera *contains* photo element that *is* digital sensor
OWL: SubClassOf(digital_photo_camera ObjectAllValuesFrom(contains_photo_element digital_sensor))

Class expression **SomeValuesFrom**, defining existential quantifiers on object properties or data properties, is derived from SBVR necessity statements solely based on binary verb concepts:

SBVR: necessity_statement → OWL:SubClassOf, ObjectSomeValuesFrom]| DataSomeValuesFrom

Example 10:

SBVR: It is necessary that photo camera *contains* photo element
OWL: SubClassOf(photo_camera ObjectSomeValuesFrom (contains_photo_element photo_element))

Example 11:

SBVR: It is necessary that product *has* product instruction
OWL: SubClassOf(product DataSomeValuesFrom (product_instruction xsd:string))

EquivalentClasses axiom in the OWL 2 ontologies is used to denote the equivalence of class expressions $CE_1, \ldots, CE_n$, $n \geq 2$, which means that CE_1 can be used instead of $CE_2, \ldots, CE_n$ without affecting the meaning of the ontology [3]. Despite the fact that equivalent classes in ontology are treated as synonyms [3], they should not correspond to SBVR synonyms, which are alternative representations of the same meaning. OWL 2 Equivalent Classes axiom for individual classes can be represented using the dedicated SBVR association:

SBVR: association 'concept$_1$ *is_coextensive_with* concept$_2$' → OWL: EquivalentClasses

Example 12:

SBVR: picture
photo
picture *is_coextensive_with* photo
OWL: EquivalentClasses(photo picture)

OWL 2 EquivalentClasses axiom also is implied from the SBVR segmentation statement together with the DisjointUnion axiom (the complete transformation of segmentation is described hereafter in Example 14).

OWL 2 EquivalentClasses axioms between individual classes and class expressions, which define how these classes are derived, can be represented by SBVR definitions:

SBVR: definition → OWL: EquivalentClasses

Example 13:

SBVR: photo artist
General concept: person
Definition: person that *has_photographed* photo and *has_participated_in* photo exhibition
OWL: EquivalentClasses(photo_artist ObjectIntersectionOf(person ObjectSomeValuesFrom (has_photographed_photo photo) ObjectSomeValuesFrom (has_participated_in_photo_exhibition photo_exhibition)))

DisjointClasses and **DisjointUnion axioms**. Class disjointness in the OWL 2 means that an individual `I` can be an instance of the only one class (class expression) CE_i from the set of disjoint classes:

DisjointClasses ($CE_1, \ldots, CE_n$), $1 \leq i \leq n$.

DisjointUnion ($C, CE_1, \ldots, CE_n$), $n \geq 2$, states that a class `C` is the disjoint union of classes $CE_1, \ldots, CE_n$, which are pairwise disjoint.

In some cases, classes that comprise the Disjoint Union in the OWL 2, do not require explicit definition of their disjointness in the SBVR. The OWL 2 Disjoint Union axioms are implied in the SBVR segmentations. The SBVR categorization schemes and segmentations mean alternative specializations of general concepts, and individuals of such specializations should be derivable in OWL 2 ontologies from the non-derivable ones [29], [30], [31], [32]. Thus, the SBVR categorization scheme or segmentation corresponds to the Equivalent Classes axiom defined for the specialization hierarchy, with the reference to the specialized primitive (i.e., non-derivable) OWL 2 class [33]. Derivation rules for derivable specializations can be specified via the SBVR definitions of specialized concepts.

```
SBVR: segmentation → OWL: EquivalentClasses,
SubClassOf, DisjointUnion
```

Example 14:

```
SBVR: camera weight
      Concept type: categorization_type
      Necessity: is_for general_concept
      photo camera
   Camera by weight
     Necessity: segmentation for
     general concept  photo camera that
     subdivides photo camera  by
     camera weight
   lightweight camera
      General concept: photo_camera
      Necessity: is_included_in
      Camera by weight
      moderate weight camera
      General concept: photo_camera
   Necessity: is_included_in
   Camera by weight  heavy camera
      General concept: photo camera
      Necessity: is_included_in
      Camera_by_weight  OWL:
      Declaration(Class(camera_by_weight))
 EquivalentClasses(camera_by_weight
   photo_camera)
   SubClassOf(lightweight_camera
   camera_by_weight)
   SubClassOf(moderate_weight_camera
   camera_by_weight)
   SubClassOf(heavy_camera camera_by_weight)
   DisjointUnion(Camera_by_weight
   lightweight_camera moderate_weight_camera
   heavy_camera)
```

SBVR as well as OWL 2 follow an open world assumption, according which the SBVR concepts as well as OWL 2 classes and properties may not be disjoint if the opposite statement is not defined. In well-formed OWL 2 ontologies, DisjointClasses axioms are required in specialization hierarchies for avoiding ambiguity [29], [30]. The OWL 2 DisjointClasses axioms are not implied in the SBVR categorizations or categorization schemes. Explicit formulations of DisjointClasses axioms can be specified using SBVR impossibility statements, or necessity statements with nor formulations:

```
SBVR: impossibility statement |
necessity statement with nor formulation →
OWL: DisjointClasses
```

Example 15:

```
SBVR: It is impossible that electronics product
is creative product or other product
```

or

```
SBVR: It is necessary that electronics product
      is nor creative product nor
      other product
OWL: DisjointClasses(creative_product
     electronics_product other_product)
```

DisjointUnionOf can be explicitly specified by the SBVR disjunction accompanied with impossibility statement or nor formulation. The OWL 2 DisjointUnionOf equivalent (consisting of ObjectUnionOf and DisjointClasses axioms) will be generated.

Example 16:

```
SBVR: photo element
        Definition: photo element is
          photographic film or photo element is
          digital sensor
        It is impossible that digital sensor is
     photographic film
OWL: EquivalentClasses(photo_element
  ObjectUnionOf(photographic_film
  digital_sensor))
  DisjointClasses(digital_sensor
  photographic_film)
```

The **ObjectUnionOf** as well as **ObjectIntersectionOf** and **ObjectComplementOf** class expressions (Fig. 6) can be explicitly specified by SBVR logical operations with the closed logical formulations.

```
SBVR: disjunction → OWL: ObjectUnionOf
SBVR: conjunction → OWL: ObjectIntersectionOf
```

(see Example 13, Example 16)

```
SBVR: logical negation →
  OWL: ObjectComplementOf
```

Example 17:

```
SBVR: photographic_film
        Definition: photo element that is not
        digital sensor
OWL: EquivalentClasses(photographic_film
  ObjectIntersectionOf (ObjectComplementOf
  (digital_sensor) photo_element)
```

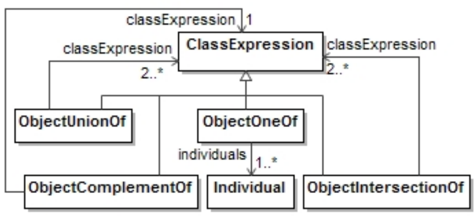

Figure 6. OWL 2 class expressions for logical operations and enumerations of individuals [3]

ObjectOneOf expression allows specifying the OWL 2 enumeration of individuals (e.g., in class definitions):

```
SBVR: definition with conjunction of
individuals
→ OWL: ObjectOneOf
```

Example 18:

SBVR: rating
Definition: Top rating or Average Rating or Low rating
OWL: EquivalentClasses(rating ObjectOneOf (Low_rating Top_rating Average_rating))

The OWL 2 ObjectHasValue class expression allows expressing object properties of individuals, i.e., their connections to other individuals. In the SBVR, such an expression can be specified as a fact, based on the verb concept, in which one role is played by an individual verb concept (usually, such expressions are used for definitions):

SBVR: fact with one Individual concept →
OWL: ObjectHasValue

Example 19:

SBVR: top_rated_product
General concept: product by rating
Definition: product by rating and *has* rating Top rating
OWL: EquivalentClasses(top_rated_product ObjectIntersectionOf (ObjectHasValue (has rating Top_rating) product_by_rating))

4.3. SBVR representation of OWL 2 Object Property Expressions and Object Property Axioms

The OWL 2 object property axioms characterize and establish relationships between object property expressions. Some of these axioms are presented in Fig. 7.

OWL 2 Inverse Object Properties axiom denotes that two object properties OP_1 and OP_2 are pair-wise inverse, i.e., they mean a bidirectional relationship between two class expressions. An Inverse Object Property corresponds to a synonymous form of a verb concept wording, which has the inverse order of verb concept roles in comparison with the primary form of the verb concept. However, the SBVR verb concept wording can have several synonymous forms with an inverse or direct order of verb concept roles. Therefore, there is a problem to denote the inverse verb concept wordings, which are desirable for representing the OWL 2 inverse object properties.

Kendall and Linehan [9] make an assumption that it makes sense to create the inverse object property for each SBVR binary verb concept that also is an association; and that is no sense for doing this for partitive verb concepts. We do not agree with such assumption.

In order to synchronize OWL 2 ontologies and SBVR vocabularies, we create inverse object properties for binary associations and partitive verb concepts if they have synonymous forms with the inverse order of roles. Also, we introduce the concept type inverse verb concept for denoting

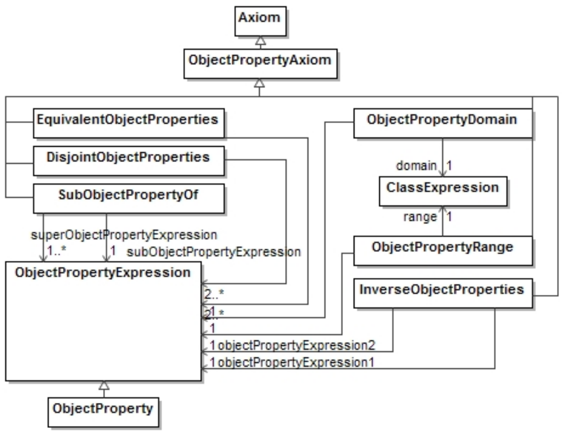

Figure 7. The subset of OWL 2 Object Property Expressions and Object Property Axioms [3]

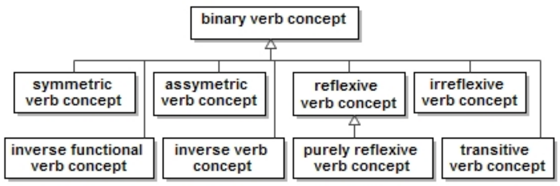

Figure 8. Categories of the SBVR binary verb concept for describing characteristics of the OWL 2 object properties and some other expressions

the synonymous form (Fig. 8), which should be transformed into the inverse object property.

Such an extension does not change the SBVR metamodel but is included in the SBVR Extensions for OWL 2 Vocabulary, which is used as the incorporated vocabulary in SBVR to OWL 2 transformations. The extension was introduced in the 2nd version of SBVR-toOWL 2 transformation, as it allows replacing the necessity statement, which was used in the SBVR-toOWL 2 version [8], for the sake of simplicity. The similar extensions (Fig. 8) were introduced for the OWL 2 characteristics of object properties and ObjectHasSelf expressions, which will be explained in the following text. Thus the OWL 2 Inverse Object Properties axiom can be specified in the SBVR:

SBVR: verb concept, inverse verb concept →
OWL: ObjectProperty, InverseObjectProperties

Example 20:

SBVR: photographer *uses* photo camera
photo camera *is_used by* photographer
Concept type: inverse verb concept
See: photographer *uses* photo camera
OWL: InverseObjectProperties (is_used_by_photographer uses_ photo_camera)

SubObjectProperty axiom is similar to the SubclassOf axiom and allows defining specialization hierarchies for verb concepts:

SBVR: categorization → OWL: SubObjectProperty

Example 21:

SBVR: photo_camera *contains* photo_element
General concept: product *contains* component
OWL: SubObjectProperty(contains photo_element contains component)

DisjointObjectProperties. In well-formed OWL 2 ontologies, the SubObjectProperty axiom requires disjointness of object properties, specializing the same more general object property (and, consequently, disjointness of classes, which define ranges of these object properties). Similarly to DisjointClasses, explicit formulations of DisjointObjectProperties axioms can be specified using SBVR impossibility statements, or necessity statements with nor formulations:

SBVR: impossibility statement for verb concepts, impossibility_statement for general concepts →
OWL: DisjointObjectProperties, DisjointClasses

Example 22:

SBVR: It is impossible that concept
'photo_camera *contains* battery' *is* concept
'photo_camera *contains* lens' or *is* concept
'photo_camera *contains* flash' or *is* concept
'photo_camera *contains* memory_card' or
is concept 'photo_camera *contains* photo element
It is impossible that battery *is* lens or *is*
flash or *is* memory_card or *is* photo_element
OWL: DisjointObjectProperties(contains_battery contains_ flash contains__lens contains_memory_card contains_photo_element)
DisjointClasses(battery flash lens memory_card photo_camera photo_element)

EquivalentObjectProperty axiom is similar to EquivalentClasses axiom and can be specified in SBVR:

SBVR: association 'concept *is_coextensive_with* concept' →
OWL: EquivalentObjectProperties

Example 23:

SBVR: concept 'product *contains* component'
is_coextensive_with concept 'product *consists_of* product part'
OWL: EquivalentObjectProperties (consists_of product_part contains_component)

ObjectPropertyChain is the more complex SubObjectPropertyAxiom of the form:

SubObjectPropertyOf (ObjectPropertyChain(OPE_1 ... OPE_n) OPE)

Such an axiom allows to derive object property assertions by stating that if an individual I_1 is connected by a chain of object property expressions OPE_1, ..., OPE_n with an individual I_2, then I_1 is also connected with I_2 by the object property expression OPE [3]. ObjectPropertyChain can be expressed by the SBVR necessity statement formulated by the implication formulation, which has the antecedent restricted by one or more projecting formulations, and the second role of its consequent coincides with the second verb concept role of the last verb concept in the projecting formulations' chain.

SBVR: necessity statement with implication formulation and projecting formulation chain →
OWL: ObjectPropertyChain

Example 24:

SBVR: It is necessary that product *is_rated_by* agent if product *has* rating that *is_given_by* agent
OWL: SubObjectPropertyOf (ObjectPropertyChain (has_rating is_given_by agent) is_rated_by agent)

ObjectHasSelf axiom allows specifying the object property that is pure reflexive in ORM2 terms [34], e.g., the replaceable product can be replaced by some other products including itself (please take the attention that the replaceable product is the first verb concept role of the verb concept; the role is played by the product and is captured by an annotation and inverse object property, if any).

SBVR: purely_reflexive_verb_concept →
OWL: ObjectHasSelf

Example 25:

SBVR: replaceable_product *has* replacing_product
Concept type: purely reflexive verb concept
OWL: SubClassOf(product ObjectHasSelf (has_ replacing_product))

Properties of OWL 2 Object Properties. OWL 2 Object Properties can have properties themselves. Some of these properties as Functional Object Property can be specified by the SBVR quantification that means that an instance of the general concept (class C_1) can be connected to at most one instance of the general concept (class C_2) using the certain verb concept (functional object property OP):

SBVR: at_most_one_quantification →
OWL: FunctionalObjectProperty

Example 26:

SBVR: It is necessary that photo_camera *has* at_most 1 camera_model
OWL: FunctionalObjectProperty(has_camera_model)

The **Inverse Functional Property** cannot be directly specified in SBVR. Besides this, there is a set of OWL 2 Object Property Axioms that come from ORM2 [14] and are important for inference:

Reflexive, **Irreflexive**, **Symmetric**, **Assymetric**, and **Transitive** Object Properties that do not have corresponding characteristics in SBVR metamodel, though the latter also is based on ORM2. For solving this problem, we propose to specialize SBVR binary verb concept similarly as in the case of inverse object property (Figure 8), e.g.:

```
SBVR: transitive verb concept
      Concept type: binary_verb_concept
```

Later, concepts of the SBVR Extensions for OWL 2 vocabulary can be reused for various domain vocabularies as concept types themselves. Transformation of SBVR verb concepts into characteristics of OWL 2 object properties is straightforward, e.g.:

```
SBVR: transitive verb concept →
OWL: TransitiveObjectProperty
```

Example 27:

```
SBVR: product consists_of component
       Concept type: transitive verb concept
OWL: TransitiveObjectProperty
    (consists_of_component)
```

The current solution simplifies the previous representation of object property characteristics by necessity statements with implication formulations used in the first version of SBVRToOWL2 transformation [8].

OWL 2 Cardinality Restrictions. OWL 2 ObjectMinCardinality, ObjectMaxCardinality, ObjectExactCardinality expressions have their direct equivalents in SBVR, whose representations have been described by us [7], [8] and other authors [9], [10]. Therefore, we do not focus on them in the current paper.

It is worth to mention that ObjectProperty axioms must satisfy the restrictions specified in [3] in order to guarantee decidability of the basic reasoning problems for OWL 2 DL: 1) ObjectMinCardinality, ObjectMaxCardinality, ObjectExactCardinality, ObjectHasSelf, FunctionalObjectProperty, InverseFunctionalObjectProperty, IrreflexiveObjectProperty, AsymmetricObjectProperty, and DisjointObjectProperties must be simple properties (i.e., they must have no direct or indirect subproperties that are either transitive or are defined by means of property chains); 2) they must satisfy the restrictions on the property hierarchy for avoiding cyclic dependencies. E.g., it is not allowed to specify an object property that is both transitive and has the cardinality restriction. Such constraints impose restrictions on SBVR vocabularies and business rules for creating or representing OWL 2 ontologies. Some of such requirements were presented in [8].

4.4. OWL 2 Data Property axioms and restrictions

The OWL 2 DataProperty axioms and restrictions are similar to ObjectProperty axioms and restrictions: **DataPropertyDomain** and **DataPropertyRange**, **Functional**, **Equivalent**, **Disjoint** data properties; **SubDataProperties** that can comprise data property hierarches; **DataMinCardinality**, **DataMaxCardinality** and **DataExactCardinality** restrictions (except **DataExactCardinality** '1' restrictions on DataProperties having `DataRange 'boolean'`; which are transformed to verb concepts of the type '`concept` *`incorporates`* `characteristic`'). We will not present here specifications analogous to these described in the previous sections. Instead we will present some peculiarities that DataProperties have.

One of such peculiarities is the fact that the **DataPropertyRange** is represented in SBVR as the elementary concept, which generalizes the role, specifying the DataProperty (Examples 6 and 7). Therefore, SBVR to OWL 2 transformation must take this into account and to distinguish, which concept is for actual generalization of OWL 2 data properties, and which concept is for the data property range. In practise, it is not a problem as the equivalents of data property ranges are specified in SBVR as elementary concepts.

The **DataIntersectionOf**, **DataUnionOf**, and **DataComplementOf** data ranges can be specified in SBVR as logical operations with elementary concepts (similarly as for object properties in Examples 13, 16 and 17). **DataOneOf** expression allows specifying the enumeration of data values (similarly as for object properties in Example 18). The **DatatypeRestriction** restricts the value space of a data type by a constraining facet. This restriction is different from object property restrictions and can be expressed in SBVR by the "restrictive" verb concept, constructed from 1) the first verb concept role, corresponding to OWL 2 data property; the verb symbol *`is_equal_to`*, *`is_greater_than`*, *`is_less_than`*, *`is_not_greater_than`*, *`is_not_less_than`*, and the second role, played by the elementary concept, corresponding to OWL 2 Datatype:

```
SBVR: restricting verb_concept →
  OWL: DataTypeRestriction
```

DataTypeRestrictions usually are used in class expressions, having DataPropertyRestrictions (**DataAll ValuesFrom**, **DataSomeValuesFrom**, **DataHas Value**), which are similar to the ObjectProperty restrictions. The **DataHasValue** class expression allows specifying the particular value of DataProperty expression. The main difference between ObjectPropertyRestrictions and DataPropertyRestrictions is that the class expressions for existential and universal quantification can be formulated over a list of data property expressions thus allowing value comparesons. OWL 2 data property restrictions with datatype restrictions can be specified in SBVR definitions:

```
SBVR: definition → OWL:
DataPropertyRestriction
```

Example 28:

```
SBVR: lightweight_camera
      General_concept: photo camera
      Definition: photo camera that has
     camera weight less_than 200 and has
     camera weight greater_than 0
OWL: EquivalentClasses(lightweight_camera
  ObjectIntersectionOf (DataSomeValuesFrom
  (camera_weight DatatypeRestriction(xsd:int
  xsd:maxInclusive "200"^^xsd:integer
  xsd:minInclusive "0"^^xsd:integer))
  camera_by_weight))
```

OWL 2 **HasKey** axioms allow defining the keys, consisting of the sets of object properties and data properties, uniquely identifying named individuals in ontology. It is not required that OWL 2 key properties were functional though they can be defined as functional. HasKey axioms in OWL 2 are DL Safe and allow inferring that two individuals are the same if their key values coincide.

The HasKey axiom corresponds to the reference scheme in SBVR, which is defined as the "chosen way of identifying instances of a given concept" [2], and consists of intentionally used verb concept roles, extensionally used verb concept roles, and characteristics:

```
SBVR: reference_scheme → OWL: HasKey
```

Example 29:

```
SBVR: photo_camera
      Reference scheme: has camera model
      is_produced_by camera_maker
      has product number
OWL: HasKey(photo_camera(has_camera_model
  is_produced_by_camera_maker)(product_number))
```

4.5. Representation of OWL 2 assertions

OWL 2 assertions about individuals correspond to SBVR propositions, which include individual noun concepts and individual verb concepts among other concepts. The ClassAssertion axiom, already discussed in Section 4.2, defines the individual being an instance of the class (e.g., as in Example 4) and corresponds to the classification in SBVR.

The OWL 2 **SameIndividual** (**DifferentIndividuals**) assertions state that several individuals are all equal (all different) to each other. In SBVR, SameIndividuals assertions are formulated by necessity statements:

```
SBVR: necessity_statement →OWL: SameIndividuals
```

Example 30:

```
SBVR: It is necessary that G. Gudas
is Gytis Gudas
OWL: SameIndividual(G.Gudas Gytis_Gudas)
```

In SBVR, DifferentIndividuals assertions are formulated similarly as DisjointClasses and Disjoint ObjectProperties axioms:

```
SBVR: impossibility statement |
necessity statement with
nor formulation →
OWL: DifferentIndividuals
```

Example 31:

```
SBVR: It is impossible that camera_model
  COOLPIX P600 is camera_model COOLPIX P340
OWL: DifferentIndividuals(COOLPIX_P600
  COOLPIX_P340)
```

OWL 2 **ObjectPropertyAssertions** and **DataPropertyAssertions** allow specifying relations between individuals and assigning data property values to them. Such assertions correspond to SBVR propositions:

Example 32:

```
SBVR: Nikon 112
        General concept: digital photo camera
      Nikon 112 has camera_model COOLPIX P600
      Nikon 112 has product_number 112
OWL: Declaration(NamedIndividual(Nikon_112))
  ClassAssertion(digital_photo_camera
  Nikon_112)
  ObjectPropertyAssertion(has_camera_model
  Nikon_112 COOLPIX_P600)
  DataPropertyAssertion(product_number
  Nikon_112 "112"^^xsd:string)
```

OWL 2 **NegativeObjectPropertyAssertions** and **NegativeDataPropertyAssertions** allow to explicitly declare that certain individuals are not related with other individuals by specified properties. In SBVR, such assertions can be made by using impossibility statements or necessity statements with negations:

```
SBVR: impossibility statement|necessity statement
with logical negation →
OWL: NegativeObjectPropertyAssertion |
NegativeDataPropertyAssertions
```

Example 33:

```
SBVR: It is impossible that digital camera
  Nikon 112 contains photographic film
  PF Kodak 0012
     It is necessary that not Nikon 112 has
   product number 151
OWL: NegativeObjectPropertyAssertion
  (contains_ photographic_film Nikon_112
  PF_Kodak_0012)
  NegativeDataPropertyAssertion
  (product number Nikon_112
  "151"^^xsd:string)
```

5, Conclusions and future works

The paper presents the consistent analysis of SBVR capabilities to represent OWL 2 ontologies with regards to the SBVRtoOWL2 transformation, which is implemented and tested for several domains. It was shown that the SBVR with minor extensions can serve as a human-friendly interface to OWL 2 ontologies; so it can be used in semantic search and other ontology-based applications including ontology development. Such tasks would require the careful attention for describing SBVR vocabularies in order to avoid inconsistent ontologies that can be raised by specifying impermissible combinations of SBVR business rules leading to incompatibility of OWL 2 axioms.

The issues addressed in the paper are also important to the solution of the reverse problem – creation of SBVR vocabularies from existing ontologies. As one can see, it would require some manual intervention for introducing missing labels or supplementing object properties with semantics of part-whole relations. Moreover, it would be impossible to transform OWL 2 object properties without domains and ranges specified. Sometimes, domains and ranges may be inferred from property subsumption hierarchies or inverse object properties. Also, we have not considered yet complex domain and range specifications and imported ontologies that would require additional efforts as well as methodology for creation of domain specific lexical ontologies from SBVR representations.

Nevertheless, to our knowledge, our SBVR to OWL 2 mapping is the most comprehensive in comparison with currently existing approaches [9], [10]. The demonstrative prototype of transformations implemented in ATL transformation language can be found at http://s2o.isd.ktu.lt/.

Acknowledgements

The work is supported by the project VP1-3.1-ŠMM-10-V-02-008 „Integration of Business Processes and Business Rules on the Basis of Business Semantics" (2013–2015), which is funded by the European Social Fund (ESF).

References

[1] **OMG, 2008.** *Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.0.* OMG Document Number: formal/2008-01-02, 2008.

[2] **OMG, 2013.** *Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.2.* OMG Document Number: formal/2013-11-05, 2013.

[3] **W3C, 2012a.** *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)* W3C Recommendation 11 December 2012, 2012.

[4] **L. Nemuraite, T. Skersys, A. Sukys, E. Sinkevičius, L. Ablonskis.** VETIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models. In: *Information Technologies' 2010: proceedings of the 16th international conference on Information and Software Technologies, IT 2010, Kaunas, Lithuania, April 21-23*, 2010, pp. 377-384.

[5] **A. Morkevičius, S. Gudas.** Enterprise Knowledge Based Software Requirements Elicitation. *Information Technology and Control,* 2011, Vol. 40, No. 3, 181-190.

[6] **T. Skersys, L. Tutkute, R. Butleris, R. Butkienė.** Extending BPMN Business Process Model with SBVR Business Vocabulary and Rules. *Information Technology and Control,* 2012, Vol. 41, No. 4, 356-367.

[7] **J. Karpovic, L. Nemuraite.** Transforming SBVR business semantics into Web ontology language OWL2: main concepts. In: *Information Technologies' 2011: Proceedings of the 17th international conference on Information and Software Technologies, IT 2011, Kaunas, Lithuania, April 27-29,* 2011, pp. 231-238.

[8] **J. Karpovic, L. Nemuraite**, **M. Stankeviciene.** Requirements for semantic business vocabularies and rules for transforming them into consistent OWL2 ontologies. In: *Information and software technologies: 18th International Conference, ICIST 2012, Kaunas, Lithuania, September 13-14, 2012.* Berlin, Heidelberg: Springer, 2012, pp. 420-435.

[9] **E. Kendall, M.H. Linehan.** Mapping SBVR to OWL2. *IBM Research Report, RC25363 (WAT1303-040) March 16, 2013*.

[10] **E. Reynares, M.L. Caliusco, M.R. Galli.** An Automatable Approach for SBVR to OWL 2 Mappings. In: *XVI Ibero-American Conference on Software Engineering* CIbSE 2013, Montevideo, Uruguay, 2013, pp. 201-214.

[11] **G. Bernotaityte, L. Nemuraite, R. Butkiene, B. Paradauskas.** Developing SBVR vocabularies and business rules from OWL2 ontologies. In: *Information and Software Technologies: 19th international conference, ICIST 2013, Kaunas, Lithuania, October 10-11, 2013*. CCIS 403, Springer: Berlin, Heidelberg, 2013, pp. 134-145.

[12] **F. Gailly.** Transforming Enterprise Ontologies into SBVR formalizations. In: *Proceedings of the CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering (CAiSE),* Valencia, Spain, June 20th, 2013, pp. 122-129.

[13] **E. Reynares, M. L. Caliusco, M. R. Galli.** Approaching the feasibility of SBVR as modeling language for ontology development: An exploratory experiment. *Expert Systems with Applications,* 2014, Vol. 41, No. 4, 1576-1583.

[14] **T. Halpin.** *ORM 2 Graphical Notation*. Neumont University, Technical Report ORM2-01, September 2005.

[15] **M. Jarrar.** Towards automated reasoning on ORM schemes – mapping ORM into the DLRifd description logic. In: *Proceedings of the 26th International Conference on Conceptual Modeling (ER 2007),* LNCS 4801, 2007, pp. 181-197.

[16] **C. M. Keet.** *Mapping the object-role modeling language ORM2 into description logic language DLR.* Technical Report KRDB07-2, KRDB Research

Centre, Faculty of ComputerScience, Free University of Bozen-Bolzano, 2007.

[17] **H. M. Wagih, D. S. ElZanfaly, M. M. Kouta.** Mapping Object Role Modeling 2 schemes to OWL2 ontologies. In: *Proceedings of the 3rd IEEE International Conference on Computer Research and Development (ICCRD), Shanghai, China, March 11-13, 2011*, Vol. 4, IEEE Press, pp. 126-132.

[18] **E. Franconi, A. Mosca.** *Reasoning on ORM2 Conceptual Schema: Theoretical Backgrounds*. EU-IST Integrated Project (IP) 2009-231875, 2009, pp. 1-41.

[19] **E. Franconi, A. Mosca, D. Solomakhin.** ORM2: Formalization and Encoding in OWL2. In: *On the Move to Meaningful Internet Systems: OTM 2012 Workshops.* LNCS 7567, 2012, pp. 368-378.

[20] **A. H. M. t. Hofstede, H. A. Proper.** How to formalize it? Formalization Principles for Information Systems Development Methods. *Information and Software Technology*, 1998, Vol. 40, No. 10, 519-540.

[21] **OMG, 2011**. *Date-Time Vocabulary (Date-Time)*. Version 1.0. OMG Document bmi/2011-08-01.

[22] **M. Kleiner, P. Albert, J. Bézivin.** Parsing SBVR-Based Controlled Languages. In: *Schürr, A., Selic, B. (Eds.) Models 2009*, LNCS 5795, Springer-Verlag Berlin Heidelberg, 2009, 122-136.

[23] **P. Buitelaar, P. Cimiano, P. Haase, M. Sintek.** Towards Linguistically Grounded Ontologies. In: *Aroyo, L. et al. (Eds.) ESWC 2009, LNCS 5554, Springer-Verlag Berlin Heidelberg*, 2009, pp. 111-125.

[24] **P. Cimiano, P. Buitelaar, J. McCrae, M. Sintek.** LexInfo: A declarative model for the lexicon-ontology interface. *Web Semantics: Science, Services and Agents on the World Wide Web 9*, 2011, pp. 29-51.

[25] **W3C 2012b**. **B. Motik, P. F. Patel-Shnaider, B. C. Grau** (Eds.) *OWL 2 Web Ontology Language Direct Semantics (Second Edition).* W3C Recommendation, 11 December, 2012.

[26] **W3C, 2012c**. **P. F. Patel-Schneider (Ed.)** *OWL 2 Web Ontology Language RDF-Based Semantics (Second Edition).* W3C Recommendation, 11 December, 2012.

[27] **W3C, 2012d. M. Schneider, B. Motik** (Eds.) *OWL 2 Web Ontology Language Mapping to RDF Graphs.* W3C Recommendation, 11 December, 2012.

[28] **The University of Manchester.** OWL Syntax Converter. http://owl.cs.manchester.ac.uk/ converter.

[29] **A. L. Rector.** Normalisation of ontology implementations: Towards modularity, re-use, and maintainability. In: *Proceedings Workshop on Ontologies for Multiagent Systems (OMAS) in conjunction with European Knowledge Acquisition Workshop*, 2002, pp. 1–16.

[30] **A. L. Rector.** Modularisation of domain ontologies implemented in description logics and related formalisms including owl. In: *Genari, J. (Ed.): Proceedings of the 2nd International Conference on Knowledge Capture K−CAP 03*, ACM, 2003, pp. 121–128.

[31] **A. L. Rector, N., Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, C. Wroe.** OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. In: *Engineering Knowledge in the Age of the Semantic Web*, LNCS 3257, 2004, pp. 63-81.

[32] **L. Nemuraitė, B. Paradauskas.** A methodology for engineering OWL 2 ontologies in practise considering their semantic normalisation and completeness. *Electronics and Electrical Engineering*, 2012, Vol. 4, No. 120, 89-94.

[33] **T. R. Gruber.** *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Technical report, KSL-93-04. Knowledge Systems Laboratory, Stanford University, 1993, pp. 1-23.

[34] **T. Halpin, M. Curland.** Enriched Support for Ring Constraints. In: *On the Move to Meaningful Internet Systems: OTM 2011 Workshops, LNCS, 7046, Springer: Berlin, Heidelberg*, 2011, pp. 309-318.

Received March 2014.