An Automatable Approach for SBVR to OWL 2 Mappings

Emiliano Reynares¹, Ma. Laura Caliusco¹, and Ma. Rosa Galli¹,²

¹ CIDISI Research Center, UTN Santa Fe, B. Lavaysse 610, S3004EBW Santa Fe, Argentina
{ereynares,mcaliusc}@frsf.utn.edu.ar
² INGAR-UTN-CONICET, Avellaneda 3657, S3002GJC Santa Fe, Argentina
{mrgalli}@santafe-conicet.gov.ar

Abstract. The Semantics of Business Vocabulary and Business Rules (SBVR) has been conceived by OMG to provide business people a way to semantically describe business concepts and specify business rules. However, benefits of having a mean of mapping SBVR expressions to ontology statements have already been recognized. Some of the most important applications of such mappings are: (1) using of ontology reasoners to prove the consistency of business domain information, (2) generation of an ontology intended to be used in the analysis stage of a software development process, and (3) the possibility of encapsulate the declarative specification of business knowledge into information software systems by means of an implemented ontology. Although previous proposals have presented some transformations, an exhaustive and automatable approach for them still is lacking. This work presents a broad and detailed set of transformations that allows the automatable generation of an OWL 2 ontology from the SBVR specifications of a business domain. Such transformations are rooted on the structural specification of both standards and are depicted along the paper through a case study.

Keywords: business rule, ontology, mapping, SBVR, OWL 2

1 Introduction

The Semantics of Business Vocabulary and Business Rules (SBVR) provides business people a linguistic way to semantically describe business concepts and specify business rules [10]. SBVR has been conceptualized for business people and designed to be used for business purposes independent of information systems designs. The linguistic approach adopted by the proposal enables the expression of business knowledge through statements rather than diagrams. That is rooted in the insight that diagrams are helpful for depicting structural organization of concepts but they are impractical as a primary means of defining vocabularies and expressing business rules.

Despite the empowerment such language grants to business people, several works
have already recognized the benefits of having a mean of mapping SBVR expres-
sions to ontology statements. Important applications of such mappings can be
mentioned. For example, ontology reasoners could be used to automatically prove
the consistency of business models [4][8][1][6]. Ontologies intended to be used in
the analysis stage of a software development process could be generated from
main business knowledge sources [3]. The mappings could also be used to gen-
erate ontologies that encapsulate business knowledge into information software
systems, enabling unambiguous representation of knowledge and efficient man-
agement of highly dynamic environments [5][14][12]. Such applications highlight
the central role semantic technologies will have on future enterprise information
systems.

Although previous proposals have presented some SBVR to OWL 2 transforma-
tions, an exhaustive and automatable approach for them still is lacking. This
work presents a broad and detailed set of transformations that allows the au-
tomatable generation of an OWL 2 ontology from the SBVR specifications of a
business domain. Transformations are rooted on the structural specification of
both standards and are depicted along the paper through a case study. OWL 2
Web Ontology Language (OWL 2) has been selected as the receipt language of
the transformations because it has evolved as a de-facto standard for a broad
spectrum of applications [17].

The rest of this paper is organized as follow. Section 2 and Section 3 provides an
overview of the SBVR and OWL 2 specifications, respectively. Section 4 presents
the SBVR to OWL 2 mappings and illustrate them through a case study. Sec-
tion 5 analyses the differences with previous works. Finally, some conclusions
and future research directions are presented in Section 6.

2 SBVR Overview

SBVR defines the vocabulary and rules for documenting the semantics of busi-
ness vocabularies, business facts, and business rules; which allows their verbal-
ization in a controlled vocabulary readily understandable by business people.
The fact-oriented approach of SBVR stems from the Business Rules Manifesto³,
stating that rules builds on facts, and facts build on concepts as expressed by
terms. Therefore, terms express business concepts, facts make assertions about
these concepts, and rules constrain and support these facts. SBVR supports such
approach by providing noun concepts and verb concepts respectively correspond-
ing to the notions of terms and facts. Figure 1 shows the structural organization
of such components.

A noun concept is a concept that is the meaning of a noun or noun phrase, which
is specialized by: (1) object types, which are noun concepts classifying things on
the basis of their common properties; (2) individual concepts, which are concept
corresponding to only one object thing and (3) roles, which are noun concepts
corresponding to things based on their playing a part, assuming a function or
being used in some situation. Additionally, fact type roles are defined as those

³ http://www.businessrulesgroup.org/brmanifesto.htm
roles that specifically characterizes its instances by their involvement in an instance of a given fact type.

A *verb concept* - also named *fact type* - is a concept that is the meaning of a verb phrase that involves one or more noun concepts, representing unary, binary or n-ary relations.

![Diagram](image)

**Fig. 1.** Noun and verb concepts in SBVR

Finally, a SBVR rule is an element of guidance that introduces an obligation or necessity and distinguishing two general types: (1) *structural rules*, which describe the way the business chooses to organize the things it deals with; and (2) *operative rules*, which govern the conduct of business activity by describing business processes. Both types of rules are built imposing restrictions over fact types by using quantifiers, logical operators, etc. Figure 2 shows the structural organization of quantifications and logical operations.

As early stated, SBVR adopts a linguistic approach that allows to define vocabularies and express operative rules. According to this insight, SBVR defines a Controlled Natural Language (CNL) - named SBVR Structured English - and describes the way to mechanically mapping such CNL expressions to SBVR formal concepts.

### 3 OWL 2 Overview

The OWL 2 Web Ontology Language (OWL 2) is the latest version of an ontology language proposed by the World Wide Web Consortium (W3C) for the development of the Semantic Web [18], but it has gradually evolved as a de-facto standard for a broad spectrum of applications.
Fig. 2. Quantifications and logical operations in SBVR

OWL 2 ontologies provide classes, properties, individuals, and data values, and are stored as Semantic Web documents. An OWL 2 ontology is a formal description of a domain of interest rooted in three syntactic categories that are interpreted under a standardized semantics, which allows useful inferences to be drawn. Figure 3, Figure 4 and Figure 5 depict the structural specification of such categories:

- **Entities** such as classes, properties, and individuals. They are the basic elements of an ontology and are identified by Internationalized Resource Identifiers (IRIs) [7]. For example, a class $a$:$Person$ can be used to represent the set of all people, the object property $a$:$parentOf$ can be used to represent the parent-child relationship and the individual $a$:$Peter$ can be used to represent a particular person called “Peter”.

- **Expressions**, representing complex notions in the domain being described. For example, a class expression describes a set of individuals in terms of the restrictions on the individuals characteristics.

Fig. 3. Entities in OWL 2 ontologies
– **Axioms**, which are statements asserted to be true in the domain being described. For example, a subclass axiom states that the class `a:Student` is a subclass of the class `a:Person`.

OWL 2 ontology language defines several concrete syntaxes that can be used to serialize and exchange ontologies. Among them, the functional style syntax is defined in the OWL 2 structural specification [18] with the aim to state the semantics of OWL 2 constructors and allow a compact writing of ontologies. Following the structural specification insight, the rest of this paper uses such syntax to state OWL 2 ontology expressions.

### 4 SBVR to OWL 2 Mappings

Mappings presented in this section allow the automatable generation of an OWL 2 ontology from the SBVR specifications of a business domain. Transformations are rooted on the structural specification of both standards and are depicted in subsections below by grouping and sequencing them according to the inherent logical order of the subject matter itself. In addition to their theoretical ex-
pression, the mappings are illustrated by building an ontology that reflects the business knowledge exposed by a case study\(^4\).

Fig. 5. Axioms in OWL 2 ontologies

Fig. 6. EU-Rent car movement among branches

\(^4\) Ontology generated from the example can be found in http://code.google.com/p/eurent-mapping-case-study/
The case study has been drawn from the Annex E of the SBVR specification, depicting the business service of a fictitious car rental company with branches in several countries. For a sake of space, only the resulting OWL 2 expressions are described while the SBVR model can be found in the specification\(^5\). Although just a fragment of the case study is presented, it result complete enough in order to depict the usefulness of the proposed mappings. Figure 6 exposes the movements of cars among branches used to illustrate the mappings.

4.1 Core Mappings

As early stated, object types, individual concepts, fact types and fact type roles constitute the core of SBVR metamodel. Therefore, their mappings to OWL 2 expressions are required for more complex translations:

1. Each object type ot is mapped to \(\text{Declaration(Class(a:ot))}\)

2. Each individual concept ic of an object type ot is mapped to:
   - \(\text{Declaration(NamedIndividual(a:ic)}\)
   - \(\text{ClassAssertion(a:ot a:ic)}\)

3. Each unary fact type uft is mapped to:
   - \(\text{Declaration(DataProperty(a:uft)}\)
   - \(\text{DataPropertyDomain(a:uft a:ClassOne)}\)
   - \(\text{DataPropertyRange(a:uft a:DataRangeOne)}\)

4. Each binary fact type bft is mapped to:
   - \(\text{Declaration(ObjectProperty(a:bft)}\)
   - \(\text{ObjectPropertyDomain(a:bft a:ClassOne)}\)
   - \(\text{ObjectPropertyRange(a:bft a:ClassTwo)}\)

5. Each fact type role ftr is mapped by using the \(\text{SubObjectPropertyOf}\) and \(\text{ObjectPropertyChain}\) OWL 2 axioms, as proposed in [8].

The core of the ontology reflecting the business knowledge exposed by the EU-Rent example is built by applying the translations before depicted. As first step, the mappings of the main object types of the case study are shown below.

\(\text{Declaration(Class(eurent:Car Movement))}\)
\(\text{Declaration(Class(eurent:Branch))}\)

Characteristics of object types are expressed. In this case, just the \text{Car Movement} class have such kind of attribute:

\(\text{Declaration(DataProperty(eurent:has Movement-ID))}\)

---

Binary fact types are expressed as properties of classes as shown following. A binary fact type translation is presented below, and the rest of them share the same structure.

In the case study, Branch object type adopts two roles by assuming the function of sending or receiving branch of a car movement. Expressions below shows the mapping of the sending role - the other one is made in the same way -.

Where is_role_of is an OWL 2 object property whose domain and range are the Sending Branch and the Branch classes, respectively.

Finally, although this portion of the case study does not consider the modelling of individuals, to the aim of illustrate the translations it is possible to suppose the existence of a Branch named LAX_Airport_Agency.

4.2 Quantifications Mappings

Quantifications are defined by SBVR as logical formulations introducing a variable and having either the meaning: all referents of the variable satisfy a scope formulation; or a bounded number of referents of the variable exist and satisfy a scope formulation, if there is one. According to such definition and the previous translations, the mappings of the SBVR quantifications are presented depending on the arity of the fact type they ranges over:

1. Universal quantification
   - If the logical formulation scopes over a unary fact type, the expression is mapped to $\text{DataAllValuesFrom}(a:\text{DataPropertyOne} a:\text{DataRangeOne})$
   - If the logical formulation scopes over a binary fact type, the expression is mapped to $\text{ObjectAllValuesFrom}(a:\text{ObjectPropertyOne} a:\text{ClassOne})$
2. **Existential quantification**
   - If the logical formulation scopes over a unary fact type, the expression is mapped to `DataSomeValuesFrom(a:DataPropertyOne a:DataRangeOne)`
   - If the logical formulation scopes over a binary fact type, the expression is mapped to `ObjectSomeValuesFrom(a:ObjectPropertyOne a:ClassOne)`

3. **at-most-n quantification**, where “n” is a non-negative integer
   - If the logical formulation scopes over a unary fact type, the expression is mapped to `DataMaxCardinality(n a:DataPropertyOne a:DataRangeOne)`
   - If the logical formulation scopes over a binary fact type, the expression is mapped to `ObjectMaxCardinality(n a:ObjectPropertyOne a:ClassOne)`

4. **at-least-n quantification**, where “n” is a non-negative integer
   - If the logical formulation scopes over a unary fact type, the expression is mapped to `DataMinCardinality(n a:DataPropertyOne a:DataRangeOne)`
   - If the logical formulation scopes over a binary fact type, the expression is mapped to `ObjectMinCardinality(n a:ObjectPropertyOne a:ClassOne)`

5. **exactly-n Quantification**, where “n” is a non-negative integer
   - If the logical formulation scopes over a unary fact type, the expression is mapped to `DataExactCardinality(n a:DataPropertyOne a:DataRangeOne)`
   - If the logical formulation scopes over a binary fact type, the expression is mapped to `ObjectExactCardinality(n a:ObjectPropertyOne a:ClassOne)`

Remaining SBVR quantifications - `at-most-one`, `exactly-one`, and `numeric-range` quantifications - are easily translatable in terms of the above presented mappings.

From the case study, the mapping of an existential quantification ranging over the function assumed by EU-Rent branches is presented below.

```
ObjectSomeValuesFrom(eurent:is_role_of eurent:Branch)
```

And a exactly-n quantification mapping (where n = 1), establishing the uniqueness of the movement identification of a EU-Rent car movement is shown following.

```
DataExactCardinality(1 eurent:has_Movement-ID xsd:string)
```

### 4.3 Logical Operations Mappings

SBVR defines logical operations as those formulations of meaning based on only the truth or falseness of the meanings of its logical operands. In correspondence with the previous translations, the mappings of the SBVR logical operations are presented depending on the types of the involved logical operands:
1. **Logical Negation**
   - If the logical operand is an object type, then the expression is mapped to \( \text{ObjectComplementOf}(a:operand) \)
   - If the logical operand is a literal, then the expression is mapped to \( \text{DataComplementOf}(a:operand) \)

2. **Conjunction**
   - If both logical operands are object types, then the expression is mapped to \( \text{ObjectIntersectionOf}(a:operand1 a:operand2) \)
   - If both logical operands are literals, then the expression is mapped to \( \text{DataIntersectionOf}(a:operand1 a:operand2) \)

3. **Disjunction**
   - If both logical operands are object types, then the expression is mapped to \( \text{ObjectUnionOf}(a:operand1 a:operand2) \)
   - If both logical operands are literals, then the expression is mapped to \( \text{DataUnionOf}(a:operand1 a:operand2) \)

4. **Equivalence**
   - If both logical operands are object types, then the expression is mapped to \( \text{EquivalentClasses}(a:operand1 a:operand2) \)
   - If both logical operands are individual concepts, then the expression is mapped to \( \text{SameIndividual}(a:operand1 a:operand2) \)
   - If both logical operands are unary fact types, then the expression is mapped to \( \text{EquivalentDataProperties}(a:\text{logicaloperand1} a:\text{logicaloperand2}) \)
   - If both logical operands are binary fact types, then the expression is mapped to \( \text{EquivalentObjectProperties}(a:\text{operand1} a:\text{operand2}) \)

Remaining SBVR logical operations - exclusive disjunction, nand-formulation, nor formulation, and whether-or-not formulation - are translatable by the logical combination of the above presented mappings.

As an example, the mapping below depicts a disjunction expression stating the geographical criteria of rentals classification are based on whether it crosses local area or international boundaries.

\[
\begin{align*}
\text{Declaration(Class(eurent:Geographical_Movement_Type))} \\
\text{Declaration(Class(eurent:InCountry_Car_Movement))} \\
\text{Declaration(Class(eurent:International_Car_Movement))} \\
\text{SubClassOf(eurent:Geographical_Movement_Type}} \text{ObjectUnionOf(eurent:InCountry_Car_Movement eurent:International_Car_Movement eurent:Local_Car_Movement))}
\end{align*}
\]
4.4 Identifiers, Specializations and Classification Mappings

1. **Reference Scheme**, is the SBVR way of identifying instances of a given concept. However, SBVR considers only characteristics to be used as reference schemes, so the expression is mapped to \( \text{HasKey}(a:\text{ClassExpression} \ a:\text{DataPropertyExpressionOne}) \)

In the case study, the `has_Movement-ID` characteristic is used as the identifier of the individuals belonging to the `Car_Movement` object type. The corresponding translation is shown below.

\[
\text{HasKey}(\text{eurent:Car_Movement} () (\text{eurent:has_MovementID}))
\]

2. **Specialization**, is a fact type representing relationships between a more general and a more specific concept.
   - If both concepts are object types, then the expression is mapped to \( \text{SubClassOf}(a:\text{concept1} \ a:\text{concept2}) \)
   - If both concepts are unary fact types, then the expression is mapped to \( \text{SubDataPropertyOf}(a:\text{concept1} \ a:\text{concept2}) \)
   - If both concepts are binary fact types, then the expression is mapped to \( \text{SubObjectPropertyOf}(a:\text{concept1} \ a:\text{concept2}) \)
   - If concept1 is an object type and concept2 is an individual concept, then the expression is mapped to \( \text{ClassAssertion}(a:\text{concept1} \ a:\text{concept2}) \)

An application of the specialization mapping over the case study has been shown in the disjunction expression stated in the previous subsection.

3. **Categorization** and **Segmentation**
   A possible situation is the need of categorize the individuals belonging to certain entity according to a set of different criteria. Such situation is presented in the case study, where the car movements are classified based on whether the car is returned to a branch different from the pick up branch - directional categorization criteria - and on whether the car crosses local area or international boundaries - geographical categorization criteria -. Moreover, the categories belonging to each criteria are disjoint between them. In the SBVR context, such modelling are called categorization and segmentation, respectively. While OWL 2 `ObjectUnionOf` is the way to map a SBVR categorization, OWL 2 `DisjointUnion` are used to translate SBVR segmentations.

Multiple categorization criteria has been already depicted in the case study through the rentals classifications based on directional or geographical aspects. The translations of such business issue is presented following.

\[
\text{Declaration(Class(\text{eurent:Directional_Movement_Type}))}
\]
\[
\text{SubClassOf(\text{eurent:Directional_Movement_Type}}
\]
\[
\text{ObjectUnionOf(\text{eurent:Round_Trip_Car_Movement}}
\]
\[
\text{eurent:OneWay_Car_Movement))}
\]
\[
\text{DisjointClasses(\text{eurent:Round_Trip_Car_Movement}}
\]

5 Related Work

Several works have already recognized that having a mean of mapping SBVR expressions to ontology statements will have strong benefits and will gain wide applicability in the future vision of enterprise information systems [4][5][8][1][6]. The feasibility of the approach by combining OWL 1 [15] and SWRL expressions [16] is explored in [4]. Models transformations chains are presented in [5] as a way to translate SBVR based vocabularies to OWL 1 and R2ML statements [13]. Some basics concepts in transforming SBVR expressions to OWL 2 is introduced in [8]. These proposals shown the usefulness of using SBVR expressions as starting point for ontology development. However, in the first cases they do not use the most mature and powerful technological solutions available, while in the last one just some primary questions are answered.

In the other hand, [1] and [6] propose a set of transformations but following a different approach. These works are rooted in the ORM conceptual modelling language [11], which is at the core of the SBVR proposal. Definition and application of an integrated method that uses ORM to generate an ontology-based query mechanism is presented in [1]. Finally, a set-theoretic semantic for ORM 2 and a formal approach to map a fragment of such language to the \(ALCQI\) fragment of Descriptions Logics [2] is introduced in [6]. Although the work also depicts a tool which implements the translations of a set of ORM 2 constraints into a OWL 2 ontology, the proposal follows a formally grounded approach by stating the logical foundations of the translations between the underlying theories of SBVR and OWL 2.
Instead, the present work depicts a broad and detailed set of transformations that allows the automatable generation of an OWL 2 ontology from the SBVR specifications of a business domain. Transformations are rooted on the structural specification of both standards rather than theoretic considerations of the language, with the aim to provide a set of mappings readily usable for business people or developers concerned with the implementation of a mapping tool.

6 Conclusions and Future Work

An exhaustive and automatable approach for SBVR to OWL 2 transformations has been presented and evaluated through a case study. However, OWL 2 is not expressive enough to model all the possible semantics of business vocabularies. As an example, it can be took the rule of the case study imposing that the sending and receiving branch of a round-trip car movement must be the same. Such restriction can not be expressed by OWL 2 statements. According to that, future theoretic works involve three main issues.

The first one is focused on the validation and formalization of the mappings. Both goals will be achieved by generating an ontological metamodel of SBVR specification by following an similar approach to that adopted in the building of the Ontology Definition Metamodel (ODM) [9]. The second one is related to the definition of mappings from SBVR to Horn Rules expressed in the SWRL language, with the aim to filling the gap between SBVR and OWL 2 expressive power. Such new set of mappings will take advantage of the mappings formalization before mentioned. The third one is about the analysis of the benefits of integrating the mappings approach to EDON [12], an evolutionary method for building ontologies intended to be used as a structural conceptual model of an information system. In a early stage, translations can be useful in a method that make use of ontologies to encapsulate business rules as a mean to raise the flexibility, extensibility and ease of maintenance of enterprise software systems.

Several study groups have been organized with the aim to obtain early feedback about the application of the mappings. Such evaluation effort have resulted in an exploratory experiment comparing the performance and attitudes of some groups of students building an ontology based on SBVR business rules expressions with another set of groups building an ontology based on traditional glossaries describing domain entities. Although a more structured and detailed analysis of the results will be presented in the future, the practitioners mainly highlighted the focus on the declarative specification of business rules that allows to obtain an implemented ontology from business knowledge in a smooth way.

Finally, practical objectives of the research will be directed towards the implementation of a prototype intended to provide automatable translations from SBVR business domain specifications to OWL 2 ontologies.

References