UTN SANTA FE

An Automatable Approach for SBVR to OWL2 Mappings

Emiliano Reynares, María Laura Caliusco, María Rosa Galli
Doctorado en Ingeniería – Mención Sistemas de Información
CIDISI UTN FRSF

CIDISI

AGENDA

1. Context
2. SBVR
3. OWL 2
4. Mappings
5. Discussion and future research directions

THE CONTEXT

Business people have the knowledge to describe business concepts and specify business rules

BUT…

- If we could use reasoners to prove the consistency of business domain information?
- If we could use such knowledge to generate an artifact intended to be used in the analysis stage of a software development process?
- If we could encapsulate the declarative specification of business knowledge into information software systems?

THE CONTEXT

A proposal

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

What's SBVR?

Semantics of Business Vocabulary and Business Rules

Conceived by OMG to provide business people a way to semantically describe business concepts and specify business rules

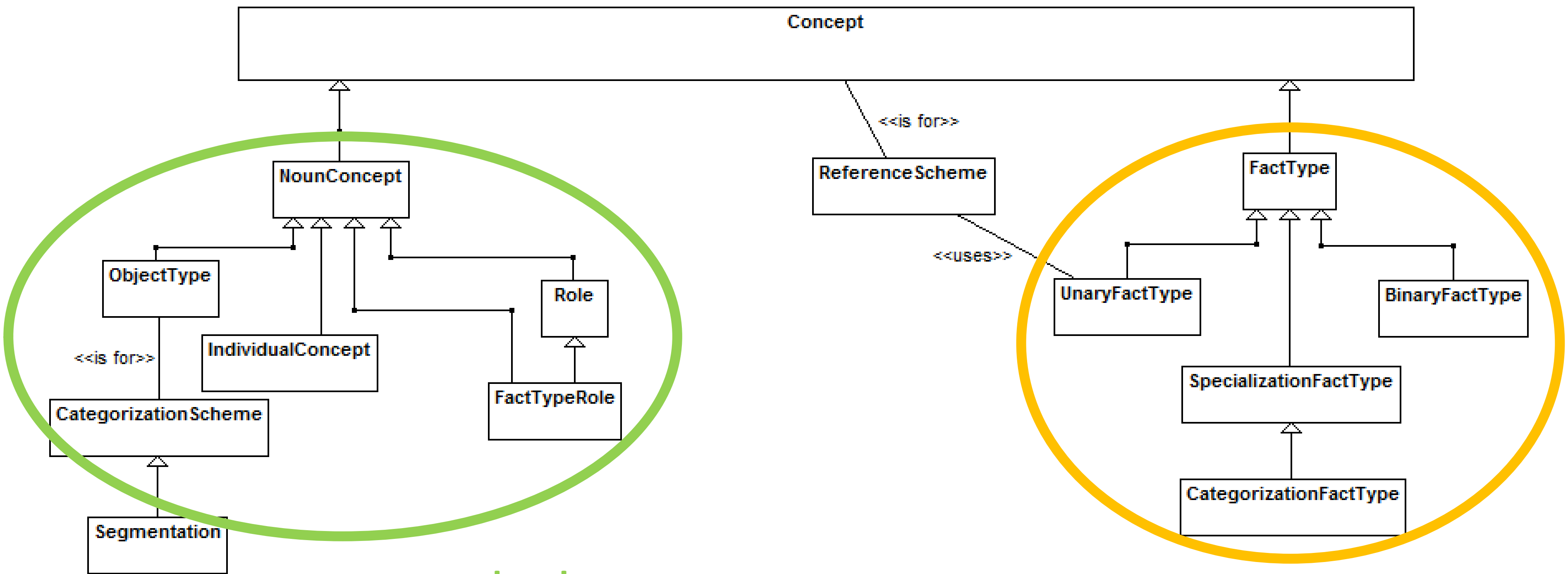
- Conceptualized for business people and designed to be used for business purposes independent of information systems designs
- Linguistic approach enabling the expression of business knowledge through statements rather than diagrams
- Defines a *Controlled Natural Language (i.e. SSE)* and rules for documenting the semantics of business vocabularies, business facts, and business rules

What's SBVR?

Fact-oriented approach of SBVR stemmed from the Business Rules Manifesto, stating that rules builds on facts, and facts build on concepts as expressed by terms

1. *Terms* expressing business concepts
2. *Facts* making assertions about these concepts
3. *Rules* constraining and supporting these facts

What's SBVR?



noun concepts ≈ terms

verb concepts ≈ facts

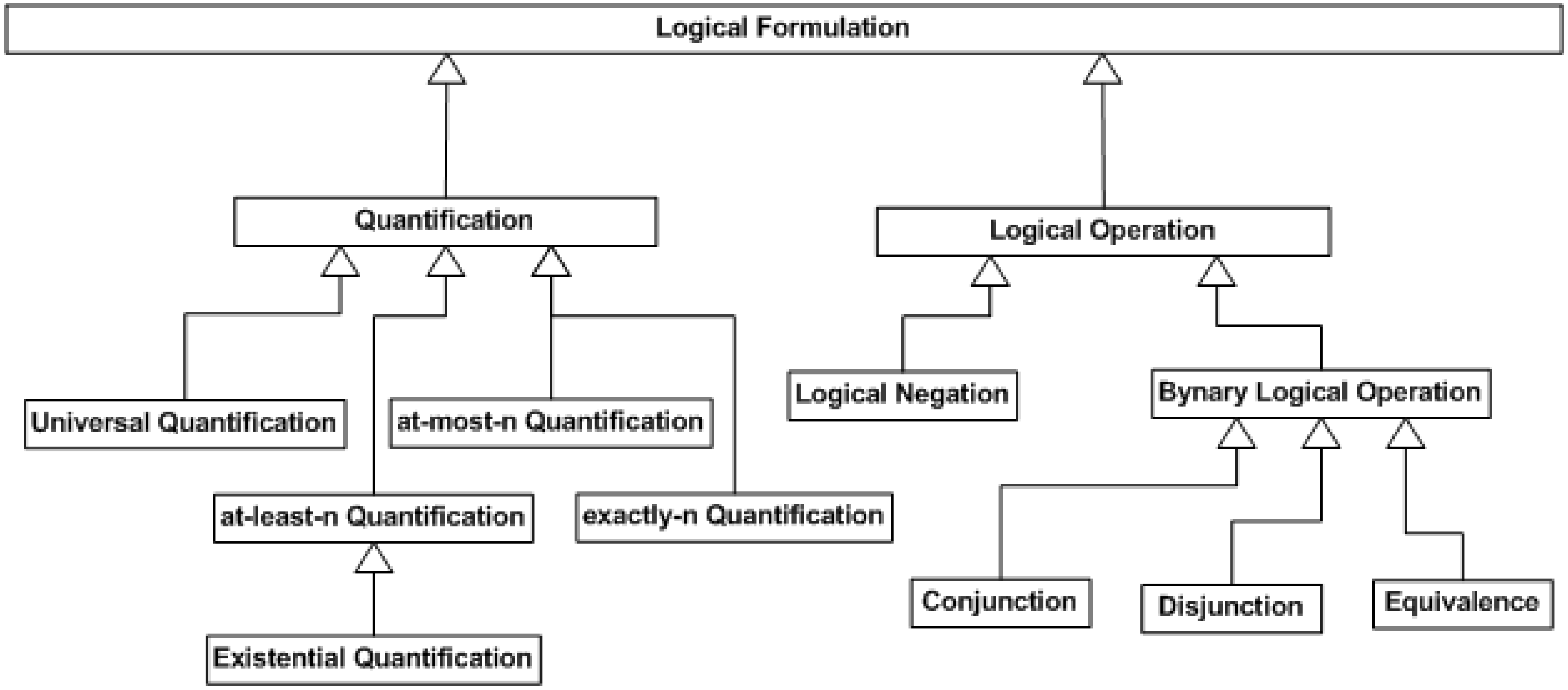
What's SBVR?

A SBVR rule is an element of guidance that introduces an obligation or necessity and distinguishing two general types:

- *structural rules*, which describe the way the business chooses to organize the things it deals with
- *operative rules*, which govern the conduct of business activity by describing business processes.

What's SBVR?

Building rules by imposing restrictions over fact types



What's OWL 2?

OWL 2 Web Ontology Language

Latest version of an ontology language proposed by the W3C for the development of the Semantic Web,
which has gradually evolved as a de-facto standard
for a broad spectrum of applications

- An OWL 2 ontology is a formal description of a domain rooted in three syntactic categories that are interpreted under a standardized semantics
- OWL 2 ontology language defines several concrete syntaxes that can be used to serialize and exchange ontologies
- *Functional-Style Syntax* states the semantics of OWL 2 constructors and allows a compact writing of ontologies

What's OWL 2?

Three syntactic categories

1. Entities

- Basic elements of an ontology
- Identified by Internationalized Resource Identifiers (IRIs)
- Represent classes, properties, and individuals

Example:
A class *"a:Person"* represents the set of all people
An object property *"a:parentOf"* represents the parent-child relationship
An individual *"a:Peter"* represents a person called *"Peter"*

What’s OWL 2?

Three syntactic categories

2. **Expressions**

- Represent complex notions of the domain being described

Example:
A class expression describes a set of individuals in terms of the restrictions on the individuals characteristics

What's OWL 2?

Three syntactic categories

3. Axioms

- Statements asserted to be true in the domain being described

Example:
A subclass axiom state that the class *"a:Student"* is a subclass of the class *"a:Person"*

Again…

A proposal

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

MAPPINGS

Core Mappings

1. **Each object type ot is mapped to:**

   Declaration(Class(a:ot))

2. **Each individual concept ic of an object type ot is mapped to:**

   Declaration(NamedIndividual(a:ic)
   ClassAssertion(a:ot a:ic)

3. **Each unary fact type uft is mapped to:**

   Declaration(DataProperty(a:uft)
   DataPropertyDomain(a:uft a:ClassOne)
   DataPropertyRange(a:uft a:DataRangeOne)

MAPPINGS

Core Mappings

4. **Each binary fact type bft is mapped to:**

   Declaration(ObjectProperty(a:bft)
   ObjectPropertyDomain(a:bft a:ClassOne)
   ObjectPropertyRange(a:bft a:ClassTwo)

5. **Each fact type role ftr is mapped by using**

   "SubObjectPropertyOf" and "ObjectPropertyChain" OWL 2 axioms, as proposed by [Karpovic et al., 2011]

MAPPINGS

Quantifications Mappings

1. **Universal quantification**

   If the logical formulation scopes over a unary fact type, the expression is mapped to DataAllValuesFrom(a:DataPropertyOne a:DataRangeOne)

   If the logical formulation scopes over a binary fact type, the expression is mapped to ObjectAllValuesFrom(a:ObjectPropertyOne a:ClassOne)

2. **Existential quantication**

   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)

MAPPINGS

Quantifications Mappings

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)

MAPPINGS

Quantifications Mappings

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

MAPPINGS

Logical Operations Mappings

1. **Logical Negation**

   If the logical operand is an object type, then the expression is mapped to ObjectComplementOf(a:operand)

   If the logical operand is a literal, then the expression is mapped to DataComplementOf(a:operand)

2. **Conjunction**

   If both logical operands are object types, then the expression is mapped to ObjectIntersectionOf(a:operand1 a:operand2)

   If both logical operands are literals, then the expression is mapped to DataIntersectionOf(a:operand1 a:operand2)

MAPPINGS

Logical Operations Mappings

3. Disjunction

If both logical operands are object types, then the expression is mapped to ObjectUnionOf(a:operand1 a:operand2)

If both logical operands are literals, then the expression is mapped to DataUnionOf(a:operand1 a:operand2)

4. Equivalence

If both logical operands are object types, then the expression is mapped to EquivalentClasses(a:operand1 a:operand2)

If both logical operands are individual concepts, then the expression is mapped to SameIndividual(a:operand1 a:operand2)

MAPPINGS

Logical Operations Mappings

4. Equivalence (cont.)

If both logical operands are unary fact types, then the expression is mapped to EquivalentDataProperties(a:logicaloperand1 a:logicaloperand2)

If both logical operands are binary fact types, then the expression is mapped to EquivalentObjectProperties(a:operand1 a:operand2)

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

MAPPINGS

Identifiers, Specializations and Classification Mappings

1. Reference Scheme

Only characteristics to be used as reference schemes, so the expression is mapped to HasKey(a:ClassExpression a:DataPropertyExpressionOne)

2. Specialization

If both concepts are object types, then the expression is mapped to SubClassOf(a:concept1 a:concept2)

If both concepts are unary fact types, then the expression is mapped to SubDataPropertyOf(a:concept1 a:concept2)

MAPPINGS

Identifiers, Specializations and Classification Mappings

2. Specialization (cont.)

If both concepts are binary fact types, then the expression is mapped to SubObjectPropertyOf(a:concept1 a:concept2)

If concept1 is an object type and concept2 is an individual concept, then the expression is mapped to ClassAssertion(a:concept1 a:concept2)

3. Categorization and Segmentation

While OWL 2 ObjectUnionOf is the way to map a SBVR categorization, OWL 2 DisjointUnion are used to translate SBVR segmentations

DISCUSSION

Distinctive features

1. **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

2. **Transformations are rooted on the structural specification**
   of both standards rather than theoretic considerations of the language

3. **Provides a set of mappings readily usable**
   for business people or developers concerned with the building of a mapping tool

EXPERIENCES

Several study groups have been organized with the aim to obtain early feedback about the application of the mappings

1. **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

2. **A more structured and detailed analysis of the results will be presented,**
   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

FUTURE RESEARCH DIRECTIONS

A practical objective

1. **Implementation of a prototype,**
   intended to provide automatable translations from SBVR business domain specifications to OWL 2 ontologies

FUTURE RESEARCH DIRECTIONS

Future theoretic works involve three main issues

1. **Validation and formalization of the mappings**
   by generating an ontological metamodel of SBVR specification by following an similar approach to that adopted in the *Ontology Definition Metamodel* (ODM)

2. **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

3. **Analysis of the integrating of the mappings approach to EDON,**
   an evolutionary method for building ontologies intended to be used as a structural conceptual model of an information system *[Reynares et al, 2012]*

Questions?

Emiliano Reynares, María Laura Caliusco, María Rosa Galli
ereynares@frsf.utn.edu.ar
mcaliusc@frsf.utn.edu.ar
mrgalli@santafe-conicet.gov.ar

CIDISI UTN FRSF