# Modeling Ontology Design Patterns with Cascaded Role Sets

Hermann Bense[1]

[1]bense.com Digital Publishing GmbH, Schwarze-Brüder-Str. 1, 44137 Dortmund, GERMANY

Abstract

Research into knowledge and ontology engineering has seen numerous attempts to introduce *Ontology Design Patterns* (ODP) to support reuse and modularization. *Cascaded Role Sets* (CRS) provide a novel method for defining ODPs. CRSs can be utilized to model complex ontological structures, to build *Controlled Vocabularies of Concepts* (CVC), and to check the accuracy of ontologies during their evolution. We also show how the entries of CVCs can be constructed using the *Genus-Differentiae-Pattern* (GDP) to compose *Compound Concepts* (CC) from *Prime Concepts* (PC).

**Keywords**

Concept Binary Trees (CBT), Concept Composition, Controlled Vocabularies of Concepts (CVC), Cascaded Role Sets (CRS), Genus-Differentiae-Pattern (GDP), Ontology Design Patterns (ODP), Ontology Evolution, Reification, Compound Concepts (CC), Prime Concepts (PC)

## 1. Introduction

In ontology and linguistics engineering there is quite a good understanding of *inheritance* and the management of taxonomies. However, the formal construction of complex structures seems to be less systematized. *Relationship types* are often also referred to as *Object Properties* (OP), and are applied in modeling roles and complex state of affairs. In this context, research communities focus on terms such as *Ontology Design Patterns* (ODP), *object aggregates* [1], *variable embodiment* [2], *clusters of relational properties* [3], *thematic roles* [4], *social roles* and *abstract roles* [5], etc. *Controlled Vocabularies* (CV) such as *Princeton WordNet* (PWN) [6] typically use semantic relations, which organize concepts into taxonomies and *sets of words which are all pairwise near-synonyms* (SynSets) [7]. How language concepts and language building blocks could otherwise be related to each other calls for necessary extensions. The aim of this paper is to introduce a methodology that formalizes the modeling of complex ontological and linguistic structures, while providing the possibility of ensuring modeling accuracy with rules within the ontology evolution process.

The key research question we answer is: What are limitations in ontology and linguistics engineering when it comes to using design patterns, and what methods can be used in overcoming the limitations. This raises other additional questions: How can

*CAOS VII: Cognition and Ontologies, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, July 19–20, 2023, Sherbrooke, Québec, Canada*
hb@bense.com (H. Bense)
https://www.ontology4.us/ (H. Bense)
0000-0002-2562-224X (H. Bense)

the (structural) accuracy of ontologies be monitored during the ontology evolution process? Utilizing *Cascaded Roles Sets* (CRS), how can meanings of words and concepts be more formally defined, synchronized cross-lingually, and collected to build a standardized *Controlled Vocabulary of Concepts* (CVC)?

The remainder of the paper is organized as follows: In the Related Work section 2, we discuss what approaches already exist for modeling ontological and linguistic structures and what disadvantages they entail. The Methodology section 3 describes the basic concepts with respect to structured modeling with *Cascaded Role Sets* (CRS). In the Examples section 4 we provide examples for modeling *relators* and *reification* with CRSs. Using modeling examples for natural language concepts, the application of CRSs is further discussed in the Concept Composition section 5. In the Discussion section 6, we evaluate the extent to which our methodology contributes to the improvement of structured conceptual modeling in ontology and linguistics engineering compared to existing approaches. In the Conclusion section 7, we summarize the results and provide an outlook for future research activities.

## 2. Related Work

On the website ontologydesignpatterns.org a large number of *Ontology Design Patterns* (ODP) have been compiled over the years. The visualization is done using *Unified Modeling Language* (UML)-like diagrams such as for the pattern 'AcademicRoles'[1]. The rules for the ODPs only to a certain extent specify which restrictions exist for the relationship types between the model elements involved. In addition, we do not see how the correct application of the ODPs can be monitored. In Smith and Spear's [1] work, on the pages 88 and 93 describing the hierarchy of *Basic Formal Ontology* (BFO) continuants, we find the concept of an *object aggregate*, which is defined as collection of objects. Examples given include 'heap of stones', 'group of commuters on the subway', 'a flock of geese', 'a symphony orchestra' etc. Since all these sets can be defined by object properties, we find the concept of an 'object aggregate' to be debatable to appear in a class hierarchy. Herre et al. in [8], page 49, discuss the mathematical aspects of sets within the scope of the *General Formal Ontology* (GFO). Furthermore, on page 41 they discuss "most complex entities" such as 'chronoids', 'topoids', 'situoids' and 'configurations' as *constituting aggregate of facts*. However, these works do not provide a concrete methodology for structuring complex entities. Gangemi et al. [9], page 13, Table 2, map the concepts of WordNet to those of the upper ontology DOLCE (*Descriptive Ontology for Linguistics and Cognitive Engineering*). The authors explain on page 7: "The common trait of aggregates is that they are endurants with no unity ... We consider two kinds of aggregates: *amounts of matter* and *arbitrary collections*. ... We may have called these arbitrary collections *groups*, or perhaps *sets*; but we refer to use *set* for abstract entities, and *group* for something having an intrinsic unity". In the paper of Guizzardi et al. [3], on page 4, *relators* are described as a means "to represent clusters of relational properties that 'hang together' by a nexus". The examples provided, such as 'enrollments', 'employments', 'citizenships', 'marriages', 'car rentals' etc., are then characterized as fully-fledged *en-*

[1] http://ontologydesignpatterns.org/wiki/Community:AcademicRoles

*durants*. Under *mediation* they can model a particular type of existential dependence relation. For example, for 'married_with', it can be required, that both spouses must exist. Beyond that, however, we do not see how it would be possible to define which participants in a 'marriage' are mandatory, and which are optional. As in the other cited works, the nesting of structures with relators is not explicitly addressed. John Sowa [10] compares different graph types, such as *Conceptual Graphs* (CG), *Correlational Nets* and *Dependency Graphs* in connection with the possibility of representing logic and semantic network relationships. Sowa also describes the concept of *Thematic Roles* (TR). One specialist aspect of thematic roles are the *social roles*, which are distinguished from *abstract roles* by Loebe in [5] and presented as part of the GFO-Ontology [8], pages 37ff. The Somers-Dick-Matrix, which Sowa in [4] uses to explain his theory of thematic roles, contains further semantic relations. The hierarchy of roles used by Sowa is more extensive than that used by Loebe. Therefore, we consider it worthwhile to investigate how roles such as 'Agent', 'subRoleOf', 'SocialRole', 'Completion', 'Destination', 'Duration', 'Effector', 'Experiencer', 'Location', 'Material', 'Origin', 'Patient', 'PointInTime', 'Recipient', 'Result', 'Start', 'Theme' etc. can be utilized for a more extensive modeling of a state of affairs. The question that arises is: What general *Ontology Design Patterns* (ODP) and graphic notations can be used to represent complex ontological structures in such a way that they can be used as a helpful tool for rapid ontology development and for checking the correctness of ontologies?

**Word Sense Definitions and WordNet:** Philip Johnson-Laird in [11], Page 216, poses the question: "Can the meaning of words be defined?" and proceeds with: "On the one hand, the most extreme claim that meanings can be analyzed into semantic elements parallels the fundamental theorem of arithmetic: just as there is only one way of decomposing a number into prime factors, so the meaning of any word can be decomposed into a unique product of semantic primitives". As an example for an adjective definition he presents *bachelor is an unmarried man*. Johnson-Laird also raises the definability-question and argues "... it is important to appreciate that the meaning of a word might be decomposable into ineffable components, and therefore impossible to define adequately". Johnson-Laird argues, that by applying an analogy to the bootstrap method for compiler construction in computer science, one can acquire meanings of other words on the basis of informal definitions. So, what remains unresolved is the question of whether one can create a controlled vocabulary with *Word Sense Definitions* (WSD) in such a way that the entries can be formally assembled from semantic primitives (*Semantic Primes* (SP)) into more complex terms (*Semantic Compounds* (SC)). One of the most comprehensive and best-known projects dealing with words, synonyms and language is undoubtedly the *Princeton WordNet* (PWN). In our opinion, PWN qualifies as an linguistic upper ontology by including the most general concepts as well as more specialized concepts, related to each other not only by the subsumption relations, but by other semantic relations as well, such as 'part-of' and 'cause'. However, it has not been formally axiomatized and thus does not grant a precise representation of the logical relations between the concepts. The paper by Bond et al. [12] describes a number of weaknesses in PWN, and proposes improvements. For example, one of their criticisms is the lack of coordination across projects for handling the cross-lingually linking of *SynSets* and that the semantical and lexical relations do not mean the same thing in different languages. They also identify large differences in vocabulary coverage and in the degree of *polysemy*. In 2002, Gangemi et al. [9] discussed the problems of PWN from the point of view of ontology engineering research. Among other aspects, they identify as critical problems "the confusion between concepts and individual", "the notorious multiplication of sense", "the heterogeneity in levels of generality", and the fact that "PWN does not sufficiently support *polysemy detection*". Therefore, we will also discuss in the following how one can back the concepts represented in controlled vocabularies with ontological definitions and vice versa.

## 3. Methodology

Within this chapter, we introduce *Cascaded Role Sets* (CRS) in order to lay the foundation for an enriched method to define *Ontology Design Patterns* (ODP). Before answering questions, we must introduce a few naming conventions and graphical notations. Figure 1 is a so-called *Ontograph* visualization of a sample ontology that makes use of naming conventions according to [13]. The open-source graphics library graphviz[2] is used to create Ontographs. The use of the allowed colors and shapes of the nodes and edges gives visual feedback for the types of the elements and support the consideration of knowledge graphs in the form that we call *Visual Thinking*. The quick switching between ontology modeling and visualization supports rapid prototyping, thus supporting error detection and correction.

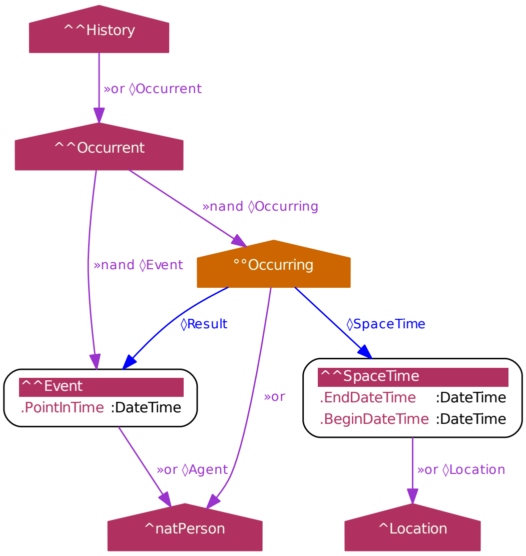


Figure 1 Basic Cascaded Role Sets

**Name Sets:** Be $\mathrm{N} = \{\mathrm{char}\}^*$ a set of character strings. As our key naming convention, the name of an ontological concept always uses the index of the designating set as its prefix: $\mathrm{N_x} = \{\mathrm{xn} \mid \mathrm{n} \in \mathrm{N}\}$. For an unambiguous designation of the names of ontological concepts we introduce the following name sets: $\mathrm{N}_{\hat{}}$= Class Names, $\mathrm{N}_{.}$= Data Property Names, $\mathrm{N}_{\Diamond}$= Object Property Names, $\mathrm{N}_{>}$= Particular Names, $\mathrm{N}_{\gg}$ =Relator Names, $\mathrm{N}_{\circ}$ =Process Names, $\mathrm{N}_{\sim}$= Function Names and $\mathrm{N}^* = \cup \{\mathrm{N_x} \mid \mathrm{x} \in \{\hat{}, ., \Diamond, >, \gg, \circ, \tilde{}\}\}$. In short, the symbols are motivated to evoke associations with hierarchies, properties, processes, things in flux, and references / pointers. This means that the name sets defined in this way form the vocabulary of a meta-language for compact and expressive notations. Next, we would like to motivate our approach and present the main features of our methodology. In [14], page 9, we find an example for roles which are logically

[2]http://www.graphviz.org

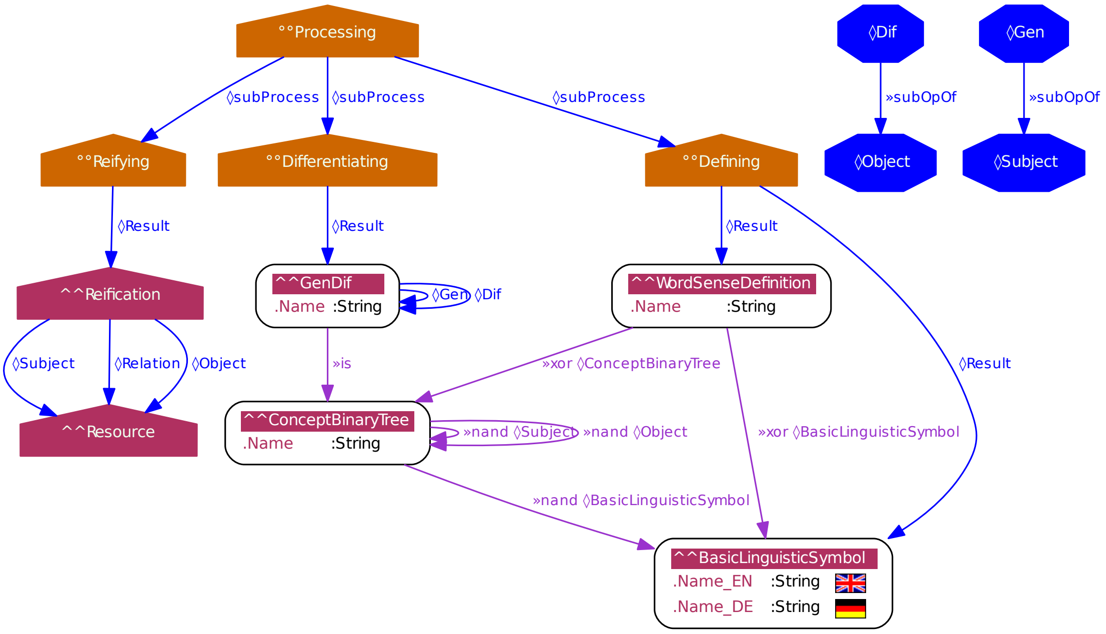


**Figure 2:** Role Sets for Concept Composition

connected with the Boolean `xor`. The example for AgentRole[3] uses `or`. We will generalize this principle to other role operators in the following. As depicted in figure 1 we prefix the name of top classes taking part in Cascaded Roles Sets with a double circumflex (^^) and names of top processes with double-degree (°°). Roles or sets of roles can be aggregated into a role set definition using OPs from the set of *Role Set Object Properties* (RSOP), which is defined as $\mathscr{RSOP}$ = `{»and, »nand, »any, »opt, »or, »xor, »nxor, »integer}`. The term `»integer` stands for any `»one`, `»two`, `»three` etc. Let be `(minCard, maxCard)` the range of the number of entities which can be related using one of the role types. Then, CRS definitions can be constructed according to the following rules / constraints.

- `»integer`: Between `n and m = integer` roles must be instantiated: `(n, m)`. `»one` is the default, if no other RSOP is specified.
- `»and`: A role must be instantiated for each `»and`: `(n, n)`
- `»nand`: Any of the `»nand` roles can be instantiated, but not all: `(1, <n)`
- `»any`: The instantiation of each `»any` role is optional: `(0, n)`
- `»opt`: Zero or one role can be instantiated for each `»opt`: `(0, 1)`
- `»or`: At least one role has to be instantiated: `(1, n)`
- `»xor`: Exactly one of the connected `»xor`-roles must be instantiated: `(1,1)`
- `»nxor`: All `»nxor`-roles must be instantiated, or none: `(0,0)` ∨ `(n, n)`

The Ontograph in figure 1 shows applications of RSOPs:

- **»or:** A `^^History` has one or more `^^Occurrents`. An `^^Event` has at least one agent.
- **»nand**: An `^^Occurrent` is either an `^^Event` or a processual role relator `°°Occurring`.
- **»one** and **»or**: The processual role relator `°°Occurring` has exactly one `^^SpaceTime` and one result and at least one agent.

[3] http://ontologydesignpatterns.org/wiki/Submissions:AgentRole

When applying the RSOPs, those with the same name aggregated directly under a ^^ definition are combined with a logical 'and' ($\wedge$). The Boolean roles '»not' and '»nor' are not required because the absence of roles can be modeled with the '»any'-Role. The Ontograph in Figure 2 introduces further processual roles for different types of definitions of ontological and linguistic structures:

- Knowledge can be reified using `∘∘Reifying` and `^^Reification`, which connects concepts by the object properties »`Subject`, »`Relation` and »`Object`.
- *Word Sense Definitions* (WSD) can be created by using `∘∘Defining` and `^^WordSenseDefinition`, which utilizes C*oncept Binary Trees* (CBT).

*Concept Binary Trees* (CBT) are utilized for modeling *Ontological Concepts* (OC) using *Genus-Differentiae Patterns* (GDP) and for modeling atomic and compound *Linguistic Concepts* (LC) and definitions. CBTs are defined by the relator class `^^ConceptBinaryTree` in Figure 2 and can be nested to any depth using two »`nxor` as 'all or nothing relators'. At the same time, with `CBT(x[,y])=` $\Delta_{x[,y]}$ a recursive function can be defined that returns the full lexical definition of a concept: The function isSC(z): $\mathscr{BLS} \rightarrow \{T, F\}$ assigns an element of the set of Basic Linguistic Symbols $\mathscr{BLS}$ the value T|TRUE|1, if it is compound, else F|FALSE|0:

**Df 3.1** (isSC = is Semantic Compound)**.** isSC(z) ← T ⇔ (∃x :(z, »Subject, x) ∧ ∃y :(z, »Object, y)) or ∃x, y: (z = x_y), otherwise, isSC(z) ← F.

**Df 3.2** (CBT() = Concept Binary Tree Function)**.** If isSC(z) then ∃x, y such that $\Delta_z \leftarrow (\Delta_x, \Delta_y)$, otherwise $\Delta_z \leftarrow z$.

For the purpose of modularization and reuse we define *Role Set Items* (RSI) and *Cascaded Role Sets* (CRS) as follows.

**Df 3.3** (Role Set Item)**.** rsi=([rsop]◊op, R) with rsop ∈ ***RSOP*** an optional Role Set Object Property, and ◊op an Object Property (OP) and R a resource.

**Df 3.4** (Cascaded Role Set)**.** »$\mathrm{CRS}_{\mathrm{Name}} = \Theta_{\mathrm{Name}} = \{\mathrm{rsi}\}^{+}$ is a non empty set of Role Set Items.

## 4. Examples

In the following we present different examples for the application of *Cascaded Role Sets* (CRS). This comprises the modeling of n-ary relators and state of affairs, the modeling with reification, and the implementation of *Concept Binary Trees* (CBT) for the maintenance of *Word Sense Definitions* (WSD). We will apply our approach to the example of how to represent 'marriage' in order to illustrate the pluses of our approach. The modeling of 'marriage', which is much discussed in the literature, see e.g. [15], page 6, is based on the cascaded role sets depicted in figure 3. We cite [9], page 3: "The main distinction between endurants and perdurants is that of *participation*: an endurant *lives* in time by *participating* in a perdurant. For example a person, which is an endurant, may participate in a discussion, which is a perdurant". In the marriage-based example, we determine that endurants are the participating agents ◊`Husband` and ◊`Wife` which are mediated by `^^Spouses` while `^^Marriage` itself is a perdurant. In [9] we also find: *Enduring entities* are also called *continuants* and *perdurant entities* are called *occurrents* where *occurrent* is a synonym for *perdurant.*

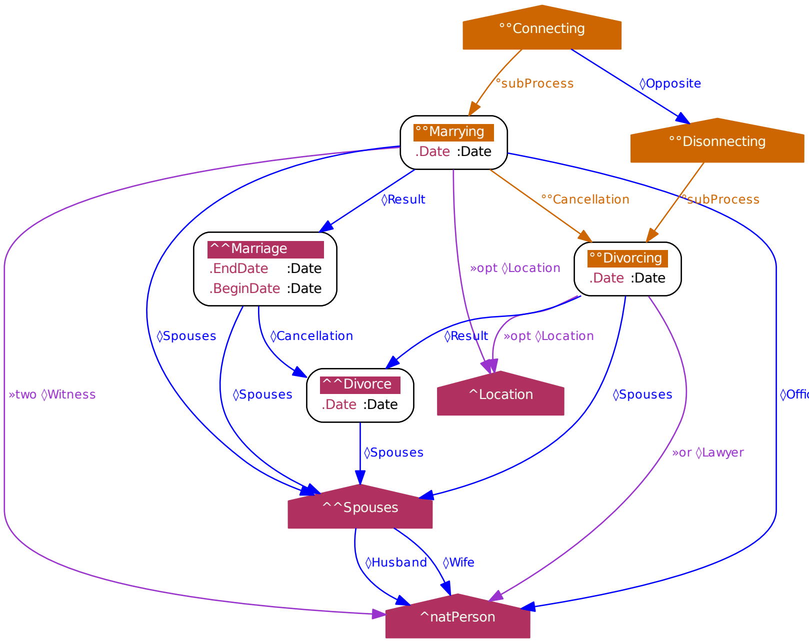


**Figure 3:** Cascaded Role Sets for Marrying and Divorcing

When it comes to modeling n-ary relations, as discussed in Use Case 3 of the w3.org page about n-ary relations[4] none of the participants of the n-ary relations is privileged / "standing out". A marriage might have as its participants the spouses, the officer and witnesses to the marriage. Figure 4 shows the modeling of 'marriage' based on *Cascaded Role Sets* (CRS) depicted in figure 3. It has the advantage of being completely symmetric because `»Husband` and `»Wife` are aggregated under `»Spouses_BT`. In this way, it was also possible to elegantly model how two people can be married several times in their lives. Here, 'marriage' is also an example for the result of the application of the processual role set ∘∘Marrying. ∘∘Marrying has in addition to spouses, an officer and two witnesses to 'marriage' and an optional 'location'. The example of 'marriage' at best illustrates the characteristics of cascading since the spouses as a CRS are embedded under 'marriage'. In contrast, ∘∘Divorcing does not require witnesses. To wrap it up: The Ontograph in figure 3 shows the concepts and structures used for 'marriage' and divorce. Marriage is a process of joining, 'divorce' is a process of separation. The result of 'marriage' is modeled with the state of affairs `^^Marriage`, that of the 'divorce' with the `^^Divorce`. What they both have in common is the couple, which is connected with the roles ◊`Husband` and ◊`Wife` using the `^^Spouses` relator. The period of the 'marriage' is defined by `.BeginDate` and `.EndDate`. Divorce is an event that takes place on a specific `.Date` and represents the transition between 'married' and 'divorced'. In contrast to other previously known modeling approaches, relationships between knowledge subjects can not only be equipped with additional attributes but can also can connect relators to other relators. Then, for 'marriage' the complete CRS definition is:

[4]https://www.w3.org/TR/swbp-n-aryRelations/#representation

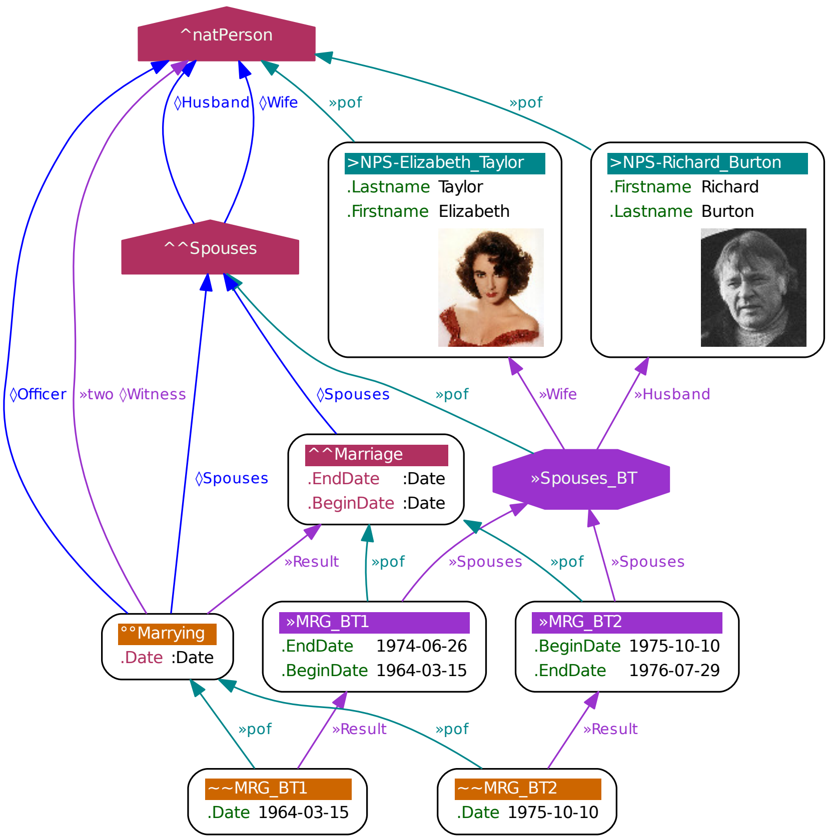


**Figure 4:** Marriage of Burton-Taylor

- $\Theta_{{}^\wedge{}^\wedge\mathrm{Spouses}} =$ `{(◊Husband, ^natPerson), (◊Wife, ^natPerson)}`
- $\Theta_{{}^\wedge{}^\wedge\mathrm{Marriage}} =$ `{(◊Spouses, ^^Spouses)}`, Θ`^^Divorce` $=$ `{(◊Spouses, ^^Spouses)}`
- $\Theta_{\circ\circ\mathrm{Marrying}} =$ `{(◊Result, ^^Marriage), (◊Spouses, ^^Spouses), (»two◊Witness, ^natPerson), (◊Officer, ^natPerson), (»opt◊Location, ^Location)}`
- $\Theta_{\circ\circ\mathrm{Divorcing}} =$ `{(◊Result, ^^Divorce), (◊Spouses, ^^Spouses), (»or◊Lawyer, ^natPerson), (»opt◊Location, ^Location)}`

## 5. Concept Composition

In this section, we present different methods for composing concepts from other concepts. These include `∘∘Differentiating` with the *Genus-Differentiae Pattern* (GDP), `∘∘Defining` with the *Concept Binary Trees* (CBT) and `∘∘Reifying` with the Reification Pattern. All three are based on the method for creating *Ontology Design Patterns* (ODP) introduced in the CRSs section 3. Creating a concept C from two concepts A and B, is called *defining*. *Defining* enables not only the linking of ontological concepts, but also that of words in a natural language to be able to define terms. This is an important aspect of ontology and linguistics modeling. `Defining` is modeled based on processual role set `∘∘Defining`. In contrast to text-only definitions, we call the formal definitions of concepts *Word Sense Definitions* (WSD). *Genus-Differentiae Patterns* (GDP) are represented by special *Concept Binary Trees* (CBT), using the CRS Definition `^^GenDif` where `(♢Gen, »is, ♢Subject)` and `(♢Dif, »is, ♢Object)` and we abbreviate `GDP(x) =` $\Gamma_x$. The Ontograph in figure 5 depicts the complete multi-lingual CBT definition `>BLS-barometer` of the *Ontological Concept* (OC) `^Barometer`. For a compound composed with the *Genus-Differentiae Pattern* (GDP) holds: $(\Delta_x, \Delta_y) = \Delta_z = \Gamma_z \Leftrightarrow$ (z, »Dif, x) $\wedge$ (z, »Gen, y). Since with `(♢Gen, »is, ♢is)` the Object Property (OP) `♢Gen` is defined as a sub OP of `♢is` it can be entailed $\Gamma_D$ = (C, $*$) $\rightarrow$ (D, »is, C) or in other words C subsumes D.

- $\Gamma_{\text{^SeaBird}}$ = `(^Bird,^Sea)`, $\Gamma_{\text{^Carnivore}}$ = `(^Eater,^Meat)`
- `(^AirPressure, »Gen, ^Pressure)` and `(^AirPressure, »Dif, ^Air)` $\rightarrow$ $\Delta_{\text{^AirPressure}}$ = `(^Pressure,^Air)` = $\Gamma_{\text{^AirPressure}}$ $\rightarrow$ `(^AirPressure, »is, ^Pressure)`
- `(^Barometer, »Gen, ^Gauge)` and `(^Barometer, »Dif, ^AirPressure)` $\rightarrow$ $\Delta_{\text{^Barometer}} = \Gamma_{\text{^AirPressure}} = (\Delta_{\text{^Gauge}}, \Delta_{\text{^AirPressure}})$ = `(^Gauge, (^Pressure, ^Air))` $\rightarrow$ `(^Barometer, »is, ^Gauge)`

*Basic Language Symbols* (BLS) correspond to *lexemes* and therefore its senses “relate to a single general meaning” [7].

Their names are prefixed with `>BLS` and associated with the *Ontological Concept* (OC) by the object property `»SymbolOf`, e.g. `(>BLS-pressure, »SymbolOf, ^Pressure)`. This corresponds to that edge in a *semiotic triangle*, where a *symbol symbolizes a reference*. This

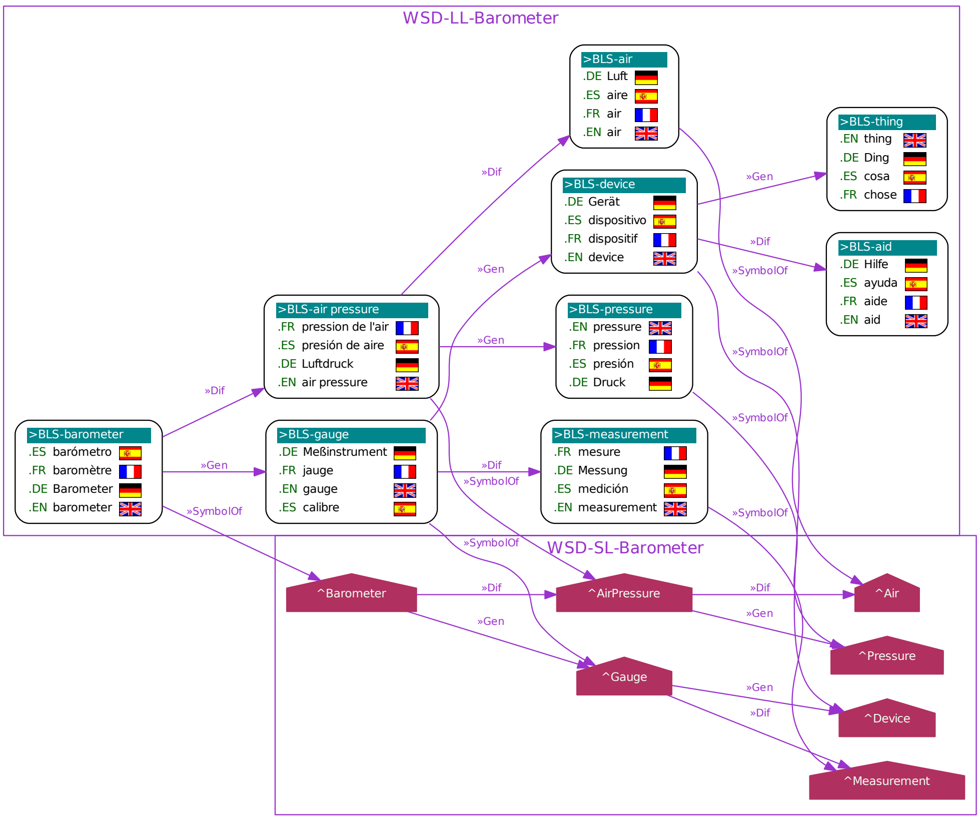


**Figure 5:** Word Sense Definition of Barometer

is called *symbol grounding*. With homonyms, each term would be assigned a unique BLS, as with `>BLS-organ_(bodypart)` and `>BLS-organ_(musical-instrument)`. This also applies to polysemes such as 'run', where we would distinguish, for example, between the verb `>BLS-run_(to)` and the noun `>BLS-run_(race)`.

**Concept Congruence Axiom:** *Ontological Concepts* (OC) defined by the *Genus - Differentiae Pattern* (GDP) are congruent (≊) to their *Word Sense Definitions* (WSD) in the following way. The association between BLSs and their classes is established by `(>BLS-gauge, »SymbolOf, ^Gauge)`, `(>BLS-air pressure, »SymbolOf, ^AirPressure)`, `(>BLS-barometer, »SymbolOf,^Barometer)` etc. Now replacing each *Basic Linguistic Symbol* (BLS) related by `»SymbolOf` to the class names `^AirPressure`, `^Barometer`, ^Gauge, and `^Pressure` yields: $\Delta_{\text{>BLS-Barometer}} = (\Delta_{\text{>BLS-Gauge}}, \Delta_{\text{>BLS-AirPressure}}) = (\text{>BLS-Gauge}, (\text{>BLS-Pressure}, \text{>BLS-Air})) \approxeq \Delta_{\text{^Barometer}} = (\Delta_{\text{^Gauge}}, \Delta_{\text{^AirPressure}}) =$ `(^Gauge, (^Pressure, ^Air))`. With the axiom 1.1 we can now define when an *Ontological Concept* (OC) is congruent to a *Linguistic Concept* (LC). Be $\text{lc} = \Delta_{\text{lc}}$ a *Prime Concept* (PC) and $\Delta_{(\text{lx, ly})}$ a *Compound Concept* (CC). Then (1) (lc, »SymbolOf, oc) $\Leftrightarrow \Delta_{\text{lc}} \approxeq \Delta_{\text{oc}} \wedge \text{lc} \approxeq \text{oc}$ and (2) (lx, »SymbolOf, ox) $\wedge$ (ly, »SymbolOf, oy) $\Leftrightarrow \Delta_{(\text{lx, ly})} \approxeq \Delta_{(\text{ox, oy})} \wedge (\text{lx, ly}) \approxeq (\text{ox, oy})$.

**Ax 1.1** (Concept Congruence Axiom CCAx). $lc \approxeq oc \Leftrightarrow \Delta_{lc} \approxeq \Delta_{oc}$.

I.e., for every ontological concept 'oc' there should be a linguistic concept 'lc' that defines its linguistic meaning and vice versa the ontological meaning. This can be checked with Axiom 1.1. If the congruence cannot be derived from the axiom, then this indicates a possible inaccuracy or incompleteness of the ontology. To the best of our knowledge, we are not aware of any approach in the literature that makes it possible to reconcile ontological and linguistic concepts in a comparable way.

**Reification:** In the RDF Primer[5] is stated: "For one thing, it is important to note that in the conventional use of reification, the subject of the reification triples is assumed to identify a particular instance of a triple in a particular RDF document, rather than some arbitrary triple having the same subject, predicate, and object". The triple `(>ASB-Earth, »OrbiterOf, >ASB-Sun)` models the fact, that 'the Earth orbits the Sun'. When querying the knowledge graph this assertion can be assumed as true. How can it be represented, that the worldview before Kopernikus was, that the Sun orbited the Earth and since Kopernikus the worldview was the opposite? Finally, the Onto-graph in figure 6 represents how the belief of Nikolaus Kopernikus can be modeled with the ^^Reification CRS as conceptual graph as defined by [10]. Using the definition of `∘CRS_Reificating` (s. Figure 2), relators can be instantiated that transform asserted facts into un-asserted facts.Taking `»Earth_orbits_Sun` as a unique identifier, the knowledge atom `(>ASB-Earth, »OrbiterOf, >ASB-Sun)` can be reified as shown in figure 6 with: $\Sigma_{\text{»Earth\_Orbits\_Sun}}$ `= {(»Earth_Orbits_Sun, »Subject, >Earth), (»Earth_Orbits_Sun, »Relation, >OrbiterOf), (»Earth_Orbits_Sun, »Object, >Sun)}` where the prefix `>ASB` stands for *AStronomical Body* and `»Belief(>NPS-Nikolaus_Kopernikus) = »Earth_Orbits_Sun`.

[5] https://www.w3.org/TR/rdf-primer/#reification

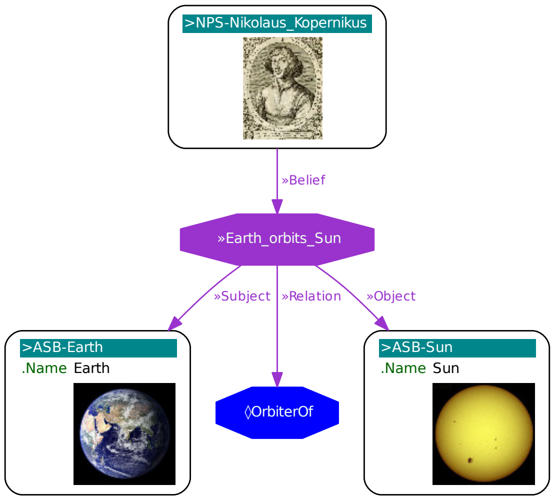


**Figure 6:** Belief of Nikolaus Kopernikus that the Earth orbits the Sun

## 6. Discussion

What are the relationships between *Object Properties* (OP) and mathematical sets? In our view, the instantiations of OPs with respect to the objects in an OP's range, create sets. For example, the definition `(^Employer, ◊hasEmployee, ^Employee)` assigns the set of employees to each employer during instantiation. This behavior does not have to be modeled explicitly but is in the nature of things. This means that object properties render a construct such as *object aggregates* superfluous in a top ontology such as BFO [1]. The cardinality of OPs can be specified explicitly by specifying `minCard` and `maxCard` information. An existential dependency can then be specified by `minCard > 0`. OPs basically form unsorted sets. The sorting can be achieved using the OPs `»first`, `»next`, `»prev` and `»last`. Further basic role types are `»Result` and `»Cause`. They establish a causal relationship between processes and the circumstances resulting from the execution, e.g. `(°°Marrying, »Result, ^^Marriage)` and this results in the inverse relationship `(^^Marriage, »Cause, °°Marrying)`. This also allows the thematic roles described in [10] and the social roles described in [5] within the GFO top ontology to be covered. The following example is based on the publication [16] and shows a part of the modeling of a school. Each of the CRS definitions can be regarded as a building block which can be used in any other CRS and thus supports modularization.

- $\Theta_{\text{SchoolEnrollment}} = \{(\Diamond\text{School}, \hat{}\text{School}), (\Diamond\text{Student}, \hat{}\text{natPerson})\}$
- $\Theta_{\text{SchoolEmployment}} = \{(\Diamond\text{School}, \hat{}\text{School}), (\Diamond\text{Teacher}, \hat{}\text{natPerson})\}$
- $\Theta_{\text{SchoolClass}} = \{(\Diamond\text{Class}, \hat{}\text{ClassOfCourse}), (\text{»or}\Diamond\text{Student}, \hat{}\text{natPerson})\}$
- $\Theta_{\text{CourseOffering}} = \{(\Diamond\text{School}, \hat{}\text{School}), (\Diamond\text{Class}, \hat{}\text{ClassOfCourse}), (\Diamond\text{Course}, \hat{}\text{Course})\}$
- $\Theta_{\text{CourseEnrollment}} = \{\Theta_{\text{SchoolEnrollment}}, \Theta_{\text{CourseOffering}}\}$
- $\Theta_{\text{CourseTeacherAssignment}} = \{\Theta_{\text{SchoolEmployment}}, \Theta_{\text{CourseOffering}}\}$

So, in summary the set of CRSs for the school modeling is: $\Theta_{\text{School}} = \{\Theta_{\text{StudentClass}}, \Theta_{\text{CourseTeacherAssignment}}, \Theta_{\text{CourseEnrollment}}\}$. Compared to the modeling in [16] we see no need for annotations like `«relator»` and `«mediation»` and the cardinalities and type of involvement of the engaged object properties can be expressed more flexible. E.g., we can explicitly specify in addition constraints like ‘at least one’, ‘all or none’, ‘exclusive or’, ‘optional’ (zero or one connection), ‘any but not all’, ‘mandatory’, ‘as many as’ etc. In top-level ontologies such as BFO, GFO, DOLCE and UFO we see no structuring methods comparable to Cascaded Role Sets (CRS). What is described as “clusters of relational properties that hang together by a nexus” in [3], page 4, has been assigned a concrete formalization with CRSs. Withs CRSs it can be specified not only which entities of a unique given kind play a role but also how many (*relational contingent sortal*).

The CRS roles `»and`, `»nand`, `»any`, `»integer`, `»or`, `»xor`, `»nxor` and `»opt` that we use allow us to specify existential dependence relations in a more general form. We have described this using the example of `∘∘Marrying` in the figures 3 and 4. The relators `»MRG_BT1` and `»MRG_BT2` absolutely require the existence of a married couple by `^^Spouses` linked by an `»and` role. Below `^^Marriage` “cascaded” comes the definition of the `^^Spouses`, which in turn, through the `»and-Role`, absolutely requires the existence of ◊`Husband` and ◊`Wife`. Based on these definitions, after performing `∘∘Marrying`, it can be checked whether the knowledge base is complete from this perspective. By defining procedural roles through cascaded role sets (figures 1, 2, 3), we have shown that temporal processes, events and complex state of affairs can be formally defined. These meet basic requirements for *variable embodiment* [17] and make it possible, for example, “to monitor the persistence of organizations through change”. Processes can be grouped into hierarchies of sub-processes and also the ‘causes’ and ‘results’ of processes can be modeled.

**Linguistics Engineering:** In line with [11], we have shown that the meaning of words can be formally defined using *Cascaded Role Sets* (CRS). We used CRSs to model *Concept Binary Trees* (CBT) and to compose terms and phrases: With the processual role set `∘∘Defining`, *Word Sense Definitions* (WSD) can be created in an easily readable form, such as `Def (word_sense) = ((one, aspect), (meaning, (a, word)))`. These can be used to create term taxonomies by `∘∘Differentiating` using `^^GenDif`. We have shown that the ontological concepts ‘air’, ‘pressure’, ‘gauge’ etc. can be derived from a linguistic definition of the term ‘barometer’ and that, conversely, the ontological terms can be applied for the WSD of terms. In contrast to PWN, we not only apply the commonly used semantic relations ‘hypernym’, ‘hyponym’, ‘synonym’, ‘antonym’ etc. Modeling the terms by means of CBTs and GDPs extends significantly beyond *SynSets*, taxonomies and troponyms and also models the connection between semantic primes and compounds from which language terms can be internally composed. Since linguistic concepts can be modeled precisely by CBTs, compared to [7] they can be used for a more exact *polysemy detection* by applying the Concept Congruence Axiom 1.1. In contrast to PWN, our methodology also enables a *Deep Semantic Search* (DSS) that takes advantage of the composition of terms from others. For example, deep searching using queries like ‘air pressure’, ‘air gauge’, and ‘pressure gauge’ answers with the concept ‘barometer’, while PWN only finds the basic terms involved[6]. On the other hand, DSS also returns the ‘barometer’ concept for all search terms combinations. At the same time, it is possible to express the *Basic Linguistic Symbols* (BLS) such that each entry of a controlled vocabulary can be utilized in any language. In addition, the WSDs can be stored in

[6] http://wordnetweb.princeton.edu/perl/webwn?s=barometer

all languages, for example, with `(>BLS-barometer, .Def_EN = (gauge, (pressure, air)))` or `(>BLS-barometer, .Def_DE = (Meßinstrument, (Druck, Luft)))`. This also means that the DSS can be performed using combinations of words from any language. The *Ontolex Lemon Lexicography Module* (OLLM)[7] is “targeted at the representation of dictionaries and any other linguistic resource containing lexicographic data, and addresses structures and annotations commonly found in lexicography”. We think that `isLexicalizedSenseOf` is synonym to what we have introduced as `»SymbolOf`. With `subComponent` they can arrange lexicographic components in hierarchies. Our modeling of semantic compounds by the GDPs and CBTs extents the possibilities described in OLLM.

## 7. Conclusion

To date, top ontologies have focused on class hierarchies. We regard Cascaded Roles Sets (CRS) as a supplementary approach for the implementation of common Object Design Patterns (ODP). The logic for their application is already specified by the CRS object properties employed. In terms of building controlled vocabularies, we consider CRSs to be essential, including the Genus-Differentiae pattern `^^GenDif` and the more general `^^ConceptBinaryTree` for building Concept Binary Trees (CBT). In addition, the correct application of the reification can be ensured by means of the `∘∘Reification`. CRSs have been designed to support a maximum degree of modularization and reusability. In addition, based on the participation of the roles defined for the object properties they may be mandatory, partially mandatory or optional. This for the execution of processes and functions allows to check whether the requirements for the role participants are met before a transaction is executed. If not, the transaction must not be started. After a transaction is terminated, requirements can again be checked against the CRS definitions. If they have not be satisfied, the transaction has to be rolled back. By applying CRSs the prerequisites have been created for formally modeling complex state of affairs and, at the same time for the monitoring the correctness of ontologies through changes.

**Future work:** Moving forward, we aim to investigate how CRSs can be applied to the numerous ODPs on the ontologydesignpatterns.org website in order to increase their expressiveness, usability and interoperability. Ultimately, it would be desirable to have a catalog of interoperable ODP building blocks that can be used to plug together complex modeling patterns. We are aware that the logic and constraints that we model with CRSs could also be modeled with OWL constraints. However, we think that modeling in OWL is much more demanding. It forces the modeler into modeling restrictions using `SubClassOf` clauses which seems not to be an obvious metaphor for modeling ODPs. It may still be obvious that one can map the Boolean operators ‘and’, ‘or’, and ‘not’ onto the union, intersection, and difference sets. However, it is no longer so obvious how one can define constraints such as ‘all or none’, ‘any but not all’, and ‘as many as’. Even if successful, it will not be trivial to infer the meaning of the constraints easily from the OWL source code. Therefore, we also want to investigate how OWL constraints for ODPs can be automatically derived from CRS design patterns.

[7] https://www.w3.org/2019/09/lexicog/

## References


[1] R. Arp, B. Smith, A. D. Spear, Building Ontologies with Basic Formal Ontology, The MIT Press, London, England (2015). ISBN: 978-0-262-52781-1.

[2] K. Fine, Things and their parts, Midwest Studies in Philosophy, 23(1) (1999) 97–105.

[3] G. Guizzardi, G. Figueiredo, M. M. H. G. Poels, Ontology-Based Model Abstraction, IEEE Thirteen International Conference on Research Challenges in Information Science, Brussels (2019). URL: https://www.researchgate.net/publication/332290797_Ontology-Based_Model_Abstraction. doi:10.1109/RCIS.2019.8876971.

[4] J. F. Sowa, Thematic Roles (2012). URL: http://www.jfsowa.com/ontology/thematic.htm.

[5] F. Loebe, Abstract vs. social roles - Towards a general account of roles, Applied Ontology 2 (2007) IOS Press (2007) 127–158.

[6] Proceedings of the 11th GlobalWordnet Conference (2021). URL: https://aclanthology.org/volumes/2021.gwc-1/, iSBN: 978-9-464027-31-0.

[7] A. A. Habibi, B. Hauer, G. Kondrak, Homonymy and Polysemy Detection with Multilingual Information, [6] (2021) 26–35. URL: https://aclanthology.org/2021.gwc-1.4/.

[8] H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, H. Michalek, General Formal Ontology (GFO) (2006). URL: http://www.onto-med.de/publications/2006/herre-h-2006-a.pdf.

[9] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider, Sweetening Ontologies with DOLCE (2002). URL: https://www.researchgate.net/publication/221630979_Sweetening_ontologies_with_DOLCE. doi:10.1007/3-540-45810-718.

[10] J. F. Sowa, Conceptual Graphs, Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (eds.), Handbook of Knowledge Representation, ISBN 9780444522115, Elsevier (2008) 213–237.

[11] P. N. Johnson-Laird, Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness, Harvard University Press. (1983). ISBN: 0521273919.

[12] F. Bond, P. Vossen, J. McCrae, C. Fellbaum, CILI: the Collaborative Interlingual Index, Proceedings of the 8th Global WordNet Conference (GWC) (2016) 50–57. URL: https://aclanthology.org/2016.gwc-1.9.pdf.

[13] H. Bense, The Unique Predication of Knowledge Elements and their Visualization and Factorization in Ontology Engineering, Kutz O, Garbacz P (eds.), Proceedings of the Eighth International Conference (FOIS 2014), Rio de Janeiro, Brazil, Sept. 22-25, 2014 (2014) 241–250. URL: https://ebooks.iospress.nl/publication/37972. doi:10.3233/978-1-61499-438-1-251.

[14] G. Guizzardi, J. P. A. Almeida, Stability Patterns in Ontology-Driven Conceptual Modeling und Demystifying Ontology (2020). URL: https://www.researchgate.net/publication/344525031_Stability_Patterns_in_Ontology-Driven_Conceptual_Modeling.

[15] N. Guarino, C. A. Welty, An Overview of OntoClean, In Steffen Staab and Rudi Studer, eds., The Handbook on Ontologies. Berlin:Springer-Verlag (2004) 151–172.

[16] G. Guizzardi, A. B. Benevides, C. M. Fonseca, D. Porello, J. P. A. Almeida, T. P. Sales, UFO: Unified Foundational Ontology, Applied Ontology 1 (2021) 1-3 (2021). URL: https://www.researchgate.net/publication/355735118_UFO_Unified_Foundational_Ontology.

[17] R. Ferrario, C. Masolo, D. Porello, Organisations and Variable Embodiments, FOIS 2018, 10th International Conference on Formal Ontology in Information Systems, 17th-21th September (2018). URL: https://www.academia.edu/37399751/Organisations_and_Variable_Embodiments.