Pattern Alternatives for Referring to Multiple Indirectly Specified Objects

Vojtěch Svátek[1], Ján Kľuka[2], Miroslav Vacura[1], and Martin Homola[2]

[1] University of Economics, Prague, W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic
{svatek,vacuram}@vse.cz
[2] Comenius University in Bratislava, Mlynská dolina, 842 48 Bratislava, Slovakia
{homola,kluka}@fmph.uniba.sk

Abstract. We introduce the general problem of capturing references to multiple objects that are indirectly specified via their type and/or relationship to an explicit object. We describe the setting in an abstract way, provide three alternative patterns for representing it in OWL (based on an existential restriction, a placeholder individual, and a shortcut property) and discuss their comparative advantages and disadvantages. The patterns are also aligned with the closest one among the popular logical pattern families, classes as property value, and other related research.

1 Introduction

A common requirement in data modeling is to refer to the existence of entities whose *identity*, but possibly also *number*, remains unspecified; we only know their common *type*. This modeling problem most typically appears when the original relationship refers to something possibly happening in the *future*: for example, a company *offers to sell* an unspecified number of physical products of some (catalog) type, or some activity *is scheduled to be carried out* by multiple agents of a certain type (i.e., whose identity and number is unknown when making the statement). A variation of the setting is obtained by replacing the common type with a common (individual) entity to which the unspecified entities are *related*. E.g., the company would be offering an unspecified number of products *produced by* a certain manufacturer, or the activity would be scheduled to be carried out by agents *employed by* a certain department. We can generalize both cases to the problem of referring to *multiple indirectly specified objects* (MISO), and will distinguish between them through the acronyms MISO-T (indirect specification via a *type*) and MISO-R (indirect specification via *relationship* to an object).

An intuitive way of expressing a relationship to anonymous entity/entities in semantic web terms is to employ an *existential or cardinality restriction* over the respective property. It allows to qualify the anonymous entity/entities with their *class*, thus addressing the MISO-T problem; the MISO-R variant can be handled by putting the related object into a *value restriction* in the filler of the existential/cardinality restriction. This modeling method can also satisfy MISO-T and MISO-R at the same time, since the existential/cardinality restriction filler can be a conjunction of a named class and the value restriction. However, these straightforward approaches have certain shortcomings. First, the existential restriction cannot distinguish between the setting with a

single (merely unknown) object and that with multiple (possibly quite many) objects. Higher-cardinality restrictions, in turn, only express an *exact* (maximal and/or minimal) number of anonymous entities to be specified.[3] A further issue with this kind of modeling is that it is beyond the expressiveness of RDFS, which simple linked data vocabularies often target; restrictions with higher cardinality are actually even beyond the expressiveness of some OWL profiles.[4] In this paper we attempt to go beyond these simple solutions by proposing a family of three patterns in variants for both MISO-T and MISO-R, considering:

- pattern design based on a unifying *background model*,
- possibilities to approximate the *cardinality* of the relationship considered,
- generic logical patterns in combination with *naming/annotation* patterns and *vocabulary reuse* considerations,
- interconnection to previously published patterns, in particular, the CPV pattern [6].

While one of the patterns is only a mild extension of the existential restriction approach, the remaining two structurally differ from it in their core.

The paper is structured as follows. Section 2 analyzes the overall situation to be modeled and explains why its OWL representation is not obvious. A short Section 3 reviews the inventory available when designing an ontology pattern for OWL. Section 4 presents the three alternative modeling patterns, including axiomatization and examples. Section 5 mentions some real (though implicit) usage of the patterns. Section 6 summarizes the patterns, points out their differences and outlines criteria for choosing among them. Section 7 compares the studied problem with pre-existing research. Finally, Section 8 concludes the paper and drafts the prospects for further research.

2 MISO: background model

The modeling situation we target can be characterized as follows, specifically for the MISO-T model: *There is a distinguished object that is related, by the same kind of relationship, to (possibly even one, but usually) multiple undistinguished objects of a certain type.* We can depict this situation as in Fig. 1.

The diagram conforms to the principles of the *PURO* method of ontological background modeling, introduced in [10,11]. Without going into details of this method, we can summarize that it aims to make explicit the ontological distinctions of 'particular vs. universal' and 'object vs. relationship (between objects) that exist behind existing, or newly designed, OWL ontologies and vocabularies. We presume that these distinctions are mostly correctly sorted out in the heads of ontology designers (as 'background model') but get obscured when the ontology ('foreground model') is crafted in OWL as language with limited expressiveness, or because of computational efficiency concerns in logical inference and linked data management.

[3] Some may argue that machine processing of semantic web data mostly relies on exact entity counts. We however foresee scenarios where even a fuzzy specification of the number would help; one of them, that of data *verbalization*, in discussed in the Conclusions of the paper.

[4] `https://www.w3.org/TR/owl2-profiles/`

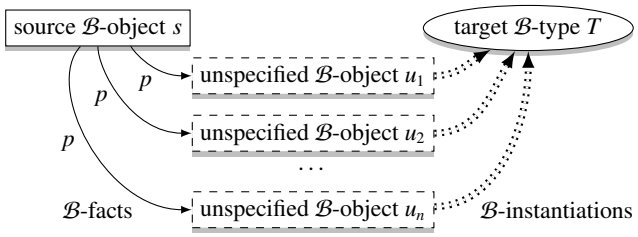

Fig. 1. Background schema of a MISO-T structure

The diagram depicts an unspecified number of 2-chains of relationships, the first, labeled as p, always being a 'fact' and the second an 'instantiation' (in PURO terms, we use the notions of $\mathcal{B}$-fact and $\mathcal{B}$-instantiation, so as to stress their 'background' nature and avoid mismatch with the analogous notions in OWL as 'foreground' language); the source object and the target type (again, $\mathcal{B}$-object and $\mathcal{B}$-type) are the same in all chains.

The diagram for MISO-R would be same, except that instead of the target type T we would have a target object t and each of the connections of an u_i to it would be another fact, labeled as q, rather than an instantiation.[5]

A natural and most 'semantically faithful' choice (in the sense of preserving the essential distinction of 'particular vs. universal') for representing the source object and target type in OWL would be to use an OWL individual and an OWL class, respectively. Since the explicit representation of the undistinguished entities is not obvious, the modeling problem can be easily perceived as that of connecting an individual (s) to a class (T); this is however somewhat tricky in OWL. The pattern – more precisely, pattern family – providing guidance in the 'classes as property values' (CPV) problem has been first coined by the W3C Ontology Engineering and Patterns a decade ago [6]. In our previous work [10] we analyzed the CPV pattern family from the W3C note [6] in depth (using the PURO apparatus) and identified that its members have different 'affinity' to expressing three different 'states of affairs': relating the individual (1) to an abstract topic derived from the class, (2) to the intension of the class, and (3) to the extension of the class. We will briefly revisit this analysis, specifically for the 'multi-instance fact' problem, in Section 7 of this paper. However, in the main part of the paper, we start from the background state of affairs (as in Fig. 1) rather than from a problem manifested on the surface and possibly having different causes.

3 Pattern modeling inventory

Before explaining in detail the members of the proposed pattern family, we will first briefly recapitulate the inventory available for an (OWL-based) ontology pattern designer. Different elements of this inventory will be then employed in the individual alternative MISO patterns in Section 4.

[5] We colloquially refer to this common MISO structure as to the '*slingshot* pattern': with some imagination, each of the unspecified objects can be seen as connected by the 'two portions of the rubber strip' to the tips of the Y-handle (source object and target type/object, respectively), and corresponding to the changing position of the 'ball' during the firing event.

Obviously, first of all, since OWL is a language with rigorous logical semantics, the pattern has to be expressed in *logical terms*. The advent of the semantic web and OWL, during which the ontology pattern repositories such as the W3C SWBPD WG collection,[6] the Manchester-based catalog[7] and the Rome-based portal[8] emerged, was dominated by purely logical interpretation of ontology structures. Furthermore, the tiny part of the popular RDF-centric catalog developed for linked data designers[9] that is devoted to ontological (in the sense of OWL T-box) modeling issues, such as the 'N-ary relation pattern', is also confined to the 'logical space'.

However, *naming conventions* gradually came into light in the last couple of years [1,8,9,12]. They can serve not only as guidance for a human inspecting the ontology (and associated data) but also as subject of automated analysis aiming to reconstruct the ontological background (as we refer to it in this paper). Therefore they are prone to make a meaningful part of any ontological pattern design effort.

Yet another kind of space for entering important information into OWL ontologies and knowledge bases are *structured annotations*. The topic of extending the expressiveness of OWL ontologies using structured canonical annotations would deserve a separate study. We believe that the provision for rich annotations embedded in OWL 2 has not yet been deployed to its full potential, and that annotations can become an integral part of best practice patterns for ontological modeling.[10]

Finally, rather than advising the pattern users how to design new ontological entities based on generic patterns, it is also possible to let them *reuse specific entities* from existing ontologies and vocabularies, especially those popular on the linked data web.

4 MISO pattern family

We outline and exemplify three alternative patterns in its both MISO-T and MISO-R variants. While the first alternative is a simple extension of the baseline approach using an existential restriction, the other two are novel, though inspired by their implicit use in common linked data vocabularies. As instructive examples we will use artificial ones from the domain of selling cars (as one in which semantic web approaches recently started to proliferate).

4.1 Alternative 1: Existential restriction with annotation

The *logical* part of the existential restriction MISO-T pattern simply states that the source individual s is an instance of an anonymous class defined by an existential restriction on property p, such that the filler of the restriction is class T. Existential restriction is a construction well anchored in the OWL standard (it is part of even simple dialects of the language[11]) as well as in ontological modeling practice.

[6] `http://www.w3.org/2001/sw/BestPractices/OEP/`

[7] `http://odps.sourceforge.net/`

[8] `http://ontologydesignpatterns.org/`

[9] `http://patterns.dataincubator.org/book/`

[10] Preliminary ideas for such deployment have been outlined in [13].

[11] This, and the fact that exact cardinality statements would be misleading when actually meaning 'vague' multiplicity, is the reason why we do not consider other cardinality restrictions here.

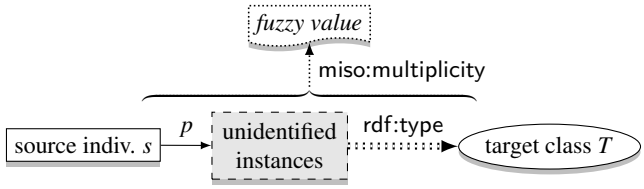

Fig. 2. Alternative 1: existential restriction MISO-T pattern

Since the existential restriction pattern cannot make the multiplicity of the relationship to anonymous instances explicit at the *logical* level, we leverage on the possibility, in OWL 2, to *annotate* the existential axiom with a specific annotation property. This property, provisionally named miso:multiplicity, indicates that the whole axiom annotated by it is a reference to multiple objects, and its value declares the fuzzy 'linguistic' cardinality of the set of unidentified instances. This value could be a simple literal (as in examples below) or, possibly a value from a dedicated classification that could also include fuzzy numerical intervals.

Figure 2 depicts the inferential product of the axiom rather than its syntactical form:[12] there is a constructed 'skolem' node representing the unindentified instances of *T* linked from *s* via property *p*.

The syntactical form of the (extended part of the) pattern in Manchester syntax is as follows:

```
Individual: s
  Types:
    Annotations: miso:multiplicity value
    p some T
```

where *value* is the fuzzy value, for instance, `"many"^^xsd:String`. For example:

```
Individual: Shop123
  Types:
    Annotations: miso:multiplicity "many"^^xsd:String
    sells some VolvoXC90
```

When porting this approach to the MISO-R scheme, the complexity of the pattern increases since, instead of a named class, the filler of the existential restriction now itself becomes a *value restriction*, i.e., an existential restriction valued with an enumerated class containing a single individual. Syntactically we can display it as:

```
Individual: s
  Types:
    Annotations: miso:multiplicity value
    p some (q value T)
```

exemplified as

[12] We use this convenient form of depiction so as to remain coherent with [6], at the cost of deviating from the syntactical structure of the anonymous 'existential' class.

```
Individual: Shop123
  Types:
    Annotations: miso:multiplicity "many"^^xsd:String
    sells some (manufacturedIn value Thailand)
```

While this still appears neat in Manchester syntax, at the level of, e.g., the Turtle RDF syntax the structural complexity increase would be quite obvious. Note also that the nesting of restrictions, and, especially, the use of nominals (enumerated classes with a single individual) would sweep the ontology away from certain OWL profiles such as OWL 2 QL.

The types of class expression can also be combined, as in:

```
Individual: Shop123
  Types:
    Annotations: miso:multiplicity "many"^^xsd:String
    sells some ( VolvoXC90 and ( manufacturedIn value Thailand ) )
```

4.2 Alternative 2: Linking to a placeholder individual

When downgrading the previous model into an OWL sublanguage disallowing existential restriction, such as RDFS, one modeling option at the logical level could be a blank node, corresponding to the result of evaluating an existential restriction in tableau reasoning. Via multiple blank nodes, an approximation of the diagram from Fig. 1 could be reconstructed, however, again for a fixed, exact cardinality (as with the higher-arity OWL cardinality restrictions, to whose product this would correspond). Moreover, using blank nodes outright as truly undistinguished (rather than just unknown to date) entities is odd.

As a pragmatic approach to explicitly modeling the 'intermediate objects' between the source object and target class/object of the MISO structure we thus propose to create a *named individual* representing all the multiple undistinguished objects as their common *placeholder* (or, possibly, 'proxy'). The placeholder then needs to be, to achieve the MISO-T effect, classified in two ways:

- First, it should declare the type of the undistinguished individuals it represents, i.e., link to the *target class*.
- Second, to avoid being mistaken for a real instance of the target class, it has to be either typed by a dedicated 'utility' class (we call it thereafter as *MI_class*) unifying all such placeholders, or equipped with an adequate property (presumably, an annotation one). We opted for the first option in our design.

Furthermore, we can express the 'fuzzy multiplicity' as in the previous approach, by assigning the placeholder individual the corresponding annotation property.

The depiction of the pattern in the MISO-T variant is in Fig. 3. Regarding the choice of *MI_class*, the designer can opt for:

- reusing the generic class from the pattern namespace; we provisionally suggest to call it miso:SomeInstances;

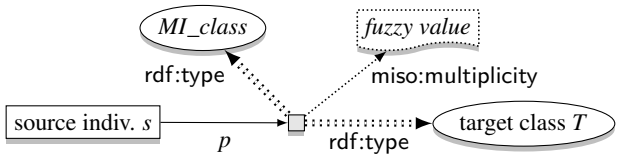

Fig. 3. Alternative 2: placeholder individual MISO pattern

- reusing a class from another namespace (below we discuss the gr:SomeItems class from the popular GoodRelations vocabulary);
- coining a proprietary class for the given domain; then the *name* of the class should indicate the multiplicity, for example, in the medical domain one could propose a class called SomePatients;
- omitting the class entirely and only using the annotation property.

The syntactical form of the pattern in Manchester syntax is as follows:

```
Individual: s
  Facts: p _:A
Individual: _:A
  Types: T
  Types: miso:SomeInstances
  Annotations: miso:multiplicity value
```

where *value* is again a suitable fuzzy value and *A* is a name of the placeholder individual. In our example

```
Individual: Shop123
  Facts: sells _:A
Individual: _:A
  Types: VolvoXC90
  Types: miso:SomeInstances
  Annotations: miso:multiplicity "many"^^xsd:String
```

The modification of the pattern for MISO-R is simple this time. Since we work in the Abox space, we can just replace the instantiation with a property assertion:

```
Individual: s
  Facts: p _:A
Individual: _:A
  Facts: q T
  Types: miso:SomeInstances
  Annotations: miso:multiplicity value
```

exemplified as

```
Individual: Shop123
  Facts: sells _:A
Individual: _:A
  Facts: manufacturedIn Thailand
  Types: miso:SomeInstances
  Annotations: miso:multiplicity "many"^^xsd:String
```

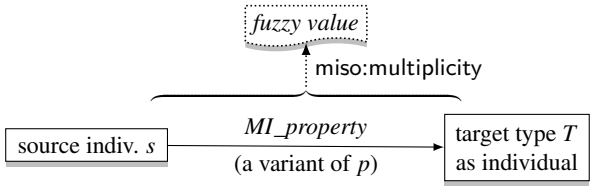

Fig. 4. Alternative 3: the shortcut property MISO pattern

Compared to the existential restriction approach, the placeholder approach has the advantage of remaining (aside the multiplicity annotation) within RDFS. The combination of statements is still possible (just moving from the Tbox level to the Abox level, compared to Alternative 1):

```
Individual: Shop123
  Facts: sells _:A
Individual: _:A
  Facts: manufacturedIn Thailand
  Types: VolvoXC90
  Types: miso:SomeInstances
  Annotations: miso:multiplicity "many"^^xsd:String
```

4.3 Alternative 3: Shortcut property with name and annotation

The third option is to 'fold' the whole central part of the MISO structure into an instance of a new property, say, *MI_property*, which directly connects the source individual with the target type/object. This allows to avoid both anonymous classes and placeholder individuals. On the other hand, some consequences of this variant are:

- The target type, in the case of MISO-T, can no longer be a syntactic class in OWL-DL: its meta-modeling with an individual is required, at least in the form of OWL 2 punning (i.e., with the same identifier as the class, yet treated as a separate entity in the inferential model).
- The new, 'shortcut' property has similar but not identical meaning to the original property *p*: it differs by its 'multiplicity' and it further includes the instantiation links (for MISO-T) or represents the chaining of two facts (for MISO-R).
- Unlike the previous two approaches, it is not possible to combine multiple pieces of information about the indirectly specified objects.

The multiplicity can be indicated and 'quantified' in the same way as in the previous alternatives: using the property miso:multiplicity, this time, however, with a property assertion in its subject. The pattern schema of the MISO-T variant is in Fig. 4. It can be further extended by a similar logical construction as in the placeholder approach: the MI property as such can be declared to an instance of a meta-class miso:MultiInstanceProperty and a subproperty of a general multi-instance property, miso:multiInstanceProperty.

The Manchester syntax form of the logical part of the MISO-T pattern variant is as follows:

```
Individual: s
  Facts:
    Annotations: miso:multiplicity value
    MI_Property T
```

given again some suitable fuzzy value *value*.

The car selling example may then look, e.g., as follows:

```
Individual: Shop123
  Facts:
    Annotations: miso:multiplicity "many"^^xsd:String
    sellsType VolvoXC90
```

For MISO-R we yield basically the same scheme, except the target individual instead of the class:

```
Individual: s
  Facts:
    Annotations: miso:multiplicity value
    MI_Property T
```

exemplified as

```
Individual: Shop123
  Facts:
    Annotations: miso:multiplicity "many"^^xsd:String
    sellsManufacturedIn Thailand
```

As noted above, since the set of indirectly specified objects is truly implicit and only 'packed' inside *MI_property*, no information can be externally provided to them. By combining the elements from the MISO-T and MISO-R examples we can only state that Shop123 sells many Volvo XC90 cars and many (possibly completely different) products from Thailand. The only freedom we have is that of creating a hierarchy of specialized properties, which could be manageable at the level of a few high-level distinctions such as 'sells cars' or 'sells from offshore provenance' but not in full scale.

5 Implicit usage of the patterns on the linked data web

Since the first alternative requires the use of an existential restriction (anonymous class) at the instance dataset level, we do not assume it to be commonly used in the linked data cloud, but rather in the Abox knowledge bases used inside DL-based servers. The pattern does not manifest itself at the Tbox level and thus cannot be detected merely by analyzing ontologies as such. For this reason we only examine the usage of the remaining two alternatives, which are amenable to linked data. The observed usage also does not attempt to express an explicit multiplicity annotation.

As regards Alternative 2, the notion of multiple-instance placeholder has been first coined by M. Hepp in the GoodRelations (GR) ontology, on which the e-commerce module of schema.org is based. The reason was detailed in Ch. 3.3.3 of the GoodRelations techreport [4]: to be able to reuse the same datasheet information for product

models, individual physical products and mass-offered products. Later on, GR has been included into the currently perhaps most popular linked data (and web markup) vocabulary, which is *schema.org*, as its e-commerce component. By simplifying the example snippet from the GR specification,[13] we get (in Turtle RDF syntax):

```
foo:offer a gr:Offering;
    gr:includes foo:product .
foo:product a gr:SomeItems;
    gr:name "Canon Rebel T2i (EOS 550D)"@en .
```

In terms of MISO-T, foo:product corresponds to the multiple-instance placeholder and gr:includes to p. Type T is not present in the snippet, but it could easily be, e.g., class obk:DigitalCamera from the GR-compliant Digital Camera Vocabulary,[14] i.e.:

```
foo:product a obk:DigitalCamera.
```

Translated to the Manchester syntax, for better comparison with Alternative 2 of the MISO-T pattern, the snippet would then look as follows:

```
Individual: foo:offer
  Facts: gr:includes _:A
Individual: _:A
  Types: obk:DigitalCamera
  Types: gr:SomeItems
```

According to the GoodRelations statistics service from June 2014,[15] the placeholder class gr:SomeItems was used nearly 86 thousand times. In schema.org the class has been adopted under the name SomeProducts.[16] The entity pages indicates its usage "Between 1000 and 10,000 domains".

As regards Alternative 3, traces of this approach can be observed, again, in the GR ontology (and thus also schema.org). It allows, for instance, to restrict an offering of a product or service to (undistinguished) possible customers from a certain category, namely to public institutions, as follows:

```
ex:offering231  a  gr:Offering ;
    gr:eligibleCustomerTypes gr:PublicInstitution .
ex:PublicInstitution a gr:BusinessEntityType .
```

The OWL individuals ex:offering231 and gr:PublicInstitution correspond to the source individual s and target type T, and the gr:eligibleCustomerTypes property assertion is the shortcut property: what is actually meant here is that the eligible customers are the unspecified number of unknown instances of gr:PublicInstitution. The individual gr:PublicInstitution is inherently not an individual object but a type; its class, gr:BusinessEntityType, also referred to in the snippet, is thus (e.g., in the sense of PURO) a

[13] http://www.heppnetz.de/ontologies/goodrelations/v1.html#SomeItems

[14] http://purl.org/opdm/digitalcamera

[15] The statistics tool was then available from http://goodrelations-stats.appspot.com/; it is not functional at the time of writing this paper.

[16] http://schema.org/SomeProducts

meta-type. In schema.org the same kind of property is called eligibleCustomerType[17] (the only difference being in the singular form 'type', which is probably more appropriate). Its reported usage is again "Between 1000 and 10,000 domains".

A similar case is the gr:acceptedPaymentMethods property (linking an offer to a payment method such as gr:PayPal), again appearing in schema.org with a singular ending.[18] Its reported usage is however smaller, "Between 10 and 100 domains" only. On this example we can see that the surface (OWL) patterns, in terms of the MISO-T/ MISO-R variation, need not always correspond to the underlying background model. For example, instead of an acceptedPaymentMethod property (which we above ascribed to MISO-R) there could also be an OWL class such as PaymentByPayPal to which an individual acting as multiple-instance placeholder would be assigned (jointly with the 'SomeObjects' class or similar), yielding the Alternative 2 of MISO-T. On the other hand, in some contexts the eligibleCustomerTypes property (ascribed to MISO-T) could have the background model of MISO-R, for example, if the types of customers were induced by their specific relationship to a specific country or institution. This phenomenon conforms to the observation that motivated the design of the PURO approach: the same background model can be expressed using different, more or less semantically faithful, OWL variants.

6 Overview and selection criteria

Table 1 summarizes the pros and cons of the three alternatives.

Table 1. Overview of the alternative patterns

Approach	*Pros*	*Cons*
Existential	Modeling flexibility	Beyond RDFS Multiplicity only hinted by annotation
Placeholder	Modeling flexibility Structural simplicity	Placeholder individuals may mix with normal ones
Shortcut property	Structural simplicity (highest)	Type meta-modeled by individual in MISO-T Hardcoded semantics – property proliferation

Alternative 1 (existential restriction) makes sense when designing ontologies that are assumed to be processed by OWL-aware tools. The MISO-R variant (or a combination of both) is more demanding for the reasoners than the MISO-T variant alone. The logical part of the pattern is sound for them even without understanding the annotation properties; however, the multiplicity aspect is not obvious then.

The remaining alternatives come into play when we do not expect the data consumer to apply an OWL reasoner, which is typically the case in linked data, as mentioned in

[17] http://schema.org/eligibleCustomerType
[18] http://schema.org/acceptedPaymentMethod

the previous section. They both allow to keep the vocabulary structurally simpler, free of anonymous classes, which may potentially discourage linked dataset designers from beyond the core semantic web community. It is however at the cost of introducing extra 'utility' entities.

Alternative 2 (placeholder) is suitable when we want to retain the possibility of arbitrarily assigning features (as classes or property assertions) to the indirectly specified objects. More specifically it allows to transfer default values, originally assigned to a 'model' such a product datasheet, through an abstract superclass, to indirectly specified objects. MISO-R does not bring extra complexity to MISO-T, since in both cases the 'feature assignment' is merely an addition of an Abox axiom.

Finally, alternative 3 (shortcut property) is most parsimonious in terms of knowledge base size, and in simple cases the modeling may appear most straightforward to inexperienced dataset designers. However, due to its hard-coding approach, it appears only usable if there is a single prominent feature to be assigned to the indirectly specified objects. Since such a feature is more likely to be a type rather than just a fact, the MISO-R variant usage of this alternative is presumably less common compared to MISO-T; no combination of them is of course possible.

7 Related research

We are aware of three pieces of research that are tighly or more loosely related to the MISO pattern proposal: the GoodRelations ontology, the CPV pattern, and the Template instance pattern.

GoodRelations and schema.org We reuse the GoodRelations notion of 'SomeItems' (discussed in the previous section) as is, for our Alternative 2, and only consider it (1) independently of the e-commerce domain, (2) as one element of the larger MISO framework, and (3) possibly extended with a fuzzy-linguistic quantifier in an annotation property. Regarding the first point, although the SomeItems class is defined in a domain-specific *vocabulary*, we assume that it is sufficiently generic to be reused beyond the e-commerce (or, even, commercial offering) domain. According to its verbal specification, its instance should be understood as 'a placeholder instance for unknown instances of a mass-produced commodity.' If we relax the adjective 'mass-produced', the class could be suitable for any use case where we consider an undistinguished number of such unknown instances. Reuse of the GR term thus looks as an optimal solution in many cases, aside reusing the more generic miso:SomeInstances class from the MISO pattern or coining a new domain-specific property (but coherently with the MISO pattern). In long term, a natural unifying step might be to declare gr:SomeItems as subclass of miso:SomeInstances,[19] in the GR ontology. In schema.org the analogous class is called SomeProducts; the use of the term 'product' somewhat limits its straightforward application beyond the domain of product offering.

[19] It is worth mentioning that the deprecated predecessor of 'SomeItems' is 'ProductOrServicesSomeInstancesPlaceholder', i.e., it already including 'SomeInstances' in its string.

CPV pattern family We already mentioned that the modeled problem is to some degree related to the problem of making a class the value of a property, which has been addressed by a W3C (pattern) note [6]. Although the nature of the source and target of the MISO-T structure (object and type) clearly refers to the CPV problem, there are some differences from the way the CPV problem has been analyzed in the W3C note. First, MISO-T only amounts to one of the three sub-categories of CPV, as identified in our previous study [10]: that of relating the individual to the *extension* of the class (rather than to its intension or to an abstract topic). Only the Approach 4 of the W3C note [6] – using an (existential) property restriction with the target class as filler – fully conforms to this sub-category.[20] We can rewrite the gist of the Approach 4 A-box example in Manchester syntax as:

```
LionsLifeInThePrideBook dc:subject some Lion
```

i.e., the book 'Lions: Life in the Pride' has as subject some (at least one) unindentified lion. Second, [6] devotes specific attention to the problem of considering the target class inside of a taxonomy, which is beyond the scope of our MISO pattern analysis. On the other hand, as reflecting in its name, the MISO pattern (family) specifically accounts for the *multiplicity* of the relationship between the source object and target class.

Template instance pattern Nyulas et al. [7] proposed a pattern similar to the logical part of the placeholder variant (Alternative 2) of the MISO-R pattern, which they call Template instance (TI) pattern. Its motivation is to compress the Abox of ontologies containing relationships that are reified, because of either their higher arity or the need to annotate them with further information. While straightforward modeling makes different source subjects link to the same reified relationship through a different reifying individual, the TI pattern makes such subjects (i.e., source objects in MISO sense) share the same reifying individual (analogous to the placeholder object in MISO), provided the whole structure of the reified relationship (i.e., all target objects, in MISO sense) is uniform. There are however clear differences, again, both in motivation and implementation. The placeholder in the TI pattern represents a set of relationships while that in the MISO represents a set of objects. The TI pattern assumes multiple source objects (while MISO normally has one source object per placeholder individual) Finally, there are typically also multiple target objects in the TI pattern (some of which can even be data values); we do not assume this in MISO by default. Finally, the MISO pattern is primarily designed for efficient data publishing rather than for solving a memory-size problem as the TI pattern.

8 Conclusions

We described the modeling problem denoted as *multiple indirectly specified objects* and provided basic descriptions of three kinds of patterns for expressing it in OWL. The

[20] It should also be noted that our study [10] also outlines three new variants of the CPV pattern, of which two, called 'SomeItems' and 'Fact-Instantiation Label', are precursors of the MISO-T Alternatives 2 and 3 (without explicitly modeling the multiplicity).

pattern is, in a way, hybrid: it contains both a logical/structural description based on examples (as common for the W3C SWBPD patterns) and a proposal for reusable entities within a common namespace[21] (miso:multiplicity property, miso:SomeInstances class and miso:multiInstanceProperty). We also positioned the new problem and patterns with respect to the closest related effort in the ontology pattern field: representing classes as property values, as well as GoodRelations as its prime source of inspiration and a related Template pattern.

The work presented in the paper is not a pattern proposal ready to be immediately published as guidance for ontology designers. It is rather a focused analysis from which a recommendation such as [6] could be distilled relatively easily. The principles seem clear; however, a discussion of a larger community is required so as to decide about, at least (1) the priority order in which the three options should be presented to the reader; (2) the precise form of naming conventions as well as rich annotations – since the number of options is quite large here. Explicit *competence questions* for the patterns would probably be worth reengineering in this context.

Of course, *novel proposals* for solutions addressing the problem described here may arrive as well. For example, a potential new alternative that popped up during the final phase of writing the paper is one focusing on 'counting', as would be, in a variant of our example (with T being Car rather than VolvoXC90), a derived property numberOfSoldCars. An annotation might then, rather than the multiplicity as such (since that would already be expressed by the new property, at the logical level) express the *degree* of fuzziness of the numerical value, based on, e.g., the reliability of an external estimate or the degree of the value fluctuation in time.

While we justified the importance of the modeling problem by pointing at its occurrence in the e-commerce domain, additional *empirical investigation* is still needed in further domains. Systematic search for pattern occurrence is however difficult due to the fact that its logical and even naming aspect would typically not allow to distinguish the constellation with *multiple unspecified objects* from that of a *single unspecified object*; this distinction, which we propose to capture via structured annotation, is possibly nowadays expressed verbally in the vocabulary specifications.

Yet another thread of future work should be an analysis of *effects* following from the application of the patterns – in terms of computational efficiency, storage requirements, as well as perception of data expressed this way by users.

We also plan to elaborate on the notion of *fuzzy multiplicity* expressed via the annotation property. While we feel that such a distinction increases the value of data openly exposed on the web, we still miss an elaborated scenario in which such a (possibly contextually dependent) distinction could be exploited by automated tools, as well as the specification of the optimal form of expressing this distinction. One straightforward application could arise in ontology/dataset *verbalization*. E.g., a popular translator of OWL to Attempto Control English [5] expresses an existential restriction as the name of the filler class with indefinite article; in our running example it would be "... sells a VolvoXC90". With the help of a multiplicity annotation an informed translator could produce something like "... sells *many instances of* VolvoXC90", which would distinguish a regular car seller from someone selling, e.g., a single used car.

[21] The URI for this namespace is yet to be set up.

Finally, we plan to implement suitable patterns into the existing *suite of tools* allowing to design the skeleton of an ontology in PURO and then transform it to alternative variants in OWL [3]. This would help decrease both the intricacy of choosing between the alternatives (since the system would order them based on 'hints' inside the PURO model) and the tedium and risk of errors in specifying the individual parts of the patterns.

Acknowledgment

We are grateful to Martin Hepp, Enrico Daga and Ronald Denaux for relevant discussions or feedback to an early version of this paper.

References

1. N. A. A. Manaf, S. Bechhofer, R. Stevens: A survey of identifiers and labels in OWL ontologies. In: OWLED 2010, San Francisco, CA, USA, June 21–22, 2010. Vol. 614 of CEUR-WS
2. L. Dodds, I. Davis: Linked Data Patterns. A pattern catalogue for modelling, publishing, and consuming Linked Data, 2012. Online: `http://patterns.dataincubator.org/`
3. M. Dudáš, T. Hanzal, V. Svátek, O. Zamazal: OBOWLMorph: Starting ontology development from PURO background models. In: OWLED 2015, Vol. 9557 of LNCS, Springer.
4. M. Hepp: GoodRelations: An ontology for describing web offerings. Technical Report 2008-05-15. Online: `http://www.heppnetz.de/projects/goodrelations/GoodRelations-TR-final.pdf`.
5. K. Kaljurand: Attempto controlled english as a semantic web language. PhD thesis, University of Tartu, 2007.
6. N. Noy (ed.): representing classes as property values on the semantic web. W3C Working Group Note, April 5, 2005. Online: `http://www.w3.org/TR/swbp-classes-as-values/`.
7. C. Nyulas, T. Tudorache, S. Tu: The template instance pattern. In: WOP 2012, Boston, USA, November 12, 2012. Vol. 929 of CEUR-WS
8. D. Schober, B. Smith, S. E. Lewis, W. Kusnierczyk, J. Lomax, C. Mungall, C. F. Taylor, P. Rocca-Serra, S.-A. Sansone: Survey-based naming conventions for use in OBO Foundry ontology development. *BMC Bioinformatics* 10, 2009
9. D. Schober, I. Tudose, V. Svátek, M. Boeker: OntoCheck: Verifying ontology naming conventions and metadata completeness in Protégé 4. *J. Biomedical Semantics* 3(S-2):S4, 2012
10. V. Svátek, M. Homola, J. Kľuka and M. Vacura: Mapping structural design patterns in OWL to ontological background models. In: K-CAP 2013, Banff, Canada, June 23–26, 2013. ACM, 2013
11. V. Svátek, M. Homola, J. Kľuka and M. Vacura: Metamodeling-based coherence checking of OWL vocabulary background models. In: OWLED 2013, Montpellier, France, May 26–27, 2013. Vol. 1080 of CEUR-WS
12. V. Svátek, O. Šváb-Zamazal, V. Presutti: Ontology naming pattern sauce for (human and computer) gourmets. In: WOP 2009, Washington D.C., USA, October 25, 2009. Vol. 516 of CEUR-WS
13. V. Svátek, M. Vacura, M. Ralbovský, O. Šváb-Zamazal, B. Parsia: OWL support for (some) non-deductive scenarios of ontology usage. In: OWLED 2008, Karlsruhe, Germany, October 26–27, 2008. Vol. 432 of CEUR-WS