Ontology Design Patterns for Winston's Taxonomy Of Part-Whole Relations*

Cogan Shimizu, Pascal Hitzler, and Clare Paul

[1] Data Semantics (DaSe) Laboratory, Wright State University, OH, USA
[2] Air Force Research Laboratory, Dayton, Ohio, USA

Abstract. While the formal modeling of part-whole relationships has been of interest, and studied, in many fields including ontology modeling, as of yet there has been no dedicated ontology design pattern which goes beyond the modeling of an absolute minimum. We correct this by providing two patterns based on Winston's landmark paper, "A Taxonomy of Part-Whole Relations."

1 Introduction

Part-whole relations are of fundamental importance for how we organize concepts. Consequently, they have been studied in philosophy [1,20,19], linguistics [3,4] geographical information systems (GIS) [2,9,18], to name just a few. Corresponding *partonomies* or *meronomies*, i.e. hierarchies built from part-whole relations, are therefore a recurring theme in ontology modeling.

Despite this, however, we have been unable to find a readily available or documented ontology design pattern for part-whole relationships, other than some very minimalistic proposals in the ontologydesignpatterns.org portal. In this paper we want to rectify this by providing such a pattern, together with a contextualized version of it. Our approach to this is to keep things as simple as possible, yet to make sure that the resulting patterns are comprehensive yet general enough to be applied in many contexts.

Concretely, we will follow an approach laid out by Winston in his 1987 landmark paper on "A Taxonomy of Part-Whole Relations" [20].[3] While this paper was based on linguistic considerations, it also provided for logical characterizations and axiomatics, which will inform our pattern. As such we do not claim much novelty, other than that we cast previous observations by us and others into reuseable ontology design patterns. In fact, the technical content of Section 3 is adapted from [8] by carrying it over to the context of ontology design patterns.

---

[*] This work will be published as part of the book "Emerging Topics in Semantic Technologies. ISWC 2018 Satellite Events. E. Demidova, A.J. Zaveri, E. Simperl (Eds.), ISBN: 978-3-89838-736-1, 2018, AKA Verlag Berlin."

[3] A discussion of different such theories in the context of logical knowledge representation for ontology engineering can be found in [10].

Relation Type	funct.	hom.	sep.	Example
component-integral object	yes	no	yes	handle and cup
feature-activity	yes	no	no	paying and shopping
portion-mass	no	yes	yes	slice and pie
place-area	no	yes	no	everglades and florida
member-collection	no	no	yes	tree and forest
stuff-object	no	no	no	gin and martini

Table 1. Types of part-whole relations according to Winston. funct. stands for functional, hom. stands for homeomerous, sep. stands for separable.

The rest of the paper is organized as follows: In Section 2 we briefly review Winston's approach to lay the ground for the technical contributions. In Section 3 we provide the basic Winston-Part-Whole Pattern. In Section 4 we provide the Contextualized Winston-Part-Whole Pattern as an extention of the one presented in Section 3. In Section 5 we describe a usage scenario. In Section 6 we briefly discuss a provenance pattern as an example for contextualization, which is essentially adapted from the core of the PROV-O ontology. Section 7 contains additional release information for the patterns, and Section 8 concludes.

2 Winston's Approach

Winston in [20] distinguishes six different types of part-whole relationships. His categorization is based on the following three aspects, a different selection of which holds for each of the types.

separable (versus inseparable): Parts can in principle be physically disconnected from the whole.

functional (versus non-funcational): Parts are in specific spatial and temporal position relative to each other which supports their functional role as parts of the whole.

homeomerous (versus non-homeomerous): Parts are similar to each other and to the whole.

The six types distinguished by Winston are listed in Table 1. The table also lists which of the just mentioned three aspects holds for each type, and an example from each, taken from [20].

Winston furthermore provides a discussion of logical properties for each type of part-whole relation. E.g., he observes that each type of relation is transitive, however if you mix types, transitivity generally does not hold. E.g., if you have two relations which are both of the component-integral object type, then transitivity holds, as in toe being part of the foot, foot being part of the leg, therefore toe is part of the leg. If you mix types, though, e.g. by mixing a component-integral object relation such as "Derek's nose is part of Derek" and a member-collection relation such as "Derek is part of the Department faculty," then transitivity would result in the nonsensical "Derek's nose is part of the Department faculty."

Rather than going through Winston's observations in detail, let us refer here to the axiomatization which we have drawn from it, and which we give in the next section.

3 The Winston-Part-Whole Pattern

We are now going to cast Winston's part-whole types into a part-whole ontology design pattern, and that will include the capturing, in OWL, of the logical relationships identified by Winston.

We will use the OWL property names

- component-integral object: po-component
- member-collection: po-member
- potion-mass: po-portion
- stuff-object: po-stuff
- feature-activity: po-feature
- place-area: po-place

and we will refer to these as *the specific part-whole relations*. We also use some other, related, relations identified and discussed by Winston. These are, in paticular, spatially-located-in as the spatial (topological) located-in relation and part-of as the generic part-whole relation of which the specific ones listed above are specializations (i.e., subProperties).

From [20] we can now draw the axioms which together constitute the pattern. They are listed in Figure 1.

Axioms (1) through (12) declare transitivity and asymmetry for each of the specific part-whole relations. According to Winston, however, we would also need to declare irreflexivity for each of the specific part-whole relations, which would render each of them a strict partial order. However this is not allowed in OWL 2 DL: according to [15, Section 11] a property cannot be both transitive (and, therefore, non-simple) and irreflexive.[4]

We believe that dropping the irreflexivity axioms should usually not cause any problems in terms of logical reasoning over the pattern, however as usual it is difficult to formally assess this. A formal declaration of irreflexivity may sometimes be helpful for ontology debugging or data curation, and of course some (correct) inferences will be missed through OWL 2 DL reasoning if the axiom is omitted. Note, though, that due to the open world assumption all inferences drawn from the OWL 2 ontology are still correct with respect to the complete theory (i.e., the one including irreflexivity).

Winston lists a number of additional axioms, however as discussed in [8] they are in fact tautologies, and while they may be informative for a linguistic

---

[4] Alternatively, we could also have dropped the transitivity axoims, but that seems less appealing. As discussed in [8], a third option would be to employ nominal schemas [12,14] and provide weaker forms of some of the axioms.

$$\begin{aligned}
\mathsf{po\text{-}component} \circ \mathsf{po\text{-}component} &\sqsubseteq \mathsf{po\text{-}component} && (1)\\
\mathsf{po\text{-}member} \circ \mathsf{po\text{-}member} &\sqsubseteq \mathsf{po\text{-}member} && (2)\\
\mathsf{po\text{-}portion} \circ \mathsf{po\text{-}portion} &\sqsubseteq \mathsf{po\text{-}portion} && (3)\\
\mathsf{po\text{-}stuff} \circ \mathsf{po\text{-}stuff} &\sqsubseteq \mathsf{po\text{-}stuff} && (4)\\
\mathsf{po\text{-}feature} \circ \mathsf{po\text{-}feature} &\sqsubseteq \mathsf{po\text{-}feature} && (5)\\
\mathsf{po\text{-}place} \circ \mathsf{po\text{-}place} &\sqsubseteq \mathsf{po\text{-}place} && (6)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}component}) && (7)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}member}) && (8)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}portion}) && (9)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}stuff}) && (10)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}feature}) && (11)\\
&\mathrm{AsymmetricObjectProperty}(\mathsf{po\text{-}place}) && (12)\\
\mathsf{po\text{-}component} &\sqsubseteq \mathsf{part\text{-}of} && (13)\\
\mathsf{po\text{-}member} &\sqsubseteq \mathsf{part\text{-}of} && (14)\\
\mathsf{po\text{-}portion} &\sqsubseteq \mathsf{part\text{-}of} && (15)\\
\mathsf{po\text{-}stuff} &\sqsubseteq \mathsf{part\text{-}of} && (16)\\
\mathsf{po\text{-}feature} &\sqsubseteq \mathsf{part\text{-}of} && (17)\\
\mathsf{po\text{-}place} &\sqsubseteq \mathsf{part\text{-}of} && (18)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (19)\\
&\mathrm{ReflexiveObjectProperty}(\mathsf{spatially\text{-}located\text{-}in}) && (20)\\
\mathsf{po\text{-}component} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (21)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}component} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (22)\\
\mathsf{po\text{-}member} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (23)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}member} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (24)\\
\mathsf{po\text{-}portion} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (25)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}portion} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (26)\\
\mathsf{po\text{-}stuff} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (27)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}stuff} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (28)\\
\mathsf{po\text{-}feature} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (29)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}feature} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (30)\\
\mathsf{po\text{-}place} \circ \mathsf{spatially\text{-}located\text{-}in} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (31)\\
\mathsf{spatially\text{-}located\text{-}in} \circ \mathsf{po\text{-}place} &\sqsubseteq \mathsf{spatially\text{-}located\text{-}in} && (32)
\end{aligned}$$

Fig. 1. Pattern axioms for the first pattern variant from Section 3.

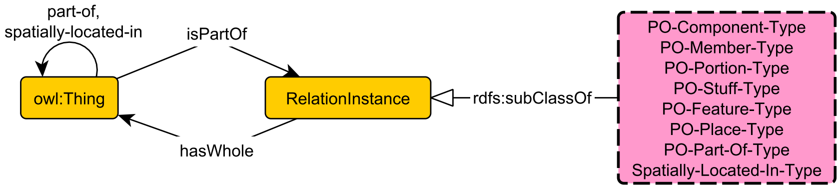

Fig. 2. Schema Diagram for the Contextualized Winston-Part-Whole Pattern. The dashed box on the right hand side lists seven different subclasses of PartWholeType. The subproperties of part-of from Section 3 are also used. Further explanations can be found in the text.

discussion, they do not really contribute to ontology modeling, and we do not want to include them in the pattern.

Please note that we do not provide a schema diagram for this pattern, as the pattern exists of related properties only.

4 A Pattern Extension Accounting for Provenance And Other Context Information

Some usages of the Winston-Part-Whole Pattern, such as the one from [8] on which this pattern is based, suggest that it would be helpful to store context information for the part-of relationship. We conceive that this would mostly be in the form of provenance information. For example, in the case of [8], part-of relationships of the various types defined by Winston were generated automatically using Hearst patterns over Web text corpora. In such a case, one may want to store confidence values, or even pointers to the exact algorithm used in each case.

In order to store this information, we now provide a contextualized version of the pattern described in Section 3; it is essentially obtained by "reifying" the properties. It is a known technique, and one could also refer to it as "lifting" or as "typecasting" of properties into classes following [13].

To explain, consider the schema diagram in Figure 2. A triple which according to the pattern in Section 3 would simply be stated as

```
:everglades   po:po-place    :florida .
```

would now be expressed using the following set of four triples—note that the original triple is still included. We use cpo as namespace, for "contextualized part-of."

```
:everglades   cpo:po-place     :florida ;
              cpo:isPartOf     :everglades-po-place-florida .
```

```
:everglades-po-place-florida   rdf:type       cpo:PO-Place-Type ;
                               cpo:hasWhole   :florida .
```

Additional context information, such as provenance information can then be attached to :everglades-po-place-florida, and we will further elaborate on this in Section 6.

We now show how to derive the axiomatization for the Contextualized Winston-Part-Whole Pattern. First of all, note that all axioms from Figure 1 are fully adopted (with adjusted namespace of course). In the following, let R denote any one of po-component, po-member, po-portion, po-stuff, po-feature, po-place, and C_R be the corresponding PO-Component-Type, . . . , PO-Place-Type from Figure 2.

$$\begin{aligned}
\mathsf{Po\text{-}Component\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (33)\\
\mathsf{Po\text{-}Member\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (34)\\
\mathsf{Po\text{-}Portion\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (35)\\
\mathsf{Po\text{-}Stuff\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (36)\\
\mathsf{Po\text{-}Feature\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (37)\\
\mathsf{Po\text{-}Place\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (38)\\
\mathsf{Po\text{-}Part\text{-}Of\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (39)\\
\mathsf{Spatially\text{-}Located\text{-}In\text{-}Type} &\sqsubseteq \mathsf{RelationInstance} && (40)
\end{aligned}$$

Then we would like to have all of the following axioms, which are here expressed using rules.

$$\mathsf{isPartOf}(x, y) \land C_R(y) \land \mathsf{hasWhole}(y, z) \to R(x, z)$$

This rule actually constitutes a generalized role chain which can be cast into OWL using the *rolification*[5] technique described in [12]. The resulting OWL axioms are as follows (please note the lowercase c_R, which is the result of typecasting the class C_R into a property).

$$\begin{aligned}
C_R &\equiv \exists c_R.\mathsf{Self} && (41)\\
\mathsf{isPartOf} \circ c_R \circ \mathsf{hasWhole} &\sqsubseteq R && (42)
\end{aligned}$$

The same axioms would be added for spatially-located-inin place of R.

Note that instead of axioms (42), we would actually have preferred to use

$$\mathsf{isPartOf} \circ c_R \circ \mathsf{hasWhole} \equiv R,$$

however this is not expressible in OWL. According to [13] use of the latter axiom would be proper typecasting between properties and classes, however this requires right-hand-side property chains, which if added to OWL DL would cause

---

[5] The name *rolification* comes from the fact that properties are called *roles* in description logics [7].

undecidability and are therefore not included in the standard. Please see [13] for a further discussion of this matter. For similar reasons, we are not able to lift most axioms from Figure 1 fully to the contextualized pattern, as they would also result in right-hand-side property chains. In fact, in addtition to the 14 axioms above we have six axioms

$$R \sqsubseteq \mathsf{part\text{-}of},$$

which correspond to axioms (13) through (18). The asymmetry declarations from Figure 1 cannot be fully lifted to the contextualized version: to the best of our abilities, they cannot be expressed in OWL, and the same holds for the reflexivity axiom. For axioms (1) to (6), (21) through (32), and (19), partial liftings could be given. However, they would be redundant, i.e., inferrable through OWL DL reasoning from the axioms already given. We thus refrain from adding them.

$$\top \sqsubseteq \forall \mathsf{isPartOf}.\mathsf{RelationInstance} \tag{43}$$

$$\forall \mathsf{hasWhole}.\mathsf{RelationInstance} \sqsubseteq \top \tag{44}$$

Finally, we give the range and domain for isPartOf and hasWhole, (43) and (44), respectively. In total, we have 32 axioms inherited from the non-contextualized pattern, plus 30 new ones, for a total of 62 axioms for the Contextualised Winston-Part-Whole Pattern.

5 Usage Scenario

We give a usage scenario for the presented patterns, from the domain of Materials Science. Materials Science is an interdisciplinary field which focuses on the discovery and design of new or enhanced materials. Of central importance to the field is the determination of materials properties using experiment or modeling and simulation. Examples of such properties include ultimate tensile strength and crack growth rate. More data than ever is being generated as the materials science and engineering domain seeks to enhance throughput through the automation of sequential experiments and greater use of modeling and simulation [16]. At the same time, there is no widely accepted ontology we are aware of to facilitate the digital exchange and integration of data in this fast-growing and very active discipline. To start filling this gap, we have begun to investigate core ontology design patterns needed for such an ontology, and this in fact prompted our development of the Winston Part-Whole Patterns based on earlier mentioned work.

The important role of part-whole relations in this context comes from the fact that engineered products are usually created by combining previously created engineered products—and that includes engineered materials. For example, fiberglass and epoxy (glue) are part of a composite material.

Product designers seek materials which possess specific properties (e.g. color, strength) to enable a function (e.g. be atheistically pleasing, resist deformation due to mechanical loads). These properties are established by combining specific materials in a particular way to achieve a certain microstructure. Once the processing is complete, the characteristic properties of the material are "locked-in." If the composition and structure of a material are described completely, a unique set of properties can be inferred. Additionally, since the processing can be associated with the composition and microstructure, it can also be associated with the unique set of properties. Thus, the recording of the parts or components of an engineered material is of importance.

Eventually, one would like to record the whole Part-Whole chain from a complex engineered product down to a very fine granularity. Examples for such relations could be the following.

- A radar system is part of a boat. – component-integral object
- An antennae radome is part of a radar system. – component-integral object
- Some composite material is part of an antennae radome. – stuff-object
- Epoxy is part of this composite material. – stuff-object
- Glass fiber is part of this composite material. – stuff-object
- Some composite material cure is part of some composite manufactoring. – feature-activity
- Some damaged area is part of some composite material surface. – place-area
- Some broken fiber is part of this damaged area. – component-integral object

It becomes apparent from these examples, that a naive approach, i.e., encoding all of these relationships using part-of only, is inferior to using a model based on Winston's work. E.g., in the former it would be incorrect, as duscussed, to declare part-of to be transitive, while our Winston Part-Whole Pattern allows for corresponding inferences where appropriate, e.g., from the above we could infer that An antennae radome is part of a boat (component-integral object) and that Glass fiber is part of an antennae radome (stuff-object).

6 A Provenance Pattern Derived From PROV-O

Provenance information is arguably among the most prominent types of context information for all kinds of data. We show in the following, how the Contextualized Winston-Part-Whole Pattern can be extended using a Provenance pattern which is derived from the core of PROV-O [5]. In a very similar way, other context information such as confidence values could be added.

The three core classes of PROV-O are Entity, Activity, and Agent. Briefly, an Entity is simply an item that has provenance. Entities are generated by Activities, which are the *execution* of some algorithm or method. The Activity or Entity may be performed by or attributed to some Agent which may be, for examples, a person or a script.

However, for use in the context of pattern-based modular ontology modeling [11], it is more convenient to have a dedicated pattern—rather than a full-blown ontology—at our disposal, although the pattern we provide is, essentially, the

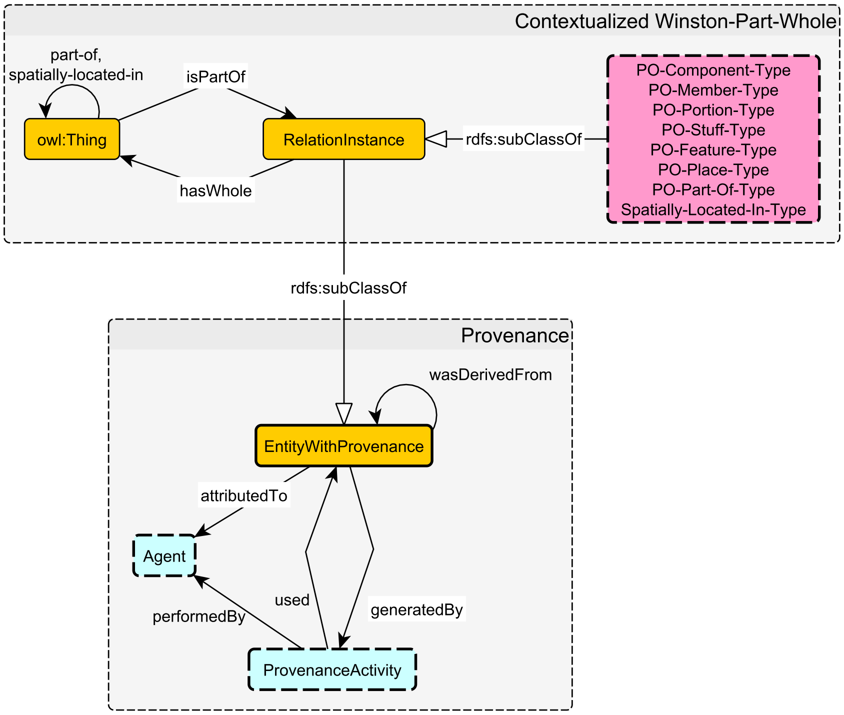

Fig. 3. Schema Diagram for the Contextualized Winston Part-Whole Pattern extended by a Provenance pattern following PROV-O.

core of PROV-O. We very simply align our extracted pattern to PROV-O via the following equivalences.

$$\begin{aligned} \mathsf{EntityWithProvenance} &\equiv \mathsf{Entity} \\ \mathsf{ProvenanceActivity} &\equiv \mathsf{Activity} \end{aligned}$$

Figure 3 provides a graphical overview of this subpattern and how it may extend the Winston-Part-Whole Pattern. The following axioms specify the behavior of this subpattern.

$$
\begin{aligned}
\mathsf{EntityWithProvenance} &\sqsubseteq \forall \mathsf{wasDerivedFrom}.\mathsf{EntityWithProvenance}\\
\forall \mathsf{attributedTo}.\mathsf{Agent} &\sqsubseteq \mathsf{EntityWithProvenance}\\
\mathsf{EntityWithProvenance} &\sqsubseteq \forall \mathsf{attributedTo}.\mathsf{Agent}\\
\forall \mathsf{generatedBy}.\mathsf{ProvenanceActivity} &\sqsubseteq \mathsf{EntityWithProvenance}\\
\mathsf{EntityWithProvenance} &\sqsubseteq \forall \mathsf{generatedBy}.\mathsf{ProvenanceActivity}\\
\forall \mathsf{used}.\mathsf{EntityWithProvenance} &\sqsubseteq \mathsf{ProvenanceActivity}\\
\mathsf{ProvenanceActivity} &\sqsubseteq \forall \mathsf{used}.\mathsf{EntityWithProvenance}\\
\forall \mathsf{performedBy}.\mathsf{Agent} &\sqsubseteq \mathsf{ProvenanceActivity}\\
\mathsf{ProvenanceActivity} &\sqsubseteq \forall \mathsf{performedBy}.\mathsf{Agent}
\end{aligned}
$$

We add some explanations of these axioms, they follow the standard templates of scoped domain and range restrictions.

1. The scoped range of wasDerivedFrom, scoped by EntityWithProvenance, is EntityWithProvenance.
2. The scoped domain of attributedTo, scoped by Agent, is EntityWithProvenance.
3. The scoped range of attributedTo, scoped by EntityWithProvenance, is Agent.
4. The scoped domain of generatedBy, scoped by ProvenanceActivity, is EntitWithProvenance.
5. The scoped range of generatedBy, scoped by EntityWithProvenance, is ProvenanceActivity.
6. The scoped domain of used, scoped by EntityWithProvenance, is ProvenanceActivity
7. The scoped range of used, scoped by ProvenanananceActivity, is EntityWithProvenance.
8. The scoped domain of performedBy, scoped by Agent, is ProvenanceActivity.
9. The scoped range of performedBy, scoped by ProvenanceActivity, is Agent.

Of course, pairs of different entities with provenance, or different agents, or different provenance activities, may in turn carry part-whole relationships, which could be expressed using Contextualized Winston-Part-Whole Pattern.

7 Pattern Release Information

We have released the Winston-Part-Whole Pattern,[6] the Contextualized Winston-Part-Whole Pattern, [7], and the Provenance Pattern[8] in OWL/XML syntax on the ontologydesignpatterns.org portal.

---

[6] `https://ontologydesignpatterns.org/wiki/Submissions:WinstonPartWhole`

[7] `https://ontologydesignpatterns.org/wiki/Submissions:ContextualizedWinstonPartWhole`

[8] `http://ontologydesignpatterns.org/wiki/Submissions:Provenance`

In addition, we have annotated the patterns with the appropriate annotations following the OPLa ontology which serves as ontology design pattern representation language [6]. The annotations were generated using the OPLa plugin for Protégé [17].

8 Conclusion

Part-whole relations are omnipresent and are fundamental to how we organize information and perceive the world. Thus, it is necessary to have a firm understanding of how to model these *partonomies* or *meronomies*. To do so, we have followed Winston's approach, as discussed in [20] and as a result, have developed two patterns: the Winston-Part-Whole Pattern and the Contextualized Winston-Part-Whole Pattern. Additionally, we provide a mechanism for augmenting the pattern with provenance.

Acknowledgements. Cogan Shimizu acknowledges funding from the Dayton Area Graduate Studies Institute.

References

1. Artale, A., Franconi, E., Guarino, N., Pazzi, L.: Part-whole relations in object-centered systems: An overview. Data & Knowledge Engineering 20(3), 347–383 (1996)
2. Casati, R., Varzi, A.: Parts and places: The structures of spatial representation. The MIT Press (1999)
3. Girju, R., Badulescu, A., Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. pp. 1–8. Association for Computational Linguistics (2003)
4. Girju, R., Badulescu, A., Moldovan, D.: Automatic discovery of part-whole relations. Computational Linguistics 32(1), 83–135 (2006)
5. Groth, P., Moreau, L. (eds.): PROV-Overview: An Overview of the PROV Family of Documents. W3C Working Group Note 30 April 2013 (2013)
6. Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A.A., Presutti, V.: Towards a simple but useful ontology design pattern representation language. In: Blomqvist, E., et al. (eds.) Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2043. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2043/paper-09.pdf
7. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
8. Jain, P., Hitzler, P., Verma, K., Yeh, P.Z., Sheth, A.P.: Moving beyond SameAs with PLATO: partonomy detection for linked data. In: Munson, E.V., Strohmaier, M. (eds.) 23rd ACM Conference on Hypertext and Social Media, HT '12, Milwaukee, WI, USA, June 25-28, 2012. pp. 33–42. ACM (2012)

9. Jain, P., Yeh, P.Z., Verma, K., Henson, C.A., Sheth, A.P.: SPARQL query rewriting using partonomy based transformation rules. In: Janowica, K., Raubal, M., Levashkin, S. (eds.) Proceedings of the 3rd International Conference on GeoSpatial Semantics. pp. 140–158. GeoS '09, Springer-Verlag, Berlin, Heidelberg (2009)
10. Keet, C.M., Kutz, O.: Orchestrating a network of mereo(topo)logical theories. In: Corcho, Ó., Janowicz, K., Rizzo, G., Tiddi, I., Garijo, D. (eds.) Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017. pp. 11:1–11:8. ACM (2017)
11. Krisnadhi, A., Hitzler, P.: Modeling with ontology design patterns: Chess games as a worked example. In: Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A., Presutti, V. (eds.) Ontology Engineering with Ontology Design Patterns – Foundations and Applications, Studies on the Semantic Web, vol. 25, pp. 3–21. IOS Press (2016)
12. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P.F. (eds.) Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures. Lecture Notes in Computer Science, vol. 6848, pp. 382–415. Springer, Heidelberg (2011)
13. Krisnadhi, A.A., Hitzler, P., Janowicz, K.: On the capabilities and limitations of OWL regarding typecasting and ontology design pattern views. In: Tamma, V.A.M., Dragoni, M., Gonçalves, R.S., Lawrynowicz, A. (eds.) Ontology Engineering – 12th International Experiences and Directions Workshop on OWL, OWLED 2015, co-located with ISWC 2015, Bethlehem, PA, USA, October 9-10, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9557, pp. 105–116. Springer (2015)
14. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Sadagopan, S., Ramamritham, K., Kumar, A., Ravindra, M., Bertino, E., Kumar, R. (eds.) Proceedings of the 20th International World Wide Web Conference, WWW2011, Hyderabad, India, March/April 2011. pp. 645–654. ACM, New York (2011)
15. Motik, B., Patel-Schneider, P., Parsia, B. (eds.): OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-syntax/
16. Nikolaev, P., Hooper, D., Perea-Lopez, N., Terrones, M., Maruyama, B.: Discovery of wall-selective carbon nanotube growth conditions via automated experimentation. ACS Nano 8(10), 10214–10222 (2014)
17. Shimizu, C., Hirt, Q., Hitzler, P.: A Protégé plugin for annotating OWL ontologies with OPLa. The Semantic Web: ESWC 2018 Satellite Events (2018), to appear.
18. Tryfona, N., Egenhofer, M.J.: Consistency among parts and aggregates: A computational model. Transactions in GIS 1(3), 189–206 (1996)
19. Varzi, A.: Parts, wholes, and part-whole relations: The prospects of mereotopology. Data & Knowledge Engineering 20(3), 259–286 (1996)
20. Winston, M.E., Chaffin, R., Herrmann, D.: A taxonomy of part-whole relations. Cognitive Science 11(4), 417–444 (1987)