A critique of rigid temporalized relations

Christopher J. Mungall [1*]

[1]Genomics Division, Lawrence Berkeley National Laboratory, MS84R017, 1 Cyclotron Road, Berkeley, CA 94720 USA

ABSTRACT

In this report I evaluate the proposed new temporalized relations strategy in which many existing relations would be replaced by two ore more relations, an *at-all-times* and an *at-some-times* form.

My findings are that the at-all-times relations have an underlying problem that renders them formally incorrect for use in ontologies that represent structures that change over time. The at-some-times relations would lose crucial inferences. These logical problems are compounded by the fact that the relations are difficult for users to understand, and will most likely lead to confusion and errors, especially in the absence of detailed documentation.

I conclude that these relations should not be adopted by ontology editors. Migrating to these relations would be an expensive, error-prone process that would alienate the users of ontologies, and the end result would be ontologies that are either formally incorrect or too weak to perform required inferences.

1 INTRODUCTION

The OBO relations ontology (**OBO-REL**) defined a set of core relations for use in biological ontologies, including is_a, part_of and derives_from[4]. The original **OBO-REL** paper has been cited 709 times[1], and has been a crucial reference in the correct usage of relations in biological ontologies.

One notable aspect of **OBO-REL** was the precise specifications of how relationships change or remain the same through the passage of time. For example, a user of **OBO-REL** could say that *every* cell nucleus is part of *some* cell at *any* given moment of time (for which that cell nucleus exists).

$\forall x \forall t : \mathsf{instance\_of}(x, \mathsf{cell\ nucleus}, t) \rightarrow \exists y \mathsf{instance\_of}(y, \mathsf{cell}, t), \mathsf{part\_of}(x, y, t)$

This relationship is known as *permanent generic parthood*.

The other notable aspect of **OBO-REL** was the distinction between type level and instance level relations. Each type-level relation connects a pair of classes and typically is defined according to an ALL-SOME-ALLTIMES pattern - for example, every cell nucleus is part of some cell at all times. The use of type level relations causes some confusion when using OWL, which does not support the same mechanism. Instead the relationship between a nucleus and a cell in OWL is explicitly quantified, but without a time argument, as all relations in OWL are binary: For example, *every cell nucleus is part of some cell*. As of 2010, the official semantics of OBO format have been as a subset of OWL, so what applies to OWL necessarily applies to OBO.

*to whom correspondence should be addressed

[1] Google scholar

This impedance mismatch between the OWL interpretation and the **OBO-REL** account has been a problem is providing a consistent formal account of relations, although it is not clear that this has caused a problem for many ontology developers or users. The standard approach has been to use a set of binary relations as specified in Table 1.

As part of the release process for the OWL translation of version 2 of the Basic Formal Ontology (BFO)[2], a number of people explored different strategies for unifying OWL binary properties with the ternary relations in the BFO2 reference specification[3]. One such strategy is the *temporalized relations* strategy, in which each reference relation relating continuants has two or more OWL cognates, rel-at-some-times and rel-at-all-times.

In this review I do not attempt to compare or even describe the different modeling possibilities, instead I focus purely on the "temporalized relations" strategy. I first provide an outline of temporalized relations, drawing on the existing release notes, attempting to fill some gaps. I then present the major problems posed by these relations: (1) the relations fail to capture the biological reality, forcing ontology editors to make a tradeoff between two unsatisfactory choices (2) the relations are confusing for users and even experienced ontology editors. These problems are related, in that ontology editors may not understand the tradeoff they are being asked to make.

Finally I end with some specific recommendations regarding the temporalized relation strategy in general and BFO2 specifically.

2 TEMPORALIZED RELATIONS

Here I distinguish between reference relations (RRs) and their manifestation in OWL as binary temporalized relations (TRs), using the temporalized relation strategy (TRS). All relations are instance level. As a typographic convention I use dashes to separated the words in a temporalized relation, and underscores in a reference relation.

The BFO2 Graz version release notes[3] specify a general template for relating RRs to TRs:

```
x rel-at-some-time y ->
 exists(t) exists_at(x,t) ->
    exists_at(y,t) and rel(x,y,t)
x rel-at-all-times y ->
  forall(t)  exists_at(x,t) ->
      exists_at(y,t) and rel(x,y,t)
```

Here we focus on part_of as an exemplar relation, whilst recognizing that similar patterns may apply to other, but not all relations.

For part_of connecting two continuants there are three TRs:

1. part-of-at-some-times
2. part-of-at-all-times
3. part-of- at-all-times- for-which-whole-exists

Note that in BFO2 the actual label is "part of continuant at some/all times", we shorten this for brevity.

On the surface, the at-all-times form appears to be the same as the **OBO-REL** interpretation. For example, the OWL axiom:

cell nucleus SubClassOf part-of-at-all-times some cell

may seem to be the same as the statement "every cell nucleus is part of some cell at all times". We might even be able to automatically translate an ontology written using the **OBO-REL** interpretation into TRs. *However, these are NOT the same, and an understanding of why is crucial to correct usage of these relations and an understanding of the consequences of using them.*

SHOW FOL HERE

The fact that these are different is of utmost importance to how ontologies are created, and affects the *characteristics* of these relations in some ways that might seem surprising.

2.1 Object property characteristics

In OWL, relations (object properties) can have certain characteristics such as *transitive*, *symmetrical*, and they may have additional logical axioms such as *inverse properties*. These are extremely useful for many purposes - transitivity has been at the core of bioinformatics applications of ontologies from the initial version of the Gene Ontology[1], if not before. Inverse properties are useful for instance level reasoning, and for finding errors in complex ontologies.

When translating a RR to a TR, it may not be immediately clear what properties of the RR should be carried over to the TR. Ideally we would be able to prove that the RR and TR are consistent, although the proof may be obvious to an expert logician. Here I use part_of as an illustrative example, although each relation may require individual examination for its properties.

Table 2 shows some of the property characteristics of the various forms of the continuant parthood relations.

2.1.1 Transitivity In the case of the RR part_of (which is transitive), the stronger at-all-times TR retains the transitivity characteristic, whereas the weaker at-some-times TR does not have this. This means the weaker version is often safer to use in an ontology, but will lead to fewer inferences.

2.1.2 Symmetricality For other relations and other characteristics, the translation may not be obvious. For example, the adjacency relation is commonly assumed to be symmetric[2]. Should this symmetricality characteristic be carried over to the temporalized form?

Currently BFO2 does not have an adjacency relation, or any other reference relation that is symmetric, so the following is based on my own understanding. I would assume that the symmetricality should be declared for the at-some-times form and not the at-all-times form. In contrast to the partOf and transitivity case, here it is the weaker form of relation that inherits the characteristic.

See table 3 for a summary.

[2] we are only considering instance level relations

Relation	Trans	Symm	Inverse Of
part_of	Yes	No	has_part
has_part	Yes	No	part_of
adjacent_to	No	Yes	

Table 1. Relation characteristics, atemporal. These are the characteristics of the instance level relations in the current RO

2.1.3 Subproperties We assume the following hierarchy:

at-some-times < at-all-times-for-which-subject-exists < at-all-times

Note how this interacts with other properties. If $x$adjacent_to at-all-times$y$, then we can infer that $y$adjacent_toat-some-times$x$.

2.1.4 Inverse Properties Ontologies frequently declare inverse relations. For example, the RR part_of is the inverse of has_part. This is fairly standard practice, and the inverse relations are extremely useful for reasoning. In OWL, it's not strictly necesary to declare an inverse, as it is possible to use an InversePropertyExpression. Here we take the view that inverses that have typically been declared in previous ontologies are useful, and should also be declared as RRs and have corresponding TRs.

The inverse of the TRs of part_of may not be completely intuitive. Naively we might guess that part-of-at-all-times would be the inverse of has-part-at-all-times, but this not the case.

In fact, the declaring the inverse of part-of-at-all-times requires declaring a third TR form, has-part- at-all-times-for-which-part-exists. The at-all-times-for-which-subject-exists form can be generated for some but not all RRs. Whilst it complicates the ontology of TRs to introduce this extra form, the alternative of not having inverses (or of being forced to write complicated inverse expressions) may be too prohibitive for many users.

2.1.5 Other characteristics We do not consider other characteristics such as antisymmetry and domain/range restrictions here.

2.1.6 Generating a TR from an RR Currently there is no "recipe" for generating a set of TRs from an RR. Different patterns may apply to different RRs.

For example, when creating the TRs for inheresIn and its inverse bearerOf, it is correct to declare bearerOfAtAllTimes InverseOf inheresInAtAllTimes.

3 EVALUATION

3.1 Temporalized Relations do not reflect the intentions of ontology editors

TRs present many challenging problems. For example, when converting an anatomy ontology that has been modeled traditionally using the relations in table 1, the ontology editor must make a choice on a per-relationship basis as to which of the relations in 2 should be used.

Relation	Trans	Inverse Of
part_of at-some-times	No	has_part at-some-times
part_of at-all-times	Yes	has-part- at-all-times-for-which-part-exists
part-of- at-all-times-for-which-whole-exists	Yes	has_part at-all-times
has_part at-some-times	No	part_of at-some-times
has_part at-all-times	Yes	part-of- at-all-times-for-which-whole-exists
has-part- at-all-times-for-which-part-exists	Yes	part_of at-all-times

Table 2. Relation characteristics for core continuant parthood relations. These characteristics are declared in the current BFO2 OWL Graz version

Relation	Symm	Inverse Of
adjacent_to at-some-times	Yes	adjacent_to at-some-times
adjacent_to at-all-times	No	

Table 3. Relation characteristics for a typical symmetric relation. This relation is not part of the BFO2 OWL Graz version

Relation	Axiom
part_of	$\forall x \forall t$: instance_of(x, cell nucleus, t) $\rightarrow$ $\exists y$: instance_of(y, cell, t), part_of(x, y, t)
part-of-at-all-times	$\forall x$: instance_of(x, cell nucleus) $\rightarrow$ $\exists y$instance_of(y, cell), part_of(x, y, t)
part-of-at-some-times	$\forall x \exists t$: instance_of(x, cell nucleus, t) $\rightarrow$ $\exists y$instance_of(y, cell, t), part_of(x, y, t)

Table 4. Summary of relations.

This is an onerous task, but this could be justified if the results were better ontologies. However, in many cases *none of the choices are appropriate*.

This is because for many ontologies, the most appropriate choice of parthood relationship is the *permanent-generic* form, as specified in the original OBO relations paper. The standard example here is the relationship between a cell nucleus and a cell. At any moment in time, a given cell nucleus is by definition part of some cell[4]. However, *this need not be the same call throughout the lifetime of the nucleus*.

In contrast, if the part-of-at-all-times TR is used then the interpretation is that the cell nucleus is always part of the same cell. This interpretation can be proved to be formally wrong in cells that undergo cell division**??**. This is because the cell nucleus is always part of a cell, *but not the same cell*.

The ontology editor can not then choose to use part-of-at-all-times without making a false statement. They may then decide to use part-of-at-some-times. Such a usage would be formally valid, but incomplete from the point of view of useful reasoning. This is because the weaker at-some-times form lacks the transitivity characteristic.

A third possibility is to use the part-of- at-all-times-for-which-whole-exists form, but this would also be false (note the above proof needs to be extended to cover this case).

The ontology editor requires the permanent generic form in order to be both accurate and to get the required inferences. However, this form is specifically excluded in the TR strategy.

The cell nucleus example is the standard one, because it is central to all of eukaryotic biology. It is by no means the only such example. The problem arises whenever we have material passed around from one carrier to another. Enumerating a list of examples is difficult because the instance level identity conditions may not be clear.

the problem is not limited to parthood relations. Use of TRs requires that all continuant relations are temporalized. This includes relations used to classify structures by phenotype.

3.2 Rigidity requirement is too onerous

The BFO2 Graz release notes state:

> Thus we only instantiate “rigid” classes, as the interpretation we take is a rdf:type C =¿ forall(t) a exists at t -¿ a instance of C at t. Temporally restricted instantiation is not supported in this version of BFO in OWL. We are working on it for the future.

To many users this may seem like an obscure point, but it is actually quite a sever restriction. A class is rigid if it is instantiated “for life”. If an individual transforms from being an instance of one class of thing to another, then those classes are not rigid.

An example of a rigid class may be “Homo sapiens”. If an individual instantiates this class at some time t, then they instantiate it all times for which they exist (barring some unusual inter-species transformation).

This clause means material entity classes such as the following may not be supported in this version of BFO:

1. ’human with Parkinson’s disease’
2. ’female organism’
3. ’infected lung’
4. ’professor’ (but *professor role* is allowed)
5. ’human patient’ (but *patient role* is allowed)
6. ’oocyte’
7. ’fractured bone’
8. ’happy human’
9. ’fetal heart’
10. ’neural crest cell’
11. ’open heart valve’
12. ’gravid uterus’
13. ’phosphorylated EGFR protein’
14. ’cytoplasmic NFkB’
15. any leaf node from PATO

[4] we would consider extruded nuclei to be transformations of cell nuclei, but instantiating a different class

In some cases the constraint may not be so onerous. It can be argued that a well structured ontology would never include a class “professor”, and that this should always be modeled using a rigid class (human) plus a role (professor role). However, these decisions should be made on a case by case basis by each ontology that than imposed from the upper ontology.

In other cases the distinction between rigid and non-rigid may not be clear. One can argue that when an EGFR protein changes state from being unphosphorylated to a phosphorylated state it is no longer the same instance - the protein literally ceases to exist and is replaced by a distinct individual an instant later, sharing all the same properties except that it is phosphorylated. In fact one could take this position for all of the above cases, in which case the TR strategy becomes similar to the temporally qualified continuant (TQC) strategy. I do not explore this further, as I assume this is contrary to the expectations of the TR proponents.

When considering BFO2 in particular, there is an interesting disjunction between the reference document, which explicitly states that *determinates* (for example, qualities such as “square”, “charged”, “cylindrical”) are non-rigid. The Graz release states that instantiation of these classes is not supported. These two seemingly contradictory statements are not explicitly linked anywhere. The modeling implications of this disconnect are not clear, and require further documentation. It cannot be ruled out that this restriction will involve further complexity.

It may be the case that future versions of the TR strategy will allow for non-rigid classes. It is not clear how this will be achieved without additional complexity. The TR strategy must be evaluated on what exists presently, and at this time the strategy comes with constraints that ontology developers should be fully aware of.

3.3 Temporalized relations add complexity

The most striking feature of an ontology that uses the TR strategy is the complexity. Whereas using traditional modeling, we may have has a single parthood relation, we now have three.

In theory some of this complexity could be tamed by additional tooling (although it is not clear who has the resources to implement this). However, even if this can be hidden from the user, the ontology developer is forced to wrestle with the complexity.

This complexity first manifests when an ontology developer chooses to migrate from a traditionally modeled ontology using relations from table 1, assuming an **OBO-REL** interpretation.

For each axiom that uses a relation that has multiple variants in TR form, the developer must make a choice of which one to use....

BAD SMELL

4 DISCUSSION

4.1 Recommendations

4.1.1 Do not use My primary recommendation is that Temporalized Relations should not be used in their current form. Ontologies should not migrate to them.

4.1.2 Documentation The TR strategy needs much more documentation if ontology developers are to use TRs. Even if TRs are abandoned in their current form (as I recommend), documentation would be useful to be able to help achieve consensus on this matter.

4.1.3 Alternate strategies Given the inherent limitations and complexity of TRs, adequate consideration should be given to alternate strategies such as Temporally Qualified Continuants (TQCs). The “null” strategy of continuing to use simple OWL object proeprties as if they has a **OBO-REL** interpretation should be the default strategy until an adequate replacement is found.

4.1.4 Use cases If adopted, TRs will require tremendous effort in omntology migration, documentation, tooling. There is little to motivate ontology developers to do this as the existing strategy works for them. The main motivating factor seems to be a desire for formal correctness, at the expense of usability and biological correctness.

5 CONCLUSIONS

Temporalized Relations would be a fundemental change to the way relationships are modeled in ontologies. They would introduce additional complexity.

Some of these costs could be justified if Temporalized Relations were on a path towards making ontologies more biologically accurate. However, due to the built in lack of support for permanent generic parthood and non-rigid classes, migrating to Temporalized Relations would lead to ontologies becoming *less* accurate. My recommendation is unambiguous in its rejection of the use of Temporalized Relations in biological ontologies.

ACKNOWLEDGMENTS

REFERENCES

[1]M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

[2]Pierre Grenon, Barry Smith, and Louis Goldberg. Biodynamic ontology: applying BFO in the biomedical domain. *Stud Health Technol Inform*, 102:20–38, 2004.

[3]Stefan Schulz, Alan Ruttenberg, Melanie Courtot, and Janna Hastings. Bfo2 owl graz version release notes.

[4]B. Smith, W. Ceusters, J. Kohler, A. Kumar, J. Lomax, C.J. Mungall, F. Neuhaus, A. Rector, and C. Rosse. Relations in Biomedical Ontologies. *Genome Biology*, 6(5), 2005.

APPENDIX

Axiom	
part_of	$\forall x \forall t$: instance_of(x, cell nucleus, t) $\rightarrow$ $\exists y$: instance_of(y, cell, t), part_of(x, y, t)
part-of-at-some-times	$\forall x \exists t$: instance_of(x, cell nucleus, t) $\rightarrow$ $\exists y$instance_of(y, cell, t), part_of(x, y, t)

Table 5. Summary of relations.