A critique of temporalized relations

Christopher J. Mungall [1*]

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

ABSTRACT

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

My findings are that the *at-all-times* relations have an underlying logical problem that renders them formally incorrect for use in many ontologies. The *at-some-times* relations are safer, but would lose crucial transitive inferences. These logical problems are compounded by the fact that the relations are difficult for users and ontology developers 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 developers as a replacement for existing relations. Migrating to these relations would be an expensive, error-prone process that would alienate the user base of an ontology, and the end result would be ontologies that are either formally incorrect or too weak to perform required inferences.

1 INTRODUCTION

1.1 The OBO Relations ontology

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[7]. 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, the intent was that an ontology could state that *every* cell nucleus is part of *some* cell at *any* given moment of time (for which that cell nucleus exists). This has a precise interpretation in first-order logic:

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

A relationship of this form is known as *permanent generic parthood*. It allows for nuclei to be passed from one cell to another, so long as the nucleus remains part of some cell.

The other notable aspect of **OBO-REL** was the distinction between type (class) 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. This causes some confusion when using OWL, which does not support these kinds of class-level relationships. 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:

```
'cell nucleus' SubClassOf 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.

This impedance mismatch between the OWL interpretation (which can be considered “atemporal”) and the **OBO-REL** account has been problematic in providing a consistent formal account of relations that is consistent with OWL semantics, 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, and to assume an **OBO-REL** type interpretation for time.

1.2 BFO2 and temporalized relations

One of the stated goals of the group developing the OWL translation of version 2 of the Basic Formal Ontology (BFO)[2] was to have relations in OWL handled so as to have a clear First-Order Logic (FOL) reading according to the BFO reference[8]. The BFO group explored different strategies for unifying OWL binary properties with the ternary relations[3]. One such strategy is the *temporalized relations* (TR) strategy, in which each reference relation relating continuants has two or more OWL cognates, rel-at-some-times and rel-at-all-times. This strategy has been adopted as the official one for the OWL translations of BFO2[6].

1.3 Outline

In this review I do not attempt to compare or even describe the different modeling possibilities (see [3] for details here); due to pressure to make temporalized relation the standard for OBO library ontologies I focus purely on this approach.

I first provide an outline of temporalized relations, drawing on the existing release notes and documentation, attempting to fill some gaps and provide additional explanations of some of the complexities. I then present the major problems posed by these relations:

1. Temporalized relations are logically flawed. Their meaning is opaque to OWL, decreasing our ability to use reasoners to detect errors and build ontologies. The approach is inherently constrained, and fail to capture dynamic situations, forcing ontology editors to make a choice between two unsatisfactory models.

*to whom correspondence should be addressed

[1] Google scholar

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 (http://obo-relations.googlecode.com)

2. Temporalized relations would lead to an unacceptable increase in complexity in most ontologies.

These problems are inter-related; the complexity of the approach with its proliferation of relations masks underlying logical problems. Ontology editors who do not understand the formalism are more likely to make errors, both in migrating to temporalized relations, and in maintaining ontologies that use them.

Finally I end with some concrete recommendations regarding the temporalized relation strategy.

My intentions are to make this review accessible to a wide audience, and to keep logical formulas to a minimum. This is difficult because to understand the motivations of the TR approach and the underlying problems it is necessary to understand some of the logical underpinnings. I have annexed some of the finer grained details into an appendix, and attempted to stick with examples a biologist would understand.

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 separate the words in a temporalized relation, and underscores in a reference relation.

2.1 Translation template

The BFO2 Graz version release notes[6] 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 I 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 in fact *three* TRs rather than two (for reasons that will be explained shortly):

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 labels are "part of continuant at some time", "part of continuant at all times" and "part of continuant at all times for which whole exists", I shorten this for brevity – here we are only concerned with relations that involve a continuant.

2.2 TRs have a different meaning from OBOREL relations

On the surface, the at-all-times form appears to be the same as the **OBO-REL** interpretation. For example, an ontology that uses TRs may include the following OWL axiom:

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

This 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 this is so is crucial if these relations are to be used correctly, and to understand the long-term consequences of using them.*

It is not the case that TRs are the same as what has come before, but with longer labels. *The semantics are fundamentally different*. This has major effects on both the OWL axioms that are crucial for reasoning over ontologies, and on the FOL reading of what the axioms in the ontology means. See **table 4** in the appendix to compare the FOL for TRs vs **OBO-REL**. For readers not versed in FOL, I will next provide an outline of how the TR semantics affect the OWL *characteristics* of these relations in some ways that might seem surprising. I will return to this in the evaluation to show how this affects real-world ontology modeling.

2.3 Object property characteristics

In OWL, relations (object properties) can have certain characteristics such as being *transitive*, *symmetrical*, and they may be related to other relations via logical axioms such as *inverse properties* and property chains. 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. If a reference relation R has an inverse R', it does not follow that the temporalized versions will be inverses. Here I use part_of as an illustrative example, although each relation may require individual examination for its properties.

Figure 1 shows the characteristics of continuant parthood relations, both atemporal (A) and temporalized (B).

2.3.1 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 (but not always) safer to use in an ontology, but will in general lead to fewer inferences.

2.3.2 Symmetricality For other relations and other characteristics, the translation may not be obvious. For example, the adjacency

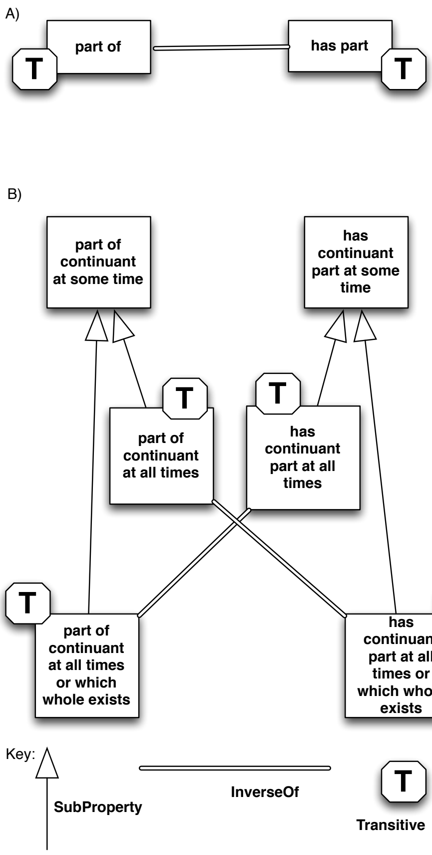

Fig. 1. Parthood relations, both atemporal (A) and temporalized (B). The atemporal form can be found in the current RO. The more complex temporalized form can be seen in the BFO2 Graz release

.

relation is symmetric in its atemporal form[2]. The symmetricality property should only be carried over in the weaker at-some-times form[3]. Table 2 shows the characteristics.

2.3.3 Sub Properties The general template for a TR hierarchy is one in which at-some-times is the most general, with sibling subproperties at-all-times and at-all-times-for-which-partner-exists. This is illustrated for parthood in figure 1.

[need more verbiage on why the hierarchy is like this. proofs?]

Note how this interacts with other properties. Given:

$$x \text{ adjacent\_to at-all-times } y$$

We infer:

$$y \text{ adjacent\_to at-some-times } x$$

2.3.4 Inverse Properties Ontologies frequently declare inverse relations. For example, the RR part_of is the inverse of has_part.

[2] I am only considering instance level relations

[3] BFO2 does not have an adjacency relation, this reflects my assumption about what the axioms should be

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

Table 2. Relation characteristics for adjacency, a symmetric relation. Note that in contrast to part_of and transitivity, it is the weaker form of the relation that inherits the characteristic

This is fairly standard practice, and the inverse relations are extremely useful for reasoning.

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, more specific, TR form, has-part- at-all-times-for-which-part-exists. This is illustrated in 1(B).

The at-all-times-for-which-partner-exists form is be generated for some but not all RRs.

2.3.5 Property Chains Property (relation) chains are an OWL construct that allows a reasoner to infer that a chain of relationships of a specified type infers another relationship type. For example, the rule that if x has part y and y is part of z then x overlaps z is written as:

```
has_part o part_of -> overlaps
```

These chains have proven extremely useful for a number of bio-ontologies. Translating these into temporalized form is complicated and leads to multiplication of axioms. I omit discussion of this for space reasons here, see the BFO tracker[5] for further details.

2.3.6 Other characteristics I do not consider other characteristics such as anti-symmetry and domain/range restrictions here[perhaps future versions].

2.4 TR formalism summary

I have presented a brief overview of how reference relations are translated to the more complicated temporalized form, using part_of as the main example. It should be noted that many - perhaps the majority - of relations will have to be temporalized in this way if a TR approach is to be adopted.

3 EVALUATION

My evaluation is split into two parts: (1) logical problems with the TR approach, leading to incorrect ontologies and limiting the utility of reasoners (2) the inherent complexity of the TR relations.

These two problems are intertwined. The TR approach with its inflation of relations and axioms is so complex that many of the problems highlighted here were not immediately obvious.

[5] http://code.google.com/p/bfo/issues/detail?id=157

3.1 Temporalized relations are logically flawed

TRs present many challenging problems. For example, when converting an anatomy ontology that has been modeled traditionally using the relations in figure 1(A), the ontology editor must make a choice on a case by case basis as to which of the relations in 1(B) should be used.

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*, resulting in an ontology that is *worse* in terms of formal correctness and reasoning power.

3.1.1 Temporalized relations lead to loss of error-checking capabilities The complexity and relation inflation of the TR approach is a potential cause of curator error in ontologies - editors will require additional training and even then subtle errors cause by using the wrong temporalized form will be common.

This problem could be mitigated if we could use reasoners to check for common errors. In fact, the use of TRs *lessens* the power of reasoners to detect errors.

As an example, consider the powerful part-disjointness pattern[5] which has been very useful for error checking in GO and anatomical ontologies. This allows us to state that nothing is part of both a mitochondrion and a nucleus:

```
(part_of some mitochondrion)
  DisjointWith (part_of some nucleus)
```

When we try and extend this to TRs we are faced with the choice of relation to use. The default choice would be the stronger at-all-times form:

```
(part-of-at-all-times some mitochondrion)
  DisjointWith
 (part-of-at-all-times some nucleus)
```

However, on the FOL level, this is far too *weak*. This weakness carries over to the OWL as well, and it is easy to demonstrate errors that reasoners cannot catch (for example, chromosomes being part of both mitochondria and nuclei). See the BFO tracker for full details[6].

In order to approximate the level of error checking we have without TRs, it's necessary to add 4 axioms, for each of at-all-times possibilities.

```
(part-of-at-all-times some mt)
  DisjointWith
(part-of-at-all-times some nucleus)
(part-of-at-all-times some mt)
  DisjointWith
(part-of-at-all-times-that-whole-exists
  some nucleus)
(part-of-at-all-times that whole exists
  some mt)
  DisjointWith
(part-of-at-all-times-that-whole-exists
  some nucleus)
(part-of-at-all-times some nucleus)
  DisjointWith
(part-of-at-all-times-that-whole exists
  some mt)
```

[6] http://code.google.com/p/bfo/issues/detail?id=156

Even with this additional complexity, the FOL still falls short of what is required.

It is simply not practical for ontology editors to have to write all such possible forms.

3.1.2 Temporalized Relations can not represent certain types of relationships One reason why TRs are formally incorrect for many ontologies is because most appropriate choice of parthood relationship is the *permanent-generic* form, as exemplified 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. However, *this need not be the same cell 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[4]. This is because in reality the cell nucleus is always part of a cell, *but not the same cell*. With part-of-at-all-times there is no “migration” allowed - a nucleus is always part of the same cell.

This is likely to cause problems when the ontology is used to model instance data[7]. Even if we do not use instance data, this exposes an inherent contradiction in the TR approach:

1. The primary justification for incurring the complexity of TRs is that they are necessary to achieve a clear FOL reading of the OWL model.
2. By adopting the TR approach we are forced to make statements in OWL that are *false* in the FOL reading.

An ontology editor might choose to circumvent this problem by choosing the weaker part-of-at-some-times relation, but this would lose transitivity, which is widely viewed to be an unacceptable compromise.

To summarize this issue, in the nucleus-cell case (which is representative of many others), 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 existing TR strategy, forcing the ontology editor to make statements that are formally incorrect.

3.1.3 Temporalized relations are ill-equipped to model change The constraints imposed by TRs are wider than the lack of permanent-generic relationships, as exemplified in the nucleus-cell example. The problems extend to a much wider range of ontology classes, including classes whose members change with respect to some property (such as location) over time. TRs are fundamentally ill-equipped to deal with these.

As an example, consider the class **premigratory neural crest cell**. This is a cell that is part of the **neural crest**[8], prior to migration. In the Cell Ontology, this is modeled as:

[7] Difficult to evaluate this at this time, as the current BFO2 TR proposal does not suggest how instances are to be modeled

[8] here we use the term 'neural crest' to denote the region of the neuroepithelium

```
'premigratory neural crest cell'
  SubClassOf part_of some 'neural crest'
```

Ontology editors interpret this correctly according to the **OBO-REL** interpretation, in that the parthood relationship is true during the period in which the cell instantiates the premigratory type.

In choosing the correct TR for this relationship we have the choice of whether to go with the weaker at-some-times or the stronger at-all-times. Again, the weaker relation loses transitivity; the stronger relation is formally wrong, because the cells can migrate. Making a false statement here will inevitably lead to major problems and contradictions later on. If the ontology contains false statements it will lead to confusion on the part of the users and developers of ontologies. It certainly complicates the task of those of us who train ontology developers.

This particular problem with TRs arises as a direct consequence of the next problem, the inability of the approach to model *non-rigid classes*.

3.1.4 Temporalized relations are incompatible with vital non-rigid classes The TR approach disallows instantiation of classes that are *non-rigid*, which is a severe constraint when developing ontologies that deal with things that change over time.

The BFO2 Graz release notes[6] 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 this is actually a severe 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 class, 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). Upper level categories like 'process' are also rigid.

The following classes are non-rigid, and therefore do not have full support in this version of BFO / the TR approach:

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

See the appendix for a full discussion of each of these cases.

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 for each ontology that than imposed from above.

In many of other other cases, forcing the ontology developer to exclude some of the classes above is too onerous. For example, many anatomical ontologies make use of phase or stage as a differentia. This TR constraint is therefore not acceptable for anatomy ontology developers.

Whilst technically the BFO2 release notes only state that instantiation of non-rigid classes is not supported, I showed previously with the neural crest cell example that non-rigid classes are not compatible with the TR approach. It is impossible to adequately represent the parthood relationships for classes such as 'premigratory neural crest cell' using the existing TR strategy.

It is in fact impossible to specify the defining relationship for non-rigid classes using TRs, as these will be by definition wrong on the FOL level.

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, rather than what it might become in the future, and at this time the strategy comes with major constraints that ontology developers should be fully aware of.

In terms of BFO2 there is a particular disconnect when it comes to qualities - this is covered in the appendix.

To summarize this criticism: non-rigid classes are commonly used in many biological ontologies. Instantiation of these classes are not supported in this version of the TR approach, and attempts to specify defining relationships for these classes are formally wrong on the FOL level.

3.2 Temporalized relations add considerable complexity

In addition to logical problems with TRs, the complexity they incur causes significant problems - including obscuring the logical problems.

3.2.1 Temporalized relations proliferate relations in a complex network 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, arranged in a counter-intuitive network of axioms; contrast (A) and (B) in figure 1. Note that in fact this figure is a simplification, as it does not show other sub-properties, such as the member part relations. In production ontologies, most relations involving continuants would have to be split in this way. Many relations are inter-related via property chains and other axioms, it is as yet unclear how much complexity an ontology rich in relations would suffer.

3.2.2 Multiple levels of quantification Another source of complexity is that ontology editors now have to handle an extra layer of quantification. Consider some of the possible ways to model the relationship between a population of organisms and an organism:

```
1. population SubClassOf
  has-part-at-some-time some organism
2. population SubClassOf
  has-part-at-all-times some organism
3. population SubClassOf
  has-part-at-all-times-that-part-exists
    some organism
4. population SubClassOf
  has-part-at-some-time only organism
```

In each case there is in fact three levels of quantification. The first level is the OWL subclass axiom, which states that the condition holds for ALL instances of a population. Also within the scope of OWL is the final SOME or ONLY quantifier. Finally, embedded within the relation is an additional layer is the temporal quantification. Note that this final layer is opaque to OWL reasoners (and thus harder to use standard tools to check).

This is in contrast to the simpler, well-documented kind of atemporal quantification ontology developers perform at the moment.

(Note that *none* of the axioms above are correct, given that populations gain and lose members over time - permanent generic parthood is required).

I have a great deal of experience in training and assisting ontology developers in the use of tools such as reasoners and in making the transition to OWL. In my estimation, the level of complexity TRs exert is simply too high.

3.2.3 Migration is complex and will be error-prone 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 complexity can be hidden from the user, the ontology developer is forced to wrestle with the complexity if they are to use the relations correctly.

This complexity first manifests when an ontology developer chooses to migrate from a traditionally modeled ontology using non-temporalized relations to a TR model.

The ontology developer must choose on a case by case basis how to translate each axiom. It may be tempting to translate all to the at-all-times form, but as I noted previously in the spatial disjointedness case, this can lead to unexpected problems. In addition, the at-all-times relation is often formally wrong on the FOL level.

Even in cases where there is an optimal way to translate to TRs, performing the conversion requires an ontology developer who has a strong understanding of the domain and of the logic. In my estimation many mistakes will be made in this process.

4 DISCUSSION

4.1 Keep it as simple as possible, but not simpler

Many engineers abhor solutions that appear overly baroque or complex compared to the task at hand. From the point of view of the person proposing the complex solution, this can be frustrating and seem like short-sightedness on the part of the engineers. It may be tempting to dismiss this as a lack of vision to foresee the time in the future when the complexity of the solution will be justified.

Based on my experience in software and ontology engineering, I firmly believe we should take heed of engineers' concerns. **Extraordinary complexity demands extraordinary justification**. I have discussed the TR solution with a number of people with expertise ranging from ontology engineering, bioinformatics, biology and formal logic. Most of them find the TR solution excessively complex, unjustified by any use case, and those with an understanding of the logic realize the underpinnings are flawed. Perhaps we are collectively shortsighted or missing some aspect of the bigger picture? But even if this were the case, the fact that such a large section of the community finds the solution to be so unworkable demands means that the proponents of the solution need to work especially hard to justify the complexity.

4.2 On the strengths and weaknesses of OWL

One of the stated goals of the BFO2 OWL project is to have a clear FOL reading of the OWL according to BFO. This is a laudable aim but must be balanced against the needs of real-world users of ontologies. Their requirements should not be trumped by the desire for formal perfection.

We must also consider whether OWL is the best mechanism for achieving this kind of perfection. OWL is by design more restricted than first-order logic (which is itself arguably inadequate to model biology in anything other than a simplistic fashion). These restrictions make it more suited to certain kinds of tasks than others. In my experience OWL is tremendously useful for building and maintaining terminological networks that model the world in a very simple (but very useful) fashion. It's important to remember that we are working with approximations. Perhaps it is the case that attempting a perfect FOL reading of an OWL version of the BFO2 spec is simply using OWL in the wrong way? I have demonstrated in this report how reasoners become less useful as more is hidden from them (see the spatial disjointness example).

Whilst I would argue that we should be prepared for the fact that the OWL approximation may not have a perfect reading in terms of the BFO2 FOL, there may be reasons to be optimistic that this reading can be achieved without the sacrifices of the TR approach. For example, it may be possible to have a dual interpretation, or to use an approach called temporally qualified continuants (TQCs). These may have the benefit of being more closely aligned with the original **OBO-REL** paper. This is outside the scope of this paper, see [3] for details.

4.3 Recommendations

Based on my evaluation I make the following three primary recommendations:

1. **Do not use.** My primary recommendation is that Temporalized Relations should not be used as a replacement for existing atemporal relations. Ontologies should not migrate their production branch to TRs.
2. **Use the traditional approach.** Ontology developers should continue using the relations that they have been using successfully for years. These relations should be given stable identifiers and be resolvable. If a clearly better approach than the default strategy arrives then like any identifier they can be obsoleted, but this should only happen when the new approach is demonstrably superior.

3. **Change priorities.** Dogma should not trump the requirements of users of ontologies. A clear reading of the OWL in terms of the FOL is laudable but may have to be postponed.

I am not against the continued investigation of TRs as a long term strategy. My additional recommendations here are:

4.3.1 Mark as experimental. These relations should be clearly marked as experimental.

4.3.2 Documentation. More documentation is required. This includes both user-level documentation (for example: how do I choose the right temporalized form?) and documentation for the experienced engineer.

4.3.3 Alternate strategies. Alternative strategies must be given due consideration.

4.3.4 Justify major changes. If a strategy emerges that seems promising, yet that strategy requires major changes to ontologies, then these changes must be justified in terms of concrete benefits of the new approach. Concrete problems with the existing approach should be demonstrated.

4.3.5 Clearly indicate shortcomings.. In a language that a typical ontology developer can understand.

4.3.6 Provide a road map. If a promising strategy emerges, then users should not be expected to migrate to it if there is no road map indicating when missing features will be implemented. This would be common sense in software engineering, the same principles should apply to ontology engineering. An example of where a road map is missing is with the TR proposal. The release notes indicates that solutions to the lack of instantiation of non-rigid classes and permanent generic parthood are forthcoming, but there is no timeline.

4.3.7 Provide a migration plan.. An alternate strategy must come with a migration plan. I have experience in migration from legacy systems (for example, the transition from obo-format to OWL). This is rarely smooth, the process needs actively managed.

4.3.8 Take advantage of FOL.. If the primary motivating factor for a new approach is an FOL reading then the success of the approach should be determined based on some machine-readable FOL.

4.3.9 Provide tools.. If the use of a different approach requires additional tools for ontology editors to work effectively, then provide them, rather than promise them.

4.3.10 Remember the limitations of OWL.. Much as I love OWL, it isn't the right solution for everything. If you appear to be overloading OWL, consider if you might be better using something different.

5 CONCLUSIONS

Temporalized Relations would be a massive fundamental change to the way relationships are modeled in ontologies. They would introduce significant additional complexity to both users and developers of ontologies.

Some of these costs could be justified if Temporalized Relations were on a path towards making ontologies more biologically accurate. However, there are are a lack motivating use cases for this transition beyond the formal desire for a reading of the OWL in terms of FOL. Migrating to Temporalized Relations would lead to ontologies becoming *less* accurate, in addition to more complex. This lack of accuracy would have negative consequences on the utility OWL reasoning, and would result in a FOL reading that is false for many ontologies.

My recommendation is unambiguous in its rejection of the use of Temporalized Relations in their current form in biological ontologies.

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] N Grewe, J Hastings, L Jansen, F Neuhaus, A Ruttenberg, CJ Mungall, and S Schulz. Expressing time-dependent relations through temporal qualifications. *in prep*, 2012. `http://code.google.com/p/bfo/source/browse/trunk/bfo/docs/fois2012/TimeDependentRelations20120616.docx`.

[4] Chris Mungall. Proof of inaccuracy of part-of-al-all-times for cell nuclei. `http://bfo.googlecode.com/svn/trunk/src/ontology/owl-group/examples/`.

[5] Christopher J Mungall. Part disjointness. *Uberon reference manual*, 2012. `http://purl.obolibrary.org/obo/uberon/references/reference_0000018`.

[6] Stefan Schulz, Alan Ruttenberg, Melanie Courtot, and Janna Hastings. Bfo2 owl graz version release notes. `http://purl.obolibrary.org/obo/bfo/2012-07-20/ReleaseNotes`.

[7] 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.

[8] Barry Smith. Bfo2 owl graz version release notes. `http://purl.obolibrary.org/obo/bfo/2012-07-20/Reference`.

APPENDIX

5.1 Non-rigid classes

A class c is non-rigid if there exists an instance i that exists at t_1 and exists at t_2, and it is not the case that i instantiates c at t_1, and i instantiates c at t_2. t_1 may precede or succeed t_2.

1. 'premigratory neural crest cell' – This is a cell that is part of the neural crest region of the neuroepithelium, prior to migration. Whilst not all instances of this class become migratory, many do. In this case we have a cell instance i that instantiates 'premigratory neural crest cell' at t_1 and then at a subsequent time t_2, it instantiates 'migratory neural crest cell'. Therefore, 'premigratory neural crest cell' is a non-rigid class. The part_of relationship between this class and the neural crest cannot be adequately represented using TRs, because the cells may migrate.
2. 'human with Parkinson's disease' – a person is not born with Parkinson's (although they may be born with genes that predispose). It is possible for a human being i who exists at t_1 and t_2 to not instantiate human-with-PD at t_1 and to instantiate human-with-PD at t_2. Therefore human-with-PD is non-rigid. The has_disposition relationship between this class and the disease class cannot be adequately represented using TRs, as we need to say the members of the class have the disposition for all times that they instantiate the class.
3. 'female organism' – Some organisms (e.g. some species of arthropod) can change sex during their lifetime. It is possible for some such instance i to instantiate 'male organism' at t_1 and then instantiate 'female organism' at t_2. If these classes are disjoint, then 'female organism' is non-rigid. Note that it is possible to define different sex concepts (gender, karyotypic sex, biological sex, ...), different arguments can be made about the rigidity of the corresponding material entity classes. The has_quality relationship between this class and the sex quality class cannot be adequately represented using TRs, as we need to say the members of the class have the quality for all times that they instantiate the class.
4. 'infected lung' – it is possible for a lung i to have the quality of being infected (alternatively: be the location of a population of invading organisms) at t_1, and then non-infected at t_2. Therefore the class 'infected lung' is non-rigid. The location_of relationship between this class and the population class cannot be adequately represented using TRs, as we need to say the members of the class are the location for all times that they instantiate the class.
5. 'professor' – It is possible for an individual i to instantiate 'human with professor role' at one time, and then not instantiate this at some later time. Therefore 'professor' (as a material entity) is non-rigid. professorhood is best represented as a role that can be gained or lost.
6. 'human patient' – see 'professor'
7. 'fractured bone' – It is possible for some bone i to instantiate the class 'non-fractured bone' at t_1 and then 'fractured bone' at some later time t_2. Therefore 'fractured bone' (in the sense of a material entity - a bone that has the quality of being fractured) is a non-rigid class. An argument can be made that i ceases to exist when it becomes fractured, and is replaced by a new individual i_2 at t_2. This could certainly be argued for severe breakages, where i is replaced to two or more bone shards.
8. 'happy human' – It is possible for a human being ii to instantiate the class 'happy human' at t_1 (by virtue of bearing a happy disposition) and then 'unhappy human' at t_2. Therefore 'happy human' is a non-rigid class. The has_disposition relationship between this class and the disposition class cannot be adequately represented using TRs, as we need to say the members of the class have the disposition for all times that they instantiate the class.
9. 'fetal heart' – it is possible for a heart i to instantiate 'fetal heart' at t_1 and and then 'newborn heart' at t_2. Therefore 'fetal heart' is non-rigid. An argument can be made that i is ceases to exist and is replaced by a new instance i_2 at t_2, but this would be unusual. An argument could also be made that there is no need for a class 'fetal heart' - the concept should be described using a rigid class 'heart' together with an occurrent 'fetal stage'. However, this would be a severely onerous penalty on many anatomy ontologies which frequently use stage as a differentia.
10. 'open heart valve' – it is possible for some heart valve i to instantiate 'open heart valve' at t_1 (by virtue of bearing the quality 'open' at this time, or, alternatively, by virtue of their being a lumen in the vessel) and then to instantiate 'closed heart valve' at t_2. Therefore 'open heart valve' is a non-rigid class.
11. 'gravid uterus' – it is possible for some uterus i to instantiate a class 'non-gravid uterus' at time t_1 (by virtue of not being the location of a developing organism), and then instantiate a class 'gravid uterus' at some later time t_2. Therefore 'gravid uterus' is a non-rigid class. The has_part class axiom between this class and the embryo class cannot be adequately represented using TRs, as we need to say the members of the class are the location for all times that they instantiate the class.
12. 'phosphorylated EGFR protein' – there are different ways to model this depending on identity conditions on the instance level (we take identity conditions on the class level as being uncontroversial - class equivalence is determined by structure for molecules). Using model M_1, we assume there to be a single instance i of an EGFR protein which transitions through different states. Here, i instantiates 'unphosphorylated EGFR' at t_1 and then later the same instance i instantiated 'phosphorylated EGFR' at t_2. Under this model, 'phosphorylated EGFR' is non-rigid. We can model this differently - call this M_2. Here i_1 instantiates 'unphosphorylated EFGR' at t_1. Then, as a phosphate group is added at t_2, i_1 ceases to exist and its place is taken by i_2, which instantiates 'phosphorylated EGFR'. Here i_1 and i_2 might be related via some relation such as 'transformation of'. This illustrates that any non-rigid class can be made rigid by changing instance-level identity conditions. At one extreme we can see life as a series of snapshots, with individuals living for an instant before being replaced by a doppelganger.
13. 'cytoplasmic NFkB' – this is similar to the EGFR case. Here the differentia is location. No structural change need take place. The located_in or part_of class axiom between this class and the cytoplasm class cannot be adequately represented using

Relation	Axiom
x part-of- at-all-times y	$\leftrightarrow \forall t$exists_at$(x, t) \rightarrow$ exists_at(y, t), part_of(x, y, t)
x part-of- at-some-times y	$\leftrightarrow \exists t$exists_at$(x, t) \rightarrow$ exists_at(y, t), part_of(x, y, t)

Table 3. Temporalized relations axioms for parthood relations. Taken from [6] and transcribed into FOL syntax

TRs, as we need to say the members of the class are part of the cytoplasm for all times that they instantiate the class.

14. any leaf node from PATO – Examples: square, open, cylindrical, hot, cold. This is a multi-faceted topic and a thorough discussion should wait until there is documentation on how to model quantities in BFO2.

5.2 Non-rigid qualities

When considering BFO2 in particular, there is an interesting pair of statements that are difficult to reconcile.

1. The reference document[8] explicitly states that *determinates* (for example, qualities such as "square", "charged", "cylindrical") are non-rigid
2. The BFO2 OWL Graz release notes[6] explicitly states that instantiation of non-rigid classes is not supported in this version of BFO2, i.e. they have no model.

The modeling implications of this disconnect are not clear (more documentation would help). A reasonable interpretation might be that leaf nodes of PATO are not supported in this version of BFO2.

It may be the case that future versions will address this, but there is no guarantee that this will not involve additional complexity when it comes to recording qualities.

5.3 First order logic axioms

This section contains some additional material on the first order logic axioms supporting the Temporalized Relations.

Table 4 shows the FOL for the two main temporalized versions of part_of. Table 3 shows the full FOL semantics of making OWL class axioms using the FOL relations.

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

Table 4. Semantics of class axioms with parthood example. The first row shows the biologically correct relationship (permanent generic parthood), given by **OBO-REL** semantics. The next two rows show two of the temporalized options - neither of these is equivalent to the **OBO-REL** version.