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 and extensive training.

I conclude that these relations should not be adopted by ontology developers as a replacement for existing atemporal relations. Migrating to these relations would be an expensive, error-prone process that would confuse and alienate users; the end result would be ontologies that are either formally incorrect or too weak to derive 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[11]. 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 would state that *every* cell nucleus is part of *some* cell at *any* given moment of time (for which the given cell nucleus exists). This statement has a precise formulation in first-order logic (FOL):

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

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 serves as a shorthand for writing complete FOL axioms. For example, writing cell nucleus $\mathsf{part\_of}^T$ cell is equivalent to writing the full FOL axiom above. (Note that in contrast to the original paper, I denote type-level relations via a ^{T} superscript).

1.2 OBO-REL and OWL

OBO-REL initially served as the foundational semantics for many ontologies in the OBO library[13]. However, with the decision to define OBO-Format in terms of the Web Ontology Language (OWL)[4, 7], the role of the **OBO-REL** interpretation became unclear. To see why, consider a typical OWL axiom from the Gene Ontology[2]:

```
'cell nucleus' SubClassOf part_of some cell
```

The part_of relation (or Object Property, in OWL terminology) used here is not the same as the ternary time-indexed instance relation from **OBO-REL**, neither is it the same as the type-level counterpart. The semantics of the OWL axiom are atemporal, and translate to:

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

This is clearly different to what is specified in the **OBO-REL** paper. In practice, the developers of ontologies subscribe nominally to the **OBO-REL** interpretation but use binary atemporal OWL object properties from the new version of the OBO Relations Ontology[1].

Assessing the negative impact of this interpretation mismatch is not straightforward. For most practical purposes it has not caused any problems, but there are concerns regarding interoperability and how to use ontologies in combination with instance data. There are ongoing efforts to align these interpretations, but I do not discuss them here - instead I focus on a proposal that does away with these atemporal relations, and replaces them with explicitly temporalized relations, coupled to the release of BFO2.

1.3 BFO2 and temporalized relations

The Basic Formal Ontology (BFO) is an upper level ontology intended for use in the sciences[5]. Version 2 of BFO (BFO2) comprises two components: (1) the BFO2 *reference*[12] which is specified independently of OWL, and (2) the OWL realization of this reference. The reference is specified using FOL and uses ternary time-indexed relations akin to the instance-level relations of the original **OBO-REL** paper. One of the goals of the group developing the OWL version was to have relations in OWL handled so as to have a clear First-Order Logic (FOL) reading according to the BFO reference. This is enshrined in principle number 1 (called P-1 from here on) from the BFO wiki[2]:

> *A clear reading of the OWL version in terms of BFO reference. A translation is a mapping from the OWL model - axioms,*

*to whom correspondence should be addressed

[1] Google scholar

[2] http://code.google.com/p/bfo/wiki/Proposal_for_how_we_manage_OWL_Reference_coordination

entities, relations, etc. To the formal language used in the reference, currently FOL. A reading can be considered a data transformation that takes asserted and inferred axioms and results in FOL using types defined in BFO2 reference. This translation should be complete - no assertions in the OWL file can be left untranslated. In our current version, the most obvious missing reading is of binary relations with the same name as the ternary time-indexed relations in BFO2 reference.

The BFO group explored different approaches for modeling relations in OWL in a way that is conformant with P-1. One strategy involves retaining retaining simple atemporal binary properties in the OWL, and adhering to P-1 by means of different or additional interpretations of the OWL semantics[6]. This can be considered a kind of "default strategy". Another approach involves treating ontologies as capturing "snapshots", so-called *Temporally Qualified Continuants*. In this document, I do not consider these alternatives, and I focus solely on the *temporalized relations* (TR) approach, in which each ternary time-indexed relation has two or more OWL ObjectProperty cognates, rel-at-some-times and rel-at-all-times. This approach was used to create the BFO2 OWL Graz release[10], which is intended to be the basis of the final version of BFO2.

This approach has proven unpopular with ontology developers and users of ontologies largely due to confusion over the proliferation of relationship types[9]. In this critique I demonstrate that the problems with this approach run deeper than mere profusion of relations, and that this approach has severe negative impact for both short and long term use in OWL ontologies.

1.4 Outline

In this critique I focus solely on the temporalized relations approach. I do not attempt to compare or even describe the different modeling possibilities (see [6] for details here).

In section 2 I 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. Then in section 3 I present the major problems posed by these relations:

1. **logic issues**: Many problems arise from the logical underpinnings of the temporalized relations approach. The semantics of these relations cannot be expressed in OWL, limiting our ability to use OWL reasoners. Furthermore, the semantics of these relations are different from those intended by ontology developers, leading to ontologies that are either formally not valid or incomplete.
2. **usability issues**: 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 unintuitive 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.

After illustrating these logical and usability problems, I end with some concrete recommendations regarding the temporalized relation strategy (section 4).

Object Property	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)

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

2.1 Preliminary definitions

To avoid confusion, it is necessary to be clear about the different forms and interpretations of relations.

1. *Reference Relations* (RRs) include instance-level relations specified in the BFO2 reference. An example is the continuant version of the part_of relation. For space reasons I focus solely on ternary time-indexed RRs in this critique. These relations cannot be directly expressed in OWL.
2. *Temporalized Relations* (TRs) are OWL object properties that correspond to an RR combined with a temporal qualifier. An example is part-of-at-all-times. As these are OWL object properties, they are necessarily binary and instance-level.
3. *Class-level Relations* (CLRs) are binary relations that hold between classes, as specified in the original **OBO-REL** paper. I focus neither on these relations nor on difficulties representing these in OWL, but sometimes it is necessary to mention these to shown comparisons with the original OBOREL paper.
4. *Atemporal Relations* (ARs) are OWL object properties that are typically in one-to-one correspondence with RRs, but are not explicitly temporalized. As these are OWL object properties, they are necessarily binary and instance-level. Table 1 shows some example ARs and their characteristics. At this time, ARs lack a defined, agreed upon temporal reading (i.e. a mapping to FOL that introduces the temporal argument), which motivates the replacement of ARs with TRs in the current BFO2 proposal.

As a typographical convention, I use dashes to separate words in a TR (e.g. part-of-at-all-times). Underscores are used to separate the words in a RR or AR (e.g. part_of). RRs and ARs are disambiguated by their context - OWL axioms use ARs and 3-argument FOL sentences use RRs.

On the occasions where CLRs are used, a ^{T} is suffixed onto the relation name (e.g. part_of^T).

2.2 Translation template

The BFO2 Graz version release notes[10] 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)
```

I interpret this as macro-expansion rules[4].

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.

2.3 TRs do not mean the same thing as OBOREL

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 OBOREL statement cell nucleus part_of^T cell. If it were, this would be very convenient - we could simply use part-of-at-all-times every time an ontology author intended the **OBO-REL** interpretation, facilitating automatic translation of an ontology to a TR form.

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 adopting the TR strategy.

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.

2.4 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[2], 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).

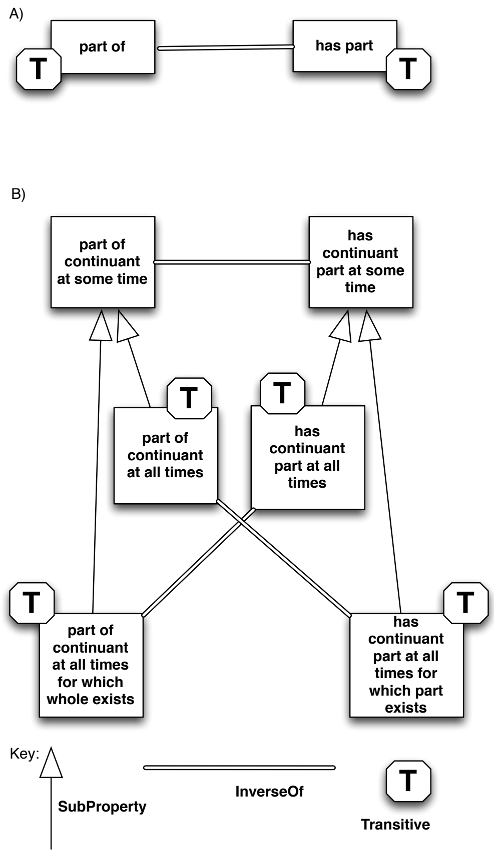

Fig. 1. Parthood relations, both atemporal (A) and temporalized (B). The atemporal form can be found in the current RO. The more complex temporalized form can be seen in the BFO2 Graz release.

2.4.1 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.

[4] Awaiting confirmation - see http://code.google.com/p/bfo/issues/detail?id=154

2.4.2 Symmetricality Symmetricality does not always transfer from a RR to a TR. This may not be obvious, and should be explicitly spelled out.

As an example, consider the traditional atemporal adjacent_to object property, which is symmetric. If this relation were part of the BFO2 reference, it would be defined as being symmetric. When translating this to TRs, the symmetricality property should only be carried over in the weaker at-some-times form. Table 2 shows the characteristics.

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. Note that here I do not explicitly name the inverse of the all-times form, but this could be done, assigning a name such as adjacent_to at-all-times-for-which-partner-exists.

2.4.3 Sub-Properties In OWL, object properties can be organized into a sub-property hierarchy. As can be seen in figure 1, the stronger temporalized relation forms are connected to the weaker forms via Sub-Property axioms. This is the general pattern for any relation that holds between continuants (regardless of whether they are in the reference).

Where the BFO2 reference introduces sub-relations (for example "memnber part of" as a sub-relation of part_of), the TR SubPropertyOf hierarchy becomes a complex lattice. See figure 2 in the appendix for an example.

2.4.4 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 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.

2.4.5 Property Chains Property (relation) chains are an OWL construct that allows a relationship to be inferred between two entities from a chain of relationships, possibly using different relations. For example, the rule that if x has_part y and y 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. To see why, consider that for any pair of RRs, there are 3×3 combinations to consider. I omit discussion of this for space reasons here, see the BFO tracker[5] for further details.

2.4.6 Other characteristics I do not consider other characteristics such as anti-symmetry and domain/range restrictions here. These may be included in a future version of this document.

2.5 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.

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 yielded better ontologies. However, in many cases *none of the TR 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 caused by use of 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[3] which has been very useful for error checking in GO and anatomical ontologies (which currently use ARs). This pattern is exemplified by the following axiom stating that nothing is part of both a mitochondrion and a nucleus[6]:

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

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

[6] For the original application of this pattern, see http://ontogenesis.knowledgeblog.org/1260

In translating this from an AR axiom to a TR axiom, we must decide on the form of TR to use. The default choice might be the stronger at-all-times form[7]:

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

However, on the FOL level, this is too *weak*. This weakness leaks into the OWL, 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[9].

In order to approximate the level of error checking we have with ARs, it is necessary to add **four** TR axioms, for each of at-all-times possibilities.

```
1. (part-of-at-all-times some mt)
    DisjointWith
   (part-of-at-all-times some nucleus)
2. (part-of-at-all-times some mt)
    DisjointWith
   (part-of-at-all-times-that-whole-exists
      some nucleus)
3. (part-of-at-all-times that whole exists
     some mt)
    DisjointWith
   (part-of-at-all-times-that-whole-exists
     some nucleus)
4. (part-of-at-all-times some nucleus)
    DisjointWith
   (part-of-at-all-times-that-whole exists
     some mt)
```

This introduced significant complexity into ontology development, probably requiring new tools. Even with this additional complexity, the FOL (i.e. the interpreted meaning of the axioms) still falls short of what is required.

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[8]. 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[10]. 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 not unique), 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 either make statements that are formally incorrect, or to make statements that are correct but are so weak the ontology cannot be used for the purposes for which it was created.

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 represents a cell that is part of the **neural crest**[11], prior to migration. In the Cell Ontology, this is modeled as:

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

The desired temporal reading is in accord with the standard **OBO-REL** pattern, in that the parthood relationship is true for the duration of 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 handle *non-rigid classes*.

[7] The part of the BFO2 documentation guiding how to translate an AR ontology to a TR ontology has yet to be written, but we assume that the stronger form is default

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

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

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

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[10] state:

> Thus we only instantiate "rigid" classes, as the interpretation we take is a rdf:type C $\Rightarrow$ forall(t) a exists at t $\rightarrow$ 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 a TR ontology such as BFO2:

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 modeling decisions should be made on a case by case basis for each ontology, rather than being 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 (e.g. "embryonic heart"). 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, and the resulting falsehoods will leak into the OWL representation.

3.2 Temporalized relations add considerable complexity

In addition to logical problems with TRs, the complexity they incur causes significant problems (including obscuring the modeling problems mentioned in the previous section).

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 figures 1 and 2. Note that in fact this figure is a simplification, as it only shows a subset of actual relations used in many ontologies. 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 — each property chain involving a chain of length n can generate 3^n possibilities when translated to TRs.

It is clear that many ontologies that currently make use of a rich set of ARs will become much more complex when translated to TRs. Furthermore, tools to detect errors in this complex network do not exist.

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-whole-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 SOME or ONLY quantifier in the superclass. Finally, embedded within the relation is an additional layer is the temporal quantification. Importantly, this final inner layer of quantification is opaque to OWL reasoners, meaning the correctness of these statements will have to be checked either by humans well-versed in logic, or using tools that are unlikely to be written for some time.

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

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 It may be argued that some of this additional complexity could be mitigated 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 end-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 demonstrated 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 RECOMMENDATIONS

I provide two sets of recommendations - one for consumers of BFO, and the other for BFO developers.

4.1 Recommendations for consumers of BFO

Here, consumers of BFO primarily refers to developers of ontologies that need to make use of core relations or classes from an upper ontology.

1. **Do not use Temporalized Relations.** My primary recommendation is that Temporalized Relations should not be used in production ontologies. They may be used in experimental or exploratory ontologies, or in experimental branches of production ontologies, but they should not be used for the main version of a production ontology. Here “production ontology” means any ontology that satisfies the OBO Foundry criteria of having a plurality of users.
2. **Continue to use atemporal relations.** Ontology developers should continue using the relations that they have been using (largely without facing any problems) for years. The Relations Ontology will continue to provide OWL Object Properties for (atemporal) part_of and so on. At this time, some of these relations are in the BFO ID space (for example, part_of has the ID BFO_0000050). Currently these are considered to be a part of RO rather than BFO, despite their ID space.

   The RO group will continue to support these relations indefinitely. It is our hope that we will be able to provide an **OBO-REL** style interpretation of these. If it at some point in the future some superior alternative to the existing relations are discovered, a migration path will be provided.

4.2 Recommendations for the BFO group

I also provide some recommendations to the BFO group, based on my experience evaluating these relations.

4.2.1 Remember that the OWL version of BFO is the most important The group developing BFO OWL is a subgroup of the larger BFO group. However, the group as a whole should take an active interest in the OWL version, because for every single user of BFO, the OWL version is synonymous with BFO. Nobody is using BFO who is not using OWL.

4.2.2 Documentation. More documentation on Temporalized Relations is required. This includes both user-level documentation (for example: how do I choose the right temporalized form?) and documentation aimed at the experienced ontology engineer.

4.2.3 Advertise limitations and problems clearly. This critique has identified or explored a number of issues with TRs. Some of these may not have been apparent to users reading the BFO documentation - some were in fact not apparent to many of the members of the BFO team.

In cases where the BFO documentation does note limitations, the consequences of the limitation are not made clear. For example, the release notes currently state that non-rigid classes are not supported at this time. To the average ontology developer, the explanation is quite opaque and the consequences of the limitation are not clear. In fact the consequences are not clear to experienced ontology engineers, as it is stated ambiguously.

If BFO is to retain the trust of the community then it must be very clear and upfront about the weaknesses and limitations of the product it is pushing. The shortcomings must be described in a language the average BFO consumer can understand. Examples must be given.

4.2.4 Mark temporalized relations as experimental. These relations should be clearly marked as experimental or exploratory.

4.2.5 Alternate strategies. Alternative strategies to adhering to P-1 should be given due consideration. The BFO team should not prematurely settle on the TR solution, especially now its limitations have been elucidated. Alternate solutions such as those currently being worked on (dual interpretations; temporally qualified continuants)[6] may turn out to preserve P-1 at less of a cost to the ability to BFO consumers to deliver a usable product to their users.

4.2.6 Justify major changes. The TR approach, if adopted, would be a major change requiring lots of resources to migrate. If the community is to be convinced that this approach is superior, then clear examples of where the previous approach leads to reasoning or interoperability problems should be provided. Alternatively, examples of where the TR approach allows you to get inferences you would not get with the traditional approach.

Currently the BFO documentation provides no such examples. The sole justification for the major shift is that the traditional approach does not yet have an agreed upon a formalluy documented way to interpret the OWL such that it satisfies P-1.

In short, extraordinary change demands extraordinary justification[13].

4.2.7 Provide a road map and migration plan. If a promising alternate strategy to atemporal relations emerges, then users should not be expected to migrate to it if there is no roadmap 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. It is not clear if the timescape is months or years, or what ontology developers are expected to do in the interim period.

The TR approach, if adopted, would require BFO consumers migrating away from atemporal relations to the temporalized form. This would be a difficult task, and users would need assistance.

4.2.8 Do a cost-benefit analysis. The primary justification for the TR approach is that it satisfies P-1. The fact that adoption of TRs would lessen the ability of BFO consumers to deliver on their use cases is regarded as secondary.

This is not how an engineering project should work. The costs of deferring on P-1 (which the community has effectively been deferring for years) should be weighed against the costs inherent in the TR approach.

A cost benefit analysis should include[14]:

> this includes aspects such as cost of training for use in development, cost to end users in understanding ontologies built using such frameworks, cost benefits measured as per metrics such as those above (e.g. answering competency questions) and risk of adoption (such as significant changes or longer term support).

4.2.9 Provide tools.. If the use of a different approach requires additional tools for ontology editors to work effectively, then these should be provided, or a strategy and timeline for their development should be included.

For example, creation of TRs involves creation of complex lattices of relations due to the proliferation of different temporalized forms. Other problems aside, adoption of TRs would require better tools for managing and presenting this.

[13] This advice should be heeded for other parts of BFO as well, such as the unpopular process profiles

[14] http://jamesmaloneebi.blogspot.co.uk/2012/07/why-choosing-ontologies-should-not-be.html

4.2.10 OWL is limited by design.. The OWL language and associated tooling is incredibly useful in bioinformatics. In particular, it has proven invaluable for automatically classifying and finding errors in large graphs of terms such as the Gene Ontology.

However, it is not the best language for everything. In particular, complex spatiotemporal reasoning may not be its forte.

If it feels like overloading, it may be time to use something else. Work to OWL's advantages, not its strengths.

5 DISCUSSION

5.1 Complex solutions are prone to failure

In this critique I have tried to focus on the inherent logical problems with the TR approach, and how these can impact ontology modeling and the use of reasoners.

An additional concern is the complexity of the approach. 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 modeling limitations and flawed logical underpinnings.

Perhaps this group of us are collectively shortsighted or missing some aspect of the bigger picture? I think this unlikely, but even if we take this to be the case, the fact that such a large section of the bio-ontology community finds the solution to be so problematic places an onus on the proponents of TRs to work especially hard to justify the complexity. If this concern is ignored, then engineers will naturally migrate to a simpler solution that allows them to deliver a working product.

5.2 BFO principle 1 and 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 (P-1). This is a laudable aim but must be balanced against the needs of real-world users of ontologies and the capabilities of the OWL languages. Their requirements must 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. OWL is tremendously useful for building and maintaining graphs of terms that model the world in a very simple (but very useful) fashion. We should be wary of attempts to over-extend these useful 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? As demonstrated in this critique, complicating the OWL representation *decreases* the utility of OWL reasoning.

The most sensible strategy is to *defer* on P-1. It may be the case that satisfying P-1 cannot be done whilst retaining an OWL model that is usable. Whilst we should be prepared for this, there is in fact grouns for optimism. It may be the case that there are alternatives to TRs that satisfy P-1 without the same sacrifices 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 [6] for details.

6 CONCLUSIONS

If adopted, 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]OBO Relations Ontology (RO), 2010. http://obo-relations.googlecode.com.

[2]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.

[3]Uberon editors. Part disjointness. *Uberon reference manual*, 2012. http://purl.obolibrary.org/obo/uberon/references/reference_0000018.

[4]C. Golbreich and I. Horrocks. The obo to owl mapping, go to owl 1.1. In *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions: June 6-7 2007; Innsbruck, Austria*, 2007.

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

[6]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.

[7]C Mungall, D Osumi-Sutherland, A Ruttenberg, and I Horrocks. The obo format specification - http://purl.obolibrary.org/obo/oboformat/spec.html. 2012.

[8]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/.

[9]Multiple ontology authors. pers. comm.

[10]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.

[11]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.

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

[13]Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, The OBI Consortium, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuermann, Nigam Shah, Patricia L Whetzel, and Suzanna Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol*, 25(11):1251–1255, Nov 2007.

APPENDIX

6.1 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. With this definition, the following classes are non-rigid:

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 a predisposition). 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 invasive pathogens) 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; but this would be curious for less severe fractures, where it is clearly the same bone instance before and after the fracture.
8. 'happy human' – It is possible for a human being i 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 spanning the length of 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. Here we assume that this is not the intent of the authors of the

Relation	Axiom
all	x part-of-at-all-times y $\leftrightarrow (\forall t \text{exists\_at}(x,t) \rightarrow$ $\text{exists\_at}(y,t), \text{part\_of}(x,y,t))$
some	x part-of-at-some-times y $\leftrightarrow (\exists t \text{exists\_at}(x,t) \rightarrow$ $\text{exists\_at}(y,t), \text{part\_of}(x,y,t))$

Table 3. Temporalized relations axioms for parthood relations. Taken from [10] and transcribed into FOL syntax. TODO at-all-times-partner-exists. TODO fix according to latest bfo

BFO document, and that continuants endure through changes in their properties.

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 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.

6.2 Non-rigid qualities

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

1. The reference document[12] explicitly states that *determinates* (for example, qualities such as "square", "charged", "cylindrical") are non-rigid
2. The BFO2 OWL Graz release notes[10] 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 (additional BFO documentation would help here). 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.

6.3 First order logic meanings of parthood

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.

6.4 Subproperty lattices

The TR strategy involves generating two or more temporalized relations for every reference relation. These are typically connected in a hierarchical fashion, with the stronger all-times forms being subproperties of the weaker some-times forms.

Axiom	FOL meaning
cell nucleus part_of^T cell	$\forall x \forall t : \text{instance\_of}(x, \text{cell nucleus}, t) \rightarrow$ $\exists y : \text{instance\_of}(y, \text{cell}, t), \text{part\_of}(x,y,t)$
cell nucleus SubClassOf part-of-at-all-times some cell	$\forall x : \text{instance\_of}(x, \text{cell nucleus}) \rightarrow$ $\exists y \text{instance\_of}(y, \text{cell}), \forall t \text{exists\_at}(x,t) \rightarrow$ $\text{exists\_at}(y,t), \text{part\_of}(x,y,t)$
cell nucleus SubClassOf part-of-at-some-times some cell	$\forall x : \text{instance\_of}(x, \text{cell nucleus}) \rightarrow$ $\exists y : \text{instance\_of}(y, \text{cell}), \exists t :$ $(\text{exists\_at}(y,t), \text{part\_of}(x,y,t))$
cell nucleus SubClassOf part_of cell	$\forall x : \text{instance\_of}(x, \text{cell nucleus}) \rightarrow$ $\exists y : \text{instance\_of}(y, \text{cell}), \text{part\_of}(x,y)$

Table 4. Various FOL meanings of different renderings of the parthood relation. The first row shows the relationship, given by **OBO-REL** semantics. This is called permanent generic parthood, and is the one intended by the ontology authors. There is currently no agreed upon way of expressing this in OWL. The second two rows show the meaning of class axioms using TRs. Note the difference between row 1 and row 2 - row 2 is known as *permanent specific parthood*. Its use is formally incorrect in this case due to the specifics of cell division [see bfo examples directory for details]. The 3rd row shows use of a weaker TR, in which transitivity us lost. The last row shows the meaning of a class axiom with an AR. This yields the intended inferences, but is problematic as it is not precisely aligned to the BFO2 reference, due to the lack of a temporal reading.

When the reference relations themselves form a hierarchy, the resulting TR structure is a lattice, as found for example in figure 2

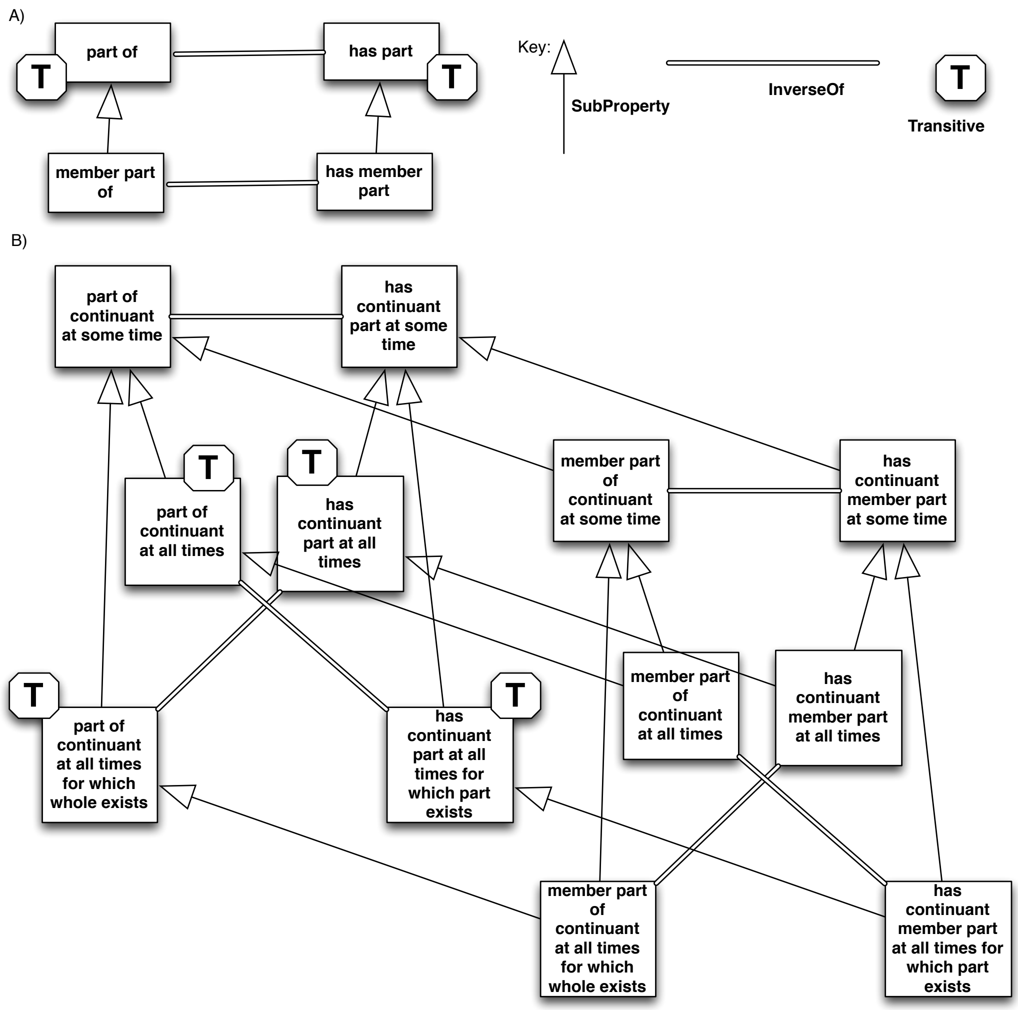

Fig. 2. Parthood and member-parthood relations, both atemporal (A) and temporalized (B). The atemporal form can be found in the current RO. The more complex temporalized form can be seen in the BFO2 Graz release. As can be seen, the already complex structure found with the basic parthood relation is made even more complex with the addition of sub-relations.

.