Original Research Articles

Pages 1–6

# A critique of rigid temporalized relations

Christopher J. Mungall [1*]

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

**ABSTRACT**

In this 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[6]. 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

As part of the release process for the OWL translation of version 2 of the Basic Formal Ontology (BFO)[2], a number of people explored different strategies for unifying OWL binary properties with the ternary relations in the BFO2 reference specification[3]. The goal was to find a way of representing the temporal aspects of BFO2 relations in OWL in a formally satisfying way. 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[5].

### 1.3 Outline

In this review I do not attempt to compare or even describe the different modeling possibilities; instead I focus purely on the temporalized relations strategy, due to pressure to make this the standard for OBO library ontologies. 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 fail to capture the biological reality, forcing ontology editors to make a choice between two unsatisfactory models.
2. Within the model there is no facility for modeling important classes that lack the 'rigidity' criterion.
3. Temporalized relations are confusing for both users and experienced ontology editors.

---

*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)

These problems are inter-related; even experienced ontology editors may not understand the choices they are being asked to make when migrating to temporalized relations.

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

My intentions are to make this review accessible to a wide audience, and to keep logical formulas to a minimum. This is difficult because one of the problems with the TR strategy is that it forces complexity upon both the users and developers of an ontology, necessitating some discussion of that complexity in order to explain why it does not yield the intended benefits. I have annexed some of the finer grained details into an appendix.

## 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[5] specify a general template for relating RRs to TRs:

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

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

For part_of connecting two continuants there are 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", we shorten this for brevity – here we are only concerned with relations that involve a continuant.

### 2.2 TRs force a different interpretation from OBOREL

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

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.* Whereas **OBO-REL** allowed permanent generic parthood (in which a nucleus must always be part of a cell, but can be transferred between cells), with TRs that possibility is disallowed. See Table 5 in the appendix for details. I will return to this in the evaluation.

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

### 2.3 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 doesn't follow that the temporalized versions will be inverses. Ideally we would be able to prove that the RR and TR are consistent, although the proof may be obvious to an expert logician. Here I use part_of as an illustrative example, although each relation may require individual examination for its properties.

Table 2 shows some of the property characteristics of the various forms of the continuant parthood relations. Figure 1 shows this in graphical form.

*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 safer to use in an ontology, but will lead to fewer inferences.

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

[2] we are only considering instance level relations

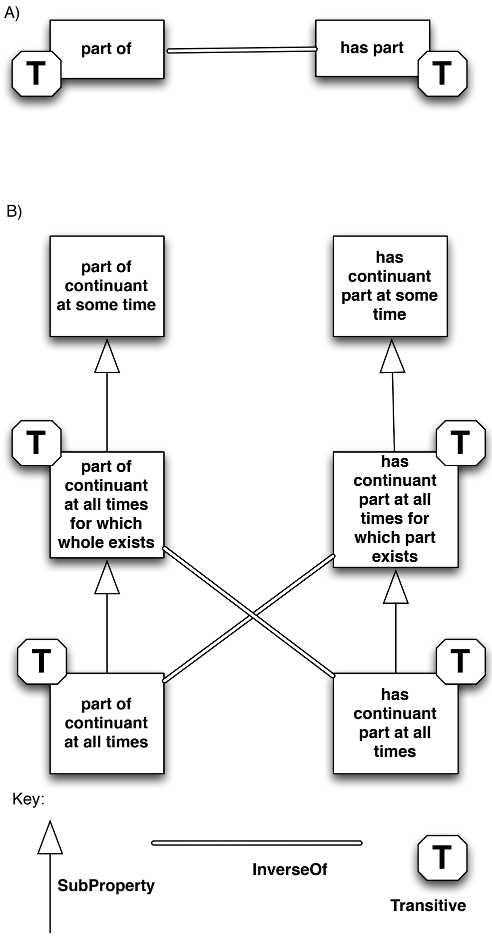


**Fig. 1.** Parthood relations, both atemporal (A) and temporalized (B)

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

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

Currently BFO2 does not have an adjacency relation, or any other reference relation that is symmetric, so the following is based on my own understanding, and is shown in 3. I would assume that the symmetricality should be declared for the at-some-times form and not the at-all-times form. In contrast to the part_of and *transitivity*, here it is the *weaker* form of relation that inherits the characteristic.

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

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

*2.3.3 Sub Properties* We assume a hierarchy in which at-some-times is the most general, with at-all-times-for-which-subject-exists intermediate, and at-all-times most specific. This is illustrated for parthood in figure 1.

Note how this interacts with other properties. If:

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

then we can infer that

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

.

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

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

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

*2.3.5 Other characteristics* We do not consider other characteristics such as anti-symmetry and domain/range restrictions here.

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

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

## 3 EVALUATION

### 3.1 Temporalized Relations do not reflect the intentions of ontology editors or the underlying biology

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

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

Understanding the consequences of incorrectly modeling the relationship in this way are not straightforward. In some situations there may be no severe problems, if the ontology is not used for instance data. However, it would be unusual to adopt a far more complex formalism (TRs) in the name of formality only to arrive at an ontology that is formally incorrect.

Given that the ontology editor can not use part-of-at-all-times without making a false statement, they must choose a weaker qualified version of part_of such as part-of-at-some-times. Such a usage would be formally valid, but incomplete from the point of view of useful reasoning. This is because the weaker at-some-times form lacks the transitivity characteristic. I assume there to be no disagreement that this is a completely unsatisfactory solution, as most ontologies are dependent on parthood transitivity.

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

In the nucleus-cell case, the ontology editor requires the permanent generic form in order to be both accurate and to get the required inferences. *However, this form is specifically excluded in the TR strategy*.

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

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

### 3.2 Rigidity requirement is too onerous

The BFO2 Graz release notes state:

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

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

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

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

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

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

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

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

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

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

### 3.3 Temporalized relations add complexity

The most striking feature of an ontology that uses the TR strategy is the complexity. Whereas using traditional modeling, we may have has a single parthood relation, we now have three. Both users and developers have to have a strong understanding of these to use them correctly.

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

```
population SubClassOf
  has-part-at-some-time some organism
population SubClassOf
  has-part-at-all-times some organism
population SubClassOf
  has-part-at-all-times-that-part-exists some organism
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 than is opaque to OWL reasoners (and thus harder to use standard tools to check), i.e. the temporal qualification.

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 much.

*3.3.2 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.

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

For each axiom that uses a relation that has multiple variants in TR form, the developer must make a choice of which of the 3 variants to use. Currently there is no documentation or guidance on how they should do this.

A conservative strategy would be to convert all continuant relations to the at-some-times form. This would result in the ontology being far less useful for inference (due to the loss of properties such as transitivity). However, we might expect it to be at least valid, since the relation is weaker.

In fact this is not universally safe. Consider the OWL axiom:

```
(\partOf\ \some\ \pr{nucleus}) \pr{DisjointWith} (\par
```

Converting these to at-some-times would actually result in an axiom that is *too strong*. Conversely, converting to at-all-times would be too *weak*, because it would admit the possibility of migratory structures being part of two spatially disconnected locations at the same time (so long as they weren't permanently part of each). It is not clear how the ontology maintainer should convert this axiom – *because all choices are suboptimal*.

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.

### 3.4 Case study: HDOT

...

## 4 DISCUSSION

### 4.1 Bad smells

Software engineers have a concept of a "bad smell" in code. This is often applied when a piece of software or library seems overly baroque or complex. Whilst this sense may be at least in part subjective, many software engineers work on this intuitive level, and will typically avoid using a piece of software if they perceive it to have a "bad smell". Sometimes there is external pressure to use the software in question - this can sometimes prolong its life but in general software engineering solutions perceived to have this quality are typically replaced by something more efficient and simpler.

From various conversations with experienced ontology engineers, bioinformaticians and logicians, I feel confident in saying that the TR solution as embodied in BFO2 is generally perceived to have a "bad smell" in the same way. I include this as a discussion point rather than part of a formal evaluation since I can't produce hard evidence of this quality. However, proponents of the TR solution ignore this engineers' intuition at their peril: it hinders the adoption of the relations as standards, and even if this is forced from above, developers will work around them. I know of various groups with plans to simply rewrite TRs back into their standard form should they become adopted.

### 4.2 Recommendations

*4.2.1 Do not use* My primary recommendation is that Temporalized Relations should not be used as a replacement for existing atemporal relations. Ontologies should in general not migrate to them.

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

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

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

*4.2.5 Road map* The existing TR proposal embodied in the BFO2 Graz release is not complete. For example, of non-rigid classes, the release notes state that "we are working on [non-rigid classes] for the future". There shoould be a roadmap indicating when these solutions are expected to transpire. Ontology developers should not be expected to commit production ontologies to an experimental project with no roadmap.

*4.2.6 Smooth transition* If TRs are to be adopted, there needs to be some incremental transition plan for migration.

## 5 CONCLUSIONS

Temporalized Relations would be a 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 no mativating use cases for this transition, and in fact migrating to Temporalized Relations would lead to ontologies becoming *less* accurate, in addition to more confusing. My recommendation is unambiguous in its rejection of the use of Temporalized Relations in biological ontologies.

## ACKNOWLEDGMENTS

## REFERENCES

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

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

[3]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.

[4]Chris Mungall. Proof of inaccuracy of part-of-al-all-times for cell nuclei.

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

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

## APPENDIX

### 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 $i$i instantiates $c$ at $t_2$. $t_1$ may precede or succeed $t_2$.

1. 'human with Parkinson's disease' – a person is not born with Parkinsons (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.
2. '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.
3. '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.
4. '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.
5. 'human patient' – see 'professor'
6. 'oocyte' –
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 $i$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.
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. 'neural crest cell' –
11. 'open heart valve' – it is possible for some heart valve $i$ to instantiate 'open heart valve' at $t_1$ (by virtue of bearing the

| 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 4.** Temporalized relations axioms for parthood relations. Taken from [5] and transcribed into FOL syntax

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.

12. '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.
13. '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 doppleganger.
14. 'cytoplasmic NFkB' – this is similar to the EGFR case. Here the differentia is location. No structural change need take place.
15. 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 First order logic axioms

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

SECTION NOT COMPLETE IGNORE FOR NOW

| 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 5.** 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.