

# Representing the UFO-B Foundational Ontology of Events in *SRIOQ*

Alessander BOTTI BENEVIDES <sup>a,1</sup>, Jean-Rémi BOURGUET <sup>a</sup>,  
Giancarlo GUIZZARDI <sup>a,b</sup>, Rafael PEÑALOZA <sup>b</sup>

<sup>a</sup>*NEMO, Computer Science Department, Federal University of Espírito Santo, Brazil*

<sup>b</sup>*Faculty of Computer Science, Free University of Bozen-Bolzano, Italy*

**Abstract** In recent years, there has been a growing interest in the application of foundational ontologies, i.e., formal ontological theories in the philosophical sense, to provide a theoretically sound foundation for improving the theory and practice of conceptual modeling and knowledge representation. This paper addresses one particular foundational theory of events termed UFO-B, which has been successfully employed as a reference model for addressing problems from complex media management, enterprise architecture, software engineering, and modeling of events in petroleum exploration. Despite that, there is still no formalization of UFO-B in a decidable knowledge representation language that could support reasoning about complex events and event relations. We address this gap by proposing a number of alternative translations from UFO-B's original axiomatization (in first-order logic and in the Alloy formal language) to the description logic *SRIOQ*, which is the formal underpinning of OWL 2 DL. Additionally, to support practical applications, we translated these *SRIOQ* theories to OWL 2 DL TBoxes, which were validated by showing that all the intended models of UFO-B (the logical models of the UFO-B specification in Alloy) that we generated are consistent with these UFO-B TBoxes.

**Keywords.** Knowledge representation, Formal Ontology, Ontology of Events, Description Logics, OWL

## 1. Introduction

In recent years, there has been a growing interest in the application of foundational ontologies, i.e., formal ontological theories in the philosophical sense, to provide a theoretically sound foundation for improving the theory and practice of conceptual modeling and knowledge representation. This paper addresses a particular philosophically well-founded ontology of events termed Unified Foundational Ontology—Part B (UFO-B) [1]. Dealing with the proper representation of subtle aspects of events is fundamental for a multitude of areas such as bioinformatics, finances, and public safety. In particular, the UFO-B ontology has been extensively tested in practice and successfully employed as a reference model for addressing problems from complex media management, enterprise architecture, software engineering, and the modeling of events in petroleum exploration. Despite that, there is still no formalization of this ontology in terms of a decidable knowledge representation language that could support reasoning about complex events and

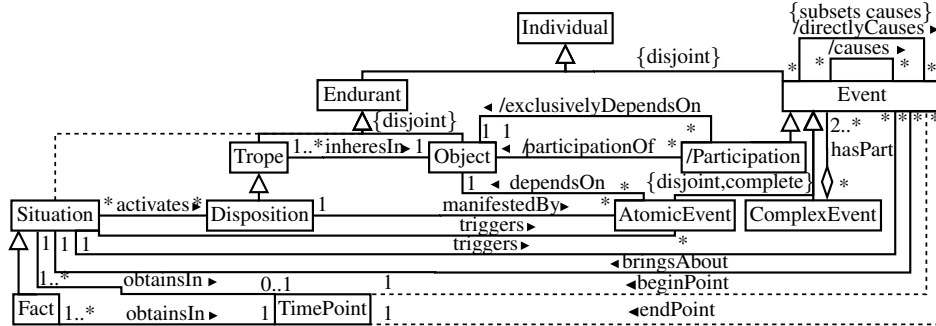
---

<sup>1</sup>Corresponding Author: Av. Fernando Ferrari, 514, Goiabeiras, Vitória, BR; E-mail: [abbenevides@inf.ufes.br](mailto:abbenevides@inf.ufes.br).

event relations. We address this gap by proposing a number of translations from UFO-B’s original axiomatization (in first-order logic (FOL) and in the Alloy formal language [2]) to the Description Logic (DL) *SR<sub>OIQ</sub>* [3], which is the formal underpinning of the OWL 2 Web Ontology Language (OWL 2 DL) [4]. The remainder of this paper is structured as follows. Section 2 briefly presents the original axioms of UFO-B together with mappings to *SR<sub>OIQ</sub>*, discussing the challenges in mapping these two formalisms and arguing that representing the totality of UFO-B in *SR<sub>OIQ</sub>* results in an ill-formed theory. In order to address the latter problem, in Section 3 we propose a number of translations that are maximal w.r.t. the syntactic constraints of *SR<sub>OIQ</sub>*, i.e., theories such that the inclusion of any other axiom makes them ill-formed. These translations are evaluated by means of codifications in OWL 2 DL terminology boxes (TBoxes), on which consistency tests are performed regarding a set of automatically generated intended models of UFO-B (logical models of the UFO-B specification in Alloy). Section 4 presents some final considerations.

## 2. The Original FOL Formalization of UFO-B and a Translation to *SR<sub>OIQ</sub>*

First, we present some notational conventions: (i) FOL predicates and DL concepts/roles are denoted in Uppercase typewriter type; (ii) we employ restricted quantification *à la Frege*, i.e., “ $\forall x:T(\varphi)$ ” is a schema for “ $\forall x(T(x) \rightarrow \varphi)$ ,” and “ $\exists x:T(\varphi)$ ” for “ $\exists x(T(x) \wedge \varphi)$ ,”; and (iii) we homogenized the names of relations defined in [1] by expanding abbreviations and replacing the forms ‘x-y’ or ‘x<sub>y</sub>’ with ‘xY’.



**Figure 1.** The union of the five class diagrams shown in [1, Figures. 1–5]. Dotted lines represent our additions.

UFO-B extended UFO-A [5] to deal with events (Figure 1), and was formalized by means of FOL formulae in [1] and also by means of an external Alloy codification. Here, we consider only the theory in [1], which is structured as follows: a definition of a parthood relation for events (Section 2.1); an explanation of the nature of events as manifestations of object dispositions (Section 2.2); constraints on how objects participate in events, and how events depend on objects (Section 2.3); an optional linear time structure and a temporal constraint on parthood (Section 2.4); and the view of events as the entities responsible for *world changes*, a sort of transition between *situations* (Section 2.5).<sup>2</sup>

<sup>2</sup>In the ontological stance adopted here, the codomain of the relation *inheresIn* is not restricted to *Objects*, since *Events* could also bear *Tropes* (we thank a reviewer for bringing that to our attention). Moreover, *Trope* is not the domain of *inheresIn*, since objectified intrinsic properties in general (e.g., modes [5]) *inheresIn* other

We formalize the (co)domain and cardinality constraints of relations according to their orientations. Two relation names occur twice in Figure 1: `obtainsIn` and `triggers`. We interpret the `obtainsIn` from `Fact` to `TimePoint` as a constraint on the domain of `obtainsIn` in case the first argument is a `Fact` (a51). On `triggers`, (i) as shown in [1, Figure 4], the codomain of `triggers` is `Event` (a30); (ii) in case of `AtomicEvents`, we have that a `Situation` `triggers` `AtomicEvent` if and only if (iff) there is a `Disposition` that is activated by the `Situation` and is manifestedBy the `AtomicEvent` [1, D3]. Finally, `TimePoint` and `Individual` are disjoint (a40).

*Subsumptions*

	<b>a4</b> <code>Situation</code> $\sqsubseteq$ <code>Endurant</code>	<b>a8</b> <code>AtomicEvent</code> $\sqsubseteq$ <code>Event</code>
	<b>a5</b> <code>Trope</code> $\sqsubseteq$ <code>Endurant</code>	<b>a9</b> <code>ComplexEvent</code> $\sqsubseteq$ <code>Event</code>
<b>a1</b> <code>Endurant</code> $\sqsubseteq$ <code>Individual</code>	<b>a6</b> <code>Disposition</code> $\sqsubseteq$ <code>Trope</code>	<b>a10</b> <code>Participation</code> $\sqsubseteq$ <code>Event</code>
<b>a2</b> <code>Event</code> $\sqsubseteq$ <code>Individual</code>	<b>a7</b> <code>Fact</code> $\sqsubseteq$ <code>Situation</code>	<b>a11</b> <code>directlyCauses</code> $\sqsubseteq$ <code>causes</code>
<b>a3</b> <code>Object</code> $\sqsubseteq$ <code>Endurant</code>		

*Domain Constraints*

	<b>a12</b> $\exists$ <code>inheritsIn</code> . <code>T</code> $\sqsubseteq$ <code>Trope</code>	<b>a18</b> $\exists$ <code>causes</code> . <code>T</code> $\sqsubseteq$ <code>Event</code>
	<b>a13</b> $\exists$ <code>hasPart</code> . <code>T</code> $\sqsubseteq$ <code>ComplexEvent</code>	<b>a19</b> $\exists$ <code>directlyCauses</code> . <code>T</code> $\sqsubseteq$ <code>Event</code>
	<b>a14</b> $\exists$ <code>dependsOn</code> . <code>T</code> $\sqsubseteq$ <code>AtomicEvent</code>	<b>a20</b> $\exists$ <code>activates</code> . <code>T</code> $\sqsubseteq$ <code>Situation</code>
	<b>a15</b> $\exists$ <code>obtainsIn</code> . <code>T</code> $\sqsubseteq$ <code>Situation</code>	<b>a21</b> $\exists$ <code>manifestedBy</code> . <code>T</code> $\sqsubseteq$ <code>Disposition</code>
	<b>a16</b> $\exists$ <code>triggers</code> . <code>T</code> $\sqsubseteq$ <code>Situation</code>	<b>a22</b> $\exists$ <code>beginPoint</code> . <code>T</code> $\sqsubseteq$ <code>Event</code>
	<b>a17</b> $\exists$ <code>bringsAbout</code> . <code>T</code> $\sqsubseteq$ <code>Event</code>	<b>a23</b> $\exists$ <code>endPoint</code> . <code>T</code> $\sqsubseteq$ <code>Event</code>
		<b>a24</b> $\exists$ <code>exclusivelyDependsOn</code> . <code>T</code> $\sqsubseteq$ <code>Participation</code>
		<b>a25</b> $\exists$ <code>participationOf</code> . <code>T</code> $\sqsubseteq$ <code>Participation</code>

*Codomain Constraints*

<b>a26</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>inheritsIn</code> . <code>Object</code>	<b>a32</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>causes</code> . <code>Event</code>
<b>a27</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>hasPart</code> . <code>Event</code>	<b>a33</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>directlyCauses</code> . <code>Event</code>
<b>a28</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>dependsOn</code> . <code>Object</code>	<b>a34</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>activates</code> . <code>Disposition</code>
<b>a29</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>obtainsIn</code> . <code>TimePoint</code>	<b>a35</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>manifestedBy</code> . <code>AtomicEvent</code>
<b>a30</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>triggers</code> . <code>Event</code>	<b>a36</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>beginPoint</code> . <code>TimePoint</code>
<b>a31</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>bringsAbout</code> . <code>Situation</code>	<b>a37</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>endPoint</code> . <code>TimePoint</code>
	<b>a38</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>exclusivelyDependsOn</code> . <code>Object</code>
	<b>a39</b> <code>T</code> $\sqsubseteq$ $\forall$ <code>participationOf</code> . <code>Object</code>

*Disjointness*

<b>a40</b> $(\text{TimePoint} \sqcap \text{Individual}) \sqsubseteq \perp$	<b>a42</b> $(\text{Object} \sqcap \text{Trope}) \sqsubseteq \perp$
<b>a41</b> $(\text{Endurant} \sqcap \text{Event}) \sqsubseteq \perp$	<b>a43</b> $(\text{Object} \sqcap \text{Situation}) \sqsubseteq \perp$
	<b>a44</b> $(\text{Trope} \sqcap \text{Situation}) \sqsubseteq \perp$
	<b>a45</b> $(\text{AtomicEvent} \sqcap \text{ComplexEvent}) \sqsubseteq \perp$

*Completeness*

**a46** `Event`  $\sqsubseteq$   $(\text{AtomicEvent} \sqcup \text{ComplexEvent})$

*Domain Cardinality Constraints*

**a47** `Trope`  $\sqsubseteq$   $(=1$ `inheritsIn`.`Object`)

**a48** `ComplexEvent`  $\sqsubseteq$   $(\geq 2$ `hasPart`.`Event`)

**a49** `AtomicEvent`  $\sqsubseteq$   $(=1$ `dependsOn`.`Object`)

---

Individuals. We ask the reader to interpret `inheritsIn` here as ‘`tropeInheritsInObject`,’ a specialization of the inheritance relation. In an extension of this paper, we deal with `Events` bearing `Tropes` and qualities.

**a50** Situation  $\sqsubseteq$  ( $\leq$ lobtainsIn.TimePoint)  
**a51** Fact  $\sqsubseteq$  (=lobtainsIn.TimePoint)  
**a52** Event  $\sqsubseteq$  (=lbringsAbout.Situation)  
**a53** Event  $\sqsubseteq$  (=lbeginPoint.TimePoint)  
**a54** Event  $\sqsubseteq$  (=lendPoint.TimePoint)  
**a55** Participation  $\sqsubseteq$  (=lexclusivelyDependsOn.Object)  
**a56** Participation  $\sqsubseteq$  (=lparticipationOf.Object)

### Codomain Cardinality Constraints

**a57** Object  $\sqsubseteq$  ( $\geq$ linheresIn<sup>-</sup>.Trope)  
**a58** TimePoint  $\sqsubseteq$  ( $\geq$ lobtainsIn<sup>-</sup>.Situation)  
**a59** Event  $\sqsubseteq$  (=ltriggers<sup>-</sup>.Situation)  
**a60** AtomicEvent  $\sqsubseteq$  (=lmanifestedBy<sup>-</sup>.Disposition)

### 2.1. Event Mereology: (M1)–(M9)

Events may be composite, e.g., “the murder of Caesar” has as parts “the stabbing of Caesar by Brutus” and “Caesar’s death.” Consider a strict partial order (that is, irreflexive (M3), asymmetric (M4) and transitive (M5)) relation `hasPart` between events. An event is atomic iff it has no parts (M1), and complex otherwise (M2). (M7) determines when two complex events *mereologically* overlap. UFO-B uses the same predicate ‘overlaps’ for mereological overlapping and temporal overlapping. Here, we call them *mereologicallyOverlaps* and *temporallyOverlaps*, respectively. UFO-B also commits to *weak supplementation* [6, p. 39, (P.4)], a whole mereologically differs from its proper parts, i.e., a whole must have at least two non-overlapping parts (M6); and *strong supplementation* [6, p. 39, (P.5)] (M8), which renders a theory in which *two* entities cannot have all proper parts in common, as shown by the theorem of extensionality [6, p. 40, (3.15)] (our (M9)).

**M1**  $\forall e:\text{Event}(\text{AtomicEvent}(e) \leftrightarrow \neg \exists e':\text{Event}(\text{hasPart}(e, e')))$   
**M2**  $\forall e:\text{Event}(\text{ComplexEvent}(e) \leftrightarrow \neg \text{AtomicEvent}(e))$   
**M3**  $\forall e:\text{ComplexEvent}(\neg \text{hasPart}(e, e))$   
**M4**  $\forall e, e':\text{ComplexEvent}(\text{hasPart}(e, e') \rightarrow \neg \text{hasPart}(e', e))$   
**M5**  $\forall e, e':\text{ComplexEvent}, e'':\text{Event}((\text{hasPart}(e, e') \wedge \text{hasPart}(e', e'')) \rightarrow \text{hasPart}(e, e''))$   
**M6**  $\forall e:\text{ComplexEvent}, e':\text{Event}(\text{hasPart}(e, e') \rightarrow \exists e'':\text{Event}(\text{hasPart}(e, e'') \wedge \neg \text{mereologicallyOverlaps}(e', e'')))$   
**M7**  $\forall e, e':\text{ComplexEvent}(\text{Overlaps}(e, e') \leftrightarrow (\text{hasPart}(e, e') \vee \text{hasPart}(e', e) \vee \exists e'':\text{Event}(\text{hasPart}(e, e'') \wedge \text{hasPart}(e', e''))))$   
**M8**  $\forall e, e':\text{ComplexEvent}((\forall e'':\text{Event}(\text{hasPart}(e, e'') \rightarrow \text{hasPart}(e', e''))) \rightarrow (e = e' \vee \text{hasPart}(e', e)))$   
**M9**  $\forall e, e':\text{ComplexEvent}(e = e' \leftrightarrow \forall e'':\text{Event}(\text{hasPart}(e, e'') \leftrightarrow \text{hasPart}(e', e'')))$

[3, p. 67] recognizes the inexpressibility of mereological notions in *SRIOQ*, as its syntactic constraints forbid roles to be asymmetric and transitive. For (M1), we capture the necessary condition (indicated by ‘-’) of `AtomicEvent` in (a61), and the sufficient condition (indicated by ‘+’) in (a62). An absence of ‘-’ or ‘+’ means equivalence. (M2) is enforced by (a45) and (a46). (M3), (M4) and (M5) are captured by (a63), (a64) and (a65), respectively. (a65) makes `hasPart` *non-simple*,<sup>3</sup> which forbids its use in (a63) and (a64).

<sup>3</sup>Let  $\sqsubseteq^+$  be the transitive closure of  $\sqsubseteq$ . A role  $R$  is *non-simple* iff  $\text{Tra}(R)$  or  $\text{Sym}(R)$ ; or there is a role composition  $\tau$  s.t.  $\tau \sqsubseteq^+ R$ ; or  $R^-$  is non-simple. A role is *simple* iff it is not non-simple. The ‘simplicity’ syntactic rule forbids non-simple roles to be used in (i) concepts of the form ‘ $\exists R.\text{Self}$ ’ and ‘ $\overset{\leq}{\exists} nR.C$ ’, and (ii) role assertions of the form ‘ $\text{Irr}(R)$ ’, ‘ $\text{Asy}(R)$ ’ and ‘ $\text{Dis}(R, S)$ ’. See [3, p. 59].

For (M7), (a66)–(a68) represent the sufficient conditions for `mereologicallyOverlaps`, where (a68) makes `mereologicallyOverlaps` non-simple. The necessary condition for this role seems inexpressible, as the right-side of a role inclusion axiom (RIA) [3, p. 58] can only have a role name. The non-simplicity of `hasPart` by (a65) makes `mereologicallyOverlaps` non-simple by (a66) or (a67). (M6) seems inexpressible even by assuming a definition of `mereologicallyOverlaps`, e.g., “`Ref(hasAtLeastTwoDisjointParts)`” with the ill-formed formula “`hasAtLeastTwoDisjointParts ≡ hasPart ◦ notMereologicallyOverlap ◦ hasPart-`” would enforce the existence of a path from a whole, then to a part, then to a non-overlapping entity that is also a part, so that the path comes to a cycle by an inverse of the parthood relation. However, `notMereologicallyOverlap` is inexpressible, e.g., even by removing the axioms that make `mereologicallyOverlaps` non-simple and asserting “`Dis(mereologicallyOverlaps, notMereologicallyOverlap)`,” *SR<sub>OLQ</sub>* could not express that one relation would be the set-complement of the other, i.e., the ill-formed formula “`U ⊆ mereologicallyOverlaps ⊔ notMereologicallyOverlap`.” Finally, it seems that (M8) and (M9) are inexpressible, since there is no way to express an equality on “generic individuals.”

<b>a61</b>	<code>AtomicEvent ⊆ Event ⊓ ¬∃hasPart.Event</code>	(M1)–
<b>a62</b>	<code>Event ⊓ ¬∃hasPart.Event ⊆ AtomicEvent</code>	(M1)+
<b>a63</b>	<code>ComplexEvent ⊆ ¬(∃hasPart.Self)</code>	(M3)
<b>a64</b>	<code>Dis(hasPart, hasPart<sup>-</sup>)</code>	(M4)
<b>a65</b>	<code>Tra(hasPart)</code>	(M5)
<b>a66</b>	<code>hasPart ⊆ mereologicallyOverlaps</code>	(M7)+
<b>a67</b>	<code>hasPart<sup>-</sup> ⊆ mereologicallyOverlaps</code>	(M7)+
<b>a68</b>	<code>hasPart ◦ hasPart<sup>-</sup> ⊆ mereologicallyOverlaps</code>	(M7)+

## 2.2. Events as Manifestations of Object Dispositions: (D1)–(D4)

Atomic events are *manifestations of* (the inverse of `manifestedBy`) unique object dispositions (D2). Dispositions `inheresIn` a unique object (D1). Whenever a disposition `inheresIn` an object and is `manifestedBy` an event, the event `dependsOn` the object (D4). A situation `triggers` an atomic event iff there is a disposition that is *activated by* (the inverse of `activates`) the situation and that is `manifestedBy` the event (D3).

<b>D1</b>	$\forall d:\text{Disposition}(\exists!o:\text{Object}(\text{inheresIn}(d,o)))$
<b>D2</b>	$\forall e:\text{AtomicEvent}(\exists!d:\text{Disposition}(\text{manifestedBy}(d,e)))$
<b>D3</b>	$\forall s:\text{Situation}, e:\text{AtomicEvent}(\text{triggers}(s,e) \leftrightarrow \exists d:\text{Disposition}(\text{activates}(s,d) \wedge \text{manifestedBy}(d,e)))$
<b>D4</b>	$\forall d:\text{Disposition}, e:\text{AtomicEvent}, o:\text{Object}((\text{manifestedBy}(d,e) \wedge \text{inheresIn}(d,o)) \rightarrow \text{dependsOn}(e,o))$

(D1) is guaranteed by (a6) and (a47), while (D2) is guaranteed by (a60). As the right-side of a RIA can only have a role name, it seems that only the sufficient condition of (D3) can be formalized as (a69). (D4) is captured by (a70). (a69) and (a70) make `triggers` and `dependsOn` non-simple, respectively.

<b>a69</b>	<code>activates ◦ manifestedBy ⊆ triggers</code>	(D3)+
<b>a70</b>	<code>manifestedBy<sup>-</sup> ◦ inheresIn ⊆ dependsOn</code>	(D4)

## 2.3. On the Participation of Objects in Events: (P1)–(P5)

Objects participate not only in the atomic events that are manifestations of their dispositions, but also in the complex events that have such manifestations as parts; e.g., Caesar

participates in both “Caesar’s death” and “the murder of Caesar.” As an atomic event is a manifestation of a unique disposition (D2), which inheres in a unique object (D1), by (D4) it follows that atomic events dependsOn unique objects (P1). exclusivelyDependsOn generalizes dependsOn to complex events (P2). By composing the relations hasPart and exclusivelyDependsOn, complex events exclusivelyDependsOn the objects that its parts exclusivelyDependsOn (P3). A Participation is an event that exclusivelyDependsOn a unique object (P4). A participationOf relation from participations to objects can be defined by means of exclusivelyDependsOn (P5).

- P1**  $\forall e:\text{AtomicEvent} \exists !o:\text{Object}(\text{dependsOn}(e,o))$   
**P2**  $\forall e:\text{AtomicEvent}, o:\text{Object}(\text{exclusivelyDependsOn}(e,o) \leftrightarrow \text{dependsOn}(e,o))$   
**P3**  $\forall e:\text{ComplexEvent}, o:\text{Object}(\text{exclusivelyDependsOn}(e,o) \leftrightarrow$   
 $\forall e':\text{Event}(\text{hasPart}(e,e') \rightarrow \text{exclusivelyDependsOn}(e,o)))$   
**P4**  $\forall e:\text{Event}(\text{Participation}(e) \leftrightarrow \exists !o:\text{Object}(\text{exclusivelyDependsOn}(e,o)))$   
**P5**  $\forall o:\text{Object}, p:\text{Participation}(\text{participationOf}(p,o) \leftrightarrow$   
 $\text{exclusivelyDependsOn}(p,o))$

(P1) is guaranteed by (a49). For (P2), it seems that *SRIOQ* cannot express the subset of the exclusivelyDependsOn relation that has AtomicEvent as its domain. As the domain of dependsOn is AtomicEvent, it seems that only the necessary condition for dependsOn (a71) is expressible, but not the opposite role inclusion. Similarly for (P3) and the subset of the exclusivelyDependsOn relation that has ComplexEvent as its domain. On (P4), together, (a10) and (a55) entail the necessary condition for being a participation, while (a72) entails the sufficient condition. On (P5), since both relations participationOf and exclusivelyDependsOn have the same domain (Participation, see (a24), (a25)) and the same codomain (Object, see (a38), (a39)), the left-to-right implication can be modeled as “participationOf  $\sqsubseteq$  exclusivelyDependsOn,” while the right-to-left implication by “exclusivelyDependsOn  $\sqsubseteq$  participationOf,” i.e., the relations are equivalent. Due to the *regularity constraint*,<sup>4</sup> such role equivalence is inexpressible in *SRIOQ*. However, one can remove one of the relations from the signature of the theory with no loss of expressivity (see [3, Footnote 2]).

- a71**  $\text{dependsOn} \sqsubseteq \text{exclusivelyDependsOn}$  (P2)-  
**a72**  $\text{Event} \sqcap (= \text{exclusivelyDependsOn}.\text{Object}) \sqsubseteq \text{Participation}$  (P4)+

#### 2.4. Temporal Relations Between Events: (T1)–(T14')

[1, T7–T13] formalizes the temporal Allen relations [7]. Also, the set of time points is totally ordered by the relation precedes [1, T1–T4], and the temporal extent of an event (improperly) includes the temporal extent of all the (proper) parts of the event [1, T14]. [1, T5–T6] constrain the functions **begin-point** and **end-point** of an event. As FOL functions are total, they can be applied to TimePoints themselves. We address this issue by turning beginPoint and endPoint into functional relations (T5'), revisiting [1, T5–T14].

- T1**  $\forall t:\text{TimePoint}(\neg \text{precedes}(t,t))$   
**T2**  $\forall t,t':\text{TimePoint}(\text{precedes}(t,t') \rightarrow \neg \text{precedes}(t',t))$   
**T3**  $\forall t,t',t'':\text{TimePoint}((\text{precedes}(t,t') \wedge \text{precedes}(t',t'')) \rightarrow \text{precedes}(t,t''))$   
**T4**  $\forall t,t':\text{TimePoint}((t \neq t') \rightarrow (\text{precedes}(t,t') \vee \text{precedes}(t',t)))$

<sup>4</sup>A strict partial order  $\prec$  on the set of roles is called *regular* iff for any role name  $S$  and role  $R$ , then  $S \prec R \leftrightarrow S^- \prec R$ . A RIA ‘ $w \sqsubseteq R$ ’ is  $\prec$ -regular iff  $R$  is a role name, and (i)  $w = R \circ R$ ; or (ii)  $w = R^-$ ; or (iii)  $w = S_1 \circ \dots \circ S_n$  and  $S_i \prec R$ , for all  $1 \leq i \leq n$ ; or (iv)  $w = R \circ S_1 \circ \dots \circ S_n$  and  $S_i \prec R$ , for all  $1 \leq i \leq n$ ; or (v)  $w = S_1 \circ \dots \circ S_n \circ R$  and  $S_i \prec R$ , for all  $1 \leq i \leq n$ . A set of RIAs is regular iff there exists a regular order  $\prec$  s.t. each RIA in the set is  $\prec$ -regular. The ‘regularity’ syntactic rule ensures that the set of all RIAs in a TBox is regular. See [3, p. 58].

- T5'**  $\forall e:\text{Event} \exists t, t': \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{endPoint}(e, t'))$   
**T6'**  $\forall e:\text{Event}, t, t': \text{TimePoint}((\text{beginPoint}(e, t) \wedge \text{endPoint}(e, t')) \rightarrow \text{precedes}(t, t'))$   
**T7'**  $\forall e, e': \text{Event}(\text{before}(e, e') \leftrightarrow \exists t, t': \text{TimePoint}(\text{endPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{precedes}(t, t'))$   
**T8'**  $\forall e, e': \text{Event}(\text{meets}(e, e') \leftrightarrow \exists t: \text{TimePoint}(\text{endPoint}(e, t) \wedge \text{beginPoint}(e', t)))$   
**T9'**  $\forall e, e': \text{Event}(\text{temporallyOverlaps}(e, e') \leftrightarrow \exists t_0, \dots, t_3: \text{TimePoint}(\text{beginPoint}(e, t_0) \wedge \text{beginPoint}(e', t_1) \wedge \text{endPoint}(e, t_2) \wedge \text{endPoint}(e', t_3) \wedge \text{precedes}(t_0, t_1) \wedge \text{precedes}(t_1, t_2) \wedge \text{precedes}(t_2, t_3)))$   
**T10'**  $\forall e, e': \text{Event}(\text{starts}(e, e') \leftrightarrow \exists t, t', t'': \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{endPoint}(e, t') \wedge \text{endPoint}(e', t'') \wedge \text{precedes}(t', t''))$   
**T11'**  $\forall e, e': \text{Event}(\text{during}(e, e') \leftrightarrow \exists t_0, \dots, t_3: \text{TimePoint}(\text{beginPoint}(e, t_0) \wedge \text{beginPoint}(e', t_1) \wedge \text{endPoint}(e, t_2) \wedge \text{endPoint}(e', t_3) \wedge \text{precedes}(t_1, t_0) \wedge \text{precedes}(t_2, t_3)))$   
**T12'**  $\forall e, e': \text{Event}(\text{finishes}(e, e') \leftrightarrow \exists t, t', t'': \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{endPoint}(e, t'') \wedge \text{endPoint}(e', t'') \wedge \text{precedes}(t', t''))$   
**T13'**  $\forall e, e': \text{Event}(\text{equals}(e, e') \leftrightarrow \exists t, t': \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t) \wedge \text{endPoint}(e, t') \wedge \text{endPoint}(e', t'))$   
**T14'**  $\forall e, e': \text{Event}(\text{hasPart}(e, e') \rightarrow ((\exists t: \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t)) \vee \exists t, t': \text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{precedes}(t, t')))) \wedge (\exists t: \text{TimePoint}(\text{endPoint}(e, t) \wedge \text{endPoint}(e', t)) \vee \exists t, t': \text{TimePoint}(\text{endPoint}(e, t) \wedge \text{endPoint}(e', t') \wedge \text{precedes}(t', t))))$

[8] discusses the impossibility of expressing the Allen's time interval relations [7] in *SRQLQ*. The best we can do is to provide partial axiomatizations of UFO-B's temporal relations between events. [1, p. 331] informally defines the (co)domain of these relations as Event (a73)–(a86). On precedes, (T1), (T2) and (T3) are captured by (a87), (a88) and (a89), respectively. (a89) makes precedes non-simple, which forbids its use in (a87) and (a88). Totality (T4) seems inexpressible (we can only think of a solution using variables). (T5') is guaranteed by (a53) and (a54). (T6') is guaranteed by (a90), which makes precedes non-simple. As the right-side of a RIA can only have a role name, the necessary conditions of the Allen relations (T7')–(T13') seem inexpressible. The sufficient conditions for before (T7') and meets (T8') are captured by (a91) and (a92), respectively, and which make them non-simple. The sufficient conditions for temporallyOverlaps (T9'), starts (T10'), during (T11'), finishes (T12'), and equals (T13'), seem inexpressible, since they would require a disjunction of role compositions, see (f1)–(f5), respectively. Finally, (T14') would break even more syntactic constraints, see (f6).

- |   |   |   |
|---|---|---|
| <b>a73</b> $\exists \text{before.} \top \sqsubseteq \text{Event}$   | <b>a77</b> $\exists \text{starts.} \top \sqsubseteq \text{Event}$                       | <b>a81</b> $\exists \text{finishes.} \top \sqsubseteq \text{Event}$ |
| <b>a74</b> $\top \sqsubseteq \forall \text{before.} \text{Event}$   | <b>a78</b> $\top \sqsubseteq \forall \text{starts.} \text{Event}$                       | <b>a82</b> $\top \sqsubseteq \forall \text{finishes.} \text{Event}$ |
| <b>a75</b> $\exists \text{meets.} \top \sqsubseteq \text{Event}$  | <b>a79</b> $\exists \text{during.} \top \sqsubseteq \text{Event}$                       | <b>a83</b> $\exists \text{equals.} \top \sqsubseteq \text{Event}$   |
| <b>a76</b> $\top \sqsubseteq \forall \text{meets.} \text{Event}$  | <b>a80</b> $\top \sqsubseteq \forall \text{during.} \text{Event}$                       | <b>a84</b> $\top \sqsubseteq \forall \text{equals.} \text{Event}$   |
| <b>a85</b> $\exists \text{temporallyOverlaps.} \top \sqsubseteq \text{Event}$   | <b>a87</b> $\text{TimePoint} \sqsubseteq \neg(\exists \text{precedes.} \text{Self})$    | (T1)  |
| <b>a86</b> $\top \sqsubseteq \forall \text{temporallyOverlaps.} \text{Event}$   | <b>a88</b> $\text{Dis}(\text{precedes}, \text{precedes}^-)$                             | (T2)  |
| <b>a89</b> $\text{Tra}(\text{precedes})$  | (T3) <b>a90</b> $\text{beginPoint}^- \circ \text{endPoint} \sqsubseteq \text{precedes}$ | (T6')   |
| <b>a91</b> $\text{endPoint} \circ \text{precedes} \circ \text{beginPoint}^- \sqsubseteq \text{before}$  |   | (T7')+  |
| <b>a92</b> $\text{endPoint} \circ \text{beginPoint}^- \sqsubseteq \text{meets}$   |   | (T8')+  |
| <b>f1</b> $(\text{beginPoint} \circ \text{precedes} \circ \text{beginPoint}^-) \sqcap (\text{endPoint} \circ \text{precedes}^- \circ \text{beginPoint}^-) \sqcap (\text{endPoint} \circ \text{precedes} \circ \text{endPoint}^-) \sqsubseteq \text{temporallyOverlaps}$ |   | (T9')+  |
| <b>f2</b> $(\text{beginPoint} \circ \text{beginPoint}^-) \sqcap (\text{endPoint} \circ \text{precedes} \circ \text{endPoint}^-) \sqsubseteq \text{starts}$  |   | (T10')+   |
| <b>f3</b> $(\text{beginPoint} \circ \text{precedes}^- \circ \text{beginPoint}^-) \sqcap (\text{endPoint} \circ \text{precedes} \circ \text{endPoint}^-) \sqsubseteq \text{during}$  |   | (T11')+   |

- f4**  $(\text{endPoint} \circ \text{endPoint}^-) \sqcap (\text{beginPoint} \circ \text{precedes}^- \circ \text{beginPoint}^-) \sqsubseteq \text{finishes}$   
(T12')<sup>+</sup>  
**f5**  $(\text{beginPoint} \circ \text{beginPoint}^-) \sqcap (\text{endPoint} \circ \text{endPoint}^-) \sqsubseteq \text{equals}$  (T13')<sup>+</sup>  
**f6**  $\text{hasPart} \sqsubseteq (((\text{beginPoint} \circ \text{beginPoint}^-) \sqcup (\text{beginPoint} \circ \text{precedes} \circ \text{beginPoint}^-))$   
 $\sqcap ((\text{endPoint} \circ \text{endPoint}^-) \sqcup (\text{endPoint} \circ \text{precedes}^- \circ \text{endPoint}^-)))$  (T14')

## 2.5. World Changes and Situations: (S1')–(S7)

Any event is *triggered by/bringsAbout* a unique situation (S3)/(S4). If a situation *triggers* an event, the situation *obtainsIn* the same *TimePoint* that is the *beginPoint* of the event (S1'). If an event *bringsAbout* a situation, the situation *obtainsIn* the same *TimePoint* that is the *endPoint* of the event (S2'). Facts are situations that *obtain* (S5). An event  $e$  *directlyCauses* an event  $e'$  “by means of” a situation that was *brought about by* (the inverse of *bringsAbout*)  $e$  and that *triggers*  $e'$  (S6). On (S7), *causes* should be the transitive closure of *directlyCauses*, what is inexpressible in FOL.

- S1'**  $\forall s:\text{Situation}, e:\text{Event}(\text{triggers}(s, e) \rightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t) \wedge \text{beginPoint}(e, t)))$   
**S2'**  $\forall s:\text{Situation}, e:\text{Event}(\text{bringsAbout}(e, s) \rightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t) \wedge \text{endPoint}(e, t)))$   
**S3**  $\forall e:\text{Event}(\exists! s:\text{Situation}(\text{triggers}(s, e)))$   
**S4**  $\forall e:\text{Event}(\exists! s:\text{Situation}(\text{bringsAbout}(e, s)))$   
**S5**  $\forall s:\text{Situation}(\text{Fact}(s) \leftrightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t)))$   
**S6**  $\forall e, e':\text{Event}(\text{directlyCauses}(e, e') \leftrightarrow \exists s:\text{Situation}(\text{bringsAbout}(e, s) \wedge \text{triggers}(s, e')))$   
**S7**  $\forall e, e':\text{Event}(\text{causes}(e, e') \leftrightarrow (\text{directlyCauses}(e, e') \vee \exists e'':\text{Event}(\text{causes}(e, e'') \wedge \text{causes}(e'', e'))))$

(S1') and (S2') seem inexpressible in *SRIOIQ*, since the right-side of a RIA can only have a role name (see (f7), (f8)). (S3)/(S4) are guaranteed by (a59)/(a52), respectively. On (S5), the necessary condition for being a fact is captured by (a51), while the sufficient condition by (a93). On (S6), it is impossible to define a role as the composition of other roles. So, we capture the sufficient condition for *directlyCauses* in (a94) (making *directlyCauses* non-simple), but the necessary condition seems inexpressible. On (S7), only the sufficient conditions for *causes* seem expressible, as in (a11) and (a95) (making *causes* non-simple); similarly to (S1'), the necessary condition seem inexpressible.

- f7**  $\text{triggers} \sqsubseteq \text{obtainsIn} \circ \text{beginPoint}^-$  **f8**  $\text{bringsAbout} \sqsubseteq \text{beginPoint} \circ \text{obtainsIn}^-$   
**a93**  $\text{Situation} \sqcap (\geq \text{lobtainsIn}.\text{TimePoint}) \sqsubseteq \text{Fact}$  (S5)<sup>+</sup>  
**a94**  $\text{bringsAbout} \circ \text{triggers} \sqsubseteq \text{directlyCauses}$  (S6)<sup>+</sup>  
**a95**  $\text{causes} \circ \text{causes} \sqsubseteq \text{causes}$  (S7)<sup>+</sup>

## 3. Well-formed *SRIOIQ* Theories Formalizing UFO-B

Figure 2 shows a regular order  $\prec$  on the set of roles and s.t. each RIA in  $\mathcal{T} = \{(a1)–(a95)\}$  is  $\prec$ -regular, what shows that  $\mathcal{T}$  is regular. However,  $\mathcal{T}$  violates *simplicity*. So, we enumerate subsets of  $\mathcal{T}$  that are maximal w.r.t. *simplicity*, i.e., theories such that the inclusion of any other axiom makes them violate *simplicity*. In order to track the incompatibilities between axioms that lead to infringing *simplicity*, we build a simple meta-level theory in propositional logic. Assume the ‘operators’  $\mathbb{I}$  and  $\mathbb{S}$  as “proposition builders” in the sense that a proposition ‘ $\mathbb{I}(i)$ ’ means that the axiom (ai) is included in the TBox, and



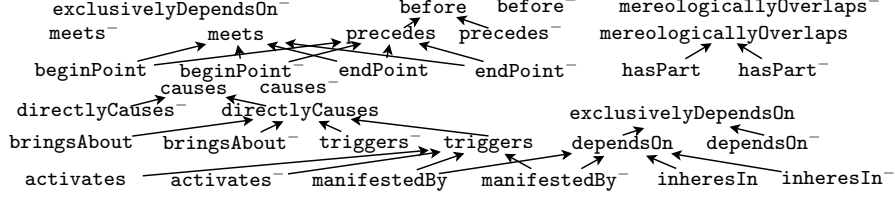


Figure 2. A strict partial order of roles that satisfy regularity for  $\mathcal{T} = \{(a1)-(a95)\}$ .

a proposition ‘ $\bar{\mathbb{S}}(r)$ ’ means that the role  $r$  is non-simple. Since our set  $\mathcal{T}$  is finite (95 axioms) and the set of role names is finite (16 role names) then the set of proposition symbols built by means of  $\mathbb{I}$  and  $\bar{\mathbb{S}}$  is also finite (111). By means of this meta-level theory, we define the problem of finding subsets of  $\mathcal{T}$  that satisfy simplicity as a SAT problem.

Firstly, we capture the reasons for non-simplicity: (1) Subsuming a role composition:  $\mathbb{I}(68) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$ ;  $\mathbb{I}(69) \rightarrow \bar{\mathbb{S}}(\text{triggers})$ ;  $\mathbb{I}(70) \rightarrow \bar{\mathbb{S}}(\text{dependsOn})$ ;  $\mathbb{I}(90) \rightarrow \bar{\mathbb{S}}(\text{precedes})$ ;  $\mathbb{I}(92) \rightarrow \bar{\mathbb{S}}(\text{causes})$ . (2) ‘Tra’:  $\mathbb{I}(65) \rightarrow \bar{\mathbb{S}}(\text{hasPart})$ ;  $\mathbb{I}(89) \rightarrow \bar{\mathbb{S}}(\text{precedes})$ . (3) Subsuming non-simple roles:  $(\mathbb{I}(65) \wedge \mathbb{I}(66)) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$ ;  $(\mathbb{I}(65) \wedge \mathbb{I}(67)) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$ ;  $(\mathbb{I}(70) \wedge \mathbb{I}(71)) \rightarrow \bar{\mathbb{S}}(\text{exclusivelyDependsOn})$ ;  $(\mathbb{I}(11) \wedge \mathbb{I}(94)) \rightarrow \bar{\mathbb{S}}(\text{causes})$ .

Secondly, the axioms forbidden by the previous non-simple roles:  $\bar{\mathbb{S}}(\text{triggers}) \rightarrow \neg \mathbb{I}(59)$ ;  $\bar{\mathbb{S}}(\text{dependsOn}) \rightarrow \neg \mathbb{I}(49)$ ;  $\bar{\mathbb{S}}(\text{hasPart}) \rightarrow (\neg \mathbb{I}(48) \wedge \neg \mathbb{I}(63) \wedge \neg \mathbb{I}(64))$ ;  $\bar{\mathbb{S}}(\text{exclusivelyDependsOn}) \rightarrow (\neg \mathbb{I}(55) \wedge \neg \mathbb{I}(72))$ ;  $\bar{\mathbb{S}}(\text{precedes}) \rightarrow (\neg \mathbb{I}(87) \wedge \neg \mathbb{I}(88))$ .

Thirdly, the meta-axioms on the incompatibilities due to the simplicity rule: (i)  $\mathbb{I}(65) \rightarrow (\neg \mathbb{I}(48) \wedge \neg \mathbb{I}(63) \wedge \neg \mathbb{I}(64))$ , (ii)  $\mathbb{I}(69) \rightarrow \neg \mathbb{I}(59)$ , (iii)  $\mathbb{I}(70) \rightarrow \neg \mathbb{I}(49)$ , (iv)  $(\mathbb{I}(70) \wedge \mathbb{I}(71)) \rightarrow (\neg \mathbb{I}(49) \wedge \neg \mathbb{I}(55) \wedge \neg \mathbb{I}(72))$ , (v)  $(\mathbb{I}(89) \vee \mathbb{I}(90)) \rightarrow (\neg \mathbb{I}(87) \wedge \neg \mathbb{I}(88))$ .

We translated (i)–(v) into 12 clauses in conjunctive normal form (CNF), more specifically, the DIMACS format, and used the SAT solver CryptoMiniSat v5.0.0 in order to find all the models. From the  $2^{15} = 32\,768$  possibilities, there are 3 969 models. However, we are only interested in the models that represent maximal TBoxes w.r.t. simplicity. These models are maximal w.r.t. truth, i.e., the change of any truth-value assignment from *false* to *true* invalidates the model. Using a script in the language R, we selected from the 3 969 models, the 24 that are maximal, leading to 24 UFO-B theories  $\mathcal{T}_i$ , which are represented by means of auxiliary sets of indexes  $\bar{\mathcal{T}}_i$  such that  $\mathcal{T}_i = \mathcal{T} \setminus \{(a_j) | j \in \bar{\mathcal{T}}_i\}$ .

$\bar{\mathcal{T}}_0 = \{48, 63, 64, 69, 70, 89, 90\}$	$\bar{\mathcal{T}}_{12} = \{48, 63, 64, 69, 70, 87, 88\}$
$\bar{\mathcal{T}}_1 = \{48, 49, 55, 59, 63, 64, 72, 89, 90\}$	$\bar{\mathcal{T}}_{13} = \{48, 49, 63, 64, 69, 71, 89, 90\}$
$\bar{\mathcal{T}}_2 = \{48, 49, 59, 63, 64, 71, 89, 90\}$	$\bar{\mathcal{T}}_{14} = \{65, 69, 70, 89, 90\}$
$\bar{\mathcal{T}}_3 = \{48, 59, 63, 64, 70, 89, 90\}$	$\bar{\mathcal{T}}_{15} = \{48, 59, 63, 64, 70, 87, 88\}$
$\bar{\mathcal{T}}_4 = \{59, 65, 70, 89, 90\}$	$\bar{\mathcal{T}}_{16} = \{49, 55, 59, 65, 72, 87, 88\}$
$\bar{\mathcal{T}}_5 = \{49, 55, 65, 69, 72, 87, 88\}$	$\bar{\mathcal{T}}_{17} = \{49, 59, 65, 71, 87, 88\}$
$\bar{\mathcal{T}}_6 = \{49, 65, 69, 71, 87, 88\}$	$\bar{\mathcal{T}}_{18} = \{49, 65, 69, 71, 89, 90\}$
$\bar{\mathcal{T}}_7 = \{49, 55, 65, 69, 72, 89, 90\}$	$\bar{\mathcal{T}}_{19} = \{49, 59, 65, 71, 89, 90\}$
$\bar{\mathcal{T}}_8 = \{48, 49, 55, 59, 63, 64, 72, 87, 88\}$	$\bar{\mathcal{T}}_{20} = \{65, 69, 70, 87, 88\}$
$\bar{\mathcal{T}}_9 = \{48, 49, 55, 63, 64, 69, 72, 87, 88\}$	$\bar{\mathcal{T}}_{21} = \{59, 65, 70, 87, 88\}$
$\bar{\mathcal{T}}_{10} = \{48, 49, 59, 63, 64, 71, 87, 88\}$	$\bar{\mathcal{T}}_{22} = \{49, 55, 59, 65, 72, 89, 90\}$
$\bar{\mathcal{T}}_{11} = \{48, 49, 63, 64, 69, 71, 87, 88\}$	$\bar{\mathcal{T}}_{23} = \{48, 49, 55, 63, 64, 69, 72, 89, 90\}$

In order to empirically evaluate the  $\mathcal{T}_i$ s, we translated each  $\mathcal{T}_i$  into a TBox $_i$  in OWL 2 DL. We assessed the consistency of each of these TBoxes by means of models generated

by the Alloy Analyzer tool for a new Alloy codification that exactly corresponds to the FOL axioms shown in Section 2. Some of these models are “fully-consistent,” i.e., they have at least one instance for each class or relation. We generated 6 models by the Alloy Analyzer: 3 models enforcing full-consistency, having a minimum scope-size of 9 and a maximum, due to memory constraints, of 28; the other 3 models do not enforce full-consistency, having a model of scope-size 1, and a maximum scope-size of 29, due again to memory constraints. Each model was transformed into an individual assertion box (ABox) and joined to each TBox<sub>i</sub> to form an OWL 2 DL knowledge base (KB). In total, this process produced 24×6=144 KBs. We used the DL reasoner Pellet version 2.4.0 to test the consistency of these KBs, all of which are consistent. Finally, the interested reader can find our DIMACS file, the SAT solutions, the R script, our Alloy codification of UFO-B, the generated Alloy models, the OWL 2 DL TBoxes, and the KBs at <https://osf.io/jas8m>.

#### 4. Conclusions

The development of suitable foundational theories is an important step towards the definition of precise semantics and sound methodological principles for modeling complex real-world phenomena. From an engineering point of view, there is a need for decidable representations of these theories that can support automated reasoning. We address the combination of ontological adequacy and decidability by providing OWL 2 DL TBoxes that (partially) represent the axiomatization of UFO-B. As a future work, we intend to (1) perform an ontological analysis of the 24 theories. (2) complement UFO-B with a fuller presentation and philosophical justification of sub-theories dealing with: (i) the differentiation of roles played by objects w.r.t. an event (the so-called processual roles); (ii) event qualities and quality structures; (iii) events of creation, destruction and modification of objects. (3) investigate alternative DL mappings for the UFO theories presented here dealing with different aspects of the trade-off between expressivity and tractability.

*Acknowledgments* The first two authors were supported by the grants 71024352 and 71047522 of the “edital FAPES/CAPES n. 009/2014.” The third author was supported by the OCEAN Project (FUB) and by the CNPq grant #312158/2015-7.

#### References

- [1] G. Guizzardi, G. Wagner, R. de Almeida Falbo, R. S. S. Guizzardi, and J. P. A. Almeida, “Towards ontological foundations for the conceptual modeling of events,” in *Conceptual Modeling - 32th International Conference, ER 2013, Hong-Kong, China, November 11-13, 2013. Proceedings* (W. Ng, V. C. Storey, and J. Trujillo, eds.), vol. 8217 of *Lecture Notes in Computer Science*, pp. 327–341, Springer, 2013.
- [2] D. Jackson, *Software Abstractions: logic, language, and analysis*. MIT press, 2012.
- [3] I. Horrocks, O. Kutz, and U. Sattler, “The even more irresistible *SRIOQ*,” *Kr*, vol. 6, pp. 57–67, 2006.
- [4] W3C, *OWL 2 Web Ontology Language Document Overview*, second edition ed., Dec. 2012.
- [5] G. Guizzardi, *Ontological foundations for structural conceptual models*. phdthesis, University of Twente, Enschede, The Netherlands, Enschede, Oct. 2005.
- [6] R. Casati and A. C. Varzi, *Parts and places: The structures of spatial representation*. MIT Press, 1999.
- [7] J. F. Allen, “Maintaining knowledge about temporal intervals,” *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.
- [8] S. Batsakis, E. Petrakis, I. Tachmazidis, and G. Antoniou, “Temporal representation and reasoning in OWL 2,” *Semantic Web*, vol. 8, pp. 981–1000, Aug. 2017.