Events as Entities in Ontology-Driven Conceptual Modeling

João Paulo A. Almeida[1], Ricardo A. Falbo[1], and Giancarlo Guizzardi[1,2]

[1] Ontology & Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo (UFES), Brazil
jpalmeida@ieee.org, falbo@inf.ufes.br

[2] Conceptual and Cognitive Modeling Research Group (CORE),
Free University of Bozen-Bolzano, Italy
giancarlo.guizzardi@unibz.it

Abstract. The Unified Foundational Ontology (UFO) has been used to provide foundations for the major conceptual modeling constructs. This ontology has led to the OntoUML Ontology-Driven Conceptual Modeling language, a UML class diagram profile reflecting the ontological micro-theories comprising UFO. So far, the focus of OntoUML has been on the representation of structural aspects of a domain (endurant types and their relations), corresponding to a fragment of UFO dubbed UFO-A. This paper extends OntoUML by addressing the representation of event types, reflecting the UFO-B foundational ontology of events. Based on the ontological distinctions and axiomatization provided by UFO-B, we define new OntoUML constructs and guidelines for the conceptual modeling of events and event relations in structural conceptual models.

Keywords: OntoUML, Events, Ontology-Driven Conceptual Modeling

1 Introduction

There has been a growing interest in the use of foundational ontologies to evaluate and (re)design conceptual modeling languages. With such use as a key motivation, a community of researchers has contributed for over a decade now to the development of the Unified Foundational Ontology (UFO), providing theoretical foundations underlying all major conceptual modeling constructs. UFO has been used to systematically design an ontology-driven conceptual modeling (ODCM) language termed OntoUML [10, 14], which has been successfully employed in academic, industrial and governmental settings to create conceptual models in a variety of domains [14].

The observation of the application of OntoUML over the years, conducted by several groups in a number of domains, amounted to a fruitful empirical source of knowledge regarding the language and its foundations. In particular, we have observed how modelers would slightly subvert the language's syntax, ultimately creating what we call *"systematic subversions"* [14]. These "subversions" would produce models that were grammatically incorrect, but which were needed to express the intended conceptualizations. Moreover, they were "systematic" because they recurred in the works of different authors in similar manners and with the same modeling intent.

Laender, A., Pernici, B., Lim, E.P., de Oliveira, J. (eds) 38th International Conference on Conceptual Modeling (ER 2019), LNCS 11788. Springer, pp. 469–483, 2019.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-33223-5_39.

One of these "language subversions" concerns the representation of events and their relations [16, 21]. Dealing with the representation of events is key to conceptual modeling and knowledge representation, given the importance of events in cognition, language and, in fact, most human endeavours. Despite their importance, the current version of OntoUML does not address event types explicitly, as it focuses on the modeling of structural aspects of the domain in accordance with the fragment of UFO that was first defined (UFO-A, an ontology of endurants).

Recently, there has been a growing interest in the explicit modeling of events in structural conceptual models. This trend can be observed in the so-called *event reification approach* in conceptual modeling [1, 2, 17] and in *behavioral modeling* in Object-Oriented structural models (class diagrams) [5]. For example, Olivé [17] writes: *"When events are entities, they are modeled in a way similar to ordinary entities: they are instance of event types (a special kind of entity type), they may participate in relationships, they can be specialized or generalized, and so on..."*. Moreover, empirical studies show the benefits of explicitly representing events in structural models. For example, Allen and March [2] show the benefits of explicit event representation in terms of faster learning about the semantics of queries over a conceptual model, and in terms of better supporting casual users in accurately recognizing when these queries are correct. They argue that effective analysis and design require *"a more substantive ontological definition of an event as an entity having both identity and properties."*

Following the same ontology-based language engineering approach that was used to create the original version of OntoUML [10], we employ here a well-founded theory of events (UFO-B) to advance an extension of OntoUML to address events and their relations. We introduce specialized stereotypes to capture event types, their (mereological and historical) relations, as well as their relations to the existing endurant types in OntoUML. Syntactic constraints in the profile guide the creation of sound models capturing event types and adhering to the rules of the underlying foundational ontology.

The remainder of this paper is organized as follows: Section 2 presents some background on OntoUML and the UFO-B ontology of events. The notions in UFO-B are used in Section 3 to introduce constructs and syntactic constraints for event modeling in OntoUML. In Section 4, to demonstrate the expressivity of the profile, we employ it in the representation of a reference ontology of software testing [20]. Section 5 positions our contribution with respect to related work and Section 6 presents some conclusions.

2 Background

OntoUML is a language whose meta-model has been designed to comply with the ontological distinctions and axiomatization put forth by UFO [10]. The ontological distinctions present in the ontology are reflected in the modeling primitives of the language via stereotypes, providing thus precise semantics grounded in the underlying ontology. In addition to that, the metamodel of the language is enriched with a number of *semantically motivated syntactic formal constraints* [7] that reflect the axiomatization of the underlying ontology. This combination of stereotypes and constraints enforces conformance, making every valid OntoUML model compliant to UFO.

The original version of OntoUML reflects a particular layer of UFO termed UFO-A, which is the *Ontology of Endurants* in UFO. Endurants are entities that exist in time and can change in a qualitative way while maintaining their identity. They are endowed with both essential and accidental properties and, for this reason, they can instantiate certain types in a *necessary* manner (static classification) while instantiating other types in a *contingent* manner (dynamic classification). Substantials (e.g., Mick Jagger, his car, the moon), relators (e.g., Bill and Ana's marriage, Mary's enrollment in Yale) and qualities (e.g., John's knowledge of Greek, Paul's Fever) are examples of endurants. Roughly speaking, they are the ontological counterparts of Objects, Reified Relationships and Weak Entities in the literature of Conceptual Modeling [9, 10].

Figure 1 exemplifies an OntoUML model. In this model, we have two *kinds* of substantials: `Organization` and `Person`. Kinds are types that classify their entities necessarily (in a modal sense) and that provide a uniform principle of identity for their instances. Instances of a kind can (contingently) instantiate different *roles* in different relational contexts. For example, a person can move in the extension of the role `Employee` by participating in `Employment` *relators*. This distinction between necessary and contingent types applies to all endurants and not only to substantials. For example, while an `Employment` is necessarily so, it can contingently be classified as a `TemporaryEmployment` and as a `PermanentEmployment`, i.e., the same instance of `Employment` (e.g., the one connecting John Smith and the UN) can move from the extension of the former to the extension of the latter. Relators (as well as qualities) are existentially dependent entities. In this example, the Employment of John Smith in the UN can only exist if both John Smith and the UN exist. This particular relation of multiple existential dependency is termed in OntoUML *mediation* [10].

Beyond the Ontology of Endurants, UFO also comprises an Ontology of Events (UFO-B). It has been formalized and model checked in [13] and, in [4], systematically mapped to the description logics SHROIQ. It has been extensively tested in practice and employed as a reference model for addressing problems from enterprise architecture [16], software engineering [20], as well as complex media management and event modeling in petroleum exploration [13]. UFO-B is composed of five sub-theories: (i) mereology – events can form partonomies. In this sub-theory, the relations between events and their parts are characterized by the axioms of the so-called *extensional mereology*; (ii) participation – events are existentially dependent on endurants. The maximal part of an event that is exclusively dependent on a particular endurant is called a *participation* (of that endurant in that event). Events can be partitioned into a set of exhaustive and mutually disjoint participations in this sense. For example, in Brutus' stabbing of

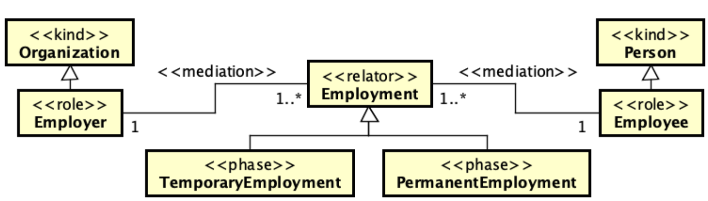

Fig. 1. OntoUML model capturing endurant types

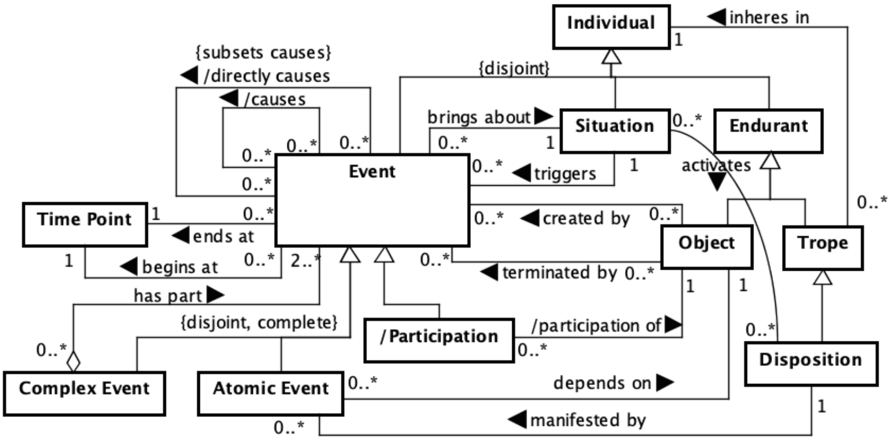

Fig. 2. A summary of UFO-B and its relations

Caesar, we have the participations of Brutus, of Caesar, and of the dagger; (iii) temporal relations – events occur in time accumulating temporal parts. Their primary properties are temporal properties. In particular, their *begin* and *end points*. Based on these properties, this theory defines all of the so-called *Allen Relations* [3] between events; (iv) events as manifestation of dispositions – this sub-theory connects endurants and events by characterizing how events are *manifestations* of particular endurants called *dispositions*, which can themselves inhere in other endurants. For example, the passing of an electrical current in a conductor is an event that is a manifestation of a disposition (electrical conductivity) inhering in a conductor. Dispositions are said to be *triggered by* certain *situations*; (v) change – events map the world from situations (that *activate* the dispositions of which they are manifestations) to situations (which are *brought about* the occurrence of that event). If an event E brings about a situation S that activates the dispositions that are manifested as event E', then we say that: S *triggers* E' and that E *causes* E'. Figure 2 summarizes these aspects of UFO-B. For a complete presentation and full formalization of this ontology, one should refer to [4, 13].

3 Extending OntoUML with Event Types

This section presents the extension of OntoUML we have defined to support the modeling of event types and their relations according to UFO-B. First of all, we introduce the stereotype ≪*event*≫ to identify those classes whose instances are events (i.e., to identify the event types in a model). We then introduce stereotypes for UML associations to provide rules for relating various event types, supporting thereby the modeling of mereological relations between events and historical dependence relations. We also introduce stereotypes to model the participation of endurants in events (≪*participation*≫, ≪*creation*≫, ≪*termination*≫), effectively connecting the extension for events we introduce here with the endurant types that had been defined previously for OntoUML (≪*kind*≫, ≪*subkind*≫, ≪*phase*≫, ≪*role*≫, ≪*relator*≫, ≪*quality*≫, etc.).

3.1 Introducing Event Types with the ≪*event*≫ Stereotype

The stereotype ≪*event*≫ identifies those classes whose instances are events (past occurrences). These classes are disjoint from any classes that model endurant types, and whose instances are endurants. As a consequence, no class may be stereotyped with both ≪*event*≫ and any of the other stereotypes defined for endurant types in OntoUML. For the same reason, no class may specialize simultaneously a class stereotyped ≪*event*≫ and a class stereotyped with any of the other OntoUML stereotypes representing endurant types.

Classes stereotyped ≪*event*≫ may be given special attributes to reflect an event's temporal properties in a suitable temporal quality structure. We consider two options here: (i) the use of two attributes, one stereotyped ≪*begin*≫ and the other stereotyped ≪*end*≫ to identify the boundaries of events; or (ii) the use of a single attribute stereotyped with both stereotypes (≪*begin*≫ and ≪*end*≫) in case begin and end coincides systematically for that class. The choice of temporal value space (and thus corresponding datatypes) is application-dependent, and typically reflects a particular model's granularity requirements. In models in which these alternatives can be applied uniformly to all event classes, the temporal attributes can be included in an abstract `Event` superclass (stereotyped ≪*event*≫). (This solution is similar to what is proposed by [6] with a single ≪*timestamp*≫ stereotype.)

The use of these temporal attributes forms the basis of the support for the well-known time interval relations proposed by Allen [3]. All these relations can be derived directly from the temporal attributes [13]. Helper OCL operations reflecting each of the temporal relations are defined in the profile (`before`, `meets`, `overlaps`, `starts`, `during`, `finishes`, `after`, `metBy`, `overlappedBy`, `startedBy`, `contains`, `finishedBy` and `equals`) and thus can be used in constraints involving event classes.

3.2 Relations between Event Types and Endurant Types

Participation. In order to model the participation of endurants in events, we use the stereotype ≪*participation*≫. An association stereotyped ≪*participation*≫ always relates a class stereotyped with ≪*event*≫ with a class denoting an endurant type. If an endurant and an event are linked through a ≪*participation*≫ association, then, either: (i) the event is a manifestation of a disposition of the participating endurant, or (ii) the event is composed of such a manifestation. The lefthand side of Figure 3 illustrates the use of the stereotypes showing the participation of persons into acts of composition. In this particular model, an act of composition is always dependent on a single person (the composer). Moreover, a person may participate in one or more acts of composition.

Creation and Termination of an Endurant. A special kind of participation is the creation of an endurant. This is identified with the ≪*creation*≫ stereotype. If an endurant is related to an event through an association stereotyped ≪*creation*≫ then that endurant is created in that event. In the example shown in Figure 3, a `Musical Piece` is created in an `Act of Composition` (or in an event that is part of it).

Fig. 3. Creation of a Musical Piece in an Act of Composition

Note that we are concerned here only with the modeling of events as past (as opposed to ongoing) occurrences. Using Lyons' distinction between experiential and historical modes of description ([15] apud [8]), we adopt the historical mode, i.e., we are concerned with "the fixed history of events as *faits accomplis*, as it were the fossil record of once-active processes." [8]. Because of this, instances of classes stereotyped with ≪*event*≫ are classified by those classes necessarily ("rigidly" or "statically"). For the same reason, any features of events are immutable (and should be marked `readOnly`), including the association ends attached to endurants in participation and creation associations (An `Act of Composition` will never have a different `Person` as composer or produce a different `Musical Piece` than it has produced. A `Person`, on the other hand, may participate in new acts of composition in time—but always accumulating past acts of composition if any).

The introduction of (past) events in a model has an important consequence to the interpretation of the endurant types in a model: since events are immutably tied to the endurants on which they specifically depend, and since events accumulate over time, related endurants must also accumulate over time. In other words, by introducing events in the model, our universe of discourse contains not only the entities that exist in a given circumstance but also all entities that have existed in that history of our universe of discourse up to that point (a view aligned with the so-called *Growing Block Universe theory* [19]), shifting from a purely "current semantics" to a "historical semantics".

A modeling consequence of "historical semantics" is that "termination" of an endurant should be considered a change in phase rather than "removal" from the universe of discourse. The termination of an endurant in the profile is represented with the introduction of the ≪*termination*≫ stereotype which relates an event type to a class stereotyped ≪*phase*≫ which is instantiated by the endurant when it takes on a "historical" nature. In such a phase, endurants have immutable properties not unlike past events. Figure 4 shows an example concerning the creation and termination of pieces of legislation by congress. Pieces of legislation may be terminated by means of a `Legislation Repeal`. When terminated, a `Piece of Legislation` instantiates the `Repealed Piece of Legislation` ≪*phase*≫ permanently.

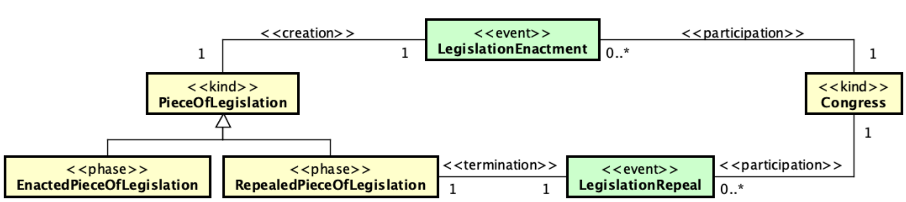

Fig. 4. Creation and Termination of a Piece of Legislation

Roles of Participants. The role an endurant instantiates in virtue of having participated in an event of a particular type can be modeled explicitly with the stereotype ≪*historicalRole*≫. Figure 5 revisits the example in Figure 3 introducing the `Composer` historical role. A historical role is required to be related to an event type through a ≪*participation*≫ association. In this case, the minimum cardinality of the association end attached to the event type must be one, reflecting that, for an endurant to play the role, it mandatorily has participated in an event of that type. In this model, any `Composer` has a participation in at least one `Act of Composition` (which does not apply to `Person` in general as shown in Figure 3). Further, according to the model, only one `Composer` participates in a certain `Act of Composition`. The pattern of historical role and participation in Figure 5 makes it explicit that a person is considered a composer in virtue of his/her participation in the act of composition.

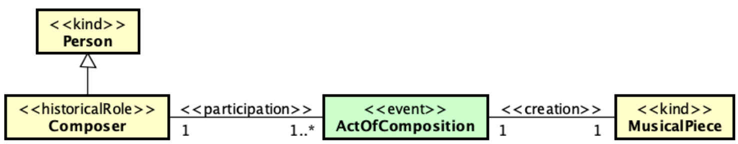

Fig. 5. Participation of a Person into Acts of Composition playing the Composer role

Historical roles can also be used to make explicit the variety of roles that endurants may play in events of a certain type. Figure 6 shows a model in which soccer players participate in soccer matches, along with a possible referee. Participation of referees is optional according to this model to cope with those informal settings when no referee is present. The model makes explicit that a person may participate in a `Soccer Match` in different roles (`Soccer Match Player` or `Referee`). (A constraint enforcing that a person does not participate in both capacities in the same match is usually required.)

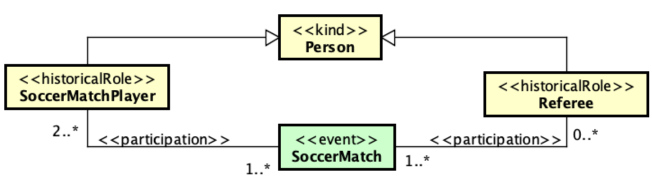

Fig. 6. Participation of Soccer Players and Referees in Soccer Matches

Furthermore, introducing historical roles allows us to distinguish explicitly between role playing in the scope of a (current) relationship and role playing in (past) events. For example, we are able to distinguish the notion of soccer player as a participant of a `Soccer Match`, i.e., someone whose dispositions were manifested in a soccer match (`Soccer Match Player`), and, soccer player as a hired professional, i.e., someone that maintains an employment relationship with a `Soccer Club`

(`Hired Soccer Player`). As shown in Figure 7, `Hired Soccer Player` is stereotyped ≪*role*≫, and is thus associated through a ≪*mediation*≫ to an `Employment` (≪*relator*≫), while `Soccer Match Player` is stereotyped ≪*historicalRole*≫ and is thus associated through a ≪*participation*≫ to a `Soccer Match` (≪*event*≫). If we consider that only "current" employments are represented, when fired by a `Soccer Club`, a `Person` no longer instantiates `Hired Soccer Player`. However, having played in a `Soccer Match`, a `Person` will always instantiate `Soccer Match Player`.

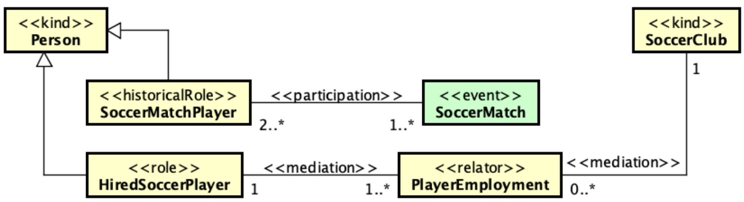

Fig. 7. Roles in virtue of relations versus historical roles

In [12], the authors discuss the duality between relators and events that are their manifestations. For example, a marriage as an event is the manifestation of properties of the marriage as a relator (mutual commitments and claims). As previously discussed, relators (and endurants in general) can change their (contingent) properties while remaining the same; events, in contrast are immutable, i.e., they cannot change in any way while keeping their identity. This aspect of (un)changeability gives us a methodological guideline for chosing to model an event or its endurant counterpart.

3.3 Mereological Relations between Events

Following Pribbenow [18], we identify different ways in which a whole may be decomposed into parts. Pribbenow discusses that parts may be: (i) "structure dependent", in which case the whole-part relations belong to the definition of the decomposed entity, e.g., the chapters of a book or the functional parts of a machine, (ii) or otherwise "constructed", in which case the whole-part relations are derived or induced using internal features of the parts or external schemes of reference. In the case of "constructed parts", by using internal features, we partition an entity into parts called "portions". For example, we may consider the "portions" of a house according to their colors. In this case, we would identify "red" parts, "brown" parts, "white" parts, etc. By using external schemes, we induce parts called "segments". An example of such external scheme is a spatial frame, so we may decompose a house into its "segment" that lies within 5m of the road, and the rest of it (the "segment" that lies over 5m from the road).

We use the classification proposed by Pribbenow to understand the ways in which an event can be decomposed in UFO-B, reflecting ultimately on our modeling recommendations. In the case of structure dependent decompositions (for example, that between a `Soccer Match` and a `Goal`, or between a `Soccer Championship` and each `Soccer Match`), no stereotype is used. Differently, for constructed decompositions, we distinguish between participational and temporal decompositions and introduce the ≪*participational*≫ and ≪*temporal*≫ stereotypes for part-whole relations.

In the case of the decomposition of an event into participations, we have "constructed" "portions", which are projected out of the whole considering their formal relation of dependence on specific endurants. Consider a meeting with multiple participants. The participation of each participant is a "portion" of the meeting in this sense. A portion is maximal with respect to the property under consideration: the portion of the meeting that is John's participation in the meeting covers all events that are part of the meeting and that depend solely from John. It is further disjoint from other portions of the meeting using the same criterion (that must per definition be participations of other participants, and hence disjoint from John's participation.) This kind of event decomposition is marked with the stereotype ≪*participational*≫. The maximum cardinality in the association end attached to the participant is always one, reflecting the rule in UFO-B that participations depend exclusively on a single endurant. Figure 8 shows the combined use of structural and participational decompositions in the soccer example.

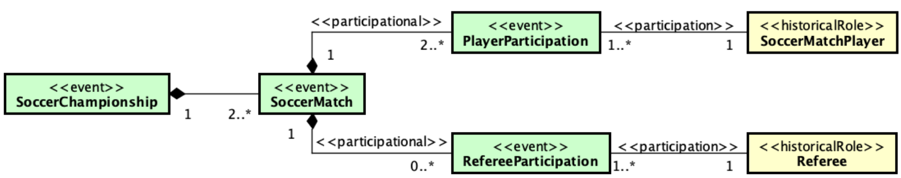

Fig. 8. Structural and participational decompositions involving Soccer Matches

In the case of the decomposition of an event into temporal parts, we have "constructed" "segments" using temporal schemes as external reference. For example, the temporal "segments" may be projected out of the whole by reference to a fixed time interval, durations, or temporal relations to other events. Consider segmenting a day-long meeting into two "segments": before and after noon. The meeting's afternoon "segment" is disjoint from the meeting's morning "segment". In addition, similarly to participational portions, temporal segments are maximal with respect to the temporal relations under consideration: in this case, there are no parts of the meeting that are fully contained in the considered time interval that are not part of the segment under consideration. This kind of event decomposition is marked with the stereotype ≪*temporal*≫.

Since events cannot be involved in part-whole relations with endurants (and vice-versa), any class stereotyped with ≪*event*≫ can only participate in part-whole relations with other classes stereotyped ≪*event*≫. Further, temporal relations can be inferred from the part-whole relations between events. More specifically, following UFO-B, the begin point of the whole must precede or coincide with the begin point of the part and the end point of the part must precede or coincide with the end point of the whole. Further, since whole part relations between events follows extensional mereology, the weak supplementation principle must be enforced. This means that the sum of lower bounds of parts must be equal to or greater than two.

3.4 Historical Dependence between Events

A final stereotype (≪*historicalDependence*≫) is defined to capture historical dependence between events. An event *b* depends historically on *a* whenever: (i) *a* (or one of its parts) *brings about* the situation that *triggers b* (or one of its parts); (ii) *a* (or one of its parts) brings about a situation that is necessary—but not sufficient—to *trigger b* (or one of its parts); (iii) *a* (or one of its parts) brings about a situation that is necessary—and more than sufficient—to *trigger b* (or one of its parts); or, (iv) *b* depends historically on an event *z* that depends historically on *a*.

Condition (i) encompasses direct causation, and, together with (iv), encompass indirect causation, which are grounds for historical dependence. Take, for instance, the relation between penalty kicks and goals. A particular `Goal` may be historically dependent on a `Penalty Kick` by being caused by it. Conditions (ii) and (iii) cover the cases of historical dependence in which there is dependence but not causation. Consider, e.g., the relation between penalties and penalty kicks. A penalty is necessary but not sufficient to cause a penalty kick (e.g., because authorization of the referee is required). A model capturing these historical relations between events is shown in Figure 9. It captures the fact that every penalty kick is historically dependent on a penalty, and that some goals (penalty goals) are historically dependent on a penalty.

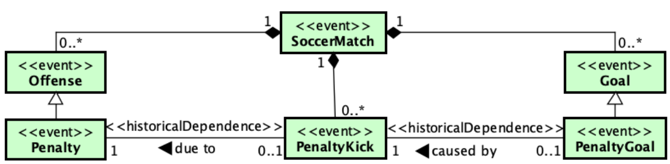

Fig. 9. Historical dependence between events of certain types in a soccer match.

4 Applying the Profile to Model Software Testing Processes

In this section, we revisit the model presented in [20], more specifically the fragment that represents software `Testing Processes`, the activities that comprise them, artifacts used and produced by those activities, and the people involved. Figures 10 and 11 present this fragment re-engineered using the profile previously discussed.

As Figure 10 shows, a `Testing Processes` is structurally decomposed into activities (events) of the following types: `Test Case Design`, `Test Coding`, `Test Execution`, and `Test Result Analysis`. To uniformly represent the temporal attributes of the events represented in the model, we included an abstract `Event` superclass that has two attributes, one stereotyped ≪*begin*≫ and the other stereotyped ≪*end*≫, capturing the temporal boundaries of the events (see Section 3.1). The historical dependencies shown in Figure 10 reflect the fact that some activities use artifacts produced by others, as shown in Figure 11. For example, `Test Execution` historically depends on `Test Coding`, since `Test Execution` uses `Test Code` produced in `Test Coding`.

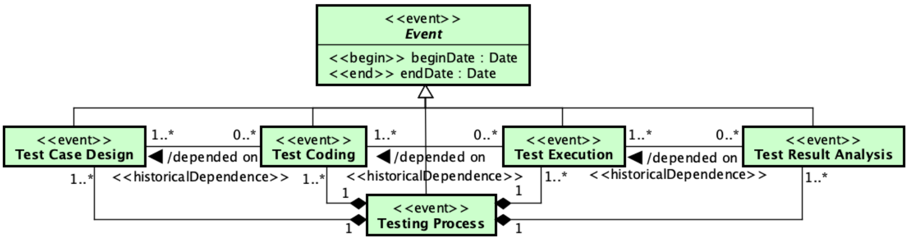

Fig. 10. Software Testing Process Model Revisited – Composition and Dependencies

The participations of endurants in the activities that compose the testing process are detailed in Figure 11. The model makes explicit the historical roles related to the participation of a `Person` (`Test Case Designer`, `Test Coder`, `Test Executor`, `Test Result Analyst`) in each activity of the `Testing Process`. These activities also involve the participation of several types of artifacts (`Document` and `Code`), some of which are created (≪*creation*≫) during the corresponding activity (e.g., `Test Code` was created during `Test Coding`); or simply participated (≪*participation*≫) in the activities (e.g., `Test Code` was used in `Test Execution`). The model omits the historical roles played by those artifacts in the corresponding activities for brevity. An exception is the case of `Tested Code`, which is the historical role of a `Code` when tested in a `Test Execution` activity. This was included to capture the difference between `Tested Code` and `Test Code`. While the former is the historical role played by a `Code` due to its participation in a `Test Execution` activity, the latter (`Test Code`) is a ≪*subkind*≫ of `Code` that was developed specifically for the purpose of testing.

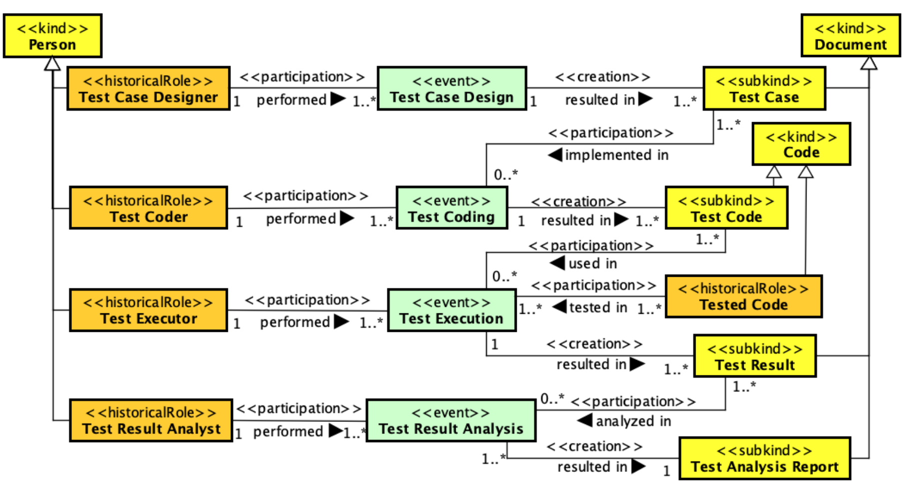

Fig. 11. Software Testing Process Model Revisited – Participations

5 Related Work

The approaches that are most closely related to ours are [5], and [17], both of which focus on UML and, similarly to our approach, treat events as entities with *"identity and properties"* which *"may participate in relationships"* and *"can be specialized"*.

In [5], we have: (a) the representation of part-whole relations between events; (b) the temporal succession of events, also with start and ending events; (c) the modeling of participants of events. Regarding (a), the proposal allows for partonomy structures and define parthood as a subtype of the Allen during relation, a point that is in conformance with UFO-B. The authors, however, are not explicit regarding the semantics of the parthood relations. For example, it is unclear under what conditions two occurrences are the same in their approach and, there seems to be no constraint proscribing the creation of an occurrence A that is composed of one single sub-occurrence B, which is different from A (thus, breaking the so-called *weak supplementation axiom*, a basic axiom in mereology). Since our approach is grounded in UFO-B, it makes clear that our semantics of composition is one of *extensional mereology*. Regarding (b), both approaches can represent Allen relations (although the authors only explicitly consider the case of *before* and *during*). However, we believe the modeling choice adopted here is superior in terms of clarity and flexibility. By marking the begin and end point of events as (immutable) attributes, we can: use different types of datatypes for representing temporal points; easily represent instantaneous events without the need for an extra construct; make explicit that *all* Allen relations are *derived* (and provide OCL derivation rules for all of them). Still regarding (b), our approach differentiates the derived relation of temporal precedence from the much stronger relation of *historical dependence*, which can be used to represent direct and indirect causation. Regarding (c), the authors represent participation by making occurrence types a specialization of UML association classes. On one hand, this makes clear that the author recognizes that occurrences are existentially dependent on its participants. On the other hand, it inherits all the problems of association classes (see [10]), including making them identical to the association to which are connected, and their consequent inability to properly model anadyc relations (e.g., we could not model an occurrence type that would have a variable number of participants from instance to instance). Moreover, since participations are modeled as association ends, there is no systematic connection between participation and parthood.

In [17], aspect (a) is simply not discussed. However, once events are represented as standard classes, one can easily represent parthood between events using, for example, UML's relations of composition and aggregation. Moreover, since events for Olivé et al., have all their properties immutable, the correct relation in this case would correspond to one of the interpretations of the composition construct in UML (the whole would be existentially dependent on its sparts). Since they do not explicitly discuss parthood for events, the authors also do not discuss the semantics on this relation for events. There is, however, an even more problematic issue. For the authors, events are always instantaneous and, hence, even if we can represent mereologically complex events, we are restrict to events which all parts happen at the exact same instant and, hence, there would be no way to represent events that unfold in time with their parts as well as non-convex events. Regarding (b), the authors represent *effects* of events by invoking operations. This can be seen as a sort of direct causality (which implies temporal precedence of the cause). However, the events that are the effects on an event are not themselves modeled as events in the same sense(!) but as operations of other classes to be invoked. As such they are not subject to all the benefits of explicit event modeling. Additionally, despite implicitly dealing with direct causality and, hence, also implicitly with temporal precedence, the authors do not discuss the modeling of all the remaining Allen relations. Regarding (c), object participation in events is modeled via regular (immutable) associations. Here, once more, there is no connection between participation and parthood and, in particular, also due to the aforementioned limitations, no way to represent mereologically complex participations. Although Olivé et al. [17] briefly considers a semantics for events in which they permanently exist as instances of the model, none of these approaches analyze the (non-trivial) consequences of introducing events in structural models. As discussed in [12], events are locked in the past and, as a consequence, introducing them in structural models changes the semantics of the model to a *historical semantics*, affecting not only them but also the objects to which they are necessarily connected. Here, we make explicit the connection between event participation and (some aspects) of dynamic endurant type instantiation, including phase changes and role playing. However, we make clear the use of events as truthmakers for role playing entails a special semantics of historical role. It is in this latter sense that, for example, in Wikipedia, Paul Newman *"is an actor"* with active years (1953–2008).

6 Final Considerations

This paper makes a contribution to conceptual modeling by proposing a profile for the representation of events in structural conceptual models. This profile was developed to address modeling requirements collected from observing the practice of the OntoUML community while creating ontology-driven conceptual models (*claim to relevance*). By employing a well-tested language engineering method, this profile was developed to reflect the ontological distinctions and formal semantics put forth by the foundational ontology UFO-B (*claim to ontological adequacy*). The proposed profile is then employed here to model a fragment of an existing reference model in the area of Software Testing. As this exercise demonstrates, the profile is able to provide specialized semantics to the various relations between events and between events and endurants. The nature of these relations remained implicit in the original model (*claim to applicability*).

Finally, this work is part of a research program aimed at addressing a fuller evolution of OntoUML [11, 14]. As discussed in [14], the development of new ontological foundations for OntoUML and the systematic redesign of its metamodel creates a number of possibilities regarding approaches to mappings from OntoUML to codification languages (e.g., OWL), model verbalization, model simulation, support for patterns and detection of anti-patterns. Revisiting the current modeling support for OntoUML in light of the developments discussed in this paper is part of our current research program.

Acknowledgments

This work has been partially supported by CNPq (407235/2017-5, 312123/2017-5), CAPES (23038.028816/2016-41), FAPES (69382549) and FUB (OCEAN Project).

References

1. Allen, G.N., March, S.T.: The ontological treatment of the 'Event' construct: Implications for system analysis and design. In: Proc. 5th Symposium on Research Systems Analysis and Design (2000)
2. Allen, G.N., March, S.T.: The effects of state-based and event-based data representation on user performance in query formulation tasks. MIS Quarterly pp. 269–290 (2006)
3. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **26**(11), 832–843 (1983)
4. Benevides, A.B., Bourguet, J.R., Guizzardi, G., Peñaloza, R., Almeida, J.P.A.: Representing a reference foundational ontology of events in SROIQ. Applied Ontology (forthcoming)
5. Bock, C., Odell, J.: Ontological behavior modeling. Journal of Object Technology **10**(3), 1–36 (2011)
6. Cabot, J., Olivé, A., Teniente, E.: Representing temporal information in UML. In: Proc. UML Conference. pp. 44–59. Springer (2003)
7. Carvalho, V.A., Almeida, J.P.A., Guizzardi, G.: Using reference domain ontologies to define the real-world semantics of domain-specific languages. In: Proc. 26th CAiSE. pp. 488–502. Springer (2014)
8. Galton, A.: Processes as continuants (abstract). In: Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06). pp. 187–187 (June 2006)
9. Guarino, N., Guizzardi, G.: "We need to discuss the relationship": revisiting relationships as modeling constructs. In: Proc. of the 27th CAISE. Springer (2015)
10. Guizzardi, G.: Ontological foundations for structural conceptual models. Ph.D. thesis, CTIT, Centre for Telematics and Information Technology, Enschede (2005)
11. Guizzardi, G., Fonseca, C.M., Benevides, A.B., Almeida, J.P.A., Porello, D., Sales, T.P.: Endurant types in ontology-driven conceptual modeling: Towards OntoUML 2.0. In: International Conference on Conceptual Modeling. pp. 136–150. Springer (2018)
12. Guizzardi, G., Guarino, N., Almeida, J.P.A.: Ontological considerations about the representation of events and endurants in business models. In: International Conference on Business Process Management. pp. 20–36. Springer (2016)
13. Guizzardi, G., et al.: Towards ontological foundations for the conceptual modeling of events. In: Proc. 32th ER Conference. LNCS, vol. 8217, pp. 327–341. Springer (2013)
14. Guizzardi, G., et al.: Towards ontological foundations for conceptual modeling: the Unified Foundational Ontology (UFO) story. Applied ontology **10**(3-4), 259–271 (2015)
15. Lyons, J.: Semantics, vol. 2. Cambridge university press (1977)
16. Nardi, J.C., Falbo, R.A., Almeida, J.P.A., Guizzardi, G., Pires, L.F., van Sinderen, M.J., Guarino, N., Fonseca, C.M.: A commitment-based reference ontology for services. Information systems **54**, 263–288 (2015)
17. Olivé, A., Raventós, R.: Modeling events as entities in object-oriented conceptual modeling languages. Data & Knowledge Engineering **58**(3), 243–262 (2006)
18. Pribbenow, S.: Parts and wholes and their relations. In: Mental Models in Discourse Processing and Reasoning, Advances in Psychology, vol. 128, pp. 359–382. North-Holland (1999)
19. Sider, T.: Quantifiers and temporal ontology. Mind **115**(457), 75–97 (2006)
20. Souza, E., Falbo, R., Vijaykumar, N.: ROoST: reference ontology on software testing. Applied Ontology **12**(1), 59–90 (2017)
21. U.S. Department of Defense (DoD): Data Modeling Guide (DMG) for an Enterprise Logical Data Model (ELDM). U.S. Department of Defense (DoD) Report (2011)