Representing Specialized Events with FrameBase

Jacobo Rouces[1], Gerard de Melo[2], and Katja Hose[1]

[1] Aalborg University, Denmark
jrg@es.aau.dk, khose@cs.aau.dk
[2] Tsinghua University, China
gdm@demelo.org

Abstract. Events of various sorts make up an important subset of the entities relevant not only in knowledge representation but also in natural language processing and numerous other fields and tasks. How to represent these in a homogeneous yet expressive, extensive, and extensible way remains a challenge. In this paper, we propose an approach based on FrameBase, a broad RDFS-based schema consisting of frames and roles. The concept of a frame, which is a very general one, can be considered as subsuming existing definitions of events. This ensures a broad coverage and a uniform representation of various kinds of events, thus bearing the potential to serve as a unified event model. We show how FrameBase can represent events from several different sources and domains. These include events from a specific taxonomy related to organized crime, events captured using schema.org, and events from DBpedia.

1 Introduction

The surge of research on large-scale knowledge bases in recent years has largely been driven by the availability of new sources of information about entities. While structured data about millions of places, people, or companies are very valuable, there have been comparably few new results on capturing events of various sorts. Most existing event-oriented ontologies have introduced only a few abstract classes of events, and typical knowledge bases tend to describe just a small number of specific types of events.

Often, however, there is a need to talk about a broad range of very specific sorts of events. For instance, one might want to distinguish battles from both gunfights and from wars, and capture the class-specific details of such events. We adopt a broad notion of events here. This includes the prototypical cases, e.g. local happenings such as concerts, gatherings, or competitions, and world events such as those reported in the news. It also encompasses the more general abstract definition of events, for instance as "happenings in the real world" [15], which would include, e.g., the birth of a person or a commercial transaction between two people. Clearly, such events make up an important aspect of the world that is relevant in knowledge representation, natural language processing, and numerous other fields and tasks. Occasionally, the term *eventuality* is used to denote a broader notion of events that explicitly includes states, e.g. two people knowing each other.

In this paper, we address this challenge of representing many different notions of events under a common schema, from the very prototypical cases to the very abstract, in a way that has both a broad coverage yet supplies sufficient detail to model event-specific properties. For this, we present a new approach for representing event information that is based on FrameBase [12], a broad RDFS-based schema made of frames and their roles. FrameBase provides a predefined vocabulary with event-specific properties for thousands of different kinds of events. For instance, FrameBase's schema accounts for the fact that a battle takes place in a certain time and place and normally involves two parties. For this, the schema draws on two linguistic resources, FrameNet [2] and WordNet [6]. As these describe important fragments of the English lexicon, their coverage is quite substantial. Additionally, as we illustrate later on, FrameBase can be easily extended.

In the following, we prove the suitability of FrameBase for representing different kinds of events by creating rules that integrate instances from different domains:

- A taxonomy of event classes relating to organized crime from the EU FP7 project ePOOLICE[3]. In the project, the event classes in the taxonomy are used as types of entities that are extracted from documents crawled from the web, as part of a strategic early-warning system. The taxonomy was originally captured using the Conceptual Graphs formalism [17]. We use and integrate the event taxonomy as it is, without ad-hoc modifications to the schema.
- The subclasses and properties of the "Event" class in schema.org, which "provides a collection of schemas that webmasters can use to markup HTML pages in ways recognized by major search providers, and that can also be used for structured data interoperability" [1].
- The subclasses and properties of the "Event" class in DBpedia [4], which are extracted from the infoboxes in Wikipedia.
- We conclude with a more general overview of how salient aspects of events [15] can be mapped into FrameBase.

This paper is structured as follows. After describing previous approaches and research in Section 2, a brief overview of the FrameBase schema is given in Section 3. Section 4 then shows how we can rely on the FrameBase schema to represent events from several different sources and domains. Finally, Section 5 provides concluding remarks and describes avenues for future research and applications of our work.

2 Related Work

Considering their importance and unique characteristics, events have been included in numerous upper-level ontologies and vocabularies. In [15], existing event models are reviewed, but these define very broad abstract categorizations or meta-models. Only few example specializations or vocabularies for narrow domains exist, and their overall size is relatively small.

For instance, the Simple Event Model (SEM) Ontology [18] introduces the four types *Event*, *Actor*, *Place*, and *Time*. While it provides a mechanism to create more specific ones by extending these, it does not actually define any specific kinds of events itself. Similarly, the LODE (Linking Open Descriptions of Events) model [16] provides very general concepts, such as the four just mentioned. The event model E [14] proposes a generic structure for the definition of events, but a specific vocabulary is provided

[3] https://www.epoolice.eu/

only for the domain of media events with sensor data. The Event Ontology [11] defines a single event class, for which time, place, agents, factors, products, and meronymic relations can be specified, and the domain of focus is music events. Likewise, the Context Ontology (CONON) is limited to the domain of pervasive computing environments [19].

FrameBase's schema instead aims at a broader coverage of many domains by building on natural language resources. Previous work has made use of natural language processing techniques to extract events from text. For instance, one study [5] relies on semantic role labelling (SRL) in conjunction with VerbNet to collect events from text and convert them to the LODE vocabulary mentioned above. Another system [10] extracts events both from text and from semi-structured data. We believe that such automatic extraction methods would benefit from being able to use a standardized wide-coverage representation schema for their output.

3 The FrameBase Schema

The FrameBase schema [12] consists of classes representing frames, and properties representing frame elements. A *frame* describes any kind of situation, state or action, in which several elements, participants (agents, patients, etc.) or properties are involved. Examples include commercial exchanges, marriages, or the act of stomping. The *frame elements* refer to the participants or properties that are involved in a particular frame instance. Common general frame elements include those of agent, patient, time, and location, but not all frames involve these. Frame elements are sometimes also referred to as semantic roles, roles, or theta roles, especially when they are very general.

The frames and the frame elements in FrameBase are organized in hierarchies of classes (based on subclass relationships) and of properties (based on subproperty relationships), respectively. There are three kinds of frames in FrameBase: LU-microframes, synset-microframes and non-lexical frames. Non-lexical frames are very general and are situated in the upper part of the hierarchy. LU-microframes (lexical unit microframes) descend from non-lexical frames, but are much more specific by being associated with the meaning of particular words (the lexical units). They come from FrameNet [7, 13]. Synset-microframes allow an intermediate level of granularity connecting synonymous LU-microframes, e.g. for *marriage* and *matrimony*. These are based on WordNet [6], and thus also have allowed us to extend the coverage of FrameBase beyond that of FrameNet. In the field of linguistics, frames are said to be evoked by words: for example, both the verb *to create* and the noun *creation* evoke the `Creation` frame.

FrameBase additionally provides direct binary predicates to directly connect certain values for elements of a given frame. For example, in a creation event, the agent and the place are directly connected via the `establishesInPlace` relation. This enables more concise queries and representations when only two elements are involved in a particular frame. The frame patterns and the direct binary predicates are logically connected by means of definite clauses that can be used with different kinds of inference systems.

For interoperability with existing resources, FrameBase relies on the standard RDF model [9], which has become a common choice for representing knowledge. This is particularly true in the context of the Linked Data [3], a large Web of datasets referring to and reusing each other's elements. The RDF model uses subject-predicate-object triples to represent statements. Each triple can also be seen as an edge in a directed labelled entity-relationship graph. SPARQL [8] is the standard query language for RDF, which is what we use in order to integrate other event representations into FrameBase.

Event frames are specific kinds of frames, subsuming a range of different notions of events, from the very abstract (e.g., "a natural abstraction of happenings in the real world" [15]) to notions with a notably narrower scope, such as that of widely-known events [10]. Frame elements correspond to what are referred to as *aspects* in the event literature [15]. However, frames can also be more general, and include what the event model E categorizes separately as entities [14]. For example, FrameNet, from which FrameBase is derived, includes a frame `People` that is evoked by lexical units (LUs) such as the noun *man*, and with frame elements such as `Age` and `Origin`.

We believe that the advantage of FrameBase over the existing event models lies on the fact that while extensible as the others, it already provides a broad-coverage vocabulary out of the box in order to bootstrap widespread adoption. Besides, its connection to natural language provides potential advantages, like interfacing with text for question answering or text mining.

FrameBase includes, from FrameNet, an Event frame, which inherits from the `Change of state scenario` frame, and includes a relatively rich hierarchy below for events like creation and destruction events (including more specific ones such as births and deaths), and some others. However, not every event must necessarily fall below this event frame, nor does doing so preclude it from being mapped to other frames that represent other conceptualizations for events, or reflect other perspectives of the frame that stress different aspects than the eventive one. Therefore, the representation of events in FrameBase is not confined to the Event frame and its subframes. We will see examples of this in the next section.

4 Integrating Events

In the first subsections of this section, we present manually built rules for integrating events from three different sources into FrameBase. Later, we add further explanations about these rules and discuss the complexity of the integration rules, and the challenges they present, in particular when they are to be established automatically.

4.1 Representing Events about Organized Crime

The following list of integration rules shows, for each instance of an event class in the organized crime conceptual graph (in bold), the corresponding representation in RDF that it would have in FrameBase. In particular, the main event instance is represented by the anonymous node `_:e`. The default prefix indicates elements that already existed in the core FrameBase schema created from FrameNet and WordNet.

- **Event** `_:e a :frame-Event-event.n`
  - **Act** `_:e a :frame-Intentionally_act-act.n`
    - **Arrest** `_:e a :frame-Arrest-arrest.n`
      - **Drug Possession Arrest** `_:e a :frame-Arrest-arrest.n .`
        `_:e :fe-Arrest-Offense _:e2 .`
        `_:e2 a :frame-Offenses-possession.n`

Human Trafficking Arrest _:e a :frame-Arrest-arrest.n .
_:e :fe-Arrest-Offense _:e2 .
_:e2 a :frame-Commerce_scenario-trafficker.n .
_:e2 :fe-Commerce_scenario-Goods :frame-People-human.n

Metal Theft Arrest _:e a :frame-Arrest-arrest.n .
_:e :fe-Arrest-Offense _:e2 .
_:e2 a :frame-Theft-theft.n .
_:e2 :fe-Theft-Goods :frame-Substance-metal.n .
_:e2 a :frame-Offenses-theft.n

Buy _:e a :frame-Commerce_buy-buy.v

Crime _:e a :frame-Committing_crime-crime.n

Illegal Drug Use _:e a :frame-Ingest_substance-use.v

Consume _:e a :frame-Ingestion-consume.v

Inhale _:e a :frame-Ingest_substance-sniff.v

Inject _:e a :frame-Ingest_substance-inject.v

Possession _:e a :frame-Offenses-possession.n

Smoke _:e a :frame-Ingest_substance-smoke.v

Organised Crime
_:e a fbe:frame-Organization-criminal%20organization.n

Theft _:e a :frame-Theft-theft.n .
_:e a :frame-Offenses-theft.n

Metal Theft _:e a :frame-Theft-theft.n .
_:e :fe-Theft-Goods :frame-Substance-metal.n .
_:e a :frame-Offenses-theft.n

Trafficking _:e a :frame-Commerce_scenario-trafficker.n

Drug Trafficking
_:e a :frame-Commerce_scenario-trafficker.n .
_:e :fe-Commerce_scenario-Goods :frame-Intoxicants-drug.n

Human Trafficking
_:e a :frame-Commerce_scenario-trafficker.n .
_:e :fe-Commerce_scenario-Goods :frame-People-human.n

Seizure _:e a :frame-Taking-seizure.n

Drug Seizure _:e a :frame-Taking-seizure.n .
_:e :fe-Taking-Theme :frame-Intoxicants-drug.n

Sell _:e a :frame-Commerce_sell-sell.v

Transaction _:e a :frame-Commercial_transaction-transaction.n

Crime Transaction
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n

Drug Trafficking Transaction
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-Intoxicants-drug.n

Human Trafficking Transaction

```
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-People-human.n
```

Metal Theft Transaction

```
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-Substance-metal.n
```

The hierarchy in the original ontology is not necessarily consistent with the hierarchy in FrameBase. Only in certain cases does a superclass relationship between two elements of the source also exist between the two elements' respective translations to FrameBase. Therefore, for each translation of an original class of event, the translations of the parents in the original ontology can be added to the set of instances (ABox) in FrameBase, and this will provide additional knowledge that would not always be inferred by the FrameBase schema alone.

We minimize the need for declaring new frames and frame elements for specialized domains by making use of the compositionality of most specialized terms, creating complex structures that combine the semantics of simpler, basic elements. For instance, the translation for the event of type "Drug Possession Arrest" declares an event of type arrest, and specifies that it is about drug possession by assigning drug possession (`Offenses-possession.n`) as the offence.

Owing to this flexibility, we merely needed to mint one single new entity that had not existed in the core FrameBase schema (the microframe `Organization-criminal%20-organization.n`, with the prefix `fbe:` denoting that this is an extension). This exemplifies the potential of FrameBase to represent events from relatively specialized domains, but at the same time the capacity to be extended to fill any possible gaps.

For representing timelines, the frame `Individual_history-history.n` can be used. Each timeline can be represented with one instance of that frame. This instance can be linked with the frame element `Individual_history-Domain` to the topic, which is preferably an entity (or alternatively, a literal or an anonymous node or dummy entity named with a literal). The instance can also be linked with the frame element `Individual_history-Event` to each of the elements in the timeline. Additional frame elements are available in FrameBase, originating from FrameNet, for expressing participants, total duration, etc.

Then, complex queries such as retrieving all events in a given timeline between two given dates, can be built in SPARQL. Similarly, sub-events can be represented with the property path: `^:fe-Part_whole-Part/:fe-Part_whole-Whole`.

4.2 Representing Events from DBpedia.org

We now turn to the `Event` class in DBpedia, and its subclasses, showing how these can be integrated into FrameBase. The integration is implemented using SPARQL CONSTRUCT rules because DBpedia is already in RDF. We only add a couple of subclasses, but most of the properties belong to the parent `Event` class itself.

Top event

```
CONSTRUCT {
  ?e a :frame-Event-event.n .
```

```
  ?e :fe-Event-Time _:timePeriod .
    _:timePeriod a fbe:frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?o1 ; fbe:fe-Timespan-End ?o2 .
  _:e2 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?e ;
    :fe-Relative_time-Focal_occasion ?o3 .
  _:e3 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?o3 ;
    :fe-Relative_time-Focal_occasion ?e .
  _:e4 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?e ;
    :fe-Relative_time-Focal_occasion ?o4 .
  _:e5 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?o4 ;
    :fe-Relative_time-Focal_occasion ?e .
  ?e :fe-Event-Reason ?o5 .
  ?e a :frame-Social_event-meeting.n ;
    :fe-Social_event-Attendee ?o8 .
} WHERE {
  ?e a dbpedia-owl:Event .
  OPTIONAL{?e dbpedia-owl:startDate ?o1}
  OPTIONAL{?e dbpedia-owl:endDate ?o2}
  OPTIONAL{?e dbpedia-owl:previousEvent ?o3}
  OPTIONAL{?e dbpedia-owl:followingEvent|dbpedia-owl:nextEvent ?o4}
  OPTIONAL{?e dbpedia-owl:causedBy ?o5}
  OPTIONAL{?e dbpedia-owl:duration ?o6}
  OPTIONAL{?e dbpedia-owl:numberOfPeopleAttending ?o7} #Omitted
  OPTIONAL{?e dbpedia-owl:participant ?o8}
}
```

For sub-classes of dbpedia-owl:Event

```
CONSTRUCT {
  ?e a :frame-Social_event-meeting.n .
} WHERE {?e a dbpedia-owl:SocietalEvent}
```

For sub-classes of dbpedia-owl:SocietalEvent

```
CONSTRUCT {
  ?e a :frame-Project-project.n .
  ?e :fe-Project-Activity dbpedia:Space_exploration .
} WHERE {?e a dbpedia-owl:SpaceMission}
```

For sub-classes of dbpedia-owl:SocietalEvent

```
CONSTRUCT {
  ?e a fbe:frame-Social_event-convention.n .
} WHERE {?e a dbpedia-owl:Convention}
```

Out of the 9 properties of the class `Event`, the only omitted one was `numberOfPeopleAttending`, because the class `Event` is too general for it, as it has sub-classes such as `NaturalEvent` (`SolarEclipse`) and `PersonalEvent` (`Birth`, etc.). The `SocietalEvent` class appears more appropriate for this.

4.3 Representing Events from schema.org

Finally, we present the translation of the Event class in schema.org. Again, SPARQL CONSTRUCT rules are used because schema.org can be expressed using RDFa, and SPARQL offers a standard way of representing knowledge graph transformations. Due to space restrictions, we omit the subclasses here, but these have very few genuine properties, and therefore the specialization is relatively simple. Besides, the taxonomy of schema.org events has some inconsistency issues that makes its use complex: the Event class is defined as capturing events such as concerts, lectures, and festivals, with properties such as "typical age range", but there are sub-events such as UserInteraction and UserPlusOnes that actually represent a more general kind of events.

```
CONSTRUCT {
  ?e a :frame-Social_event-meeting.n .
  ?e :fe-Social_event-Time _:timePeriod .
    _:timePeriod a fbe:frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?Osta ; fbe:fe-Timespan-End ?Oend .
  ?e :fe-Social_event-Duration ?Odur . ?e :fe-Social_event-Place ?Oloc .
  ?e :fe-Social_event-Attendee ?Oatt . ?e :fe-Social_event-Host ?Oorg .
  ?e :fe-Social_event-Occasion ?Osup . ?Osub :fe-Social_event-Occasion ?e .
  ?Ooff a :frame-Offering-offer.v ;
    :fe-Offering-Theme ?e .
  ?e a :frame-Performing_arts-performance.n ;
    :fe-Performing_arts-Performer ?Oper ;
    :fe-Performing_arts-Performance ?Owor .
  _: a :frame-Recording-record.v ;
    :fe-Recording-Phenomenon ?e ;
    :fe-Recording-Medium ?Orec .
} WHERE {
  ?e a sch:Event .
  # Unambiguous translation
  OPTIONAL{?e sch:startDate ?Osta}     OPTIONAL{?e sch:endDate ?Oend}
  OPTIONAL{?e sch:duration ?Odur}      OPTIONAL{?e sch:location ?Oloc}
  OPTIONAL{?e sch:attendee ?Oatt}      OPTIONAL{?e sch:organizer ?Oorg}
  OPTIONAL{?e sch:superEvent ?Osup}    OPTIONAL{?e sch:subEvent ?Osub}
  OPTIONAL{?e sch:offers ?Ooff}        OPTIONAL{?e sch:performer ?Oper}
  OPTIONAL{?e sch:workPerformed ?Owor} OPTIONAL{?e sch:recordedIn ?Orec}
  # Ambiguous translation
  OPTIONAL{?e sch:doorTime ?Odoo}
  # No translation
  OPTIONAL{?e sch:eventStatus ?Oeve}
  OPTIONAL{?e sch:typicalAgeRange ?Otyp}
  OPTIONAL{?e sch:previousStartDate ?Opre}

}
```

The only extension of the FrameBase schema used here was the frame `:frame-Timespan-period.n` with the start and end frame elements, used to denote periods of time. This, however, is not an ad-hoc extension motivated by a particular need of only one source, but a very general one. Out of the 16 properties of the `Event` class, 12 were translated without loss of meaning. One was translated with partial loss of meaning (doorTime, translated as a generic start time) and 3 of them were not translated. Whether these can be integrated too, by means of more complex structures, is something we are investigating.

4.4 Mapping Event Aspects to Frame Elements

The survey by Scherp and Mezaris [15] proposes a classification of salient aspects of events. We use this classification to show in a more general way how event aspects can relate to frame elements in the FrameNet-based schema of FrameBase.

- **Time** and **Space**: When applicable, frames include frame elements Time and Place.
- **Participation**: The classification defines this as "participation of objects in event, where objects can be any living as well as non-living things and include people, buildings, and other even intangible objects like the roles a person plays in a specific situation" [15]. FrameBase provides a large inventory of more specific roles to capture such participants. Often, these correspond to what are sometimes called the proto-agent and proto-patient roles, whose realization in FrameBase depends on the frame. Some examples are :fe-Commerce_buy-Buyer, :fe-Destroying-Destroyer and :fe-Destroying-Undergoer, which are subproperties of :fe-Getting-Recipient, :fe-Transitive_action-Agent and :fe-Transitive_action-Patient, respectively.
- Relations between events.
  - **Mereology**: The relation between two events, when one is part of another. Some frames will have a frame element that will fill this role, like :fe-Social_event-Occasion in the example of the Event class in schema.org. In other cases, an additional frame instance of type :frame-Part_whole can be used.
  - **Causality**: One event is the cause of another. Some frames will have a frame element that will fill this role, like :fe-Event-Reason in the example for the Event class in DBpedia. In other cases, an additional frame instance of type :frame-Causation can be used.
  - **Correlation**: When "two (or more) events have a common cause, but this common cause cannot be explained". If we can assume there is a common cause as in the definition, then the causal relationships can be represented with two instances of :frame-Causation connecting with an anonymous node for the unknown cause.
- **Documentation**: Events can be "documented using some media like photos or videos captured during the event". This relation is between an event and such documentation. It can be expressed connecting the events by an additional frame of type :frame-Recording-document.v, :frame-Recording-record.v, and :frame-Recording-register.v, or some extension if needed.
- **Interpretation**: This aspect aims at capturing "subjectivity that may exist on the other aspects of events". This is a very broad category that may include different phenomena. The perspectivization relation in FrameNet [13] connects frames representing objective events with frames describing them from a particular perspective. For instance, :frame-Commerce_Sell and :frame-Commerce_Buy are perspec-

tivizations of :frame-Commerce_Scenario. In other cases, an additional frame instance of a pertinent type can be used, for instance :frame-Becoming_aware.

4.5 Complex Transformations

Most of the integration rules we have described follow a pattern which involves an event *class* in the source being translated as a frame class, and each of their outgoing properties being mapped to individual frame elements. However, there are multiple ways in which the rules can differ from this basic pattern.

1. Sometimes, a class integration rule may need to instantiate multiple frames rather than just a single one. We distinguish two main types of this phenomenon.
   a) The instantiated frame instances may be connected by frame elements. Examples of this include the frame :frame-Timespan-period.n created to represent time periods, and the subframes of Relative_time to express precedence between events (all in the example for dbpedia-owl:Event). The same applies when a frame element is used to specify a frame beyond the lexical unit (see the rule for dbpedia:Space_exploration).
   b) Several frames can also be evoked separately, without the instances being directly connected by any frame element. When these frames describe different perspectives of the same event, there is the possibility that FrameNet links them by means of *perspectivization*, and therefore FrameBase can infer one from another. For example, classes :frame-Commerce_buy-buy.v and :frame-Commerce_sell-sell.v, which are used for classes Buy and Sell in the organized crime taxonomy, are both perspectivizations of :frame-Commerce_goods-transfer. In this case, inference is possible because RDFS subclass and subproperty properties are used in FrameBase to reflect the perspectivization relation between frame classes and frame elements respectively. Another example are :frame-Receive_visitor_scenario and :frame-Visit_host, which are perspectives of :frame-Visitor_and_host. However, in other cases one cannot rely on existing inference. For instance, see how the rule to translate Event from schema.org, besides frames Event-event.n and Timespan-period.n, also instantiates Performing_arts-performance.n, Recording-record.v and Offering-offer.v when certain properties are present.
2. Another possible source of complexity is that frame elements can be inverted. In this case, the integration rules need to invert the order of the arguments, like in the second appearance of :fe-Social_event-Occasion in the integration rule for the class Event in schema.org.
3. Oftentimes, a *property* (rather than a class) in the source can be translated as evoking a frame on its own. In this case, the two involved entities become connected to the new frame by means of frame elements. This would be the case for a property like fightAgainst, which might evoke an event or frame of type armed conflict, about which additional information could be added. None of the examples we have

covered above are of this kind, because we use sources that explicitly represent, or *reify*, events. In other sources, however, this phenomenon appears quite frequently.

Arbitrary combinations of these phenomena are possible (e.g. the rule integrating the Event class from schema.org). Overall, this makes automatic generation of the integration rules a very hard task, because it generates so many free variables that any attempt to train a system would face extreme sparsity. In some cases, it may thus make sense to sacrifice some recall, developing a system that only covers simpler transformations.

4.6 Representational Flexibility

Finally, another potential challenge for data integration is that even when a homogeneous schema such as FrameBase is used, certain kinds of knowledge can still be expressed in multiple possible ways.

- One example is that there are several ways of narrowing down the meaning of a frame instance. One is creating a new sub-microframe associated with a new lexical unit. Another one is assigning a value to a frame element (see example for `SpaceMission`), as mentioned above. This may lead to divergent choices of representation even within the core part of the schema that comes from FrameNet.
- Another example of this is when a frame element needs to be reified, i.e. represented as a frame instance, to express something additional about it (as would be the case of the property `previousStartDate` in schema.org), or when there is no direct frame element available and creating it would lead to a combinatorial explosion in the size of the schema. An example of the latter is the difference between our proposal for using the frame `Part_whole` for expressing sub-event relations, and how we used the frame element `Occasion` for the frame `Social_event`, but this is a particularity of that frame. Again, this may lead to an incoherent representations in the knowledge base. One potential way of addressing this would be extending the reification–dereification mechanism of FrameBase [12].

5 Conclusion

We have shown how events from specialized domains can be represented with the FrameBase schema under a unified model, integrating events in the prototypical sense with more general kinds of events in the sense of abstract happenings or situations. This model has proven to have a high degree of coverage because it needed just few extensions to accommodate the integrated knowledge, and we have illustrated how these extensions can be performed when needed. We have also discussed the various challenges and problems one faces when the integration rules from disparate structured sources of event information are to be built automatically.

Extremely specialized domains, such as quantum physics, may produce lower coverage and need more extensions, although in some cases the creators of FrameNet have also been involved in projects that led to the inclusion of specific scientific and technical domains.

The integration rules that we produce can be used in the future as gold standards for training and testing automatic methods for creating rules from other schemas. We are currently performing research on these methods to integrate further sources such as YAGO2s, Freebase, and Wikidata.

Please refer to `http://framebase.org` for information on using FrameBase and the integration rules.

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. FP7-SEC-2012-312651 (ePOOLICE project). as well as China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61450110088.

References

1. Schema.org. http://schema.org.
2. C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet Project. ICCL '98, pages 86–90, 1998.
3. C. Bizer, T. Heath, and T. Berners-Lee. Linked data–the story so far. *IJSWIS*, 5(3):1–22, 2009.
4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia-A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.
5. P. Exner and P. Nugues. Using semantic role labeling to extract events from Wikipedia. DeRiVE '11, 2011.
6. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
7. C. J. Fillmore, C. R. Johnson, and M. R. Petruck. Background to Framenet. *International journal of lexicography*, 16(3):235–250, 2003.
8. S. Harris and A. Seaborne. SPARQL 1.1 Query Language. W3C Recommendation, W3C Consortium, Mar. 2013.
9. P. Hayes and P. Patel-Schneider. RDF 1.1 semantics. Technical report, W3C, 2014. http://www.w3.org/TR/rdf11-mt/.
10. E. Kuzey and G. Weikum. Extraction of temporal facts and events from wikipedia. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, pages 25–32. ACM, 2012.
11. Y. Raimond and S. Abdallah. The event ontology. Technical report, Oct. 2007. http://motools.sf.net/event.
12. J. Rouces, G. De Melo, and K. Hose. FrameBase: Representing N-ary Relations using Semantic Frames. In *Proceedings of the 12th Extended Semantic Web Conference, ESWC*, 2015.
13. J. Ruppenhofer, M. Ellsworth, M. R. Petruck, C. R. Johnson, and J. Scheffczyk. *FrameNet II: Extended Theory and Practice*. ICSI, 2006.
14. A. Scherp, S. Agaram, and R. Jain. Event-centric media management. In *Electronic Imaging 2008*, pages 68200C–68200C. International Society for Optics and Photonics, 2008.
15. A. Scherp and V. Mezaris. Survey on modeling and indexing events in multimedia. *Multimedia Tools and Applications*, 70(1):7–23, 2014.
16. R. Shaw, R. Troncy, and L. Hardman. LODE: Linking Open Descriptions of Events. In *ASWC '09*, Lecture Notes in Computer Science, pages 153–167, 2009.
17. J. F. Sowa. Conceptual graphs. In *In Handbook of Knowledge Representation*, pages 213–237. Elsevier, 2008.
18. W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber. Design and use of the Simple Event Model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011.
19. X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22. Ieee, 2004.