A Contextual Framework for Reasoning on Events

Loris Bozzato[1], Stefano Borgo[2], Alessio Palmero Aprosio[1], Marco Rospocher[1], and Luciano Serafini[1]

[1] Fondazione Bruno Kessler,
Via Sommarive 18, 38123 Trento, Italy
[2] Laboratory for Applied Ontology, ISTC CNR,
Via alla Cascata 56 C, 38123 Trento, Italy

`{bozzato,aprosio,rospocher,serafini}@fbk.eu, stefano.borgo@cnr.it`

Abstract. In this paper we investigate the contextualized representation of events. In particular, we re-interpret the formalization of events and situations adopted in the NewsReader Event and Situation Ontology (ESO) according to the notions of context and module proposed in the Contextualized Knowledge Repositories (CKR) framework. This contextualized formalization sets the basis for exploiting logical reasoning on events and situations, enabling to automatize tasks such as: recognizing incompatible event descriptions or inconsistent situations, inferring missing or implicit events.

1 Introduction

With the growth of technologies managing the extraction of events and their participants from texts (e.g. [1,7]), interest has spread in using these low level tools as the base for event processing and reasoning in a higher level abstraction for events. The idea is to be able to discover, starting from low level descriptions of events extracted from text, complex events and their succession (viz. *stories*) with respect to their participant entities.

This is one of the goals of the *NewsReader project*[3]. In this project, Natural Language Processing (NLP) technologies, such as Named Entity Recognition (NER) and Semantic Role Labelling (SRL), are exploited to process large streams of news in multiple languages, to extract events and event participants, locations, dates. At the end of the extraction and processing, events and related information are represented using RDF. In order to define and reason over the features and effects of events, an OWL ontology has been defined for such data, called the *Event and Situation Ontology (ESO)* [11].

In this paper we propose to reinterpret and formalize the event model defined by the ESO ontology using a context-based framework for representation of Semantic Web data, the *Contextualized Knowledge Repositories (CKR)* framework [12,2,4,5]: this enables to exploit the structure and reasoning possibilities offered by the contextual framework in order to perform complex inferences about and inside knowledge associated with events.

Intuitively, CKR is a description logics (DL) based framework defined as a two-layered structure: a lower layer contains a set of knowledge bases representing each context, while the upper layer contains context independent knowledge and meta-data

---

[3] `www.newsreader-project.eu`

defining the structure of contexts. The CKR framework has not only been presented as a theoretical framework, but also actual implementations based on its definitions [5,3] have been proposed. In particular, in [5] we presented an implementation for the CKR framework over state-of-the-art tools for storage and inference over RDF graphs: intuitively, the CKR architecture can be implemented by representing the global context and the local object contexts as distinct RDF named graphs, while inference inside (and across) named graphs is implemented as SPARQL-based forward rules.

From a formal ontology point of view, in the approach we propose in this paper we clearly distinguish *ontology* from *knowledge*. The upper layer of our system contains the underlying ontology, that is, the description of the organization of the world. This part lists types of entities, relations and constraints that are assumed to exist or to simply be possible. More generally, the upper layer should be thought as including two elements: a general (foundational or domain) ontology describing what can exist, and the organization of a set of knowledge modules characterizing roles and relationships in (typically social) standardized scenarios like economic transactions or soccer games. Regarding the first, we do not make a specific commitment towards one ontology or another although we explicitly commit to the existence of physical and social objects, like people and organizations, and temporal happenings like weddings and thunderstorms. These entities come with their usual physical and temporal properties: weight, shape, duration and so on. Regarding the modules, we do explore their role in the system to infer new facts from given knowledge and thus we will present their general setting and how they are used. The lower layer of the system, on the other hand, collects claims about what happens in the world, that is, claims about how things are or change at some time. Throughout the paper these are what we call *events*, but note that they are not events in the ontological sense, rather they are descriptions of happenings.

Each event is associated with three state-like entities, namely happenings characterized by some continuously holding property or relation like “begin married”, “being running” or “being employed by” (e.g. see the notion of *stative perdurant* in DOLCE [9]). These state-like entities are called *situations* in our system; more specifically the situation holding before the event is called *pre-situation*, the one holding after *post-situation* and the one holding during the event itself is the *during-situation*. Informally, these situations are used to make explicit relevant properties that (*a*) persist during the whole event, or (*b*) hold (or don’t hold) before/after the event and their truth values change “because” of the event. We will use situations to formalize (and reason on) the preconditions for an event of a certain type to happen, the postconditions (or consequences) of its happening, and what can be taken as stable during the whole event of that type.

Although we take the received events at face value, it is possible that the news are imprecise or incomplete, that the text processing used to acquire the news is faulty and misinterprets them, that statements extracted from different sources about the same happenings contradict each other. For this reason, we take each piece of information extracted from an outside source as a contextual perspective on the world. This means that the content of a news is not equated to absolute knowledge (it is not indisputable). Indeed, the extracted information constitutes a context annotated with its original source (possibly with different degrees of reliability which can change over time). This allows to integrate pieces of information coming from different sources into large and articulated event descriptions by integrating different contexts (in turn, news statements). These extrapolated events reconstruct how an entity like a person or an organization, changes over time by changing its status (size, capacity, death) or its relationships with other entities (property acquisition, employment, marriage). Furthermore, missing information on these complex events can be detected via logical reasoning based on the ontology at the upper layer, leading to infer changes that have not been reported (or detected) in the news. Finally, note that, when a contradiction arises, the system can isolate the conflicting pieces of information and establish which contexts are to be kept apart and need to be verified.

In this paper we propose the first sketch and example of use for the contextualized ESO model: in the next section we briefly present the ESO model and provide an example of event modelling; in Section 3, we summarize the base definitions for the CKR framework; in the following section we describe the *CKR-ESO model*, that is our contextual based realization of ESO, we show how it models the previously presented event example and provide some insights in the advantages of such representation; we conclude by outlining some of the current and future directions for our work.

2 Events and Situations

One of the objectives in the NewsReader project is the representation of events and of their effects on the entities participating in them. For example, in a "giving" event, we have at least two actors (people, organizations) and an object, e.g., a person A giving something to another person B at some time T. The event also describes a change in these entities: at the time T, person A owns the object, while person B does not (aka, *pre-situation*); after T, the opposite is true (aka, *post-situation*).

To achieve this objective, the Event and Situation Ontology (ESO) [11] has been developed. It defines two main classes of entities: *events* and *situations*. An *event* describes an happening, typically a change, in the world (for instance, person A giving an object to person B at time T), has a certain number of participants (here, the two people and the object) and an associated period of time (here, T). A *situation* describes a state, i.e., takes a set of statements as describing (part of) the world at some point or interval of time. In the example "person A gives an object to person B at time T", we can identify a *pre-situation* (the state of the world at the initial time point of T):

- person A owns the object,
- person B does not own the object

and a *post-situation* (the state of the world at the ending time point of T):

- person A does not own the object
- person B owns the object.

For instance, in the sentence: "*The chairman of India's Tata Group has confirmed that his company is acquiring the Jaguar and Land Rover businesses from Ford.*", an event is described: the acquisition by Tata Group of Jaguar and Land Rover from Ford. The event is represented in ESO terminology as follows:

⟨*:evID, rdf:type, sem:Event*⟩
⟨*:evID, rdf:type, eso:Getting*⟩
⟨*:evID, rdfs:label, acquire*⟩
⟨*:evID, eso:possession-owner_1, dbp:Ford*⟩
⟨*:evID, eso:possession-owner_2, dbp:Tata_Group*⟩
⟨*:evID, eso:possession-theme, :Jag_and_L._Rover*⟩
⟨*:evID, sem:hasTime, #tmxID*⟩

where *#tmxID* is the RDF representation of "August 26th, 2007".

According to ESO, the event *eso:Getting* has both *pre-* and *post-situation*, containing *eso:hasInPossession* and *eso:notHasInPossession* assertions, respectively. In the *pre-situation* of our example, Ford has in possession Jaguar and Land Rover, and Tata Group does not have in possession them; in the *post-situation*, the contrary holds. In the ESO model, such facts specific to a situation are stored in a RDF named graph identified by the URI of the situation. Therefore, the following set of *n*-tuples is generated:[4]

⟨*:evID, eso:hasPreSituation, :evID_pre*⟩
⟨*:evID_pre, rdf:type, eso:Situation*⟩
⟨*:evID_pre, sem:hasTime, #tmxID*⟩
⟨*dbp:Ford, eso:hasInPossession, :Jag_and_L._Rover, :evID_pre*⟩
⟨*dbp:Tata_Group, eso:notHasInPossession, :Jag_and_L._Rover, :evID_pre*⟩

⟨*:evID, eso:hasPostSituation, :evID_post*⟩
⟨*:evID_post, rdf:type, eso:Situation*⟩
⟨*:evID_post, sem:hasTime, #tmxID*⟩
⟨*dbp:Ford, eso:notHasInPossession, :Jag_and_L._Rover, :evID_post*⟩
⟨*dbp:Tata_Group, eso:hasInPossession, :Jag_and_L._Rover, :evID_post*⟩

In NewsReader, all the events and related information, instantiated according to the ESO metamodel, are stored, together with the original news article from where they were extracted in the KnowledgeStore [6], a scalable, fault-tolerant, and Semantic Web grounded storage system to jointly store, manage, retrieve, and query both structured and unstructured data.

3 Contextualized Knowledge Repositories

In the following we provide an informal summary of the definitions for the CKR framework: for a formal and detailed description and for complete examples, we refer to [5].

A CKR is a two layered structure: (1) the upper layer consists of a knowledge base $\mathfrak{G}$, called *global context*, containing (a) *meta-knowledge*, i.e. the structure and properties of contexts, and (b) *global* (context-independent) *object knowledge*, i.e., knowledge that applies to every context; (2) the lower layer consists of a set of *(local) contexts* that contain locally valid facts and can refer to what holds in other contexts. The intuitive structure of a CKR knowledge base is depicted in Figure 1: in the following we detail its formal components and interpretation.

Syntax. The meta-knowledge of a CKR is expressed in a DL language containing the elements that define the contextual structure: the *meta-vocabulary* Γ is a DL signature containing, in particular, the sets of symbols for *context names* $\mathbf{N}$, *module names* $\mathbf{M}$ and *context classes* $\mathbf{C}$, including the class of all contexts Ctx. Intuitively, modules represent pieces of knowledge specific to a context or a context class. The role mod defined on $\mathbf{N} \times \mathbf{M}$ expresses associations between contexts and their modules. The *meta-language* $\mathcal{L}_\Gamma$ of a CKR is a DL language over Γ.

The knowledge in contexts of a CKR is expressed via a DL language $\mathcal{L}_\Sigma$, called *object-language*, based on an object-vocabulary Σ. The expressions of the object language are evaluated locally to each context, i.e., contexts can interpret each symbol

---

[4] Whenever applicable, default named graph is omitted.

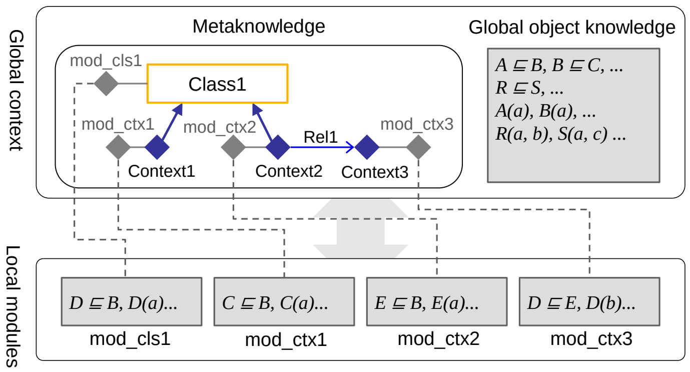

Fig. 1. CKR structure

independently. To access the interpretation of expressions inside a specific context or context class, we extend $\mathcal{L}_\Sigma$ to $\mathcal{L}^e_\Sigma$ with *eval expressions* of the form $eval(X, \mathsf{C})$, where X is a concept or role expression of $\mathcal{L}_\Sigma$ and C is a concept expression of $\mathcal{L}_\Gamma$ (with $\mathsf{C} \sqsubseteq \mathsf{Ctx}$). Intuitively, $eval(X, \mathsf{C})$ can be read as "the interpretation of X in all the contexts of type C".

We define a *Contextualized Knowledge Repository (CKR)* as a structure $\mathfrak{K} = \langle \mathfrak{G}, \{\mathrm{K}_\mathsf{m}\}_{\mathsf{m} \in \mathbf{M}} \rangle$ where: (i) $\mathfrak{G}$ is a DL knowledge base over $\mathcal{L}_\Gamma \cup \mathcal{L}_\Sigma$; (ii) every K_m is a DL knowledge base over $\mathcal{L}^e_\Sigma$, for each module name $\mathsf{m} \in \mathbf{M}$. We note that the knowledge in a CKR can be expressed by means of any DL language: in our work, we consider $\mathcal{SROIQ}$-RL [5] as language of reference. $\mathcal{SROIQ}$-RL is a restriction of $\mathcal{SROIQ}$ syntax corresponding to OWL RL [10].

Semantics. The semantics of CKR basically extends the usual model-based semantics of DL knowledge bases to the two layered structure of the framework. A *CKR interpretation* is a structure $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t.: (i) $\mathcal{M}$ is a DL interpretation of $\Gamma \cup \Sigma$ (respecting the intuitive interpretation of Ctx as the class of all contexts); (ii) for every context $x \in \mathsf{Ctx}^\mathcal{M}$, $\mathcal{I}(x)$ is a DL interpretation over Σ (with same domain and interpretation of individual names of $\mathcal{M}$). The interpretation of ordinary DL expressions on $\mathcal{M}$ and $\mathcal{I}(x)$ is as usual while *eval* expressions are interpreted as follows: for every $x \in \mathsf{Ctx}^\mathcal{M}$, $eval(X, \mathsf{C})^{\mathcal{I}(x)}$ represents the union of all elements in $X^{\mathcal{I}(\mathsf{e})}$ for all contexts e in $\mathsf{C}^\mathcal{M}$.

A CKR interpretation $\mathfrak{I}$ is a *CKR model* of $\mathfrak{K}$ iff: (i) for $\alpha \in \mathcal{L}_\Sigma \cup \mathcal{L}_\Gamma$ in $\mathfrak{G}$, $\mathcal{M} \models \alpha$; (ii) for $\langle x, y \rangle \in \mathsf{mod}^\mathcal{M}$ with $y = \mathsf{m}^\mathcal{M}$, $\mathcal{I}(x) \models \mathrm{K}_\mathsf{m}$; (iii) for $\alpha \in \mathfrak{G} \cap \mathcal{L}_\Sigma$ and $x \in \mathsf{Ctx}^\mathcal{M}$, $\mathcal{I}(x) \models \alpha$. Intuitively, this means that $\mathfrak{I}$ verifies the contents of global and local modules associated with contexts and global object knowledge has to be propagated to local contexts.

Materialization calculus. Reasoning in CKR has been formalized as a materialization calculus [8], a datalog-based calculus for instance checking in $\mathcal{SROIQ}$-RL CKRs.

Intuitively, the calculus is based on a translation to datalog of the input CKR. It has three components: (i) the *input translations* I_{glob}, I_{loc}, I_{rl}, where given an axiom α and $\mathsf{c} \in \mathbf{N}$, each $I(\alpha, \mathsf{c})$ is a set of datalog facts or rules encoding the contents of input global and local DL knowledge bases; (ii) the *deduction rules* P_{loc}, P_{rl}, which are sets of datalog rules representing the inference rules for the instance-level reasoning over the translated axioms; and (iii) the *output translation* O, where given an axiom α and

$\mathsf{c} \in \mathbf{N}$, $O(\alpha, \mathsf{c})$ is a single datalog fact encoding the ABox assertion α that we want to prove to be entailed by the input CKR (in the context c).

Intuitively, $\mathcal{SROIQ}$-RL input I_{rl} and deduction P_{rl} rules provide the translation and interpretation of $\mathcal{SROIQ}$-RL axioms from the input CKR. Global input rules in I_{glob} encode the interpretation of Ctx in the global context. Similarly, local input rules I_{loc} and deduction rules P_{loc} provide the translation and rules for the local *eval* expressions. The rules in O provide the translation of ABox assertions that can be verified to hold in a context c by applying the rules of the final program.

The translation of a CKR $\mathfrak{K}$ to its datalog program $PK(\mathfrak{K})$ proceeds in four steps: we first translate $\mathfrak{G}$ in the *global program* $PG(\mathfrak{G})$ by applying input rules I_{glob} and I_{rl} to $\mathfrak{G}$ and adding deduction rules P_{rl}; then, for every context name $\mathsf{c} \in \mathbf{N}$ appearing in $PG(\mathfrak{G})$, we compute its knowledge base K_{c} as the set of modules $\mathrm{K}_{\mathsf{m}} \in \mathfrak{K}$ s.t. $\mathsf{mod}(\mathsf{c}, \mathsf{m})$ is proved by $PG(\mathfrak{G})$; we translate each *local program* $PC(\mathsf{c})$ by applying input rules I_{loc} and I_{rl} to K_{c} and adding deduction rules P_{loc} and P_{rl}; the final *CKR program* $PK(\mathfrak{K})$ is then obtained as the union of $PG(\mathfrak{G})$ with all local programs $PC(\mathsf{c})$. We say that $\mathfrak{K}$ *entails* an axiom α in a context $\mathsf{c} \in \mathbf{N}$ if $PK(\mathfrak{K}) \models O(\alpha, \mathsf{c})$. We can show (see [5]) that the presented rules and translation process provide a sound and complete calculus for instance checking in $\mathcal{SROIQ}$-RL CKR.

CKR implementation on RDF. We recently presented a prototype [5] that implements the forward reasoning procedure over CKR defined by the materialization calculus. The prototype accepts RDF input data expressing OWL-RL axioms and assertions for global and local knowledge modules: these different pieces of knowledge are represented as distinct named graphs, while we encoded in a OWL vocabulary the CKR contextual primitives (e.g. the class Context of all context individuals, the class Module of all modules and the property hasModule corresponding to the role mod). The prototype is based on an extension of the Sesame RDF Framework[5] and structured in a client-server architecture: the main component, called *CKR core* and residing in the server-side part, offers the ability to compute and materialize the inference closure of the input CKR, add and remove knowledge and execute queries over the complete CKR structure.

The distribution of knowledge in different named graphs asks for a component to compute inference over multiple graphs in a RDF store, since inference mechanisms in current stores usually ignore the graph part. This component has been realized as a general software layer called *SPRINGLES* (*SParql-based Rule Inference over Named Graphs Layer Extending Sesame*) [5]. Intuitively, SPRINGLES provides methods to demand a closure materialization on the RDF store data: rules are encoded as (named graphs aware) SPARQL queries and it is possible to customize both the used ruleset and the evaluation strategy.

In our case, the ruleset basically encodes the rules of the presented materialization calculus. The rules are evaluated with a strategy that follows the same steps of the translation process defined for the calculus. The plan goes as follows: (i) we compute the inference closure on the graph for global context $\mathfrak{G}$, by a fixpoint on rules corresponding to P_{rl}; (ii) we derive associations between contexts and their modules, by adding dependencies for every assertion of the kind $\mathsf{hasModule}(\mathsf{c}, \mathsf{m})$ in the global closure; (iii) we compute the closure of the contexts, by applying rules encoded from P_{rl} and P_{loc} and resolving *eval* expressions by the metaknowledge information in the global closure.

---

[5] `http://www.openrdf.org/`

4 Representing events in CKR: CKR-ESO ontology

We can now describe how we translated and implemented a first prototype of the ESO model in the form of a contextualized ontology for the CKR, that we call the *CKR-ESO ontology*.

In this model, the event and situation structures are modelled in the metaknowledge. Similarly to the ESO model, each event instance is associated in the metaknowledge with its pre-, during- and post-situations using the object properties hasPreSituation, hasPostSituation and hasDuringSituation, subproperties of hasSituation. Situation elements associated with events can be generated automatically by SPRINGLES rules when importing an event.

Each event is represented in the metaknowledge as an instance of the class Event: in particular, each event is associated, analogously to the ESO model, with a subclass of the Event class that determines the type of associated situations: in particular, DynamicEvents (e.g. *ChangeOfPossession*, *Constructing*) are typically characterized by their pre- and post-situations, while StaticEvents (e.g. *BeingOperational*) by their during-situations.

This classification is provided by restrictions over the definition of such classes. For example, for the ChangeOfPossession event class, the CKR-ESO ontology states that:

$$\mathsf{ChangeOfPossession} \sqsubseteq \forall \mathsf{hasPreSituation.Pre\_ChangeOfPossession}$$
$$\mathsf{ChangeOfPossession} \sqsubseteq \forall \mathsf{hasPostSituation.Post\_ChangeOfPossession}$$

Each event individual is associated with a knowledge module that corresponds to the RDF graph of the event in the ESO model. This association is represented in the metaknowledge by the property hasEventModule. The following chain axiom is defined over this property, asserting that situations related to an event inherit the facts asserted in the event module: $(\mathsf{hasSituation})^- \circ \mathsf{hasEventModule} \sqsubseteq \mathsf{hasModule}$. As defined by the ESO model, we expect to find in the event module the instantiation for all the required roles involved in the event.

The class Situation is defined as a subclass of the Context class in the CKR vocabulary: in other words, in our model we consider situations and their local knowledge as contexts. The particular (pre, post and during) situations associated with event types are modelled by specific context classes. Thus, for example, we have that the pre- and post-situations for events of type ChangeOfPossession are classified as members of the classes Pre_ChangeOfPossession and Post_ChangeOfPossession. The association between such type of situations and their local axioms (i.e. what its modelled in the ESO ontology by situation assertions) is performed by linking specific knowledge modules to these context classes. For example, in CKR-ESO we declare that every pre-situation of ChangeOfPossession is associated with the knowledge module pre_change-of-possession-m and post-situations to post_change-of-possession-m:

$$\mathsf{Pre\_ChangeOfPossession} \sqsubseteq \exists \mathsf{hasModule}.\{\mathsf{pre\_change\text{-}of\text{-}possession\text{-}m}\}$$
$$\mathsf{Post\_ChangeOfPossession} \sqsubseteq \exists \mathsf{hasModule}.\{\mathsf{post\_change\text{-}of\text{-}possession\text{-}m}\}$$

Situation assertions are thus encoded inside these specific modules: the assertions can be basically translated to chain axioms across the roles specified in the event. For example, assertions for pre-situations of ChangeOfPossession stating that:

$$hasInPossession(possession\text{-}owner\_1, possession\text{-}theme)$$
$$notHasInPossession(possession\text{-}owner\_2, possession\text{-}theme)$$

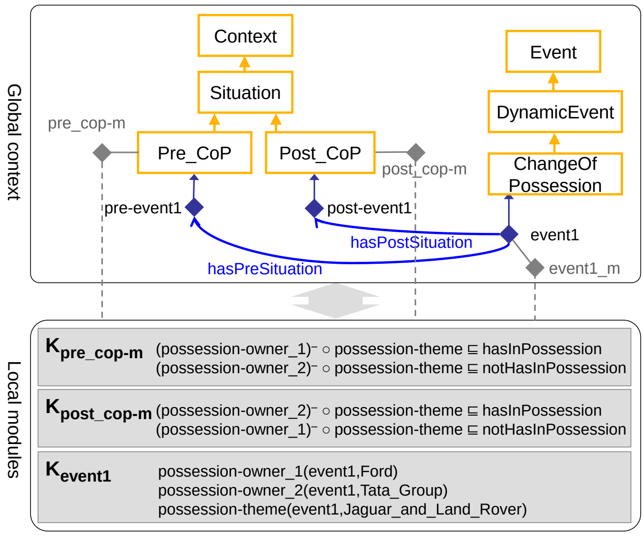

Fig. 2. Example event in CKR-ESO model.

is translated in CKR-ESO to these chain axioms across role properties:

$$
\begin{array}{l}
(possession\text{-}owner\_1)^{-} \circ possession\text{-}theme \sqsubseteq hasInPossession \\
(possession\text{-}owner\_2)^{-} \circ possession\text{-}theme \sqsubseteq notHasInPossession
\end{array}
$$

We now can show how to represent our example event from Section 2 using the CKR-ESO model: we depict this modelling in Figure 2. In the global context, we define event1 to be of type ChangeOfPossession and associate it with its situations:

$$
\mathsf{ChangeOfPossession}(\mathsf{event1})
$$

$$
\begin{array}{l}
\mathsf{hasPreSituation}(\mathsf{event1}, \mathsf{pre\text{-}event1}) \\
\mathsf{hasPostSituation}(\mathsf{event1}, \mathsf{post\text{-}event1})
\end{array}
$$

By the above axioms for such event type, we know that the pre- and post-situations of event1 have to be of type Pre_ChangeOfPossession and Post_ChangeOfPossession. By metalevel reasoning, this implies that:

$$
\begin{array}{l}
\mathsf{hasModule}(\mathsf{pre\text{-}event1}, \mathsf{pre\_change\text{-}of\text{-}possession\text{-}m}) \\
\mathsf{hasModule}(\mathsf{post\text{-}event1}, \mathsf{post\_change\text{-}of\text{-}possession\text{-}m})
\end{array}
$$

and thus the situation assertions associated with the pre- and post-situations of ChangeOfPossession are imported in the two situations[6]. Moreover, the graph associated with the original event is now defined as a module associated with event1 in the metalevel: hasEventModule(event1, event1_m). The event1_m module (i.e. the associated RDF graph) now contains the following facts that are shared with all the situations associated with this event:

$$
\begin{array}{l}
possession\text{-}owner\_1(\mathsf{event1}, Ford) \\
possession\text{-}owner\_2(\mathsf{event1}, Tata\_Group) \\
possession\text{-}theme(\mathsf{event1}, Jaguar\_and\_Land\_Rover)
\end{array}
$$

---

[6] In Figure 2 we abbreviate classes of pre- and post-situations of ChangeOfPossession with Pre_CoP and Post_CoP and their modules with pre_cop-m and post_cop-m.

Using the situation assertions in the module associated with the pre-situation, the CKR thus derives the following facts in the context of pre-event1:

$$\begin{array}{l} hasInPossession(Ford, Jaguar\_and\_Land\_Rover) \\ notHasInPossession(Tata\_Group, Jaguar\_and\_Land\_Rover) \end{array}$$

Similarly, in the context of post-event1 we obtain:

$$\begin{array}{l} notHasInPossession(Ford, Jaguar\_and\_Land\_Rover) \\ hasInPossession(Tata\_Group, Jaguar\_and\_Land\_Rover) \end{array}$$

We note that representation of the described event can be completed with its associated during-situation: among the facts that are known to hold during the event, for example, we can assert the existence of the actors and of the possession theme (using the $exists$ property in the ESO).

This contextual re-interpretation of the ESO model can bring several advantages from the point of view of reasoning capabilities inside and across events. First of all, every aspect of the reasoning procedure is now strictly ruled by logical reasoning: situation assertions and their association with the type of situation are now directly modelled by the CKR structure and local axioms, without demanding an external reasoner to consider the situation rules and the local reasoning inside situations. Furthermore, the propagation of global object knowledge to local knowledge allows the use of context independent background knowledge in local reasoning. In our example, we can assert in the global knowledge that both actors ($Ford$ and $Tata\_Group$) are classified as car companies and their features can be used in local reasoning. More in general, the advantages of an explicit and structured representation of contexts (as the one offered by the CKR) with respect to a modelling based on reification have been shown in [2].

On the other hand, the clear separation of meta and object level reasoning can be exploited to exchange information across the two levels. For example, depending on the type and specific patterns of situation and events, by adding custom SPRINGLES rules it is possible to generate implicit events that have to occur for the completion of the event sequence in a story. In our example, if we have a second event event2 representing another ChangeOfPossession of $Jaguar\_and\_Land\_Rover$ between two companies $Company1$ and $Company2$, different than the two companies from event1, and event2 has a timestamp greater than event1, then we can infer that there have been another two events (possibly being the same one): in one $Jaguar\_and\_Land\_Rover$ has been sold from $Tata\_Group$ and in the other it has been acquired from $Company1$. Similarly, we can recognize cases in which we can assert the equality of certain situations: this can be used to compile sequences of events in a story.

Metalevel information for situations and events can be derived from local reasoning: we might recognize incompatible descriptions of the same event from different news. For example, let us suppose a different representation of the scenario shown in the example in Figure 2: assume that event1 is now classified as Buying (subclass of ChangeOfPossession) while another event event2 is extracted as Selling (also subclass of ChangeOfPossession), but they both represent the same conceptual event (i.e. the acquisition of $Jaguar\_and\_Land\_Rover$ by $Tata\_Group$ from $Ford$). Thus, at the level of the metaknowledge, the two events are modelled as:

Buying(event1)
hasPreSituation(event1, pre-event1)
hasPostSituation(event1, post-event1)

Selling(event2)
hasPreSituation(event2, pre-event2)
hasPostSituation(event2, post-event2)

Since they represent the same happening, the event modules event1_m and event2_m basically share the same contents: that is, the actors are the same and they take the same role. However, suppose that, due to the extraction from different news sources, the metamodel property *sem:hasTime* associated to post-event1 has value *"August 26th, 2007"* while the value associated to pre-event2 is *"August 28th, 2007"*. Then, using this metalevel information and the local contents of the event modules, we can easily write a reasoning rule that recognizes that the two events are incompatible and adds the assertion event1 incompatibleWith event2 in the global context. Similarly, we can recognize inconsistent situations by local reasoning: this can be used both to exclude further inferences from inconsistent contexts, by marking as "inconsistent" the situation individual in the metaknowledge, but also to repair (possibly with some ad-hoc rules) the local axioms. We note that, on the other hand, this kind of reasoning requires to define ad-hoc rules to recognize such different situations.

Another interesting possibility is the one of having inter-situation knowledge propagation. For example, if two situations or two events are recognized as consequent in a story, unmodified knowledge from the previous situations can be propagated to subsequent situations (e.g. the marital status of Obama did not change when he was elected US president). This clearly presents problems of non-monotonicity, since one has to consider which knowledge can be seamlessly propagated without incurring in contradictory states. In this regard, we recently introduced in CKR a notion of *defeasible axioms* and their overriding across different contexts [3].

5 Conclusions and future works

In this paper we introduced the model of the CKR-ESO ontology, a re-interpretation of the Event and Situation Ontology under the contextual view of knowledge offered by the CKR framework. We discussed informally the advantages of such representation and demonstrated its application by means of an example.

We are currently completing the translation of the full ESO ontology to its contextualized version: we remark that, given the direct translation across the two models, we can easily automatize this transformation.

Our goal is to be able to apply some of the proposed complex reasoning services to the events currently represented in the KnowledgeStore of the NewsReader project: to this aim, we plan to integrate the RDF-based CKR implementation with the KnowledgeStore and encode such reasoning services with respect to CKR contextual model.

Acknowledgments. The research leading to this paper was supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

References

1. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labelling. In: Proceedings of CoNLL-2009. Boulder, CO, USA (2009)
2. Bozzato, L., Ghidini, C., Serafini, L.: Comparing contextual and flat representations of knowledge: a concrete case about football data. In: K-CAP 2013. pp. 9–16. ACM (2013)
3. Bozzato, L., Eiter, T., Serafini, L.: Contextualized Knowledge Repositories with Justifiable Exceptions. In: DL2014. CEUR-WP, vol. 1193, pp. 112–123. CEUR-WS.org (2014)
4. Bozzato, L., Homola, M., Serafini, L.: Towards More Effective Tableaux Reasoning for CKR. In: DL2012. CEUR-WP, vol. 824, pp. 114–124. CEUR-WS.org (2012)
5. Bozzato, L., Serafini, L.: Materialization Calculus for Contexts in the Semantic Web. In: DL2013. CEUR-WP, vol. 1014. CEUR-WS.org (2013)
6. Corcoglioniti, F., Rospocher, M., Cattoni, R., Magnini, B., Serafini, L.: Interlinking unstructured and structured knowledge in an integrated framework. In: Proc. of 7th IEEE International Conference on Semantic Computing (ICSC), Irvine, CA, USA (2013), (to appear)
7. Das, D., Schneider, N., Chen, D., Smith, N.: Probabilistic frame-semantic parsing. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. HLT'10 (2010)
8. Krötzsch, M.: Efficient inferencing for OWL EL. In: JELIA 2010. Lecture Notes in Computer Science, vol. 6341, pp. 234–246. Springer (2010)
9. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: DOLCE: a descriptive ontology for linguistic and cognitive engineering. WonderWeb Project, Deliverable D17 v2 1 (2003)
10. Motik, B., Fokoue, A., Horrocks, I., Wu, Z., Lutz, C., Grau, B.C.: OWL 2 Web Ontology Language Profiles. W3C recommendation, W3C (Oct 2009), http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/
11. Segers, R., Vossen, P., Rospocher, M., Serafini, L., Laparra, E., Rigau, G.: ESO: a Frame based Ontology for Events and Implied Situations. In: Proceedings of the MAPLEX 2015 Workshop (2015)
12. Serafini, L., Homola, M.: Contextualized knowledge repositories for the semantic web. J. of Web Semantics 12 (2012)