CKR:Live Demo: Representing an Evolving Scenario by Contexts and Exceptions

Loris Bozzato, Luciano Serafini, and Gaetano Calabrese

Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy

{bozzato,serafini,calabrese}@fbk.eu

Abstract. Contextualization of knowledge has recently gained interest in the Semantic Web community and a number of logic based solutions have been proposed. In this regard, in our previous works we have introduced the Contextualized Knowledge Repository (CKR) framework: we recently extended CKR with the possibility to reason with defeasible axioms and their exceptions.
The goal of this demo is to demonstrate the use of contexts and exceptions in representing evolving situations: the demo visualizes the evolution of a soccer match by showing the (non-monotonic) changes in the knowledge across different events composing the match.

1 Introduction

The interest in context representation applied to Semantic Web data has led to the proposal of a number of logic based solutions in recent years, like e.g. [4,5,6]. In this direction, in our previous works we have introduced the DL based *Contextualized Knowledge Repository (CKR) framework* [5,2,3]. In the latest formulation of CKR presented in [1], we introduced an extension of the framework with the possibility to reason with defeasible axioms and their *justifiable exceptions* in context dependent axioms.

In this demo, we demonstrate the use of contexts and local exceptions of the CKR framework in representing a scenario evolving across several subsequent knowledge states. We chose to model the example of the evolution of a soccer match: this allows us to expose the potentialities of our framework inside a simple and well-understood scenario, which, on the other hand, provides a rich set of evolving information.

Goal of this demo is to show in practice the use of contexts and their relations for interpretation of local knowledge and its propagation in an event structure. Moreover, we show how a combination of knowledge import axioms and exceptions allows for modelling a non-monotonic evolution of knowledge across states. From the point of view of the implementation, the architecture of the demo system demonstrates how to use the *CKR datalog rewriter* (CKR*ew*) and the reasoning implemented by its datalog translation [1] to compute inferences on an input CKR with defeasibility and integrate this reasoning in a pipeline for managing contextualized OWL/RDF data.

2 CKR with Justifiable Exceptions

We provide an informal summary of the elements of the CKR framework with justifiable exceptions: for a formal description and complete definitions, we refer to [3,1]. 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 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 meta-knowledge of a CKR is expressed in a DL *meta-language* $\mathcal{L}_\Gamma$ containing the elements that define the contextual structure: for example, it contains sets of symbols for *context names*, *context classes* (i.e. named classes of contexts) and *module names* (i.e. pieces of local knowledge associated to a single context or a context class).

The knowledge inside contexts of a CKR is expressed via a DL *object-language* $\mathcal{L}_\Sigma$. The expressions of object language are evaluated locally to each context: every context can interpret each symbol independently. To access the interpretation of symbol X inside a specific context or context class C, we allow in local object knowledge *eval expressions* of the form *eval*(X, C), which can be read as *"the interpretation of* X *in all the contexts of type* C*"*. The global context $\mathfrak{G}$ can contain statements $\mathrm{D}(\alpha)$ with $\alpha \in \mathcal{L}_\Sigma$: this states that α is a *defeasible axiom*. Intuitively, in the local contexts, α is applied to all its local instantiations except for those generating a contradiction, i.e. a local "overriding" for some instances of α is tolerated. E.g., $\mathrm{D}(A \sqsubseteq B) \in \mathfrak{G}$ means *"in general, every* A *is a* B*"*: in a local context, the axiom might be contradicted by assertions $\{A(e), \neg B(e)\}$ that override the axiom for the "exceptional" individual e.

Reasoning in CKRs with defeasibility has been formalized in the form of a translation to datalog: the resulting datalog program is interpreted under answer set semantics, which provides the non-monotonic reading needed for the interpretation of defeasible axioms and exceptions. The datalog translation has been implemented in a *CKR datalog rewriter* (CKR*ew*) prototype, available at `http://ckrew.fbk.eu/`.

3 CKR:Live Demo System Architecture

The demo system is composed by two components: a *compiler*, which prepares the input RDF files and computes the inferences, and an *interface*, which visualizes the results of the inferences in terms of the presented demo scenario.

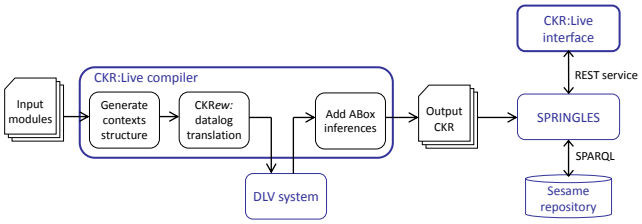

Fig. 1. System architecture and compilation process

The *compiler* first loads a set of RDF files containing the ABox information for the global and local modules (i.e. the global and event specific knowledge for the considered soccer match). It then generates the CKR structure by linking each input module to its intended context and adds a set of fixed and context dependent TBox axioms defining the scenario (see next section) to local contexts. The CKR so generated is then passed to CKR*ew*, which computes its datalog translation. By interfacing with the *DLV solver*[1],

[1] `http://www.dlvsystem.com/dlv/`

the answer sets of the datalog translation are computed: the newly inferred ABox facts are then extracted from the models and added to the generated CKR. Finally, the CKR is saved to RDF files: these are loaded via *SPRINGLES* [3] as a single Sesame repository.

The CKR:Live demo web *interface* uses a REST interface to communicate with the SPRINGLES platform: each update to the web interface is reflected by a SPARQL query (targeting the currently visualized context) implemented as a REST service. As we present in Section 5, the web interface provides an intuitive visualization of the evolution across the states of the represented system: it gives the possibility to select specific events and visualize the changes in their associated knowledge.

4 Scenario Description: Evolution of a Soccer Match

In the example scenario presented in this demo, our goal is to reason on the evolution of knowledge in the development of a soccer match. In fact, while some information is stable during the match (e.g. the number and playing position of a player), some other changes during the *events* taking place in the course of the match (e.g. the number of yellow cards assigned to a player). Moreover, most of the information increases monotonically (e.g. scored goals), but some other knowledge can be retracted across events: in our example, this is the case of player substitutions, in which a player previously in game is exchanged by a team mate previously waiting at the bench.

We can now show how we represented this scenario using a CKR. The stable information is contained as facts in the global context as global object knowledge: this consists of the name of the host and home teams and the name, position, team membership and number of each player. Then, local contexts represent events happening during the match: we consider goals, substitutions, card bookings, kick-off, half-time and end of the match. Events are ordered by their time of happening in the match: they represent the match evolution as a discrete succession of states. In the local knowledge of each event, the new information added by the happening is specified as ABox assertions: e.g., in the case of a goal, the scoring player and team are recorded. In particular, in the first state s_0 (kick-off), we store information about the initial formations by classifying each $Player$ as $PlayingNow$ (in game) or $notPlayingNow$ (in the reserves).

The local information of a state needs to preserve the (unchanged) information of its predecessor state: this is achieved by means of *eval* axioms. In the case of monotonic information, using *eval* we can "import" the instances of a predicate from the previous state: for example, in context s_2 we import previous goals using the axiom $\mathit{eval}(GoalNow, \{s_1\}) \sqsubseteq GoalNow$. In the case of non-monotonic information (i.e. substitutions), we use defeasible axioms: in the global context, we include the axioms (1) $\mathrm{D}(Player \sqsubseteq Unchanged)$, i.e. *"Players are generally unchanged (from previous state)"*; (2) $PlayingBefore \sqcap Unchanged \sqsubseteq PlayingNow$, i.e. *"If a player is unchanged and he was playing before, then he still plays now"*. In local contexts, knowledge about $PlayingBefore$ is obtained again using *eval* axioms: e.g., in state s_2 we can specify that $\mathit{eval}(PlayingNow, \{s_1\}) \sqsubseteq PlayingBefore$. In contexts representing substitutions, we can create an exception to the defeasible axiom by asserting that the substituted player is $Changed$ (with $Changed \sqcap Unchanged \sqsubseteq \bot$) and $notPlayingNow$: the first assertion defines an exception to (1) which is then not applied to the player, thus negating the local application of (2). The same mechanism of "defeasible propagation" is used for $notPlayingNow$ instances (i.e. players currently at the bench).

5 Online Demo Overview

An online version of the demo is available at `http://bit.ly/ckrlive17`.

During the demo we will demonstrate the usage of the system on two example matches: *Brazil vs. Germany*, semi-final of 2014 FIFA World Cup (`http://bit.ly/BRAvsGER-FIFA2014`), and *Italy vs. Spain*, from round of 16 of UEFA Euro 2016 (`http://bit.ly/ITAvsSPA-UEFA2016`). We chose these matches as they provide a fair number of events and event types. Their information has been extracted from official websites of the competitions.

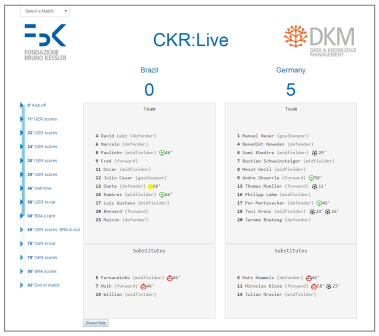

Fig. 2. CKR:Live demo web interface

Using the menu at the top of the page, the user can choose one of the matches. At the left of the page, a slider gives the possibility to select a state and scroll across the events composing the match: a short descriptive label is associated to each event marker. At the center of the page, all of the knowledge associated to the current state is visualized: it is shown the current score for each team and the team formations (divided by currently in-game players and substitutes). Each player is listed with his number, name and position: whenever he scores a goal, receives a yellow/red card or enters/exits the field, an icon and time of happening appear besides its name.

By moving the slider across states, it is possible to note the changes in knowledge contents at each event. E.g. considering the first match (*Brazil vs. Germany*): at the initial state, the initial team formation is shown. The first event of the match is a goal: one can note that its information is (monotonically) preserved when passing to next states. On the other hand, in the event of a substitution, previously in-game players are replaced by reserve players: this is the case of the *half-time* event in the shown match, in which both teams had multiple player substitutions. If a player has not been involved in a substitution, then its position is preserved to the subsequent states. By clicking on the *Show/Hide* button, it is possible to visualize the underlying data retrieved from the CKR repository: a text box shows the results (in JSON) of the call to the REST service, corresponding to a SPARQL query about players information in the current context.

References

1. 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)
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., Serafini, L.: Materialization Calculus for Contexts in the Semantic Web. In: DL2013. pp. 552 – 572. CEUR-WP, CEUR-WS.org (2013)
4. Klarman, S.: Reasoning with Contexts in Description Logics. Ph.D. thesis, Free University of Amsterdam (2013)
5. Serafini, L., Homola, M.: Contextualized knowledge repositories for the semantic web. J. of Web Semantics 12, 64–87 (2012)
6. Straccia, U., Lopes, N., Lukácsy, G., Polleres, A.: A general framework for representing and reasoning with annotated semantic web data. In: AAAI 2010. AAAI Press (July 2010)