Poster

Implementing Usage Control Policies Using Reification with RDF-Star and SPARQL-Star

Ines Akaichi[1,*], Giorgos Flouris[2], Irini Fundulaki[2] and Sabrina Kirrane[1]

[1]Institute for Complex Networks, WU, Vienna, Austrian
[2]Institute of Computer Science, FORTH, Heraklion, Greece

Abstract

In decentralized environments usage control (UC) is crucial for governing asset and resource usage. In an earlier work, we proposed GUCON, a generic graph pattern based policy framework for UC enforcement, which offers a formal semantics for specifying and reasoning over UC policies. Although GUCON caters for the expression of various rules using graph patterns, the incorporation of temporal constraints is cumbersome. In this paper, we propose an instantiation of the GUCON framework that demonstrates how RDF-star can be used for rule representation and SPARQL-star for compliance checking. Additionally, we extend the original policy language to cater for temporal constraints.

Keywords

RDF-star, SPARQL-star, Usage Control, Policies

1. Introduction

In modern decentralized systems, such as the *Internet of Things (IoT)* and *data spaces*, usage control becomes crucial for managing assets and resources [1]. In previous work [2], we proposed the *generic graph pattern-based policy framework for usage control enforcement* (GUCON), which provides an abstract structure with formal semantics for UC policy specification and includes algorithms for policy-specific reasoning tasks such as compliance, requirements, and consistency checking. GUCON uses conditional deontic rules based on graph patterns and deontic concepts (permission **A**, prohibition **P**, obligation **O**, and dispensation **D**), with formal semantics rooted in graph patterns [3]. It introduces a *Knowledge Base* (KB) to capture domain knowledge and events, serving as the foundation for reasoning about and enforcing UC policies. An essential aspect of UC is expressing and reasoning over environmental and system constraints (e.g., temporal, spatial, purpose, events) [4, 5]. Although GUCON can express various conditions on data usage through graph patterns, adding constraints such as temporal constraints can make the expressions cumbersome. To address this, we propose using reification to express constraints as part of UC rule conditions. Specifically, we use RDF-star for rule expression and

ISWC 2024 Posters and Demos: 23rd International Semantic Web Conference, November 11–15, 2024, Maryland, USA
*Corresponding author.
ines.akaichi@wu.ac.at (I. Akaichi); fgeo@ics.forth.gr (G. Flouris); fundul@ics.forth.gr (I. Fundulaki); sabrina.kirrane@wu.ac.at (S. Kirrane)
0000-0002-6020-5572 (I. Akaichi); 0000-0002-8937-4118 (G. Flouris); 0000-0002-4812-9896 (I. Fundulaki); 0000-0002-6955-7718 (S. Kirrane)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

```
{
?share
    rdf:type splog:SharingEvent ;
    rdf:subject ?x ;
    rdf:predicate eg:share ;
    rdf:object egmc:l.premierleague.com-2020-coverage ;
    splog:occurs  ?t .

?x
    rdf:type eg:Person ;
    eg:position eg:Journalist .

FILTER (?t > "2021-02-25T21:00:00Z"^^xsd:dateTime)
}
->
{
A {?x  :share  egmc:l.premierleague.com-2020-coverage }
}
```

Listing 1: Policy Representation using GUCON.

SPARQL-star for our compliance checker. We illustrate the application of our solution with a football embargo use case. In this scenario, media outlets are permitted to share specific extracts from a football match, such as highlights, photographs, and interviews, only after a specified time or event (e.g., the end of the match). To model this information, we developed an ontology based on the Sport Schema ontology[1], which describes a sports event as part of a competition structure (e.g., league, championship), and the ontology for media resources[2], which details media coverage (e.g., photos, interviews) as part of an event. In this paper, we use the 2020 English Premier League competition as a recurring example.

2. Related Work

RDF reification is the standard way to express metadata about RDF triples. Reification involves converting a basic triple to an `rdf:Statement`, that describes the original triple as a resource. While RDF reification is a standard approach, it can be verbose and cumbersome [6]. Alternative approaches include the use of named graphs, or RDF-Star[3]. When it comes to UC specifically, Robaldo [7] proposed to use reified I/O logic to express norms and make use of representation languages, such as SHACL, to implement compliance checking [8]. While their work highlights the significance of reification in compliance checking, it primarily focuses on normative reasoning. Additionally, expressing environmental or system constraints is outside of their focus. In this paper, due to their enhanced expressivity and efficiency, we propose to use RDF-star to express constraints on top of UC rules and SPARQL-star to build a compliance checking algorithm on top of GUCON.

[1]IPTC Sport Schema, https://sportschema.org/schema-overview/
[2]Media Resources 1.0, https://www.w3.org/TR/mediaont-10/
[3]RDF-star and SPARQL-star, https://w3c.github.io/rdf-star/cg-spec/editors_draft.html

```
{
 <<?x eg:share egmc:l.premierleague.com-2020-coverage>>   splog:occurs  ?t .

 ?x rdf:type eg:Person;
    eg:position eg:Journalist .

 FILTER (?t > "2021-02-25T21:00:00Z"^^xsd:dateTime )
}
->
{
A {?x eg:share egmc:l.premierleague.com-2020-coverage }
}
```

Listing 2: Policy Representation using GUCON-star.

3. Using Reification to Model Usage Control Policies

A KB is an RDF graph describing the set of actual knowledge. A GUCON rule is of the form: $cond \rightsquigarrow da$, where *cond* is a SPARQL graph pattern, and *da* is a deontic pattern, where $d \in \{\mathbf{A}, \mathbf{P}, \mathbf{O}, \mathbf{D}\}$, and *a* is a called an action pattern. An action pattern is a SPARQL triple pattern where the subject, property, and object refer to an entity name, an action name, and a resource name, respectively, with variables present in all three positions. A UCR can be read as follows: if the condition (*cond*) is satisfied by the KB, then the deontic pattern (*da*) may (**A**), must not (**P**), must (**O**), or need not (**D**) be satisfied. In Listing 1, we express the permission rule from our use case using GUCON (we assume that 2021-02-25T21:00:00 implies the end of the match).

In this paper, we propose to extend GUCON with reification in order to express constraints on top of the rules, also called GUCON-Star. In particular, the action pattern representing the action of sharing data is reified by using a *quoted triple*. This quoted triple can then be used in the body of the rule as a way to describe constraints, in our case, a temporal constraint. The same rule from Listing 1 is expressed using GUCON-Star in Listing 2.

4. Compliance Checking in RDF-star/ SPARQL-star

The original GUCON compliance algorithm from [2] checks for only obligation and prohibition compliance, assuming permissions and dispensations are always valid. With the addition of constraints, rule semantics change, such as temporal permissions becoming valid only after a specific time. We extend our compliance checker to evaluate permissions, prohibitions, obligations, and dispensations with constraints. Below, we present a KB extract containing logs from a system tracking media coverage sharing for football matches. The KB is described using the SPECIAL Policy Log Vocabulary[4]. The log shows `eg:JohnSmith` sharing a media resource from the English Premier League 2020 Season, including details like occurrence time and description. More details describing the logs can be found in our GitHub[5]. The compliance checker uses a UC rule and a KB as input. It determines rule condition matches via ASK SPARQL-star queries over the KB, deciding compliance accordingly. For temporal permissions

[4]The SPECIAL Policy Log Vocabulary, https://ai.wu.ac.at/policies/policylog/
[5]https://github.com/Ines-Akaichi/GUCON-star/tree/main

```
eg:JohnSmith
    rdf:type eg:Person ;
    eg:position eg:Journalist .

eg:JohnSmith   eg:share  egmc:l.premierleague.com-2020-coverage .

eg:log20210225T203050
    rdf:type splog:Log ;
    dct:title  "Log of tracking application of media sharing of the premier league
    match"@en ;
    dct:description "This contains all traces of  media coverage sharing"@en ;
    dct:issued  "2021-02-25T20:30:50Z"^^xsd:dateTimeStamp  ;
    prov:wasAttributedTo  eg:TrackingSystemR2D2  ;
    splog:event <<eg:JohnSmith  eg:share  egmc:l.premierleague.com-2020-coverage >> .

<<eg:JohnSmith  eg:share  egmc:l.premierleague.com-2020-coverage >>
    rdf:type   splog:SharingEvent ;
    dct:description          "We tracked a new sharing of match materials"@en ;
    splog:instanceData  egmc:l.premierleague.com-2020-coverage  ;
    splog:occurs  "2021-02-25T20:30:50Z"^^xsd:dateTime .
```

Listing 3: Extract of a Log.

and dispensations, the engine checks whether the KB contains a match for the UC rule's body that satisfies the temporal constraint; if so, the KB is non-compliant. For obligations and prohibitions, a match means the KB is compliant for obligations and non-compliant for prohibitions. The sharing that occured at "2021-02-25T20:30:50Z", makes `eg:JohnSmith` non compliant with the rule defined in Listing 2. The compliance checker and inputs are available on our GitHub page.

5. Discussion and Future Work

In this poster, our focus was on demonstrating how RDF-star and SPARQL-star syntax can be utilized to serialize GUCON policies with various constraints. As an important next step, we aim to define the semantics of this new serialization so that future implementations of compliance engines are able of consistently and accurately enforcing GUCON UCPs. Furthermore, our compliance checker primarily handles temporal constraints and filter operators. We aim to expand its functionality to accommodate other types of constraints. Given that UC involves managing dynamic policies, temporal constraints can evolve, such as when a match ends, thus we are interested in exploring the mutability of constraints within the context of GUCON. Additionally, due to the absence of benchmarking in UC [9], we plan to represent GUCON using various representation languages such as Datalog, ASP, etc. This will allow us to compare the performance of different engines.

References

[1] C. Jung, J. Dörr, Data Usage Control, 2022.

[2] I. Akaichi, G. Flouris, I. Fundulaki, S. Kirrane, Gucon: A generic graph pattern based policy

framework for usage control enforcement, in: Rules and Reasoning: 7th International Joint Conference, RuleML+RR 2023, Oslo, Norway, September 18–20, 2023, Proceedings, 2023. URL: https://doi.org/10.1007/978-3-031-45072-3_3.
[3] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of sparql, in: The Semantic Web - ISWC, 2006.
[4] M. Hilty, D. Basin, A. Pretschner, LNCS 3679 - On Obligations, Technical Report, 2005.
[5] M. Hilty, A. Pretschner, D. Basin, C. Schaefer, T. Walter, A policy language for distributed usage control, in: Computer Security – ESORICS, 2007.
[6] O. Hartig, Foundations of rdf* and sparql* (an alternative approach to statement-level metadata in rdf), in: Alberto Mendelzon Workshop on Foundations of Data Management, 2017. URL: https://api.semanticscholar.org/CorpusID:2010537.
[7] L. Robaldo, Towards compliance checking in reified I/O logic via SHACL, Proceedings of the 18th International Conference on Artificial Intelligence and Law, ICAIL 2021 (2021).
[8] L. Robaldo, S. Batsakis, R. Calegari, F. Calimeri, M. Fujita, G. Governatori, M. Morelli, F. Pacenza, G. Pisano, K. Satoh, I. Tachmazidis, J. Zangari, Compliance checking on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies, Artificial Intelligence and Law 32 (2023).
[9] I. Akaichi, S. Kirrane, Usage control specification, enforcement, and robustness: A survey, 2022. URL: https://arxiv.org/abs/2203.04800.