Ontology-Based Event Detection for Wastewater Treatment

Yi Sun, Zhe Wang and Kewen Wang[1]

[1] Griffith University, Brisbane, Australia
{yi.sun, zhe.wang, k.wang}@griffith.edu.au

Abstract. Event detection systems can help detecting incidents of interest that occur within data streams generated by interconnected sensor devices. Corresponding personnel need to be notified about the incidents so that appropriate actions can be taken to address them. One of the typical use cases of event detection systems is monitoring the quality of recycling water during wastewater treatment. Existing event detection systems in this domain usually focus on dealing with low level primitive events and thus are not capable of detecting more complex events which contain relations and background knowledge. In this paper, we present a framework for detecting complex events with background knowledge in event streams, by using ontology and query rewriting techniques.

Keywords: wastewater, ontology, event detection

1 Introduction

In today's business environment, complex events occurs in real time and companies can benefit from efficient real-time processing of data streams generated by interconnected sensor devices to discover such complex events. However, some complex events cannot be described using low-level event detection rules based on pattern-matching. The definition of such events usually requires domain expertise and background knowledge. Using ontology and rules to model complex events significantly enhances the accuracy of detecting such events. One of the use cases is monitoring the quality of recycled wastewater, where the water quality is monitored using sensing devices. A system that is capable of detecting complex events can provide an early indication of water quality changes, for relevant personnel to take appropriate actions to adjust the water treatment accordingly.

Existing stream modeling ontologies such as the SSN/SOSA [1] ontology are based on RDF(S) and are limited in their expressive power. To model complex events in the wastewater scenario, more expressive OWL ontologies and a framework that supports such ontologies are required. Also, while some system incorporates ontological reasoning with stream processing such as SANSA Stack [2], its inference is achieved via expanding the data by ontology, which is ineffective in handling large data streams. Hence, an alternative ontological reasoning approach is needed for event detection.

In this paper, we present a framework for detecting complex events with rich background knowledge over event streams, by using ontology and query rewriting techniques. We evaluate the performance of our framework and event detection using historical time-series data collected from one of the wastewater treatment plant in Australia and queries representing common water-quality-change events.

2 Our Framework

Our framework contains three components highlighted in blue in Figure 1. Sensor data flow into the framework as streams, and the ontology component validates and transforms the data into a format which streaming component can process. Users specify the events of interest as a set of queries Q. The query rewriting component rewrites Q into a more complicated query Q' based on the ontology model, and then delivers the transformed query Q' to streaming component. Finally, the transformed data run through Q' at the streaming component and the events of interest are detected through query answering. The framework was designed to be modular, so that each component can cater for different technologies.

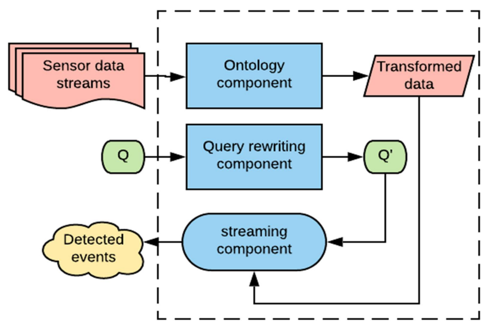

Fig. 1. Overview of the framework

The ontology component contains two parts. One is an ontology model which is a representation of generic complex events, regardless of which domain they belong to. The other one is a validation rule set which validates the input data using SHACL shapes. For the ontology model, we build an event ontology by extending the well-known SSN/SOSA [1] ontology with additionally 16 classes and 8 object properties for events and related notions. We also add OWL class definitions, together with expressive OWL axioms to capture complex ontological relationship which largely enhances the ontology expressiveness. The main classes created specifically for event detection include *ComplexEvents*, *EventObservation*, *EventProperty* and *EventLocation.* The ontology is developed under the aim of high re-usability and interoperability. That is, our ontology should be easily adaptable to other domain with minimal modification and extension.

Due to the nature of streaming data that data may arrive late or completely missing, we decided to implement a validation rule set. This rule set validates a complex event by comparing data with the axioms of *ComplexEvent* class in the ontology. This partly alleviates the problem of missing data by eliminating unqualified event. For example, a potential fire event must contain temperature and smoke data. If temperature data is missing at any particular time, our validator will not transform the data at this specific time. Once the validation is done, incoming data is transformed into another format, such as RDF, and is sent to the streaming component.

The query rewriting component is responsible for rewriting simple queries from non-expert user who does not have domain knowledge, into queries with incorporated domain knowledge. Query rewriting is a technique that uses ontology axioms to create more specific queries from the original most general query [3]. A set of queries are created based on ontology axioms, and then evaluated directly over the data (independent from the ontology). The answers to the rewritten queries are as if the ontology were consulted during the querying [3].

For example, users may want to ask ‘What time does an event of interest occur at which barrier of the treatment plant A?’ This question is translated to the below SPARQL query:

```
SELECT ?t ?b ?s WHERE {
      ?s a Barrier; name ?b;  hasEvent [a ComplexWaterEvent] .                    (1)
FILTER (?s hasName ‘A’)}
```

Axiom: IndustryDischarge subClassOf ComplexWaterEvent (i)

After query rewriting based on the ontology axiom (i), query (1) is transformed into

```
SELECT ?t ?b ?s WHERE {
     {?s a Barrier; name ?b;  hasEvent [a ComplexWaterEvent]} UNION              (2)
     {?s a Barrier; name ?b; hasEvent [a IndustryDischarge]}
FILTER(?s hasName ‘A’)}
```

The component was designed to be modular so that any query rewriting tools can be used in the component. The original queries Q are transformed into Q' after rewriting, and then propagated to the streaming component.

The streaming component is the place where incoming streams of the transformed data meets the rewritten query Q'. There are several candidates which are suitable to use as our streaming component, such as Apache Spark, Storm etc. An alert is generated as output of this component, once an event of interest is detected.

3 Evaluation

We tested our framework in the wastewater domain, using historical time series data collected from one of the wastewater treatment plant in Australia. The original data was in CSV format which consists total number of 79306 individual records. We first simulated the streaming data using a small program which reads the CSV file and ingest each record at speed of 30 seconds interval per record. Then, we added 2 extra classes which defines 3 types of complex events, ‘Industry Discharge’, ‘Heavy Rainfall’ and ‘Plant Fault’ in the wastewater domain to our original ontology. These 3 complex events are our targets to be discovered during the evaluation of our framework. The ontology we used in this test is depicted in Figure 2.

For the query rewriting component, we adopted Ontop [4] to perform the rewriting process. Ontop was primarily designed to query relational database using query rewriting technologies [4]. While SANSA Stack [2] has an inference module which can also perform query answering, it is done through expanding data by ontology axioms instead of query rewriting, which is ineffective for processing data streams.[1] Lastly, we have chosen Apache Flink as our streaming component for its stateful computation and the abilities to process data stream distributively. *SANSA Stack* [2] was also used to read the 'transformed data' into Flink and query the final results. As expected, we have successfully detected 11 'Industry Discharge' events, 1 'Plant Fault' event and 4 'Heavy Rainfall' events in the sample dataset.

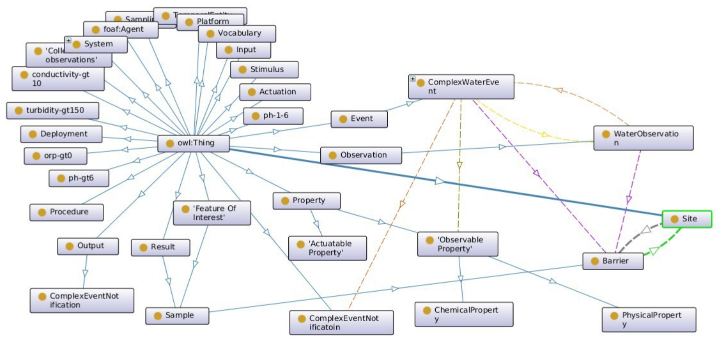

Fig. 2. Ontology for wastewater event detection

4 Conclusion

In this paper, we presented an ontology based framework to detect meaningful event and evaluated it using data collected from wastewater treatment process. Our system provides a innovative way for event detection in the water monitoring domain. In our future work, we would be focusing on reducing the system raw data conversion time, as well as enhancing the ability of detect multiple and more complex events.

References

1. Semantic Sensor Network Ontology [WWW Document], n.d. URL https://www.w3.org/TR/vocab-ssn/ (accessed 6.16.19).
2. Lehmann, J.; Sejdiu, G.; Bühmann, L.; Westphal, P.; Stadler, C.; Ermilov, I.; Bin, S.; Chakraborty, N.; Saleem, M.; Ngonga, A.-C. N. & Jabeen, H. (2017), Distributed Semantic Analytics using the SANSA Stack, *in* 'Proceedings of 16th International Semantic Web Conference - Resources Track (ISWC'2017)' , Springer, , pp. 147--155 .
3. Rodríguez-Muro, Mariano & Calvanese, Diego. (2012). Quest, a System for Ontology Based Data Access. CEUR Workshop Proceedings. 849.
4. Rodríguez-Muro M., Kontchakov R., Zakharyaschev M. (2013) Ontology-Based Data Access: *Ontop* of Databases. In: Alani H. et al. (eds) The Semantic Web – ISWC 2013. ISWC 2013. Lecture Notes in Computer Science, vol 8218. Springer, Berlin, Heidelberg

[1] Although the latest release notes claim that Ontop has been incorporated, but it seems query rewriting has not been fully implemented.