Ontology driven complex event pattern definition

Conference Paper · October 2016

CITATIONS
0

READS
141

5 authors, including:

Francois-élie Calvier
French Institute of Health and Medical Research
11 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Antoine Zimmermann
École Nationale Supérieure des Mines de Saint-…
85 PUBLICATIONS **599** CITATIONS
SEE PROFILE

Ontology Driven Complex Event Pattern Definition

short paper

Francois-Élie Calvier[1], Abderrahmen Kammoun[1], Antoine Zimmermann[2], Kamal Singh[1], Jacques Fayolle[1]

[1] Laboratoire Hubert Curien, Université de Saint-Etienne, Jean Monnet, F-42000, Saint-Etienne, France. email: firstname.lastname(at)univ-st-etienne.fr
[2] Laboratoire Hubert Curien, University of Lyon, MINES Saint-Etienne, F-42023, Saint Etienne, France. email: firstname.lastname(at)emse.fr

Abstract. Complex Event processing (CEP) usually focuses on analyzing raw atomic events in order to detect composite events. Usually, a composite event is defined as the pattern actively searched by a CEP system. However, considering uncertainty in some paradigms, such as internet of things, is still an open issue. In current approaches the confidence value related to the occurrence of an event is usually not communicated to the CEP system. As a consequence, a complex event pattern doesn't take this information into account. Nevertheless, even if static, they are useful for pattern definition and particularly for a more accurate constraint definition. We propose to manage this information through domain ontologies. In this paper we describe the architecture for the enrichment of CEP queries to enable evolutivity and flexibility in CEP systems according to event sources [9].

Acknowledgment: This work is supported by ITEA 3 project Water-M with the funding from DGE France.

1 Introduction

The internet of things paradigm is changing our world by providing smart solutions to numerous sectors such as urbanization, utilities (water and energy, health, etc). For example, water and energy networks are seeing an evolution in terms of a big upgrade of traditional meters to sensors and smart meters which allow us to improve our reaction time to network disrupts. However, the smart solutions are not limited to just relying on these observations and monitoring. Suppliers are more interested in disrupt prevention and efficiency improvement of their networks. Traditional performance indicators are still relevant, and the dynamic processing of such indicators, considering their evolution over time, is still an open challenge. But the availability of more frequent measures bring the possibility to discover novel indicators and tendencies.

CEP is a well defined solution for the cross analysis of dynamically evolving event streams. CEP engine implements a processing tool for high-level events that may result from low level factors or low level raw events. A sequence of raw events is considered as a complex event pattern. The processing step then aims at identifying the patterns within a raw event flow from multiple streams

of real-time data. This identification allows to take effective and immediate action in response to specific scenarios, such as the appearance of a problem in the network. Some CEP approaches based on relational models do not integrate semantic models. Because of these limitations processing dynamic and heterogeneous data from multiple sources, is still an open question. It has been shown that the exploitation of knowledge in ontological form, and the events of relations with other non-event concepts can improve the quality of CEP.

CEP uses a pattern as a standing query, defined by an administrator, to evaluate the event stream for detecting complex events. Fig.1. illustrates the pattern extraction process. The administrator exploits a knowledge base providing in-

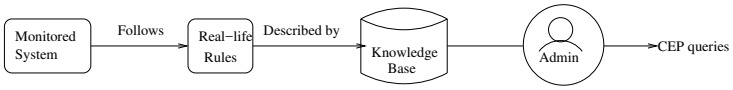

Fig. 1. CEP query extraction

formation about constraints over the events in the domain of the monitored system. The resulting patterns are then added as standing queries to the CEP system, which uses it to analyze the flow of events. Fig.2. shows the architecture of a CEP system.

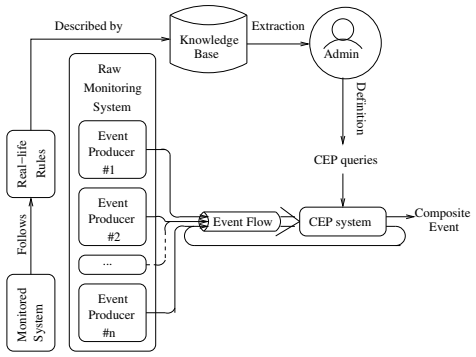

Fig. 2. CEP Architecture

A raw monitoring system is added which provides data, about the activity of the observed system, to the CEP system. The definition of patterns relies on domain knowledge to identify both final events and the possible sequences of events leading to each final event. Thanks to a formal conceptualization of the domain knowledge such as ontologies[3], we propose to integrate information about the event sources to enable a more accurate and meaningful pattern definitions.

This paper is organized as follows. In the next section, we present a set of related works. Then we show an overview of our architecture and pattern model in sections 3 and 4. Finally, we describe our proposals in section 5 before concluding and presenting future works.

2 Related Works

In section 1, we introduced CEP systems and how an association with ontologies would enhance their accuracy. In this section we present works based on CEP

systems. We roughly classified these works as follows: works proposing CEP features which are interesting for our proposal build-up, and works proposing solutions for a similar objective, but exploiting other technologies, and works exploiting an association of ontologies and CEP systems, but having different objective than ours.

2.1 Interesting CEP features

The common architecture of CEP systems is quite coarse and several works propose enhancements for the management of both raw and complex events. Information Flow Processing (IFP) is defined in [5] as an umbrella term for technologies such as Data Stream Management System (DSMS) and CEP. DSMSs are an extension of DataBase Management Systems which focus on outputting query answers for continuously changing input data. However, detection of complex events involving sequences and ordering of events are not tackled by DSMSs. CEP systems address these limitations by tackling such patterns. Queries from DSMS and patterns from CEP are both part of IFP rules. Both in DSMSs and CEP systems, rules are usually defined once and are mostly neither added, nor suppressed, nor modified. To overcome this, in [1], the authors propose dedicated patterns that catch some update raw events and make the system enter in the updating state.

In our approach, we propose to integrate static data from the source description. This extension of the pattern definition from *detection only* rules to *transformation and detection* rules would permit this integration. Above all, complex event patterns need to be evolutive and interactive. Thus we need such a solution to apply our pattern modifications on event source evolution.

2.2 Similar approaches using different technologies

Adequacy of CEP patterns with event sources can be partly seen as a trust problem and can be expressed as "how reliable an event source is". In CEP context, occurrence of an event is a binary problem. Thus, the trust problem mainly focuses on the loss of an event (i.e. events that occur but were not transmitted to the CEP system). Nevertheless, some approaches use probabilities and fuzzy logic to introduce uncertainty in CEP. In [2], the authors identify uncertainty source and then model it through a Bayesian network. Constraints from the complex event pattern are then relaxed using the Bayesian networks to detect and propagate the uncertainty in the CEP.

The introduction of uncertainty in those approaches leads to a modification of the core CEP system. Our approach is somewhat different since we do not modify the existing CEP system itself nor its input model .

2.3 Approaches based on ontology association

Exploiting ontologies as background knowledge for CEP system is not a novel idea and numerous works have studied this association to propose Semantic Complex Event Processing (SCEP) systems . A survey is proposed in [6].

Some rare approaches like [4] describe domain ontology CEP which consist in defining a pattern as a semantic query on a domain ontology. CEP is then the

processing of the defined queries over the data populating the domain ontology. In most cases, event ontologies are exploited in CEP to describe how events interact. Their exploitation can then be similar to domain ontology CEP like in [10]. However, the event ontology can also be exploited as an interface to define patterns in the CEP native query language like described in [7]. Usually, event ontologies are domain specific and do not describe an abstract framework on which any domain ontology could be inserted. Nevertheless, in [8] the authors propose a modular upper ontology based architecture for CEP. They try and define a reusable Events ontology. This modular ontology integrates Time, Space and Agents ontologies and defines a *situation module*, which in turn is the domain specific part of the ontology.

Yet, the association of ontologies and CEP in existing approaches mainly aims at proposing a high level and semantically rich language for pattern definition. In our approach, we propose to focus on the description of the event sources to fit the CEP pattern within the CEP or SCEP existing languages. Our point is not to propose a language, but to provide the ability of self-adaption and evolution to CEP systems, according to the event sources.

3 Architecture for complex event pattern enrichment

CEP patterns are neither designed for evolution nor adaptation. To enable this, we need a dedicated module which can create and manage a link between the patterns and the properties of event sources. We illustrate our proposal in Fig.3.

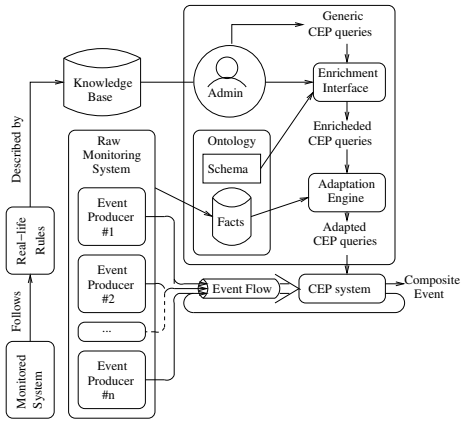

Fig. 3. Proposed architecture

To the existing CEP architecture, we propose to add a semantic layer composed of an ontology (schema and facts) describing the event sources, an interface enabling the administrator to be aware of this semantic description and an automata to generate the adapted CEP queries from the enriched ones.

Thus, in our architecture, query adaptation is a semi automatic process composed of two steps. During the first step called query enrichment, the administrator defines the enriched CEP queries from the existing generic CEP queries exploiting the schema of the ontology. The model of enriched CEP queries is decribed in section 4. Concurrently, information from the event source populates

this ontology. During the second step called query personalization, the resulting data are integrated to generate the adapted CEP queries. This automatic, second step is detailed in section 5.

4 CEP pattern model

We aim at defining CEP queries adapted to the different event sources by using their semantic description. Thus, we have to define a CEP pattern model that allows for adaptability and evolutivity. We focus on the following points to deal with: 1) Uncertainty over data, 2) Time within event sequences, and 3) Composite events as an input.

In this section, we first describe how the semantic description of the event sources meets those three features. Then, we present our enriched CEP pattern model which captures these requirements.

4.1 Uncertainty awareness

Patterns are well known rules from the domain which the users want to detect. However, there is a gap between what the CEP manager wants to capture and the patterns he actually defines. This gap has multiple causes:

Data encoding: rules often use units. Without information about units, the event detection of a CEP system can be faulty. For example, considering a pattern addressing the preparation of a boiled-egg, the sequence would be E_1: the water is warming, E_2: the water reaches 80 °C, E_3: the egg is put in the water, E_4: 10 minutes past, E_5: the egg is boiled. If the thermal sensor is sending data in Fahrenheit then the CEP will be faulty. More broadly, event transmission relies on a protocol defining how to extract the source, the value and the timestamp from the bit sequence, but does not allow to describe the value type. A misinterpretation of the value can cause a faulty detection in a CEP system.

Observation errors introduced in the system: the well known rules are theoretical and should be tuned according to the information about the event source. Coming back to our example, if the thermal sensor or the timer has an error rate then the prediction of the E_5 state will not be precise. Furthermore, an event source may also have a limited data range. This piece of information should be taken into account when defining patterns in order to first, prevent unreachable states (waiting an event out of source range) and second, to detect faulty events. In our example, if a mistake leads to the use of a thermal sensor with a range from 0 °C to 50 °C then E_3 can never turn into a true state. On the other hand, if a -5 value is sent then the system should detect that there was a problem with the source relative to E_3.

4.2 Time awareness

Usually, an event is represented as a triple (s, v, t) where s is the event source, v is the value sent and t is the time stamp of the event. Up to now we have explained that the semantic description of s should be taken into account for the evaluation of v. We will now discuss its interactions with the t parameter.

A peculiarly interesting property for CEP can be the update frequency of a given source. A pattern can involve a time relation between two events (i.g.

time sequence, time windows). But the sources of the involved events may have different update frequencies. In our example, the thermal sensor may send data every minute and the egg position sensor may send data when the position changes. If the water reaches 80 °C one second after the sensor has sent data then the *put the egg in the water* event can be uncaught by the CEP because it will be sent before the *boiling water* event. In this case, the final state of the pattern is unreachable even if it has been reached by the monitored system.

Another similar application is event loss. If the frequency of an event source is defined, then when processing the event flow, the CEP system can check if the sequence of events from a same source is correct or not. If the difference between the time-stamps of two following events is higher than the update frequency of that event source, then it means that an event is missing.

4.3 Composite event as an input

Once a pattern has been caught by a CEP system, the resulting composite event can be further used as input by another CEP system detecting another pattern. Thus, CEP system can also be a source of events. Such a source of composite events should then be described like any other event source. However, the semantic description of a pattern is not its semantic definition. Indeed, the needed description should enable the detection of faulty results from this pattern detection. In the former case, the sequence of events is not useful, but the description of causes of a given fault (the source of the events, the faulty events resistance, etc.) is mandatory. In the latter, fault detection can not be taken into account and the event sources can not be semantically described.

4.4 CEP pattern model description

Referring the triple representation (s, v, t) of an event (cf. section 4.2), a raw event pattern is then represented as a triple $(s, f(v), t)$ where f is a Boolean function of v. Therefore, a complex event pattern is a Boolean relation involving several raw event patterns and Boolean functions of their t attributes. In our example, E_2 is represented as $(thermalSensor, isUpper(v, 80), t_2)$ and the pattern is represented as $(E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5 \wedge t_1 < t_2 \wedge t_2 < t_3 \wedge t_3 < t_4 \wedge t_4 < t_5)$.

The functions involved in those patterns are static. Our enriched query pattern model extends this static definition with the introduction of a (set of) parameter(s) to these functions. We propose to extract the additional parameters from the *data property* set of the ontology schema. In our running example, focusing on error rate of the sensor, the function $isUpper(v, 80)$ of E_2 would then become $isUpper(v, 80, Thermometer.errorRate) \equiv isUpper(v, 80 + Thermometer.errorRate)$.

To match time awareness, the enriched patterns should not rely on sequences of events, but on their co-occurrences within a time frame. A time frame would then be defined as a function of the update frequencies of each of the involved events. One solution is to define a pattern for each couple of sequence events. In our example, the sequence E_1, E_2, E_3, E_4, E_5 would become CE_1: the water is boiling $\equiv ((E_1, E_2) \vee (E_2, E_1$ within $S_{E_1}.frequency))$, CE_2: the egg is in boiling water $\equiv ((CE_1, E_3) \vee (E_3, CE_1$ within $S_{E_3}.frequency)), CE_3$: the egg is in boiling water since 10 minutes $\equiv ((CE_2, E_4) \vee (E_4, CE_2$ within $S_{E_4}.frequency))$.

We describe in section 5 how to exploit this enriched pattern model to produce adapted queries.

5 Pattern enrichment and adaptation

In section 4, we presented our enriched CEP pattern model able to deal with the semantic information of the event sources. In this section, we describe how this model can be exploited to produce adapted queries.

5.1 Exploiting the properties from the ontology

Our pattern enrichment relies on *data properties* from the ontology. Any event source of the system instantiates concepts from the domain ontology. Then, some static information about the event source are stored in the data properties defined by its relative concepts. We define them as proper information. They strictly match with the description of our pattern model.

Considering a given pattern p, some relevant information may be stored in the ontology but not as a data property of the event sources involved in p. Beside proper information, we thus define cross-layer information as the piece of information that cannot be inferred from the data description of an event source (data properties), but that also involves the description of other instances (object properties). Contrarily to proper information, cross-layer information do not strictly match the description of our pattern model since the additional parameter is not a single data property but a chain of object and data properties. Moreover, the other instances involved in cross-layer information can also be event sources for another pattern. When it happens, cross-layer information can be extracted from those sources and thus, be dynamic.

Using a semantic description, the manager of the CEP system can be aware of the additional information and decide which ones to integrate and how to integrate them within the domain rules. A pattern is thus a sequence of events parameterized by properties from a domain ontology.

5.2 From enriched to adapted pattern

The integration of static source information can be realized either at raw event level like in an IFP system or at pattern definition step. The former solution lets the theoretical rules unchanged, and thus easily reproducible, but leads to additional computation for each event.. The latter generates only one additional computation for each pattern but modifies the theoretical rule. Moreover, both solutions are not evolutive nor self-adaptative. The CEP manager has to suit each instance of a same rule to its event sources. Using our enriched CEP pattern model, each rule becomes a parameterized static definition of a given pattern and an instantiation becomes a set of parameters' values. Besides, on each source modification, an update event is sent to the system and each complex event pattern is updated according to its parameters' values. In our example, E_2 becomes E_2'': the water reaches 80 °C($sensor_1.errorRate$) and E_4 becomes E_4'': 10 minutes past($timer_1.errorRate$).

This instantiation is done during the personalisation step. The introduction of confidence parameters would lead to the withdrawal of events. Indeed, if the

source of an event is known to be sending false or irrelevant values, the associated flow can be disabled. Then, a pattern waiting for an event e to be originating from a faulty source can be disabled, if e is not part of an event branch of the underlying detection logic, for example an intervening state of a Non-deterministic Finite Automata (commonly used in CEP systems as an event detection model). Such a pattern can not be matched on the event flow and its processing is computationally extensive. For the same reason, if e is part of an event branch eb_1, the pattern definition should be modified by cutting out eb_1. Thus, if a modified pattern p can not reach a final state then p should also be disabled.

6 Conclusion and future works

In this paper we propose an ontology driven adaptation process for the definition of complex event patterns in CEP. Our method allows more accurate patterns by presenting a knowledge representation about the sources of events. The CEP administrator can choose to integrate the additional knowledge to the rules either in a static way or in a dynamic way. Our method can be used with any existing or future CEP system since it does not modify the semantics nor the syntax of the CEP queries. Our proposal of query parameterization with ontological properties can be adapted to any system needing evolutive ability and adaptability.

References

1. Bhargavi, R., Pathak, R., Vaidehi, V.: Dynamic complex event processing—adaptive rule engine. In: Recent Trends in Information Technology (ICRTIT), 2013 International Conference on. pp. 189–194. IEEE (2013)
2. Cugola, G., Margara, A., Matteucci, M., Tamburrelli, G.: Introducing uncertainty in complex event processing: model, implementation, and validation. Computing 97(2), 103–144 (2015)
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5(2), 199–220 (Jun 1993), http://dx.doi.org/10.1006/knac.1993.1008
4. Liu, J., Guan, X.: Complex event processing for sequence data and domain knowledge. In: Mechanic Automation and Control Engineering (MACE), 2010 International Conference on. pp. 2899–2902. IEEE (2010)
5. Margara, A., Cugola, G.: Processing flows of information: from data stream to complex event processing. In: Proceedings of the 5th ACM international conference on Distributed event-based system. ACM (2011)
6. Schaaf, M., Grivas, S.G., Ackermann, D., Diekmann, A., Koschel, A., Astrova, I.: Semantic complex event processing. Recent Researches in Applied Information Science pp. 38–43 (2012)
7. Taylor, K., Leidinger, L.: Ontology-driven complex event processing in heterogeneous sensor networks. In: Extended Semantic Web Conference. Springer (2011)
8. Teymourian, K., Coskun, G., Paschke, A.: Modular upper-level ontologies for semantic complex event processing. In: WoMO (2010)
9. Teymourian, K., Paschke, A.: Enabling knowledge-based complex event processing. In: Proceedings of the 2010 EDBT/ICDT Workshops. p. 37. ACM (2010)
10. Zhu, T., Bakshi, A., Prasanna, V.K., Gomadam, K.: Applying semantic web techniques to reservoir engineering: Challenges and experiences from event modeling. In: Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations. pp. 586–591. ITNG '10, IEEE Computer Society, Washington, DC, USA (2010), http://dx.doi.org/10.1109/ITNG.2010.183