Complex pattern detection and specification from multiscale environmental variables for biodiversity applications

Jacqueline Midlej do Espírito Santo[1], Claudia Bauzer Medeiros[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
13.083-852 – Campinas – SP – Brazil

jacqueline.santo@students.ic.unicamp.br, cmbm@ic.unicamp.br

Abstract. *Biodiversity scientists often need to define and detect scenarios of interest from data streams concern meteorological sensors. Such streams are characterized by their heterogeneity across spatial and temporal scales, which hampers construction of scenarios. To help them in this task, this paper proposes the use of the theory of Complex Event Processing (CEP) to detect complex event patterns in this context.*

1. Introduction

Biodiversity broadly means the abundance, distributions and interactions across genotypes, species, communities, ecosystems and biomes. Countless problems in biodiversity studies require data collected and analyzed at multiple space and time scales, correlating environmental variables, living beings and their habitats [Hardisty and Roberts 2013]. An open problem in this context is how to specify and detect patterns from environmental variables in multiple scales to help scientists to analyze phenomena and correlate results with data collected on the field.

To help solving the problem, this work proposes to use Complex Event Processing (CEP), the technology to process data streams concern meteorological sensors via event detection. The main goal is to detect event patterns in near real-time, in order to signal situations of interest [Sen et al. 2010]. The idea is to allow researchers to specify and combine events that characterize such situations, in the context of biodiversity applications. For now, scenarios of interest are usually built case-by-case; sensor events are sometimes captured by customized software. The paper extends the framework proposed by [Koga 2013] for this purpose.

2. Basic Concepts

In CEP, the word *event* means the programming entity that records an occurrence of something in a domain [Etzion and Niblett 2010]. Events are classified into primitive and complex. Primitive events represent an occurrence at a given place and time. Complex events are formed by combinations of primitive or complex events.

The main task of CEP is to detect complex events, in order to identify within a set of events those that are significant to an application domain. Such a detection occurs through matching events with patterns. Patterns represent models of scenario of interest composed by specification of events and their relationships. Patterns can be defined on a hierarchy of events in which the highest level events are formed by inferences from lower level events.

3. Related Work

Depending on the context, the structure and components of events can change. [Koga 2013] defines 4 attributes to specify events in environmental applications: measured-value, nature, spatial-variable, and timestamp. However, this representation only describes primitive events. The description of complex events must define relationships between events. For example, [Sen et al. 2010] represents complex events in business applications by a model based on semantics which, besides the basic attributes, has reference to operators that connects events.

Patterns are specified by event processing languages. These languages are mainly defined using approaches based on logics (logic-based) or automata (automata-based). Many research efforts in defining more powerful languages. For instance, [Barga and Caituiro-Monge 2006] describe the language *Complex Event Detection and Response* (CERD) for expressing patterns that filter, generate and correlate complex events in business applications.

Logic-based patterns are defined as combinations of predicates on events. Examples of works using this approach are [Motakis and Zaniolo 1995] and [Obweger et al. 2010]. The first authors define a model for active databases in which the pattern composition is described by $Datalog_{1S}$ rules. For biodiversity applications, our target, this model is limited because $Datalog_{1S}$ only supports one temporal operator. Scenarios that have more complex temporal relationships and/or have spatial relationships cannot be represented. On the other hand, [Obweger et al. 2010] do not limit the predicate to the use of specific operators. In addition, their model allows users to compose hierarchical patterns using an interface that abstracts the definition of sub-patterns.

In automata-based approaches, regular expression operators are used to compose patterns. This approach limits the temporal relationships to the notion of precedence and does not support spatial operators. Examples of papers in this line are [Pietzuch et al. 2004] and [Agrawal et al. 2008]. The first one performs event detection in distributed systems. The latter focuses on improving the runtime performance of pattern queries over event streams, for business applications.

4. Partial Results

This work has two main parts. The first one aims at formalizing specification of events on biodiversity, inspired by literature proposals applied to different domains (e.g., [Etzion and Niblett 2010, Barga and Caituiro-Monge 2006, Sen et al. 2010]). It must: allow the hierarchical events composition, such as [Sen et al. 2010]; combine heterogeneous data sources, such as [Koga 2013]; and consider the place where the event occurs, such as [Koga 2013]. It must also extend the semantics of operators to support spatial and temporal multiscale data. This specification can express biodiversity scenarios of different complexity, from excessive rain to situations combining river data with vegetation and relief data;

The second contribution of this work is the development of a mechanism that allows patterns composition and detection in order to assist biodiversity applications. This step extends the work of [Koga 2013], which allows integrating data from heterogeneous sources; however, it is limited to the detection of primitive event patterns. The Figure 1 illustrates the adapted architecture, horizontally drawn, of the extended framework. The architecture has two main aspects: the use of Enterprise Service Bus (ESB) to process data streams uniformly and the use of CEP to detect patterns. Environmental data are pre-processed and translated into events, which pass through the ESB and are processed by CEP.

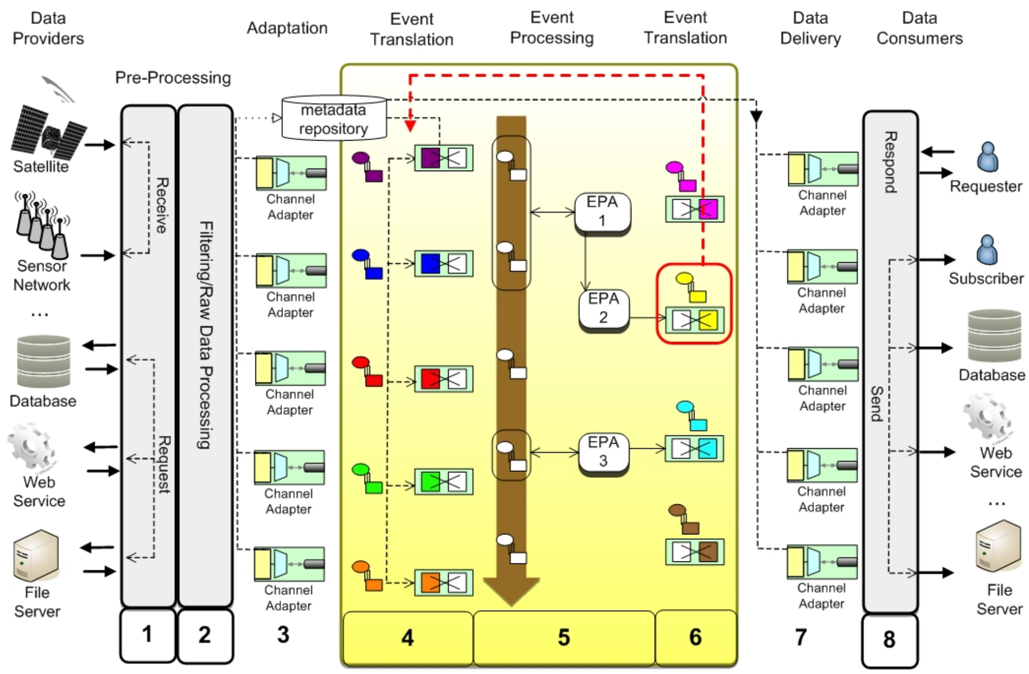

Figure 1. Adapted architecture from [Koga 2013]

From bottom to top, steps 1 and 2 filter data according to the goal of application. At step 3, the data are encapsulated into messages that are standardized by channel adapters (ESB template). Steps 4 and 5 correspond to the translation of messages into events and their processing by CEP. If a pattern is detected, step 6 encapsulates the matched event into a new message. At the steps 7 and 8, this message is standardized and sent to the interested user.

Our work complements the architecture adding complex pattern composition and detection, illustrated by the red arrow from step 6 to step 4 in Figure 1. This adaptation provides more representative patterns. The detected composition of events is sent back to the ESB bus, and forwarded back into the pipeline, creating a hierarchical structure. The output of a complex event may become part of more complex compositions, generating composite events at a higher level.

Using the architecture, biodiversity scientists can represent scenarios (as deforestations and forest fires) by complex patterns and detect them. For instance, detecting climatic changes as the arrival of a cold front in Campinas involves the monitoring of several environmental variables. A short logic-based pattern for this scenario can be:

$$
\begin{aligned}
&\exists Et1 | Et1.temp < 5\,^{\circ}\mathrm{C} \wedge dist(Et1.space, Campinas) < 200km \wedge \\
&\exists Et2 | Et2.temp > 20\,^{\circ}\mathrm{C} \wedge touch(Et1.space, Et2.space) \wedge \\
&\exists Ew | Ew.windSpeed > 60km/h \wedge overlap(Ew.space, Et1.space) \wedge movDir(Ew) = Campinas \\
&overlap(Et1.time, Et2.time, Ew.time)
\end{aligned}
$$

This pattern searches for a composition of event $Et1$ signaling low temperature (cold air mass), "meeting" with $Et2$ signaling high temperature in Campinas (hot air mass), and Ew which shows the presence of strong wind carrying the cold front to Campinas. The detection process finds events $Et1$ and $Et2$, generating complex event $CE1$. This event is fed back to the bus. Next, $CE1$ and Ew are detected, generating the complex event $CE2$ that confirms the cold front. At the detection hierarchy, when $CE1$ and $CE2$ are generated, they form a higher hierarchical level.

This framework will be validated over sensor data, provided by Cooxupé, cooperative of coffee farmers, from 14 weather stations in Minas Gerais and São Paulo, data used to validate the work of [Koga 2013]. Each weather station continuously collects at least 26 types of measurements, e.g., temperature, humidity, barometric pressure and so on.

5. Conclusions

This paper proposes a software framework to help biodiversity scientists to quickly detect scenarios of interest. These scenarios are specified by event patterns. The expressiveness of patterns and events is considered in their specification, and the handling of multiscale data is considered. The detection is made by a hierarchical and logic-based approach. Future directions include defining the pattern language and partial implementation.

Acknowledgements Work partially financed by FAPESP (2013/02269-7), FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the MSR-FAPESP Virtual Institute (NavScales project), CNPq (MuZOO Project), FAPESP-PRONEX (eScience project), INCT in Web Science, and grants from CNPq.

References

Agrawal, J., Diao, Y., Gyllstrom, D., and Immerman, N. (2008). Efficient pattern matching over event streams. In *ACM SIGMOD*, pages 147–160.

Barga, R. S. and Caituiro-Monge, H. (2006). Event correlation and pattern detection in cedr. In *EBDT*, pages 919–930.

Etzion, O. and Niblett, P. (2010). *Event Processing in Action*. Manning Publications Co.

Hardisty, A. and Roberts, D. (2013). A decadal view of biodiversity informatics: challenges and priorities. *BMC Ecology*, 13(1).

Koga, I. K. (2013). *An Event-Based Approach to Process Environmental Data*. PhD thesis, Instituto de Computação - Unicamp. Supervisor Claudia Bauzer Medeiros.

Motakis, I. and Zaniolo, C. (1995). Composite temporal events in active database rules: A logic-oriented approach. In *DOOD*, volume 1013 of *LNCS*, pages 19–37.

Obweger, H., Schiefer, J., Kepplinger, P., and Suntinger, M. (2010). Discovering hierarchical patterns in event-based systems. In *SCC*, pages 329–336.

Pietzuch, P., Shand, B., and Bacon, J. (2004). Composite event detection as a generic middleware extension. *IEEE Network*, 18(1):44–55.

Sen, S., Stojanovic, N., and Stojanovic, L. (2010). An approach for iterative event pattern recommendation. In *DEBS*, pages 196–205.