Complex Pattern Detection and Specification to Support Biodiversity Applications

Jacqueline Midlej do Espírito Santo[1], **Claudia Bauzer Medeiros**[1] **(advisor)**

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

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

Level: Master's degree
Admission: March 2013
Qualifying Exam: 29th November 2013
Expected Conclusion: March 2015
Completed Steps: Bibliographic research; Event and pattern specification for biodiversity context.
Future Steps: Extend Event Processing Language (EPL) from the Esper engine; Adapt a framework to detect complex patterns using the language extended; Evaluate the framework.

Abstract. *Biodiversity scientists often need to define and detect scenarios of interest from data streams delivered from meteorological sensors. For example, scenarios such deforestation or forest fire need to be detected in order to reduce impacts over the environment. Such data streams are characterized by their heterogeneity across spatial and temporal scales, which hampers detection of events and construction of scenarios. To help scientists in this task, this work proposes the use of the theory of Complex Event Processing (CEP) to define and detect complex event patterns in this context. The two main contributions focus on the specification of events and patterns for the biodiversity context and on the mechanism to detect these patterns. The first one requires to extend an Event Processing Language (EPL) to include spatial relationships in the pattern. The second one will extend Koga's framework [Koga 2013], which integrates heterogeneous data sources, with the detection of complex patterns. This paper extends the short paper accepted for the Brazilian Workshop e-Science (BreSci) 2014 with the specification for events and patterns.*

Keywords: *Complex Pattern Detection, Complex Pattern Specification, Multi-scale Data, Environmental Variable, Spatial and Temporal Relationship.*

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]. For example, environmental monitoring requires data from environmental variables mainly generated by meteorological sensors. When this monitoring involves animals, there are other sensors for motion and sound. An open problem in this context is how to specify and detect scenarios of interest (as climate change, deforestation or water pollution) from environmental variables, in multiple scales, to help scientists analyze phenomena and correlate results with data collected on the field.

To help solve the problem, this work proposes to use Complex Event Processing (CEP) to process data streams of meteorological sensors. The main goal of CEP is to detect event patterns in near real-time, in order to signal situations of interest [Sen et al. 2010]. Our idea is to allow researchers specify and combine events that characterize such situations and detect their occurrence in the context of biodiversity applications.

The paper presents two main contributions: (i) The specification of events and complex patterns for biodiversity context. This context requires patterns composed by combining events with temporal and spatial relationships (the latter unavailable in CEP). Therefore, this paper proposes to extend an Event Processing Language (EPL) to support these particulars. (ii) The development of a mechanism to detect complex patterns. For this purpose, the paper extends the framework used to integrate heterogeneous data sources proposed by [Koga 2013], which deal with simple events. This paper extends the short paper accepted for the Brazilian Workshop e-Science (BreSci) 2014 with the specification for events and patterns in the biodiversity context.

2. Theoretical Foundations

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. Primitive events represent observations outside from the event system, while complex events represent events defined inside the system [Pietzuch et al. 2004]. Examples of primitive events from environmental variables are measurements about temperature, barometric pressure and wind speed. Examples of complex events are cold front, fire, or poor water quality.

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 detection occurs through matching events with event patterns (patterns for short). Patterns represent models of scenario of interest composed by specification of events and their relationships [Etzion and Niblett 2010]. Patterns can be defined on a hierarchy of events in which the patterns specify how the highest level events are formed by inferences from lower level events. Their composition are defined by Event Processing Languages (EPL). Sometimes, the literature refers to composition of patterns using the term complex event, and other times, the term complex pattern. This paper discusses the subject regardless of the term used in the literature. One of the contributions of the dissertation is the uniformization of related work, which misses term such as event and pattern, with respect to a wide variety of similar operations and constructs.

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. 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, references to operators that connect events.

Event Processing Languages, used to specify patterns, are mainly defined using approaches based on logics (logic-based) or automata (automata-based). Many research efforts are concerned with defining more powerful languages. For instance, [Barga and Caituiro-Monge 2006] describe the language *Complex Event Detection and Response* (CEDR) for expressing patterns that filter, generate and correlate complex events in business applications.

Logic-based patterns are defined as combinations of logic predicates on events. Examples using this approach are [Motakis and Zaniolo 1995] and [Obweger et al. 2010]. The first 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. Contributions

This work has two main parts. The first one aims at formalizing specification of events and patterns on the biodiversity context, inspired by proposals applied to different domains (e.g., [Etzion and Niblett 2010, Barga and Caituiro-Monge 2006, Sen et al. 2010]). It must: allow 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 hydrographic data with vegetation and relief data.

The second contribution of this work is the development of a mechanism that allows patterns composition and detection to assist biodiversity applications. This step extends the work of [Koga 2013], which allows integrating data from heterogeneous sources, but it is limited to the detection of primitive event patterns. Figure 1 illustrates the architecture, horizontally drawn, of the extended framework. Kogas's proposal 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. We extend this work by providing continuous event feedback into the bus, to allow event composition and detection of complex events.

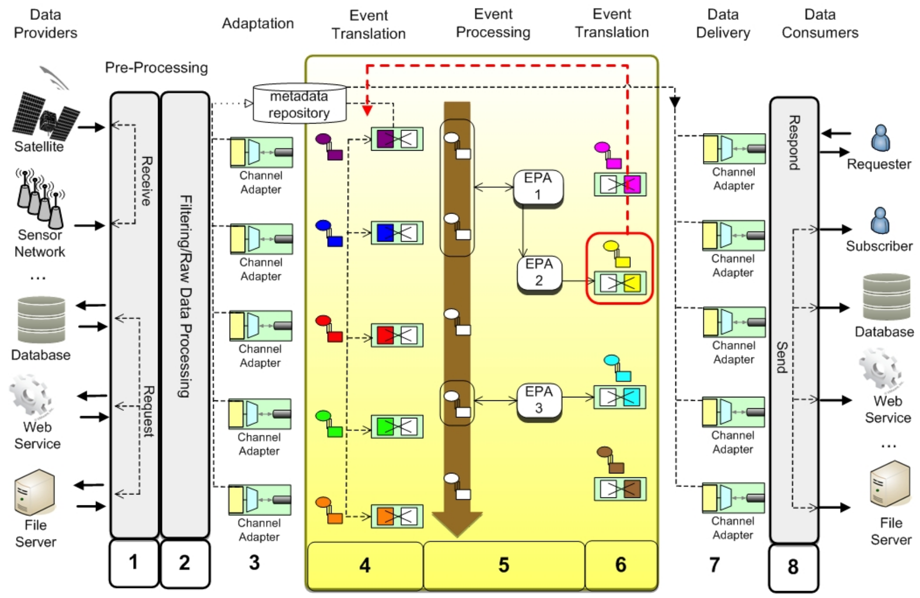

Figure 1. Adapted architecture from [Koga 2013]

From left to right, 1 through 3 filter data according to the goal of the application and encapsulate events into messages. 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 steps 7 and 8, the message is sent to the interested user.

Our work complements the architecture adding complex pattern composition and detection, illustrated by the red dotted 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. Though represented by one small arrow, this extension will require considerable design and implementation efforts.

This framework will be validated over meteorological sensor data combined with hydrographic, topographic and relief data. Meteorological data are provided by Cooxupé, the largest coffee cooperative in the world, from 14 weather stations in cities located in Minas Gerais and São Paulo. The stations continuously collect at least 26 types of measurements, e.g., temperature, humidity and barometric pressure. Each station records all measurements once a minute. However, the other data sources correspond to records collected at distinct temporal granularities, and distinct geospatial coverage, thereby instantiating the multiscale aspect in time and space. We will create patterns that specify scenarios of interest exploring spatial and temporal relationships among these data. The framework will hierarchically detect data that match a pattern (starting from atomic events upwards). We point out that, as far as we know, event detection approaches using CEP do not take spatial relationships into consideration, nor do they consider handling multiscale data.

5. Ongoing Work

5.1. Biodiversity Event Definition

The definition of events in the biodiversity context requires using data from different sources and scales on time and space. In the following list, we define the set of attributes required to represent primitive events, obtained as part of our research:

- *Id* refers to the attribute to identify a single event.
- *Source* refers to device, database, sensor or application that produces the event. This attribute filters the set of data in which the user is interested. Different sources can measure different variables with varied quality. For example, an expert may know which sensors are in a better location or provide more reliable data.
- *Type* refers to which type of variables were observed or measured. In other words, what this event is about.
- *Value* refers to the value measured from the variable specified by *Type*.
- *UnitOfMeasure* refers to the unit of measure from the *Value* attribute. Events with the same *Type* can be stored with different units of measure due to, for example, an user's cultural habits.
- *Space* refers to the location where the event occurs. It can handle data generated at multiscales on space: a point at space, a line or a polygon.
- *Time*, composed by *StartValidTime* and *EndValidTime*, refers to the time when the event is considered valid for the event system. The meaning of this attribute is adapted from the CEP concept. Originally, CEP uses an interval to represent the time in which the event is happening. Our adaptation provides a way to handle different types of events generated at multiscales on time. Events concerning soil data, for example, do not change quickly, so they may be generated just once a year. Because of that, these events are considered valid for the entire year. On the other side, temperature events have a short valid time and are generated frequently.

These attributes support multiscale data; we believe they are also sufficient to represent events from heterogeneous sources in biodiversity. We point out that we can pre-process data before transforming them into events (steps 1 and 2 from Figure 1). For example, satellite images or animal sounds can be represented by descriptors. These descriptors are formed by a set of measured variables and values, which can become a set of events.

Complex events, however, do not represent a single observation. Thus, they do not keep the *Type* and *Value* attributes from the primitive event schema. They have an *Id*, a *Source* (the event system) and they keep aggregated information about space and time.

The values of *Space* and *Time* of complex event E are computed from the values of the events that compose E (immediate lower level events). *Space* considers the smallest area and *Time* considers the smallest time interval containing all the immediate lower level events. In addition to these events attributes, the complex event has a list of operators and a list of references to events. The first represents the relationships between the events. It is the operator used to aggregate the lower level events into complex event. The later consists of references to the immediate lower level (primitive or complex) events.

In our solution, complex events are defined by hierarchical composition of less complex events. Events are aggregated into complex events using different kinds of relationships. This approach allows the lower level events be traced back from the complex event. In CEP, this backtracking task is called *drill down*. On our specification, each complex event keep information about the events that directly creates it, stopping the drill down at primitive events.

5.2. Biodiversity Pattern Definition

Using the logic-based approach, we define the aspects that a pattern language should consider for the biodiversity context: connectors, quantifiers and operators over event attributes. The connectors are logical operators used on the combination of two or more predicates. The main connectors are $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$(negation). The main quantifiers are $\forall$ (universal quantifier) and $\exists$ (existential quantifier). Operators should consider the nature of each attribute. Examples of operators over numerical attributes are =, $<$ and $>$. Examples of operator over temporal attributes are [Allen 1983]'s relationships: *before/after, meets/is-met-by, overlaps/is-overlapped-by, finishes/is-finished-by, contains/during, starts/is-started-by* and *equals*. Examples of spatial operators are distance and topological relationships between two objects (*contained-in/contains, overlaps/disjoint, equals* e *touches*).

It is important to notice that biodiversity scenarios need to express relationships between events considering temporal and spatial aspects. We have not found EPL that provide spatial operators in the CEP literature. We aim to extend the EPL from the Esper Engine[1], an open source software, to support spatial operators. This SQL-like EPL has good documentation and is the most complete pattern language found. It supports numerical and temporal operator, besides the connectors and quantifiers required by our definition.

5.3. Running Example

Using the specifications and framework proposed, biodiversity scientists can represent scenarios (as deforestations and forest fires) by complex patterns and detect them. For instance, detecting climate 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:

$$\begin{aligned}
&\exists Et1|Et1.type = temp \wedge value < 31 \wedge dist(Et1.space, Campinas) < 200km \wedge \\
&\exists Et2|Et2.type = temp \wedge value > 54 \wedge touch(Et1.space, Et2.space) \wedge \\
&\exists Ew|Ew.type = DirWind \wedge value = southwest \wedge overlap(Ew.space, Et1.space) \wedge \\
&overlap(Et1.time, Et2.time, Ew.time)
\end{aligned}$$

[1] http://esper.codehaus.org/

This pattern contains 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 wind carrying the cold front to Campinas. At the framework, the detection process finds events $Et1$ and $Et2$, generating complex event $CE1$ with the operators *touch* and *overlap*. This event is fed back to the bus. Next, $CE1$ and Ew are detected, generating the complex event $CE2$ with the operator *overlap* that confirms the cold front. When $CE1$ and $CE2$ are generated, they form a higher hierarchical level.

6. Conclusions and Future Work

This paper proposes a software framework to help biodiversity scientists to specify and detect scenarios of interest. These scenarios are specified by event patterns. The expressiveness of patterns and events and the handling of multiscale data are considered in their specification. The detection is made by a hierarchical and logic-based approach. Future directions include extending Esper's EPL to support pattern with spatial relationships and extending the [Koga 2013]'s framework to detect the biodiversity scenarios described by this language.

Acknowledgments 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.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

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.