An Extended Complex Event Processing Engine to Qualitatively Determine Spatiotemporal Patterns

Foued Barouni, Bernard Moulin
Département d'informatique et de génie logiciel, Université Laval
Foued.barouni.1@ulaval.ca, Bernard.moulin@ift.ulaval.ca

Abstract

Spatiotemporal events are widely used in data acquisition systems and take an important space in spatiotemporal databases. When spatiotemporal events are combined together and respect a given structure, they form interesting situations called spatiotemporal patterns. Complex event processing has been used to represent such spatiotemporal patterns and detect them from the event cloud. In this paper, we propose an extension of a complex event processing engine for qualitative spatiotemporal patterns. Fuzzy spatial relations are used to express the spatial relationship between events' spatial attributes and to improve expressiveness of patterns.

Keywords: spatiotemporal patterns, events, fuzzy spatial relations, complex event processing, qualitative spatial reasoning.

1. INTRODUCTION

With the emergence of data acquisition systems, spatiotemporal events take a huge space in spatiotemporal databases. Such data may be used by humans or software agents for decision making in several areas (telecommunications, video monitoring, ecology, weather, road safety, etc.) (Sakr, 2010). Data related to spatiotemporal events is often available in digital format (sampling points), but it is not suitable for human reasoning and hence for decision making since it is neither intuitive nor close to natural language. Furthermore, accessing such data from current databases is not easy. Users must often build complex queries using languages with very limited expressiveness (such as SQL) in order to access and process relevant and interesting data. Such interesting data, which is basically a set of spatiotemporal events combined together, is also called situation of interest (Haddad and Moulin, 2010) and can be considered as spatiotemporal patterns when it occurs on a regular basis. According to (Erwig, 2003) a spatiotemporal pattern is a set of spatiotemporal events which have a certain configuration and respect a given structure.

In the literature, works on spatiotemporal patterns can be divided in two main approaches: data mining approach and logical approach. The first approach is interested in discovering patterns in databases containing large amounts of data. In other words, data mining is a research area that essentially aims at proposing statistical techniques allowing a user to discover interesting data structures (called patterns) in databases. Once found, these patterns (or structures) are stored in specific tables called "Field View" existing in spatiotemporal databases. Further details and a review of this approach can be found in (Erwig, 2003). The second approach (logical approach) focuses on the spatiotemporal pattern itself and proposes models for pattern definition, querying and reasoning. In this approach several works propose languages to build spatiotemporal queries, assuming that spatiotemporal data (not necessarily events) is already structured in databases and patterns are already defined. The proposed query languages do not seem to be efficient because of their lack of generality and expressiveness (Gorawski, 2010). Other works in this approach concentrated on reasoning and detecting spatiotemporal patterns using artificial intelligence techniques. First order logic and predicates have been widely used to express spatiotemporal patterns. Even if these works mitigate the lack of expressiveness of classic query languages, they still remain limited because they are application driven.

Both approaches have some drawbacks. **First**, they focus on pattern definition and detection rather than on processing. When databases receive a large amount of events in a short time frame, detecting patterns by running queries on the databases becomes complicated and inefficient. The research community on geospatial event processing has studied the location based aspects of events and proposed a representation model for geo-spatial events. As an example, Worboys proposed a spatial event representation (Worboys, 2004) and a spatiotemporal event representation (Worboys, 2005). However, these models focused on event representation rather than on event processing. Furthermore, these models only enable the reaction to a single event and do not support to deal with event composition and pattern matching in event histories (Etzion, 2010).

Second, most of pattern representations models are using spatial and temporal relations without considering vagueness and uncertainty. When reasoning about spatial data in real applications, the available information is often partially true or uncertain. Thus, it is necessary to move from numerical representations of spatial and temporal data to representations that are closer to the way user reasons using natural language.

To overcome such limitations, an emerging research community proposes to use a spatial extension of complex event processing engines. Complex event processing (CEP) delivers high-speed processing of many events across, identifying the most meaningful events within the event cloud, analyzing their impact, and taking subsequent action in real time (Wikipedia, 2011).

Originally, CEP has been developed and used in the financial and economic industry to predict market developments and exchange rate trends. CEP patterns emerge from relationships between events attributes *time*, *cause* (causal relation between events) and *aggregation* (significance of an event's activity towards other events). Geospatial event processing requires the extension of CEP to include location attributes (Resh el al, 2010). Spatial relations are used to combine location aware events and to create spatiotemporal patterns.

In the literature, several works proposed spatial extensions to CEP engines. Among them, we note *SpatialRules* (SpatialRules, 2011) which is a complex event processing engine for geospatial data and is compliant with OGC geospatial specifications (OGC, 2011). Event processing is performed through rules that offer geospatial and temporal operators. Geospatial operators are mainly topology or distance based. *GCEP* (GCEP, 2011) is an extension of *ESPER* (Esper, 2011) complex event processing engine that allows the use of OGC Geospatial Functions in the rules for filtering events. The *GCEP* Engine offers 12 topological functions for Esper for Java that conforms to Open Geospatial Consortium standards. *ruleCore CEP* Server (ruleCore, 2011) is a complex event processing engine used for real-time detection of complex event patterns. The system is scalable and can be used to implement event driven architecture solutions. *ruleCore* enables defining rules using location information. Location data can be collected from GPS or other sensors and can be natively processed by *ruleCore*. The engine allows stream creation based on events coming from specific geographic zones.

Although these solutions support the processing of spatiotemporal events, they do not deal with uncertainty and vagueness and therefore they are not close enough to users' qualitative reasoning which is essentially based on natural language, using Zadeh's linguistic variables (Zadeh, 1965). In this paper, we propose an extension of a complex event processing engine by the introduction of fuzzy spatial relations between events to enable the definition of qualitative spatiotemporal patterns. In particular, we address the fuzzy distance relation between two events.

The paper is structured as follows: Section 2 provides a brief review on qualitative proximity formalism. Section 3 describes the distance calculation method and our spatiotemporal patterns model. Section 4 describes the architecture that we propose to extend the complex event processing engine and outlines a short case study. Section 5 concludes the paper and discusses possible enhancements of the proposed solution.

2. QUALITATIVE PROXIMITY FORMALISM

Spatial information is an important component of human reasoning. Thanks to geographic information systems (GIS), it becomes possible to access databases containing large amounts of spatial data. This information is now routinely used by humans for daily reasoning (parking a car, path planning, driving, walking, etc) and by companies to support decision processes (i.e. finance, security monitoring, utility, path planning, etc). Although GIS solutions provide tools for data retrieval, they remain limited since most of them are based on numerical methods and require a user to be knowledgeable in specialized areas such as data processing and query languages such as SQL. Hence, current GIS do not offer mechanisms and functionalities naturally compatible with human spatial reasoning which is essentially based on qualitative data taking into account uncertainty and vagueness. Most people find numerical methods non intuitive. For example, a proposition such as "the building is at latitude 30 minutes, 10 degrees, etc." is less relevant than "the building is in front of the Emirates Stadium".
Guesgen and Albrecht (Guesgen and Albrecht, 2000) proposed a new approach called Qualitative Proximity (QP) formalism in order to reason about distance relationships between spatial objects. A distance relationship between two spatial objects is a linguistic variable which is used in a statement as follows: the distance relation between Object A and Object B is the *linguistic variable*. Generally speaking, a linguistic variable is a variable whose values are words or phrases in a natural language (Guesgen, 2004). The values of a linguistic variable are called linguistic values. Guesgen's model considered the following set of linguistic variables as possible relationships between two spatial objects: {touching, very near, near, moderately near, moderately far, far, very far}. Although linguistic variables allow expressing qualitative spatial relations, they are not sufficient to integrate qualitative and quantitative spatial reasoning. To overcome this limitation, fuzzy logic (Zadeh, 1965) has been combined with qualitative methods to address the issue of vagueness of spatial information. Fuzzy sets can be used to associate quantitative and qualitative information. Formally, a fuzzy set F of a domain D is a set of ordered pairs, $\{d, \mu_F(d)\}$, where $d \in D$ and $\mu_F : D \rightarrow [0, 1]$ is the membership function of F. The membership function replaces the characteristic function of a classical subset F $\subseteq D$, which maps the set D to $\{0, 1\}$ and thereby indicating whether an element belongs to F (indicated by 1) or not (indicated by 0). (Shultz et al, 2006).

Guesgen proposed to assign a membership function to every distance relationship and to calculate the truth degree of the relationship between two spatial objects. For example, if Object A is *near* Object B, the membership function will be greater than 80%. The membership function of the relation *very far* about the two spatial objects will be 0%. Hence, the further away a relation is from the reference relation, the lower its membership grade (Shultz et al, 2006).

3. A MODEL FOR SPATIOTEMPORAL PATTERNS

In this section, we present a model that we propose to represent spatiotemporal patterns. First, we introduce a distance calculation method that will be used to determine spatial relationships between spatial objects. Then, we propose a general definition of spatiotemporal events before presenting our spatiotemporal pattern model.

3.1 Distance calculation method and proximity

Researchers of various disciplines, including GIS, have studied how human assess distance and proximity. According to Bloch (2003), numerous factors influence people's perception of distance. For example, temporal measures lead to distance expressed as travel time. Economic measures in terms of travel cost are also considered. The perception of distance between objects also depends on the presence or absence of other objects in the environment. Guesgen and Albrecht (2000) referred to the notion of *absolute distance* which is the most intuitive form of reasoning about proximity and simply based on the absolute distance between two objects. Furthermore, *relative distance* can be used when reasoning about the proximity of two objects: it is evaluated by considering the other objects as being part of a given environment. For example, considering the distance from Quebec to Montreal which is qualified by *near*, and the distance from Montreal to Miami which is qualified by *very far,* will help a resident in Quebec to conclude that the distance from Quebec to Miami is *very far*. The Euclidian distance is one of the simplest ways to compute the absolute distance between spatial objects (Guesgen and Albrecht, 2000). For the rest of this paper, we consider the absolute distance to evaluate proximity between two objects and the Euclidian distance for calculation purposes. Note that our model remains open to integrate other methods to evaluate distance relationships between spatial objects.

Given a 2D spatial object A with coordinates (x_a, y_a) and a spatial object B with coordinates (x_b, y_b), the distance between the two objects is given by the formula:

$$\text{Distance(A,B)}=\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

Once distance between two objects is calculated, a membership value can be assigned to each distance relationship. As illustrated in Figure 1, every spatial relationship between the spatial object A and the spatial object B has a membership value based on the absolute distance between the two spatial objects.

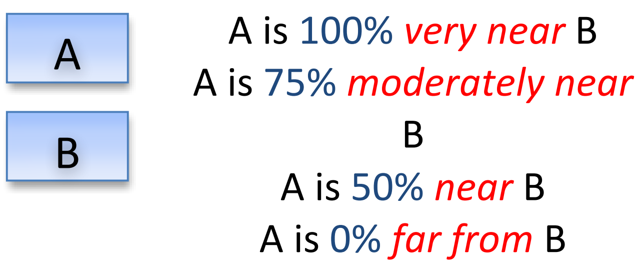

Figure 1: Spatial relationships between two spatial objects with associated membership values

3.2 Spatiotemporal events

As defined by Luckham (Luckham, 2002), events are "*something that happens in reality*". Haddad and Moulin (Haddad and Moulin, 2010) defined spatiotemporal events as "*events that happen in a certain location and at a certain point in time*". Hence, spatiotemporal events must have spatial and temporal attributes. From a design and implementation perspective, it is interesting to define a general structure of events. When events are received by an event processing system and have similar structure and meaning, we say that they belong to the same *event type*. In the literature on event processing systems, event types have a common general structure but they can be customized by the user who can change event attributes. The general structure of an event type can be divided into three main parts: Header, Payload and Event Relations. Figure 2 shows the general structure of the definition element. Meta-information about the event, like its type, occurrence time and detection time is contained in the event header. Specific information about the event itself is contained in the event payload. The last part of the event definition is the event relation which is optional. It allows defining a semantic relationship between different event types. According to Etzion (2010), we can consider four types of relationships between event types: membership, generalization, specialization and retraction.

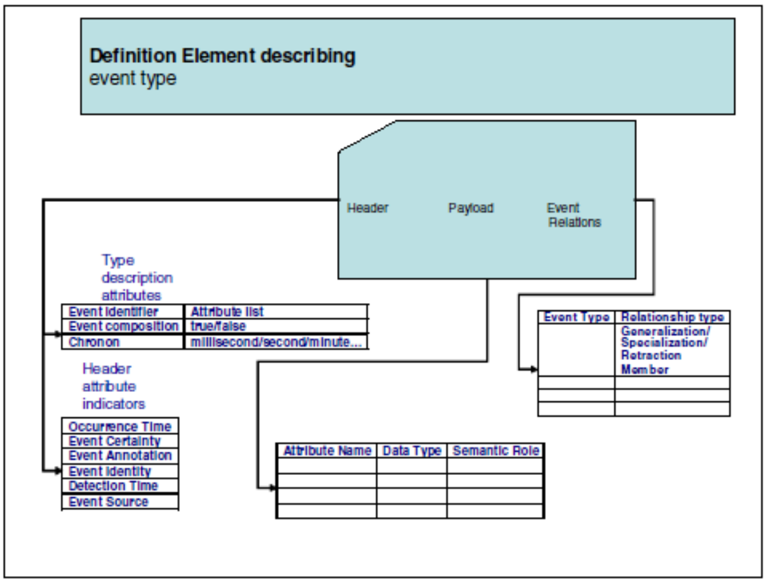

Figure 2: general definition of event type (Etzion, 2010)

Since we are interested in spatiotemporal events, spatial and temporal event properties are the most important elements that we consider in the event attributes and they will be introduced in the following section.

Event temporal aspect: According to Etzion (Etzion, 2010), events may have two main temporal properties: detection time and occurrence time. Detection time is defined as the point in time when the event becomes available for processing. Occurrence time is the temporal point when the event occurs in the real time. Usually, occurrence time is earlier than detection time. For the rest of the paper, we assume that detection time and occurrence time are the same.

Event spatial aspect: Events can occur at different types of locations that can be defined as points, lines and areas. Those location types are also part of the payload. An example of point is the 2D GPS coordinates of an object. An example of line is: "24^{th} Avenue east". Finally, an example of area can be: "Emirates Stadium, London".

In event processing systems, events can be correlated by spatial and temporal relationships and can also be aggregated. Spatiotemporal patterns are the results of such operations on spatiotemporal events.

3.3 Spatiotemporal Patterns:

According to Erwig (Erwig, 2003), interesting phenomena are not random but rather follow certain rules and structures. When spatiotemporal events are combined, related and comply with a certain configuration, they form so-called *spatiotemporal patterns*. In other words, a spatiotemporal pattern is a set of events combined together using temporal and spatial relationships. A simple example of a spatiotemporal pattern is a single event with spatial and temporal attributes. Another simple example of a spatiotemporal pattern is composed of an event which has a spatial attribute qualified by a spatial relation with respect to an external spatial object (Figure 3-a). Another example of a simple spatiotemporal pattern is an event which has a temporal attribute qualified by a temporal relation with respect to a time reference (Figure 3-b).

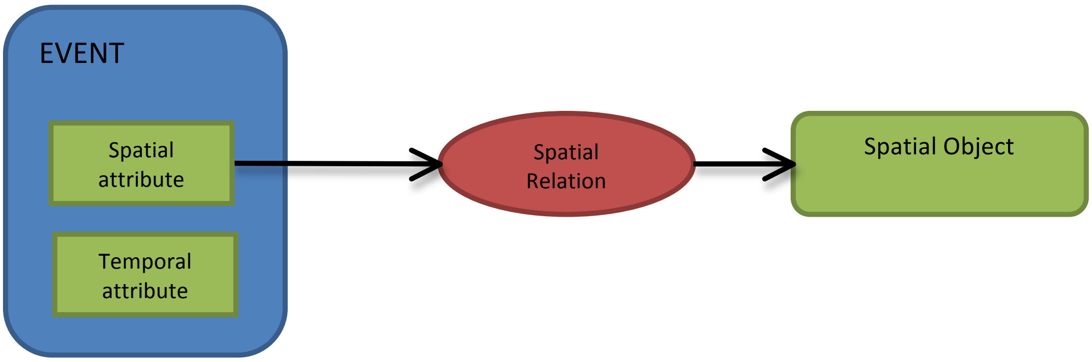

Figure 3-a: a simple spatiotemporal pattern

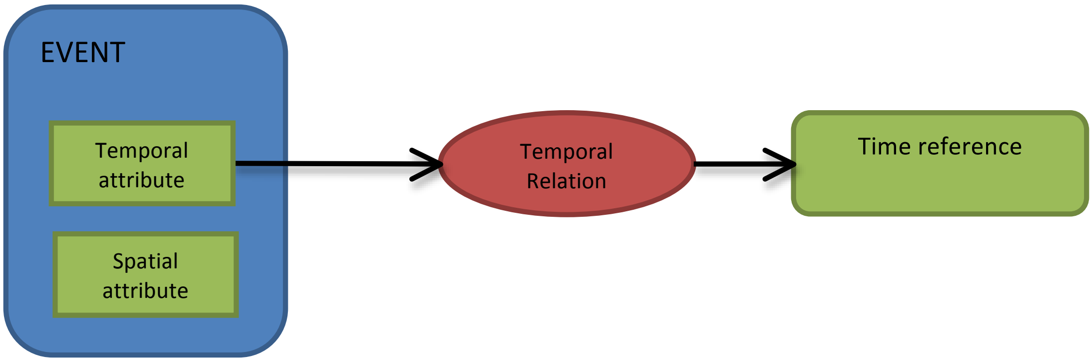

Figure 3-b: a simple spatiotemporal pattern

Figure 3-c: a simple spatiotemporal pattern

Here is an example of a simple spatiotemporal pattern expressed in English: A car accident ***far*** from Quebec City (Figure 3-c). The event accident occurred in Montreal, and its location can be qualified as being far from Quebec City.

Formally, we define a spatiotemporal pattern as follows:

$$P = (E, TR, SR, L, T)$$

Where:
-***P*** is the spatiotemporal pattern.
-***E*** is a set of spatiotemporal events.
-***TR*** is a set of temporal relations between events. In our model, we consider Allen's temporal relations (Allen, 1983). They are also supported by the *Esper* complex event processing engine. Note that the temporal relation between two events is obtained from the relation between the two events' time attributes.
-***SR*** is a set of spatial relationships between spatial objects. Three types of spatial relations are defined in the literature: distance relations, direction relations and topologic relations. Note that a spatial relation between two events can be specified as a direction relation between the two events' spatial attributes.
-***L*** is a set of spatial locations. An event can be qualified with respect to a given spatial location and forms a simple spatiotemporal pattern (as illustrated in Figure 3-a).
-***T*** is a set of time references. An event can be qualified with respect to a given time reference and forms a simple spatiotemporal pattern. A simple pattern defined as: "*Accident event before noon*" will match when a car accident event occurs at 11:00 AM (as illustrated in Figure 3-b).
Figure 4 describes a generic form of a spatiotemporal pattern.

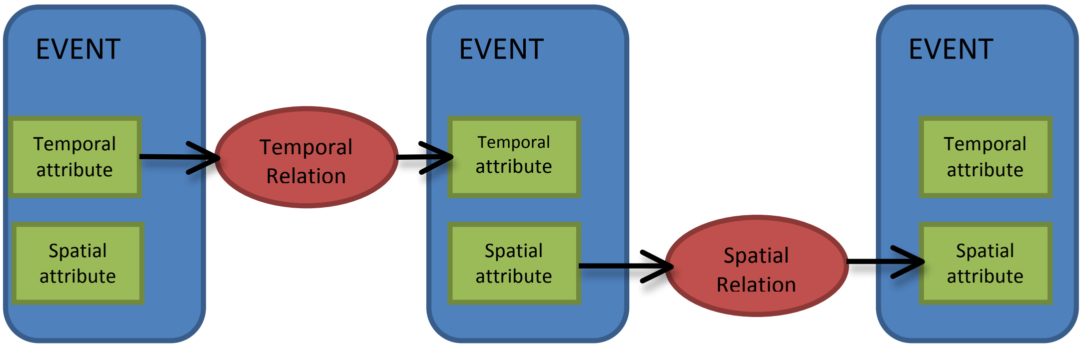

Figure 4: a generic form of a spatiotemporal pattern

4. ARCHITECTURE AND CASE STUDY:

In this section, we propose an implementation model of our qualitative extension of the complex event processing engine, followed by a case study.

4.1 Architecture:

We developed a spatial extension of a complex event processing engine to support qualitative spatiotemporal patterns based on the spatiotemporal pattern definition presented in the previous section. The proposed architecture needs to respond to several requirements:
-*Processing events*: events will be received from the environment. They need to be managed, classified and processed.
-*Pattern definition*: spatiotemporal patterns will be defined using a specific language and will be stored in a knowledge base. They will be accessed by the application during pattern matching operations.
-*Temporal operators*: they will be used for temporal relations between events.
-*Fuzzy spatial operators*: they will be used to express spatial relations between events.

-*GIS* compatibility: spatiotemporal patterns are defined in relation to an environment. Events must be connected to spatial locations from a given GIS.

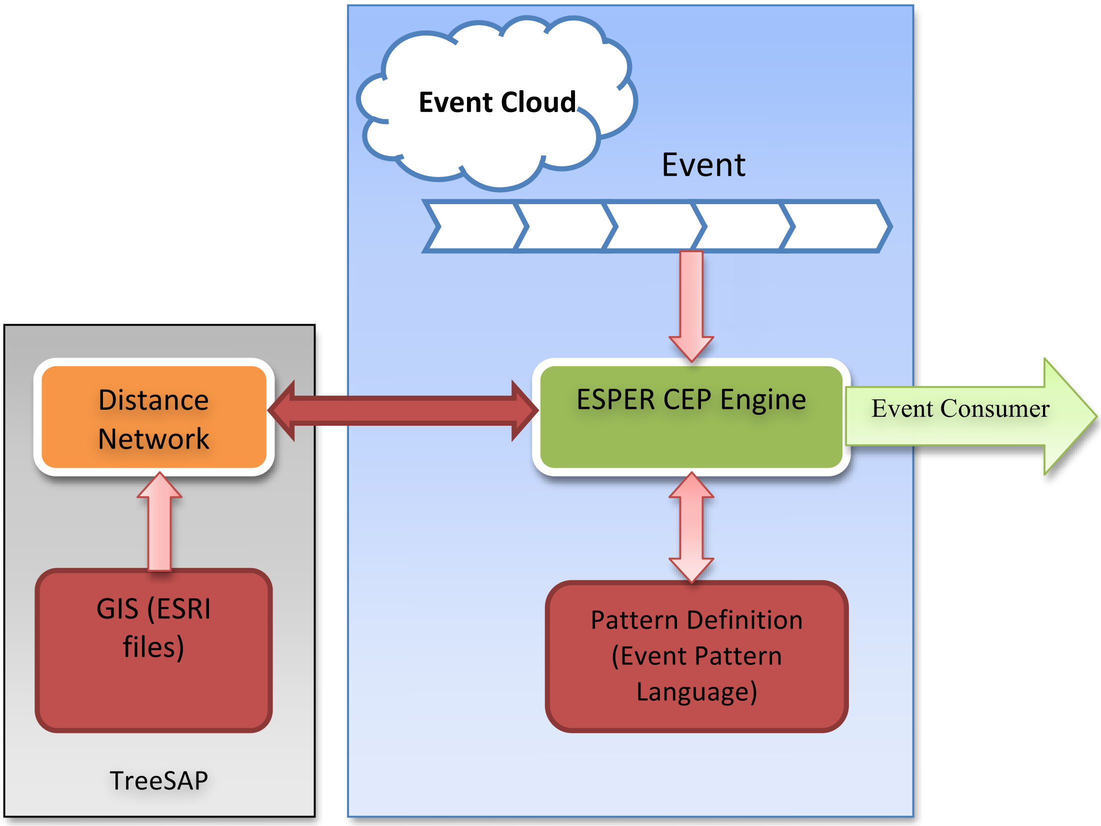

Figure 5: overview of qualitative spatiotemporal pattern processing architecture

To achieve the requirements listed above, we propose a-two levels architecture (Figure 5). The first level is based on *TreeSAP*, an application developed by Guesgen's team which implements qualitative spatial relations formalism based on fuzzy logic. *TreeSAP* offers a graphical user interface written in Java and uses the *JTS* (Java Topology Suite) framework for geographic data manipulation. Using *TreeSAP*, a user is able to load a set of shape files (ESRI format). The application engine builds a set of spatial relationships with membership grades between every pair of spatial components loaded from the shape files. This operation creates a distance network which contains all possible distance relationships between the loaded GIS components. Note that the framework that comes with *TreeSAP* allows for an extension of spatial relations to include fuzzy topology relations using an extension of Allen's formalism and using of fuzzy logic (Guesgen , 2004). Accessing the distance network is possible using a query mechanism that returns the spatial relations between two spatial objects with related membership grades.

The second level of our architecture is based on *ESPER*, an open source framework for complex event processing (Esper, 2011). It provides a rich declarative language for pattern definition called Event Pattern Language (EPL) which is SQL based and offers all SQL operators extended with temporal operators according to Allen's temporal logic (Cugola, 2010). The Esper Pattern language and engine are available with Java and .Net (NEsper).
Spatiotemporal patterns are defined in the *ESPER* knowledge base pattern and they are used by the pattern matching process. As events are received with spatial attributes and need to be evaluated, we propose to make a connection between the

ESPER CEP engine and the Distance network of *TreeSAP* in order to evaluate spatial relationships between event locations in spatiotemporal patterns.

4.2 Workflow:

The user starts by loading shapes files into the *TreeSAP* application. A new distance network is generated and contains all fuzzy distance relations between spatial objects. The ESPER engine is initialized and ready for event processing. When events are detected by the ESPER engine in the 'event cloud', they are directed to streams defined according to the user's preferences. The CEP engine uses patterns defined by the event pattern language from the pattern knowledge base and the spatial information from distance network to perform pattern matching operations.

4.3 Case study: Fiber optic network monitoring solutions:

Fiber optic networks are becoming one of the most important technologies used by the telecom industry. With the high demand for bandwidth, telecom companies use fiber optic networks for data transmission and to provide high quality services to business and individual customers. The large deployment of such networks and the need to reduce the mean time to repair (MTTR) to ensure optimal quality of service, led research communities to offer fiber optic monitoring solutions. The Optical Time Domain Reflectometer (OTDR) is one of the technologies used for fiber optic characterization. It helps to verify fiber status by detecting possible degradations and breaks and by identifying different network components.
OTDR technology is currently used by the SCADA systems (supervisory control and data acquisition) and implemented in local sensors (or remote test units, (RTU)). When sensors detect a new event in the fiber (degradation, break, etc.), they send alerts to the SCADA server which is responsible to dispatch information to the end-user. With the development of domain oriented GIS applications, a new generation of monitoring solutions offer to map fiber events into a GIS interface that offers an inventory of fiber optic networks and allows visualizing events generated by sensors and collected by the SCADA server. The following Figure 6 illustrates an overview of a network monitoring solutions architecture.

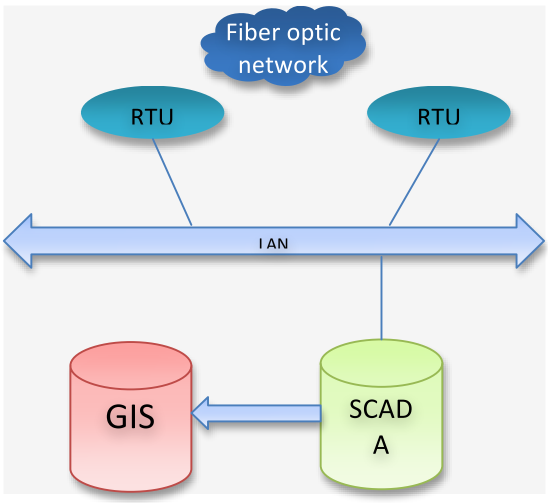

Figure 6: an example of fiber optic network monitoring solution

Since monitoring solutions generate events that need to be mapped into a GIS, the model that we propose can help end-users define and detect interesting patterns.

Several situations may interest a user monitoring the reception of spatiotemporal events from SCADA. For example, when a degradation event is followed by another degradation event on the same fiber and the two events locations are close, the user may decide to send a technical team to perform a fiber inspection and avoid any risk of fiber break. An example of such spatiotemporal pattern is illustrated in Figure 7:

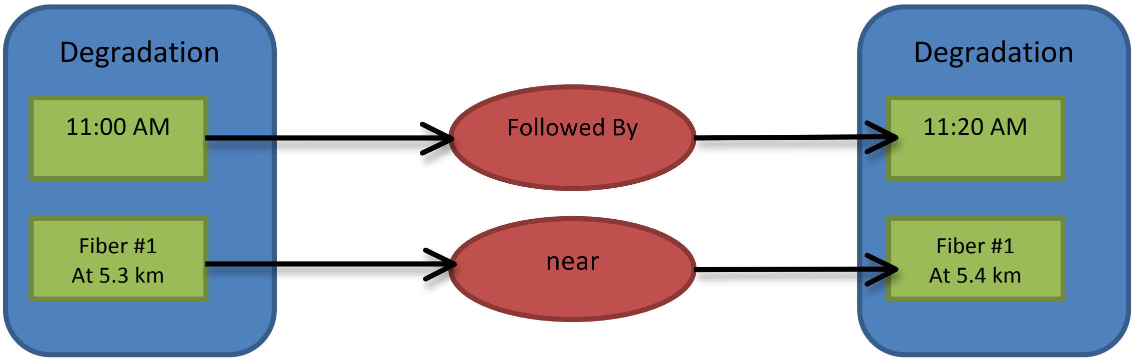

Figure 7: an example of spatiotemporal pattern

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an extension of a complex event processing engine thanks to the introduction of fuzzy spatial relations between events to enable the definition of qualitative spatiotemporal patterns. In particular, we addressed the fuzzy distance relation between two events. As mentioned in the previous sections, spatiotemporal patterns represent situations of interest. There are several studies in the literature which define two types of spatiotemporal situations: static and dynamic (see (Haddad, 2009) for a complete review of such studies). Static situations express stability and cannot account for change. They are generally used to describe a static state of an object. Dynamic situations are situations that change something in the universe and describe movements or spatiotemporal changes of states of a given object.

Hence, the notion of change is an important aspect of spatiotemporal situations. The model that we currently propose does not take into account this aspect. An example of such a change is when users are interested to include the evolution of the state of a fiber optic from degradation to break in the spatiotemporal pattern. As a future work, we aim to propose an extension of our model to allow representing and processing spatiotemporal patterns including the notion of change.

We also intend to extend the proposed model by introducing fuzzy topological relations. Finally, we will introduce a knowledge base that allows enhancing pattern expressiveness and processing as it was stated in Section 3.2.

REFERENCES

Allen, J.F (1983). Maintaining Knowledge about Temporal Intervals. *ACM Common*, 26(11):832–843

Cugola G. and Margara A. (2011). Processing flows of information: From data stream to complex event processing. http://event-based.org, [accessed 15 November 2011]

Erwig, M. (2003).Toward spatiotemporal queries. *Spatiotemporal Databases. Springer, 29-54*.

Esper (2011). http://esper.codehaus.org/ [accessed 15 November 2011]

Etzion O. and N. Zolotorvesky.(2010). Spatial Perspectives in Event Processing, Event Composition, Correlation, Detection. *Springer Verlag* pp. 85–107.

Gcep (2011). www.surna.org [accessed 15 November 2011]

Gorawski M., Jureczek P. (2010), "A Proposal of Spatio-Temporal Pattern Queries," *International Conference of Complex, Intelligent and Software Intensive Systems, February 15-February 18 2010, Krakow, Poland*, pp. 587-593.

Guesgen H.W. and Albrecht J. (2000), Imprecise reasoning in geographic information systems, *Fuzzy Sets and Systems (Special Issue on Uncertainty Management in Spatial Data and GIS)*, 113, pp. 121–131.

Guesgen H.W. (2004), Fuzzy reasoning about geographic regions, *Fuzzy modeling with spatial information for geographic problems*, pp. 1-14.

Haddad H. (2009). Une approche pour supporter l'analyse qualitative des suites d'actions dans un environnement géographique virtuel et dynamique. L'analyse « What-if » comme exemple, *PhD Thesis*, Laval University, Quebec, Canada.

Haddad H. and B. Moulin (2010). A framework to support qualitative reasoning about COAs in dynamic spatial environment, *Journal of Experimental & Theoretical Artificial Intelligence,* 22(4): 341-380.

Luckham, D.C. (2002): *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*: Addison-Wesley.

Open Geospatial Consortium (2011). www.opengeospatial.org [accessed 15 November 2011]

Resch, B., Lippautz, M. and Mittlboeck, M. (2010) Pervasive Monitoring – A Standardised Sensor Web Approach for Intelligent Sensing Infrastructures. Sensors. Special Issue "Intelligent Sensors 2010", pp. 11440-11467.

Rulecore (2011). www.rulecore.com [accessed 15 November 2011]

Sakr M., Guting R., (2010). Spatiotemporal Pattern Queries, *Geoinformatica* 15(3): 497-540

Schultz, C.P.L. Guesgen, H.W. and Amor, R. (2007). A System for Querying With Qualitative Distances in Networks, *Fuzzy Systems Conference*, 2007. FUZZ-IEEE 2007. IEEE International , London, United Kingdom.

SpatialRules (2011). www.objectfx.com [accessed 15 November 2011]

Wikipédia. (2011). www.wikipédia.org [accessed 15 November 2011]

Worboys, M.F. and Hornsby, K. (2004). "From objects to events. GEM, the geospatial event model". *GIScience*, Springer, 3234:327-344

Worboys, M.F (2005). "Event-Oriented Approaches to Geographic Phenomena". *International Journal of Geographical Information Systems*. 19:1-28

Zadeh, L (1965). Fuzzy sets. *Information and Control*, 8:338–353

Proceedings of Global Geospatial Conference 2012
Québec City, Canada, 14-17 May 2012