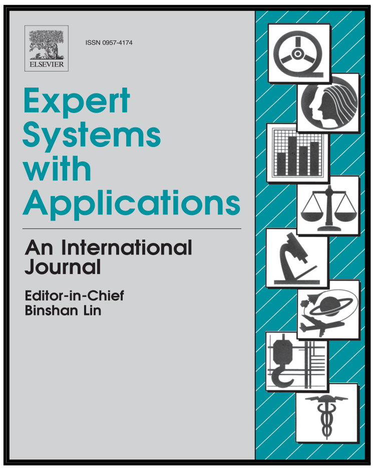
Accepted Manuscript



Enabling Situation Awareness with Supply Chain Event Management

George Vlahakis , Dimitris Apostolou , Evi Kopanaki

PII: S0957-4174(17)30688-7
DOI: 10.1016/j.eswa.2017.10.013
Reference: ESWA 11596

To appear in: *Expert Systems With Applications*

Received date: 3 May 2017
Revised date: 4 October 2017
Accepted date: 5 October 2017

Please cite this article as: George Vlahakis , Dimitris Apostolou , Evi Kopanaki , Enabling Situation Awareness with Supply Chain Event Management, *Expert Systems With Applications* (2017), doi: 10.1016/j.eswa.2017.10.013

Highlights

- Supply chain event management refers to methods that process supply chain events.
- We present a graph approach to represent supply chain events and their relations.
- We describe a framework that implements event correlation and situation detection.
- Key benefits include capability to adapt to dynamic conditions and scalability.

Enabling Situation Awareness with Supply Chain Event Management

George Vlahakis
University of Piraeus
Karaoli & Dimitriou 80
Piraeus 18534, Greece
g78vlahakis@hotmail.com

Dimitris Apostolou
University of Piraeus
Karaoli & Dimitriou 80
Piraeus 18534, Greece
dapost@unipi.gr

Evi Kopanaki
University of Piraeus
Karaoli & Dimitriou 80
Piraeus 18534, Greece
evik@unipi.gr

Abstract Supply chain event management exploits synergies between IT and logistics and refers to the set of methods and technologies used to efficiently integrate events from all actors and processes of the supply chain. In the context of supply chain event management, we examine how events can be used to leverage situation awareness. In our approach, situation awareness is facilitated by providing the capability to detect situations, which are represented as correlations between simple events, complex events and supply chain objects (e.g., suppliers, 3PL companies, retailers and material resources). We introduce a two-phase event correlation method which first correlates simple events into complex events and then events with supply chain objects. We describe how the proposed model has been implemented in a software framework and we conduct evaluation tests to examine its situation detection capabilities.

Keywords supply chain management; supply chain event management; situation awareness; event processing

Abbreviations Supply Chain Management (SCM), Supply Chain Event Management (SCEM), Complex Event Processing (CEP), Supply Chain (SC), Situation Awareness (SA), Latitude (LAT), Longitude (LONG), Event Listener (EL), Correlation Agent (CA), Key Performance Indicator (KPI), Event Description Language (EDL), Service Oriented Architecture (SOA), Continuous Query Language (CQL)

1 Introduction

Market uncertainty and intense competition forces organisations to search for ways to improve their responsiveness, effectiveness and quality of products and services to customers. To acquire strategic advantage, increase corporate profit and improve market share, companies need to replace 'standalone' business strategies with strategies that include their trading partners (Iansiti & Levien, 2004). Therefore, they need to collaborate closely with their suppliers or customers and try to improve processes at the level of their supply chain. A supply chain can be described as consisting of all organizations that are involved in the successive stages of design, manufacturing, distribution, marketing and retailing of a product or a service (Holland, 1995).

Supply Chain Management (SCM) is one of the key factors aiming to enhance organizational effectiveness (Park et al., 2005) and operational efficiency along the supply chain, from raw materials through first and second-tier suppliers to final customers. As a term, SCM first appeared in the early 1980s and has been in widespread use ever since. It involves the efficient management of materials, processes and information along the whole supply chain. It requires control and coordination activities, as well as information flow between heterogeneous information systems (Ryoo & Kim 2015; Marra et al. 2012).

As supply chain processes become highly data intensive, increasingly more events are generated and consumed by supply chain actors, systems and processes. Events can affect the process flow and execution by influencing relevant sub-processes e.g., an order submitted, a product shipped, a transportation truck wracked, raw material stock reduced, etc. The increasing association of events with supply chain processes has led developers and researchers, form both fields of logistics and IT, to coin the new term Supply Chain Event Management (SCEM) referring to the set of methods and technologies used to efficiently integrate all events involved in the planning, production and distribution of the materials and products in the supply chain so as to satisfy customers expectations (Ijioui, Emmerich & Ceyp 2007; Knickle and Kemmeter, 2002). These events may be happening across the various stages of a supply chain. Or, they may be news items, traffic reports, weather reports, or other kinds of data affecting supply chain operations.

SCEM benefits from event processing methods and technologies for tracking and processing streams of data about supply chain events and deriving a conclusion from them. Complex Event Processing (CEP) is event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. The premise of CEP is to provide organizations with a new way to analyse supply chain patterns in real-time and help the business side to make decisions based on a real-time analysis of incoming data streams. Specifically, decision making is facilitated with rules that enable reaction to complex events sampled from across event streams, not just a single stream.

But CEP can do more for SCEM than just reacting to streams of data about the status of a supply chain entity, towards determining the situation of a supply chain entity. Consider, for example, an airplane cruising at a certain altitude: in addition to information about its mechanical sub-systems, its situation may include information about current weather conditions, forecasted weather at destination and other significant information that potentially concern the flight and its crew. A CEP-based approach can leverage situational awareness by processing all relevant event data over a wide-enough time window and correlating them with relevant domain knowledge.

Situation awareness refers to the persecution of the elements in the environment with a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" (Adi and Etzion, 2004). To realize systems for situation awareness, individual pieces of raw information (e.g. sensor data) should be interpreted into a higher, domain-relevant concept called situation, which is an abstract state of affairs interesting to specific applications. In the context of enterprise computing systems, situation is defined as an event occurrence that might require a reaction (Edsley 2016).

Our work focuses on utilizing CEP in the context of SCEM to detect the operational situation of supply chain objects, such as suppliers, 3PLs and material resources and to propose an SCEM software framework for the development of SCEM systems that will be able to deal with the following aspects: (i) Situation awareness through the detection of situations at different stages of the supply chain; (ii) Event correlation at different time windows; (iii) Scalability of the framework and capability to represent complex supply chains.

This paper is structured as follows: Section 2 presents a literature review of SCM's main objectives and challenges such as supply chain flexibility, vulnerability, risk management as well as related works in situation awareness and event management in supply chain management. Section 3 presents the conceptual approach of situation awareness that is the base of our framework development. Section 4 presents the architecture of our SCEM framework and describes its main components and their basic functionalities. A technical description of the framework is described in section 5, while an evaluation of the framework is presented in section 6. The final section of the paper includes our concluding remarks and the description of our future work.

2 Literature Review

Regardless of the industry, SCM objectives and challenges are similar. The most commonly referenced SCM objectives are improved demand planning (Frochlich & Westbrook, 2002), reduction of lead time (Buxmann et al., 2004), speed of delivery, control and reliability (Ross, 2004; Schnetzler, Nobs, & Sennheiser, 2004; Slack et al., 2004), quality of service (Hoffman & Mehra, 2000) and reduction of costs (including production, transportation and purchase costs (Buxmann et al., 2004), as well as inventory-driven costs and product returns costs (Callioni et al., 2005)). Additional objectives include tight coordination of activities (Lee, 2002), risk management (Christopher & Lee 2004; Ho et al. 2015) and resilience (Christopher & Peck 2004; Brandon-Jones et al. 2014).

SCM objectives and challenges are easier to meet through improved collaboration (Scholten, & Schilder, 2015) and exchange of information between trading partners. Improved collaboration is further facilitated through the use of technology, which integrates business processes along the supply chain, facilitates the processing of information, and supports the sharing of knowledge between trading partners (Malhotra et al., 2005; Premkumar, 2000). The use of appropriate information technology does not only improve the efficiency of the supply chain, but also increases its ability to deal with uncertainty.

Uncertainty is one of the main challenges that SCM has to face (Hsu, 2005; Vilko et al. 2014; Ho et al 2015). Davis (1993) distinguishes between manufacturing uncertainty, demand uncertainty and supply uncertainty. Manufacturing uncertainty refers to machine breakdowns, computer errors and poor product designs that may affect the quality of products and cause disruptions or bottlenecks in production. Demand uncertainty is related to changes in demand that may lead to out-of-stocks or excessive inventory. It is also related to irregular orders that may lead to wrong forecasts and ineffective production planning. Lee (2002) distinguishes between markets where demand uncertainty is high, e.g., markets of innovative products or the fashion industry, where demand is subject to the vagaries of fashion, and markets where demand uncertainty is low, e.g., industries of essential products or utilities consumption. He also examines supply uncertainty, i.e., inability to supply on time due to defective material or unexpected problems during, e.g., transportation and distinguishes between low uncertainty or stable processes, e.g., basic manufacturing, and high uncertainty processes, e.g. weather-related production, such as farming.

In addition to the ‘everyday’ uncertainties of supply and demand, companies have to face competitive uncertainty, referring to competitors’ moves and strategies (Kopanaki 2014). They also have to face unexpected events, such as earthquakes, wars, price rises, problems with trading partners (Blos et al 2012), accidents or production plants damages.

In an uncertain business environment the exposure of supply chains to disturbances or disruptions has increased (Christopher and Lee 2004). As supply chain disruptions can have significant impact on a firm's short-term performance (Tang 2006), the need to effectively manage or confront potential risks is imperative. This is one of the main reasons leading to the development supply chain risk management (Tang & Musa, 2011). According to Wieland and Wallenburg (2012), supply chain risk management refers to the implementation of strategies with the objective of managing both everyday and exceptional risks along the supply chain. A risk management process can be represented as a cycle of risk identification, evaluation, control and monitoring (Tummala, & Schoenherr, 2011; Ho et al. 2015). It is based on continuous risk assessment aiming to reduce supply chain vulnerability (Wieland and Wallenburg 2012).

According to lahmar et al. (2016) the concept of supply chain vulnerability has been defined by many authors, leading to varied explanations. One of the most commonly used definitions is given by Svensson (2000) who explains it as the existence of disturbances leading to deviations from normal, expected or planned schedules or activities, with negative effects or consequences. The problem of vulnerability, both to everyday changes and more rare events, strongly implies a need for resilience (Pettit et al. 2010; Brandon-Jones et al. 2014) in supply chains. Resilience is an evolving concept and differs from traditional risk management (Pettit et al. 2010) as it does not focus on the assessment and management of risks, but on the ability of the organizations or supply chains to respond or absorb environmental disturbances. Supply chain resilience can be defined as the ability of a supply chain to return to normal operating performance, within an acceptable period of time, after being disturbed (Christopher & Peck, 2004; Brandon-Jones et al. 2014). "Resilient supply chains incorporate event readiness, are capable of providing an efficient response, and often are capable of recovering to their original state or even better post the disruptive event" (Ponomarov and Holcomb, 2009 p.124). Therefore, a resilient supply chain must be able to handle a large numbers of events, both expected and unexpected. According to Liu et al. (2007) the unexpected events are also called exceptions, as they are not part of the supply chain planning. Examples of exceptions are product out-of-stock, production faults, shipment delays, truck accidents and machine breakdowns.

The increased need of supply chains to respond to events (especially to exception events) dynamically and in an efficient way has led to the development of the new discipline of SCEM. The goal of SCEM is to introduce a control mechanism for managing events, in particular, exception events, and responding to them dynamically. Transparency, information sharing and supply chain visibility are critical factors for effective SCEM (Dießner and Rosemann, 2008; Baader and Montanus, 2008). According to Otto (2003), SCEM has received little attention as a field of academic research. To explain SCEM, he analyses it from three complementary perspectives: as a management concept, as a software solution and as a software component. Our research combines these perspectives and develops a conceptual approach and a software framework to support situation awareness, in the context of SCEM.

Similar to other domains (see for example Machado et al. 2017; Cimino et al. 2012; Feng et al. 2009), achieving situational or situation awareness in a supply chain requires a system that is capable of sensing the operational environment, including supply chain participants and objects, their interactions and changes of their state. This is not an easy task for a system to accomplish and the majority of existing approaches lack of capability of handling events that are originated not only from within the Supply Chain (SC) itself but also from the entire operational environment of the SC. Hence, our primary motivation has been to design a framework that could be capable of detecting, processing and managing events not only from the SC but rather from exogenous factors that affect the operation of the SC. Moreover, the framework should enable the development and deployment of software systems or services that would make use of processed events.

Events affecting a SC can appear from sources outside the SC and their affecting duration can exceed a few minutes or hours. Stream CEP systems are capable of processing events for a limited time window and require queries with high computational cost. The extended use of graph theory in social engineering (El Kassiri & Belouadha, 2015; Heer & Boyd, 2005; Lawrence & Latha, 2015) inspired us to consider SC events as graph nodes and their relations as edges composing a network that could describe the SC environment. In other words, our work focuses on utilizing CEP and graph databases, in the context of SCEM, to determine the operational situation of supply chain objects and to identify the operational requirements and the features needed for the development of SCEM systems.

Liu et al. (2007) proposed time and coloured Petri nets to represent case data as a formalism for managing supply chain events. They designed seven basic patterns to capture modelling concepts that arise commonly in supply chains. They argue that these basic patterns may be used by themselves and also combined to create new patterns. To demonstrate this, they combined patterns to build a complete Petri net and analysed it using dependency graphs and simulation. In addition, they used simulation to analyse various performance indicators (e.g., fill rates, replenishment times, and lead times) under different strategies.

Simulation is used by many researchers to study SCEM. Longo and Mirabelli (2008) describe a flexible and parametric simulation model, starting from an event simulation package. This model can be used as a decision making tool, since it enables supply chain managers to analyse different supply chain scenarios by changing input parameters (e.g., inventory policies, lead times, forecast methods and demand variations) and to observe the effects of such changes on multiple performance measures.

To address event management many researchers have suggested agent-based approaches. To detect and predict disruptive events along a schedule execution, Fernández, et al. (2015) presented an agent-based approach for the implementation of a service - oriented monitoring subsystem. This subsystem is part of an integral service-oriented architecture for collaborative management of disruptive events in supply chains. It can be used by supply chain members and can provide monitoring services with two main functionalities: collection of data related to supply process execution; processing and assessment of collected data to detect and/or anticipate disruptive events. Bearzotti et al. (2012) presented a collaborative, distributed agent-based approach for SCEM, aiming to perform autonomous corrective control actions, to minimize the impact of deviations on a plan under execution. Autonomous control actions minimize the disruptive event effect by distributing the variation between supply chain members, using the plan's slack in a collaborative way. Finally, Bodendorf and Zimmermann (2005) also used agent technology, but for implementing proactive SCEM system. These scholars argue that proactive SCEM solutions can substantially reduce supply-chain troubleshooting costs.

3 Supply Chain Situation Awareness Model

A generally accepted definition of Situation Awareness (SA) was introduced by Mica Endsley (2016; 2008) describing it as "the persecution of the elements in the environment with a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" (see also Endsley & Connors 2008; Endsley, 1995; Magoutas, et al. 2011). SA is based on the perception of the operational environment where time, space, objects and their state are well defined. SA involves being aware of what is happening in the vicinity to understand how information, events, and one's own actions may impact goals and objectives, both immediately and in the near future. A human with an adept sense of situation awareness generally has a high degree of knowledge with respect to inputs and outputs of a system, an innate 'feel' for situations, people, and events that play out because of variables the subject can control. Lacking or inadequate situation awareness has been identified as one of the primary factors in accidents attributed to human error (Nullmeyer et al. 2005). Thus, SA is especially important in work environments such as SCs where the information flow can be quite high and poor decisions may lead to serious consequences such as delaying delivery of health & safety critical products.

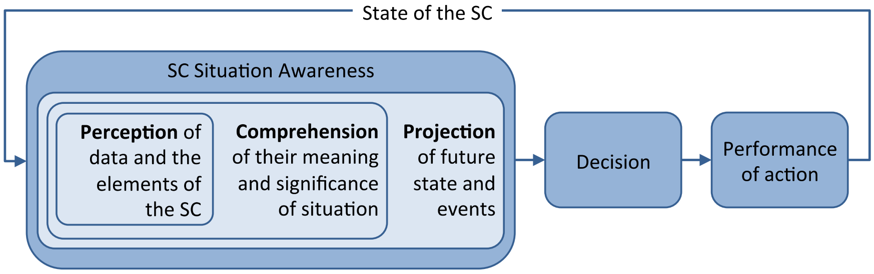


Fig.1. SC Situation Awareness Model (adapted from Endlesey, 2016; 1995)

According to the given definition, SA can be considered as a goal-directed process in which certain steps should be taken: a) information detection and collection; b) situation detection; c) prediction of new situations using knowledge from the past situation analysis; and d) implementation of actions to achieve the desired goal within certain constraints. These steps are depicted in the supply chain situation awareness model (fig. 1), which is based on the model proposed by Endlesey (2016; 1995). In this model, the first step in achieving SA is to perceive the status, attributes, and dynamics of relevant objects in the SC environment. Thus, perception involves the processes of monitoring, event detection, and simple recognition, which lead to an awareness of multiple SC objects (people, systems, environmental factors) and their current state (locations, conditions, modes, actions). Comprehension, the next step in SA involves a synthesis of disjointed perception elements through the processes of event pattern recognition, interpretation, and evaluation. Comprehension requires integrating this information to understand how it will impact upon SC objects. Projection, the third step of SA involves the ability to project the future actions of the elements in the environment. The work presented in this paper focuses on the first two steps of the model; it does not address projection of future states and prediction of new situations.

A central element in the situation awareness model is monitoring the operational SC environment and determining the state of SC objects. A SC object represents a logical or physical item, e.g., a customer's order or a container, an activity within a logistic process, as well as a whole logistic process (Winkelmann et al. 2009). The dynamic environment of the SC comprises partners collaborating with each other. Hence, in order to achieve SA, there is a need to consider synergistically all SC objects. An SCEM system for SA should be capable of not only revealing the current state of an SC object but also make references to past situations, as well as anticipate future situations. This capability can be achieved by an SCEM system that can detect current situations and store them in a knowledge base, which in turn can be used for finding similarities with past situations and for predicting future ones.

Before describing how situation detection can be achieved, we first describe SC objects, events and situations as well as their relations. Consider a typical SC consisting of SC objects such as suppliers, 3PLs, trucks, production machinery and the relations among them. Each SC object has properties related to its role in the SC. For example, a product is described with properties, such as its dimension, price and weight, while a 3PL company is described with properties such as location and transportation capabilities. Relations between SC objects have also properties specifying additional information, e.g., when a Cooperates relation type is time-limited or when a partner is more preferable than another a weight value can be added. Table 1 presents indicative SC objects and their relations.

Table 1. SC Objects and their Relations in a SC

SC Object	Notation	Properties
Supplier	S	{id, Name, Location, lat, long, NumberOfEmployes, ProductionCapabilty}
3PL	T	{id, Name, Location, lat, long, NumberOfTransports}
Product	Pr	{id, Name, Price, Weight, Length, Height , Width}
Truck	Tr	{id, Capacity, lat, long}
Order	O	{id, Quantity, Address, Price , CustomerName, lat, long}
Relation	**Notation**	**Description**
Cooperates	$\rightarrow^{Cop}$	Relation Between Supply Chain Partners
Undertakes	$\rightarrow^{Und}$	Relation Between Order and Partners
Produces	$\rightarrow^{Prod}$	Relation Between Supplier and Products
Operates	$\rightarrow^{Oper}$	Relation Between 3PL Company and Truck

We proceed with an analysis of events that are linked to SC objects and define or change their state. SC processes and systems are the main sources of events. Suppliers, 3PLs, manufactures and wholesalers produce events through their operations in the supply chain. Moreover, SC processes such as procurement and transportation generate events that are typically structured. Further event sources include sources that are not directly related to the supply chain processes, such as events from the physical, economic and political environment of the SC. Physical disasters is an example of an external event source which influences the SC (Vlahakis & Apostolou 2015). In practice, prominent event sources include sensors, RFID tags and GIS systems, which provide the necessary status data, as well as ERP systems, which provide information about customers, orders, production processes and stock levels. With regard to external events sources, information is typically provided through web services over the Internet.

We distinguish between two types of events (Table 2): simple events and complex events. Simple events are those that appear uniquely and are described by properties, such as id for unique event recognition, the time of event appearance, the duration of the event, the description of the event, a degree of event influence in the system, the producer of the event and location properties, like LAT and LONG. Complex events are those that are synthesized from simple events.

To associate events with other events and with SC objects, we use the following types of relations: an Affects relation refers to a relation between an SC object and a simple event; an Associated relation refers to a relation between two or more simple events; a Correlated relation refers to a relation between simple events that synthesize a complex event. A Trigger relation refers to a relation between events and situations and will be explained at the end of this section.

Table 2. Events and their Relations in a SC

Event	Notation	Properties
Simple Event	e	{id, time, date, duration, eventDescription, degreeOfInfluence, Producer, lat,long}
Complex Event	Ce	{id,time,date,duration,CeDescription,degreeOfInfluence}
Relations Between Event and SC Objects	Notation	Description
Affects	$\rightarrow^{Af}$	The relation between an event and a SC object
Associated	$\rightarrow^{As}$	The relation between simple events
Correlated	$\rightarrow^{Cor}$	The relation between simple and complex events
Triggers	$\rightarrow^{Tr}$	The relation between a (complex) event and a situation

SC events can be classified based on several factors, depending on their nature (asset event, delivery event, etc.), on the expectancy of their appearance (expected or unexpected) or on their degree of influence on the supply chain. Alternative categories of supply chain events (Liu, et al. 2007) include task status related events (e.g., beginning or end of production) and events produced by a task (e.g., out of stock). External events need also to be classified in order to be managed sufficiently and to facilitate the monitoring, correlation and identification of relevant patterns. Table 3 presents a classification of events in the SC process depending on their possibility of appearance, their degree of influence on the SC and their initial source. Especially for complex events, we can distinguish between a B2B event, a type of event that affects two supply chain partners, an order to business O2B event, an environment to business

E2B event and other similar types of complex events denoting the main correlating business entities.

Table 3. Event Classification Scheme

Classification Factor	Class	Description
Possibility of Appearance	Expected	It may happen immediately or shortly. Expected to happen frequently in specific process or participants.
	Prospective	It may happen in the long term. Expected to happen frequently to specific processes or participants.
	Likely	Expected to happen sometime in the future.
	Unlikely	Unlikely to happen.
Degree of Influence	Disaster	It may cause a disaster to the supply chain. Loss of crucial facilities that serve the supply chain. e.g., fire in a supplier factory.
	Critical	It may affect severely the supply chain. Severe and longtime effect on the supply chain, e.g. economic blockade of a specific market.
	Limited	It may have a limited effect on the supply chain process, for a limited period of time, e.g., employee strike.
	Minor	The effect of this event has little effect on the performance of the supply chain process and mitigating actions are easy to apply to avoid them. E.g., delayed status events.
Internal Events	Task Related	E.g., Production finished or Delivery cancelled.
	Event Produced by a Task	E.g., Machine Failure on Production.
External Events	Social Political	E.g., strikes that announced and influence the production or the delivery date.
	Economical	E.g., the price of raw materials.
	Traffic	E.g., Road blocks and traffic conjunctions.
	Weather	Extreme weather conditions that may affect the production or the delivery of products.

Situations are initiated or terminated by events which have an effect on important real-world entities and represent an important state of entities (Etzion 2009). We define supply chain situations as event occurrences that indicate a certain state in SC objects and may require a reaction in terms of decision making or corrective actions. In our approach a situation is represented using a predefined set of correlations between simple events, complex event and SC objects and events. A situation can be triggered (see last row of Table 2 and Figure 2), i.e., initiated or terminated, by a simple or complex event, which is linked to one or more SC objects and has an effect on the objects' state.

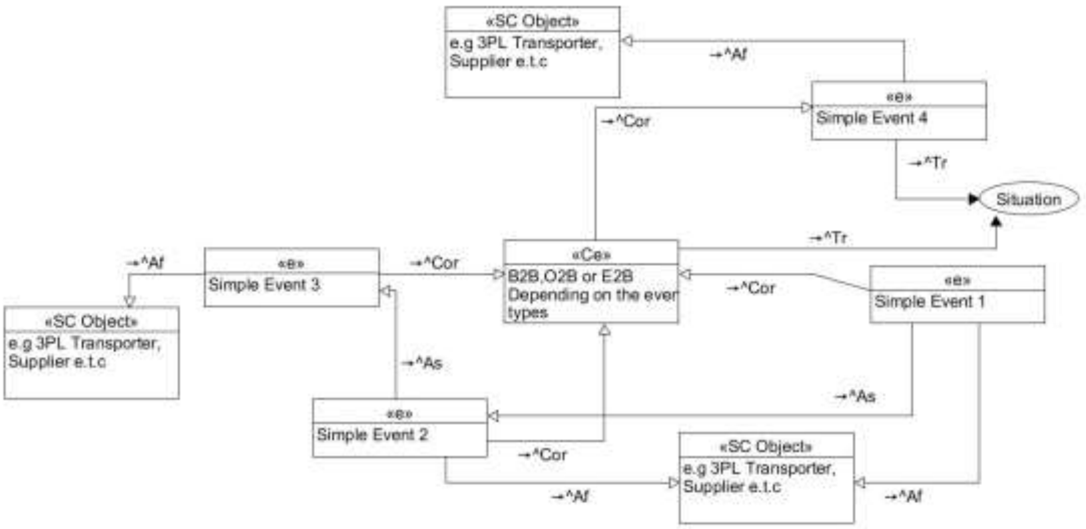


Fig. 2. A Situation can be Triggered by a (complex) Event Linked to SC Object(s)

An example situation representation is shown in figure 3. In this diagram, we present a situation where «S» receives an order «O» from a customer. The supplier connects with the 3PL company «Tr» with a type relation «Cooperates» which implies their business relation. The 3PL company «Operates» two trucks «Tr_01» and «Tr_02», which undertake the delivery of products. During this process unexpected events «Affects» both 3PL company and supplier. These events are «Associated» and «Correlated» to create a complex event «Ce». An example of a complex event is a detected transportation delay in a particular city in which trucks are moving slowly, due to road constructions, and a fall in the demand resulting from decreasing quality of service in this particular area. This example shows that complex events can be created from two events that happened in a short time window or by two or more events that happened in two different time periods but still influence the SC. The correlation of two or more simple events creates complex events whose influence is not restricted to a single SC participant but extends to a greater part or even to the whole SC, effectively indicating the initiation or a termination of a situation.

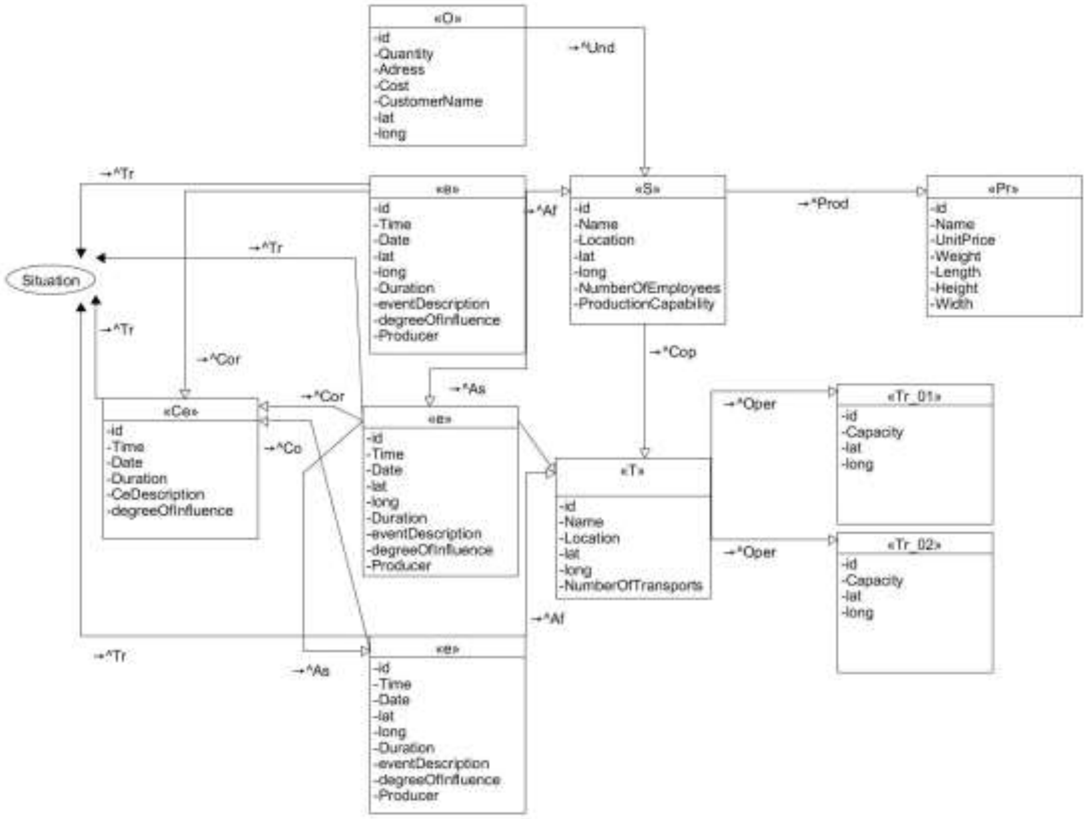


Fig. 3. An Example Situation

4 Realising Situation Detection Through Event Correlations

The situation representation approach described in the previous section provides a well-defined model of situation triggers which are based on correlations of simple and complex events with SC objects and which, in turn, describe the state of an SC in time. Further, the state of an SC in space is described using the location of the events occurrence and the place that SC objects are located in with LAT and LONG data. Situation detection is realised through a two-stage event correlation process which supports correlating in real time simple events into complex events as well as associating (complex) events with SC objects. Event correlations are based on event operators and event rules that are used in two stages (fig. 4).

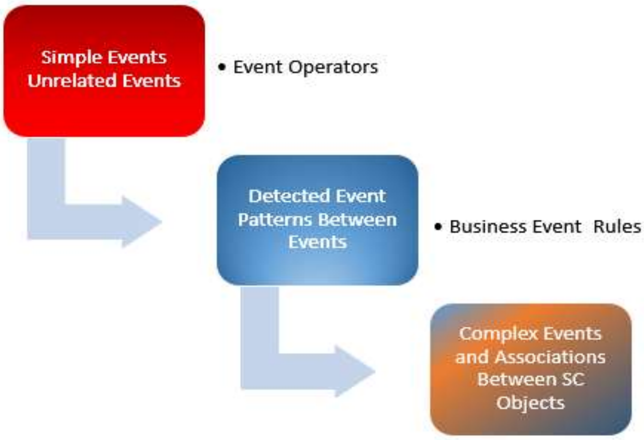


Fig. 4. Event Correlation Stages

In the first stage, the basic operators are used for the detection of the occurrence of simple events and for their combination. Event correlation is a technique for making sense of a large number of events and pinpointing the few events that are really important in that mass of information. This is accomplished by looking for and analysing relationships between events in time, so-called complex event patterns, and aggregating simple events into complex events. Table 4 presents the event operators used for simple event correlation, as proposed in (Mei & Madden, 2009; Wu, et al. 2006; Akdere et al. 2008).

Table 4. Event Operators

Event Operator	Symbol	Description	Supply Chain Event Example
Sequence	Seq^t	An event E_1 is followed by the event E_2 in a particular time window t.	A machine failure followed by the production of goods outside specifications.
Negation	$!^t$	The negation represents the absence of event E. It can be combined with time window or not.	The order completion in a period of time t.
Conjunction	$+^t$	Event E_1 and event E_2 both can occur in a specific time window but their order does not matter.	A machine failure event followed by a lack of quantity of raw materials.
Disjunction	$\times^t$	Event E_1 or event E_2 or both can be occur in a specific time window t	A machine failure event followed by a lack of quantity of raw materials
Repetition	$E^{v,t}$	An event Ecan be occur v times in a specific time window t which can be limited or not.	Repeated strikes by employees.

The example in figure 5 shows how event correlation helps making sense of a large number of events and pinpointing the few event patterns that are of importance, in this case (A Se B), i.e., a sequence pattern between events A and B. Event correlation takes place in specific time windows, in this case the window is five time units. At t -3, we have two matches of the aforementioned sequence pattern which result into the identification of two complex events: (A2 Se B2) and (A3 Se B3). At t-2, we have one match (A2 Se B2) and zero new matches. At t-1, we have again one match and zero new matches. At t, we have two matches, one of which is the new complex event (A1 Se B1).

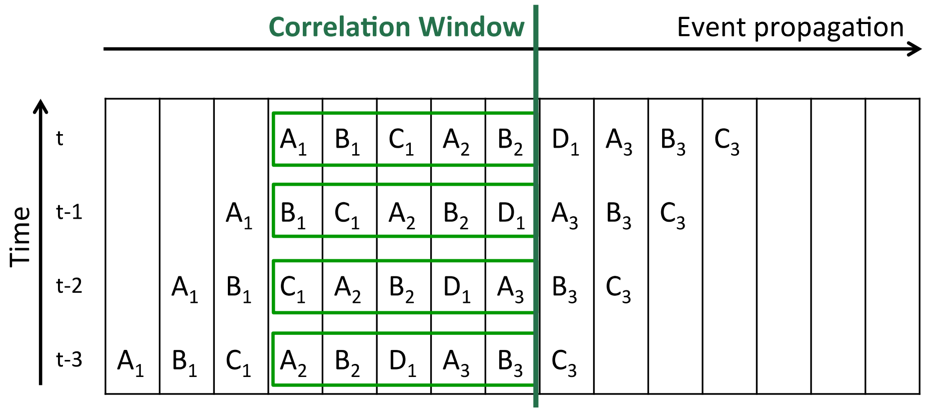


Fig. 5. Illustration of Event Correlation using Time Windows

In the second stage, event rules are used to make correlations at a business logic level. Such correlations enable the creation of associations between simple and complex events and SC objects leading to the creation of new complex event patterns that can trigger the initiation or termination of a situation. We propose a formal approach for correlating events with SC objects so that the state of SC objects in time is defined. Further, space is described using the location of the events occurrence and the place that SC objects are located in with LAT and LONG data.

Event rules are executed as event queries and, in order to perform the correlation, make use of the relations between events and the SC object, as well as the attributes of the events and the SC object. The general structure of this type of queries is presented in fig. 6:

MATCH E1, E2
WHERE relations between events and supply chain object e.g. E1→O←E2
AND Constrains in event properties criteria, SC Object and event operators.
CREATE EventPattern2Object complex event. (E2Order, E2Supplier etc.)

Fig. 6. General Structure of Queries Correlating Events E1 and E2 with Object O

The case where a correlation leads to the association of more than one SC objects, is handled by the type of queries shown in fig. 7:

MATCH E1, E2
WHERE E1→O1-[Relation Between the two Objects]-O2←E2
AND Constrains in relation between SC Objects, event properties criteria, and event operators.
CREATE B2B, E2B, O2B complex events

Fig. 7. General Structure of Queries Correlating Events E1 and E2 with SC objects O1 & O2

In the case of event rules representing business logic, the time window used is typically larger than the one used to correlate simple events into complex ones, enabling events from different time periods to be combined and associated with SC objects.

4. SCEM Framework for Situation Awareness

We designed our SCEM framework (fig. 8) with the aim to enable supply chain situation awareness by facilitating situation detection using event correlations. Specifically, we aimed at supporting SC participants with situation detection capabilities at all stages of SC processes, e.g., order submission, warehouse lookup, transportation, etc. Our framework has been designed to meet the following five core functionalities (Tröger & Alt 2010; Knickle and Kemmeter, 2002; Winkelmann, et al. 2009):

- **Monitor**. A basic role of SCEM is the monitoring of SC objects in respect of occurring events. This functionality is responsible for tracking data and detecting any abnormality. It is typically implemented as a network of sensors, RFID tags, web-services and generally any source of information that can produce events in the supply chain.
- **Notify.** This refers to the capability of notifying the right person, at the right time about undesired or abnormal events. Relevant stakeholders (e.g., procurement manager) should be made aware about an abnormality in real time.
- **Simulate.** This refers to capabilities assisting SC managers in assessing the impact of a SC change.
- **Control.** This refers to capabilities assisting the SC managers in accomplishing a change.
- **Measure.** The last function bridges the gap between SCEM and SCM by analysing data and enabling companies to identify certain patterns related to the causes of supply chain event bottlenecks in order to optimize their supply chain processes and structures. After the analysis, the results are fed back into the system to improve the efficiency of business rules.

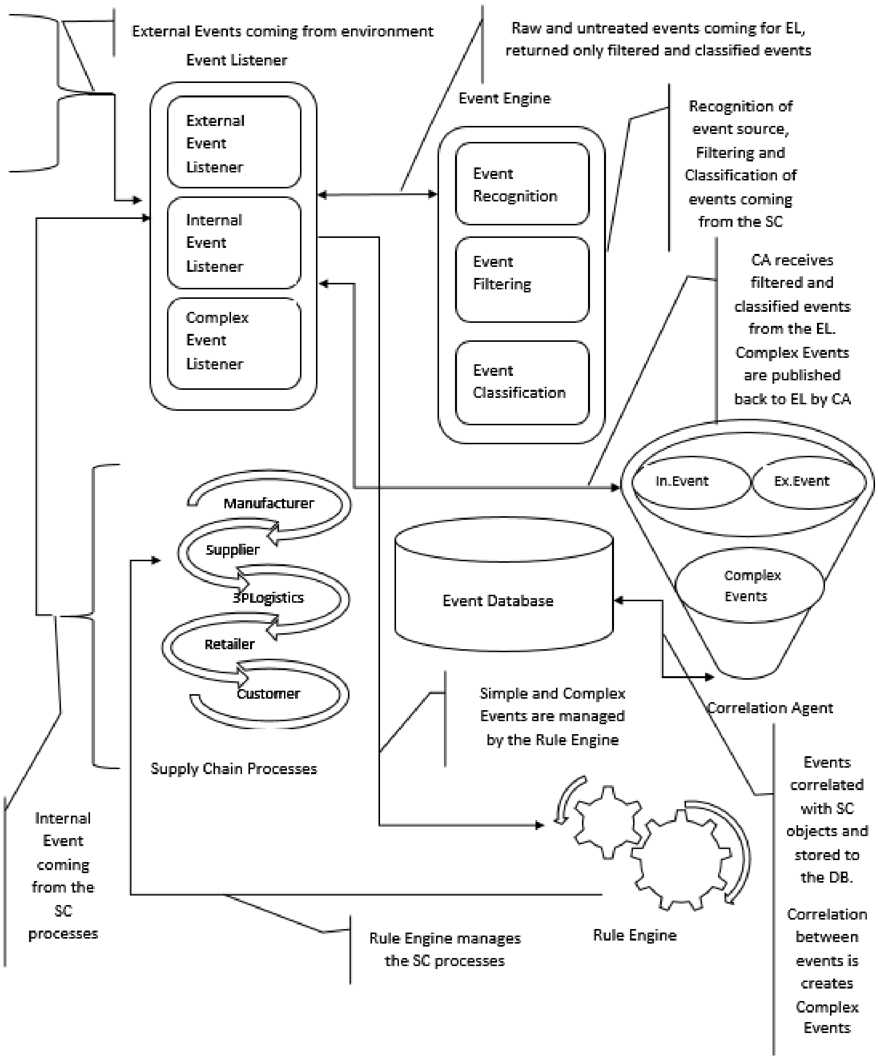


Fig. 8. Conceptual Architecture of SCEM Framework for Situation Awareness

Table 5 presents the mappings between the framework components and their functionalities, which are explained next.

Table 5. Mapping between SCEM Basic Functionalities and Framework Components

SCEM Basic Functionalities	Framework Components
Monitor	Event Listener-CEP engine
Notify	
Simulate	Correlation Agent-Event Database
Measure	
Control	Rule Engine

Event listener: Event Listener (EL) is an agent responsible for the monitoring and the notification of events. It consists of three sub-components, the internal event listener, the external event listener and the complex event listener. The internal event listener is responsible for listening to events coming directly from SC systems and processes, the status of SC objects and their changes e.g., ‘an order is submitted’, or ‘the product is shipped’. The external event listener handles events coming from outside of the supply chain. To achieve this, a series of external information sources are monitored to detect events that are likely to be useful for decision making. These events may occur from various web-services which provide economic, weather, traffic or other pertinent information. External and internal events are published to a CEP engine through an enterprise channel bus for analysis and evaluation of their importance and assessment of their impact on the SC, through filtering, classification and association with the SC objects. After the processing of the events, these are returned to the EL and are sent to the Correlation Agent for further processing. The complex event listener focuses on the detection of complex events that have been published to the enterprise event channel by the Correlation Agent component.

CEP Engine: The CEP engine operates to support the EL by means of event filtering, event recognition and event classification. Event filtering is discarding unwanted or not important events according to predefined rules. These rules concern basic characteristics of the events, such as the time of their appearance, the source etc. Event filtering is important for two reasons: Firstly, because there is a need to reduce the storage and the traffic costs of the event database by archiving unnecessary, mostly expected events, such as status events coming from supply chain objects. Secondly, because there is a need to reduce the events that are subject to be correlated by the corresponding component (the Correlation Agent). Event recognition aims to identify the source that generated the events that will be used afterwards by the Correlation Agent to perform associations between the event and the SC Object. Events are classified to determine their importance in the supply chain process as described in section 3. After their analysis, the events are propagated to the rest of the framework through the Event Listener, in order to trigger action recommendation by the Rule Engine or to be processed by the Correlation Agent.

Correlation Agent: The Correlation Agent (CA) detects situations based on the relations between the detected events and SC objects representing situations. To perform this action CA operates in parallel phases (fig 9). In synergy with EL, CA anticipates classified events processed by the CEP engine. When a new classified event is published by EL, CA undertakes its management: the event is stored as a node in the Event Database and is correlated to the rest of the SC objects that either created the event or are affected by its existence. Additionally, using business logic and previous knowledge depicted in queries, CA searches the database for relations between simple events and SC objects to formulate complex events. When a complex event is detected, it is published to the EL for further analysis by the CEP engine. Complex event analysis results are returned to CA that stores them to the Event Database and a new series of queries is executed from the beginning.

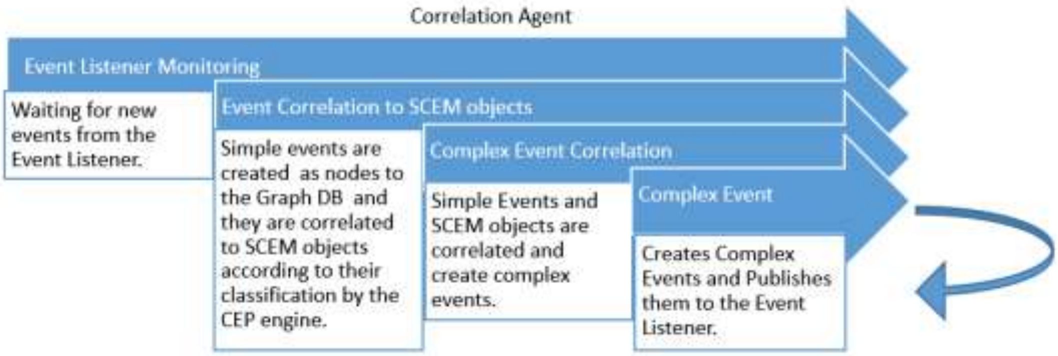


Fig. 9. The Correlation Agent

Event Database: The event database accomplishes two main tasks. Firstly, it stores schematically the representation of the supply chain consisting of SC objects, namely the suppliers and the 3PLs, as well as their collaborative relations. Each SC object is represented as a node and one or more edges indicate its relation with other nodes. Whenever a new participant is added to the supply chain, a new node is added and the relevant connections are created. Secondly, in addition to the SC environment, the Event Database stores simple and complex events processed by the CEP engine and captures the relations between the SC Objects and (complex) events. New event nodes are stored in the Event DB by CA, which is responsible for creating event nodes in the SC schema.

Because of the high computational cost and the flexible storage capabilities associated with the sophisticated queries needed for implementing this functionally, traditional relational DBs are deemed inappropriate. In contrast, the use of NoSQL databases and especially graph databases, which are playing an important role in the area of social engineering, are considered to be a solution to the problem of correlating events and revealing cause and effect relationships in the SC environment. Based on graph theory, graph databases use graph structures with nodes, edges and properties for representing and storing data. In comparison with relational databases, where links between the data are based on the data itself, a graph database uses edges to relate the stored nodes. The advantage of this type of storing is that they allow simple and rapid retrieval of complex hierarchical structures, whereas a relational database for the same data retrieval would use complex queries with higher computational costs, making the use of graph database effective in performance issues. Another advantage of graph databases is becoming evident when performing queries that are more than one level deep. For example, when searching which 3PL company operates a single truck and who was the driver on a particular date, a relational database would require three or more datatables such as Trucks, Drivers and Date datatables depending on the model and the number of existing joins. In contrast, in a node database, such a query has only to ‘jump’ between already related nodes that store the related data. These types of queries require less computational power and are much simpler than those of relational databases.

SC objects of the SC environment consist of SC participants, such as suppliers and 3PL companies, and their operational means. Business relations refer to the relations between the SC objects. By considering the SC as a network of objects and relations, we can formulate its representation as a graph database, where participants constitute the nodes and their relations are the edges of the schema. A graph database, when used for representing and modelling the environment of the SC, has advantages, such as deep level queries for retrieval of information regarding the operational state of each SC object and the relation between each other. With traditional relational databases, this type of queries demands significant programming effort and performance cost. Therefore, the advantage of storing data as nodes and edges, in a graph database, proves to be suitable for the representation of a complex relational SC environment. Queries in the node database enable the combination of information pertaining SC objects, the provision of information about their state, the detection of event patterns at any stage of the supply chain and the combination of events that affect more than one supply chain objects. Once a situation is detected, it is published to the Rule Engine for recommending corrective actions.

Rule Engine: It controls the SCEM processes and recommends corrective actions to be taken by SC managers once a situation is detected. Actions concern possible re-configurations of the SC process by selecting alternative paths, e.g., selecting another supplier or 3PL company, or involving more SC partners to the process, modifying the transportation route, or making modifications in the production process. This is made possible via business rules that are based on supply chain Key Performance Indicators (KPIs) and business logic. When a rule is fired due to a detected situation, mitigating actions are recommended. Corrective actions implemented in the past are stored together with their triggering situations as well as feedback indicating how successful the action was and a description about its positive or negative effect on the SC. This information is used as a reference and is presented to the user when and if the same situation is detected again.

5. Framework Implementation

5.1 Technical Considerations

Technically, the framework follows an event-driven, service-oriented architecture that connects several loosely connected components via a common service bus. This architecture is suitable for our design , since it allows us to achieve reusability, loose coupling of key components and abstraction, features that are important for a framework aiming to serve a changing business environment,. By being service-oriented, the proposed framework exhibits the following characteristics (Erl 2008):

- *Vendor neutral*: In order not to be depended on one vendor only, we use software standards and open source software like BPEL 2.0, as well as programming languages like Java and C#. By not depending on a particular software vendor, framework components with ageing technologies can be replaced in the future without the need to replace other components or redesign the framework entirely.
- *Business driven*: The framework is designed for the needs of the supply chain and thus takes into consideration the environment of the involved partners and their requirements.
- *Enterprise Centric*: The framework takes into account enterprise requirements for SC processes and supports services which are reusable across SC partners, e.g., 3PLs share common service to interact with the rest of the SC process.
- *Composition Centric*: Based on SOA, our framework can deal with changing requirement by making use of existing components and by connecting third-party components by means of Web Service standards.

5.2 Implementation

To implement and simulate SC processes in our framework, we use BPEL 2.0 (http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html), which is a standard for implementing processes-aware systems using Web Services. We model all core processes of the supply chain, such as order submission, order processing and request for transportation. These processes are designed in Oracle Fusion Middleware platform, Jdeveloper 12c Studio and are deployed as web services in a WebLogic 12c server. Event delivery is achieved with Oracle Event Delivery Network, which is a java based message service and works as an intermediary for event publishers and event consumers. This technology allows us to publish and consume events to and from the entire framework efficiently. Events are defined with the use of Event Description Language (EDL) that is a schema used to build business event definitions in Service Oriented Architecture (SOA) composite applications. The technical architecture of the framework is depicted in figure 10, showing all technologies used.

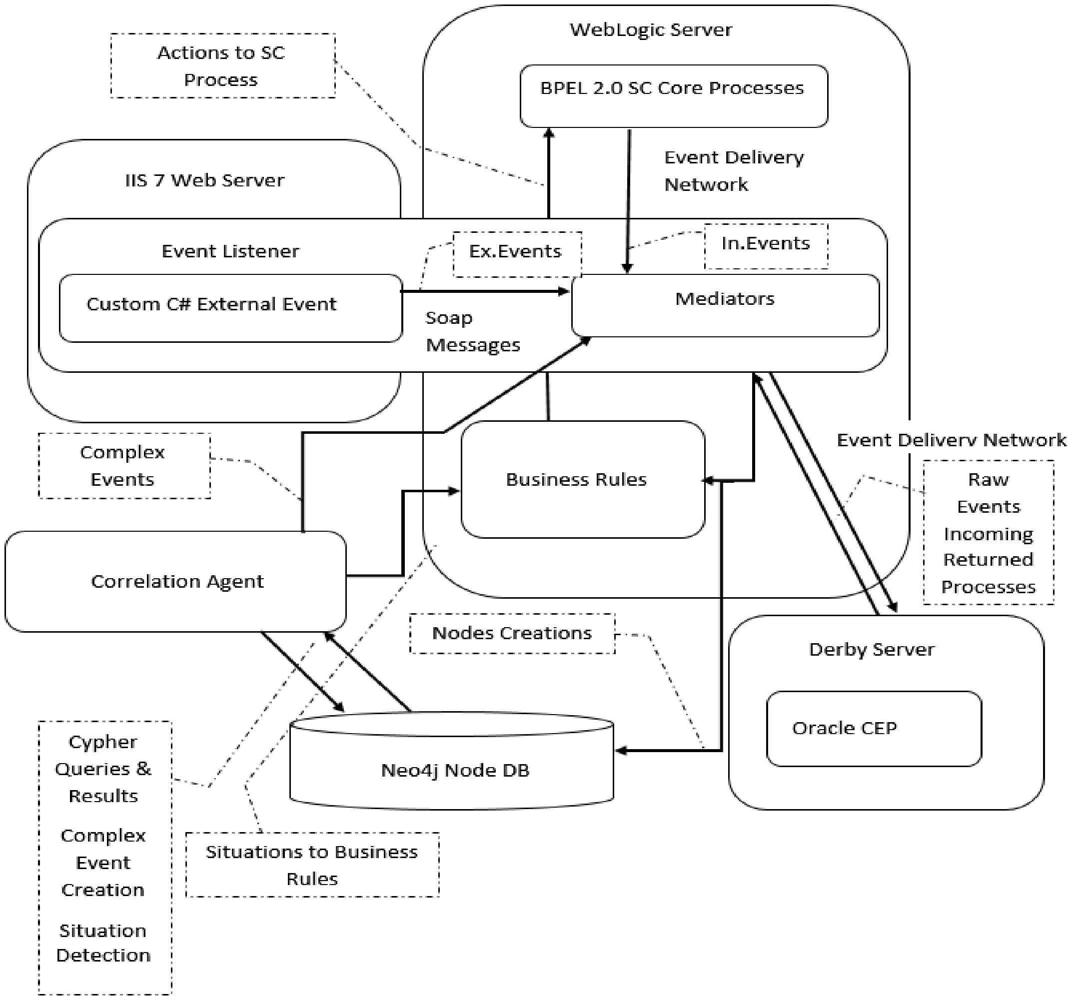


Fig. 10. Framework Technical Architecture

We implemented the EL using .Net framework and C# language as well as Oracle Mediators, a useful feature of Oracle Fusion Middleware. The external EL sub-component is implemented as a web service with the use of custom programming. The operation of this sub-component is summarized to the detection of events from external sources, such as sensors, GPS devices, as well as weather and traffic web services (e.g., Bing Traffic: http://msdn.microsoft.com/en-us/library/hh441726.aspx and National Oceanic and Atmospheric Administration's http://graphical.weather.gov/xml/). Oracle Mediators implement the internal and complex EL subcomponents supporting the detection of internal and complex events discovered in the SC processes through the Event Delivery Network. All the events delivered to the EL are in raw state, unfiltered, unassociated, unclassified and delivered by the EL to the CEP engine, which performs filtering, classification and association with the proper source for further processing by the Rule Engine.

CEP engine has been implemented with Oracle Event Processing. Processes of filtering, classification and association with the proper SC object, are achieved via Continuous Query Language (CQL) queries running in a Derby Server. The server accepts incoming events from the EL component, processes them and publishes them back to the EL. Even though this tool is very useful for a CEP application and provides the necessary flexibility related to the time window of the event recognition and correlation, it doesn't address the problem of correlation between two events in different time windows.

Business Rule engine is another Oracle Middleware SOA platform feature that the framework uses to enable business logic and control in our framework. The type of rules, available to this feature, are either "if then else" clauses or logic tables This component is published in the Web Logic server as a service, allowing a loosely coupled interaction with the rest of the processes that imitate the sub processes of the SC. The main purpose of the Business Rule engine is the management of the events coming for the EL component and the control of the SC process. When an event arrives in this component, rules are triggered using business logic according to predefined policies or thresholds. Actions control the SC process either by re-directing it, e.g., by selecting another supplier when the original one cannot respond to specific requirements, or by applying business policies, e.g. discounts in products' prices in case of delays.

Correlation Agent is a continuous running process based on .Net framework and implemented in C# programming language. It consists of classes responsible for the correlation of simple events as the detection of situation, querying the node database in short periods of 2 to 5 mins, depending on the type of the query and the desired correlation or detection that we wish to achieve. Queries are written in Cypher language (figs. 11 and 12), synthesizing events and SC objects nodes properties, as well as their relation to perform the correlation and situation detection. Because of the fact that events are stored in the node database the time window property can be a parameter in the Cypher language. Due to the continuous periodic running of the CA, the disadvantage of the limited time window in the stream processing is overcome. Situations detection is performed querying simple and complex events stored in the db. Once a situation is detected, it is published to the business rule engine and actions are taken to the SC.

```
MATCH Event e, Supply Chain Object (T, S, Pr, O) sc
WHERE (e) - [:af] -> (sc) <-[:af] - (e)  AND
```

```
              (e) – [:af] -> (sc)  AND
              (e) - [:as]-> (e)     AND
              (e) - [:as]-> (e)   AND
       IF query result is true THEN
{CREATE Complex Event ce   (Complex event stored in DB as node)
WHERE (e)-[:Cor]->(ce)<-[:Cor]-(e)  AND
      (e)-[:Cor]->(Ce)}
     ELSE
      {TIMER SET TO 2 min}
LOOP
```

Fig. 11. Example of Complex Event Detection in Cypher

```
MATCH Event e, Complex Event ce, Supply Chain Object (T, S, Pr, O) sc
WHERE (Ce)-[:Cor]->(e)-[:Af]->(sc) AND {Criteria and Constraints
Related to events and SC properties}
       IF query result is true THEN
{CREATE Situation S    (S stored in DB as node)
WHERE (Ce)-[:For]->(S)<-[:For]-(e)}
    ELSE
       {TIMER SET TO 5 min}
LOOP
```

Fig. 12. Example of Situation Detection in Cypher

We use Neo4j node db to store events as well as to create the SC depiction in a relational network. In a node db, there are two main types of nodes, the SC objects related with each other imitating their business relation in real life and the event nodes. Subtypes of the SC objects are the supplier nodes, 3PL nodes, orders nodes, supplies nodes and means of transportation nodes. Subtypes of the event node type are the internal event node types, external event node types and complex event types. Events and SC objects are linked by the CA with edges that represent the relation that are created after the three stage process of the CA.

6. Evaluation

6.1 Evaluation Scenario

For the illustration and evaluation of our framework, we designed a typical logistics scenario. The scenario consists of four suppliers and three 3PLs located in different areas, as well as a product retailer, which submits orders to the system. Each 3PL company and supplier has different transportation and replenishment capabilities which provide the constraints to the process.

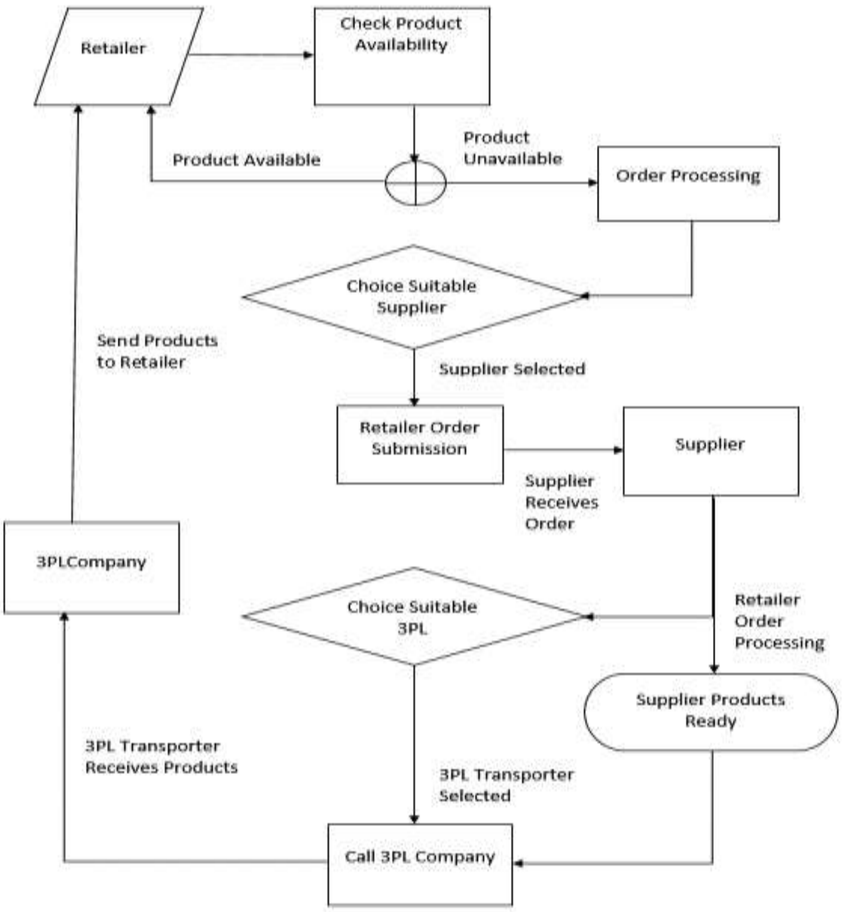


Fig. 13. SC Process Diagram

The simulation creates different scenarios based on products availability and guides the process accordingly. As demonstrated in figure 13, each scenario starts with the calculation of products' availability at the retailer's side. If the quantity of products is not sufficient, an order is processed and the application searches for a suitable Supplier to fulfil the order requirements. The retailer submits the order to the selected supplier. The supplier starts processing the order and searches for a suitable 3PL company to handle the delivery.

When the products are ready, the supplier calls the 3PL company and the products are delivered to the retailer.

6.2 Evaluation Goals & Methodology

Our primary evaluation objective has been to assess quantitatively the situation awareness capabilities of the proposed framework and prove the potential gains for SCEM in terms of:

(i) Situation detection at different stages of the supply chain. By detecting situations from events that happened in two or more different supply chain objects and correlating them, we enable SA at all stages and, hence, enhanced capability for decision making.

(ii) Event correlation regardless of the event time window. Events happening in different time windows that vary from minutes to days but still affecting the process of the supply chain can be correlated in order to identify a situation.

(iii) Capability of the framework to detect situations before their appearance.

(iv) Scalability of the framework with respect to the number of generated events.

To identify events that could affect the logistics process, described in the previous section, we talked to professionals who gave us examples of simple events that could disrupt that process. To simulate the events in the SC we had to specify a probability of appearance as well as a level of criticality. We generate pseudo-randomly typical internal and external events occurring at different supply chain stages. To simulate the external events, we use an external event generator that also populates pseudo-randomly external events pertinent to the SC. The types of the generated events and the possibility of their appearance are presented in table 6.

Table 6. Internal and External Events in the SC and their possibility of appearance

Internal Events		
Event Type	Event Description	Probability of Appearance when an event is published
Supplier Event	Employees on strike, Delivery date will be delayed.	20%
	Problem on product specification.	20%
	Insufficient Quantity.	30%
	Production Machine Failure.	30%

3PL Company Event	Employees on strike, Delivery date will be delayed.	30%
	Truck failure. Delays.	30%
	Problem of delivery. Number of products damaged during transportation.	20%
	Quantity too large. Products will be partly transported.	20%
External Events		
Event Type	Event Description	Probability of Appearance when an event is published
Weather Events	No important delays due to weather.	33%
	Bad Weather conditions. Delays may occur.	33%
	Extreme Weather Conditions. Process possibly affected.	33%
Economic Events	Loan Application Rejected.	30%
	The merger agreement has been cancelled.	20%
	Liquidity Problems, payments delay.	40%
	The partner temporarily suspend its operation due to economic problems.	10%
Traffic Events	Conjunction delays.	40%
	Delays due to construction works.	30%
	Road Closed, Detour must be taken.	15%
	Truck had a road accident.	15%

Randomly generated events of each type have duration, time and date attributes, simulating the events of a real SC. Event filtering is preformed based on their degree of influence in the SC, as shown in Table 7. Depending on the attributes of the each event (e.g. duration of the event), the CEP engine calculates in real time the degree of influence considering various business level thresholds. For evaluation purposes, we have limited the duration of the events in a time window of 30 days for the internal events and 10 days for the external events.

Table 7. The impact of the events according their duration

	Duration in Days	Impact in the SC
Internal Events	0≤e_{dur} ≤2	Minor
	3≤e_{dur} <10	Limited
	10≤e_{dur} <15	Serious
	15≤e_{dur} <30	Critical
External Events	0≤e_{dur} ≤2	Minor
	3≤e_{dur} <5	Limited
	5≤e_{dur} <8	Serious
	8≤e_{dur} ≤10	Critical

Based on our discussion with the industry professionals, we designed five use case scenarios described below that combine different internal and external events. The scenarios aim to illustrate capability to detect situations created from events that appear in different SC objects and a variety of time windows. They also illustrate the capability of the framework to detect the tested situations before being terminated.

A series of correlated events, in each scenario, represent an initial state $\mathrm{S_{init}}$ of a situation or trigger a final state $\mathrm{S_{fin}}$. The $\mathrm{S_{fin}}$ is triggered, when the rule engine does not have a suitable rule to fire in order to manage the situation or when the rule fired is not capable of preventing the situation, e.g. the selection of another 3PL company or another supplier is not possible because of the business restrictions such as diversified production line or insufficient transport capability. The situations that reach their $\mathrm{S_{fin}}$ state are considered as lost, i.e., not detected on time.

The events involved in the situation detection process are evaluated by the CEP engine which considers their duration and description indicating the impact of the event in the SC process as demonstrated in table 8. Only events with critical or serious impact are used for situation detection.

Cases that may arise from the evaluation of events are presented below:

- A simple event is evaluated as having a minor or limited impact. Business rules are involved to improve or limit the impact of the event.
- A simple event is evaluated as serious or critical, but without correlation with other events. Business rules are applied to deter the event impact on the SC process.
- A set of events are evaluated as critical or disaster and an initial state appears (S_{init}). Business rules are applied or the process is reoriented to deter the S_{fin} state of appearing.

- The prevention of the situation is not possible from the framework and a S_{fin} is triggered. Business rules are applied to limit the situation impact on the SC process.

The scenarios were tested in a series of experiments. The scalability as a feature of the framework was tested using different sets of test runs with increasing number of internal and external events. The experimental test runs were performed in a sequence of five groups with an increasing step of 100 tests. The total number of tests executed was 1500. All tests were performed in a Core i5 2.6 MHz processor with 6 GHz RAM. The scenarios that were executed are described below:

Scenario 1: The employees of one of the major 3PL companies announced repeated strikes. Possibly due to improper maintenance, which resulted because of the strike, a transportation truck broke down, but the company was still capable to deliver the order, using another truck. However, as a result of all these, the retailer announced that many products were delivered in bad condition. The events that describe the initial state S_{init} are in sequential order: e_1 which describes the repeated strikes from the personnel of the 3PL company, e_2 event describes the truck failure as the e_3 describes the announcement of the retailer about the condition of the products delivered, which also represents the final state S_{fin} of the situation. The situation is described in expression (1). For the situation detection only the events with degree of influence serious and above are considered. The result of the situation of this scenario is the creation of a complex $B2BEvent$ that is managed by the rule engine with a proper rule action according to a predefined rule that takes into consideration the importance of the complex event as calculated by the CEP Engine. An example of the event rule that fired when this type of $B2BEvent$ is published is presented in figure 14.

$$Seq^{t2}(e_1^{vt1}, e_{2,}\, e_3) \qquad (1)$$

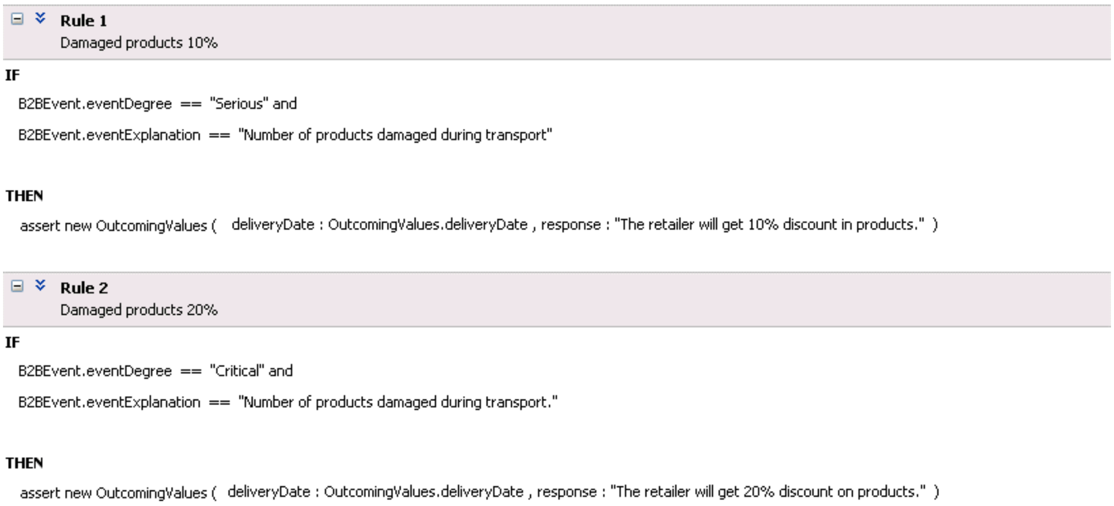


Fig.14. The Business Rule for Scenario 1

Scenario 2: A machine used in a production process of a supplier breaks down; this creates a shortage of the associated product part to the SC partners that use this part in their production processes. These two events are generated by different partners and are not related to each other when populated to the event network. The third event is that the order cannot be completed because the final product assembly is hampered by the lack of the specific part and hence the order is addressed to an external partner with an additional cost to the SC in time and resources. The event operators used to describe the situation pattern are presented in expression (2). The result of this situation is a complex $O2BEvent$ that affects both the supplier and the 3PL company. The initial state S_{init} for this scenario is described by the event e_1 which refers to the announcement of the first supplier that its machine failed and the event e_2 that describes the shortage of the supply. The final state S_{fin} is the announcement of the order cancelation. The event rule that manages this type of $O2bBEvent$ is presented in fig.15.

$$Seq^{t2}[(e_1 + e_2)!\, xe_3^{t1}, e_4] \qquad (2)$$

```
Rule:1 The order quantity is more than 25000 and less than 50000
IF
O2BEvent.eventDegree=="Serious" and
O2BEvent.eventDescription=="The order may cancelled." and
OrderInputValues.orderQuantity < 50000 and
OrderInputValues.orderQuantity >= 25000
THEN
Assert new OutcomingValues( deliveryDate: XMLDate add days to (OrderInputValues.orderDeliveryDate,
10), response:"The order will be registered to third party partner.")

Rule 2:The order quantity is more than 50000
IF
O2BEvent.eventDegree=="Critical" and
O2BEvent.eventDescription=="The order may cancelled." and
OrderInputValues.orderQuantity > 50000
THEN
Assert new OutcomingValues( deliveryDate: XMLDate add days to (OrderInputValues.orderDeliveryDate,
0), response:"The order will be cancelled.")
```

Fig.15. The Business Rule for Scenario 2

Scenario 3: The 3PL company announces that one of the trucks is out of service, but also states that the company is capable to deliver the products and complete the order with a short delay inside the delivery limits. During transportation, a major road is closed and a detour has to be taken. A number of products are inspected by the final recipient to reveal that they were damaged during transportation. The correlation of these events produce a complex $en2bevent$, environment to business event, that associates events coming both from the SC and the operational environment. The pattern used for the correlation in this case is a sequence of events as follows in expression (3). The initial state S_{init} for this situation includes the e_1 event and xe_2, describing the truck failure and the road closed event. The situation is completed with S_{fin} when the e_3 event, indicating the condition of the delivered products, is announced to the 3PLcompany.

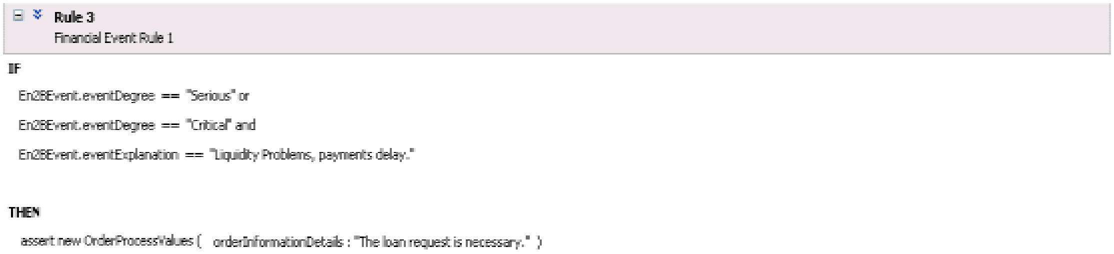
$$Seq^t[(e_1 + xe_2), e_3] \qquad (3)$$

The complex $En2BEvent$ is managed by the rule engine using the rules of scenario 1, offering a discount to the retailer according to the damage occurred.

Scenario 4: A partner expects a business deal, which will positively affect the economics of the business. The financial agreement is cancelled and liquidity problems appear. In parallel, employees go on repeating strikes. The initial state S_{init} is consisted by the xe_1 that describes the announcement of the deal cancelation and the liquidity problems, delays on employees payments xe_2 and the repeating strikes by the employees of the partner xe_3. The situation final state is S_{fin} , which arises when event e_4 is published, refering to the danger of cancelation of pending orders. The pattern detection using the event operators is as follows in (4):

$$Seq^{t3}[(!\,xe_1^{t1}, xe_2)e_3^{vt2}, e_4] \qquad (4)$$

Time intervals $t1, t2$ refer to the time passed between publishing the event, $t3$ refers to the total duration of the situation, v to the number of the repeating strikes. The situation creates two complex events $En2BEvent$ and $O2BEvent$; the first managed with the rule of fig.16 and the latter with the previous rules of scenario 2 and according to the quantity of the order.



```
Rule 3
Financial Event Rule 1
IF
  En2BEvent.eventDegree == "Serious" or
  En2BEvent.eventDegree == "Critical" and
  En2BEvent.eventExplanation == "Liquidity Problems, payments delay."

THEN
  assert new OrderProcessValues ( orderInformationDetails : "The loan request is necessary." )
```

Fig.16. The Business rule for Scenario 4

Scenario 5: Bad weather conditions in the wide area of an order delivery lead to a road closure, which is in the delivery route. The truck driver has to use a bypass road to accomplish the delivery. During transportation, the vehicle is involved in a traffic accident. The order is not delivered because of this unexpected situation. S_{init} includes the external events xe_1, xe_2; the first refers to the weather conditions that led to the road closure and the second event to the need for the 3PL company to use a detour to deliver the order. xe_3 describes the road accident and completes the situation of this scenario. The complex event $En2BEvent$ that is produced by this situation is managed by the rule engine. The pattern detection using the event operators is as follows in expression (5) as the rule fired in fig.17.

$$Seq^{t}(xe_1, xe_2, xe_3) \qquad (5)$$

Fig.17. The Business Rule for Scenario 5

The ability of our framework to detect the situations in the scenarios described above was evaluated though simulation. We simulated objects, events, relations and processes. We conducted several simulation runs with random data sets, which generated events in the logistics process described in the previous section. The simulation process aimed to detect simple events and if needed correlate them to produce complex events. It also aimed to check the occurrence of scenarios. The detection of situations is based on the expressions presented above. These expressions show the relation and sequence of events verifying the scenario (i.e. triggering the situation). Through this process we measured the number of events detected, the number of complex events created and the specific situations detected.

6.3 Evaluation Results

The capability of the framework to detect a situation in its initial state per each run set is presented in Table 8. Analysing the results, we conclude that the detection and prevention ratio is above 70%. This ratio was calculated in each test run for every scenario adding the number of states detected (S_{init}, S_{fin}) that represent the total number of potential situations, and dividing the result by the number of S_{init} state. This means that from a total of 100 situations in their S_{fin} appearing in each scenario, up to 70 were detected before their occurrence and prevented in their S_{init} state. This percentage indicates that the framework is capable to detect and finally prevent a significant amount of situations before their S_{fin} state. Due to randomness of the generated events this percentage varies, but form the executed tests we conclude that 70% is a limit for the situation detection and prevention capability of the framework.

Table 8. Percentage of Situations Detection

Runs	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
100	75%	100%	-	-	-
200	76,92%	81,8%	80%	100%	-
300	71,8%	73,68%	77,78%	87,5%	-
400	71,42%	75,86%	81,25%	76,92%	80%
500	70,5%	71,05%	77,78%	70,83%	81,82%

Table 9. Scalability with respect to Processed Events

Runs	Inter. Events created	Ext. Events created	Scenario1		Scenario2		Scenario3		Scenario4		Scenario5	
			S_{init}	S_{fin}	S_{init}	S_{fin}	S_{init}	S_{fin}	S_{init}	S_{fin}	S_{init}	S_{fin}
100	75	10	12	4	3	0	0	0	0	0	0	0
200	109	33	20	6	9	2	4	1	2	0	0	0
300	158	49	28	11	14	5	7	2	7	1	0	0
400	193	83	35	14	22	7	13	3	10	3	4	1
500	305	141	43	18	27	11	21	6	17	7	9	2

Table 9 shows the scalability of our framework with respect to the number of events processed. We observe that the number of situation detections either in initial state S_{init} or in final state S_{fin} is in accordance to the number of events published. As the data set in each simulation run is randomly generated, certain runs may not produce events (e.g. an order completed without problems). It should be noted that scenarios 3, 4 and 5 are less likely to occur because they rely on a complicated pattern that is not easily verified. This is reflected in Table 9, which shows that scenarios 3, 4 and 5 are only detected after a large number of runs. Note that the number of events does not seem to have an impact on the performance of the framework, as the response time from the test run does not transcend the 50000 milliseconds, which was the specified limit for a single test run (fig. 18).

Request | **Response**

Stress Test Status	Executed 30 of 500 tests
Number of Tests with Errors	0
Average Response Time (ms)	13183
Minimum Response Time (ms)	12284
Maximum Response Time (ms)	16809

View ▾

Thread	Loop	Duration (ms)	Start Time	End Time	Invocation Status
0	1	16476	9:22:45 PM	9:23:02 PM	Passed
1	1	16476	9:22:45 PM	9:23:02 PM	Passed
4	1	16478	9:22:45 PM	9:23:02 PM	Passed
3	1	16585	9:22:45 PM	9:23:02 PM	Passed
2	1	16809	9:22:45 PM	9:23:02 PM	Passed
2	2	12578	9:24:52 PM	9:25:05 PM	Passed
0	2	12921	9:24:52 PM	9:25:05 PM	Passed
1	2	12925	9:24:52 PM	9:25:05 PM	Passed
4	2	12922	9:24:52 PM	9:25:05 PM	Passed
3	2	12756	9:24:52 PM	9:25:05 PM	Passed
4	3	12566	9:25:55 PM	9:26:07 PM	Passed
3	3	12539	9:25:55 PM	9:26:07 PM	Passed
0	3	12594	9:25:55 PM	9:26:07 PM	Passed
2	3	12666	9:25:55 PM	9:26:07 PM	Passed
1	3	12643	9:25:55 PM	9:26:07 PM	Passed
2	4	12284	9:26:57 PM	9:27:10 PM	Passed
0	4	12350	9:26:57 PM	9:27:10 PM	Passed
3	4	12359	9:26:57 PM	9:27:10 PM	Passed

Fig.18. The response time from the set of 500 test runs

Summarizing the test results, we conclude that the proposed framework is capable of correlating events produced from different SC objects and detecting situations in the SC in a variety of time windows. The scalability of the framework and its capability to detect situations in a percentage greater than 70%, was also proved in the test runs executed.

7. Conclusion and Further Work

Data intensive supply chain processes generate a plethora of events that deliver useful information to supply chain participants. The management of such events, especially of exception events, and the reaction to them is the main objective of SCEM. In this paper, we designed and implemented an SCEM framework that is capable of detecting and processing events from different stages of the SC, correlating events within an extended time window and offering situation awareness capabilities that go beyond those of existing SCEM systems. To achieve this, we incorporated methods and technologies from CEP and graph databases under a SOA approach. With the proposed framework, situation awareness is facilitated with situation detection capabilities and achieved through the delivery of transparent information via correlations between events and between events and SC objects. Such information reveals the state of the SC in time and space and can be useful for supporting decision making and for coordinating actions by supply chain participants with the final aim to improve SC effectiveness and efficiency.

We implemented and evaluated the framework using a number of SC scenarios. We concluded that the framework is capable of correlating events produced from different SC objects and detecting more than the 70% of situations in the SC, in a variety of time windows. We can therefore argue that the framework supports our research objectives and is in synergy with the core requirements and functionalities of an SCEM system.

The key characteristics of our framework are adaptability to dynamic SC conditions, scalability with respect to number and variety of events, as well as easy technology integration using widely adopted standards such as BPEL and Web Services. The graph database utilised proved to be highly useful for event correlation and situation detection because of the way it stores events and its capability to execute complex, two or more level-deep queries, as required for event correlation and ultimately situation detection.

The main contributions of our work is on one hand the formal modelling of SC situations as a correlation of (complex) events and SC objects and on the other hand the software framework for detecting and storing situations. Researchers can build on the situation model in order, for example, to extend it with formal semantics and to add inferencing capabilities so that new situations can be derived from the ones defined explicitly. Practitioners can benefit from the situation detection capabilities of the framework and use it to populate its knowledge base with mitigating actions and receive automatic action recommendations. The framework is build using industrial strength technologies and can support the development of third party application / user interfaces. Our future work includes the development and integration into our SCEM framework of recommendation methods and algorithms that will quantify similarity between situations and hence enable the generation of recommendations based not only on actions asserted by experts but also on inferred actions which have been implemented successfully in the past for situations which were similar to the current one. To this end, we will extend our previous work on collaborative filtering recommendation techniques. Moreover, we plan to extend the framework by incorporating algorithms that can predict emerging situations, which will enhance the capabilities of supply chain managers to plan and execute processes proactively.

8. References

Adi, A. & Etzion, O. (2004). Amit - the situation manager, *The VLDB Journal*, vol. 13, no. 2, pp. 177–203, May 2004.

Akdere, M., Çetintemel, U., & Tatbul, N. (2008). Plan-based complex event detection across distributed sources. *Proceedings of the VLDB Endowment*, *1*(1), 66-77, August 23-28, ACM 978-1-60558-305-1/08/08

Baader, A. and Montanus, S. (2008). Transparency in Global Supply Chain Networks — Methods and Tools for Integrated Supply Chain Event Management. *Strategies and tactics in supply chain event management* (Vol. 368). Ijioui, R., Emmerich, H., & Ceyp, M. H. (eds.). Springer.

Bearzotti, L. A., Salomone, E. & Chiotti, O. J. (2012). An autonomous multi-agent approach to supply chain event management. *International Journal of Production Economics* 135.1: 468-478.

Blos, M. F., Wee, H. M., & Yang, W. H. (2012). Supply chain risk management: resilience and business continuity. Handbook on Decision Making, 219-236.

Bodendorf, F., & Zimmermann, R. (2005). Proactive supply-chain event management with agent technology. *International Journal of Electronic Commerce*, *9*(4), 58-89.

Brandon-Jones, E., Squire, B., Autry, C. W., & Petersen, K. J. (2014). A contingent resource-based perspective of supply chain resilience and robustness. *Journal of Supply Chain Management*, 50(3), 55-73.

Buxmann, P., von Ahsen, A., Diaz, L. M., & Wolf, K. (2004). Usage and evaluation of Supply Chain Management Software - results of an empirical study in the European automotive industry. *Information Systems Journal* , 14, 295-309.

Callioni, G., de Montgros, X., Slagmulder, R., Van Wassenhove, L. N., & Wright, L. (2005). Inventory-driven costs. harvard business review, 83(3), 135-141.

Chou, D. C., Tan, X., & Yen, D. C. (2004). Web technology and supply chain management. *Information Management & Computer Security*, 12 (4), 338-349.

Christopher, M., & Lee, H. L. (2004). Mitigating supply chain risk through improved confidence. *International Journal of Physical Distribution & Logistics Management*, 34 (5), 388–396.

Christopher, M., Peck, H.: Building the Resilient Supply Chain. *International Journal of Logistics Management*, 15(2), 1 (2004).

Cimino, M. G., Lazzerini, B., Marcelloni, F., & Ciaramella, A. (2012). An adaptive rule-based approach for managing situation-awareness. *Expert Systems with Applications*, 39(12), 10796-10811.

Davis, T. (1993). Effective supply chain management. *Sloan Management Review* , 34 (4), 35-46.

Dießner, P. and Rosemann, M. (2008). Supply Chain Event Management: Managing Risk by Creating Visibility. *Strategies and tactics in supply chain event management* (Vol. 368). Ijioui, R., Emmerich, H., & Ceyp, M. H. (eds.). Springer.

El Kassiri, A., & Belouadha, F. Z. (2015, October). Towards a unified semantic model for online social networks analysis and interoperability. In *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)* (pp. 1-6). IEEE.

Endsley, M. (2016). *Designing for Situation Awareness: An Approach to User-Centered Design*, Second Edition. CRC Press, 2016,ISBN: 9781420063585.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32-64.

Endsley, M. R., & Connors, E. S. (2008, July). Situation awareness: State of the art. In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE* (pp. 1-4). IEEE.

Erl, T. (2008). *SOA design patterns*. Pearson Education.

Etzion, O. (2009). Complex Event Processing, *Encyclopedia of Database Systems*, Springer, pp 412-413.

Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1), 455-463.

Fernández, E., Toledo, C. M., Galli, M. R., Salomone, E., & Chiotti, O. (2015). Agent-based monitoring service for management of disruptive events in supply chains. *Computers in Industry*, *70*, 89-101.

Frochlich, M. T., & Westbrook, R. (2002). Demand chain management in manufacturing and services: web-based integration, drivers and performance. *Journal of Operations Management*, 20, 729-745.

Heer, J., & Boyd, D. (2005, October). Vizster: Visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (pp. 32-39). IEEE.

Hoffman, J. M., & Mehra, S. (2000). Efficient Consumer Response as a supply chain startegy for grocery businesses. *International Journal of Service Industry Management*, 11 (4), 366-373.

Holland, C. P. (1995). Cooperative supply chain management: The impact of interorganizational information systems. *Journal of Strategic Information Systems*, 4 (2), 117-133.

Hsu, L.-L. (2005). SCM system effects on performance for interaction between suppliers and buyers. *Industrial Management & Data Systems*, 105 (7), 857-875.

Iansiti, M., & Levien, R. (2004). Strategy as ecology. *Harvard business review*, 82(3), 68-81.

Ijioui, R., Emmerich, H., & Ceyp, M. (2007). *Supply Chain Event Management*. Physica-Verlag Berlin Heidelberg, Chicago.

Knickle, K., & Kemmeter, J. (2002*). Supply Chain Event Management in the Field: Success With Visibility*. AMR Research, Boston.

Kopanaki, E. (2013), Dealing with Uncertainty in a Global Business Environment: The Need for Supply Chain Flexibility, *Proceeding of the 25th Conference of the Association for Global Business*.

lahmar, A., Galasso, F., Chabchoub, H., Lamothe, J. (2016) Conceptual Framework of Supply Chain Vulnerability, 6th International Conference on Information Systems, Logistics and Supply Chain ILS Conference, June 1 – 4, Bordeaux, France

Lawrence, E. E., & Latha, R. (2015, March). Analysis of Six degrees of separation in Facebook using Ant colony optimization. In *Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on* (pp. 1-5). IEEE.

Lee, H. L. (2002). Aligning Supply Chain Strategies with Product Uncertainties. *California Management Review*, 44 (3), 105-119.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *Sloan Management Review*, 38 (3), 93-102.

Lee, H. L., Padmanabhan, V., & Whang, S. (2004). Information distortion in a supply chain: The bullwhip effect. *Management Science*, 50 (12), 1875-1886.

Liu, R., Kumar, A., & Van Der Aalst, W. (2007). A formal modeling approach for supply chain event management. *Decision Support Systems*, *43*(3), 761-778.

Longo, F., & Mirabelli, G. (2008). An advanced supply chain management tool based on modeling and simulation. *Computers & Industrial Engineering*, *54*(3), 570-588.

Machado, A., Maran, V., Augustin, I., Wives, L. K., & de Oliveira, J. P. M. (2017). Reactive, proactive, and extensible situation-awareness in ambient assisted living. *Expert Systems with Applications*, 76, 21-35.

Magoutas, B., Mentzas, G., & Apostolou, D. (2011, August). Proactive situation management in the future internet: the case of the smart power grid. In *2011 22nd International Workshop on Database and Expert Systems Applications* (pp. 267-271). IEEE.

Malhotra, A., Gosain, S., & El Sawy, O. A. (2005). Absorptive Capacity Configurations in Supply Chains: Gearing for Partner - Enabled Market Knowledge Creation. *MIS Quarterly*, 29 (1), 145-187.

Marra, M., Ho, W., & Edwards, J. S. (2012). Supply chain knowledge management: A literature review. *Expert systems with applications*, 39(5), 6103-6110.

Mei, Y., & Madden, S. (2009, June). Zstream: a cost-based query processor for adaptively detecting composite events. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (pp. 193-206). ACM.

Nullmeyer, Robert T., et al. (2005). Human factors in Air Force flight mishaps: Implications for change. Proceedings of the 27th Annual Interservice/Industry Training, Simulation, and Education Conference. Arlington, VA: National Training Systems Association, APA.

Otto, A. (2003). Supply chain event management: three perspectives. *The International Journal of Logistics Management*, *14*(2), 1-13.

Park, J. H., Lee, J. K., & Yoo, J. S. (2005). A framework for designing the balanced supply chain scorecard. *European Journal of Information Systems*, 14, 335-346.

Pettit, T. J., Fiksel, J., & Croxton, K. L. (2010). Ensuring supply chain resilience: development of a conceptual framework. *Journal of Business Logistics*, 31(1), 1-21.

Ponomarov, S. Y., & Holcomb, M. C. (2009). Understanding the concept of supply chain resilience. *The International Journal of Logistics Management*, 20(1), 124-143.

Premkumar, G. P. (2000). Inter-Organizational Systems and Supply Chain Management - An Information Processing Perspective. In H. R. Hansen, M. Bichler, & H. Mahrer (Ed.), Eighth European Conference on Information Systems, (pp. 1017-1028). Vienna.

Ross, D. F. (2004). *Distribution, Planning and Control: Managing in the Era of Suplly Chain Management*. Kluwer Academic Publishers.

Ryoo, S. Y., & Kim, K. K. (2015). The impact of knowledge complementarities on supply chain performance through knowledge exchange. *Expert Systems with Applications*, 42(6), 3029-3040.

Schnetzler, M., Nobs, A., & Sennheiser, A. (2004). Decision Support and Strategy Formulation in Supply Chain Management. In P. Cunningham, & M. Cunningham, eAdoption and the Knowledge Economy: Issues (pp. 1231-1238). Amsterdam: IOS Press.

Scholten, K., & Schilder, S. (2015). The role of collaboration in supply chain resilience. *Supply Chain Management: An International Journal,* 20(4), 471-484.

Svensson, Goran (2000), "A Conceptual Framework for the Analysis of Vulnerability in Supply Chains," International Journal of Physical Distribution and Logistics Management, Vol. 30, No. 9, pp. 731-749.

Tang, C. S. (2006). Perspectives in supply chain risk management. International Journal of production economics, 103(2), 451-488.

Tang, O., & Musa, S. N. (2011). Identifying risk issues and research advancements in supply chain risk management. *International Journal of Production Economics*, 133(1), 25-34.

Tröger, R., & Alt, R. (2010, September). Service-oriented Supply Chain Event Management-A Case Study from the Fashion Industry. In *ISSS/BPSC*(pp. 31-42).

Tummala, R., & Schoenherr, T. (2011). Assessing and managing risks using the supply chain risk management process (SCRMP). *Supply Chain Management: An International Journal*, 16(6), 474-483.

Vilko, J., Ritala, P., & Edelmann, J. (2014). On uncertainty in supply chain risk management. *The International Journal of Logistics Management*, 25(1), 3-19.

Vlahakis, G., & Aposotlou, D. (2015, July). A FCM analysis for supply chain management. In *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on* (pp. 1-6). IEEE.

Wieland, A., Wallenburg, C.M., (2012).Dealing with supply chain risks: Linking risk management practices and strategies to performance. *International Journal of Physical Distribution & Logistics Management*, 42(10).

Winkelmann, A., Fleischer, S., Herwig, S., & Becker, J. (2009). A conceptual modelling approach for supply chain event management (SCEM). In *ECIS*(pp. 2363-2374).

Wu, E., Diao, Y., & Rizvi, S. (2006, June). High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (pp. 407-418). ACM.