Complex Event Processing Framework for Big Data Applications

Chapter · July 2016

DOI: 10.1007/978-3-319-31861-5_2

1 author:

R. Bhargavi
VIT University
19 PUBLICATIONS **164** CITATIONS

SEE PROFILE

Chapter 1:
Complex Event Processing Framework for Big Data Applications

Dr. R. Bhargavi

School of Computing Sciences and Engineering, VIT University, Chennai, India.
bhargavi.r@vit.ac.in

Abstract: The fundamental requirement for modern IT systems is the ability to detect and produce timely reaction to the occurrence of real-world situations in the system environment. This applies to any of the Internet of Things (IOT) applications where number of sensors and other smart devices are deployed. These sensors and smart devices embedded in IOT networks produce huge amounts of data continuously. These data streams from heterogeneous sources arrive at high rates and need to be processed in real-time in order to detect more complex situations from the low level information embedded in the data. Complex Event Processing has emerged as an appropriate approach to tackle such scenarios. Complex Event Processing is the technology used to process one or more streams of data/events, identify patterns of interest from multiple streams of events and derive a meaningful conclusion. This chapter proposes CEP based solution to continuously collect and analyze the data generated from multiple sources in real time. Two case studies: Intrusion detection in a heterogeneous sensor network and automated healthcare monitoring of geriatric patient are considered for experimenting and validating the proposed solutions.

Keywords: Complex Event Processing, IOT, Data Stream, Event, Intrusion detection, Geriatric health monitoring

1.1 Introduction

A wide range of IoT applications have been developed and deployed in recent years. IoT has provided promising solution to several real time applications by leveraging the growing ubiquity of radio-frequency identification (RFID), and wireless, mobile, sensors and other smart devices [7]. Big Data is a term encompassing the use of techniques to capture, process, analyze and visualize potentially large datasets in a reasonable time frame not accessible to standard IT technologies [11]. The platform, tools and software used for this purpose are collectively

called “Big Data technologies”. It refers to the ability to crunch vast collections of information, analyze it instantly, and draw conclusions. Big data technologies deal with petabytes of records, files, transactional data either arriving as streams or in batches. The RFIDs, sensors, smart devices etc., embedded in IOT networks produce huge amounts of data/events continuously [19]. This data has the characteristics like volume, variety, velocity, variability, veracity, and complexity. Some of the characteristics of the data/event streams and their implications are as follows:

- Data streams are continuous and sequential and are ordered by a timestamp or any other attribute value of the data item. Therefore data items which belong to the same stream are processed in the order they arrive.
- Data streams are generated by external sources and are sent to a processing system. Hence the data stream processing system does not have any direct control over the data sources.
- The input characteristics and the rate of a data stream are unpredictable. The input rate can be very irregular and at times, bursty in nature. Also, the nature of the input does not allow one to make multiple passes over the data while processing.
- The amount of data is very large and unbounded. Therefore, processing requirements may not permit persistence followed by processing. However data or summary of data can be stored for archival or other purposes.
- The data types of the data items can be structured, semi-structured or unstructured.
- Data items in a data stream are not error free because the data sources are external. Some data may be corrupted or discarded due to network problems.

These data streams need to be processed and analyzed to identify some interesting patterns and take actions if necessary. Processing these continuous data/event streams in real time to identify the patterns among them is a herculean task and has raised new research challenges over the last few years [3, 5]. A large number of solutions exist in terms of systems, middleware, applications, techniques, and models proposed by researchers to solve different challenges [13]. Data generated from multiple sources have logical and spatio-temporal relations among them. There is a need for data fusion as the data from a single source may not be enough for taking accurate decision. Conventional process oriented control flow software architectures do not explicitly target the efficient processing of continuous event streams. Complex Event Processing is the technology used to process and analyze one or more streams of data/events, identify patterns of interest from multiple streams of events to derive a meaningful conclusion and respond by taking appropriate action.

The remaining part of the chapter is organized as follows. Section 1.2 presents CEP preliminaries and event modeling. Semantic intrusion detection using complex event processing is elaborated in section 1.3. Section 1.4 discusses the CEP enabled Geriatric health monitoring, and Section 1.5 concludes the chapter.

1.2 Complex Event Processing

Complex Event Processing is relatively new, but it has got wider acceptability due to its systematic and multi-level architecture driven concept approach. An event is an object that is a record of an activity. The event signifies the activity. A key stroke, the output reading produced by a sensor etc., are couple of examples of an event. CEP allows one to set request for an analysis or some query and then have it executed continuously over a period of time against one or many streams of events in a highly efficient manner. CEP is all about the processing of events that combines data from many sources to infer events or patterns that represent more complicated circumstances. In contrast to Hadoop's two-stage disk-based MapReduce paradigm, CEP's push based paradigm supports faster processing of data streams.

1.2.1 CEP Architectural Layers

The architectural layers of the CEP system are shown in Figure 1.

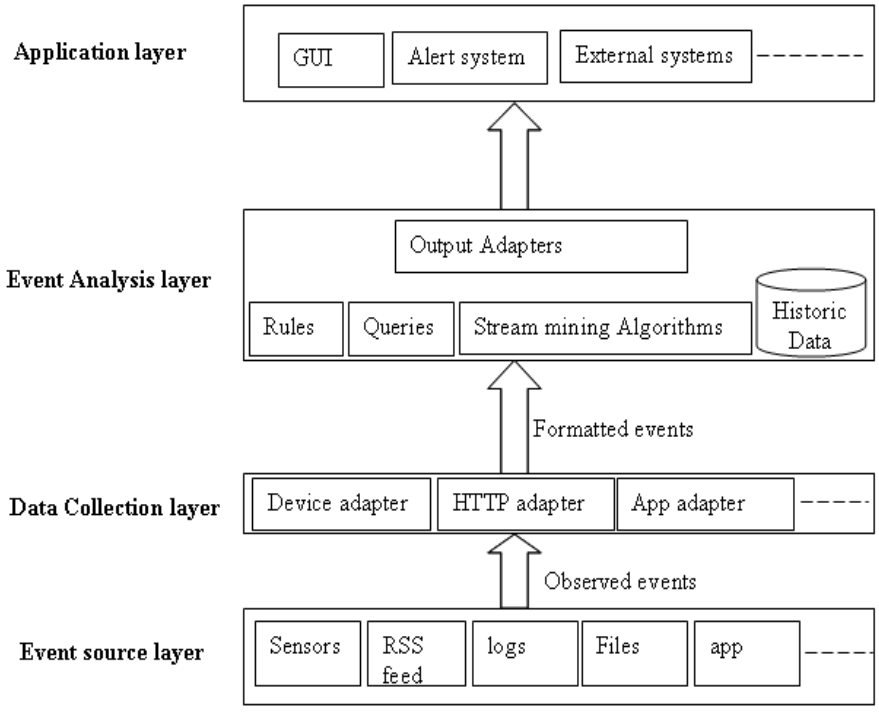

Figure 1 Architectural layers of CEP

Event Source Layer

This layer consists of the event sources. Event sources can be sensors, RFID readers, RSS feed, feed from network monitoring systems, web logs etc.

Data Collection Layer

Data collection layer is responsible for collecting the data coming from various sensors, and filtering the data. Data collection could be distributed or centralized based on the requirement of the application. Basic capabilities in WSN involve mechanisms like routing, tunneling, data aggregation, clustering to collect information from nodes and forward them to a sink node.

Event Analysis Layer

The event analysis layer is responsible for data mining on streams of data. An important feature of this layer is pattern matching and correlation across multiple event streams. In this layer few fixed continuous queries act on the incoming data streams to identify patterns of interest. Comparison of real-time data with historical or static data is also often required for event analysis. In many instances historical or static data is also often required along with the real-time data for analysis. Thus the event analysis layer should contain facilities for connecting to data base.

Application layer

This layer consists of event listeners i.e. modules/systems that receive the processed events. Examples of event listeners are event storage systems, mobile phones and pagers, or other systems that can take actions based on the results of event processing (e.g. GUI receives the processed events and displays the information).

There are two underlying stages involved in CEP. The first step is to detect meaningful events or pattern of events which signifies either threats or opportunities from the streams of events. The second step is to send alerts to the responsible person/entity for the identified threat or opportunity for quick response. CEP solutions and concepts can broadly be classified into two categories.

1. Computation oriented CEP
2. Detection oriented CEP

In computation oriented CEP solutions, on-line algorithms are executed whenever an event or data enters the system. Simple example is to calculate the moving average temperature sensed by a temperature sensor.

Detection oriented CEP concentrates on detecting combinations of events or event patterns. Simple example is to look for a sequence of events.

Complex Event Processing system is made up of number of modules like input adapters, output adapters and event processing modules such as event filtering modules, in-memory caching, aggregation, database lookups module, database writes module, correlation, joins, event pattern matching, state machines, dynamic queries etc as shown in Figure 2. In order to support more flexibility and adaptability for different use cases more number of I/O adapters must be supported by the CEP. The main component of CEP is the continuous queries which monitor streams of simple/raw events for so-called complex events, that is, events that manifest themselves in certain temporal, spatial or logical combinations. Querying events over data streams is different from traditional querying with database, in the way that traditional database querying is pull based whereas continuous querying of events is push based.

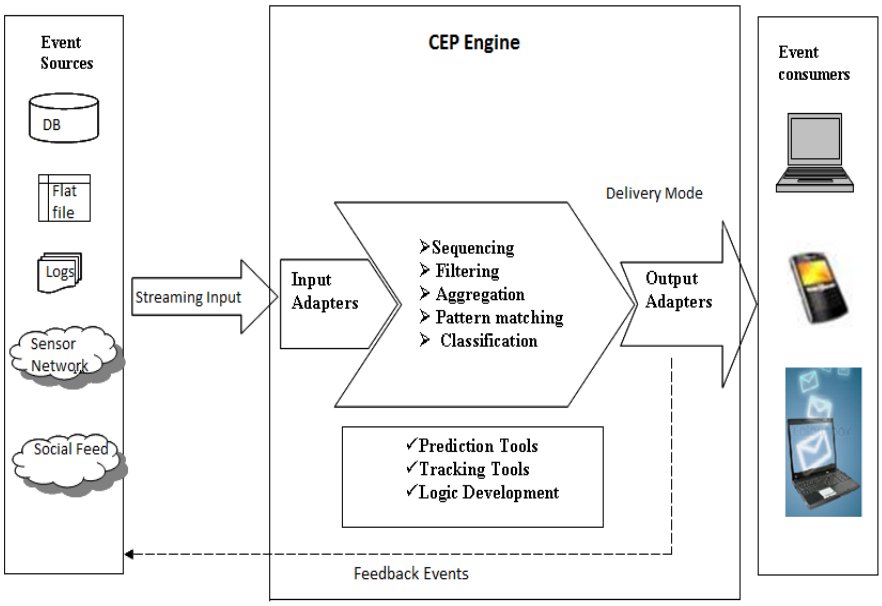

Figure 2 Logical view of CEP

Many of the real time distributed applications require continuous monitoring and processing and analysis of information or data in a timely fashion as it flows from periphery to the system. Intrusion detection in surveillance and healthcare monitoring are a couple of such applications. Traditional pull based approach can hardly address the requirements of timeliness, response generation etc. Hence this chapter proposes CEP based solution to continuously collect and analyze the data generated from multiple sources in real time. Two case studies: Intrusion detection in a heterogeneous sensor network and automated healthcare monitoring of geriatric patient are considered for experimenting and validating the proposed solutions.

1.2.2 Event Modelling

An event is "anything that happens, or is contemplated as happening" [10] in the real world and normally it is of interest to some group of people. A key stroke, a sensor outputs a reading etc., are couple of examples of an event. Sometimes these events, in turn, may produce secondary events internally. Real world occurrences can be defined as events that happen over space and time. Events are of two types: 1 Basic / primitive events, 2 Complex events. Events have event attributes. An event attribute is a property of the event. For example, the entry of an identified person in a restricted area could be treated as an activity. Then the form of the event instance could be composed by unique id of the person, time, and location (geographical coordinates). An event is an object that represents, encodes, or records an event, generally for the purpose of computer processing [10].

A basic event is atomic, indivisible and occurs at a point in time. Attributes of a basic or primitive event are the parameters of the activity that caused the event. An event instance is a record of an activity, which has three features such as:

- Significance, which gives its semantics.
- Form, which gives activity information that will be processed by the computer. E.g., unique id, time etc.
- Relation with other event instances.

Computer systems process the events by representing them as event objects. From a software application perspective, an event is something that needs to be monitored and may trigger a specific action. Specifying an event is therefore providing a description of the happening. A common model of an event is a tuple represented as:

$E = E (id, a, t)$

here id - is the unique ID of an event,

$a = \{a_1, a_2, \ldots\ldots, a_m\}$, $m > 0$, is a set of attributes

t is the time of occurrence of the event.

For example an RFID event can be described with some set of dimensions which includes source of event, location of event, time at which the event occurred and a possible set of operations for combining events. An RFID event is denoted as E = e (o, r, t) where o is the tag EPC, r is the reader ID and t is the time stamp of the event.

Complex events are composed of basic events. Complex events are defined by connecting basic events using temporal, spatial or logical relations. A common model for a complex event is as follows:

$E = E (id, a, c, t_b, t_e)$, $t_b <= t_e$

where $C = \{e_1, e_2, \ldots..,e_n\}$, $n > 0$ is the vector that contains basic events and complex events that cause this event to happen; t_b, t_e are starting and ending times of the complex event.

Attributes of complex events are derived from the attributes of the constituent primitive events. Event constructors and event operators are used to express the relationship among events and correlate events to form complex events.

Any basic event or a complex event is specified by an Event Expression. An event expression is a mapping from histories (domain) to histories (range) [6].

E: histories → histories.

Since event expressions are equivalent to regular expressions it is possible to implement event expressions using finite automata.

For example Composite/ Complex Event = $\mathbf{E_1}$ ^ $\mathbf{E_2}$ can be represented using the finite automata as shown in Figure 3.

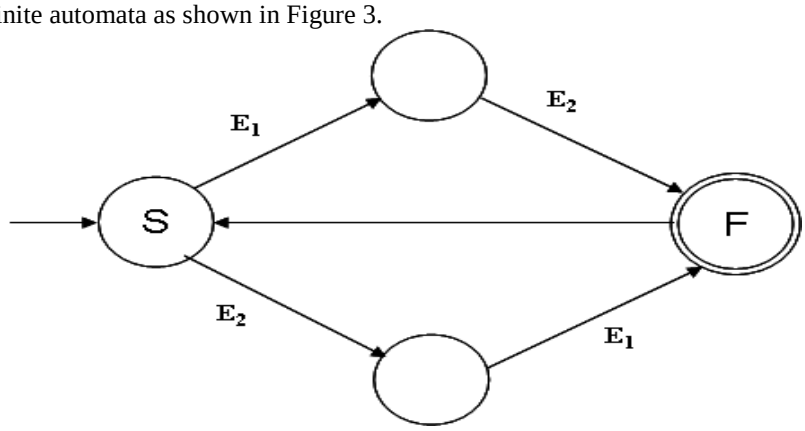

Figure 3 Finite automata representation of composite event E1 ^ E2

Event expression is formed by combining events with the event constructors. Many of the event processing engines support different types of logical and temporal constructs.

1.3 Semantic intrusion detection using complex event processing

There is an increasing demand for security solutions in the society. This results in a growing need for surveillance activities in many environments. Recent events, like terrorist attacks, have resulted in an increased demand for security in society. There is a growing interest in surveillance applications, because of the availability of cheap sensors and processors at a reasonable cost. Intelligent remote monitoring systems allow users to survey sites from remote location. Sensor networks bridge the gap between the physical world and the virtual world of processing and communication. It is envisioned that sensor networks can reduce or eliminate the need for human involvement in information gathering and processing in surveillance applications if the ability of the sensor networks can be suitably harnessed. Distributed sensors and smart devices in the surveillance application produce huge data [14]. Processing, analyzing and detecting abnormal patterns from this data are very complex in nature. There is a need for multisensor data fusion in sensor net-

work applications as the data from homogeneous sensors may not be enough for taking accurate decision [17]. Also the data generated from multiple sensors have logical and spatio-temporal relations among them. Special processing and mining algorithms are proposed in the literature for distributed and In-network processing. Existing solutions for surveillance and intrusion detection are based on pull based architecture where the data from sensors are stored and later, the user runs queries to retrieve the data. There are also solutions based on machine learning techniques but these solutions again work on the static data. Traditional approaches which are pull based can hardly address the requirements of timeliness, response generation etc. The proposed CEP based Semantic Intrusion Detection System (CSIDS) addresses the above mentioned problems.

In the context of surveillance, the evaluation of available technologies shows that the security bottleneck is not the hardware, but rather the real-time analysis and correlation of data provided by various sensors. The objective of proposed CEP based Semantic Intrusion Detection System is early detection and prevention of security compromises/risks by identifying abnormal situations and quickly responding with appropriate action. To achieve this goal, sensors (such as cameras, RFID readers etc.) are installed in the places to be monitored and a control center receives the information transmitted by sensors. This information is processed and analyzed by an event processing engine to detect the intrusive patterns. There are few factors that make this task rather difficult. First, the raw data to deal with is very huge, and second the information sources are heterogeneous. Therefore, there is a need to fuse or aggregate the data coming from such sensors to construct a global view of the situation, and even more to go one step beyond to the intent assessment.

The proposed CSIDS allows fusion of information generated by heterogeneous sensors that support the goal of providing a global situational view for intrusion detection. The goal of this fusion is to transform lower-level data into higher-level quality information and to improve the certainty of situation recognition i.e., reduce the false alarm rate. The fusion is realized by taking into consideration the semantics of the information. This means that, there should be a representation or model of the situation that need to be detected (such as an intrusion in a restricted area). In this context, a situation is viewed as a combination of several activity elements (or smaller situations), each of them appearing at some place and time. As a consequence, the finer the granularity of these elements, the more difficult the situation modeling task will become. This task is further complicated if any of the elements appear at different time instants and do not relate to the same situation. Furthermore, the set of heterogeneous sensors capturing these sub-situations have an asynchronous behavior, meaning each of the sensors compute the situation independently, unconnected of each other.

CSIDS supports online detection of event patterns which represent anomalies. The association of an event pattern, a constraint and an action is referred to as a rule. The approach consists in time, content and context based selection of a subset of events described by their pattern. Finally after the event pattern is matched,

one or more appropriate actions are executed such as, creation and sending of a new event or sending an alert message to the concern personnel etc.

The architecture of the Complex Event Processing System for Semantic intrusion detection system is shown in Figure 4. CSIDS has multiple event receivers. Each event receiver receives the data/ events coming from a different data/event source. The event receiver on receiving the data from the source converts them into event streams. Data generated by different sources follows a different format hence Event receivers also convert the data from different sources to the specific format suitable for processing further by the event processing engine. The events generated by the event receivers are inserted in to a FIFO or a Queue. Events are organized in the queue in the order of their detection time. To avoid the out of order arrival of the events which is caused by the network delays, the events are stored internally in the queue for some tome T. Hence the events are dequeued after a time T for processing. Any event getting generated at time t will be processed after t+T time by the CEP engine. Any event arriving the queue with a time stamp smaller than the time stamp of the already dequeued events is ignored. This leads to missing of events. Missing events due to network delays can be avoided by having larger value of T i.e. by storing the events internally for longer duration. But this will delay the event processing time. Hence there is a tradeoff between missing events due to network delay and latency.

Event processing engine processes the event streams. The events generated from heterogeneous sensors are collected, aggregated using logical and spatiotemporal relations to form complex events which model the intrusion patterns. All the rules and patterns are to be registered initially. The listeners are intimated whenever the corresponding rule is hit or pattern is matched for which it is configured. Modeling of the complex events using event expressions is explained later in this section. Notifiers are used for intimating the Listeners about the rule or pattern occurrences. Listeners are the modules that take necessary action on notifications.

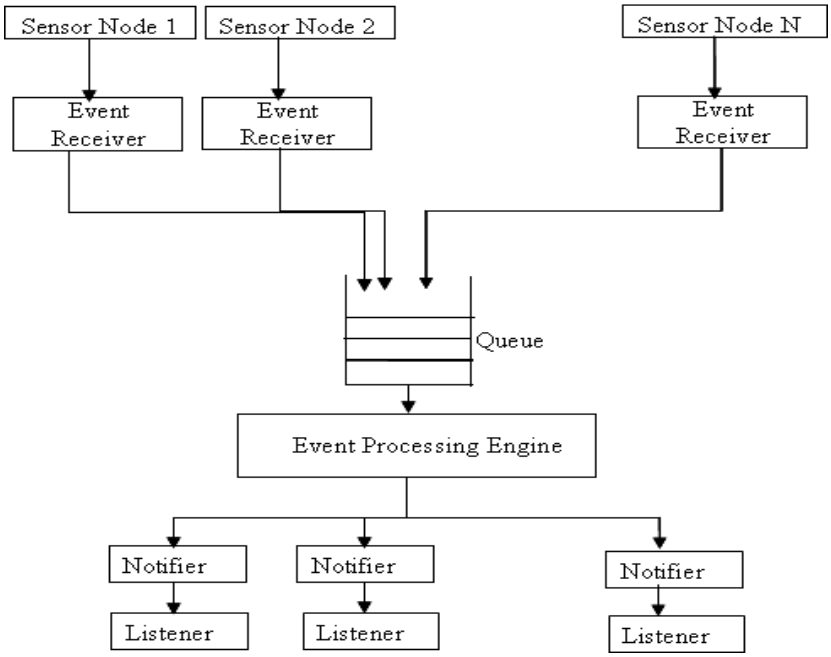

Figure 4 CEP architecture for Semantic Intrusion Detection System

There are several supporting modules in Semantic IDS to perform the activities on the occurrence of certain events. These modules are Kalman tracking module, person detection module, authentication module, etc. Person tracking is done using kalman filter. All these modules are listeners to the CEP engine.

Figure 5 shows how the physical events generated from different sensors can be aggregated to generate a complex event which represents a pattern or a scenario of interest. Sensed information from the sensors that interact with the physical environment/world is collected by the event receivers. Various scenarios representing simple and complex events have been modeled using event expressions.

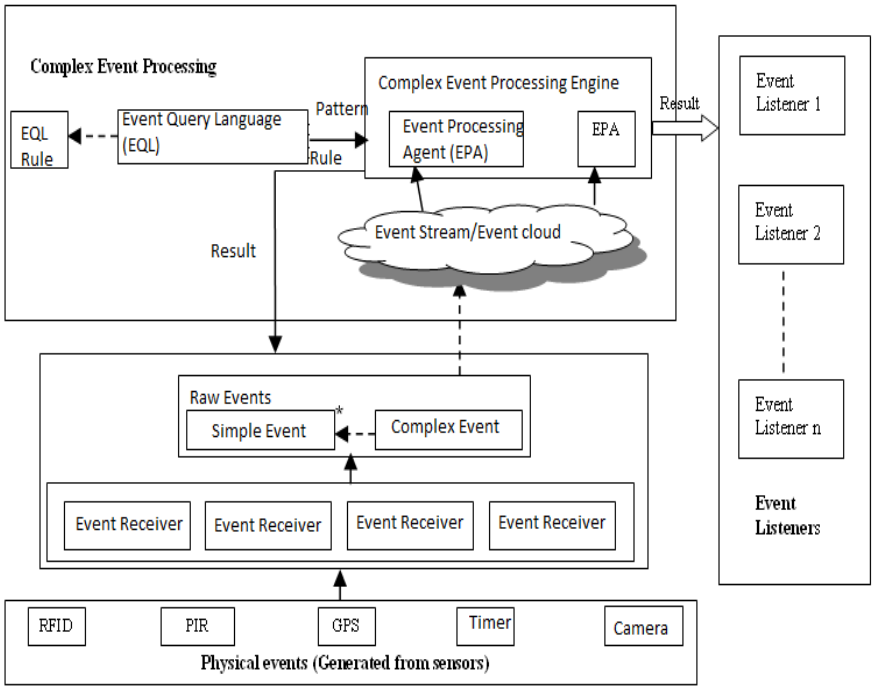

Figure 5 CSIDS

Following Primitive events considered for the present study:

- Events generated during the interaction between the RFID readers and tags
- PIR readings generated whenever a person crosses the sensor
- GPS readings indicating the location of event occurrence
- Time of Event occurrence
- Images captured by the camera

One of the complex event scenarios and its modeling using event expressions are discussed below:

Example: Unauthorized entry

When a person who does not possess a valid RFID tag tries to enter a location just behind an authorized person it is called as tailgating. This scenario can be captured as explained below.

CEP queries the wireless data for an event where PIR is present but RFID is zero. Once such an event is identified, CEP gets the image from the database corresponding to that event's timestamp and gives the image to Haar Face Detection module. The Haar Face detection algorithm counts the number of persons present and gives the count back to CEP. This scenario can be modeled as follows:

E1: Office entry

E1 = (s1, o1, t1) type(s1) = RFID event

E2: Office entry

E2 = (s2, o1, t2) type(s2) = PIR event

E3: Multiple people sensed by Image sensor

The complex pattern can now be formulated as:

every((E2: ¬ (E1)) ^ E3)

Now the rule can be developed to rise an alert on the above complex event using IF – THEN condition

IF (True)

Get the Image with the same Time stamp and Node ID from the Data Base.

Send the location of the Image in the Data Base to the Image processing module for checking number of people.

Receive the response from the Image processing module.

Generate alert if needed.

End

Implementation and Validation

The proposed CSIDS is validated using a sensor network. A wireless sensor node is devolved which consists of a Passive Infra-Red sensor, a Radio Frequency Identifier reader, GPS receiver, and timer. 10 such sensor nodes are used for deploying the wireless sensor network. Some of the nodes are connected with cameras. The camera is triggered by the PIR sensor and is connected to a Personal Computer. The images captured are stored in PC and transmitted to the server via wired network. A unique ID is assigned to each sensor node. Data from the RFID and PIR sensors are sent to the sink using wireless communication (using Zigbee module). Data from the cameras is sent to the server using Ethernet. This is done to overcome the bandwidth, memory, processing and power limitations of Sensor Node. The complete CSIDS is developed using JAVA. ESPER event processing engine is used to implement and execute the rules. Patterns representing intrusion are modeled as complex events which in turn are aggregated from base events and other complex events using logical and spatiotemporal relations. Rules have been developed to identify the unauthorized entry of the people in to the security zones, tailgating issue and few other scenarios. It is observed that the proposed CSIDS identifies the intrusion patterns efficiently in near real time. It is also observed that the proposed system out performs the pull based solutions in terms of detection accuracy and detection time. The quality of results depends on the quality of sensors and processes of their data. For example if the sensors take more time to sense the data, process and communicate, then this introduces a delay between the occurrence of real situation and its recognition by the application which can become considerable in sensitive situations.

1.4 CEP enabled geriatric health monitoring

Healthcare monitoring is another application which can best exploit the advantages of CEP. This section explains the proposed CEP based Geriatric Health Monitoring System (CGHMS). Advancements in wireless body area networks have led researchers to exploit the usage of these technologies in healthcare applications [4, 15]. Availability of wearable and cost effective physiological and motion sensors allow automated health monitoring of elderly people. There are a number of healthcare applications using sensor networks. Within the hospital, there are three applications: (1) to track people and objects around the hospital [16], (2) to safeguard use of equipment and (3) to assist medical personnel with their jobs. Another important application is elder care in home. Global increase in the ratio of elderly people [8] in the world requires alternatives to the traditional home care technologies that are available today. This in turn indicates that there will be challenges related to giving proper care to the elderly people since care giving requires enough people and resources. Hence, there is a need for alternatives that automate the home care application domain.

The important requirement of healthcare monitoring is automatic anomaly detection of vital parameters. Another important requirement is that when something dangerous or critical occurs, it is important that the system detects this immediately and that the delay is minimal. There are several existing solutions for monitoring the activity of a person or abnormality of a particular health parameter. But there is no solution for identifying the abnormal situations like fall of a person by combining the vital parameters, activities of the person and the context information. CEP enabled geriatric healthcare monitoring system (CGHMS) collects data generated from physiological and environmental sensors and detects the abnormalities in vital parameters and fall of a person.

The main objective of the CEP enabled geriatric healthcare monitoring is to collect health parameters from the patient, detect the presence of an abnormality in vital parameters and provide feedback to the rules in decision making about the health condition and falls of a geriatric patient. In the geriatric health care monitoring domain, Complex Event processing involves analyzing raw sensor data and recognizing next level of events like low BP, high BP, high temperature to complex events like person has fallen etc. The data/event streams are filtered correlated and aggregated to identify any abnormal patterns. Rules/patterns of interest corresponding to various scenarios are stored in the knowledge base. Rules are executed on the incoming data to detect the anomalies. Every rule has associated action to be executed like sending SMS to care giver or doctor or patient. Figure 6. Shows the architecture of the CEP enabled geriatric health monitoring system. The physical and semantic data flows are shown in figure. The proposed CGHMS uses wearable BP sensor, pulse oximeter, Bioharness 3 device, a Zephyr product which contains both bio sensors and tri-axial accelerometer for measuring the vital parameters like respiration/ breathing rate, heart rate, ECG and movement along X, Y and Z axis respectively. CGHMS also uses RFID reader which can be con-

nected and used with IPAQ PDA device. RFID tags and environmental sensors are placed throughout the area to be monitored to get the context information. Data generated by these sensors are collected and sent to a central server where CEP engine. The knowledge base consists of CEP rules which model the various abnormal health condition events. The CEP engine executes all the rules whenever it receives the sensor data/facts to detect abnormality situations. Every rule has associated action to be executed like sending SMS to the care giver or doctor or patient.

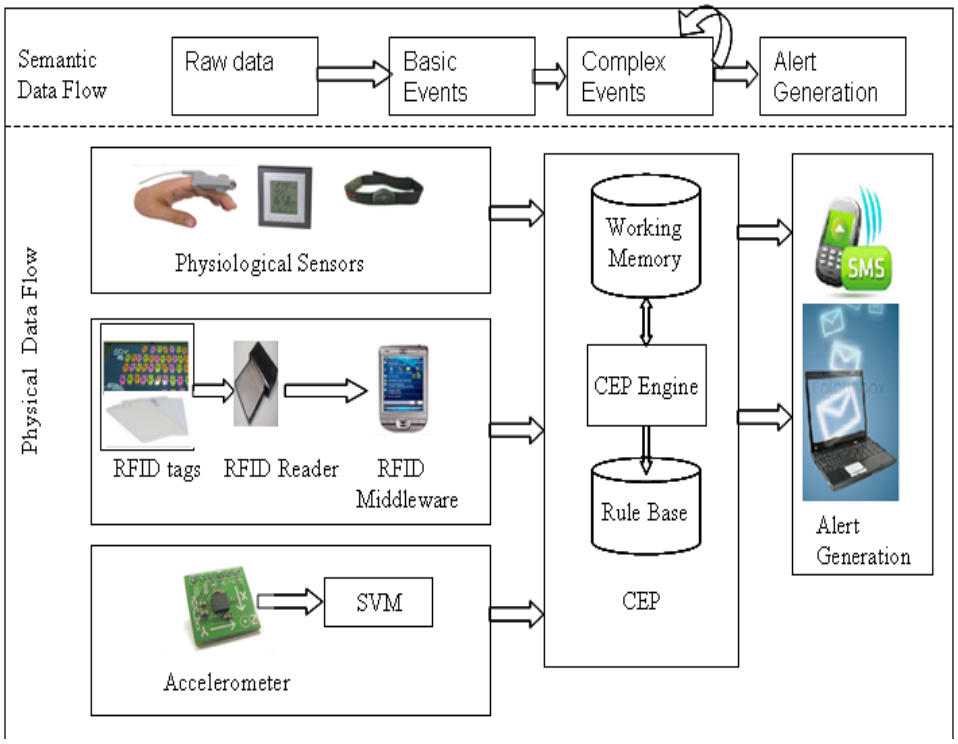

Figure 6. CGHMS architecture

Implementation and Validation

The complete CGHMS is implemented in JAVA. JAVA based DROOLS expert engine is used for rule development. DROOLS APIs are extended to support CGHMS. To reduce the false alarms in fall detection, the proposed CGHMS uses vital parameter data, activity of the person, RFID information which gives the contextual information and a camera. For activity classification using tri axial accelerometer and gyroscope SVM classifier is used [12].The proposed CGHMS

provides continuous accessibility to vital parameters and other relevant data, and analysis of the data. The CGHMS improves the quality of life of the elderly people by monitoring the well-being and alerting in case of emergency. The proposed CGHMS is tested and validated in real time environment using Zephyr BioHarness device, Pulseoxio meter, BP sensor, for measuring the vital parameters, RFID tags & reader to get the context information, activity of a person using accelerometer and gyroscope, and camera. It is observed that CGHMS detects the abnormalities in the vital parameters and fall of a person more precisely compared to using the individual sensors. As discussed earlier, performance of the proposed CGHMS depends on the quality of the sensors used. Also, the wearable sensor device must be worn by the person correctly in the right position, otherwise the captured data by the sensors may be incorrect or sometimes the parameters may not be captured at all. Other problems which affect the performance of the system include, operating condition of the device, battery charging, availability of the communication links etc.

1.5 Conclusion

To conclude, this chapter discusses CEP framework for Big Data applications. Two real time case studies have been dealt thoroughly. Early identification of significant complex events provides situational awareness and better decision making. Complex Event Processing enables sense-and-respond behavior, in which incoming events or information is used to assess the current situation and generate a response in a timely fashion. This chapter proposed CEP based solutions to Internet of Things where early identification of significant complex events provides situational awareness and better decision making. Two real time applications, Sematic Intrusion Detection and Healthcare monitoring are implemented and validated to demonstrate the power of CEP. Patterns representing intrusion and abnormal situations are modeled as complex events which are aggregated from base events and other complex events using logical and spatiotemporal relations. The CEP based solutions provide the capabilities of heterogeneous information fusion coming from several kinds of sensors to get a global situation view and take necessary action in near real time. The proposed semantic intrusion detection system is found to outperform the existing solutions in terms of detection latency (i.e. time delay between the event occurrence and event detection), and detection accuracy i.e. event identification with less false positives and false negatives. The proposed CEP based solutions require domain knowledge of the application. All the situations/scenarios of interest should be well understood and modeled as rules using the events and available constructors. Failing to which, the pattern representing the complex event (event of interest) may not be captured as expected. Missing events is another common problem in case of wireless sensors. Missing basic events will lead to the failure of getting the high-level complex situations captured. The quality of results depends on the quality of sensors. For example if the

sensors take more time to sense the data, process and communicate, then this introduces a delay between the occurrence of real situation and its recognition by the application which can become considerable in sensitive situations.

In any of the Complex Event Processing based applications, the system is initially configured with all the rules (defined by the domain experts) which define the complex events or patterns to be identified and alerted. But there are several applications in which new rules need to be added or deleted at run time. Hence the proposed solutions can be extended to support dynamic rule addition and rule deletion. CEP is a rule based technology for detecting known patterns of events and reacting to the identified situations in real-time. Incremental or dynamic learning is used to learn the dynamic environments effectively. Hence the proposed solutions can be extended to incorporate dynamic learning, and the newly learned knowledge/ concepts can be given as feed back to the CEP engine to add new rules and identify new patterns or situations of interest. This makes the complete system more adaptive and intelligent.

References

1. Bhargavi, R & Vaidehi, V (2013) "Semantic Intrusion Detection with Multisensor Data Fusion using Complex Event Processing", Sadhana – Academy Proceedings in Engineering Sciences, ISSN: 0256-2499, vol. 38 , no. 2, pp. 169 – 185
2. Bastian Hoßbach and Bernhard Seeger (2013) "Anomaly management using complex event processing: extending data base technology paper", In Proceedings of the 16th International Conference on Extending Database Technology (EDBT '13). ACM, New York, NY, USA, pp.149-154, 2013
3. Cantoni, V, Lombardi, L & Lombardi, P (2006) "Challenges for data mining in distributed sensor networks". ICPR, vol. 1, pp. 1000-1007
4. Carmen CY, Poon, Qing Liu, Hui Gao, Wan-Hua Lin & Yuan-Ting Zhang (2011) "Wearable Intelligent Systems for E-Health", Journal of Computing Science and Engineering, vol. 5, no. 3, 2011, pp. 246-256
5. Elnahrawy E (2003) "Research directions in sensor data streams: solutions and challenges". DCIS, Technical Report DCIS-TR-527, Rutgers University
6. Gehani N H, Jagadish H V and Shmueli O (1992) "Composite event specification in active databases: Model and implementation", VLDB '92: Proceedings of the 18th International Conference on Very Large Data Bases, 327–338. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
7. Jiong Jin, Gubbi J, Marusic S, Palaniswami M (2014) "An Information Framework for Creating a Smart City Through Internet of Things" Internet of Things Journal, IEEE , vol.1, no.2, pp.112 – 121 doi: 10.1109/JIOT.2013.2296516

8. Kevin Kinsella & Wan He (2009) "An aging world: 2008. International population reports", U.S. Department of Health and Human Services

9. Luckham D C (2010) The power of events: An introduction to complex event processing in distributed enterprise systems. Boston, MA, USA: Addison Wesley Longman Publishing Co., Inc

10. Luckham, DC & Schulte R (2008) "Event Processing Glossary – Version 1.1.", Event Processing Technical Society. URL: <http://www. epts.com/component/option,com_docman/task, doc_download/gid, 66/Itemid,84/>.

11. NESSI White Paper, December 2012.

12. Palaniappan, Adithyan, Bhargavi, R, & Vaidehi, V (2012) "Abnormal human activity recognition using SVM based approach", proceedings of IEEE International Conference on Recent Trends in Information Technology (ICRTIT 2012), Chennai, India, pp. 97-102, April 19-21

13. Perera C, Zaslavsky A, Christen, P, Georgakopoulos D (2014) "Context Aware Computing for The Internet of Things: A Survey," Communications Surveys & Tutorials, IEEE , vol.16, no.1, pp.414-454

14. Tian He., Sudha Krishnamurthy., Liqian Luo., Ting Yan., Lin Gu., Radu Stoleru., Gang Zhou., Qing Cao., Pascal Vicaire., John A Stankovic., Tarek F Abdelzaher., Jonathan Hui., Bruce Krogh(2006) "VigilNet: An integrated sensor network system for energy-efficient surveillance", ACM Trans. Sen. Netw, Vol.2, No.1, pp. 1–381, 2006

15. Vaidehi, V, Bhargavi, R, Ganapathy, Kirupa, Sweetlin Hemalatha, C (2012) "Multi-sensor based in-home health monitoring using Complex Event Processing", proceedings of IEEE International Conference on Recent Trends in Information Technology (ICRTIT 2012), Chennai, India, pp.570-575, April 19-21

16. Wen Yao, Chao-Hsien Chu, and Zang Li Yao, W., Chu, C., Li, Z (2011) "Leveraging complex event processing for smart hospitals using RFID", Journal of Network and Computer Applications, Vol.34, Issue 3, pp. 799-810

17. White, Jr.F.E (1987), "Data fusion lexicon",Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, Naval Ocean Systems Centre, San Diego

18. Wood A, Virone, G, Doan, T, Cao, Q, Selavo, L, Wu, Y, Fang, L, He, Z, Lin & Stankovic, S (2006), "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring", Technical Report, Department of Computer Science, University of Virginia. Wireless Sensor Network Research Group

19. Zaslavsky A, Perera C, and Georgakopoulos D (2012) "Sensing as a service and big data," in International Conference on Advances in Cloud Computing (ACC-2012), Bangalore, India, July 2012, pp. 21–29

Index