International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 2
ISSN: 2088-5334

Activity Recognition for Smart Building Application Using Complex Event Processing Approach

Rabiah Adawiyah Shahad[#], Mohamad Hanif Md Saad[*], Aini Hussain[#]

[#]*Department of Electrical, Electronic & System Engineering, Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia*
E-mail: rabiahshahad@siswa.ukm.edu.my, draini@ukm.edu.my

[*]*Department of Mechanical & Material Engineering, Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia*
E-mail: hanifsaad@ukm.edu.my

***Abstract*— Activity recognition has become one of the most interesting and challenging subjects in performing surveillance or monitoring of smart building system. Although there are several systems already available in the market, limitations and several unresolved issues remain, especially when it involves complex engineering applications. As such, activity recognition is purposely incorporated in the smart system to detect simple and complex events that happen in the building. In all existing event detections, the complex event processing (CEP) approach has been used for the detection of complex events. The CEP is capable of abstracting meaningful events from various and heterogeneous data sources, filtering and processing both simple and complex events, as well as, producing fast mitigation action based on specific scenarios. The work reported in this paper intends to explain in detail on the development of activity recognition application using CAISER™ and NESPER© platform as well as the complex event detection that uses the CEP approach. In assessing the system performance, Matthew Coefficient Correlation (MCC) has been used as the main performance parameter. Results obtained showed that the Temporal Constraint Template Match Detector (TCD) is more accurate, stable and better in complex event detection compared to NESPER© detector.**

***Keywords*— complex event processing; Matthew Coefficient Correlation; multi-layered event detector; surveillance system; smart building**

I. INTRODUCTION

The evolution of smart buildings is increasing at a tremendous phase. It unofficially began in 1950 [1] when pneumatic actuator was used for building automation purposes. In 1960, the technology changed from air compressors to the analog controller and then to the digital controller. Then, the home automation became popular from the year 1970 to 2010 [2]. The open protocol has been introduced in 1990, whereas the wireless technology was introduced since the year 2000. From 2000 onwards, home or building automation was established and stabilised with the presence of the internet and ambient intelligence. Smart buildings are equipped with the automation system facility and integration technology [3] to measure, monitor, control and optimize operations and maintenance, life security, telecommunication, consumer system and facility management systems [4].

Activity recognition is an important part of the smart building system. it involves the use of image processing and artificial intelligence. Activity recognition monitors the environment or infrastructure automatically, with minimal intervention by manpower [5]. This system involves an automatic detection and traction as well as advanced analysis. There are several components that contribute to input data acquisition such as sensor devices, computer vision and social messaging.

There are a number of challenges when developing an activity recognition system in smart buildings such as detection of a complex event, handling and managing huge amounts of data, controlling the mechanical and magnetic motor for the event actor, etc. It is obvious that there are lots of data produced from the network of sensors attached, especially in a building. The data produced contains information that can be used for prediction and action. It is needed so that the user can extract meaningful information / events wisely.

Currently, among the most popular techniques for event detection includes Hidden Markov Model (HMM), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Conditional Random Fields, Naïve Bayes and Tree (NBTree). Artificial Neural Network (ANN) based technique such as Multi-Layer Perceptron (MLP) is also quite popular among researchers in this field. [6]. To further

increase the accuracy of detection, some researchers fused several detection techniques and selected the best-predicted result using the ensemble-based approach. Some researchers also used CEP based framework utilizing tools such as Drools, NESPER©, Siddhi, and APAMA to detect meaningful events as well as produce fast mitigation action for related events. These tools mostly use the Sliding Windows (SW) technique to detect complex events from a sequence of simple events. This paper highlights the development of activity recognition, which is a form of a complex event, in smart buildings using the CEP based event detector.

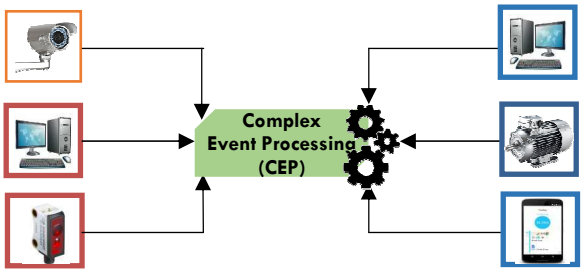

Fig. 1 CEP concept diagram

A. Terms & Definition

There are many terms used in event processing. The important terms are highlighted in Table 1.

TABLE I
TERMS & DEFINITIONS USED IN EVENT PROCESSING

Term	Definition
Event	Something significant happens in real-time and can lead to an increase in other events
Event Stream	A group of events that arrive in a context to a processor that has an event topic.
Simple Event / Raw Event / Primitive Event	Refers to any change in events, either in the form of value or things taken out directly from the real world. Events that occur directly and not from a combination of other events.
Complex Event / Composite Event	Extraction from a combination of simple events or when added to previous complex events
Event Producer	Detection of changes in circumstances that come from a variety of sources and is represented as an event.
Event Processor	Identify and process events as well as generate appropriate mitigation actions.
Event Consumer	Event User receives information or notification about events.
Event Actor	The final component that receives instructions to implement mitigation action.
Event Channel / Event Communication	Routes that connect the event producer to the event processor and from the event processor to the event consumer.

B. Complex Event Processing

CEP is defined as an approach to acquiring, analyzing, detecting and processing voluminous events in real-time and produces a quick mitigation action based on specific scenarios. CEP consists of three main parts, which are the event observer (sources), event processor, and event consumer (sinks). The CEP concept is illustrated in Fig. 1. CEP is responsible for filtering and combining messages/notifications that predict higher level events that are then notified to sinks. CEP uses an event processing language (EPL) for event detection. The event represents the entity that occurs in the CEP field or state change [7], [8]. Events consist of two main types, which are the simple events (*smoke_triggered*, *temperature_high*) and complex events (*smoke_triggered* followed by *temperature_high* that predicts the *Fire_detected*). CEP correlates several simple events to detect higher level complex events [7]. It is commonly applied in environmental monitoring, financial trading, business process automation, and control systems.

C. CEP Engine

There are lots of CEP platforms present in the market such as Drools, RuleCore, Progress Apama, Coral8, StreamBase, Esper, etc. The performance of these CEP engines is summarised in Table 2.

TABLE II
PERFORMANCE SUMMARY OF CEP ENGINE

Vendor	Inf. Type	License	Supp.	Dev.	Comp. Lang.
Progress Apama	Rule-based	Commercial	Strong Market Presence	Studio	EPL
StreamBase	Rule-based	Commercial	Good	Studio	EPL
Coral8	Rule-based	Commercial	Good	Java API	EPL
Esper and NEsper	Rule-based	Open Source	Support	API C# Java	EPL
TIBCO BusinessEvents	Rule-based	Commercial	Good	Studio	EPL
MS Streaminsight	Rule-based	Commercial	Good	API C#	EPL
Rapide	Rule-based	Open Source	Not Support	API Java	EPL
RuleCore	Rule-based	Free	Not Support	API Java	EPL

Esper is an open-source CEP engine with two implementations, which are Esper© for Java and NEsper© for .NET. Esper uses rule-based as an inference engine and API C# as well as API Java in the platform development. Esper is usually used in production environments, provides better documentation (well-written, extensive documentation spanning hundreds of pages), easy to understand, has clear concept of combining composition operators with data stream constructs, provides an Eclipse plugin and is easy to install and use, can access data using the API for implementing languages and can accept database query from event queries.

D. CEP Architecture

There are four main modules for CEP system, such as Event Producer, Communication Channel, Event Processor, and Event Consumer. Fig. 2 depicts the overall block

diagram for the CEP system. The event producer is responsible for identifying simple events as an input stream and send it to the Event Processing Network (EPN) or complex event processor via suitable communication channels. The communication channel connects the input or output adapter to the EPN or complex event processor. There are a number of communication channels, such as TCP/IP, database, JSON, XML, web service, etc. The complex event processor is used to detect the complex events by using several correlation techniques, such as temporal correlation, logical or causal correlation, and spatial as well as dimension-based correlations. Event consumers or event actors are responsible for connecting to the action gate. The mitigation actions initiated could be unified messaging or triggering the actuator.

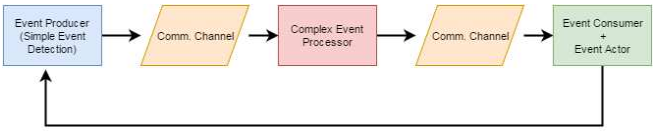

Fig. 2 Block diagram of CEP system

E. CEP Event Detection

There are two main methods for event pattern detection in CEP, which are Exact Match and Similarity Matching.

1) Exact Match

In this technique, all elements in event sequences need to be matched [9]. This technique has a low computational cost. However, it is exposed to detection failure due to the noise and system error that normally comes with event sequences. The clean sequences will result in high accuracy and precision with fast detection and low computational costs.

2) Similarity Matching

There are several criteria used in similarity matching such as Hamming distance [10] and Editing operational numbers (delete, add, edit), which are executed to make both sequences similar [11]. Whereas, [12] uses distance in between patterns and sequences approach to determine the level of similarity matching. These matching approaches have been used in Non-Finite Automata (NFA) [13] and in the artificial intelligence technique, such as Hidden Markov Method (HMM) [14], [15] applications.

F. CEP Application

There are several CEP applications that have been implemented. The application highlighted in this study is the engineering field. The conventional data analysis and mining system need human operators to analyse the huge volumes of generated data. In addition, the system still lacks the unknown event pattern identification. Hence, the automated technique was developed by using a combination of statistical learning theory algorithms and event processing systems. The complex event processing is highly recommended to be deployed since it is capable of obtaining various sources of events, predict what would happen and can be applied in real-time data analysis.

1) Wireless Sensor Network

The increase in sensor networks has led to the need for managing and detecting meaningful events (simple and complex). There are two sensor network approaches, namely Centralized and Distributed processing [16]. The centralized approach sends input sources to the gateway for processing. However, this approach uses more energy and transmission latency causes a delay in event detection. Whereas, the distributed approach allows evaluation on the node before sending the events to the gateway. These approaches reduce communication overhead and increase the system's performance.

In [17], an engine that allows detection of complex events and meaningful information from raw/primitive events is proposed. There are three main parts in this system architecture, which are the server side, sink side, and node side. Results show that their in-network complex event will increase the lifetime of the network compared to the centralized approach

2) Intrusion Detection

The intrusion detection system is a critical system. The system needs to have the ability to process various sources of data in real-time and produce multiple actions such as notifying engineering personnel and also providing relevant data to stop the attempts. Hence, the CEP methods are highly suitable for adoption in this kind of system. A generic Intrusion Detection and Diagnosis System (ID2S), has been implemented in [18]. These systems are capable of detecting and diagnosing complex intrusion scenarios in Large-Scale Complex Critical Infrastructures using comprehensive alert correlation workflows. They also use hybrid and hierarchical approaches for on-line detection and diagnosis. In addition, a CEP has been used for complex event correlation.

3) Smart Homes

A smart home [16] can be described as a home/building that is equipped and integrated with surveillance and ambient devices with the purpose of creating safety, security, comfort, communication, entertainment and technical management. The technology used in a smart home can be the Bluetooth, bus operated systems, X10 standards, main borne communication systems, radio frequency transmission, infrared communication, and insteon. CEP technology uses a framework infrastructure work best suited for the smart home application. Smart homes can increase safety, automate the task, monitor home activity remotely, enhancing energy efficiency, technology accessibility and provide comfort to the user. Smart home systems still need further development in terms of providing security for all connected items and also needs to find efficient ways to reduce energy consumption as well as faster and efficient actions that need to be taken.

DIGIHOME [19] aims to provide comprehensive and simple solutions in a pervasive environment whenever dealing with context processing. DIGIHOME uses Esper with a Java open source stream event processing engine to process event management and make a decision.

4) Activity Recognition System

CEP has been used in an activity recognition system. [20] uses CAISER™ as a CEP engine to monitor and analyse the

complex event produced by the CCTV and door sensors. The strength of the CEP platform is in its complex event detection. In addition, these platforms provide multiple built-in classifiers for comparison purposes. The system has an average accuracy of more than 90%. However, these systems are limited to single people tracking. In [7], event-driven architecture (EDA) is used for data processing and will process in near real-time. The communication between data and other entities is via messages. [7] deploys the CEP paradigm in its system since it allows all EDA features, filter information, and enable high-level information to be inferred. The system combines EDA and CEP engines in the overall system.

5) Event Monitoring System

[19], [21] uses NESPER© in the CEP engine for the SAPHE (Smart and Aware Pervasive Healthcare) project. This system aims to monitor vital signs and patient activity. However, this system has a limitation of discovery new services and unclear interaction between actuators and the engine.

II. Materials And Method

A. Region of Interest

Event recognition infrastructure has been installed in the Innovation 2 Laboratory, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia. Fig. 3 shows the exact Region of Interest (ROI) inside the laboratory environment. There are 9 CCTV units, 4 door sensor units, a door actuator, a lamp switch and an air conditioner switch. Whereas, there are 14 ROIs that are marked in order to obtain the actor movement reading.

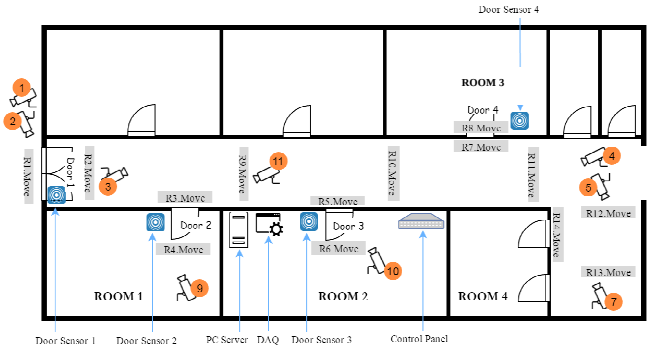

Fig. 3 Region of interest in smart building

B. Event Classification

Table 3 gives a description of simple events (SE). For example, door sensors are annotated as `Door1.Open`, `Door1.Closed`, `Door2.Open`, `Door2.Close`, etc. Whereas, CCTV ROI are annotated as `R1.Move`, `R2.Move` until `R14.Move`.

Table 4 depicts the sample of complex events (CE) description. For example, `Door1_NotClosedProperly` is triggered when the door is opened and not closed for a certain period of time (e.g., 3 minutes). The rules for triggering the `Door1_NotClosedProperly` is `Door1.Opened` → `Door1.NotClosed` → `Door1.NotClosed` → `Door1.NotClosed`.

TABLE III
Simple Event Description

Item	Event Class	Item	Event Class

Door Sensor	`Door1.Open`
	`Door1.Close`
	`Door2.Open`
	`Door2.Close`
	`Door3.Open`
	`Door3.Close`
	`Door4.Open`
	`Door4.Close`
		CCTV	`R1.Move`
			`R2.Move`
			`R3.Move`
			`R4.Move`
			`R5.Move`
			`R6.Move`
			`R7.Move`
			`R8.Move`
			`R9.Move`
			`R10.Move`
			`R11.Move`
			`R12.Move`
			`R13.Move`
			`R14.Move`

TABLE IV
Sample of Complex Events Description

Complex Event	Description	Rules
`Door1_Not Closed Properly`	Door1 is not closed for 3 minutes	`Door1.Opened` → `Door1.NotClosed` → `Door1.NotClosed` → `Door1.NotClosed`
`Person1_Exit Room3`	Person1 exited room3	`Door4.Opened` → `R8.Move` → `R7.Move`
`Person1_Enter Room2`	Person1 enters room2	`R5.Move` → `Door3.Opened` → `R6.Move`
`Person1_Enter Room1From Room2`	Person1 enters room 1 from room2	`R6.Move` → `Door3.Opened` → `R5.Move` → `R9.Move` → `R3.Move`

C. CEP Platform

There are two CEP platforms used in this research study, which are CAISER™ and NESPER©. CAISER™ [22] is a component-based CEP development platform suitable for developing CEP systems for Engineering and Scientific applications. It is equipped with a set of readily build Event Generator and Event Actuator Adapters (such as OPC Server Adapter, Data Acquisition Board Adapter, Remote Terminal Unit Adapter, Smart CCTV Adapter and Indoor Surveillance Robot Adapter). It is also equipped with numerous Communication Channel Adapters; thus, enabling it to receive data as well as relay commands and information via SMS, Email, XMPP and bare TCP/IP. CAISER™ also supports inter-component communication via popular social messaging applications such as Telegram, Skype, Twitter, and Jabber. CAISER™ utilizes several Complex Event detection algorithms to detect Complex Events by fusing previous and current Simple Event information with previously detected Complex Event information. With CAISER™, developing CEP-based intelligent engineering and scientific applications are as simple as connecting the components and configuring the event detection and mitigation rules. Whereas, the NESPER© [23] provides a CEP-ESP component for .NET applications. It is an open

source software available under the GNU General Public License (GPL).

D. Rules Language & Tool

In general, event detection rules are represented ontologically to ease rule creation. This approach is used not only for CEP applications but in other applications as well, especially when the data comes from heterogeneous sources and the data analysis is made based not only on current information but utilizes historical data patterns as well. [24]. For example, NESPER© uses SQL-like query language, which is ESPER© Event Processing Language (EPL), to represent the rules; whereas, CAISERTM uses its CAISERTM Event Processing Language (CEPL) to represent its rules.

E. CEP Framework

Fig. 4 shows the CEP framework for this research. The status of movements in ROI's, sensors and triggered switches are sent as simple events directly to CAISER™ event detector to be processed in online mode. The event detector then analyses the pattern of past and present simple events and past complex events to identify current complex events. For comparison purposes, this study extracted the simple events from CAISER's database and re-annotated the complex events so that comparative offline analyses could be made between CAISER's Complex Event Detector (CED) and NESPER©.

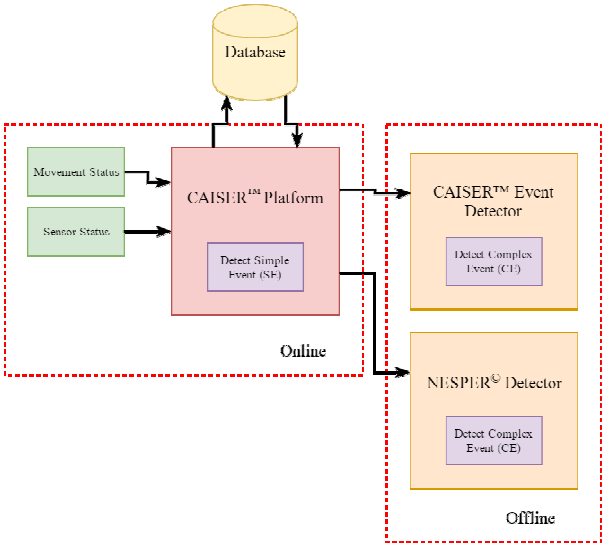

Fig. 4 CEP Framework

Table 5 depicts the details of the number of complex events that occurred in the provided dataset for experimental purposes. There are 26 types of complex events that occurred in the experiments with the total sequences of 972 simple events. The data was taken from the real-time data with several manual annotations.

TABLE V
DETAILS ON THE NUMBERS OF COMPLEX EVENTS

#	Complex Event (Level 2)	Σ	#	Complex Event (Level 2)	Σ
1	Door1. NotClosed	11	14	Actor1.Enter Room2From Room3	3
2	Door2. NotClosed	9	15	Actor1.EnterRoom2 FromOutside	3
3	Door3. NotClosed	19	16	Actor1.EnterRoom3 FromRoom2	3
4	Door4. NotClosed	13	17	Actor1. EnterRoom3From Room4	3
5	Actor1. EnterBuilding	11	18	Actor1.ExitBuilding FromRoom1	3
6	Actor2. ExitBuilding	14	19	Actor1.ExitBuilding FromRoom2	3
7	Actor1. EnterRoom1	12	20	Actor1.ExitBuilding FromRoom3	4
8	Actor1. ExitRoom1	9	21	Actor1.PassedBy Room2	15
9	Actor1. EnterRoom2	22	22	Actor1.PassedBy Room4	22
10	Actor1. ExitRoom2	9	23	Actor1.Waiting Outside	24
11	Actor1. EnterRoom3	14	24	Actor1.ExitBuilding FromRoom1	3
12	Actor1. ExitRoom3	8	25	Actor1.EnterRoom1 FromRoom4	1
13	Actor1. EnterRoom1 FromRoom2	1	26	Actor1.EnterRoom1 FromOutside	4
Total Sequence = 972					

F. Complex Event Detector

There are 4 CE detectors used in this experimental study, which comprises the sliding window detector (SWD), weighted sub-window sliding window detector (WSD), temporally constrained template matching (TCD) and NESPER© [22]. Both NESPER© and SWD uses similar sliding window techniques for detection but different ones in the rule language usage. NESPER© uses event processing language (EPL), whereas SWD uses CAISER™ event processing language (CEPL). On the other hand, WSD algorithm is based on SWD, with the addition of weight function for matching the algorithm. Meanwhile, TCD adds a temporal constraint to its algorithm. All of these CE detectors produce different performances.

Fig. 5 shows an example of the SWD detection technique [22]. For example, 5 sequences of events, such as **a → d,b → b → b → e,c,** are matched with 4 sub-rules templates (**a → d → b → e**) based on a window size of 4. Exact matches with the current sub-rules are coloured in green and give a confidence factor value of CF=1. The unmatched sequences are coloured in red and produce CF=0.

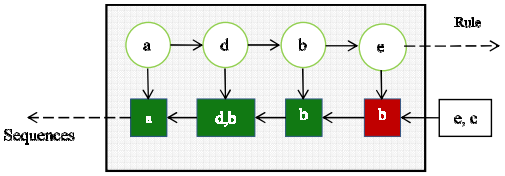

Fig. 5 SWD detection technique

Fig. 6 depicts the detection technique referred to WSD. For example, 5 sequences of events, such as **a → d,b → b → b → e,c**, were matched with 4 sub-rules templates of **a→ d → b → e** with a window size of 4. The green colour represents the exact matches rules, whereas the orange colour represents the matches rules for events detected before or after the current rules. Each detection will produce different CF based on the weighting function that has been set.

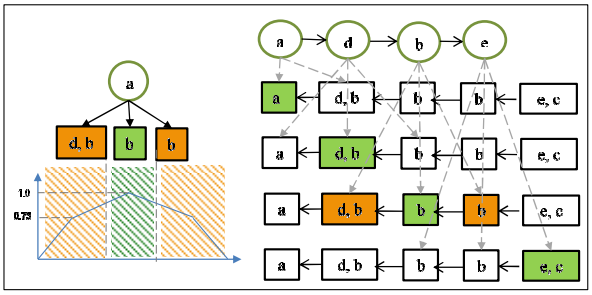

Fig. 6 WSD detection technique

Fig. 7 depicts the TCD detection technique. This method takes into account an alpha and beta parameter and is adjusted based on the needs of any temporal challenge that is detected. The alpha value represents the number of confidence factors, maximum overall waiting time and minimum overall waiting time. Whereas, beta values represent the sub-rules parameter, which is the waiting time before detection starts and the waiting time for a current sub-rule matching move to the next sub-rule if there is no detection in current time. For example, 7 sequences of events, such **as e → b,a → d,a → a → b → b,c → e**, need to be matched with 4 sub-rules, which are **a → d → b → e**. In row 1, Beta (0,1) = 0,2 represents the waiting time before starting the detection that equals to zero and the waiting time for the current sub-rule matching the move to the next sub-rule that equals to two. Thus, the unmatched detection of the first sub-rule of a for the first sequence of e will wait until 2 seconds to move to the next sub-rule. The matches were found at the next second for sequences of **b,a**. Further explanation of these detection techniques is found in the main developer manuscript [22].

Fig. 8 shows the NESPER© detection technique. The matching process between event sequences and the rules template will take into account the desired window size, pattern matching used and quantifier search that has been set. All of these parameters are in the standard EPL parameter. For example, 5 sequences of events, such **as a → d,b → d,e → b → e** need to be matched with 4 sub-rules templates (**a → d → b → e**) with the desired sliding window of 5 and the quantifier search used being ***.** In this case, the quantifier search used is written in this form **A B* C D**. This means that the sub-rule number 2 can occur more than once. The brief explanation is found in Table 6.

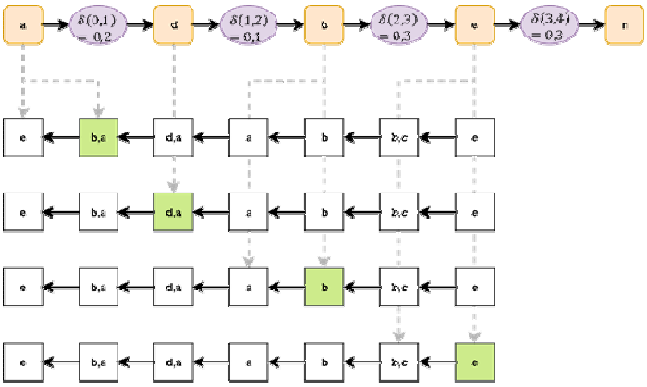

Fig. 7 TCD detection technique

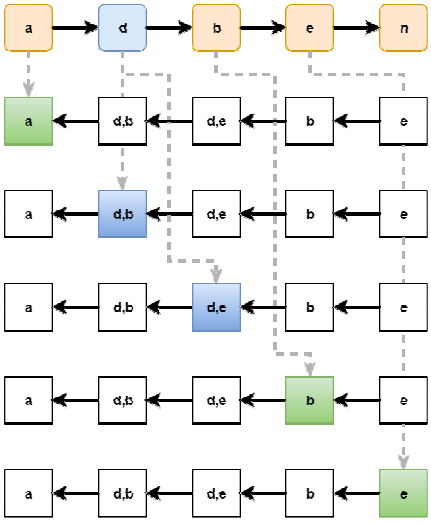

Fig. 8 NESPER© detection technique

TABLE VI
EXAMPLE OF NESPER DETECTION PROCESS

Row	Description
Row 1	Exact match is detected (Green colour box). Current sub-rule is **a,** and the current sequence is **a**. No quantifier used for this first sub-rule.
Row 2 & Row 3	Quantifier ***** was detected (Blue colour box). Thus, this rule can check more than once. The sequences of **d,b** (row 2) and **d,e** (row 3) are matched with the sub-rule number 2, which is **d**.
Row 4	No matches with sub-rule number 2 and quantifier. Thus, proceed to the next sub-rule. Exact match is detected (Green colour box). Current sub-rule is **b,** and the current sequence is **b**. No quantifier used for this first sub-rule.
Row 5	Exact match is detected (Green colour box). Current sub-rule is **a,** and the current sequence is **a**. No quantifier used for this first sub-rule.

III. RESULTS AND DISCUSSION

The performance parameter in this study was measured using the Confusion Matrix (CM) and Matthew's Coefficient

Correlation (MCC). CM a method used to describe the classification model's performance with known true values on a set of test data. It is a common technique for the performance measure. However, CM has an unreliable metric for accuracy when there were unbalanced data sets that eventually led to misleading results. Hence, the MCC technique is used to support the detection of a balance measurement by considering true and false positives and negatives.

The results show that the performance for each activity is different. Table 7, Table 6, Table 9 represents activities such as "*Door3.NotClosed*", "*Actor1.ExitBuilding FromRoom3*" and "*Actor1.PassedByRoom2*". Based on the average results from the event analysis in Table 10, the fastest average operation time was by the TCD detector (2.355 ms) followed by SWD (8.689 ms), WSD (24.19 ms) and lastly, NESPER© (34111.88 ms). TCD is the fastest because it has no weighting function. While SWD and NESPER© have similar detection technique are dependent on the size of the sliding window. The bigger window size, the longer the searching period. NESPER© has the longest detection due to bigger window size and complex rule structure. The event processing language (EPL) structure of NESPER© can be improved more so that the processing time can be smaller at least almost similar to the SWD detector.

For overall average accuracy and specificity value, all event detectors achieved the highest value of above 95% with the TCD detector (Acc.=99.5%) leading the accuracy performance. However, this study will also consider the MCC value to support the accuracy parameter. The acceptable MMC value is supposedly above 50%. The highest average MCC was obtained by the TCD detector (74.4%) followed by NESPER© (68.8%) and SWD (60.0%). The MCC value plays an important role in the performance parameter measure due to the balance of true / false positive and negative values. WSD had an average MCC of lower than 50% due to the unstable weighting function included in its algorithm.

The sensing system was evaluated based on human movement activities, which were entering and exiting a building / room. The overall highest average for precision and sensitivity values belonged to the TCD detector with 73.6% and 81.1% respectively. The results considered the amount of noise present in the background environment. For example, the ambient light present during daylight and night light might affect the movement detection. Hence, the user needs to set up a higher sensitivity level for both daylight and night light. In this study, we have applied the sensitivity from 0.8 to 0.95.

TABLE VII
RESULTS OF "DOOR3_NOTCLOSED"

Detector	Acc.	Sen.	Spec.	Prec.	MCC	t_{ave} (ms)
SWD	0.99	1	0.99	0.63	0.789	8.689
WSD	0.92	1	0.918	0.179	0.405	24.190
TCD	0.986	1	0.985	0.548	0.735	2.355
NESPER	0.989	1	0.988	0.607	0.775	34111.88

TABLE VIII
RESULTS OF "ACTOR1_EXITBUILDINGFROMROOM3"

Detector	Acc.	Sen.	Spe.	Pre.	MCC	t_{ave} (ms)
SWD	0.999	0.75	1	1	0.866	8.689
WSD	0.973	1	0.973	0.133	0.36	24.190
TCD	0.999	0.75	1	1	0.866	2.355
NESPER	0.999	1	0.999	0.75	0.866	34111.88

TABLE IX
RESULTS OF "ACTOR1_PASSEDBYROOM2"

Detector	Acc.	Sen.	Spe.	Pre.	MCC	t_{ave} (ms)
SWD	0.997	0.833	0.999	0.909	0.869	8.689
WSD	0.954	1	0.953	0.25	0.488	24.190
TCD	0.996	0.833	0.998	0.833	0.831	2.355
NESPER	0.993	0.444	0.998	0.667	0.541	34111.88

TABLE X
1 RESULTS OF OVERALL AVERAGE

Detector	Acc.	Sen.	Spe.	Pre.	MCC	t_{ave} (ms)
SWD	0.992	0.703	0.993	0.561	0.600	8.689
WSD	0.960	1.000	0.960	0.172	0.395	24.190
TCD	0.995	0.811	0.996	0.736	0.744	2.355
NESPER	0.993	0.925	0.993	0.553	0.688	34111.88

Based on overall average performance from all CE detectors, it was found that the TCD detector was better than the NESPER© detector. Thus, it is recommended that the TCD detector becomes the baseline comparison detector for future algorithm development.

IV. Conclusions

This paper had provided a basic concept of activity recognition in smart buildings using the CEP approach. The experimental studies involved the use of two CEP engines, namely CAISER™ and NESPER©. Several complex event detection techniques were also used such as TCD, SWD, WSD, and NESPER©. Results showed that TCD was the best detector for complex event detection with lower time latency, higher MCC, precision value as well as higher accuracy. NESPER© was also found to be good for complex event detection. However, NESPER© is limited to exact matching technique, and the parameter is restricted. TCD is better in terms of detailing in detection rule, and the parameter is easy to vary based on the situation.

It is recommended that future research use more complex activities and standard data using similar CEP detectors such as TCD. The TCD detector could be used as a baseline detector for future algorithm development. On the other hand, the NESPER© rule can be improved by managing the EPL structure and developing the weighting function for each sub-rule. To conclude, this study believes that the research in complex event processing will proliferate as the demand for an efficient surveillance system in smart buildings increases in order to ensure stability, security, safety and a comfortable environment for society and mankind.

Acknowledgment

This work was funded in parts by Universiti Kebangsaan Malaysia grant code DIP-2015-012 and Ministry of Higher Education grant code FRGS/1/2016/ICT02/UKM/02/7. The authors would also like to express their sincere gratitude to CAISER team, affiliates of the CAISER@SESRG laboratory, for their guidance and assistance in this project.

References

[1] C. Solutions, "The Evolution Of Smart Buildings," *t.pt.*, 2017. [Online]. Available: http://controlyourbuilding.com/the-evolution-of-smart-buildings. [Accessed: 02-Feb-2017].

[2] J. Towler, "Evolution of Smart Building and Their Place in the Internet of Everything," in *14th International Conference for Enhanced Building Operations*, 2014, no. September.

[3] I. Strictest, C. Confidence, P. One, I. Strictest, and C. Confidence, "Smart Building enable Smart City," p. 22, 2016.

[4] S. Wendzel, J. Tonejc, J. Kaur, and A. Kobekova, "Cyber Security of Smart Buildings," *Secur. Priv. CyberPhysical Syst. Found. Appl. Chapter 16, Ed. H. Song, G. Fink, S. Jeschke, G. Rosner, Wiley, Press*, pp. 1–28, 2016.

[5] S. Ibrahim, "A comprehensive review on intelligent surveillance systems," vol. 1, pp. 7–14, 2016.

[6] M. Sufyian, M. Azmi, and N. Sulaiman, "Accelerator-Based Human Activity Recognition Using Voting Technique with NBTree and MLP Classifiers," vol. 7, no. 1, pp. 146–152, 2017.

[7] R. Gad, M. Kappes, J. Boubeta-Puig, and I. Medina-Bulo, "Employing the CEP paradigm for network analysis and surveillance," in *Proceedings of the Ninth Advanced International Conference on Telecommunications*, 2013, pp. 204–210.

[8] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, 1st ed. Addison-Wesley Professional, 2002.

[9] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman, "Querying event sequences by exact match or similarity search: Design and empirical evaluation," *Interact. Comput.*, vol. 24, no. 2, pp. 55–68, 2012.

[10] A. M. Gil-lafuente, "Decision-making techniques with similarity measures and OWA operators," vol. 36, no. January 2012, pp. 81–102, 2013.

[11] P. Moen, *Attribute, Event Sequence and Event Type Similarity Notions for Data Mining*. 2000.

[12] H. Obweger, "Similarity Searching in Complex Business Events and Sequences thereof," *Citeseer*, no. 0, pp. 1–117, 2009.

[13] Y. Mei and S. Madden, "ZStream : A Cost-based Query Processor for Adaptively Detecting Composite Events Categories and Subject Descriptors," *Proc. 35th SIGMOD Int. Conf. Manag. data*, vol. pages, pp. 193–206, 2009.

[14] R. Agrawal, K. Lin, H. S. Sawhney, and K. Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," *Proc. 21st Int. Conf. Very Large Databases*, pp. 490–501, 1995.

[15] S. Guler, W. H. Liang, and I. A. Pushee, "A video event detection and mining framework," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, 2003, vol. 4, p. 42.

[16] K. S. Pooja, K. T. Chandrashekar, M. Thungamani, G. B. C. N, A. W. Is, and A. S. Home, "Complex Event Processing In Smart Homes," no. 3, pp. 544–550, 2015.

[17] O. Saleh, "Complex Event Processing in Wireless Sensor Networks," *Wiat 2010*, pp. 211–214, 2010.

[18] M. Ficco and L. Romano, "A Generic Intrusion Detection and Diagnoser System Based on Complex Event Processing," *2011 First Int. Conf. Data Compression, Commun. Process.*, pp. 275–284, 2011.

[19] D. Romero, G. Hermosillo, A. Taherkordi, R. Nzekwa, R. Rouvoy, and F. Eliassen, "RESTful integration of heterogeneous devices in pervasive environments," in *IFIP International Conference on Distributed Applications and Interoperable Systems*, 2010, pp. 1–14.

[20] R. A. Shahad, G. B. Leow, M. H. M. Saad, and A. Hussain, "Complex Event Detection in an Intelligent Surveillance System using CAISER Platform," *2016 Int. Conf. Adv. Electr. Electron. Syst. Eng.*, 2016.

[21] G. E. Churcher and J. Foley, "Applying and extending sensor web enablement to a telecare sensor network architecture," in *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, 2009, p. 6.

[22] M. H. M. Saad, "Pemprosesan Peristiwa Kompleks Untuk Aplikasi Sistem Kejuruteraan Pintar," Universiti Kebangsaan Malaysia, 2017.

[23] EsperTech, "NEsper for .NET," *EsperTech Inc.* [Online]. Available: http://www.espertech.com/esper/about_nesper_dotnet.php. [Accessed: 01-Jan-2017].

[24] F. Ramli, S. Azman, and M. Noah, "Building an Event Ontology for Historical Domain to Support Semantic Document Retrieval," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, pp. 1154–1160, 2016.