ENHANCING PERFORMANCE AND EXPRESSIBILITY OF COMPLEX EVENT PROCESSING USING BINARY TREE-BASED DIRECTED GRAPH

BABAK BEHRAVESH

UNIVERSITI TEKNOLOGI MALAYSIA

ENHANCING PERFORMANCE AND EXPRESSIBILITY OF COMPLEX EVENT PROCESSING USING BINARY TREE-BASED DIRECTED GRAPH

BABAK BEHRAVESH

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

FEBRUARY 2016

To my wife, *Khatereh Abron*, for her support and kindness.
To my dears, mother *Narges*, brother *Behzad*, and late *father* and *brothers*.
To my kind supervisors, *Professor Dr. Siti Mariyam Hj. Shamsuddin*
and *Dr.Alex Sim Tze Hiang*.
To my adviser and friend *Dr.Hassan Chizari* for his kind support.
And to all who supported me in my study.

ACKNOWLEDGEMENTS

In preparing this thesis, I was in contact with many researchers and academicians who have contributed towards my understanding and thoughts. In particular, I would like to thank to my internal supervisors, Professor Dr. Siti Mariyam Hj. Shamsuddin and Dr. Alex Sim Tze Hiang for their encouragement, guidance, critics and financial support. I would like to thank UTM Big Data Centre and Soft Computing Research Group (SCRG) for the inspiration in making this study a success, Prof.Samuel Madden and Dr.Yuan Mei from Massachusetts Institute of Technology for providing good comments on this work to carry out the experiments. Moreover, I would like to thank the authority of Universiti Teknologi Malaysia (UTM) for providing me with a good environment and facilities which I needed during the process.I would also like to thank the developers of the utmthesis LATEX project for making the thesis writing process easier for me, so I could focus on the content of the thesis, and not waste time with formatting issues. This work was partially supported by the Flagship project (RG QJ 130000.2428.02G38) and Applied Soft Computing- Big Data Research Centre.

Babak Behravesh, Kuala Lumpur

ABSTRACT

In various domains, applications are required to detect and react to complex situations accordingly. In response to the demand for matching receiving events to complex patterns, several event processing systems have been developed. However, there are just a few of them considered both performance and expressibility of event matching as focusing only on performance can cause negative effect on the expressibility or vice versa. This research develops a fast adaptive event matching system (FAEM), a new event matching system to improve expressibility and performance measures (throughput and end-to-end latency). This system is designed and developed based on a novel binary tree-based directed graph (BTDG) as a unified basis for event-matching. The proposed system transforms a user-defined query into a set of system objects including buffers, conditions on buffers, cursors, and join operators (non-kleene and kleene operators) and arranges these objects on a BTDG. Provided BTDG the enhancement in performance of non-kleene operators applied through developing a batch removal method to remove the events that are located out of time-window, and an actual time window (ATW) which can improve performance of event matching. To improve performance of kleene operators, this research introduces a twin algorithms for kleene operator which is match to BTDG. These two kleene algorithms apply grouping on events and reduce the number of intermediate results and apply combination algorithm in final stage. Transformation of queries containing join operators into BTDG enhances the expressibility of the proposed CEP system.

ABSTRAK

Dalam pelbagai domain, aplikasi diperlukan bagi mengesan dan bertindak sewajarnya terhadap situasi-situasi kompleks. Dalam memberi respons kepada permintaan bagi menyesuaikan peristiwa-peristiwa yang diterima pada bentuk-bentuk kompleks, beberapa sistem memproses peristiwa telah dibangunkan. Walau bagaimanapun, hanya beberapa sistem sahaja yang mempertimbangkan prestasi dan kebolehan penyataan penyesuaian peristiwa kerana dengan memberi fokus hanya ke atas prestasi yang boleh mendatangkan kesan negatif ke atas kebolehan penyataan atau sebaliknya. Kajian ini membangunkan sistem Penyesuaian Peristiwa Adaptif Segera (FAEM), satu sistem penyesuaian peristiwa baru bagi memperbaiki kebolehan penyataan dan langkah-langkah prestasi penghasilan dan kependaman hujung ke hujung. Sistem ini telah direka bentuk dan dibangunkan berdasarkan satu Graf Binari Terarah Berasaskan Pokok (BTDG) novel sebagai satu asas yang disatukan untuk penyesuaian peristiwa. Sistem yang dicadangkan mengubah sesuatu pertanyaan pengguna yang didefinisikan kepada satu set objek-objek sistem termasuk penimbal, keadaan-keadaan penimbal, kursor dan operator-operator gabungan iaitu operator *kleene* dan bukan *kleene* dan menyusun objek-objek tersebut di atas satu BTDG. Peningkatan prestasi operator-operator bukan *kleene* telah diperolehi melalui kaedah penyingkiran kelompok yang menyingkirkan peristiwa-peristiwa yang terletak di luar tingkap masa dan tingkap masa sebenar (ATW) yang meningkatkan prestasi penyesuaian peristiwa. Bagi memperbaiki prestasi operator-operator *kleene*, kajian ini memperkenalkan satu algoritma kembar bagi operator *kleene* yang sesuai dengan BTDG. Kedua-dua algoritma *kleene* mengaplikasikan pengumpulan ke atas peristiwa-peristiwa, mengurangkan jumlah keputusan pertengahan dan mengaplikasikan kombinasi algoritma pada tahap akhir. Transformasi pertanyaan-pertanyaan mengandungi gabungan operator ke dalam BTDG meningkatkan kebolehan penyataan sistem Pemprosesan Peristiwa Kompleks (CEP) yang dicadangkan. Eksperimen ke atas dua set data menunjukkan bahawa sistem yang dicadangkan telah mencapai prestasi yang lebih tinggi berbanding dengan yang lain. Di samping itu, kebolehan penyataan sistem tersebut telah dibandingkan dengan satu set sistem yang menunjukkan kemajuan signifikan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BTDG	-	Binary Tree-based Directed Graph
CEP	-	Complex Event Processing
DFA	-	Deterministic Finite Automaton
DSMS	-	Data Stream Management System
FAEM	-	Fast Adaptive Event Matching
FOPA	-	Fast Optimal Plan Algorithm
FSM	-	Finite State Machine
NFA	-	Non-deterministic Finite Automaton
Pub/ Sub	-	Publication/ Subscription
XML	-	Extensible Markup Language

LIST OF SYMBOLS

& - Conjunction operator

| - Disjunction operator

end_ts - End timestamp

e - Event

! - Negation operator

; - Sequence operator

start_ts - Start timestamp

ρ - Threshold in plan adaptation

t - Timestamp

CHAPTER 1

INTRODUCTION

1.1 An Overview on Complex Event Processing Systems

With rapid development of technology a new generation of communication system emerged that generates huge volumes of data.A few years ago computers and sensing devices were the sources of these data interactions, but because of their limited mobility, could not collect and analyze some useful data that people generates in their real daily life. Today our smart phones even our watches and glasses are offered in a packet size packages and generate data much more than any time in the past. These devices present interesting applications and services and can capture every mome3nt of our lives from the most profound to the most minor events including our location, purchases, friends, social interactions, likes and dislikes, and even our vital signs. This huge collection of data from everyone and everything can transform every industry on earth. For instance, Facebook with 1.3bn users collects more than 1 pedabyte of data a day. Processing this huge volume of data needs powerful analysis tools. In Big Data other than the huge volume of data, the velocity of data that is received in a high rate, the variety of homogeneous and/ or heterogeneous data sources that generate data demand a low latency system to notify interesting situations (opportunities or threats) with an acceptable veracity. Traffic in cities for instance, needs to deal with high volume of endless data and needs a fast event processing system to monitor and take timely actions (de Fabritiis *et al.*, 2008, Ding *et al.*, 2011, Sokha *et al.*, 2013); similar demands exists on detection suspicious event patterns in computer networks(Huang *et al.*, 2007) and event clouds (Suntinger, 2008, Widder *et al.*, 2007). In energy consumption, smart energy grids needs to efficiently manage energy consumption through semantic web using event processing systems (Ciuciu, 2012, Mauri, 2008, Simmhan and Kumbhare, 2011). In stock market timely notification on large number of fluctuations in stock price needs monitoring for proper reaction using event processing systems (Bülow *et al.*, 2014, Mangkorntong and Rabhi, 2007, Rozsnyai *et al.*, 2007). Crisis management systems needs to monitor environmental changes in form of a wide range of events and discover their correlation in order to take proactive timely decisions (Cugola and Margara, 2012, Itria *et al.*, 2014). In companies, manufacturing needs concurrent monitoring of several processes, and reaction can be taken adaptively. Internet banking systems need to monitor transactions and check the received requests with banks in a real-time manner to avoid fraud or money laundry utilizing event processing systems. Regarding the demand in different areas and the complexity of relationship between events, complex event processing (CEP) systems is considered as an important area of information systems(LeHong and Fenn, 2012).

In 2001, the term was coined by David Luckham and according to his definition (Luckham, 2001), event processing consists of set of tools and techniques to present solution for event based information systems. However, the history of developing today CEP systems returns to 1950's with developing discrete events simulations which has had been employed to predict weather and factory line production. In 1960's and 1970's every major company developed its own discreteevent simulator. Some of these systems are Simscript, GPSS and Simula67. In 1960's another event processing system was emerged from developing computer networks in ARPA project. Provided this system packets of data can be transmitted through a computer network, where each event contains a sequence of data. Standardization of hardware description languages happened in 1980's and 1990's by developing VHDL and Verilog as commercial discrete event simulators. Demands in real-time process was a cause to emerge active database technology in late the 1980's. These systems implemented on top of traditional database management systems and allowed defining simple patterns to trigger event condition action rules. Since placing event processing on top of traditional systems was not efficient due to the need for real-time process, developing a middleware between application layer and network layer to perform CEP task started in mid 80's by Tibco (Guide, 2010, Palmer, 2013). According to (Luckham, 2008) CEP from late 90's to 2005 was only applicable for simple situations, from 2004 to 2012 was creepy CEP era, and then from 2010 to 2025 it is going to become a ubiquitous technology in event driven world.

In a simplified CEP system, Figure1.1, user defines query and sends it to an event matching unit where events from various event sources are received to discover their matching over the given pattern. The pattern matching process is visualized and evaluated based on mainly two modeling systems. Finite state machine also known as FSM-based (Agrawal and Diao, 2008, Cugola and Margara, 2012, Demers *et al.*, 2007) and tree-based (Mei and Madden, 2009) are two event matching models for event processing which are widely used.

Figure 1.1 A simplified event processing system

Performance of pattern matching in CEP systems is measured based on throughput and end-to-end latency. In addition, the size of allocated memory is considered as an important factor because the process is performed in random access memory (RAM). Performance of event matching is influenced by changes in rate of events, distribution of events, cost of different event operators, selectivity of events, number of event classes in query, type of operators in query, and size of sliding time-window. The expressibility is measured through investigating model and algorithms to find if they can support a query. A query language that can support a wider range of operations is considered more expressible than others.

1.2 Problem Background

In order to detect an interesting pattern of changes, namely among stock market events, a user needs to define pattern of interest in form of a query to be run over stream of endless stock market events. Defining a query needs a language which is not only limited to define one query but the language should be expressible for a wide range of questions that might be asked by the user. The language to create event processing patterns called event processing language. Pattern-queries are presented with a similar structure to standard query language (SQL). SQL is used in relational database systems where data is mostly static, so the process is enhanced for disk access. In case of need to acquire data from database, we may issue a query. The problem for running a SQL query on stream of events arises when system needs to run the query as a new event detected, say 100 times in a second, while the application may run thousands of queries. Triggers in database systems can be employed to fire an event when an update happens, but triggers are rather slow to perform the operation for one hundred times in a second where each trigger needs to perform checking logical criteria and takes action. The process in CEP systems mostly performs in-memory (in random access memory) due to the need for low latency responses. To overcome the issue of keeping a query running on a non-stop stream of receiving events, another query language was required to deal with stream of events. Continuous query language (CQL) is an extension of SQL to deal with events, relationship between events, additional semantic and temporal constraints(Chakravarthy and Jiang, 2009). By the time of writing this thesis, there is no a unified continuous query language to define queries as SQL for static database systems. Although the structure of query in different research work is similar, their syntax are varied. This study is about to improve the performance and expressibility of event matching in CEP systems and does not propose a standard language for continuous query.

In CEP systems an expert user defines a query, that needs a rich query language to support queries for various demands. The flexibility of a model and join algorithms of event processing systems has a great impact on performance of a CEP system. In addition, the expressibility of a query language depends on defining join algorithms. The modeling system, of CEP systems were mainly implemented on petri-net, tree, and finite state machine based structures. Due to the problems in verification of decidability and complexity of Petri net (Esparza, 1998, Mei and Madden, 2009), Petri net modeling system is not widely used in CEP applications. The most common structures in CEP systems are finite state machine (FSM) and tree-based structures. In FSM-based systems, matching is performed from either beginning or end of pattern section regarding to limitation of FSM based systems. Non-deterministic finite automaton-based (NFA) as a derivation FSM-based structure borrows many of its characteristics from FSM-based systems, however, in (Agrawal and Diao, 2008, Diao *et al.*, 2007) they defined some strategies to ignore some events non-deterministically to improve the system's responsiveness. Tree-based systems(Chakravarthy *et al.*, 1994, Mei and Madden, 2009) shows more degree of flexibility in reordering joins because unlike the FSM-based models, they are not limited to left or right-deep joins. T-rex as an FSM-based systems presented an expressible language TESLA and performs matching in left-deep order (Cugola and Margara, 2012). In tree-based models, fine granularity helped to improve performance measures through reordering joins regarding the load changes. For instance, ZSTREAM (Mei and Madden, 2009) performs reordering joins (plan adaptation) when the rate of input events shows a significant change. Then, the system switches to an efficient plan to reduce memory allocation and improves throughput.The limitation of finite state machine based structure are mostly on plan adaptation. These systems can either perform matching in left to right (Cugola and Margara, 2012), or right to left (Agrawal and Diao, 2008, Wu *et al.*, 2006) method. This research considers the limitation of FSM and develops a version of tree-based structure which is not only limited to left and right plan.The promissing system, has a more degree of freedom to switch to other plans.

In order to improve performance and memory allocation for the process, these systems (Mei and Madden, 2009, Wu *et al.*, 2006) also wait for a final event to start event matching. This may later have a converse effect on the process when there are many events waiting for evaluation and the final event is received late. The problem can be even worse when the size of time window is defined rather extended, rate of selectivity is low and a long queue of events are waiting for event matching.The large, extended, time window covers more number of events that are subjected for evaluation. In addition, the process is performed like breadth first algorithm so, it needs much time to produce the final match cases. This strategy in some scenarios causes a large lag time. On the other hand, the strategy to wait for final event instance that are seen in (Agrawal and Diao, 2008, Mei and Madden, 2009) may not be efficient in proactive decision making, where the system needs to notify a potential threat or opportunity before arriving the final event.

In event processing, events have a life cycle based on time-window of a pattern query. Earliest allowed time of considering an event is calculated by subtraction of now, current time, from the length of time window. So, events with the time stamp less than earliest allowed time are discarded. Provided information in above, ZSTREAM(Mei and Madden, 2009) starts event matching as arriving a new instance of final event, say *e_f*. The problem arises when many event instances are stashed in the system while still waiting for *e_f*. As soon as arriving the *e_f* the process starts where many comparisons between events are required and events are composed if they meet conditions. These processes take time so that some events that were valid by the time of detecting *e_f* are now invalid. In this case ZSTREAM results some false negatives.

During event matching, irrelevant events in both systems are removed when the criteria in time window constraint is not met. They check events from beginning to the end to find invalid events and delete them from their respective buffers one by one(Agrawal and Diao, 2008, Demers *et al.*, 2007, Mei and Madden, 2009). This process can be performed in a rather efficient way.

Aggregation is performed based on grouping events. The aggregation is performed based on Kleene operator. Kleene can be either (+, *, num), where + represents one or more instances, * zero or more instances, and num is for a certain number of event instances. In ZSTREAM (Mei and Madden, 2009), two simple event classes confine a kleene event class, by put it in the middle, to make limitation on it. Then grouping is applied on the middle when criteria from left and right is applied. After the process a composite event is generated. The problem arise when another event class is joint with the kleene ternary and invalidates one of the kleene event items in composite event. In this case, if condition is not satisfied the composite event is remove. This create false negatives for Kleene operation. This point is extensively described in chapter 2.

Similarly, in SASE+ grouping is applied based on grouping on kleene buffer, while the condition from previous events in are not taken into account. In addition, in(Agrawal and Diao, 2008) through sharing results which can reduce memory allocation and CPU time because it prevents repetitive event matching task for other event instances in the same computation state no matter the history of event. However, this point can be violated because criteria on data values of events may cause failure as the process continues. The anti-thesis on the issue is presented in chapter 2. Sharing results based on a single hierarchical is a good solution to reduce burden of event matching as addressed in(Stakhanova *et al.*, 2009), while in contrast to this study, their method did not address how to manage time-window, and how to use the structure for prediction of next possible event.

Recently, many works tried to develop distributed CEP systems (Cao *et al.*, 2013, Ottenwälder and Koldehofe, 2014) on their proposed distributed environment but they used the existing single query systems as their basis. However, this study covers some critical gaps and then solutions to rectify shortcomings on single node systems. Enhancing the process on single node CEP systems can improve the process on many of these distributed systems. In this study, we scrutinize the process on existing FSM and tree-based CEP systems and enhance the process so that the enhanced system can be employed for high performance event matching; then contribute to high performance distributed CEP systems.

One of the positive points of ZSTREAM that highlights this work from the other CEP systems is the ability of ZSTREAM to reorder the priority of applying join operators based on a cost model. Reordering of join operators based on cost of operators and population of events, help their system to adapt with the changes and be able to switch to an efficient order of applying join operators (also called plan adaptation). If the cost of event matching passes a certain threshold, ZSTREAM attempts to calcualte cost of alternative plans and switch to a plan with the minimal cost. The time complexity of plan adaptation algorithm in ZSTREAM is addressed as $O(n^3)$. Their algorithm creates a powerset of event classes with timecomplexity of $O(n^2)$ and then in each round the algorithm performs a linear search to find the pair of event classes with the minimal cost. They addressed the overall time complexity of their plan adaptation algorithm is $O(n^3)$.

Two factors that influence the performance of event matching are, throughput, and end-to end latency. Throughput considers the number of events which are processed in a second. End-to-end latency considers the duration (in seconds) between the time stamp of receiving event to the system and finalizing its process. Various event matching systems show a different end-to-end latencies for identical set of events. This research delves into this difference in end-to-end latency and discovers stems from the both structure and join algorithms. The majority of the previous research work tried to focus on throughput (Agrawal and Diao, 2008, Cugola and Margara, 2012, Mei and Madden, 2009). In contrast with previous research work on CEP systems, this research highlights the importance of both throughput and end-to-end latency. This research presents a directed graph on the structure to direct events to do matching with a minimal delay which this functionality is supported by the join operator algorithms.

The expressibility of the event matching determines the range of queries a query language supports. The limitations in defining queries is because of adjacency of join operators with different natures in a single query. For example, A in Sequence join with B and B is in a conjunction join with C. This research discovers the origin of this limit, and overcomes the issue by developing a new set of join operators. This research separates operators as Non-kleene algorithms that are binary operators while kleene operators are unary operators. Previous research work support binary operators without covering the expressibility of the queries, however, this research highlights the shortcoming of their system (Mei and Madden, 2009) in supporting both kleene and non-kleene operators on a query with respect to expressibility as well as performance. This research presents more than 10 queries that are not supported by previous studies due to the limitation in expressibility.

The evaluation of event matching mostly concentrated on enhancement in order to improve performance of pattern matching, and expressibility of the query language to cover a wide range of queries.

1.3 Problem Statement

Regarding the limitations of the previous studies on events processing that addressed in problem background, current study focuses on gaps related to performance measures including throughput, end-to-end latency and size of allocated memory through developing enhanced algorithms on a flexible and robust structure, and expressibility of join algorithms to support various queries.Regarding these two measures problem statement is presented as:

a) *How to design a unified structure to overcome the performance and expressibility shortcomings of previous systems?*
b) *How to enhance join operator algorithms in accordance to the unified structure in order to improve performance and expressibility of event matching?*

1.4 Aim of the Reasearch

The aim of the research is to develop a fast adaptive event matching system (FAEM) through developing a new model, adaptive reordering joins and enhancing join algorithms to improve performance and expressibility measures of event matching.

Although this study focuses on enhancement of event matching, this component needs to be in harmony with other components that this research plans to develop in the future. Figure 1.2 shows our proposed event processing systems. An expert user defines queries and send them to enhancement unit. At the beginning there is no prior knowledge about history of event matching. The end user runs the query while events are received from event sources, and matching is performed in event matching unit. Some information about event matching including number of partial match cases and full match cases, the number of comparisons, and other statistical information about event matching are sent to history of matching for later use. The other result of event matching is a notification message in case of finding a match case. Provided, some information on history of matching, this information is consumed by prediction unit. The prediction results then, are sent to pattern-query enhancement (enhancement) unit to enhance pattern matching process (e.g., reordering join operators). In addition, the prediction results are used for taking proactive actions.Proactive decision making in CEP systems is performed based on current state of events. Given the current events, the CEP system predicts the future event(s). Prediction of next events helps the system to be proactive in deals with a threat or opportunity. For example, to predict an attack to a computer network based on logs in a real-time manner. However, the focus of this research is on event matching and the research leaves the proactive decisions as a future work.

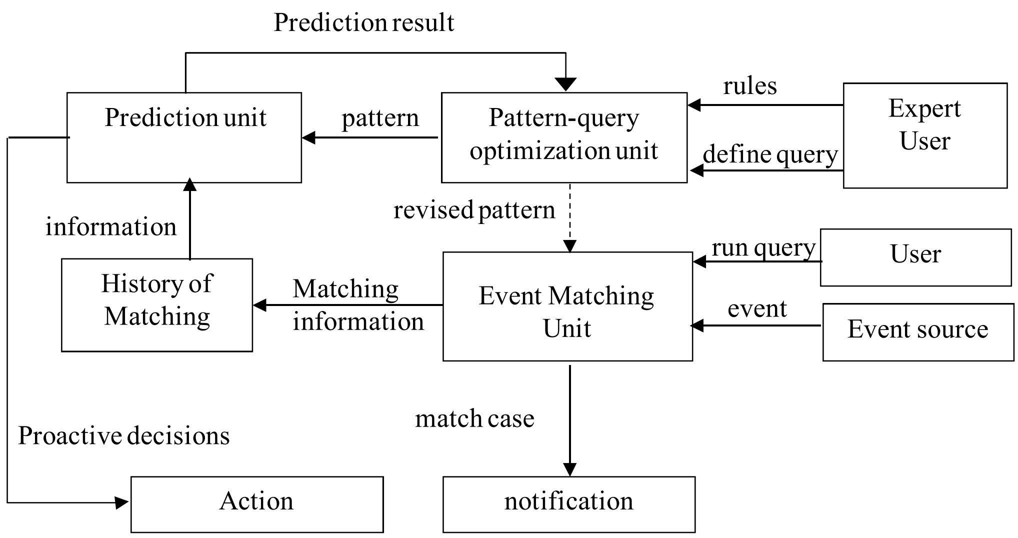

Figure 1.2 The proposed framework for event processing system

The process starts when the expert user defines a query and send it to the system to check its validity as it is shown in Figure 1.3. If the query is valid then system transforms the query into a set of system objects (event buffers and cursors) to form a logical tree-based structure. However, the arrangement of the buffers and pointers is decided based on a cost operaotors and population of events in each buffer. After the construction of event matching plan, events are received from event sources.The system checks the events against the condition of each leaf buffer; if an event is match to a buffer then the event is inserted into the buffer and follows the cursor to check its sibling buffer to evaluate. If a pair of event fulfill the criteria then they are assembled and make a composite event. The composite event is inserted into their parent buffer. Then the evaluation continues with the parent buffer and its sibling.

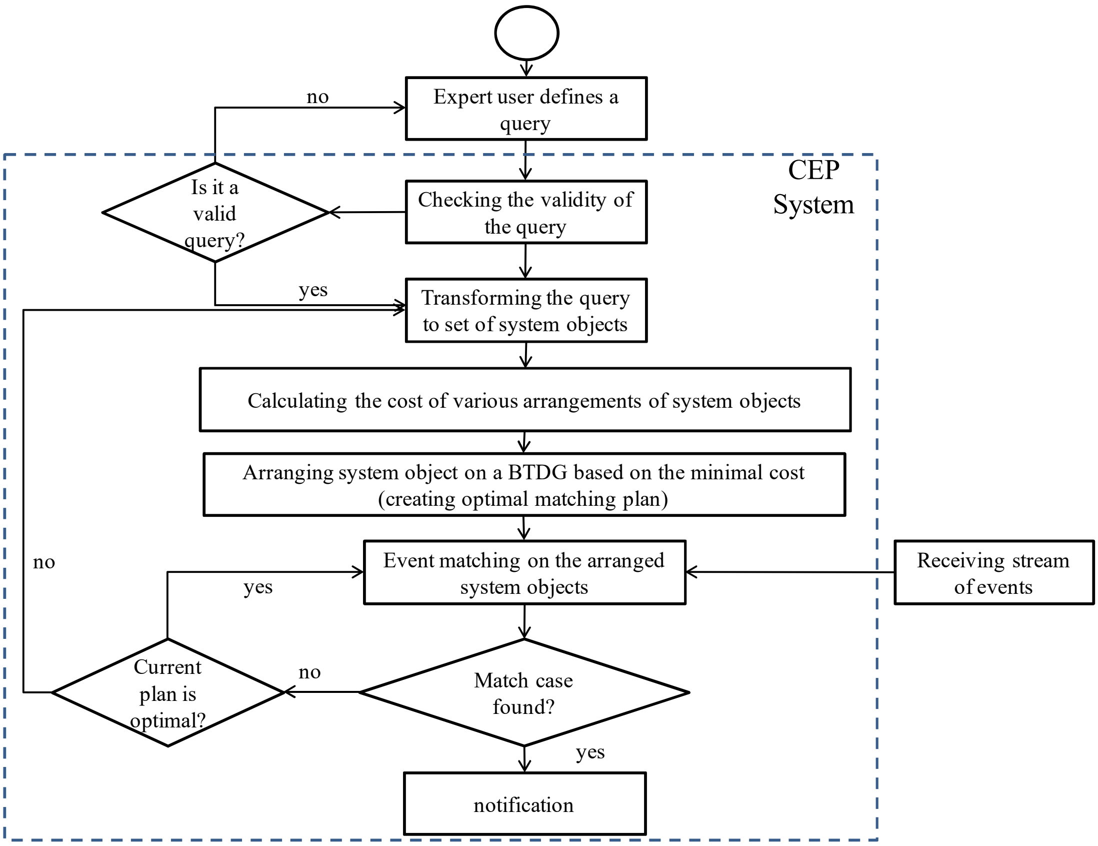

Figure 1.3 The flowchart of the proposed system

Finally, if a composite event reached to the root of the tree, the system generates a notification. However, the system does not stop to receive and evalute events while stream of events enter to the system. The system also check if the plan is optimal based on a cost model. Switching to a plan with a lower cost improves the performance according to (Mei and Madden, 2009). In this case system attempt to reconstruct the matching plan.

1.5 Objectives of the Research

The main goal of the proposed framework by this study is to develop an expressible high performance event matching system. Therefore, according to background and problem statement, three main objectives of this research can be expressed as follows:

a) To develop a binary tree-based directed graph (BTDG) as a basis to define join operators' algorithms and to event matching and To creates a unified binary tree-based directed graph to transform user-defined query into BTDG. In BTDG event buffers are linked by join operators and criteria are placed on their respective buffers in order to improve performance and expressibility measures.

   - To design enhanced non-kleene join operator algorithms including sequence, negation, conjunction, and disjunction that improves throughput and end-to-end latency of the proposed CEP system.
   - To develop batch removal of out-of-range of time window events that helps to reduce number of comparisons, memory consumption, and excessive shifting of events in buffers.
   - To implement a time-window policy (actual time-window) on join algorithms to reduce latency gap which can improve performance of event matching process. This can be achieved through applying directed graph and modifying the join operators' algorithms.
   - To construct a directed graph to reduce latency gap so that when a new event comes event matching starts which helps reducing latency gap.

b) To construct a new set of algorithms for kleene operators based on BTDG to improve expressibility of defining queries, flexibility of reordering joins, and improves the performance of kleene num to reduce excessive intermediate results.

- To develop two algorithms to support kleene operators 1) when the left event class is a kleene and the right event class is a non-kleene, and 2) when the left event class is a non-kleene and the right event class is a kleene. These twin algorithms improves expressibility of kleene operator in dealing with non-kleene operators. In addition, reordering joins can be performed in a good degree of flexibility.
- To propose a new combination algorithm for *kneene num* which is applied on top of hierarchy. The algorithm uses results of previous rounds and can improve performance of event matching.

c) To develop a fast optimal plan adaptation algorithm (FOPA) to reconstruct BTDG. This algorithm dynamically constructs and switches to the plan with a minimal cost through reordering join operators.

- To develop fast optimal plan algorithm to switch to a query plan with a minimal cost to improve throughput of event matching; The proposed method adapts with respect to the cost model regarding, 1) population of events and, 2) cost of join operators.
- To discover for the first time a property in join operators which influences results of event matching, associativity. In addition, an algorithm is designed to avoid generating inaccurate composite events as a result of considering this property.
- To design plan adaptation algorithm in a low time complexity as $O(n^2)$ which performs better than its counterparts.

1.6 Scope of the Study

This study concentrates on enhancing event matching in event processing systems by increasing expressibility of the system and enhancing performance of pattern matching. The scopes are as follows:

a) The proposed system is tested on two data sets, Yahoo! music in KDD Cup 2011(Dror *et al.*, 2012)as a real-world dataset with more than 1,000,000 tuples and a synthesized stock market dataset.
b) Join operator algorithms including sequence, negation, conjunction, disjunction and kleene (*, +, and num) operators. In addition, event matching strategies at various levels of relaxation in accepting or rejecting events are covered.
c) Expressibility of query languages and their underlying algorithms are investigated. The expressibility factors that are considered in this research are, 1) adjacency of different types of join operators in a query, and 2) weather a query can support various sizes of event classes in its pattern.
d) Performance factors are considered that may cause 1) CPU-related (throughput and end-to-end latency) and 2) memory-related (size of allocated-memory) effects.
e) Some of the efficient techniques that were used ZSTREAM (Mei and Madden, 2009), SASE+ (Agrawal and Diao, 2008) and T-Rex (Cugola and Margara, 2010) for inquery transformation are investigated extensively.
f) Plan adaptation in reordering join operators in CEP systems is included.

In this research, we concentrate on enhancing event matching process on a continuous query which can be affected by several factors including the rate of receiving events, type of events, selectivity of events, number of match and mismatch cases, size of sliding time window, cost of join operators, and adaptive behaviour of event matching. In existing work a set of modelling systems isemployed to compare and visualize pattern matching processes. In addition, the expressibility of different CEP systems is covered in details.

1.7 Significance of Study

Regarding the demand for employing a high performance event processing systems in a wide range of applications, it is useful to enhance event matching process. On the other hand, improving the expressibility of queries can help to define wide range of queries.

The solution proposed in this study is in very bottom line of CEP system where many applications can employ it. Some of these applications are, traffic estimation and prediction in floating cars, decision making in stock markets, reducing catastrophe in critical management, traffic management in communication network, and real-time healthcare services.

1.8 Organization of the Thesis

In this thesis a new CEP system is developed which concentrates on enhancing expressibility and performance. The rest of this dissertation is organized as follows.

Chapter 2, at first presents the definitions on event processing systems. Then we present definitions and applications of CEP systems and investigate the interactions between CEP components in existing CEP systems. Then, with a closer look, we investigate the methods proposed in each area through similarities and differences of techniques, to discover each method shortcomings and advantages. These positive and negative points on existing systems helps us to propose our method.

Chapter 3 covers the methods that is applied on this research. The method stems from scrutinized investigation on the literature. The steps of this research works are presented in few diagrams followed by detailed descriptions which includes the model to develop event matching, the algorithms to develop non-kleene and kleene join operators and plan adaptation algorithm. This chapter finally describes datasets and evaluation methods which are used in this research.

Chapter 4 presents the proposed structure to manage process through transformation of user-defined queries into binary tree-based directed graph. This structure is used as a basis to perform event marching for all of the join operator including kleene and non-kleene operators. This chapter also covers developing non-kleene operators including four enhanced algorithms for sequence, negation, conjunction and disjunction as event operators (join operators). The enhancement focuses on management of event buffers and sliding time window. It is shown that the enhancement on each operator improves performance of event matching.

Chapter 5 covers three kleene operators including kleene plus, star and num. We first discover the some shortcomings of kleene operators in FSM-based and tree-based systems which have not been addressed previously. Then, the steps for developing the algorithms are presented for kleene plus, star, and num. These algorithms are matched with the requirements of the proposed BTDG that is presented chapter 4. In addition, for kleene num a new combination algorithm is developed that tries to reuse results of previous combinations so that it can helps to improve throughput while maintains the accuracy of the process.

Chapter 6 develops a fast optimal plan algorithm (FOPA) to construct BTDG in a low time-complexity. FOPA also differentiates various joins in construction of BTDG to avoid generating undesired composite events.

Chapter 7 summarizes the thesis, restates the contributions, and suggests direction of future research.

REFERENCES

Adi, A. and O. Etzion. (2004). Amit – the situation manager. *VLDB Journal*. **13**: 177-203.

Aggarwal, C.C., J. Han, J. Wang, and P.S. Yu. (2004). A framework for projected clustering of high dimensional data streams. *Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment: 31 August-3 September. Toronto, Canada. 852-863.

Agrawal, J. and Y. Diao. (2008). Efficient pattern matching over event streams. *In Proceedings of ACM SIGMOD international conference on Management of data*. Vancouver, BC, Canada: ACM. 2008. 147-160.

Arasu, A., S. Babu, and J. Widom. (2006). *The CQL Continuous Query Language : Semantic Foundations and Query Execution ∗. VLDB Journal*. 121-142.

Babcock, B., S. Babu, and M. Datar. (2002). *Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM. 1- 16.

Babu, S. and J. Widom. (2001). Continuous queries over data streams. *ACM Sigmod Record*. **30**: 109-120.

Bai, Y. (2007). Data stream processing and query optimization techniques. *ProQuest*.

Barga, R. and H. Caituiro-Monge. (2006). Event correlation and pattern detection in CEDR. *Current Trends in Database Technology–EDBT*. 2006: 919-930.

Brenna, L., A. Demers, and J. Gehrke. (2007). Cayuga: a high-performance event processing engine. *Proceedings of ACM SIGMOD international conference on Management of data*. 1100-1102.

Bülow, S., M. Backmann, and N. Herzberg. (2014). Monitoring of Business Processes with Complex Event Processing. *Business Process Management Workshops*. 277-290.

Cao, K., R. Li, and F. Wang. (2013). The Research on CEP Based on Query Rewriting for CPS. *Internationa journal of modelling and optimization.* **3**(10): 509-514.

Cao, Z., Y. Diao, and P. Shenoy. (2009). Architectural considerations for distributed RFID tracking and monitoring. *In Proceedings of the 5th ACM Workshop on Networking Meets Databases (NetDB)*. ACM: 14 October, Montana,USA.

Carmona, J., J. Cortadella, and M. Kishinevsky. (2008). A region-based algorithm for discovering Petri nets from event logs. *Business Process Management.* 358-373.

Chakravarthy, S., E. Anwar, L. Maugis, and D. Mishra. (1994). Design of sentinel: An object-oriented DBMS with event-based rules. *Information and Software Technology*. **36**(9): 555-568.

Chakravarthy, S. and Q.C. Jiang. (2009). Stream data processing: a quality of service perspective: modeling, scheduling, load shedding, and complex event processing. *Processing.*

Chakravarthy, S., V. Krishnaprasad, E. Anwar, and S.K. Kim. (1994). Composite events for active databases: Semantics, contexts and detection. *Proceedings of the International Conference on Very Large Data Bases*: Santiago, Chile. 606-617.

Chandrasekaran, S. and M.J. Franklin. (2003). PSoup: a system for streaming queries over streaming data. *The VLDB Journal The International Journal on Very Large Data Bases*. **12**: 140-156.

Chen, J., D.J. DeWitt, F. Tian, and Y. Wang. (2000). NiagaraCQ: A scalable continuous query system for internet databases. *ACM SIGMOD Record*. **29**: 379-390.

Chhillar, R.S. and B. Kochar. (2009). A New Efficient Approach for Effective Warehousing of RFID Data: Readers Load Sentient Scheme. *American Journal of Scientific Research.* **3**: 85-95.

Ciuciu, I.G., Robert Meersman, haram Dillon. (2012). Social network of smart-metered homes and SMEs for grid-based renewable energy exchange. in *International Conference on Digital Ecosystems Technologies (DEST)*. 1-6. IEEE

Cugola, G. and A. Margara. (2010). TESLA: a formally defined event specification language. *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*. 50-61. ACM.

Cugola, G. and A. Margara. (2012). Complex event processing with T-REX. *The Journal of Systems & Software*. **85**: 1709-1728.

Cugola, G. and A. Margara. (2012). Processing flows of information: From data stream to complex event processing. **V**: 1-70. ACM

Dayal, U., B. Blaustein, A. BUchmann, and U. Chakravarthy. (1988). The HiPAC Project: Combining Active Databases and Timing Constraints*. *In ACM SIGMOD*. 51-70.

de Fabritiis, C., R. Ragona, and G. Valenti. (2008). Traffic Estimation And Prediction Based On Real Time Floating Car Data. *International IEEE Conference on Intelligent Transportation Systems*. Ieee: Beijing. 197-203.

Demers, A., J. Gehrke, B. Panda, M. Riedewald, V. Sharma, W.M. White, and Others. (2007). Cayuga: A general purpose event monitoring system. *Third Biennial Conference on Innovative Data Systems Research*. CIDR: 7-10 January. Asilomar, CA, USA. 412-422.

Diao, Y., N. Immerman, D.G. (2007). Sase+, and An. agile language for kleene closure over event streams. *In UMass, Technical Report*. **7**: 1-14.

Ding, J.-X., H.-J. Huang, and Q. Tian. (2011). A traffic flow cellular automaton model to considering drivers' learning and forgetting behaviour. *Chinese Physics B*. **20**. 2. 028901.

Dong, L., D. Wang, and H. Sheng. (2006). Design of RFID Middleware Based on Complex Event Processing. *2006 IEEE Conference on Cybernetics and Intelligent Systems*. 1-6.

Drogba, G., N. Koenigstein, Y. Koren, and M. Weimer. (2011). The Yahoo! Music Dataset and KDD-Cup'11. in *KDD Cup*. 8-18.

Dror, G., N. Koenigstein, Y. Koren, and M. Weimer. (2012). Pattern recognition on Yahoo! Music Dataset. *Proceedings of KDD Cup 2011 competition*. 3-18.

Esparza, J. (1998). Decidability and complexity of Petri net problems—an introduction. *Lectures on Petri Nets I: Basic Models*. 374-428.

Etzion, O. and P. Niblett. (2010). Event Processing in Action. Processing XML Streams with Deterministic Automata. Manning Publication Co.

Fischer, P., A. Garg, and K. Sheykh Esmaili. (2010). Extending XQuery with a Pattern Matching Facility. *Database and XML Technologies*. 48-57. Springer: Berlin Heidelberg.

Fodor, P., D. Anicic, and S. Rudolph. (2011). Results on out-of-order event processing. *Practical Aspects of Declarative Languages*. 220-234. Springer: Berlin Heidelberg

Forgy, C.L. (1982). Rete : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*. *Artificial intelligence.*. **19**: 17-37.

Fu, T.-c., F.-l. Chung, R. Luk, and C.-m. Ng. (2007). Stock time series pattern matching: Template-based vs . rule-based approaches. *Engineering Applications of Artificial Intelligence*. **20**: 347-364.

Fülöp, L., G. Tóth, R. Rácz, and J. Pánczél. (2010). Survey on complex event processing and predictive analytics. *In Proceedings of the Fifth Balkan Conference in Informatics*. 26-31.

Gatziu, S., A. Geppert, and K.R. Dittrich. (1995). The SAMOS Active DBMS Prototype. *In SIGMOD conference.* 480-481.

Guido, A.S. (2010). Tibco Business Events Extreme Application Architect's Guide. *Tibco technical report*.

Gyllstrom, D., and J. Agrawal. (2008). On supporting kleene closure over event streams. *In International Conference in Data Engineering (ICDE).* 10- 24.

Gyllstrom, D., Y. Diao, E. Wu, P. Stahlberg, and G. Anderson. (2007). SASE : Complex Event Processing over Streams. *The Third Biennial Conference on Innovative Data Systems Research (CIDR)*. ACM: Asilomar, California, USA. 407-411.

Hassan, M.M., B. Song, and E.-N. Huh. (2009). A dynamic and fast event matching algorithm for a content-based publish/subscribe information dissemination system in Sensor-Grid. *The Journal of Supercomputing*. 330-365.

Hirte, S., Andreas Seifert, Stephan Baumann, Daniel Klan, K-U. Sattler. (2012). Data3-a kinect interface for olap using complex event processing. *In International Conference on Data Engineering (ICDE),*. 1297-1300. IEEE

Hoßbach, B., 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*. 149-154. ACM.

Huang, W., W. Tang, and C.F. Beedgen. (2007). Storing log data efficiently while supporting querying to assist in computer network security. *US Patent App. 11/966,078*.

Itria, M.L., A. Daidone, and A. Ceccarelli. (2014). A Complex Event Processing Approach for Crisis-Management Systems. *In Distributed Event Based Systems* (*DEBS)*. ACM Press: New York. 238-249.

J. Xie, J.Y., Y. Chen. (2008). A sampling-based approach to information recovery. *Internationa Conference in Data Engineering (ICDE)*. 476- 485.

Jegadish. N, and Gehani, N.H. (1991). Ode as an active database: Constraints and triggers. *In VLDB*. 327-336.

Jean-Claude Mamou, L.S.. (2004). *Real time data integration services for health care information data integration.* US20050228808 A1.

Jerzak, Z. and C. Fetzer. (2007). Prefix forwarding for publish/subscribe. *Proceedings of inaugural international conference on Distributed event-based systems - DEBS '07*. ACM Press: New York, New York, USA. 238-249.

Jin, C., J. Carbonell, and P. Hayes. (2004). ARGUS: Rete+ DBMS= Efficient Continuous Profile Matching on Large-Volume Data Streams. *Foundations of Intelligent Systems*. 142-151. Springer: Berlin Heidelberg.

K. Gatziu, S.D. (1994). Events in an active object-oriented database.

Krishnaprasad, V., S. Chakravarthy, and S. Kim. (1994). *Composite Events for Active Databases: Contexts and Detection Semantics, Contexts and Detection. Proceedings of the 20th VLDB Conference, Santiago, Chile*: Santiago, Chile. 606-617.

Lee, W.B., B.C.F. Cheung, and S.K. Kwok. (2009). Digital Manufacturing and RFID-Based Automation. *Springer Handbook of Automation.* 859-879.

LeHong, H. and J. Fenn. *Key Trends to Watch in Gartner 2012 Emerging Technologies Hype Cycle*. from: http://www.forbes.com/sites/gartnergroup/2012/09/18/key-trends-to-watch-in-gartner-2012-emerging-technologies-hype-cycle-2/#67c5d9207f7c. 2012

Lemans, S.J.J., D. Fahland, and W.M.P. van der Aalst. (2014). Discovering block-structured process models from event logs containing infrequent behaviour. *In Business Process Management Workshops*. 66-78. Springer International Publishing

Li, C. and R. Berry. (2014). CEPBen: A Benchmark for Complex Event Processing Systems. *Performance Characterization and Benchmarking.* 125-142.

Liu, J., Q. Wu, and W. Liu. (2009). Temporal Restriction Query Optimization for Event Stream Processing. *Advances in Web and Network Technologies, and Information Management.* 25-35.

Luckham, D. (2008). A Short History of Complex Event Processing 1 Part 2 : the rise of CEP. *Technical Report*. 1-8.

Luckham, D.C. (2001). The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Professional.

Mangkorntong, P. and F.A. Rabhi. (2007). A Domain-Driven Approach For Detecting Event Patterns in E-Markets: A Case Study in Financial Market Surveillance. *Web Information Systems Engineering–WISE.* 147-158.

Mauri, G., Diana Moneta, Paolo Gramatica. (2008). *Automation systems to support smart energy behaviour of small customers. In SmartGrids for Distribution, CIRED Seminar*. pp: 1-4.

Mei, Y. and S. Madden. Zstream: a cost-based query processor for adaptively detecting composite events. (2009). *Proceedings of the 35th SIGMOD international conference on Management of data*2009. Rhode Island, USA. 193-206. ACM.

Mendes, M.R.N., P. Bizarro, and P. Marques. (2009). A Performance Study of Event Processing Systems. *Performance Evaluation and Benchmarking*. pp: 221-236. Springer: Berlin Heidelberg.

Motwani, R., J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. Manku, C. Olston, J. Rosenstein, and R. Varma. (2003). Query processing, approximation, and resource management in a data stream management system. *Proc. First Biennial Conf. on Innovative Data Systems Research (CIDR).* 1-12.

Muthusamy, V., H. Liu, and H.-a. Jacobsen. (2010). Predictive publish/subscribe matching. *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*. ACM: Cambridge, UK. 14-25.

Nielsen, S., C. Chambers, and J. Farr. Systems and methods for complex event processing of vehicle information and image information relating to a vehicle. (2013). *US Patent*. **243547564**.

Ottenwälder, B. and B. Koldehofe. (2014). RECEP: selection-based reuse for distributed complex event processing. *In 8th ACM International Conference on Distributed Event-Based Systems(DEBS)*: Mumbai, India. 59-70.

Owens, T.J. Survey of event processing. (2007). *Journal of Air Force Research*. **8**(2): 24-33.

Palmer, M. (2013). Big Data Streaming Analytics and the New Physics of Fast Data. Tibco. from: http://www.tibco.com/blog/2014/07/22/big-data-streaming-analytics-and-the-new-physics-of-fast-data/.

Park, H.K. and W.S. Lee. (2008). Multiple continuous queries evaluation over data streams. *Proceedings of the 8th conference on Applied computer scince. World Scientific and Engineering Academy and Society (WSEAS)*. Venice, Italy. 346-350.

Peng, S., Z. Li, Q. Li, Q. Chen, H. Liu, Y. Nie, and W. Pan. (2010). Efficient Multiple Objects-Oriented Event Detection Over RFID Data Streams. *Web-Age Information Management.* 97-102.

Pongthawornkamol, T., Klara Nahrstedt, Guijun Wang. (2010). Probabilistic QoS modeling for reliability/timeliness prediction in distributed content-based publish/subscribe systems over best-effort networks. *In Proceedings of the 7th international conference on Autonomic computing,* pp. 185- 194. *ACM.*

Qin, X. and W. Lee. (2005). Attack plan recognition and prediction using causal networks. *In 20th Annual Conference in Computer Security Applications.* Tucson, US. 370-379.

Rao, J., S. Doraiswamy, H. Thakkar, and L.S. Colby. (2006). A deferred cleansing method for RFID data analytics. *Proceedings of the 32nd international conference on Very large data bases.* 175- 186.

Ré, C., J. Letchner, M. Balazinksa, and D. Suciu. (2008). Event queries on correlated probabilistic streams. *In proceedings of ACM SIGMOD international conference on Management of data*. Vancouver, BC, Canada. 715-728. ACM

Rizou, S., F. Durr, and K. Rothermel. (2011), Fulfilling end-to-end latency constraints in large-scale streaming environments. *In 30th International conference on Performance Computing and Communications Conference (IPCCC)* Orlando. 1-8.

Rosenblum, D.S. and A.L. Wolf. (1997). A design framework for Internet-scale event observation and notification. *ACM SIGSOFT Software Engineering Notes.* **22**(6): 344-360.

Rozsnyai, S., R. Vecera, J. Schiefer, and A. Schatten. (2007). Event Cloud - Searching for Correlated Business Events. *The 9th IEEE International Conference on E-Commerce Technology, E-Commerce, and E-Services*. 409-420.

Schmidt, K.U., R. Stuhmer, and L. Stojanovic. (2008). Blending Complex Event Processing with the RETE Algorithm. in *1st International workshop on Complex Event Processing for the Future Internet colocated with the Future Internet Symposium FIS2008*. 1-10. ACM

Schultz-Møller, N.P., M. Migliavacca, and P. Pietzuch. (2009). Distributed complex event processing with query rewriting. in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems - DEBS '09*. Nashville, TN, USA: ACM Press. 4-5.

Shen, Z. and S. Tirthapura. (2006). Faster Event Forwarding in a Content-Based Publish-Subscribe System through Lookup ReuseEvent. *Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)*. 77-84. IEEE

Simmhan, Y. and A. Kumbhare. (2011). An analysis of security and privacy issues in smart grid software architectures on clouds. *IEEE International Conference in Cloud Computing (CLOUD), 2011* 4- 9 July. 582- 589. IEEE

Sokha, Y., K. Jeong, J. Lee, and W. Joe. (2013). A Complex Event Processing System Approach to Real Time Road Traffic Event Detection. *Journal of Convergence Information Technology(JCIT)*. **8**: 142-148.

Srivastava, U. and J. Widom. (2004). Flexible time management in data stream systems. in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '04*. New York, New York, USA: ACM Press. 263- 273.

Stakhanova, N., A. Ghorbani, and W. Bird. (2009). Graph Structures for event matching. *US Patent*: US.

Stipkovic, S., Ralf Bruns, Jurgen Dunkel. (2013). Pervasive Computing by Mobile Complex Event Processing. *IEEE 10th International Conference in e-Business Engineering (ICEBE)*. 318-323. IEEE.

Stojanovic, N. D., and L. Stojanovic. (2013). Tutorial: personal big data management in the cyber-physical systems-the role of event processing. *In Proceedings of the 7th ACM international conference on Distributed event-based systems*. Mumbai, India: ACM. 281-288.

Stojanovic, N., Yongchun Xu, Aleksandar Stojadinovic, Ljiljana Stojanovic. (2014). Using mobile-based complex event processing to realize collaborative remote person monitoring. *In Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. Oslo, Norway: ACM. 225-235.

Stonebraker, M., U. Çetintemel, and S. Zdonik. (2005). The 8 requirements of real-time stream processing. *ACM SIGMOD Record*. **34**(4): 42-47.

Suntinger, M., Hannes Obweger, Josef Schiefer, E. Groller. (2008). The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. *In Visualization Symposium, IEEE Pacific*. Kyoto, Japan. 111- 118. IEEE

Tatbul, N. and S. Zdonik. (2006). Window-aware load shedding for aggregation queries over data streams. *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment: Seol, Korea. 799-810.

Tran, T., C. Sutton, R. Cocci, Y. Nie, Y. Diao, and P. Shenoy. Probabilistic inference over rfid streams in mobile environments. (2009). *In ICDE '09: Proceedings of the 2009 IEEE International Conference on Data Engineering*. Shanghai, China. 1096-1107. IEEE

Turchin, Y., A. Gal, and S. Wasserkrug. (2009). Tuning complex event processing rules using the prediction-correction paradigm. *In Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. Nashville, TN, USA: ACM. 10-11.

van der Aalst, W.M.P. and B.F. van Dongen. (2013). Discovering Petri Nets from Event Logs. *In Transactions on Petri Nets and Other Models of Concurrency VII.* 372-422.

Viglas, S.D. and J.F. Naughton. (2002). Rate-based query optimization for streaming information sources. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data - SIGMOD '02.* 37- 48.

Walzer, K., T. Breddin, and M. Groch. (2008). Relative temporal constraints in the Rete algorithm for complex event detection. *In Proceedings of the second international conference on Distributed event-based systems - DEBS '08*. Rome, Italy. 147-155. ACM Press.

Walzer, K., M. Groch, and T. Breddin. (2008). Time to the rescue-supporting temporal reasoning in the rete algorithm for complex event processing. *Database and Expert Systems Applications*. 635-642.

Wang, D. and E. Rundensteiner. (2013). Active complex event processing: applications in real-time health care. *Proceedings of the VLDB Endowment.* 1545-1548.

Wang, F., C. Zhou, and Y. Nie, Managing and Mining Sensor Data. Boston, MA: Springer US.

Wang, Y.M., L. Qiu, D. Achlioptas, G. Das, P. Larson, and H.J. Wang. (2007). Subscription partitioning and routing in content-based publish/subscribe systems. *16th International Symposium on DiStributed Computing (DISC'02)*.

Widder, A., R. Ammon, P. Schaeffer, and C. Wolff. (2007). Identification of suspicious, unknown event patterns in an event cloud. *Proceedings of the 2007 inaugural international conference on Distributed event-based systems*. ACM. 164-170.

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

Xu, Y., N. Stojanovic, L. Stojanovic, and D. Kostic. (2013). An Approach for Dynamic Personal Monitoring based on Mobile Complex Event Processing. *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*. ACM. 464.