A CRIME INVESTIGATION TOOL BASED ON PATTERN MATCHING IN CRIME EVENT STREAMS

by

Osamai Osbert

Reg. No: 2010/HD18/1259U

BITC (KYU), DCS (KYU),ITIL

Department of Networks

School of Computing and Informatics Technology, Makerere University

E-mail: omeja4ever@gmail.com,omeja4ever@yahoo.com

Tel: +(256)712 738530

A Project Report Submitted to the School of Graduate Studies in Partial Fulfillment for the Award of a Master of Science in Data Communication and Software Engineering Degree of Makerere University.

OPTION: Mobile Computing

November, 2013

Declaration

I Osamai Osbert do declare to the best of my knowledge that the information presented in this report is original and my own work and effort and has never been submitted to any college, institution or University for any form of award whatsoever.

Signed................................ Date..................................

Osamai Osbert

Approval

This project report has been submitted with my approval as supervisor.

Signed................................. Date...................................

Dr. Benjamin Kanagwa, PhD

Department of Networks,

School of Computing and Information Technology,

Makerere University.

Dedication

This work is dedicated to the men and women of the Uganda Police who tirelessly serve and protect our nation.

Acknowledgement

I would like to thank and appreciate all those who contributed time and effort towards this report. Special thanks go to my supervisor Dr. Benjamin Kanagwa for every support accorded to me in completing this project.

I am greatly indebted to my family (My wife Lucy, daughter Alba, sisters Kevine and Suzan and brothers Euphrasius and Godfrey) for all the support and encouragement they offered throughout the course. I also express my gratitude to my workmates at Tropical Bank Uganda especially Mr. Nicholas Ssesinde for the morale and encouragement they provided me during this course.

May the Almighty God reward them abundantly.

LIST OF ACRONYMS

EJB	Enterprise Java Beans
SOA	Service Oriented Architecture
CEP	Complex Event Processing
XHTML	Extensible HyperText Markup Language
JPA	Java Persistence API
EIS	Extended Information Server
PDF	Portable Document Format
IEEE	Institute of Electrical and Electronic Engineers
EL	Expression Language
HTML	HyperText Markup Language
SNA	Social Network Analysis
GIS	Geographic Information System
SPP	Spatial Point Patterns
AMIT	Active Middleware Technology
GPL	General Public License
SQL	Structured Query Language
ECA	Event Condition Action
POJO	Plain Old Java Object
EPL	Event Processing Language
SDLC	System Development Life Cycle
UML	Unified Modeling Language
JSF	Java Server Faces
Java EE	Java Enterprise Edition
CEL	Cayuga Event Language
ESBs	Event Service Buses
ICT	Information and Communication Technology
CCTV	Closed-Circuit Television
GPL	General Public License
API	Application Programming Interface

Abstract

Crime investigation world over is considered a difficult and laborious process that heavily relies on efficient crime analysis to ensure accurate and timely conclusions. This is not helped by the fact that law enforcement agencies deploy manual investigation processes and computerized systems that cannot quickly identify complex crime patterns.In Uganda, the situation is no different with these agencies facing massive delays in crime solving and increasing numbers of sophisticated crimes, most of them of the organized category. Every year tens of thousands of cases are carried forward to the following year, uncompleted.Majority of these crimes evolve in a long period of time making them even more difficult to predict. Therefore the rate of organized crime is on the rise most of which are orchestrated in vast geographical areas using these complex techniques.

Criminals have become very intelligent due to the advancement of technology and therefore they conduct crimes in an untraceable manner. Intelligence and law enforcement agencies are often faced with the dilemma of having too much data, which in effect makes too little value and the lack of sophisticated network analysis tools and techniques to utilize the data effectively and efficiently.

In this project the phenomenon of Complex Event Processing (CEP) is used to detect patterns in crime related events using the Esper engine for high throughput and performance. CEP analyses low level events to produce a single complex event and has been successfully used in Stock Trading, Network Analysis and other areas. By using this tool, not only will crimes be investigated faster but some future crimes may be prevented based on analysis of current patterns.

Contents

1 CHAPTER ONE: Introduction

Crime investigation world over is considered a difficult and laborious process that heavily relies on efficient crime analysis to ensure accurate and timely conclusions. This is not helped by the fact that law enforcement agencies deploy manual investigation processes and computerized systems that cannot quickly identify complex crime patterns.

In Uganda, the situation is no different with these agencies facing massive delays in crime solving and increasing numbers of sophisticated crimes, most of them of the organized category. Every year tens of thousands of cases are carried forward to the following year, uncompleted. As the usual circle of crime would dictate, fresh cases are reported every day, and, gradually, older cases left uncompleted lose the urgency they initially generated and, inadvertently, they die a natural death [22].

To avert this problem there is need to deploy sophisticated tools, technologies and resources that can enable crime investigators quickly reach reasonable conclusions by identifying patterns of behavior in criminals. In so doing, not only will crimes be investigated faster but some future crimes may be prevented based on recurring patterns identified. However, intelligence and law enforcement agencies are often faced with the dilemma of having too much data, which in effect makes too little value. On one hand, they have large volumes of raw data collected from multiple sources: phone records, bank accounts and transactions, vehicle sales and registration records, and surveillance reports. On the other hand, they lack sophisticated network analysis tools and techniques to utilize the data effectively and efficiently [26].

In this project the phenomenon of Complex Event Processing (CEP) is used to detect patterns in crime related events using the Esper engine for high throughput and performance. CEP analyses low level events to produce a single complex event and has been successfully used in Stock Trading, Network Analysis and other areas.

1.1 Statement of the problem

Due to the manual nature of crime investigations in Uganda, case backlog remains a critical issue leading to delayed justice, unpunished criminals and of course a loss of confidence in the agencies and government by the citizenry. These manual methods often lead to fatigue, poor statistical analysis and the inability to solve crimes through pattern detection and analysis.

Criminals have become very intelligent due to the advancement of technology and therefore they conduct crimes in an untraceable manner. Majority of those crimes evolve in a long period of time making them even more difficult to predict. Therefore the rate of organized crime is on the rise most of which are orchestrated in vast geographical areas using these complex techniques.

Therefore, manual techniques of analyzing such data with a vast variation have resulted in lower productivity and ineffective utilization of manpower[13].There is need to develop a tool to quicken the investigation process by accurately guiding investigators in evidence analysis and also use recurrent patterns to prevent future crimes.

1.2 Objectives

1.2.1 General Objective

To develop a crime investigation tool that helps investigators by detecting crime patterns in event streams to provide investigative leads.

1.2.2 Specific Objectives

Specifically, the objectives of the study are:

(i) Identify existing crime patterns associated with crimes related to causing financial loss.

(ii) Generate events from existing data source into the esper engine for processing.

(iii) Setup and detect crime patterns in events using the esper engine.

(iv) Display to the user/client events matching the pattern and the rate(success percentage).

1.3 Scope

The project focused on crime investigations related to causing financial loss to government in ministries and the tool will be used by the Uganda Police. The tool reads incoming events from an existing data source/existing system presented in a structured data format.

1.4 Significance of the study

To Enhance the crime investigation process by using the Esper engine to detect patterns matched in good time and perform efficient correlations of events. The expected goals of the study are:

(i) Reduce case backlog through quick and multiple processing of case files to restore public confidence in the Police.

(ii) Reduce human intervention by officers in terms of cross referencing and analysis of crime data.

(iii) Provide a foundation for prediction of future crimes based on current crime trend analysis.

(iv) Reduce government expenditure and reliance on manual crime analysis methods.

2 CHAPTER TWO:Literature Review

This section provides a general literature review of major data mining techniques used in existing crime investigation systems,discuses some studies and systems at the Uganda Police related to this project and explains the concept of Complex Event Processing which is the preferred technology for this project.

2.1 Current Technologies in Crime Investigation Systems

Existing crime investigation systems tend to vary in terms of their overall capabilities and technical operation. In one study [2]the existence of prominent criminal investigation software like HOLMES2, BRAINS and Analysts' Notebook which are used by criminal analysts in the United Kingdom and Holland was acknowledged. This category can be classified as early generation systems that mainly focused on analyzing evidence separately without linking multiple sets of evidence in order to solve crimes. They were also not designed to communicate with existing case management systems to facilitate data exchanges.These systems make use of relational database queries using SQL which is slow in terms of processing.

Second generation systems around the world rely heavily on data mining techniques to query large datasets for meaningful patterns in order to help investigators solve crimes. These methods however fall short in terms of supporting decision making largely due to poor processing speeds and querying algorithms.These systems mainly produce graphical representations of links between criminals and other crime entities.

Link Analysis is a technique used in data mining. These tools have for long been used by law enforcement agencies to identify, analyze and visualize relationships between crime entities. In a study [18] , it is revealed that through association paths linking suspects and victims in crime, link analysis discovers information about motives and hence provides investigative leads.Link analysis requires an extensive amount of data preparation and is highly labour intensive.The performance of this technique detoriates with increasing data amounts and is therefore suitable for smaller observation sets.

In order to correlate entities , investigators must manually search for associations by examining a large number of documents that may range from structured database records of crime incidents to unstructured report narratives. Link Analysis is similar to the breadth-first search alogrithm in which a search tree rooted at one of the known enitities. The process involves examining one or more documents and consumes a considerable amount of time.

Another problem with link analysis is high branching factors as a result of very many asociations between entities which increases the complexity of the search alogrithm. Also paths found during analysis may not be useful as they may contain unimportant links.Link Anlaysis heavily relies on domain knowledge and experience making it very difficult to automate the process. Investigators need to determine whether an association between two crime entities is important for uncovering investigative leads. As a result this is a highly costly technique though effective in a way.

Another technique used in crime pattern detection involves several data mining steps like hotspot detection, crime clock, crime comparison and crime pattern visualization. Numerous algorithms are used to relate multiple crime scenes, represent a number of crime scenes on a daily basis, compare different crimes to estimate growth rates and visualize the changes in crime occurrence frequencies [13].

A study on crime network analysis [26] suggests that law enforcement agencies need to deploy reliable data and sophisticated tools as critical tools in the discovering useful patterns in data. The study introduces a data mining technique called Social Network Analysis (SNA) which is used to discover hidden patterns in large volumes of crime related data. An approach of SNA referred to as block modeling is used in criminal networks to reveal associations between subgroups based on a link density measure. Discovery of new structural patterns during this process can enable prevention of crimes and also modify conventional view of certain crimes by investigators.This approach is expected to provide more advanced analytical functionality to assist crime investigation.Sophisticated structural analysis tools are needed to go from merely drawing networks to mining large volumes of data to discover useful knowledge about the structure and organization of criminal networks [26].

2.2 Related Work in Uganda

In Uganda, some studies have been conducted in the area of crime investigation. One such study [21] discusses a model for forensic investigations that performs detection of incidents through system monitoring and performs data analysis to unearth the crime scene, suspect and how the crime was perpetrated. The study proposed a new model based on five iterative phases that were meant to strengthen the crime detection and analysis process. The model suggested depicts the forensic process as iterative as opposed to linear,differentiates the investigations at the primary (suspect) and secondary (victim) crime scenes, introduces a new phase (Traceback) that would reflect the process of arriving at the perpetrators scene, re-defines the phases in the physical and digital crime scene investigation phases in the previous models, re-defines the Deployment phase in the previous model to include the physical and digital crime investigations,reserves only one reconstruction (at the end) but provides for investigative hypotheses during the entire process and is suitable for cyber-crime investigations.

Another study [16] on crime prevention suggested a combined application of data mining techniques alongside GIS (Geographical Information Systems) to discover crime data in disorganized settings like Uganda. Spatial Point Patterns (SPP) based on coordinates of events such as locations of crime incidences and the time of occurrence are used.All or a sample of point pattern may be plotted on the map. The aim of SPP analysis is to detect whether the point pattern is distributed at random, clustered or regular. SPP is typically interpreted as analysis of clustering. A dot map is commonly used to represent SPP. The tool effectively used for analysis of clustering effects is the K function. This method assesses clustering of crime incidences in detection of hot spots wheretime and space relationship analysis is required, the methods used are Knoxs method,Mantels Method and K-nearest neighbour method.

2.3 Existing Systems at Uganda Police

Currently there is no crime investigation system being used at the Uganda Police but there are ongoing efforts to deliver a case management system to be used by all police stations in the country.Most crime related systems in uganda centre on monitoring crime and criminals rather than investigations and they include;online tracking systems for monitoring cyber criminals,mobile phone tracking system to track stolen or lost phones,camera tracking software to monitor identified criminal vehicles via CCTV cameras.

2.4 The Crime Investigation Process

An investigation is an examination, a study, a survey and a research of facts and/or circumstances,situations, incidents and scenarios, either related or not, for the purpose of rendering a conclusion of proof. An investigation, therefore, is based upon a complete and whole evaluation and not conjecture, speculation or supposition.Crime detection and investigation is both an art and a science; a collaboration of common sense, judgment, intellect, experience and an innate intuitiveness along with a grasp of relative technical knowledge. The criminal investigator must continually apply those skills, acquired through study and experience, to the examination and observation of the criminal and his behavior, as well as his social and physical environment [4].

There are several basic types of investigations that law enforcement personnel may undertake in the routine discharge of their duties: Investigations of incidents, which are violations of laws and/or ordinances that include; criminal acts (robbery, assaults, larceny, burglary, murder, illegal weapons, etc) and traffic accident investigations (serious injuries, likely to die, property damage). Personnel investigations into the background, character and suitability of persons in an effort to determine their eligibility for positions of public trust. Investigations of illegal conditions or circumstances, which if left unchecked would cause an increase in traditional crimes. These conditions may include the following: narcotics sales, illegal weapons trafficking, vice type crimes (prostitution, gambling), street gang activity, organized crime, terrorist front activities, fraud and con games, identity theft and computer crimes. Although many of these conditions would dictate self-initiated investigations based upon intelligence rather than reacting to a citizen crime complaint, there are however, times that investigations will in fact result from such individual crime complaints [4].

The official purpose of criminal investigation in most countries is to retrieve information that can be used as evidence in court.The obtained evidence then becomes the basis for judges' and juries' decisions concerning the guilt of prosecuted defendants and the sentences imposed on those found guilty. From the above description it is evident that investigative activities cannot be fully understood if viewed detached from its context, but should be seen as intertwined with other components of the criminal justice system. Therefore, it is helpful to consider how investigators' work relates to the prosecution process. In a prosecutor's application for a summons, a claim is to be made concerning the criminal behavior of a defendant in the past. A prerequisite for issuing a summons is, first and foremost, that the identity of the defendant is clear. Furthermore, the criminal act must be specified with regard to the time and place of the offense. Finally, the circumstances surrounding the offense should be detailed and proven to fulfill the legal requisites for the specified crime classification. The investigative work carried out by the police authority serves to provide the prosecutor with all the above information [1].

2.5 Complex Event Processing

A Complex event is an event that abstracts or aggregates simple (or member) events [11]. Simple and complex events are normally represented in linear ordered sequences called Event Streams. These streams are usually bound by time intervals and may contain different types of events.

Complex Event Processing (CEP) is defined as the process of detecting complex events using continuously incoming events on a lower abstraction level [3]. This study justifies the need for CEP given the fact that single events on their own may not be sufficient in determining certain patterns.

CEP is a foundational technology for detecting and managing the events that happen in event driven enterprises. It is a collection of methods, tools and techniques applied in processing events as they happen. In order to achieve a lot from CEP, happenings of events in enterprises need to be well understood. This can be achieved by organizing events into structures or hierarchies, identifying relationships among events (causal, time, aggregation) and organizing events in different views from different personnel. In CEP, higher- level knowledge is derived from lower-level events which are a combination of various occurrences. CEP can be viewed in two types, the first one involving specification of complex events as patterns and detecting them effectively, whereas the other type involves detecting new patterns as complex events. In the first case, event query languages offer convenient means to specify complex events and detect them efficiently. In the second case, machine learning and data mining methods are applied to event streams.

Detection of complex events is, of course, not an end in itself; an event-driven information system should react automatically and adequately to detected events. Typical reactions include notifications (e.g., to another system or a human user), simple actions (e.g., buy stocks, activate fire extinguishing installation), or interaction with business processes (e.g., initiation of a new process, cancellation or modification of a running process).

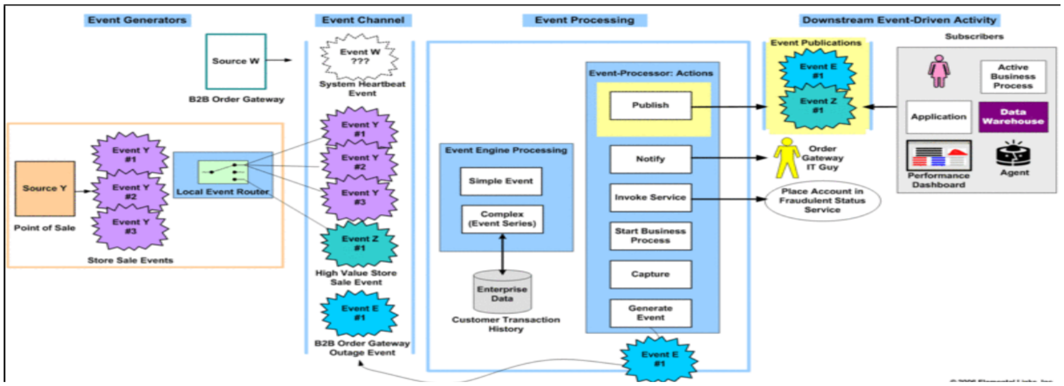


Figure 1: Event Driven Architecture:(Source: Event Processing Masters Thesis by Paul Dekkers 2007)

A study about the history of CEP [10] traces its roots to university and company research groups in the late 90's which were involved in the areas of active databases, event driven simulation, networking and event processing in middleware. This explains partly why the CEP query languages are based on SQL syntax having been influenced by research on active databases. CEP products at that point did not generate much interest until the late 2000's when CEP was deployed as add-ons on SOA architectures and ESBs.

Databases are distinct from event queries used in CEP because of the latter's ability to continuously detect events as they happen rather than just acting on stored datasets. Event processing languages need to enable the possibility of joining several individual events together, so that their combined occurrences over time yield a complex event and complex events must contain the element of time, to track times when events occur.One study [9] introduced the need for revision of events in cases of erroneous data. In practice, there are a number of reasons requiring revisions in event stream processing. For example, an event was reported by mistake, but did not happen in reality (and the mistake was realized later); an event happened, but it was not reported (due to failure of either a sensor, or failure of the event transmission system); or an event was triggered and later revoked due to the transaction failure. Also very often streaming data sources contend with noise (e.g., financial data feeds, Web streaming data, updates etc.) resulting in erroneous inputs and, therefore, erroneous complex event results.

Through Complex Event Processing (CEP), companies and organizations can manage processes in close to real-time. It is however noted that due to the complexity of generic event processing frameworks offered by the industry, the configuration and setup of CEP applications are left to external experts who are more knowledgeable in complex event logic. A CEP application retrieves events for all noteworthy incidents in the business environment. In various parts of the application, event-pattern rules are applied on the incoming event stream to detect relevant patterns, e.g., an uptrend in application errors or execution delays. In response to such patterns, the CEP engine proactively intervenes in the business environment, e.g., by temporarily allocating additional resources, throttling uncritical business tasks or notifying system administrators [10].

Pattern matching is a key feature of all CEP technologies which involves finding subsets of data matching a given pattern and also relationships between those subsets. A study about CEP under uncertainty [24] explains the role pattern matching plays in allowing users to look beyond individual events and find specific collection sets.

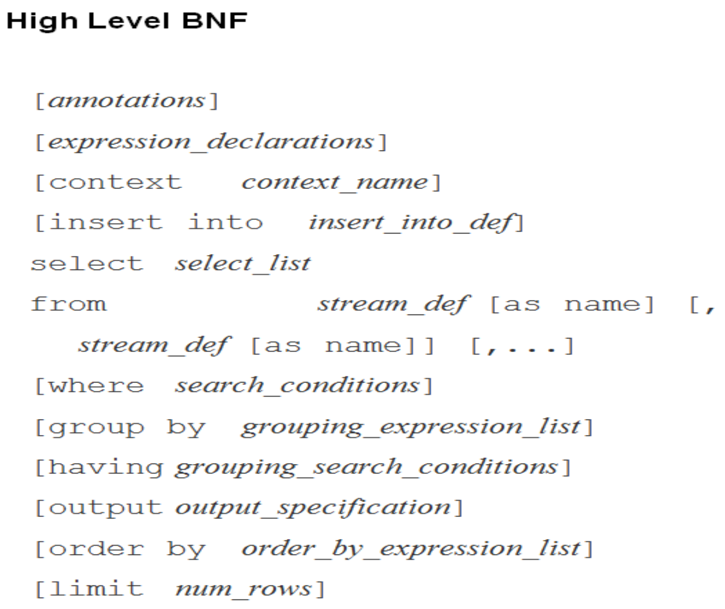
Solution templates are proposed [15]to perform data mining procedures on historical data to identify event patterns besides real-time monitoring. The central concept of any template's event processing infrastructure is the event processing map, a predefined orchestration of event adapters and event services. Event adapters may be considered the actual interface to the underlying source system: Depending on their implementation, event adapters translate real-world actions (such as a user actually placing a bet in an online gambling platform) into event representations of a certain event type, and vice versa. Event services receive events from event adapters or other event services, process them based on implementation of specific logic, and respond back to the map.Detecting events is based on some considerations like, some events sharing time elements, the order of events, time bounds within events and detection of events of long time lags. To gain an insight into processes there is need to include the following components in CEP application, facilities (graphical or textual) for precise description of complex patterns of events, scalable performance, modular rules engines to detect complex patterns of events, facilities for defining and composing event pattern triggered rules for pattern abstraction.

2.6 Alogrithms and BNF for Esper Engine

The engine used for this project is the esper engine which makes use of indexes, a data structure that improves the speed of data retrieval operations. For sorted access it may prefer a binary tree index while a hash-based index is great for key lookups.For efficient matching of incoming events to statements the engine uses inverted indexes.Multi-version concurrency control is a concept used for variables and also for filters to allow concurrency and reduce locking.The match-recognize pattern matching functionality is built using nondeterministic finite automata (NFA).Query planning based on the analysis of expressions used in the where-clause is another technique used by the engine. The execution strategy may choose nested-loops versus merge joins [5].

The Esper Engine uses the EPL(Event Processing Language) to define statements stored in the engine. EPL queries are created and stored in the engine, and publish results to listeners as events are received by the engine or timer events occur that match the criteria specified in the query. Events can also be obtained from running EPL queries via the safeIterator and iterator methods that provide a pull-data API.The select clause in an EPL query specifies the event properties or events to retrieve. The from clause in an EPL query specifies the event stream definitions and stream names to use. The where clause in an EPL query specifies search conditions that specify which event or event combination to search for [5].

EPL queries can be simple queries or more complex queries.A simple select contains only a select clause and a single stream definition. Complex EPL queries can be built to feature a more elaborate select list utilizing expressions, may join multiple streams, may contain a where clause with search conditions and so on [5].Below is the syntax/BNF for the EPL language used by the Esper engine.



High Level BNF

```
[annotations]
[expression_declarations]
[context    context_name]
[insert into    insert_into_def]
select  select_list
from              stream_def [as name]  [,
   stream_def [as name]]  [,...]
[where  search_conditions]
[group by  grouping_expression_list]
[having grouping_search_conditions]
[output output_specification]
[order by  order_by_expression_list]
[limit  num_rows]
```

Figure 2: BNF for Esper Engine(Source: Esper Tutorial)

2.7 Event Processing Engines

An analysis of Event Processing Languages [3] revealed the need to shift from using general-purpose languages like C, Java, C++ e.t.c for CEP applications due to low-level complexities. Using such languages along with complexities like data structures and algorithms can only complicate the development process. The study provided a detailed analysis of existing CEP programming Languages and platforms based on their expressivity and integration capabilities. Expressivity is measured by one of the following abilities, filtering streams by event type, processing a subset of events (windows), data extraction and aggregation of data over events, performing conjunctions and disjunctions, show temporal relations between events, showing causality of events, negation and counting of events, event instance selection and consumption to prevent reuse of events in pattern detection and integration of event data and non-event data (data from outside). Languages are also grouped into the categories of data stream query languages, composition-operator-based languages, production rule languages and logical formulas.

STREAM (Stanford Stream data Manager) is a language whose focus was to develop methods to manage and query data in data streams and was a result of a research project at Stanford University. The project also produced a CEP engine called STREAM and an Event Query Language called Continuous Query Language to query events. STREAM was a basis for other data stream languages like Esper and its querying syntax resembles SQL very strongly.

Borealis is a CEP engine developed at Brandeis University and MIT that uses a ”boxes and arrows” approach. Queries are described graphically with queries as boxes and streams as arrows connecting boxes. The approach was first used in an earlier engine, Aurora. The main difference between Borealis and stream languages is the focus on query evaluation that Borealis offers resulting in less abstract queries than STREAM.

Active Middleware Technology (AMiT) enables IBM middleware to become event-based. This technology is implemented in several products, most notably extending Web-Sphere Broker with CEP capabilities. As WebSphere is a commercial product, it is not freely available (requires registration .Basic events are declared with their attributes in event tags. Lifespans are windows defined by two events, an initiator and a terminator event. Lifespan types are therefore declared by referencing start and end event types. Whenever an event matching the initiator specification is detected, a new lifespan of this type is opened, and when an event matching the terminator specification is detected, the lifespan is closed. Complex events are called situations. A situation consists of at least one data attribute (it has to carry at least one kind of information), exactly one operator, and a lifespan type. Situations are only tried to be detected in lifespans of its type. A lifespan may be referenced by multiple situations

RuleCore is a CEP engine developed by Analog Software, building on research at the University of Skyde. As the name suggests, rules are the central concept of ruleCore. The ruleCore engine processes events using ECA (Event-Condition-Action) rules that consist of three parts: for every event (basic or complex), check a condition; if it is true, execute the action. ruleCore has two implementations; an open source variant called ruleCore, released under the terms of the GPL; as well as a commercial version called ruleCore CEP Server. RuleCore uses so-called detector trees for event detection. Leaf detector nodes (detector nodes without children) detect single events (they pick up events of their type). They are inactive until an event of their type is delivered to the rule (usually by entering the system, although exceptions are mentioned in the next paragraph), after which point they are always active. To detect complex events, a detector tree is built: the leaves detect simple events, and inner nodes detect complex events depending on whether its children detected events.

SASE+ is a CEP system developed at the University of Massachusetts, Amherst. It is an extension of the older SASE system. The system is designed for event streams with many events per time unit and also queries using large time windows, creating new issues regarding efficient query execution. The project’s purpose is to devise techniques for high-performance querying of event streams, using a declarative, composition-operator-based language. Although SASE+ is an agile language and concentrates only on pattern matching on streaming data, the pattern matching properties of SASE+ can be used in more general contexts.

Esper is an open-source CEP engine, developed by EsperTech Inc. and volunteers, released under the GNU General Public License (GPL v2). As stated on the official web site, it is designed for CEP and Event Stream Processing (ESP).There are two implementations of Esper, Esper for Java and NEsper for .NET. Both supply an API to access the engine features, such as deploying queries, sending events into the engine and retrieving events out of the engine, in their respective language. Events are objects in their respective language; for Esper, events can be instances of java.util.Map,org.w3c.dom.Node (Java representations of XML documents), or other Java objects. Regardless of the implementation language, queries are stated in a SQL-like language called Event Processing Language (EPL).

Cayuga is a research CEP engine developed at Cornell University. It sets itself apart from other engines in that it deliberately sacrifices expressivity for performance, targeting applications running large numbers of queries. It is free software, available under the terms of the BSD license. Cayuga uses an Event Query Language called Cayuga Event Language (CEL). While its syntax resembles SQL, like many data stream languages, it also offers patterns, although using a different approach compared to Esper, inspired by regular expressions.

Drools, also known as JBoss rules, is a production-quality business rule management system, including a production rule engine. It is free software, released under the Apache License. The Drools engine is implemented in Java, as is JBoss, and is also controlled using that language. Initialization of the engine and deployment of rules is implemented in Java. Also, as unusual for production rule engines, rules never fire by themselves, but are issued to do so by the Java program that controls the engine. In addition, Drools can be extended by defining so-called Domain Specific Languages. These are languages that may have a different syntax than the standard Drools syntax to write queries in. Rules in Domain Specific Languages are then translated into the Drools language when inserted into the engine.

XChangeEQ is a research Event Query Language. It is developed at the University of Munich and designed for automated reasoning on the Semantic Web. XChangeEQ introduces a new style of event querying. It separates event query features into four so-called dimensions: Data extraction, event composition, temporal relationships, and event accumulation. Most operators belong to exactly one of these dimensions. This was done to define clear semantics. As XChangeEQ is designed for use on the Web, it works best at processing tree-structured events, such as XML messages. Queries are generally structured like the XML representations of the events queried. For querying simple events, it embeds the Xcerpt language. Xcerpt queries apply patterns to XML documents, similar to templates. Change is a reactive programming language. Using Event-Condition-Action rules, it allows Web sites to react to changes at other Web sites, for example by updating its own data.

TelegraphCQ, from the University of California at Berkeley, is designed to provide event processing capabilities alongside relational database management capabilities by utilizing the PostgreSQL open source code base. The existing architecture of PostgreSQL is modified to allow for continuous queries over streaming data. Several components of the PostgreSQL engine underwent very little modification, while others were significantly changed. The most significant component of the TelegraphCQ system is the ”wrapper,” which allows for data to be pushed or pulled into the Telegraph processing engine, and custom wrappers allow for data to be obtained from any data source .

BEA Systems in 2007 introduced of their WebLogic Real Time and WebLogic Event Server systems. More specifically, their Event Server technology is a focus on event-driven service oriented architecture which provides a response to events in real-time. As part of the package, they provide a complete event processing and event-driven service-oriented architecture infrastructure that supports high-volume, real-time, complex, event-driven applications. This is one of the few commercial offerings of a complete, integrated solution for event processing and service-oriented architectures.

Truviso is a commercial event processing engine that is based on the research toward the Telegraph CQ project at UC Berkeley. The”claim to fame” for Truviso is that it supports a fully functional SQL, and integrates PostgreSQL relational database alongside a stream processing engine. The integration of PostgreSQL leads to other aspects of the Truviso system. The queries are simply standard SQL with extensions that add functionality for time windows and event processing. Carried over from PostgreSQL are user-defined functions, as well as JDBC and ODBC interfaces. In addition, the use of an integrated relational database allows for easy caching, persistence, and archival of data streams, as well as queries that include not only real-time data, but also the historical data [8].

3 CHAPTER THREE:Methodology

The development of the system followed the SDLC development methodology.In addition, object oriented approach was used in design and implementation.

3.1 Requirements Gathering

3.1.1 Interviews

Interviews were conducted with investigative officers at Police headquarters, Jinja Road Police station and the Special Investigation Unit, Kireka. This was carried out to ascertain the nature of financial crime cases, the investigation processes involved and the criteria used to zero down on a suspect. Telephone interviews were also conducted in situations where physical access was challenging.

The questions asked were related to the following:-

(i) Type of evidence gathered and used in such investigations.

(ii) The patterns which are common in most cases and which the system will base upon.

(iii) Any existing systems used for crime investigation and formats of existing crime records.

3.1.2 Existing Literature

Existing literature on Complex Event Processing was read to determine the best design approach for the project and to discover the benefits of using the CEP model.Implementations of the Esper engine in particular were studied to provide a benchmark for the project implementation.

The Annual Police report was also consulted to acquire an in depth understanding of the impact of crimes to society and government. Case statements some handwritten were also analyzed and these provided guidance on data to capture and track in the system.

3.1.3 Questionnaires

Questionnaires were distributed to investigation officers and their superiors to determine the need for such a tool and the benefits it would bring to the Police force. The returned forms portrayed a clear need for the system as it would reduce on processing time and reduce case backlog.

3.2 Design

This section describes the system environment, interactions, requirements, architecture and input/output formats as well as processing logic. The system was modeled using UML (Unified Modeling Language) tools based on the object oriented design approach.

3.2.1 System Overview/Context Diagram

At the highest level, the system interfaces with an existing crime database system in order to retrieve records which are converted to events and processed.

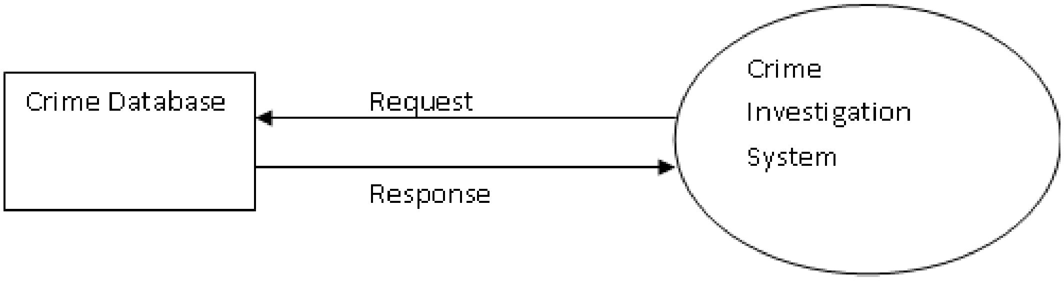


Figure 3: Context Diagram

3.2.2 System Architecture

The architectural design comprises of the external database which provides input data, the application server which describes business logic and links all components (Client, Database and Web

Server), the web client and the web server that processes requests and provides responses.

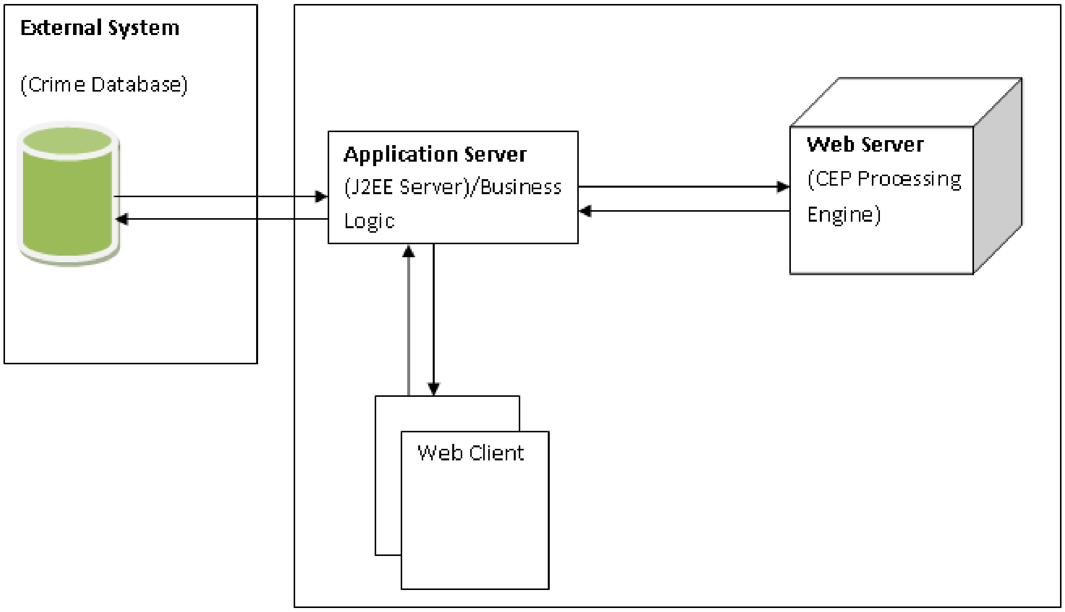


Figure 4: Architecture

3.2.3 Data Flow Diagram

The process was designed to start from the web client (user) making a request for records from an existing database. The returned data is sent to the processing engine and checked against the defined pattern. Based on the computation a response is displayed to the client in form of matching events and match percentages.

3.3 Implementation

The tool was developed as a web-based application to enable access by users throughout the Police network and to provide capabilities for access by mobile devices as well. Other reasons for choosing a web application include; OS platform independence, ease of maintenance and limited

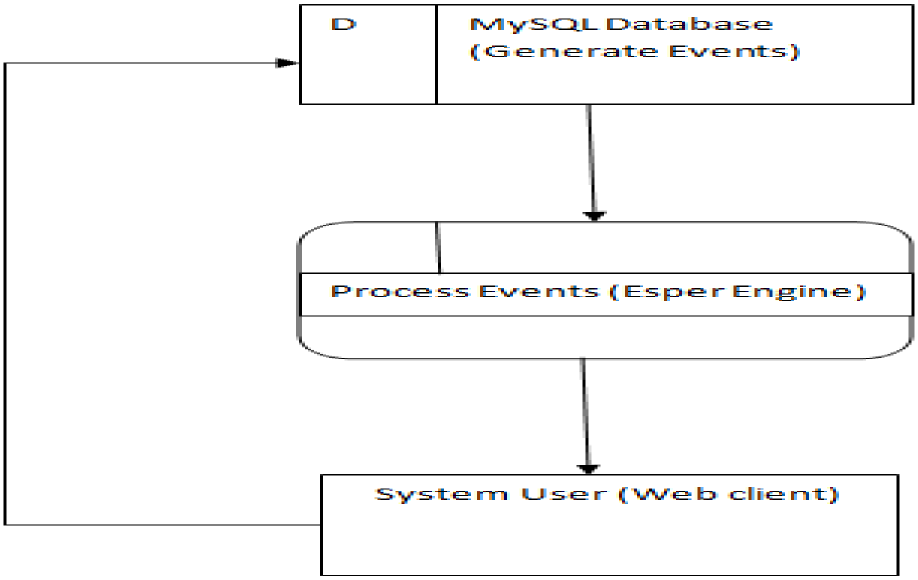


Figure 5: Data flow Diagram

upgrades from the client side. It was built on the Java EE platform using the glassfish application server and the JSF 2.1 framework.

The Java EE platform uses a simplified programming model. XML deployment descriptors are optional. Instead, a developer can simply enter the information as an annotation directly into a Java source file, and the Java EE server will configure the component at deployment and runtime. These annotations are generally used to embed in a program data that would otherwise be furnished in a deployment descriptor. With annotations, you put the specification information in your code next to the program element affected. In the Java EE platform, dependency injection can be applied to all resources a component needs, effectively hiding the creation and lookup of resources from application code. Dependency injection can be used in EJB containers, web containers, and application clients. Dependency injection allows the Java EE container to automatically insert references to other required components or resources, using annotations [6].

Java Server Faces technology is a user interface framework for building web applications. The main components of Java Server Faces technology are as follows: A GUI component framework, a flexible model for rendering components in different kinds of HTML or different markup languages and technologies. A Renderer object generates the markup to render the component and converts the data stored in a model object to types that can be represented in a view, a standard

Render Kit for generating HTML/4.01 markup. The features supporting the GUI components include; Input validation, Event handling, Data conversion between model objects and components, Managed model object creation, Page navigation configuration and Expression Language (EL). For this project the Java EE 6 platform was used with Java Server Faces 2.1 and Expression Language 2.2 [6].

The Client tier was developed using XHTML and using the Expression Languages to connect the business layer (Managed Beans). The PrimeFaces library was used to provide a desktop look and feel to the interface. PrimeFaces is a lightweight library that hides complexities from users while providing extended capabilities to the default XHTML controls as well as enabling quick software development.

This JavaEE Server contains business logic in form of managed beans (POJOs) interacting with the MySQL database through the Java Persistence API (JPA) 2.0 and servlets.The Java Persistence API (JPA) is a Java standards-based solution for persistence. Persistence uses an object/relational mapping approach to bridge the gap between an object-oriented model and a relational database. The Java Persistence API can also be used in Java SE applications, outside of the Java EE environment. Java Persistence consists of the following areas; the Java Persistence API, the query language and Object/relational mapping metadata [6].

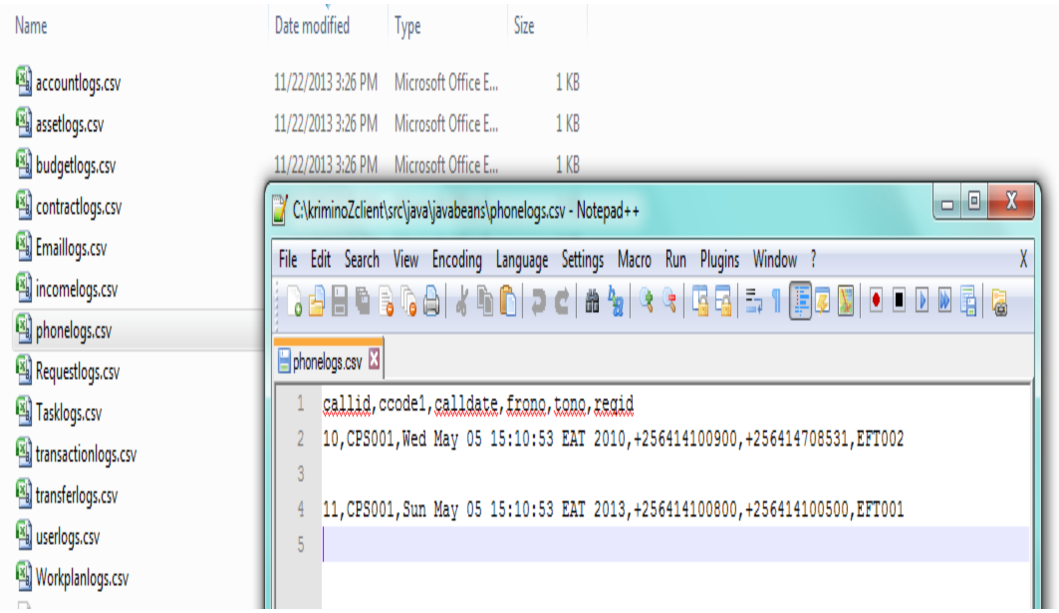
The EIS (Extended Information Server) Tier is the external system that provides crime records to be analyzed. It is JSF web application running a MySQL Server 5.5 database. The relevant queries were written to return records to the main system through the JPA entities.JPA entity classes are generated from the database tables and these then provide a link to the crime investigation web application via the JPA.

3.3.1 Generating Events from Existing Database

This was achieved using the Java Persistence API (JPA) of the JavaEE6 platform. The Java Persistence API (JPA) is a Java standards-based solution for persistence. Persistence uses an object/relational mapping approach to bridge the gap between an object-oriented model and a relational database. The Java Persistence API can also be used in Java SE applications, outside of the Java EE environment. Java Persistence consists of the following areas; the Java Persistence API, the query language and Object/relational mapping metadata [6].

Persistence units are used to define a set of all entity classes that are managed by the Entity Manager instances in an applciation. The entity classes represent the data contained within a single data store.

A parameter (case file number) is passed from the client interface to a parameterized query of an event entity class to return all records pertaining to the supplied case file number.The returned records/events are extracted programmatically from the list set and written in CSV file format to the project class path as depicted below for the phonelogs event.



Name	Date modified	Type	Size
accountlogs.csv	11/22/2013 3:26 PM	Microsoft Office E...	1 KB
assetlogs.csv	11/22/2013 3:26 PM	Microsoft Office E...	1 KB
budgetlogs.csv	11/22/2013 3:26 PM	Microsoft Office E...	1 KB
contractlogs.csv			
Emaillogs.csv			
incomelogs.csv			
phonelogs.csv			
Requestlogs.csv			
Tasklogs.csv			
transactionlogs.csv			
transferlogs.csv			
userlogs.csv			
Workplanlogs.csv			

C:\kriminoZclient\src\java\javabeans\phonelogs.csv - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
phonelogs.csv

```
callid,ccode1,calldate,frono,tono,regid
10,CPS001,Wed May 05 15:10:53 EAT 2010,+256414100900,+256414708531,EFT002

11,CPS001,Sun May 05 15:10:53 EAT 2013,+256414100800,+256414100500,EFT001

```

Figure 6: Phonelogs event generated from the Database in the project path

3.3.2 Setting up and Detecting Pattern Matches using Esper

Esper is a component/engine for complex event processing (CEP) and event series analysis, available for Java as Esper, and for .NET as NEsper.Esper enables rapid development of applications that process large volumes of incoming messages or events, regardless of whether incoming messages are historical or real-time in nature. Esper filters and analyzes events in various ways, and responds to conditions of interest.

Complex event processing (CEP) delivers high-speed processing of many events across all the layers of an organization, identifying the most meaningful events within the event cloud, analyzing their impact, and taking subsequent action in real time.

Esper offers a Domain Specific Language (DSL) for processing events. The Event Processing Language (EPL) is a declarative language for dealing with high frequency time-based event data.Esper and Event Processing Language (EPL) provide a highly scalable, memory-efficient, in-memory computing,SQL-standard, minimal latency, real-time streaming-capable Big Data processing engine for historical data, or medium to high-velocity data and high-variety data.

Esper was the preferred engine over other CEP engines and database systems for the following reasons ;

(i) Esper scales well with a throughput of between 10,000 to 200,000 events processed per second in single-threaded mode.It also scales well with additional Java Virtual Machines, CPUs and/or memory.

(ii) Esper has low latency given that data is not saved and queried later as is the case with databases and all the computing is in-memory computing where data arrives in streams and is processed using continous SQL-conforming queries.

(iii) Esper combines Stream Processing(filtering,joins and aggregation) and Complex Event Processing(Causality) into one single language/platform(EPL).

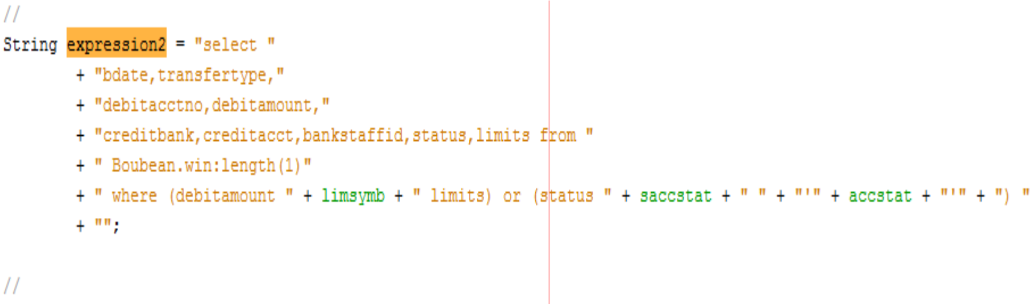
(iv) Esper's EPL language for statement definitions is similar to SQL which is a standard query language.

(v) Esper supports multiple event formats(XML,CSV,HashMaps e.t.c) which makes it convenient to use.

(vi) Esper was written to simplify the complexity of pattern detection and detects many kinds of patterns and is also open source.

Esper is an event stream processing (ESP) and event correlation engine (CEP) written in Java. Basically instead of working as a database where you put stuff in to later poll it using SQL queries, Esper works as real time engine that triggers actions when event conditions occur among event streams. A tailored Event Processing Language (EPL) allows registering queries in the engine, using Java objects (POJO, JavaBean) to represent events. A listener class - which is basically also a POJO - will then be called by the engine when the EPL condition is matched as events come in. The EPL allows expressing complex matching conditions that include temporal windows, and join different event streams, as well as filter and sort them [5].

The Event Processing Language (EPL) was used to define sub-patterns for each of the five events(Phonelogs, Funds Requests, Funds Release, Payments and Emails) which in combination form one pattern. The EPL uses SQL-like syntax to define conditions for expected output and in this project the operators and some values are passed from the client web interface to the EPL statements as seen below.



```
//
String expression2 = "select "
        + "bdate,transfertype,"
        + "debitacctno,debitamount,"
        + "creditbank,creditacct,bankstaffid,status,limits from "
        + " Boubean.win:length(1)"
        + " where (debitamount " + limsymb + " limits) or (status " + saccstat + " " + "'" + accstat + "'" + ") "
        + "";

//
```

Figure 7: Sample EPL statement for Funds Release event stream

The above expression applies to a java bean class related to a given event (Boubean class belonging to the Funds Release Event) and bean is connected to the generated CSV file and to the engine using a hashmap configuration and CSV Adapter API in Esper. The listener class records all returned records matching set criteria/pattern and writes to a text file which is then read and displayed to the user/web client.

3.3.3 Displaying Computed Results to Web Client using JSF framework/PrimeFaces

The JavaServer Faces technology framework was used to develop the client input ans display interfaces interacting with managed beans using the Expression Language and combining with PrimeFaces libraries to provide a rich user interface.

JavaServer Faces technology is a server-side component framework for building Java technology-based web applications.JavaServer Faces technology consists of the following:An API for representing components and managing their state; handling events, server-side validation, and data conversion; defining page navigation; supporting internationalization and accessibility; and providing extensibility for all these features. Tag libraries for adding components to web pages and for connecting components to server-side objects JavaServer Faces technology provides a well-defined programming model and various tag libraries. The tag libraries contain tag handlers that implement the component tags. These features significantly ease the burden of building and maintaining web applications with server-side user interfaces (UIs) [6]. Reasons choosing JSF technology include;

(i) JavaServer Faces technology offers a clean separation between business logic and presentation for web applications.

(ii) JavaServer Faces technology leverages familiar component and web-tier concepts without limiting you to a particular scripting technology or markup language.

Events processed by the engine are read from the text files periodically and displayed to a client XHTML web page using the Expression Language (EL) which interacts with the JSF managed bean to return computed results.The EL allows page authors to use simple expressions to dynamically access data from JavaBeans components.

The PrimeFaces library was used to provide a desktop look and feel to the interface. PrimeFaces is a lightweight library that hides complexities from user while providing extended capabilities to the default XHTML controls as well as quick software development. Records matching the pattern are displayed in tabular format along with the percentage matching rate computed. Another output in PDF format is also created to the local disk of the user using the itextpdf library.

3.4 System Testing

Junit was used to test the system for the state of managed beans, confirming if the navigation commands lead to the correct pages, checking if invalid error messages appear, testing application configurations and database connections.

4 CHAPTER FOUR:Crime Investigation Tool

4.1 Introduction

The tool operates in a request-response style providing the user with a web interface that allows them to define search conditions and determine the kind of events they are interested in. Based on the selection criteria a response is sent back to the web page and a PDF format file is generated to a local location. The logic used in the development involves processing streaming records as events and detecting those matching set patterns.

4.2 3-Tier Architecture

The system consists of four components at the highest level. These are the Web Client, Processing Engine (Esper), Data Source (MySQL Database) and Application Server (Glassfish 3.2).

4.2.1 Web Client

This is used by the crime investigators to select record of particular cases to be sent to the engine for processing and also to receive response from the engine through web pages. The client communicates with the database to generate input data and interacts with managed beans (business logic) to send and receive events from the processing engine.

4.2.2 Processing Engine

This is the component that filters incoming events based on defined crime patterns. Records generated from the database and converted to events provide input to the engine. The engine sends regular responses to the web client showing events that have matched the described pattern. The engine implements a listener which constantly receives events as they stream in.

4.2.3 Data Source

Records requested by the client are generated in CSV (Comma Separated Values) format and sent to the engine for processing. The records are stored in a MySQL Database for all cases that are being investigated. The events are represented in POJO (Plain Old Java Object) format with getters and setters in a form understandable to the CEP engine.

4.2.4 Application Server

This enables the all the other components to communicate and provides a central control point for the web application to deploy, run and compile as enable other configurations.

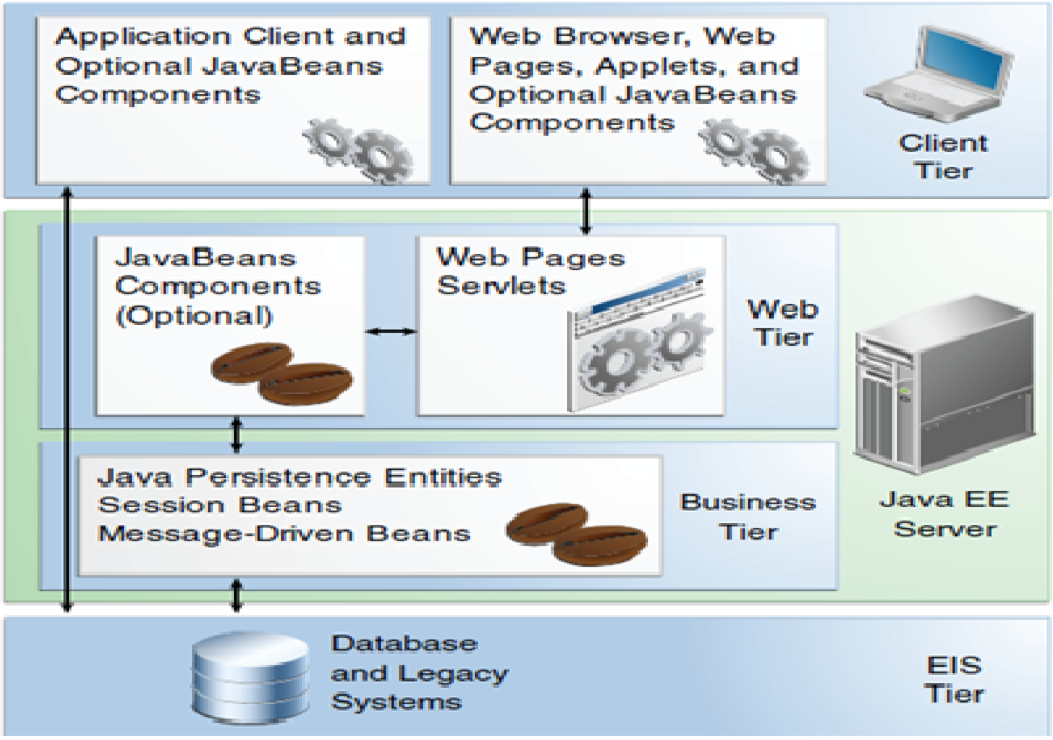

Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
JavaBeans Components (Optional)
Web Pages Servlets
Web Tier
Java Persistence Entities Session Beans Message-Driven Beans
Business Tier
Java EE Server
Database and Legacy Systems
EIS Tier

Figure 8: High Level Architecture:

source: http://docs.oracle.com/javaee/6/tutorial/doc/bnaay.html

4.3 Crime Pattern Used

The pattern is made of five major events (Funds Request, Funds Release, Payments, Phone Logs, Email Logs) which together represent the general pattern used.Any violations detected in any of the five events leads to a degree of pattern match.

4.3.1 Funds Request By Ministry

Before any amount of money is disbursed to any government entity there must be a budget designed at the Ministry of Finance (government) stipulating funds available for different activities. These funds are availed on a periodic basis to the entity most commonly quarterly.

Likewise the government entity requesting for funds must provide a detailed budget to the Ministry of Finance if it is another ministry or through the Ministry of local government for Local Governments. This work plan provides a form of accountability to government for the released amounts.Requests for release of funds must duly be authorized by a designated official permitted to act on behalf of the entity.

Therefore for this event the parameters used are;

1. Identification of authorization level of the requesting officer (Must be the Permanent Sectary or authorized replacement)
2. Checking the availability of a workplan for a given request.

4.3.2 Funds Release By Central Bank (BOU)

This is the process of releasing funds to requesting entity on a periodic basis as per the budget arrangements. It involves transferring funds from the government account at the Central Bank (Bank of Uganda) to the respective accounts via electronic payment systems (EFTs).

The parameters used here are;

1. Checking that the debited account is not in a dormant status(Dormant Accounts not to be debited).

2. Checking that the disbursed amount does not exceed the set limit for daily releases to ministries.

4.3.3 Emails and Phone Logs Events

These are closely linked to the Funds Release event given that before money is released a confirmation email mut be sent from the treasury endorsing the request and a phone call must be made from the Central Bank to the Permanent Secretary confirming the authorizer.

Therefore for these two events,the parameters are ;

1. Availability of a confirmation email record from the treasury.
2. Availability of confirmation phone call record from Bank of Uganda to the Permanent Secretary.

4.3.4 Payments Event

This is a record of how the funds were spent and may include payment details, account statements from the entity and suppliers accounts and also account opening details.A report of activities completed periodically must be availed to get value for money. Here the work done is verified against the budget or work plan submitted at the point of request for funds.

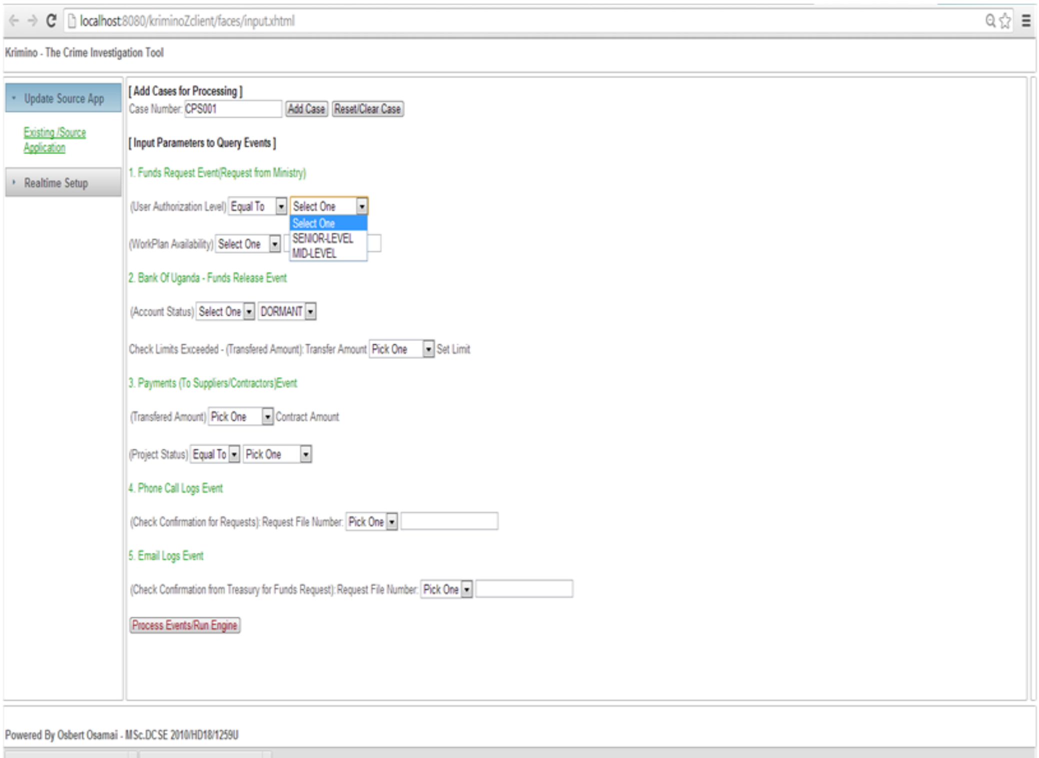
The parameters include;

1. Checking if amount paid exceeds agreed amount in contract.
2. Checking that work being paid for was completed and not pending.

4.4 Tool Functionality/Operation

4.4.1 Client Request

In the single mode one case file number is added for processing while in the multi-mode more than one case is selected or added for processing.Selections are made at the different event sections defining conditions for returned records and the request is submitted.


localhost:8080/kriminoZclient/faces/input.xhtml
Krimino - The Crime Investigation Tool
Update Source App
Existing /Source Application
Realtime Setup
Add Case
Reset/Clear Case
Process Events/Run Engine
Powered By Osbert Osamai - MSc.DCSE 2010/HD18/1259U

[Add Cases for Processing]
Case Number: CPS001

[Input Parameters to Query Events]

1. Funds Request Event(Request from Ministry)

(User Authorization Level) Equal To ▾ Select One ▾
Select One
SENIOR-LEVEL
MID-LEVEL

(WorkPlan Availability) Select One ▾

2. Bank Of Uganda - Funds Release Event

(Account Status) Select One ▾ DORMANT ▾

Check Limits Exceeded - (Transfered Amount): Transfer Amount Pick One ▾ Set Limit

3. Payments (To Suppliers/Contractors)Event

(Transfered Amount) Pick One ▾ Contract Amount

(Project Status) Equal To ▾ Pick One ▾

4. Phone Call Logs Event

(Check Confirmation for Requests): Request File Number: Pick One ▾

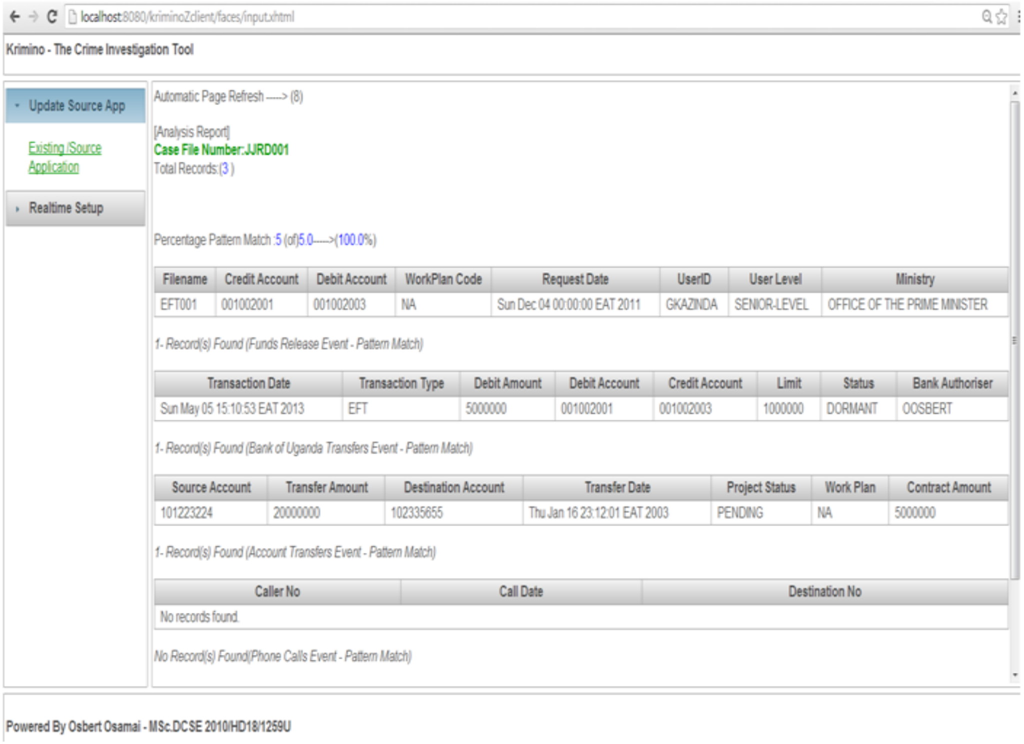
5. Email Logs Event

(Check Confirmation from Treasury for Funds Request): Request File Number: Pick One ▾

Figure 9: Case Input and Selection screen

4.4.2 System Response

The output is form of a single web view and PDF format files depending on the input mode (single or multi-mode).For the multi-mode a PDF file is created for the same number of cases file numbers processed and a single web view for one of the cases as below;


localhost:8080/kriminoZclient/faces/input.xhtml
Update Source App
Existing /Source Application
Realtime Setup

Krimino - The Crime Investigation Tool

Automatic Page Refresh -----> (8)

[Analysis Report]
Case File Number:JJRD001
Total Records:(3)

Percentage Pattern Match :5 (of)5.0----->(100.0%)

Filename	Credit Account	Debit Account	WorkPlan Code	Request Date	UserID	User Level	Ministry
EFT001	001002001	001002003	NA	Sun Dec 04 00:00:00 EAT 2011	GKAZINDA	SENIOR-LEVEL	OFFICE OF THE PRIME MINISTER

1- Record(s) Found (Funds Release Event - Pattern Match)

Transaction Date	Transaction Type	Debit Amount	Debit Account	Credit Account	Limit	Status	Bank Authoriser
Sun May 05 15:10:53 EAT 2013	EFT	5000000	001002001	001002003	1000000	DORMANT	OOSBERT

1- Record(s) Found (Bank of Uganda Transfers Event - Pattern Match)

Source Account	Transfer Amount	Destination Account	Transfer Date	Project Status	Work Plan	Contract Amount
101223224	20000000	102335655	Thu Jan 16 23:12:01 EAT 2003	PENDING	NA	5000000

1- Record(s) Found (Account Transfers Event - Pattern Match)

Caller No	Call Date	Destination No
No records found.		

No Record(s) Found(Phone Calls Event - Pattern Match)

Powered By Osbert Osamai - MSc.DCSE 2010/HD18/1259U

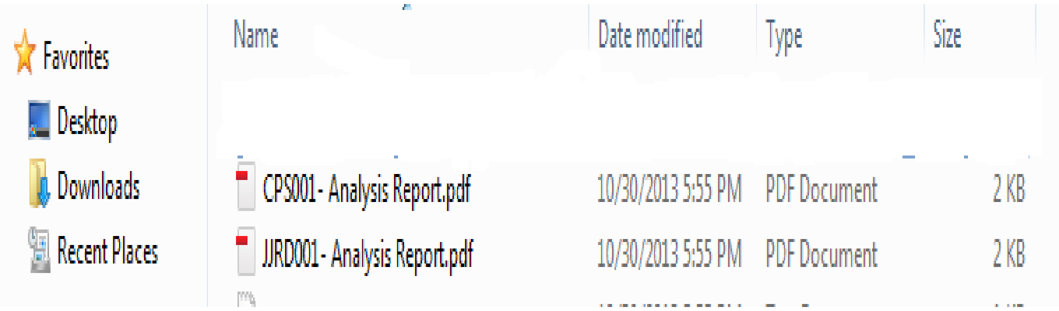
Figure 10: Web View Display

4.4.3 Realtime Capability

The system provides realtime monitoring capabilities in the event that values are edited in the data source.Computations for match rate and number of records are recomputed each time the source is changed. An interface is provided to set one case which needs to be monitored and only one case is monitored at a time as the screen refreshes every 10 seconds.

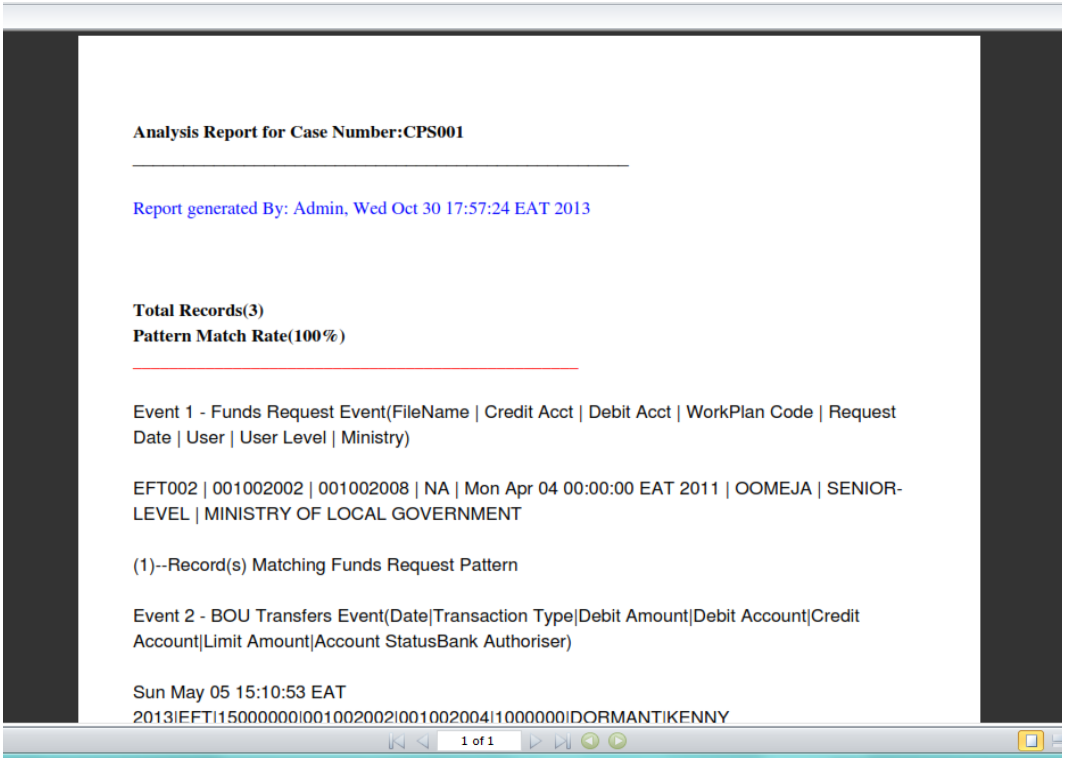
4.4.4 Events Logs

The system keeps track of all events processed by the engine with details of when they were processed and this information is held in a text file in a local location.



Name	Date modified	Type	Size
CPS001- Analysis Report.pdf	10/30/2013 5:55 PM	PDF Document	2 KB
JJRD001- Analysis Report.pdf	10/30/2013 5:55 PM	PDF Document	2 KB

Figure 11: Generated PDF files for multiple Case Files



Analysis Report for Case Number:CPS001

Report generated By: Admin, Wed Oct 30 17:57:24 EAT 2013

Total Records(3)
Pattern Match Rate(100%)

Event 1 - Funds Request Event(FileName | Credit Acct | Debit Acct | WorkPlan Code | Request Date | User | User Level | Ministry)

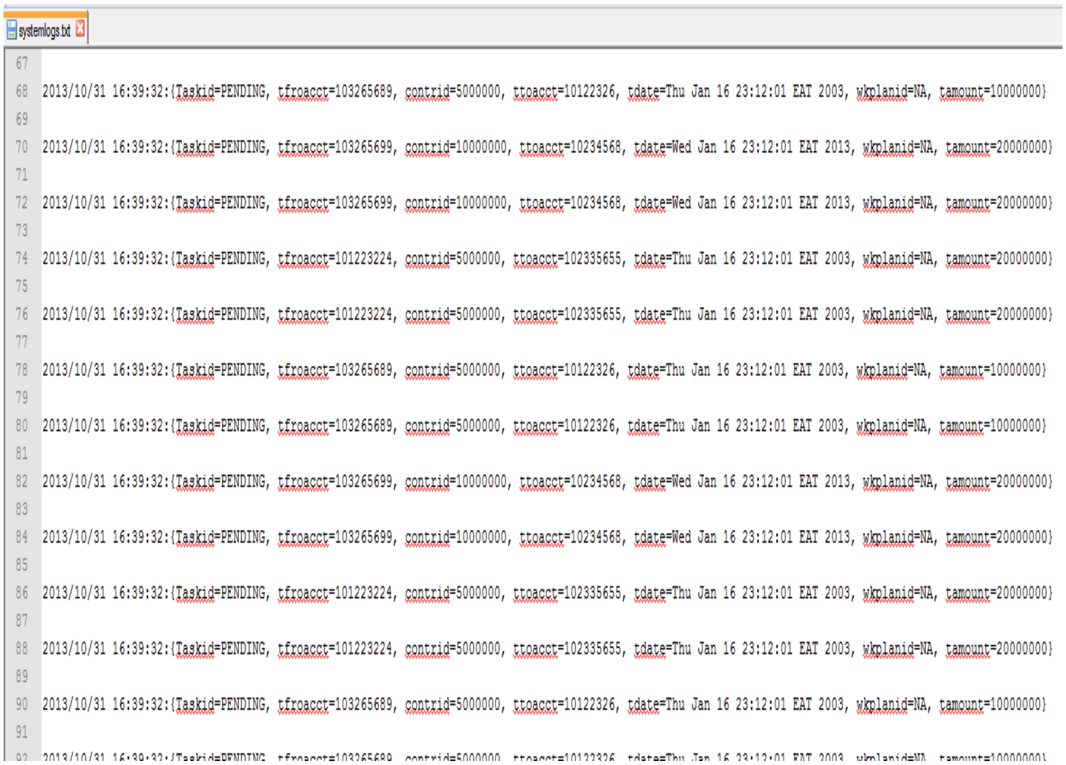
EFT002 | 001002002 | 001002008 | NA | Mon Apr 04 00:00:00 EAT 2011 | OOMEJA | SENIOR-LEVEL | MINISTRY OF LOCAL GOVERNMENT

(1)--Record(s) Matching Funds Request Pattern

Event 2 - BOU Transfers Event(Date|Transaction Type|Debit Amount|Debit Account|Credit Account|Limit Amount|Account StatusBank Authoriser)

Sun May 05 15:10:53 EAT 2013|EFT|15000000|001002002|001002004|1000000|DORMANT|KENNY

Figure 12: PDF File Display



```

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265689, contrid=5000000, ttoacct=10122326, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=10000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265699, contrid=10000000, ttoacct=10234568, tdate=Wed Jan 16 23:12:01 EAT 2013, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265699, contrid=10000000, ttoacct=10234568, tdate=Wed Jan 16 23:12:01 EAT 2013, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=101223224, contrid=5000000, ttoacct=102335655, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=101223224, contrid=5000000, ttoacct=102335655, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265689, contrid=5000000, ttoacct=10122326, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=10000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265689, contrid=5000000, ttoacct=10122326, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=10000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265699, contrid=10000000, ttoacct=10234568, tdate=Wed Jan 16 23:12:01 EAT 2013, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265699, contrid=10000000, ttoacct=10234568, tdate=Wed Jan 16 23:12:01 EAT 2013, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=101223224, contrid=5000000, ttoacct=102335655, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=101223224, contrid=5000000, ttoacct=102335655, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=20000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265689, contrid=5000000, ttoacct=10122326, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=10000000}

2013/10/31 16:39:32:{Taskid=PENDING, tfroacct=103265689, contrid=5000000, ttoacct=10122326, tdate=Thu Jan 16 23:12:01 EAT 2003, wkplanid=NA, tamount=10000000}
```

Figure 13: System Logs View

5 CHAPTER FIVE: Conclusion

Crime investigations are an integral part of any law enforcement agency and if mismanaged can effect negatively on both government and the populace.By deploying techniques such as Complex Event Processing, Crime Investigation Systems can perform better and faster in analyzing data and discovering hidden patterns.Such tools will enable speedy investigations,overcome the labour shortage issue at the force and also improve public faith in the agencies and government at large.

5.1 Challenges

The following challenges were faced during developement;

(i) Enabling the realtime mode given the complex architecture.

(ii) Representing crime data in a structured format to feed the engine.

(iii) Learning the Esper engine API which was time-consuming.

(iv) Failure to interview some top Police officials who claimed to be busy.

5.2 Contribution

(i) To computing, using CEP techniques and the esper engine to solve a problem in crime investigations which proved more effective than data mining techniques.

(ii) To citizens, enabling quick crime anlaysis leading to a good turn around time as well as managing small work force using technological initiatives.

(iii) To government and the Police,public confidence will be restored.

5.3 Recommendations

The following recommendations were made ;

(i) Develop a well organized database to supply the tool engine with data and deploy it on a country-wide Police network.

(ii) Agree and create case file numbers unique to every Police Station in the country.

(iii) Agree and arrange channels linking the system with other stakeholders' systems to provide straight-through communication e.g to the judiciary.

(iv) The Uganda Police needs to train investigators in computer-enabled investigation and equip them with neccesary ICT skills.

5.4 Future Work

To improve the performance and usefullness of the tool, the following are suggested ;

(i) Extending the tool to detect new emerging patterns that can help in crime prevention based on available records.

(ii) Include a criminal monitoring module for continuous surveillance of criminals/areas based on known patterns.

(iii) Extend the client module to support mobiles to enable remote requests.

References

[1] K. Ask. Criminal investigation:motivation, emotion and cognition in the processing of evidence, 2006.

[2] F. Bex, H. Prakken, B. Verheij, and G. Vreeswijk. Sense-making software for crime investigation: how to combine stories and arguments, 2007.

[3] H.-L. Bui. Survey and comparison of event query languages using practical examples, 2009.

[4] I. Charles M. Alifano, Worldwide Law Enforcement Consulting Group. Fundamentals of criminal investigation, 2006.

[5] Codehaus. Esper reference tutorial, 2013.

[6] O. coporation. The java ee 6tutorial, 2011.

[7] P. Dekkers. Complex event processing-sl-securenet: Intelligent policing using data mining techniques, 2007.

[8] N. Dindar, B. Güç, P. Lau, A. Ozal, M. Soner, and N. Tatbul. Dejavu: declarative pattern matching over live and archived streams of events. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 1023–1026, New York, NY, USA, 2009. ACM.

[9] A. A. Kulkarni. Arcade - abstraction and realization of complex event scenarios using dynamic rule creation. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, DEBS '11, pages 23–28, New York, NY, USA, 2011. ACM.

[10] D. Luckham. A short history of complex event processing1 part 3: the formative years.

[11] D. Luckham. Complex event processing in financial services, 2010.

[12] D. Luckham. complex or simple event processing, 2010.

[13] S. M.A.P. Chamikara, Y.P.R. D. Yapa and J. Gunathilake. Sl-securenet: Intelligent policing using data mining techniques. In *International Journal of Soft Computing and Engineering (IJSCE)*, 2231-2307, Volume-2, Issue-1, 2012.

[14] S. Myles. Criminal investigation case management., 2000.

[15] H. Obweger, J. Schiefer, M. Suntinger, F. Breier, and R. Thullner. Complex event processing off the shelf.

[16] D. O. P. Okwangale Fredrick R. Survey of data mining methods for crime analysis and visualization, 2010.

[17] PrimeTek. Prime faces users guide 3.5, 2013.

[18] J. Schroeder, J. J. Xu, H. Chen, and M. Chau. Automated criminal link analysis based on domain knowledge. *JASIST*, 58(6):842–855, 2007.

[19] N. P. Schultz-Møller, M. Migliavacca, and P. Pietzuch. Distributed complex event processing with query rewriting. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 4:1–4:12, New York, NY, USA, 2009. ACM.

[20] F. Technologies. Accelerating complex event processing with memory- centric database (mcdb), 2008.

[21] F. Tushabe. Computer forensics for cyberspace crimes.

[22] U.Police. Annual crime and road safety report 2010., 2010.

[23] K. H. Vellani. Crime analysis for problem solving security professionals in 25 small step.

[24] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *Proceedings of the second international conference on Distributed event-based systems*, DEBS '08, pages 253–264, New York, NY, USA, 2008. ACM.

[25] C. Westphal. Anatomy of a financial crime.

[26] J. Xu and H. Chen. Criminal network analysis and visualization. *Commun. ACM*, 48(6):100–107, June 2005.