Road Operations Orchestration Enhanced with Long-short-term Memory and Machine Learning (Position Paper)

Fuji Foo[1], Poh Ju Peng[1], Robert Kuo-Chung Lin[1] and Wenwey Hseush[2]
[1]Certis Group, Singapore
[2]BigObject, Taiwan

Keywords: Complex Event Processing, In-memory Computing, Working Memory.

Abstract: Road traffic management has been a priority for urban city planners to mitigate urban traffic congestion. In 2018, the economic impact to US due to lost productivity of workers sitting in traffic, increased cost of transporting goods through congested areas, and all of that wasted fuel amounted to US$87 billion, an average of US$1,348 per driver. In land scare Singapore, congestion not only translates to economic impact, but also strain to the infrastructure and city land use. While techniques for traffic prediction have existed for many years, the research effort has mainly been focused on traffic prediction. The downstream impact on how city administration should predict and react to incidents and/or events has not been widely discussed. In this paper, we propose Artificial Intelligence enabled Complex Event Processing to only identify and predict incidents, but also to enable a swift response through effective deployment of critical resources to ensure well-coordinated recovery action before any incident develop into crisis.

1 INTRODUCTION

As a practitioner in Security and Enforcement industry, situational awareness is critical to our daily operations. The perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of the future state enables us to make informed decision across a broad range of situations, ranging from integrated security services for commercial buildings, aviation security, to critical infrastructure protection.

The proliferation of IoT sensors and smart devices has enabled us with unprecedented amount of data and we have applied Complex Event Processing (CEP) aiming to identify meaningful events in real time and respond accordingly with right actions in a control and secured way. It often works well in the beginning when scenarios are not that complex and event patterns, either in normal or anomalous situations, are predictable and easy to plan with in advance. Along with the development of the applications, however, the existing CEP solutions may become inadequate to support the new business requirements for the applications.

AI-enabled CEP is critical to detecting events patterns and predicting trends or security threats in real time for the road conditions. Two key issues that prevent road operation orchestration from being well managed as planned are (1) concept drift (Tsymbal, 2004) and (2) impractical labelling in real time. Concept drift shows that the statistical properties of the target variables evolve over time in an unanticipated way. This leads to a situation where the predictions become less accurate as time passes. Even concept drift is detectable (Gama, *et al*., 2004; Gama, *et al*., 2014) when it occurs, labelling is either impractical in real time due to intensive labour requirement or infeasible in reality for lack of observable classification evidence.

Once a noteworthy sign or a meaningful event is identified by an AI-enabled CEP, the successive matter is to respond to a sequence of analytic questions instinctively or interactively in order to grasp the vital figures behind the detected patterns. It is often necessary to recall the relevant events from the past for reasoning and tracking down the cause of a threat. Such a business requirement with capacity of reasoning based on logic and evidence suggests that recent events (i.e., hot data) as well as historical events (i.e., cold data) must be properly stored and well managed in an in-memory database, ready to answer ad-hoc queries in a near real-time manner.

The idea of this work is to resolve the problems arisen from road operation orchestration by developing a framework that supports operators, managers and planners to achieve the following two objectives:

1. Planned CEP: Support predictive complex event analysis in real time.
2. Unplanned CEP: Support ad-hoc complex event analysis and reasoning in near real time.

Planned CEP defines the scope of work that can be planned in advance, either by machine learning or programming. Unplanned CEP defined the work dealing with unknown or unexpected situations, which may require human interaction or an adaptive approach to explore and learn potential solutions.

In order to address the needs for both planned CEP and unplanned CEP, we adopt a model that mimics short-term memory and long-term memory in human memory system to maintain recent events and historical events in timely order in two separate stores.

The short-term memory store manages an efficient data structure of recent events within a fixed time period *t* (e.g., say 24 hours). Any event that lives longer than *t* will be immediately removed from the store and eventually moved into the long-term memory store, where events are organized, stored for future reference and intermittently trained for future inference.

The short-term memory store keeps and manages events in two aspects: time and space (i.e., temporal-spatial events). The store is not only aware of where and when events happened, but also capable of measuring and understanding all the geographic and temporal relations (e.g., distance, range) among events in the store.

While events are directed from the short-term memory into the long-term memory store, they can be configured to run through a learning processor (i.e., trainer in machine learning). During the process, behavioural patterns with respect to certain contexts are constantly learned and the trained models are then stored as parts of the long-term memory for future use. This approach is critical to future applications such as a large group of patrol robots autonomously and co-ordinately working in an airport or a shopping mall for service and security purposes.

The long-term memory store keeps two types of data, (a) entire event history and (b) trained patterns or trained models. This store is analogous to recognition memory, a subcategory of human long-term explicit memory, with two distinct processes, *recollection* and *familiarity*, sometimes referred to as *"remembering"* and *"knowing"* (Medina, 2008), respectively. Recollection is the retrieval of details of events. In contrast, familiarity is the classification that the events were previously learned, without recollection of individual events. Recollection is a slow, controlled search process in the memory store, whereas familiarity is a fast, automatic process to identify meaningful situations or predict drifting trends.

With long-term memory store, the planned CEP is the process of familiarity supported by ML-trained models stored in the long-term memory store, while the unplanned CEP is the process of recollection supported by complex queries in database.

With short-term memory store, the planned CEP can be implemented by SQL-like declarative language together with geographic functions and temporal functions to identify certain event patterns. The unplanned CEP is supported with ad-hoc analytic queries and a codeless data visualization tool for exploring the unknown space of events.

2 BACKGROUND

We have applied Complex Event Processing in many applications on the fields, where multiple real-world event streams are captured and analysed in real time in order to respond to threats or opportunities, and further foresee potential trends or developing drifts.

Learning from many business scenarios, we have compiled a classification of four CEP approaches as shown in Figure 1. Our intention is to develop a CEP framework and architecture that meets the various business requirements.

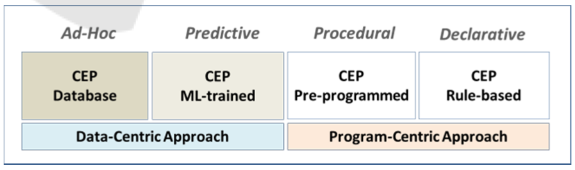


Figure 1: Four CEP Approaches.

1. Rule-based approach: A set of rules is pre-defined to process the incoming events. The CEP approach based on declarative rules (Anicic *et al*., 2010) shows expressive enough to describe various complex event patterns. In term of programming complexity, code in declarative and logic programming is often significantly simpler and smaller than that in procedural programming. The complex event language proposed in (Wu, Diao & Rizvi, 2006) allows

query rules to filter and associate events and transform the relevant events into higher-level and meaningful events for comprehension.

2. Pre-programmed approach: The logic of CEP is written in a programming framework using a procedural language to process incoming event streams. Several Examples include Apache Storm, Apache Flink and Apache Spark Streaming, etc. Apache Storm (Iqbal & Soomro, 2015) is a distributed real-time computational system, where an application is designed as a "topology", a directed acyclic graph (DAG) with spouts and bolts as vertices. Directed edges on the graph are data streams from one node to another. Together, a DAG serves as a data transformation pipeline. Apache Flink (Carbone *et al*., 2015). is a framework and distributed processing engine for stateful computations over data streams. Flink is a universal dataflow engine designed to perform both streaming and batch analytics. The framework supports a CEP API for Java and Scala. Apache Spark Streaming (Zaharia *et al*., 2016) uses Spark Core's fast scheduling capability to perform streaming analytics. It processes data in mini-batches and performs transformations on those mini-batches of data.
3. Pre-trained approach: Incoming events are classified by a pre-trained ML (Machine-Learning) model. It can serve as a predictive analytics (Fülöp *et al*., 2012) where events are prevented before they occur. However, concept drifting (Li, Wu, & Hu, 2010; Wu, Li, & Hu, 2012) is a key challenge for many existing systems, where models need to be re-trained every time when a new concept happens. Most methods used for drift detection assume the availability of class labels immediately after a data sample arrives. Nevertheless, it is unrealistic (Kim & Park, 2017) to assume acquisition of all labels when processing data streams, as labelling costs are high.

4. Database approach: Incoming events are captured and analyzed while referencing to the recent or historical events stored previously in a database. The system designed by (Gyllstrom *et al*., 2006) contains a MySQL database to support querying over historical data and to allow query results from the stream processor to be joined with pre-loaded data. DejaVu system, using MySQL database as well (Dindar, Fischer, & Tatbul, 2011), provides declarative pattern matching query language over current and archived event streams and offers new capabilities such as identifying causal relationships among complex events across multiple time scales. This approach includes SQLStream and ParStream. SQLstream is a SQL-based, real-time analytics platform for streaming data. ParStream (Hummel, 2010) is positioned for real-time analytics in the area of IoT streaming data.

The first two approaches are program-centric, where pre-defined programs or rules are used to process incoming events, current or last few ones, without maintaining the history of events in storage. The last two are data-centric approaches, where histories of events are captured and maintained in storage. ML-trained CEP is the AI-enabled (machine learning) approach that labels and trains models with the stored event data in advance, and predictively analyses incoming events in runtime. The database CEP is an interactive analytics used on an ad hoc basis. The database must be a time-series relational database system that is capable of supporting complex queries in an efficient way.

We are developing a flexible CEP framework, which supports not only the ML-trained approach, but also ad-hoc CEP and pre-programmed CEP approaches since business requirements are changing frequently and any pre-trained or pre-programmed approaches may fail to respond to an incident due to concept drift.

3 PROBLEMS AND ASSUMPTIONS

In Security and Enforcement industry, we face different challenges in many data-related applications on the fields, where multiple real-world data streams are captured and collected for analysis. Most data are considered as streaming data, which has three properties:

1. Non-stop data streams
2. Immutable data (fact data)
3. Hot value, where more recent data contains more business values

Streaming data is live fact data, which is always associated with time and a time-series database is needed to maintain immediate, recent and historical data in timely order. Example data sources in road traffic monitoring are CCTV(closed-circuit television camera), Lidar(device that measures distance and amplitude to an object), etc.

There are some problems and challenges we are facing on the fields as follows:

- Multi-sensory detection relating the real-time occurrences of different events such as video analytics from CCTV, seismic sensors and radar.
- Unusual situation detection against experience : detect a situation where the average speed of vehicles in a location for the last 10 minutes is 20 miles slower than that in the same location same time segment for the last 10 weeks.
  - Autonomous robot patrolling: identifying, detecting and differentiating between unattended objects, temporary and permanent fixtures in the course of security rounds.

To design a powerful CEP system for the problems in Security and Enforcement industry, some assumptions are as follows:

- Labeling is difficult or impractical.
- Periodicity by week, month or year is assumed.
- Public places where group behaviors are statistically similar in usual situations.

3.1 CEP Issues

Roth et al., 2010 suggested a list of observations for existing CEP systems in the following areas:

1. temporary memory (i.e., no persistent state),
2. no sophisticated analytical data processing such as forecasting, classification, clustering, etc.,
3. limited time window and aggregation, and
4. poor performance when calculating exact answers for complex aggregate queries with huge windows.

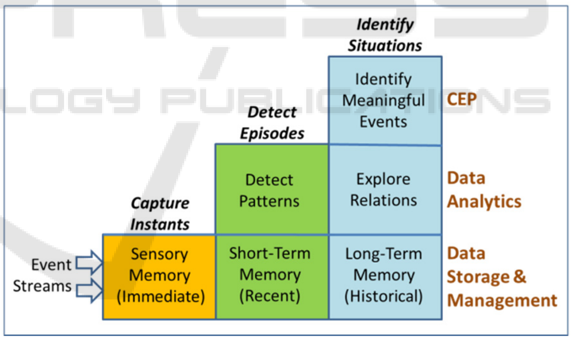
Since 2010, new data technologies, AI and analytics tools have help to resolve and improve these issues. Nowadays, our industrial experience shows several areas waiting to be addressed.

1. Ad hoc and interactive streaming analytics is still difficult. CEP is often planned in advance. Programs or rules must be pre-defined, and models needs to be trained in advance.
2. Traditional relational databases used in CEP do not support large-volume data processing and usually perform poorly for long-term historical data.
3. An efficient time-series relational database is expected to process hundreds of millions of records on a commodity server in seconds. An in-memory database can be used for this purpose.
4. A framework combining different CEP approaches is important.
5. Applications in Security industry often require the features related to computation of complex aggregates with huge windows.

4 PROPOSED CEP FRAMEWORK

A CEP framework is designed to support both planned and unplanned CEP approaches. It consists of three layers over three types of memory stores as in Figure 2:

1. CEP: define the approach for CEP, either database CEP or ML-trained CEP.
2. Analytics: define analytics on short-term memory, which detects patterns of recent events, or analytics over long-term memory store, which predicts situations or explore event behaviours.
3. Database: define sensory memory store, short-term memory store or long-term memory store. Sensory memory is a memory buffer that only keeps events instantly for on-the-fly streaming process, without maintaining internal persistent state. Short-term memory store is an in-memory database that only keeps recent events, which expire once staying longer than a pre-defined time period and will be removed from the database immediately. Long-term memory store manages two types of data, (a) entire history of events in details and (b) trained patterns or models.



Identify Situations
Detect Episodes
Capture Instants
Identify Meaningful Events
CEP
Detect Patterns
Explore Relations
Data Analytics
Event Streams
Sensory Memory (Immediate)
Short-Term Memory (Recent)
Long-Term Memory (Historical)
Data Storage & Management

Figure 2: CEP Framework.

The system architecture that supports the CEP framework is shown in Figure 3. This is a reliable system architecture that consists of 6 modules:

1. Streaming data queue: To capture the incoming events persistently before forwarding to the streaming process. This module is critical to reliable data collection while event streams flow in non-stop. Check points are used to ensure recoverability if case of system failure.
2. On-the-fly streaming process: The first stop to process the current event or the last few events without keep additional state information. For example, check if the average speed of the last 5 vehicles is less than 5 KM per hour.

3. Short-term memory database: An in-memory relational database to keep recent events. Queries in SQL are processed for multi-dimensional analysis.
4. Long-term memory database: An in-memory data warehouse to keep the history of events.
5. ML trainer: Events stored in the short-term memory database will be retrieved for labelling and training.
6. Predictor: The AI-enabled process to predict situation.

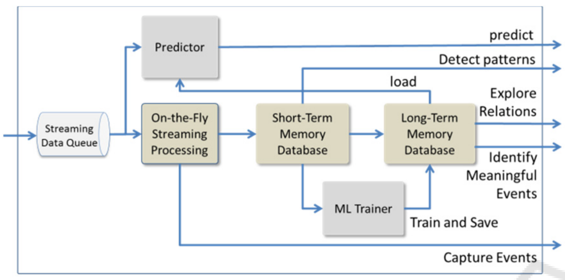


Figure 3: CEP Architecture.

5 ROAD TRAFFIC MONITORING

We have piloted a road traffic monitoring system with CEP in various levels. The objective of the system is to monitor traffic for major highways. The business requirements for road traffic monitoring are listed as follows:

- A swift response and coordinated recovery action before any incident develops into a crisis;
- An effective oversight on real-time traffic flow condition on roadways; and
  - Sensors to measure traffic demand for trending, analysis purposes for deployment of critical resources.

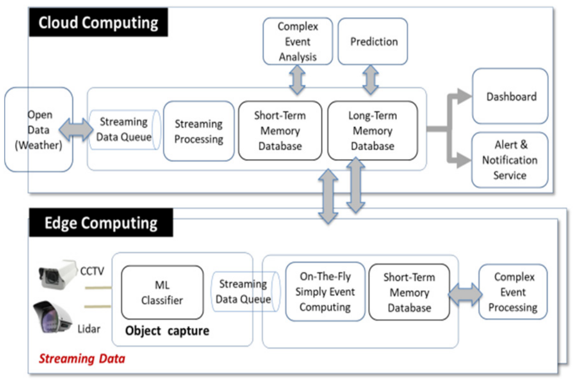


Figure 4: Distributed CEP Architecture for Road Traffic Monitoring.

Through a multi-faceted approach of applying a range of ground sensors and collecting real-time events, we were able to anticipate road traffic conditions as well as detect traffic incidents and violations. The pre-emptive approach enables business continuity through better roadway management with speeder responses and deployment of road marshals and vehicle recovery to handle traffic incidents that could paralyse downstream operations.

The Road Traffic Monitoring CEP is deployed in a distributed environment, where a cluster server is located in the cloud and tens of edge nodes are located in local edge devices, as shown in Figure 4. Edge node is equipped with (a) a machine-learning classifier (i.e., the predictor) to perform image-processed object detection tasks, (b) an on-the-fly processor for simple event computing and (3) a short-term memory database. All local events are forwarded to the cluster server, which has an additional module, long-term memory database. Besides the events collected from edge nodes, the master node in the cloud also collects weather information for prediction purpose.

Figure 5 shows a hierarchy of five analytics tasks in the CEP system as the following (from bottom level to the top level):

1. L1 (edge): Extract vehicle information (tracking ID, location) for each frame from CCTV.
2. L2 (edge): Extract more information such as speed about vehicles from multiple frames.
3. L3 (edge): Anomaly detection for stationary vehicles.
4. L4 (cloud): Prediction for the traffic condition for the next 30 minutes.
5. L5 (cloud): Ad Hoc CEP for interactive analytics and dashboard.

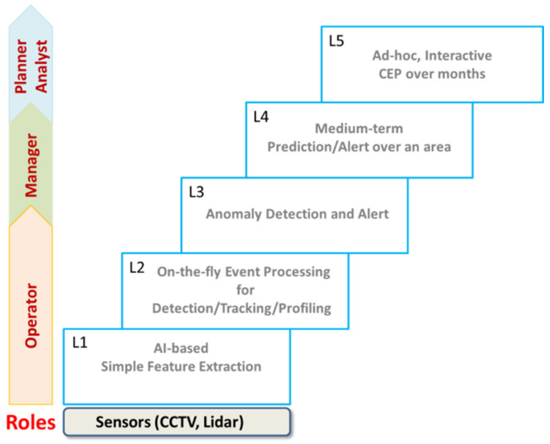

Planner Analyst
Manager
Operator
Roles
L5
Ad-hoc, Interactive CEP over months
L4
Medium-term Prediction/Alert over an area
L3
Anomaly Detection and Alert
L2
On-the-fly Event Processing for Detection/Tracking/Profiling
L1
AI-based Simple Feature Extraction
Sensors (CCTV, Lidar)

Figure 5: Analytics Hierarchy for Road Traffic Monitoring.

6 CONCLUSIONS

To keep pace with an increasingly complex security landscape, security and enforcement practitioners are leveraging on information dominance as the force multiplier. In order to achieve information dominance to Deter, Detect, Deny, Delay and Defend, the ability to identify an event, associate with a short-term memory and capture in long-term memory for referencing is critical.

Our work mentioned in this paper lays the foundation for a highly flexible event processing platform with interactive and predictive analytics to support the dynamic security and enforcement industry where requirements are changing rapidly, and any pre-trained or pre-programmed approaches may fail to respond to an incident. The long-short-term memory provides the context to CEP in solving sequence and time series related problems.

The potential application of the work can be extended to unmanned protection of critical infrastructure through IoT sensors and smart devices, centralized management of digital twins through the convergence of physical and digital environments and orchestration of autonomous digital workforce such as robots and drones. The exponential growth of data from the sensors network and events requires a more systematic and holistic approach to sense, analyse, respond and learn.

REFERENCES

Anicic, D., Fodor, P., Rudolph, S., Stühmer, R., Stojanovic, N., & Studer, R. (2010). A rule-based language for complex event processing and reasoning. In *International Conference on Web Reasoning and Rule Systems* (pp. 42-57). Springer, Berlin, Heidelberg.

Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* 36(4).

Dindar, N., Fischer, P. M., & Tatbul, N. (2011). DejaVu: a complex event processing system for pattern matching over live and historical data streams. In *Proceedings of the 5th ACM international conference on Distributed event-based system* (pp. 399-400). ACM.

Fülöp, L. J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., & Farkas, L. (2012). Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics. In *Proceedings of the Fifth Balkan Conference in Informatics* (pp. 26-31). ACM.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Brazilian symposium on artificial intelligence* (pp. 286-295). Springer, Berlin, Heidelberg.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys* (CSUR), 46(4), 44.

Gyllstrom, D., Wu, E., Chae, H. J., Diao, Y., Stahlberg, P., & Anderson, G. (2006). SASE: Complex event processing over streams. arXiv preprint cs/0612128.

Hummel, M. (2010). ParStream–a parallel database on GPUs. In *GPU Technology Conference*, San Jose Convention Center, CA.

Iqbal, M. H., & Soomro, T. R. (2015). Big data analysis: Apache storm perspective. *International journal of computer trends and technology*, 19(1), 9-14.

Ji, Y. (2013). Database support for processing complex aggregate queries over data streams. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops* (pp. 31-37). ACM.

Kim, Y., & Park, C. H. (2017). An efficient concept drift detection method for streaming data under limited labeling. *IEICE Transactions on Information and systems*, 100(10), 2537-2546.

Li, P., Wu, X., & Hu, X. (2010). Mining recurring concept drifts with limited labeled streaming data. In *Proceedings of 2nd Asian Conference on Machine Learning* (pp. 241-252).

Margara, A., Cugola, G., & Tamburrelli, G. (2014, May). Learning from the past: automated rule generation for complex event processing. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (pp. 47-58). ACM.

Medina, J. J. (2008). The biology of recognition memory. *Psychiatric Times*, 25(7), 13-15.

Roth, H., Schiefer, J., Obweger, H., & Rozsnyai, S. (2010, May). Event data warehousing for complex event processing. In *2010 Fourth International Conference on Research Challenges in Information Science* (RCIS) (pp. 203-212). IEEE.

Tsymbal, A. (2004). The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2), 58.

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

Wu, X., Li, P., & Hu, X. (2012). Learning from concept drifting data streams with unlabelled data. *Neurocomputing*, 92, 145-155.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., & Ghodsi, A. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11), 56-65.