An Application of Sensor and Streaming Analytics to Oil Production

Krishnamurthy Viswanathan[1], Chetan Gupta[1], Choudur Lakshminarayan[1]
Ming Hao[1], Umeshwar Dayal[1],

Ravigopal Vennelakanti[2], Paul Helm[2], Sumitha Rangaiah[2],
Harikrishnam-Raju Sagiraju[2], Sunil Doddmani[2]

[1]Hewlett Packard Labs, [2] Hewlett Packard Enterprise Business

{firstname.lastname@hp.com}

Abstract

At HP Labs, we are building "Live Operational Intelligence (Live OI) System" – a system that ingests streams of operational data generated by multiple sources such as sensors and operational logs, and provides the operational staff real time insights in terms of suggested actions, event correlations, predictions, root cause analysis and visualization. In a Live OI framework some models are learnt offline and then deployed online, and some models are learnt online. Live OI system also supports querying of historical data to find past occurrences of patterns and suggested actions, and a dashboard for humans to monitor and interact with the operational system. This paper describes the highlights of the Live OI system as applied to monitoring oil production operation, through the discussion of use cases.

1. Introduction

Live Operational Intelligence (Live OI) is a framework for developing, deploying, and executing applications that mine and analyze large amounts of data collected from multiple data sources such as sensors and operational logs, in order to help operations staff take informed decisions during management of operations. It has applications in many industries such as hydrocarbon production and exploration, transportation, smart grid, etc. In this paper, we focus on use cases from oil and gas production.

The oil and gas industry heavily relies on sensor data to capture and record system parameters for trending analysis and alert processing (typically using threshold

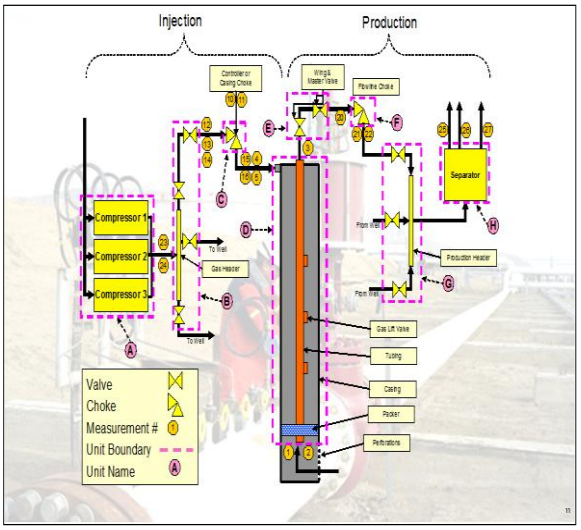

Figure 1: Well and Sensor Schematic

based approaches). During production, temperature, pressure and flow rate are monitored at various points in the well with the well head having the greatest collection of sensors (See Figure 1). However, this large amount of data is currently not being used to provide real time insights to the operations staff manning the operations center.

An example problem that requires a Live OI like framework is that of detection of onset of *unpredictable*

International Conference on Management of Data
COMAD 2011, Bangalore, India, December 19--21, 2011

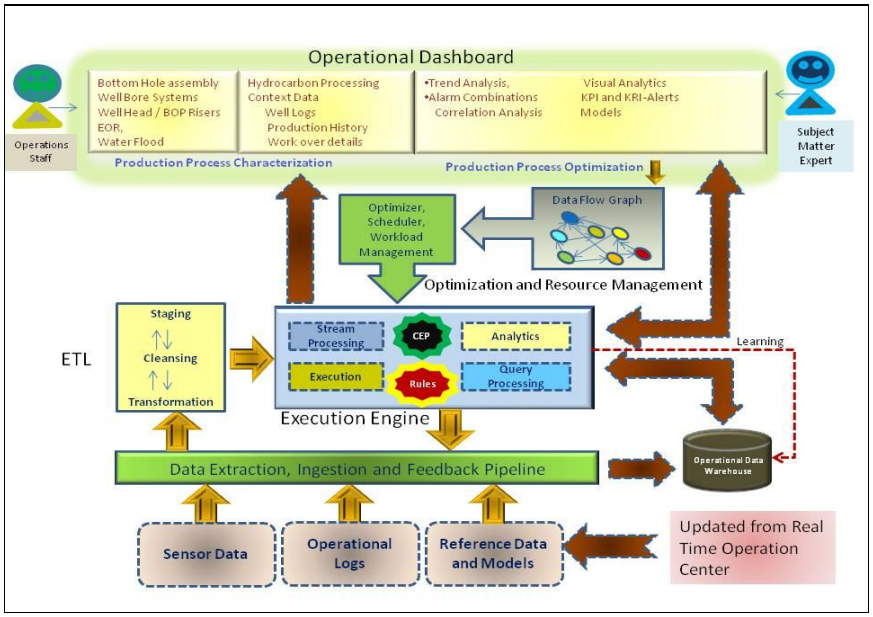

Figure 2: Reference Architecture for LOI

flow rates. In hydrocarbon production, there is a direct correlation between flow rate and revenue generated by a well. Though high flow rates are generally preferable, steady consistent flow is of equal importance. Highly variable flow results in unpredictable conditions, both physically and economically and investigation shows that a high variable flow rate can lead to a reduction in overall productivity of up to 5%. Onset of such unpredictable flow rates cannot be done with simple threshold based methods, but rather it requires more sophisticated models. Furthermore, once the onset has been detected, it is useful to help the operator respond in the correct fashion based on the type of disturbance in the flow. To address problems such as these, a Live OI system allows for building and deploying sophisticated models, as well suggesting actions to the operator.

Developing Live OI application that addresses problems such as these, presents many challenges. First, data of many disparate types – structured and unstructured, streaming and historical – has to be integrated, managed, and analyzed. Today, these different types of data are typically managed and processed by separate systems, each with its own idiosyncratic application interfaces and development environment and there is a lack of algorithms and techniques that span different types of data. Second, data from different sources has to be combined and aligned. For instance, to build a rational analytics solution sometime requires combining textual and time series data to obtain a data set in the appropriate time sequence. Third, since data is collected from multiple processes, cross-process analysis is only possible if the variability induced due to differences in calibration, data collection procedures, sampling rates, and terminology are properly comprehended and adjusted. In addition, events such as sensor malfunctions, equipment failures, missing data, bogus values, and a myriad others, impose challenges. Fourth, in addition to automated data analysis, the application must incorporate knowledge from human experts.

Live Operational Intelligence (Live OI) system consists of many components. Figure 2 describes an outline of the multiple components of the Live OI system layered in a multi-level configuration. The bottom layer is the ETL layer, where the system assimilates data from multiple sources (sensors, operational logs, historical repositories, and others), and subjects the data to standard ETL (Extraction, Transformation, and Loading) operations. Besides storing the transformed data, the Live OI system provides functionality for stream processing of this transformed data. In other words, Live OI allows for analytics over both stored and streaming data.

The Live OI supports analytical operators that broadly fall into the following functional categories: Multivariate time series analysis, pattern matching and discovery, anomaly detection, correlation of events, prediction over time series data and event streams, Bayesian causal models for diagnostics and root cause analysis over event streams and Visual analytics. The Live OI application provides a plug-and-play approach to analytical operators where different operators can be composed into an operator flow. In order to facilitate the multiplicity of operations, an optimizer module is incorporated for scheduling and managing workloads efficiently.

In the top layer, a dashboard is provided to display streaming data and results from analytics through a set of sophisticated visualization tools. The dashboard also

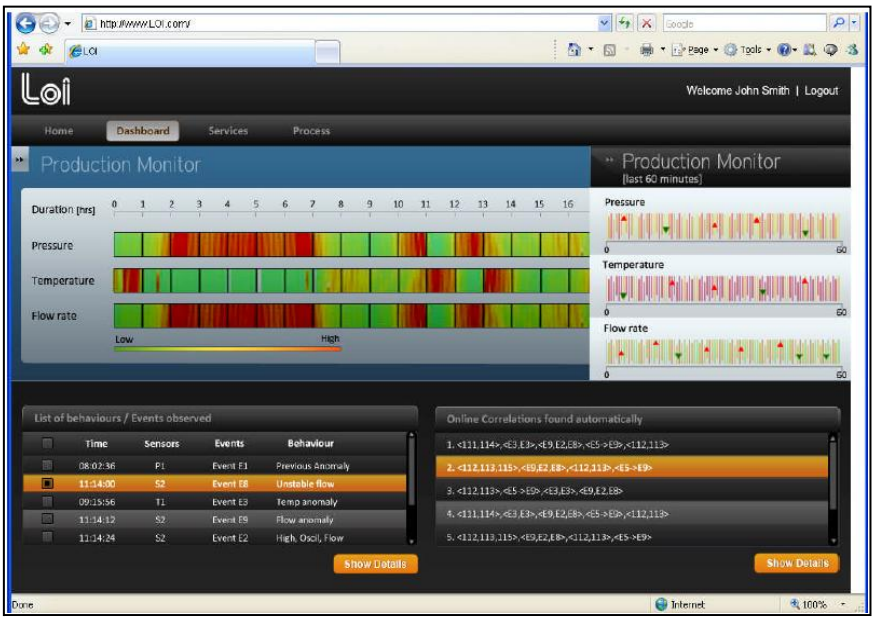

Figure 3: Screenshot for LOI Dashboard

provides an interactive drop and drill-down menu to assist the on-line operator in querying the historical database for pattern analysis, matching, retrieving precedent anomalous events sequences, and evaluating complex event correlations in order to take corrective actions proactively.

In this paper, instead of focusing on the details of the architectural components of Live OI [1], we focus on the oil production use case and the analytic components needed for live operations management for oil production. In the following sections, we will present use cases that will demonstrate the functionality of the Live OI system in order to assist in the detection, prediction, and analysis of complex events over streaming data in oil production.

2. Use Cases

We present use cases that arise when monitoring an oil well during production. As discussed earlier, several sensors are placed at different points in the well (Figure 1). For example, some of the sensors are located at the well-head and others at the bottom of the well. These sensors measure different physical parameters such as the pressure, temperature and flow rate. In a Live OI system, the measurements of these sensors are monitored through a dashboard which shows the result of algorithms aimed at detecting interesting events/correlations and advising the user on future actions based on the detected events/correlations. The dashboard also contains visual analytic tools for visual inspection of time-series data. The dashboard and some of the functionality is shown in Figure 3, where at the top left there is a representation of three important variables in Visual time series [7], which allows for succinct representation of large amount of historical time series data. In the top left, we show the current values of time series, with indicators for events detected on them. These events are of many types, for example, in the context of oil production, there are event detectors for oscillation, outliers, etc. Bottom left is the list of events (in some detail) in the current window and bottom right is a panel showing the online correlation discovered on these events.

The system monitors the measurements recorded by various sensors placed at different locations in the well. The measurements are inputs to algorithms that aim to detect specific events related to oil production in the data. As mentioned before, the system's dashboard has an *event panel* that displays the events being detected. The system first detects events that are local to each individual time series, such as an anomaly in the pressure measurement. These are *primitive events.* As soon as a primitive event is detected by the system, it is recorded in an *event log table* that maintains the time of occurrence of the event, the event type and the location of the time series in the well. The system also has the ability to detect *complex events* which may be a sequence or series of primitive events that occur within a short time window. These events are detected based on the entries into the event log table. The system also automatically detects whether the events detected in the recent past co-occur frequently with other events. This *co-occurrence* is defined in terms of time as well as location within the well. Any discovered co-occurrences or correlations are displayed in another panel as mentioned earlier.

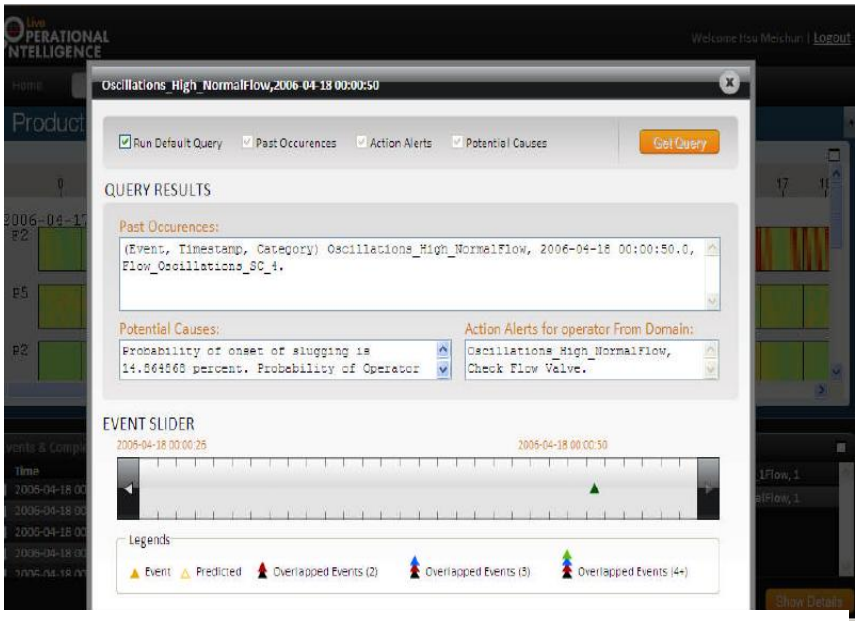

Figure 4: Screenshot Capturing Query Results in LOI Dashboard

The user is permitted to select one or more of the events observed in the event panel and issue queries regarding the events (Refer to Figure 4). The queries may include one or more of the following: past occurrences of the event (s), potential causes of the events (s), suggested action for the event (s) and the set of event (s) that might occur given that this event (s) has occurred.

Past occurrences are a table look up. The suggested actions are generated from an action table that maintains a recommended action for certain events. If no such recommended action is available, the system searches the action log, which records the actions taken and the time of the action, for actions taken when the event occurred in the past and suggests the most common ones. The user can request the system to predict events that might happen within a fixed time window beginning at the time the query was issued. This prediction is made by the system based on the entries in the event log table by computing the conditional probability of observing each event given the events observed when the query is issued.

The user can also dig deeper to unearth any serious potential causes for the events that are being observed. Potential causes are either learned or suggested by domain experts and stored in a table for quick look-up. The learning is performed using probabilistic inference based on a causal model. This model takes into account all the events that have occurred during a time window that ends at the instant the query is issued, as well as the human actions that were taken and computes the conditional probability of several candidate causes based on the observed events. This is output to the user who can use this information to guide future courses of action.

A specific use case that illustrates these functionalities is discussed below. For example, the operator observes a complex event sequence consisting of: <"Extreme Value" followed by "Unstable Process" and "High Frequency Oscillation">. He issues a query on this complex event sequence that notifies him that in the past the complex event in question is often followed by an event "Flow rate out of control". The system computes the probabilities of two candidate causes, first being that an operator tripped a valve during a maintenance event in a time window preceding the complex event and the second being onset of *slugging* (Slugging is a phenomena in the oil and gas production where high amplitude, high frequency oscillations are observed in flow of a fluid). Based on the probable cause, the system also recommends actions that may be taken to address the situation. An example of how the output would appear on the Live OI dashboard is shown in Figure 4.

Another example is where the Live OI system detects three events: instability in temperature time series, pressure characterized by a negative gradient, and oscillation in flow. We may conclude that temperature instability and pressure gradient may be temporally correlated indicating a possible serious deterioration in the oil flow and a warning is issued. The absence of a response from the on-line crew, indeed leads to flow rate *churn* in the well-head. (Churn in the flow rate is a phenomenon characterized by a high frequency oscillation which often results in turbulence that detrimentally affects production).

Besides event detection and correlation, Live OI allows for analysis of multivariate time series. This includes prediction of time series values.

The Live OI system is also designed for the user to describe interesting events and construct detectors for these events. In an online setting, the streaming data is

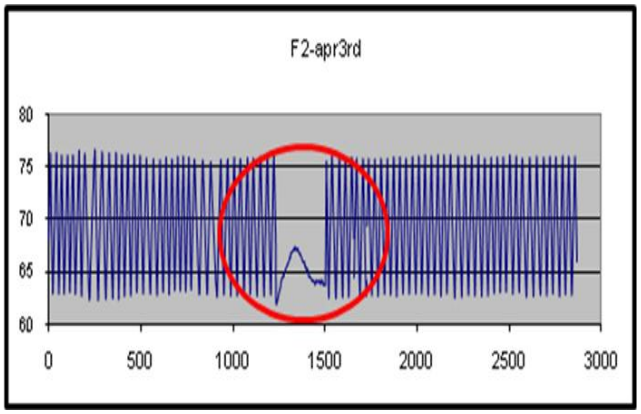

Figure 5: Sample Query for Offline Discovery of Patterns

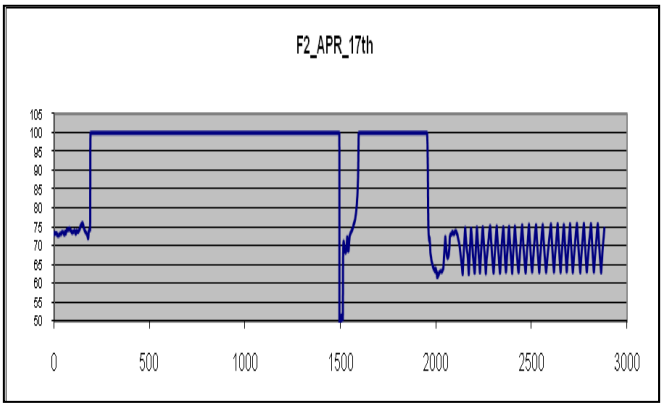

Figure 6: Figure Depicting Different Regimes in Flow

taken as input by the detection algorithms to identify pre-determined patterns of interest. However, as the system is used regularly, we anticipate that the user may observe other interesting patterns visually, analyze their occurrence and might eventually want to add it to the list of events that need to be detected. This functionality is provided in an offline setting. It is meant to be used as a diagnostic tool for identifying new events based on patterns in the data.

As shown in Figure 5, the user might select a segment of one or more time series as a pattern of interest. The system then searches the historical data for all occurrences of segments that are similar to the one selected. The similarity can be measured using any number of distances. The results can be used to diagnose the importance of the pattern. Queries such as event co-occurrence and prediction which were discussed earlier can be run on the results to gauge whether this is a significant pattern. Once that is determined, the pattern can be classified as an event to be monitored. Specialized detectors can be developed and added to the existing ones. Alternately, one can use the raw time segment, a distance measure and a threshold on the distance between the pattern and a candidate segment to determine if a given time segment corresponds to this pattern. The precise threshold can be inferred via offline diagnosis.

3. Algorithmic Approaches

In this section we discuss in some detail the algorithms behind some of the use cases discussed in the previous section. We will begin by discussing multivariate time series methods that are used for detection of primitive events. We then discuss the algorithm for online discovery of correlations between events (we do not discuss the offline discovery of correlations). We then discuss root cause discovery through Bayesian models and finally visual analytic techniques.

3.1. Multivariate time series methods

The analytics engine inside the Live OI system utilizes a variety of algorithms for anomaly detection, pattern detection, and time-series prediction. We discuss these below:

3.1.1. Anomalies

The anomaly detection (outlier detection) relies on traditional threshold methods based on the Gaussian distribution. In order to detect trends and anomalies, we implemented the Western Electric Company (WECO) rules [2]. These rules are based on the Gaussian distribution and determine trend in the time-series by observing sequences of data points within $\pm\sigma$, between $(\pm\sigma, \pm 2\sigma)$ and $(\pm 2\sigma, \pm 3\sigma)$, and $\leq -3\sigma$ and $\geq 3\sigma$. Tracking sequences of observations within ranges of the Gaussian distribution provides the ability to monitor system dynamics and take proactive actions. Point anomalies outside the $\pm 3\sigma$ limits point to excursions and merit investigation from the on-line crew.

3.1.2. Oscillation Detection and Prediction

Typical flow-rates are composed of multiple regimes characterized by normal oscillation (NO), high amplitude oscillation (HAO), and low amplitude oscillation (LAO) (Figure 6). The data is both non-stationary and non-linear. These inherent conditions render fast detection of regime change of non-linear time series difficult in an on-line fashion. The LAO is a structured time series superimposed by stochastic noise. In this case, frequency domain approaches are appropriate for the LAO regime as its unique characteristics can be extracted by spectral features. The HAO regime is characterized by highly periodic structure with very little noise. Deterministic

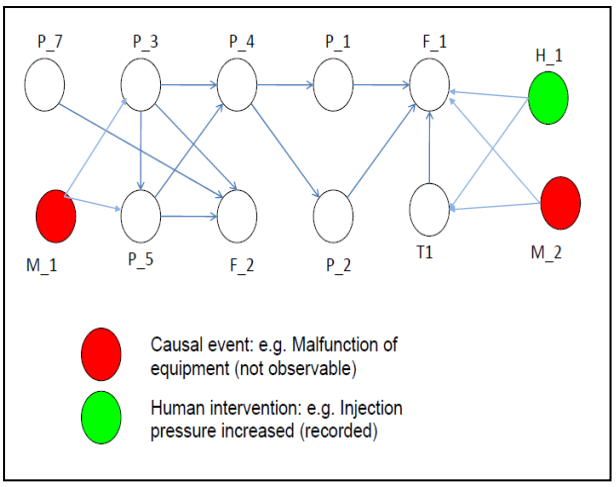

Figure 7: Bayesian Model for Root Cause

methods from non-linear dynamical systems theory are suitable since these provide a model independent representation of the dynamics that generate the time series. The HAO signal which is deterministic with predictable periodicity (Figure 5) can be abstracted by the well organized *attractor* in a multi dimensional phase space [3,4]. Lastly, oil flow rate corresponding to normal oscillation (NO) is a purely stochastic signal and is modelled by an autoregressive (AR) processes from the linear time series theory. AR methods appear to model the series suitably when the flows are in a steady state of normal oscillation.

3.2. Online Event Correlation

The purpose of online event correlation is to find correlation between events flowing into the system in real-time. Event correlation is a widely used term, but for our purpose we say that two events are *correlated* if they are in the same "time-space neighbourhood". For finding neighbours in time sliding windows are maintained and for the purpose of finding neighbours in space we maintain a data structure called *Hierarchical Neighbourhood Tree (HNT)* [5], which allows us to find neighbourhoods at different abstraction level efficiently.

For the purpose of online event correlation we store all the events in a time window in the HNT. Whenever, two or more events occur in the same hierarchical neighbourhood we a correlation event is issued. Each correlation is weighted based on the size of neighbourhood and the frequency of events involved, with the weight being higher for smaller neighbourhoods and lower for events with higher frequencies.

3.3. Bayesian Causal Models

Determining root causes of observed events is a crucial subtask for improving the operations in oil production. We use Bayesian causal networks to perform this root cause analysis. A Bayesian network is a graphical model for representing the joint distribution of a collection of random variables. The graph is a directed acyclic graph, its vertices represent the random variables and its edges the conditional independence information. A Bayesian causal network is a Bayesian network where the direction of the edges further encodes cause-effect relationships amongst the variables (see [6]). Given the value of some of the random variables, it is possible to infer the posterior probability of the causes conditioned on the observed variables. The structure can be obtained with the help of domain experts and the probabilities can be learned from the data.

The model presented in Figure 7 is an example of causal model for the oil and gas production use case. The uncoloured (white) vertices represent events associated with the various quantities being measured. These events are detected by the detectors and the variable corresponding to the event takes the value '1' if the event occurred in the time window under consideration and the value '0' otherwise. The green coloured nodes (light grey) represent actions undertaken, and their values are obtained from event logs. The red coloured (dark grey) nodes represent potential causes. These are unobservable and the goal is to infer their posterior probabilities. This is done by setting the values of the other variables to the observed or recorded values and running an inference algorithm on the causal network.

3.4. Visual Analytics

Visual Analytics combines visualization and analytics techniques. For instance, the pixel time series map shown in Figure 8 allows users to visualize streaming data over several days at once. The figure shows an oil well production pixel cell time series map [7, 8], depicting two phenomena of interest in oil and gas production and mentioned earlier: *slugging* characterized by high frequency high amplitude oscillation in the flow through well head and *churning* characterized by low amplitude-high amplitude oscillation in the flow through the well head.

Using our visualization techniques, we can also compute correlations among time series variables and visualize them: for instance, Figure 8 shows that the flow rate F_2 and pressure P_7 are correlated after a large drop in the flow (grey box to the right).

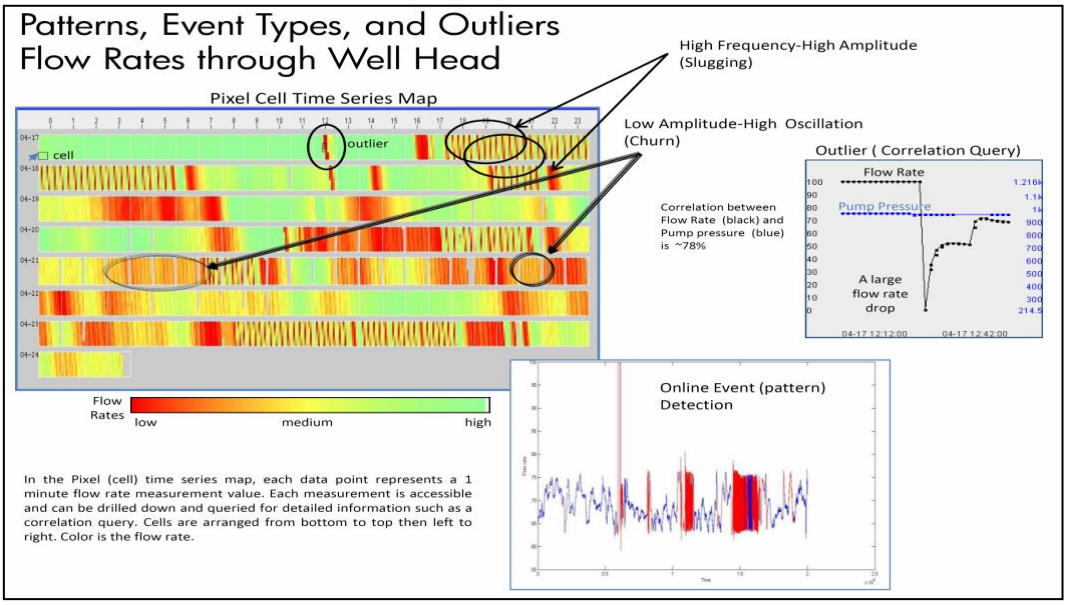

Figure 8: Visual Analytics

4. Related Work

In terms of applying event detection techniques for oil production, there has been some work [9], but unlike our work it is a point solution. There are some data models used in the oil and gas industry: PPDM Data Model[1] – designed to provide an E&P standard model, xML – standards evolving from proprietary data transmission and exchange covering, WITSML – real-time data related to drilling and completions, PRODML – related to production operations, RESML – real-time data related to Reservoir Management, PIDX – includes standards for secure data exchange, transaction documents and business processes. There has been some work in the area of "slugging" [10] but these are not complex event based approaches such as ours. To the best of our knowledge, and from speaking with many practitioners in the oil and gas domain, we believe that there is no other Live OI like comprehensive solution available to the oil and gas industry.

In terms of underlying computational technology, Complex Event Processing (CEP) and Data Stream Management Systems (DSMS) are designed to work with streaming data. CEP [11] engines arose from expert systems and systems research. They are well designed for responding in real time. However, rule-based CEP engines are primarily meant for matching of events and do not contain analytic operators such as outlier detection. DSMS engines [12], [13] can handle complex operations and are designed for fast streams. However, traditional DSMS are designed for continuous SQL query processing and they do not provide the rich analysis functionality needed for oil and gas production applications and there is no direct mechanism for real time action.

Our Live OI engine aims to capture the combined functionality of a DSMS and a CEP engine.

5. Conclusions

We have built a prototype of the Live OI system and implemented use cases from oil production. We have validated the results with domain experts – meaning that the event patterns we discovered are in fact useful indicators of problems in oil production operations. We have also extended the Live OI prototype to monitor drilling operations [1] and are building an instance of Live OI engine for monitoring traffic and predicting congestions.

Besides building solutions for the various industry verticals, we are working on adding new analytic functionality to our engine with new algorithms for time series matching, and event correlation.

References

1. Chetan Gupta et al, "Better Drilling through Sensor Analytics: A Case Study in Live Operations Management," ACM SIGKDD workshop on sensor analytics, 2011, San Diego, USA. To Appear.

[1] http://www.ppdm.org/about-ppdm

2. Douglas C. Montgomery (2005), Introduction to Statistical Quality Control (5 ed.), John Wiley and Sons.
3. Evan Kriminger, Choudur Lakshminarayan, Jose C. Principe. Modified embedding for multi-regime detection in nonstationary streaming data, IEEE, International Conference on Acoustics, Speech, and Statistical signal Processing, 2011, Prague, Czech Republic.
4. Holger Kantz and Thomas Schreiber, Nonlinear Time Series, Analysis, Cambridge University Press, New York, NY, USA, 2003
5. Malu Castellanos, Song Wang, Umesh Dayal, Chetan Gupta. SIE-OBI: a streaming information extraction platform for operational business intelligence, SIGMOD 2010.
6. Judea Pearl. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2000.
7. Hao, M., Dayal, U., Keim, D. A., Schreck, T. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. Proceedings: IEEE VGTC Symposium on Visualization, EuroVis 2007
8. Hao, M., Marwah, M. Dayal. U, Janetzko, H., Keim D., et al, Visualizing Frequent Patterns in Large Multivariate Time Series. Information Visualization VDA11, CA.
9. M. Hill, M. Campbell, Y.-C. Chang, and V. Iyengar, "Event detection in sensor networks for modern oil fields," in DEBS '08: Proceedings of the second international conference on Distributed event-based systems. New York, NY, USA: ACM, 2008, pp. 95–102.
10. F Di Meglio et al., 2006. Reproducing slugging oscillations of a real oil well, IEEE conference on Decision and Control, 2010, 4473 – 4479.
11. R. S. Barga, J. Goldstein, M. H. Ali and M. Hong, Consistent Streaming Through Time: A Vision for Event Stream Processing. CIDR , pp. 363-374. (2007).
12. D. Abadi, et. all. Aurora: A New Model and Architecture for Data Stream Management. VLDB Journal 12(2), 120–139 (2003).
13. Chetan Gupta, et. all. CHAOS. A Data Stream Analysis Architecture for Enterprise Applications, IEEE CEC, pp 33-40, (2009).