Better Drilling Through Sensor Analytics: A Case Study in Live Operational Intelligence

Chetan Gupta,
Krishnamurthy Viswanathan,
Lakshminarayan Choudur,
Ming Hao,
Umeshwar Dayal

HP Labs
firstname.latname@hp.com

Ravigopal Vennelakanti,
Paul Helm,
Anil Dev,
Sunil Manjunath,
Sastry Dhulipala,
Sangamesh Bellad

HP Enterprise Business
firstname.lastname@hp.com

ABSTRACT

In this paper, we present our Live Operational Intelligence (LOI) framework for developing, deploying, and executing applications that mine and analyze large amounts of data collected from multiple data sources to help operations staff take more informed decisions during management of operations in various industry verticals. We illustrate the use of the LOI framework with a case study from oil and gas drilling operations. The application involves characterizing and profiling on-shore wells being tapped for natural gas, and using this knowledge to construct a real-time operational intelligence engine for monitoring oil and gas drilling operations.

Keywords

Operational Intelligence, Sensor Data Analytics, Case Study.

1. INTRODUCTION

The oil and gas industry collects massive amounts of data from operations spanning exploration, drilling, and production via sensors installed in the oil wells, other measurement devices, and textual operational logs. The data repository often serves as a 'system of record' for audit of activity and occurrences. In addition, the time indexed data is a rich source of information that can be utilized for process control, quality control, and optimization of drilling operations, for instance to reduce *non productive time* [1]. The Non Productive Time related downtime during drilling operations costs the Oil and Gas industry over $40B per year. Collecting, archiving, analyzing, and visualizing the data requires a comprehensive framework. Indeed, there is a need for an analytics infrastructure packaged into a real-time monitoring and management solution and applied to the sensor and other operational data to track the state of the system at various stages of operation. In this paper, we present our Live Operational Intelligence (LOI) framework for developing, deploying, and executing applications that mine and analyze large amounts of data collected from multiple data sources to help operations staff take more informed decisions during management of operations. We illustrate the use of the LOI framework for an oil and gas drilling application. The application involves characterizing and profiling on-shore wells being tapped for natural gas.

Developing such applications is a daunting task due to a diversity of 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. Second, data from different sources has to be combined and aligned. For instance, to build a rational analytics solution requires - at the outset - combining textual and time series data to obtain a data set in the appropriate time sequence. Third, since data is collected from multiple wells using equipment supplied by multiple vendors, cross-well 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.

Our approach overcomes a majority of these challenges through an architecture, and a careful process that is separated into off-line and on-line phases. In the off-line phase, analytical algorithms drawn from a variety of disciplines are used, in conjunction with domain experts, to learn models and do application design. In the on-line phase, the application is executed by the LOI engine over streaming and historical data to raise alerts, produce actionable insights, and recommend actions for the operations staff. A key characteristic of our approach is that applications are specified declaratively in the form of data flow graphs. The operators that comprise the data flow include Extract-Transform-Load (ETL) operators, traditional SQL operators, advanced analytic operators, Complex Event Processing (CEP) operators, visualization operators, etc. Starting with *raw time series sensor data* and *external event data,* data flows through the series of operators producing automated actions and actionable insights.

SensorKDD '11, August 21, 2011, San Diego, CA, USA.

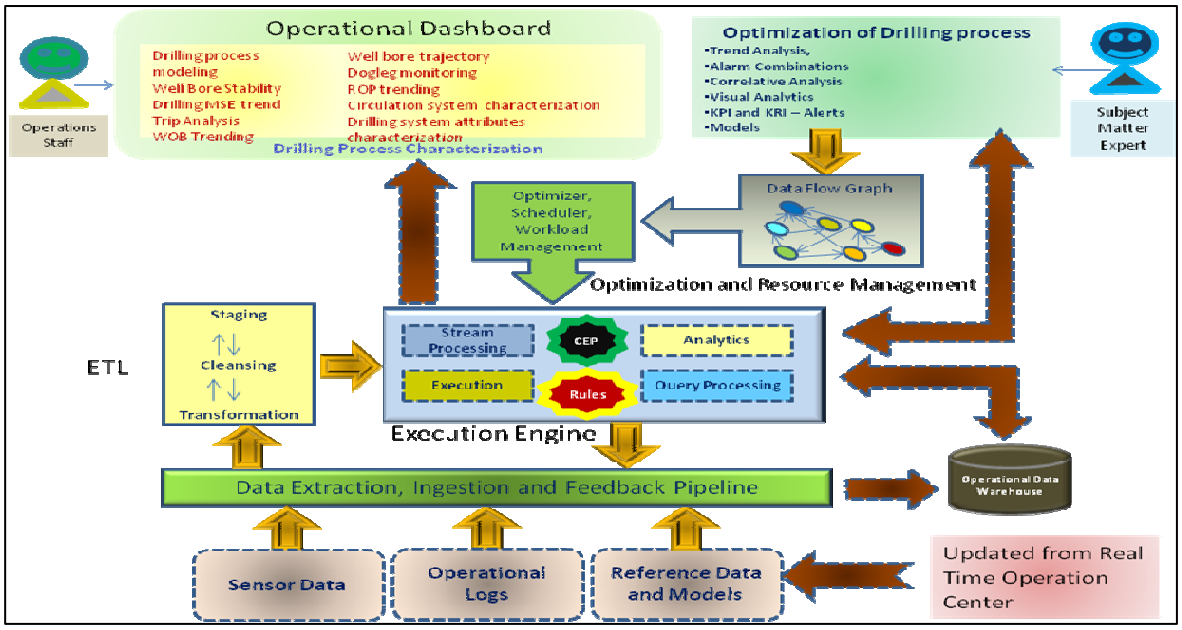

Figure 1: LOI Reference Framework

The LOI architecture includes several software components such as an ETL engine, a data management system, a CEP engine, a stream processing engine, etc. These components are common across various applications, but each application requires its own logic to be implemented. We discuss the software architecture in Section 2. Different applications share common design principles, but the actual operators comprising the application logic need to be constructed in consultation with domain experts. The overall process is discussed in Section 3, and Section 4describes the analytics techniques used to develop the flows. Section 5 presents some example flows for the drilling process, Section 6 shows validation results. We discuss related work in Section 6 and we conclude in Section 7.

2. Solution Architecture

To address the above challenges, we are designing a comprehensive Live Operational Intelligence framework and execution engine for data analysis of large scale operations.

The LOI framework (Figure 1) provides a plug-n-play approach to developing and executing analytical applications. Data from many different sources (streaming and batch-oriented, structured and unstructured) is ingested into the system through a data integration bus and an ETL component, which includes data staging, cleansing, and transformation capabilities. Depending on the latency requirements, fast moving data streams can be piped directly into the execution engine without loading into a date warehouse, or data can be captured in a repository for off-line analysis and model building. The execution engine provides parallel execution of query operators, analytic operators, complex event processing operators, and visualization operators. Applications are specified as data flow graphs, whose nodes are these operators. Associated with the data flow graph are performance requirements (for example, latency or throughput), which are used by the optimizer and scheduler component to produce efficient execution plans. In this paper, we don't describe these execution components in detail, since our objective is to focus on analytics for the drilling operations.

We have focused on a group of analytic operators by looking at real customer use cases from drilling operations. The operators currently supported 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
- Information extraction
- Attribute trending and multi-dimensional vector clustering
- Visual Analytics

3. Building the Application Logic

The overall process of application design is a two step process:

1. In the first step, called *Model learning and application design*, we use a combination of analytics and consultation with domain experts to define the set of flows that comprise the application.
2. In the second step, we implement the flows on an LOI engine, which optimizes and executes them in real time.

In Figure 2, we describe the overall process. The model learning and design phase is an offline phase. As a first step we received sensor data logs and drilling reports from our customer. The data was both structured and unstructured and data was obtained from more than 200 wells. The unstructured data contained close to 25000 files and the structured sensor data contained close to 15 million records.

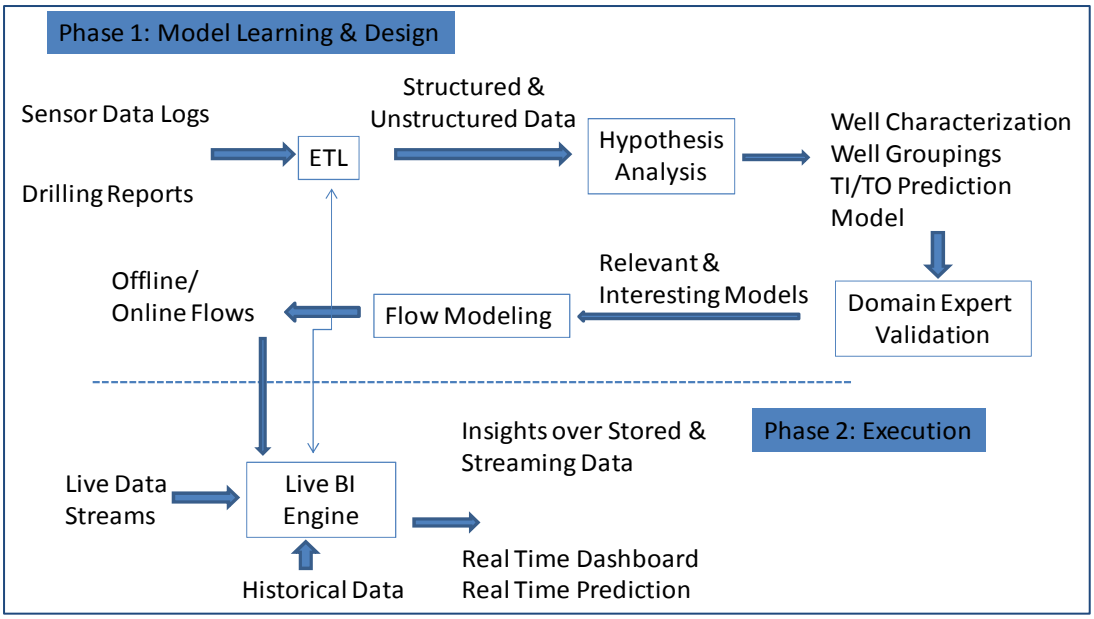

Figure 2: Two Phase Flow for Building the Application Logic

The sensor data logs are the logs of measurement during the drilling process. Typically for a single well, there are very many sensors measuring various physical attributes. These physical measurements can be divided into two categories: *drilling variables* and *circulation variables*, where the drilling variables measure quantities directly related to drilling (such as rate of penetration) and circulation variables measure quantities pertaining to fluids being circulated through the drilling system. There are typically 100 drilling variables and 100 circulation variables each. These measurements are periodic, typically at the rate of a reading per sensor every 10-15 seconds.

Besides the sensor data logs, we also received daily drill-reports, which are essentially manual natural language entries at the drill site about the drilling process. Through specific PDF adaptors, we were able to extract useful structured data from this large source of unstructured data. We observed that the two data sets had complementing context information about the entire system. The sensor data captured a large number of measurements while drilling. However, the system functions during non-drilling time were less obvious from the sensor data. The unstructured data set had a lot of context about sub-activities when the system was in idle state or during service –maintenance state. Furthermore, the un-structured data set was used to segment the various activities during both the drilling and non-drilling states to help better understand what activities consumed the time during the entire duration of the drilling process. In some cases, the unstructured data also captured information related to sensor system calibration which was used while smoothing the data set for comparative analysis. We observed that some wells manifested time-alignment issues due to deployment of multiple measurement systems. These time alignment issues of the drilling system and circulation system sensor data for correlation and analysis purposes were enhanced by referencing the un-structured data sets.

This actual sensor data received was messy containing among other artifacts, bogus values and missing values, etc. The sensor data attributes tagged by the vendor of the sensor instrumentation service did not have a common ontology and hence caused disparate naming conventions. Our solution required the attributes identified from both the structured and unstructured data sources to be standardized using a common ontology. On establishing a common reference ontology by leveraging industry taxonomies from web-sources, we were able to identify over 200 attributes that serve as complementary context information to the structured data sources. The table (Table 1) below provides a broad set of subsystem information that was derived from both the structured and un-structured data sets.

Table 1: Sub-System Information

Sub-System	Structured Data	Unstructured Data
Drilling System	X	X
Circulation System	X	X
Well Log System	X	--
Well Survey System	X	X
Vendor Profile System	--	X
GIS Information	--	X

As shown in Figure 2, through the process of ETL, from the raw sensor logs we stored clean structured data for the purpose of hypothesis formation and analysis.

In general, the schema after the ETL process for the incoming sensor data can be understood as:

<id, timestamp, sensor-id, value>

Where *id* refers to the id of the well for the record under consideration, *timestamp* refers to the time at which the sensor measurement was taken, *sensor-id* refers to the sensors which measured the physical quantity (there can be many sensors measuring the same physical quantity in different locations) and *value* refers to the actual measurement. This schema is not very conducive for the purpose of analysis, since for time series analysis it is useful to have all measurements taken at the same time be in one row or form one vector. So we transformed our sensor data to the following schema:

<Well-Id, time, location, drilling variables, circulation variables, drill-mode>

Where drill-mode is a "state variable" that indicates the overall operation being performed in the drilling process.

Once the structured and unstructured data is loaded into the DB we are ready for analysis. The purpose of analysis is ultimately to come up with flows that will be implemented on the LOI engine. The flows can help with the prediction of drill-mode, or to construction of rules that can take some action when the drilling process is not proceeding in the right way. However, before such flows can be arrived at, we need to do three things:

1. Perform a systematic analysis of the data
2. Validate the results of analysis with domain experts and choose relevant and interesting models than will need to be implemented on the LOI engine
3. Construct detailed operator flows for the chosen models.

For the purpose of systematic analysis of data, it's useful to start with the schema. Given the schema, besides the statistical characterization and visualization, following are the analysis tasks that were performed:

1. At the level of drilling and circulation variables:
    a. Find dependent/independent among drilling and circulation variables. This was done through both linear and logistic regression.
    b. Find frequent patterns in drilling and circulation variables
2. At the level of well-ids:
    a. Similar behaving well ids w.r.t. measured variables. This was done through ANOVA analysis
    b. Group wells by various sets of attributes such as location, drilling and circulation variables, drill mode. Once various clusterings are obtained, we do an association analysis to understand which well group together through various clusterings.
3. At the level of drill-mode:
    a. Predicting the drill mode. This was done through a Naïve-Bayes classifier.

We present some of the details in the next section.

4. Analytic Components

In this section we describe in some detail the various analytic components that constitute the system.

4.1 Basic Statistical Analysis

As a first step to achieving improved efficiencies in drilling operations, we statistically characterized the behavior of various physical parameters. Towards this end, analysis targeted characterizing the statistical relationships between various quantities both within a single well, as well as across multiple wells. Typical drilling operations monitor variables such as such as torque, rate of penetration, weight on bit, rotary revolutions per minute (RPM), hook-load, and bit-depth. The collected data is beset with problems such as incorrect values (outside the allowable range of the variables), missing values (sensor or measurement system malfunction), unacceptable variation of the measuring device impacting *repeatability* and *reproducibility*, *variation* due to differences in measuring systems and a host of others. These aspects of data collection affect data quality which has detrimental effects in downstream processing and analytics. In order to overcome the problems inherent in data collection, we perform basic statistical analysis to prepare the data for further analysis. These include: 1. deleting incorrect observations, 2. imputing missing values, 3. removing statistical outliers, and 4. de-noising to remove variations induced by measuring systems. After this preliminary data cleaning, basic statistical analysis is performed to profile the data. The profiling involves determining the statistical distribution of the data, computation of metrics such as skewness, kurtosis, and correlations between the variables. This initial step helps choose appropriate statistical methods to apply for monitoring and analysis.

4.2 ANOVA Analysis for Discriminating Between Wells

An important step towards understanding the behavior of the wells is to analyze the with-in well and across wells variability among variables of interest. Variability is best understood by performing statistical analysis of variance (ANOVA). The purpose of ANOVA to decompose variability into two components: *with-in well variability* and *between-well variability* to determine a signal to noise ratio (SNR). Given a set of wells and corresponding measurements, the wells can be divided into similar and dissimilar wells. Dominant between-well variance suggests that the wells may be dissimilar and further pair-wise comparison of wells can help group wells into similar classes by the method of multiple comparisons (Montgomery). A dominant with-in well variation indicates difference in variability among wells is due to behaviors specific to a well. ANOVA is extended to multivariate analysis of variance [2] to simultaneously analyze the effect of a multiplicity of variables on well(s) variability. This methodology is effective in assigning wells to similar classes. This enables the operator to transport *best-in-class* procedures to other wells in the same class to optimize drilling performance.

4.3 Clustering Of Wells, Signatures & Association Analysis

During drilling, different wells behave differently during the drilling operation. So it is useful to characterize each well's drilling behavior by a signature. Once the signature of a particularly good well is identified, we can try and achieve that signature during drilling. Similarly, if a signature for a particularly bad well is identified, we can steer the drilling away from that. To obtain the signature for a well we compute the Pearson correlation coefficient between every pair of variables over time. Pearson correlation coefficient can be efficiently computed in one pass. For d dimensional data, this results in a $0.5*d*(d-2)$ length vector. For large d (~100) the above is a very large number. Hence, we compute separate drilling and circulation signatures for each well and only take a subset of the dimensions for each to arrive at the signature. The drilling variables behave differently during different drill modes. Hence, we compute separate signatures for each drill mode. So if there are k drill modes, it leads to $2k$ signatures per well.

Additionally it is also useful to group or cluster the wells and compute signatures for each cluster. A domain expert can then study the signature of different clusters instead of looking at the signatures of each well, and can also gain insights into which wells are behaving similarly and which set of wells are behaving differently. To cluster the wells, we use the signature as the vector

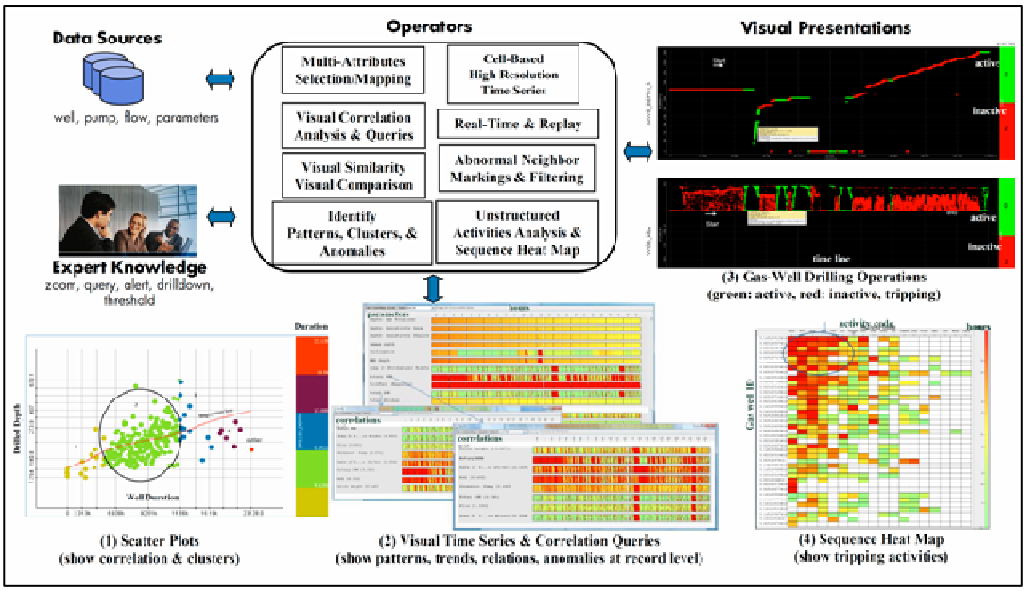

Figure 3: Visual Analytics Techniques & Flow

describing the well. These signatures are then fed to an EM algorithm. Since there are *2k* signatures per well this led to *2k* different clusterings. We also cluster wells by their location to obtain a total of *2k+1* clusterings. By this process we have grouped the wells with respect to all the variables described before. This helps build insights into the drilling operations.

Since, we obtained *2k+1* groupings for wells, we wanted to find out which wells are always found together in different groupings. This might better help us better understand the drilling operation. To achieve this we deployed a market basket analysis [12], where each basket essentially is the cluster membership. For example, for clustering *j*, and cluster number *k*, the basket M_{jk} would contain all the cluster ids in that cluster. This way we obtain $\sum_{i=1}^{2k+1} |C_i|$ baskets, where $|C_i|$ is the number of clusters in clustering *i*. Finding frequent itemsets over these baskets is equivalent to finding the sets of wells which occur together through various clusterings.

4.4 TI/TO Prediction

While drilling, a drill bit rests on the floor of the well and is rotated causing it to displace material. The drill string is a column that transmits torque and drilling fluid to the drill bit. During the drilling process, the drill string is often pulled out of the wellbore and then lowered back in. This process is referred to as tripping. Tripping is performed for a variety of reasons such as replacing the drill bit or correcting malfunctions in downhole drill equipment. Minimizing the time spent tripping can reduce the time and therefore the cost of a drilling project. While some of the time spent tripping is perhaps inevitable, long tripping windows are perhaps a sign of unusual problems and can be classified as non-productive time. Therefore, attempting to diagnose reasons for abnormally long trip durations or providing advance warnings for the same can point towards drilling practices that reduce this non-productive time.

The measurement system records the state of the drilling process from which the time windows when tripping occurred can be identified. Our goal is to predict the duration of the current tripping window as soon as the tripping commences. This prediction is made using the quantities that are monitored as part of the drilling and circulation system. We are interested in determining if the parameters in the period leading up to the beginning of a trip are informative about its duration. If so, then further analyzing the informative quantities can be helpful towards designing more efficient drilling practices. Also, the advance warning provided by this prediction mechanism can indicate if the drilling is proceeding on a path that could lead to abnormally large trip durations.

A clustering algorithm was employed on trip durations aggregated from all the wells to first classify the durations into three bins labeled short, medium and long. The goal of the prediction problem was to predict, at the onset of a trip, which bin the trip duration will fall into. The features that were used for prediction were 30 minute time windows of the drilling and circulation parameters. There were two phases to this task – model building and online prediction. In the former, we built a model that relates the trip duration to the features. In the latter, we used the model to predict the classification of the duration in an online fashion.

We constructed a Naïve Bayes model [3] for the problem. We did so for two reasons - lack of prior knowledge about the dependencies among the different features, and the presence of a large number of candidate features. The latter problem constrained our ability to build more general probabilistic models that admit all the measured variables as features as the data was insufficient to learn complicated dependencies. Formally, the dependent class variable C denotes whether the window is short, medium or long. This prediction is made based on features such as torque, RPM and depth. They are denoted by $F_1, F_2, \ldots, F_n$, the probability model for a classifier is a conditional model, $P(C| F_1, F_2, \ldots, F_n)$. Since the number of features n is large, it is it is assumed that the features $F_1, F_2, \ldots, F_n$ are conditionally independent given *C*, leading to:

$$P(C \mid F_1, \ldots, F_n) = Z^{-1} P(C) \prod_{i=1}^{n} P(F_i \mid C)$$

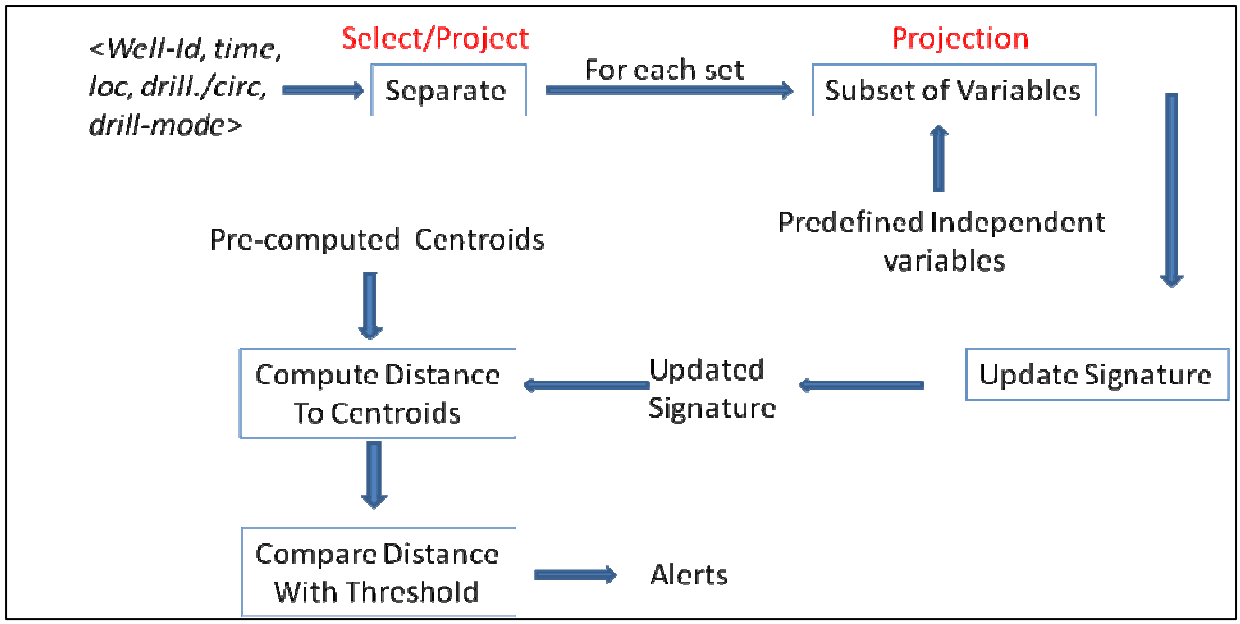

Figure 4: Flow for Signature Matching

Where Z is a scaling factor dependent only on $F_1, F_2,...,F_n$. The features $F_1, F_2,...,F_n$ were discretized by clustering 30 minute windows of different measured quantities such as RPM, torque etc. In the online phase, the model generated thus was applied to the data. Only the pairwise joint distributions *P(C, Fi)* need to be stored. The results are discussed in Section 6.

4.5 Visual Analytics

The goal of visual analytics is to merge rich information visualization techniques with above analytic algorithms for users to explore and analyze large scale volumes of data. For managing live oil/gas operations, it is essential to devise techniques that work with high-dimensional and high-speed real-time streaming data. However, today's visualization (i.e.; scatter plots, time series, spread sheets) either often have a high degree of overlapping or use aggregation and therefore, important patterns and trends may get lost in high density areas. We used a variety of advanced visualization techniques using a cell-based placement technology. Each data point represents a data item. Each data point is accessible and can be queried for detailed information. Furthermore, we define a group of operators to map input parameters into visual objects from multiple sources, and then generate a suite of visualization presentations [4, 5, 6] according to the characteristics of the input data objects.

To visualize the data of the gas drilling process as shown in Figure 3, we use five visual analytics techniques: (1) non-overlapping scatter plots to cluster over hundreds wells with outliers according to the drilled depth and well duration, (2) visual time series map to show patterns, trends, and correlations at record level, (3) real-time animation to show gas-well drilling operations in both active and inactive states to identify the root-cause of non-productive time, and (4) a sequence heat map to show tripping activities and their duration (hours). Furthermore, we combine user's expert knowledge into the data analysis process. The effect of the zooming and queries is controlled by the user. Users can quickly drilldown to the detailed information as needed.

5. Implementation on the Live BI Engine

In this section we describe the detailed flows for two of our analysis tasks. As mentioned earlier, the operations that need be performed are implemented as flows. These flows can be for offline execution or online execution.

For example, some of the analysis tasks mentioned in the previous section, such as computing cluster signatures, needs to be done periodically. For these flows are created and optimized manually, so that they can be executed whenever the need arises. We are currently working on an optimizer called QoX (Quality of X) [7], that would optimize and construct an executing plan for the flow.

Besides the offline flows are the online real-time flows. These flows ingest data as it streams in and produce either visualizations, actions or predictions. In the next two subsections, we describe two flows, one that produces an action and another that produces a prediction.

5.1 Signature Matching

The purpose of the flow is to create an early warning if the drilling for a well is proceeding in an un-desirable fashion. As the data for a well as it flows in, it updates the signature of the well. This signature of the well is compared to a set of stored signatures which correspond to undesirable behaviors. If the distance of the signature from any of the stored signature is less than a threshold, then an alert is created. The detail flow for this process after the ETL part of the flow is depicted in Figure 4.

After the ETL part of the flow, the incoming schema for the data looks like *<Well-Id, time, location, drilling variables, circulation variables, drill-mode>*. This schema is desirable in the analysis section it allows for easy analysis over multiple variables for a single well. In the first operation are *2k* Select/Project operations that construct *2k* tuples from the incoming tuple by selecting and projecting for the combination of drilling/circulation and *k* drilling modes. Then in the second operator, from each of the *2k* tuples, those dimensions are projected out that have been deemed to be useful through offline analysis. Notice that these are two traditional SQL operations. The third operator is an operator

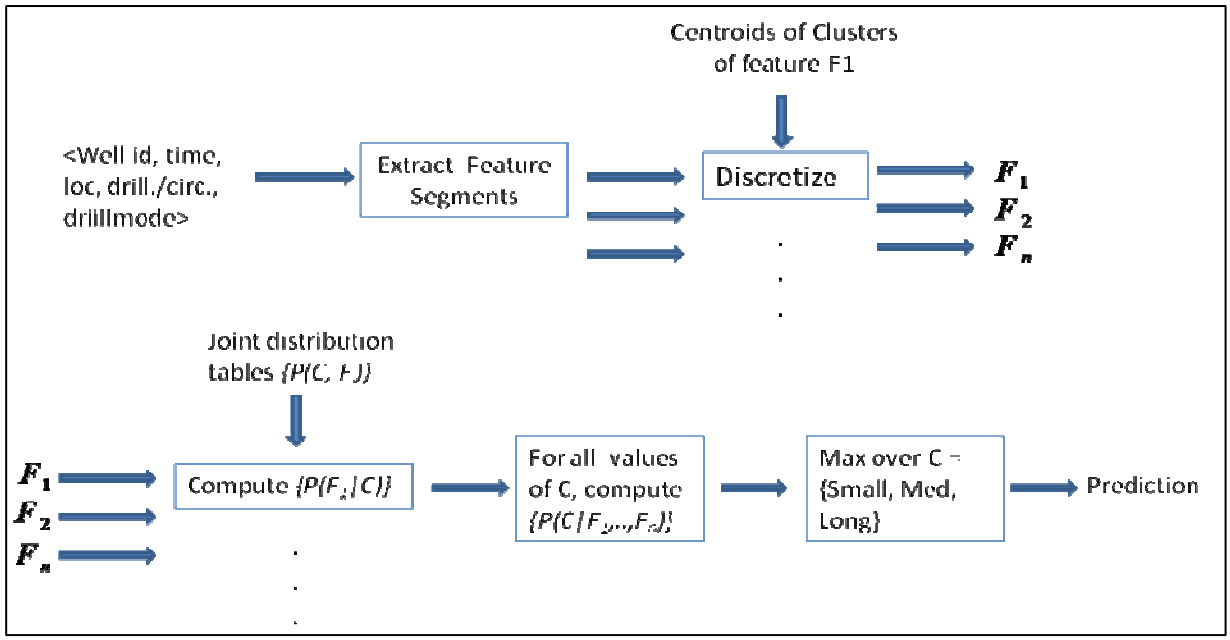

Figure 5: Flow for Prediction of TI/TO Duration

written specifically for this problem. This operator updates the centroids for each well for each *2k* combinations. This variable is a *statefull* variable and needs to maintain *2k* sums and sums of squares per well to update the Pearson Coefficient. The updated centroids are sent to a rules engine, where all pair distances between the centroid and the stored set of centroids is computed. As mentioned before if any of the distances is less than a threshold, an alert is issued.

5.2 TI/TO Predictive Model Execution

We describe the flow for predicting the tripping duration. As described in Section 4.4., the goal is, at the onset of a tripping window to predict if it would fall in the class short, medium or long. This classification is determined based on the statistics of tripping windows across all wells. The features used in the classification are 30 minute time windows (prior to the onset of tripping) of the various drilling and circulation parameters. There are two phases to this prediction problem. The first is an offline training phase where the models are constructed, and the second is the online phase where the models are employed for prediction.

In the offline phase, the following tasks are undertaken. For each parameter, 30 minute windows preceding tripping windows are collected and clustered into a discrete number of classes. These constitute the features in the model. The centroids of these clusters are selected to be representatives of the clusters. Further, the joint probability distributions of each of the features and the duration of the tripping window is computed and stored. The data flow during the online phase of the prediction is described in Figure 5. Once the onset of a tripping window is detected (this is usually done by monitoring the drill mode), the 30 minute segment just preceding the tripping window is extracted for all the drilling and circulation parameters. For each parameter, these segments are discretized by comparing them to the set of cluster centroids corresponding to that parameter that were generated in the training phase. This operation involves a nearest neighbor search in high dimensions. The output of this operation is a set of features $F_1, F_2,\ldots,F_n$ corresponding to the current tripping window. For each F_i, the conditional probability distribution $P(C|\ F_i)$ of the duration of tripping window (C = small, medium or long) is computed from the joint distribution tables that were generated in the training phase. Then the conditional distributions are used in a Naïve Bayes model as described in Section 4.4 to compute $P(C|\ F_1, F_2,\ldots,F_n)$, the conditional distribution of C given the features $F_1, F_2,\ldots,F_n$. Once this is computed, the value of C that maximizes this probability is declared the prediction

6. Validation

In this section we discuss some of the results of this project. The first part details the outcome of the prediction exercise and the second part, outlines the overall value that the LOI framework and analysis produced.

Given that the prediction component aimed at determining whether the ensuing trip duration was short, medium or long, a random guess would result in 66.66% error. From the distribution of the trip durations it was observed that 53.41% of the time the durations are small, 38.88% of the time the durations are medium and 7.73% of the time the durations are large. Informed by these statistics and no other accompaying information, the best prediction of a given trip window is to predict small and this would bring down the error to 47% (since we will be wrong whenever the duration is medium or large).

Using the features extracted from Drilling variables and the naïve Bayes model, we were able to reduce the prediction error to 35%. This was nearly a 25% improvement in the prediction error. Using the features extracted from Circulation variables and the naïve Bayes model, we were able to reduce the prediction error to 39%. This was nearly a 20% improvement in the prediction error.

We believe that with further analysis involving more complicated models, and incorporating further domain knowledge, it may be possible to drive down the prediction error futher.

Insights into operational process data assembled from applying several of the above data mining techniques on multiple disparate data sources can provide opportunities to improve the operational efficiencies of the system. Drilling operations of an oil and gas well today has very high costs associated with standard

procedures on a day to day basis. Our customers are exploring options of extracting further insights into drilling parameters to study areas of improving efficiencies which may help drive costs down by over 40%. In addition to driving costs down through optimization, the methods described above also provide the ability to validate many of the dependent conditions during drilling to make step changes in the overall instrumentation and execution of the drilling process. We shared the results we obtained with domain experts from the industry, and received very positive feedback from them on the insights obtained.

7. Related Work

For decades, the oil industry has used computers in operations to maximize profit. More recently, they are contemplating using supercomputers, software systems and related techniques to let supercomputers at different locations share the workload. Usage of computing and software is envisioned mostly to couple computer models with the reservoir geology [8]. Furthermore, in oil production operations, the phenomena of slugging and churn can cause severe disruption. These are simulated, modeled, and analyzed by mathematical and data mining techniques [9, 10]. Also, some work is done related to optimizing *non productive time* in oil drilling [11], but is mostly related to analyzing wellbore stress. There are some data models used in the oil and gas industry: PPDM Data Model – 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.

In our view, the live operations intelligence is only system that combines the collecting, archiving, mining, analyzing, and modeling into a unified framework.

8. Conclusions

In this paper, we presented our Live Operational Intelligence (LOI) framework. We illustrated the use of the LOI framework with a case study from oil and gas drilling operations. Specifically, we discussed the solution architecture, which discussed the various components that comprise an LOI framework and their inter-relationships. Then we discussed the steps we used to construction the application logic. The steps include starting from ETL operations, to performing analytic tasks, to constructing operational flows in conjunction with domain experts. The analytic tasks range from basic statistical techniques to building a predictive model to visualization techniques. We also presented two flows that can be executed in real time to obtain alerts and predictions.

We continue to build out our LOI framework. Some interesting research challenges we are exploring are: enriching the slate of analytic and visualization operators that can be used to construct flows (offline & online flows); coming up with new analytic problems/solutions given the space of LOI; devising new execution strategies; constructing an optimizer for efficient execution of flows; and applying the technology to additional operations management problems.

9. REFERENCES

[1] http://www.successful-energy.com/wp-content/uploads/2011/01/2009_OTC_20220_Eliminating_Non-Productive_Time.pdf

[2] Montgomery, D.C., 2000, *Design and Analysis of Experiments*, 5th Edition, 2000, Wiley

[3] Domingos, Pedro & Michael Pazzani (1997) "On the optimality of the simple Bayesian classifier under zero-one loss". *Machine Learning*, 29:103–137.

[4] 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.

[5] Keim, D. A., Hao, M. C. Dayal, U., Janetzko, H., and Bak, P., Generalized Scatter Plots. Information Visualization Journal (IVS), 2009.

[6] Hao, M., Marwah, M. Dayal. U, Janetzko, H., Keim D., et al, Visualizing Frequent Patterns in Large Multivariate Time Series. Information Visualization VDA11, CA.

[7] Simitsis, A., Wilkinson, K., Castellanos, M., Dayal., U. QoX-driven ETL design: reducing the cost of ETL consulting engagements, SIGMOD 2009, 953-960.

[8] Kurc, Tahsin et al., 2005, *A simulation and data analysis system for large-scale, data-driven oil reservoir simulation studies*, Concurrency and computation: practice and experience, concurrency computat.: pract. exper. 2005; 17:1441–1467

[9] Di Meglio F, et al., 2006. Reproducing slugging oscillations of a real oil well, IEEE conference on Decision and Control, 2010, 4473 – 4479

[10] Singh, A, et al.,2011, Local frequency based estimators for anomaly detection in oil and gas applications, Joinst Statistical Meetings (to appear)., Miami, Florida, 2011

[11] http://www.oilproduction.net/HalliburtonTechnologySolutionReduceDrilling.htm

[12] Agrawal R., Imielinski T. , Swami A., "Mining Association Rules between Sets of Items in Large Databases. SIGMOD 1993