An Event Based Framework for Improving Information Quality That Integrates Baseline Models, Causal Models and Formal Reference Models *

Joseph Bugajski
Visa International
PO Box 8999
San Francisco, CA 94128
jmbugajski@yahoo.com

Robert L. Grossman †
and Eric Sumner
Open Data Partners
1145 Westgate Street
Oak Park, IL 60301
{rlg1@, esumner@}opendatagroup.com

Tao Zhang
Bearing Point
1676 International Drive
McLean, VA 22102
tao.zhang@bearingpoint.com

ABSTRACT

We introduce a framework for improving information quality in complex distributed systems that integrates: 1) Analytic models that describe baseline values for attributes and combinations of attributes and components that detect statistically significant changes from baselines. These models determine whether a significant change has occurred, and if so, when. 2) Casual models that help determine why a statistically significant change has occurred and what its impact is. These models focus on the reasons for a change. 3) Formal business and technical reference models so that data and information quality problems are less likely to occur in the future. In this note, we focus on the first two types of models and describe how this framework applies to data quality problems associated with electronic payments transactions and highway traffic patterns.

1. INTRODUCTION

In this note, we introduce a framework for monitoring, exploring and ameliorating the information quality of event based data. We are interested in data and information quality problems for complex, distributed real time systems. Here are two motivating examples that are described in more detail below.

The first example is the processing of electronic payments. A payments card transaction is an example of an event and involves several parties, namely the cardholder, the merchant, the merchant's bank, the cardholder's bank and the payment processor. Each of these independent parties is involved in the decision of whether to accept the transaction, decline the transaction, or request further information about the transaction. Poor data and information quality can increase the rate of improper declines and improper approvals.

The second example is the real time analysis of traffic patterns over a metropolitan region in order to quickly identify accidents and other anomalous behavior. In this example, we assume that traffic sensors produce real time information about the speed and volume of traffic. The resulting sensor readings are examples of events which are aggregated to produce features summarizing the traffic at a particular time and location. Traffic patterns can be quite complex and vary with the time, location, weather, local events, etc. Obtaining accurate data can be very challenging.

There are several challenges common to both of these examples:

1. The data sizes are large and the data is heterogeneous.
2. The data is produced and processed by several different parties and this sometimes introduces data and information quality issues.
3. The data and system is sufficiently complex that establishing baseline data quality and information levels can be quite challenging.

In this note, we describe a framework for monitoring, exploring and ameliorating the data and information quality for systems with these types of challenges. The framework has four components:

Building Baselines. The first component of the framework is an analysis engine that:

1. analyzes event based data
2. divides the event based data into appropriate segments
3. computes features or states from these events for each segment
4. and from these features estimates appropriate baselines for each segment.

The idea is that although the system has a whole may be quite complex by dividing the data into enough cells (by

*This work was supported in part by the Visa International Data Interoperability Program and the U.S. Army Pantheon Project.

†Robert L. Grossman is the corresponding author. He is also a faculty member at the University of Illinois at Chicago.

IQIS 2005, June 17, 2005, Baltimore, MD, USA.

restricting to appropriate ranges of values along each dimension), the data becomes homogeneous enough to analyze, addressing the first challenge. In this context, we call each such cell a *segment* (as in segmented modeling). In our experience, analyzing appropriate features associated with various entities of interest instead of directly analyzing the events themselves helps address the second challenge.

Monitoring. The second component of the framework is a monitor that:

1. monitors streams of event based data
2. computes summary or state information,
3. uses this information as input to statistical measures and models
4. compares the outputs of the measures and models to previously computed baselines, and
5. issues alerts in case of statistically significant deviations.

The goal of monitoring is to determine *whether* a statistically significant change has occurred. In other words, rather than starting with a certain expectation of data or information quality, the approach is to detect as quickly as possible changes in data or information quality, addressing the third challenge.

Root Cause Analysis. The third component of the framework is a process for exploring the monitored data to understand casual relationships between data and defined outcome variables. A variety of techniques can be used to understand causality, including contingency tables [1], discriminant analysis, regression, and classification and regression trees [13]. The challenge is to understand whether different variables are causally related or simply correlated.

Here is a simple example from the analysis of payments card transactions: The decline rate of transactions is an outcome variable that has obvious business significance. Some declines are due to insufficient funds or fraudulent usage, while others are due to data quality problems. Errors in how a merchant processor sets up an e-commerce system can lead to hidden data quality problems and higher than usual declines. The role of the root cause analysis process is to understand some of the casual reasons for statistically significant changes in baselines. In other words, the goal of root cause analysis is to determine *why* something has happened.

Amelioration. Once one one or more root causes are identified, the goal of the fourth component of the framework is to take actions to ameliorate the problems. In the example, above this may involve educating the merchant processor so that the identified data quality problems do not occur in the future.

In our experience, data quality problems for complex distributed systems are often the result of documentation that is hard to understand or difficult to interpret. We have been exploring the use of model driven architecture [7] to provide formal business and technical reference models and methods that can directly address this difficulty.

In this paper, we describe this framework and provide some high level experiences of some of the implementations we have done.

Although monitoring, causal analysis and amelioration are components for several different data and information quality methodologies [6], [14], [15], as best as we can tell from reading the literature, our paper makes the following contributions:

1. Most data and information quality methodologies [14], [17], [15] do not distinguish carefully between transaction or event data and summary or profile data that is aggregated from it. This is an important distinction for our targeted applications. As a simple example, the data and information quality issues are quite different for payments card transactions and summary information at the merchant, account, issuer, or acquirer level.
2. A common approach to data and information quality is to measure the quality of data along several dimensions. For example, accuracy, completeness, validity, timeliness, etc. (see, for example, [20]). In contrast, our focus is not on the dimensions themselves but on effective procedures for creating small cells or segments of data (defined by dimensional ranges) that have both business and statistical significance and building effective baselines for each cell. For example, we view data for a transaction process as being naturally divided into cells by logical entity (issuer, acquirer, type of payments card or payment product) and temporal entity (weekday, holiday, weekend, etc.)
3. Our methodology is closely tied to standards, in particular, the Predictive Model and Markup Language or PMML, that dominant standard for statistical and data mining models. This has several important implications. In particular, this allows us to instantiate a data quality in a standards based fashion as an XML file.

2. RELATED WORK

There is now quite a bit of research in the field of data and information quality, and several books [18] and [3]. In this section, we briefly discuss some of the research that is most directly relevant.

Our approach to data and information quality is statistical. This tradition goes back at least to Deming [4]. In particular, our focus is on establishing baselines and measuring statistically significant deviations from baselines. This is a standard approach in change detection [2]. In contrast, many approaches for data and information quality are business systems or engineering based (see for example, [14] or [18]).

Once deviations from baselines are detected, a statistical analysis is undertaken to try to determine the underlying reasons. Today, there are a wide variety of approaches for trying to determine causality, including root cause analysis [19], contingency tables [1], discriminant analysis, regression, and classification and regression trees [13].

A common approach to data and information quality is to measure the quality of data along several dimensions. For example, DOD Guidelines recommend using accuracy, completeness, consistency, timeliness, uniqueness and validity. As another example, Strong et. al. [20] introduce 16 dimensions organized into four categories (intrinsic information quality, contextual information quality, representational informational quality, and accessibility information quality).

In this note, we use some, but not all, of these standard dimensions. In particular, most of the work described below are based on metrics measuring completeness, consistency, and validity.

In this note we distinguish formally between input events and persistent states. Although this is standard in dynamical systems, automata theory, and control theory, but does not appear to be a standard approach in statistics or data mining [11].

Most data and information quality methodologies [14], [17], [15], [6] include components for defining, measuring, analyzing, and improving data and information quality issues, as our does. On the other hand, the approach sketched below differs in two significant ways from [14], [17], [15], [6] and related work:

1. Our approach is closely tied to standards based architectures. As many approaches do today, we employ a data warehouse. In addition, we employ a monitor for monitoring streaming data, a component for building baselines, and a scoring engine [12] for measuring the deviation of the streaming data from the baseline.

2. Second, our approach is closely tied to standards for data mining and statistical models [10] and [5], such as the XML-based Predictive Model Markup Language.

3. EXAMPLES

We have applied the framework described here to several examples, including those involving payments card transactions, highway traffic data, and multi-modal sensor data. In this section, we provide a bit of background for two of these examples in order to make this note more self contained.

Here is a simplified description of some of the steps involved in a payments card transaction.

1. A cardholder (the card has an account number) purchases an item at a merchant.

2. The merchant has a relationship with a bank called the acquiring bank, which agrees to process the payments card transactions for the merchant. The acquiring bank provides the merchant with a terminal or other system to accept the transaction and to process it.

3. The acquiring bank has a relation with financial payment system, such as those operated by Visa, MasterCard, Discover, etc. The transaction is processed by the acquiring bank and passed to the payment system.

4. The payment system the transaction and passes the transaction to the bank that issued the payments card to the card holder (the issuing bank). In other words, one of the essential roles of the payment system is to act as an intermediary between the acquirer and the issuer.

5. The issuing bank processes the transactions and determines if there are sufficient funds for the purchase, if the card is valid, etc. If so, the transaction is authorized; the transaction can also be declined, or a message returned asking for additional information. In each of these cases, the path is reversed and the transaction is passed from the issuing bank back to the payment system, from the payment system back to the merchant bank, and from the merchant bank, back to the merchant.

One of the challenges of a problem like this is to monitor in real time data and information quality problems for the various different parties when the data is processed transaction by transaction. Our approach is to use event based processing model and to create different summary or feature vector for each entity of interest. In this case, this includes the cardholder, the merchant, the acquirer, the issuer, and the payment system.

As another example, consider the problem of understanding highway traffic congestion. The Gateway System employs over 800 sensors to collect volume, speed, and occupancy data in a three state, fifteen county Gary-Chicago-Milwaukee (GCM) corridor [16]. The Pantheon Gateway Testbed augments this data with data about weather, special events that may effect highway conditions, and related information.

Here is a question that can be posed using this data: On a Monday, around 7 am, that is not a holiday, and when it is beginning to rain lightly, what is the average speed and volume for traffic on Interstate - 290 near the Austin exit? What will the average speed and volume be around 8 am if the rain continues?

This is an important motivating question and suggests our approach. Rather than try to understand data and information quality problems for the system as a whole, our approach is divide the data into relatively homogeneous segments or cells (such as traffic on Mondays around 7 am with light rain) and to establish appropriate baselines for each such segment. With this knowledge, understanding data and information quality problems becomes much easier.

4. OVERVIEW

Broadly speaking, our technical approach is as follows:

1. We assume that data consists of events, such as transactions or sensor readings. Events are first divided into segments or cells that are relatively homogeneous. An event can be associated with multiple segments. Of course, there are many different ways of doing this and a discussion of these is outside the scope of this paper.

2. For each segment, events are processed to update state or feature vectors containing persistent features associated with the events. This is described in more detail in the section below.

3. For each segment, appropriate baselines are established for collections of state or feature vectors. Baselines may be temporal, geospatial, logical, or some combination. Determining good break points for dimensions appears to be a difficult problem and is outside the scope of this paper.

4. Deviations from baselines are detected using simple threshold models or more complex change detection models [2]. Deviations are used to determine as quickly as possible whether something has changed.

5. Separately, casual analysis is used to determine whether conditions or combinations of conditions are likely to

effect outcome or impact variables. To begin n by m contingency tables are used as a starting point for this analysis [1]. This is supplemented as required by discriminant analysis, linear regression, and nonlinear regression techniques, such as classification and regression trees [13].

6. When important casual conditions are identified, formal models [7] are used to begin to improve the condition. The use of formal models for this purpose is also outside the scope of this paper.

5. EVENT BASED DATA PROCESSING

It is useful when developing baselines to distinguish between data and derived attributes following [5].

A *data attribute* is simply an attribute present in the data itself, while a *derived attribute* is an attribute derived from the data or aggregations of the data. For example, given a payments card transaction the raw amount of the transaction is in the data itself, while currency related attributes, interchange fees, the amount of transactions for an account holder during the past hour, the number of declined transactions that are e-commerce-related, etc. are all examples of derived attributes.

In this note, we follow an event based approach to analyze information [11]. We assume that we are given:

- A stream of events α_1, α_2, ... in $\mathbf{R}^m$. Attributes in the events are data attributes.
- A finite collection ξ of feature vectors (also called state vectors)

$$\xi = \{x_1, x_2, \ldots, x_n \in \mathbf{R}^N\}.$$

- An update rule (denoted dot) specifying how an event α updates the collection of feature vectors

$$\xi' = \alpha \cdot \xi.$$

  Attributes in the feature or state vectors consist of derived attributes formed from the event data through transformations and aggregations.

- A function of the state space

$$f : \mathbf{R}^N \longrightarrow \mathbf{R}^1$$

  representing a statistical or data mining model producing scores or other outputs.

We illustrate this using our running example of payments card transactions. In this case, the events are payments card transactions, while the state vector represents information associated with a related entity, such as a payments card or issuing bank. For example, if the state vector represents a payments card transaction, the a component of the state vector might be the number of transactions during the the previous 60 minutes. If the state vector represents an issuing bank, then a component of the state vector might be the number of declined transactions during a day. In both cases, the model might be a change detection model indicating that the observed feature is statistically different than a previously computed baseline level.

For another example, assume that events consist of sensor readings for a collection of sensors and that there is a feature vector for each sensor that maintains the number of readings, the average of the readings, the min sensor reading, and the maximum sensor reading for each sixty minute period, for each of the $168 = 7 \times 24$ sixty minute periods during a seven day week. Here the update rule, updates the features corresponding to the appropriate sixty minute period.

6. BASELINE MODELS

In this section, we review a standard approach for detecting deviations from baselines [2]. In the methodology described here, this is applied to each segment or cell separately. We assume that one have mean and variances representing normal behavior and behavior that is not normal.

More explicitly, assume we have two Gaussian distributions with mean μ_i and variance σ_i^2, $i = 0, 1$.

$$f_i(x) = \frac{1}{\sqrt{2\pi}\mu_i} \exp \frac{-(x-\mu_i)^2}{2\sigma_i}$$

The log odds ratio is then given by

$$g(x) = \log \frac{f_1(x)}{f_0(x)}.$$

and can now define a CUSUM algorithm as follows [2]:

$$Z_0 = 0.$$

$$Z_n = \max\{0, Z_{n-1} + g(x_n)\}.$$

Streaming data is compared to existing baseline data and deviations are noted and flagged for investigations. Baseline models like these are used to determine whether something has changed, and, if so, when it changed.

Baseline models aggregate data along several dimensions (in the running example of payments card transactions, baseline models aggregate data by issuer, acquirer, region, temporal period, type of transaction, etc.) At too high a level of granularity, too much information is lost. At too fine a level of granularity, it is too difficult to discern what is important.

For example, it is important to know that the overall authorization rate is 93%, but this doesn't easily lead to actions that improve the authorization rate. On the other hand, knowing that the authorization rate for transactions with inconsistent point of sales data is 20% higher than the average provides some very important information. The challenge with baseline models is to choose the right of level of granularity so that the baselines are meaningful and can uncover opportunities for improvement. From this perspective, advanced baseline models can dive down into the data to uncover homogeneous pockets of data and establish appropriate baselines for each pocket.

The Predictive Model Markup Language or PMML is an XML based language to describe statistical and data mining models. As part of the work described here we have produced PMML models for baselines and to detect deviations from baselines. Using the terminology of [5], an application that produces baselines is a PMML producer and an application that monitors baselines is a PMML consumer.

7. ROOT CAUSE ANALYSIS

Different applications structure the root cause analysis differently. For example, when analyzing payment data the current approach consists of two steps:

1. In the first step, the relation between different conditions and different outcome or impact variables is examined using to generate alerts, together with confidence levels. The goal of this step is to provide an initial identification of conditions that are correlated with variables of business interest.

2. In the second step, an investigation is undertaken involving subject matter experts to explore the relationship and to determine whether the relation is *casual*, and, if so, to estimate its business impact.

We now briefly describe each of these steps in the running example of payments card transactions.

A simple way to analyze data is to use data and derived attributes to define conditions and then to examine the relation between the conditions and certain outcomes using contingency tables [1]. Recall that contingency tables capture the relation of two categorical variables. See the Table below for a simple example. In addition to contingency tables, we currently beginning to explore multivariate techniques, such as discriminant analysis or classification trees [13] to generate alerts.

Alert conditions are defined by specifying values or ranges of values for data or derived fields. Defining binary indicator variables is a very simple way of defining conditions. Here is a simple example. A transaction has a field indicating that it is e-commerce related. For example, an indicator attributed can be defined by defining a condition to be 1 if the transaction is e-commerce related in this sense and 0 otherwise. As another example, a payments card transaction also has a field indicating the type of merchant. An indicator variable can be defined if the type of merchant is a casino and 0 otherwise. More complex types of conditions can also be defined. For example, conditions with three, four or more different values can also be defined. Conditions defined in these ways are examples of statistical factors.

Outcome and impact attributes can be data attributes, but are generally derived attributes. Examples include a binary variable indicated whether a financial transaction is approved or not. As another example, a binary indicator variable indicating whether a transaction is cleared, or whether or not a transaction is associated with a charge back or not.

	Outcome - State 1	Outcome - State 2
Alert Condition Present	n_{11}	n_{12}
Alert Condition Not Present	n_{21}	n_{22}

Table 1: A 2x2 contingency table that is sometimes a helpful step in the root cause analysis of alerts.

In addition to baseline models, our framework also uses more complex statistical to help determine what conditions and combination of conditions are likely to result in certain outcomes, such as a decrease in authorizations. In conjunction with this, our framework also employs an investigative process involving subject matter experts to help determine why an outcome variable (such as the approval rate or charge back rate) has changed and if so what the impact is? We call these *casual models*.

8. STATUS

To date, we have undertaken several projects using this methodology. Here we give a brief summary of the status of two of these.

Payments card transactions. We have begun to analyze data and information quality problems associated with declines for a large financial transaction processor. The status is as follows: some measures for incomplete, invalid, and inconsistent fields have been developed. Using this measures, we are currently developing baselines and identifying combinations of conditions capturing common data and information quality problems. We are also examining casual relations between these conditions and impact variables, such as the rate of declines. Finally, preliminary business and technical reference models for some of the more important fields have been developed [7].

Highway traffic data. The Gateway System collects near real time data from over 800 highway traffic sensors covering the three state, fifteen county Gary-Chicago-Milwaukee (GCM) corridor. This data is archived by the Pantheon Gateway Project [16] and overlaid with data about special events, such as concerts or sports events, and data about the weather. In addition, data about accidents is collected. Using this data, we have established preliminary baselines used a real time scoring engine employing PMML-based change detection models to detect statistically significant changes from these baselines. To date, CUSUM-based and threshold based change detection models [2] have been developed and deployed. Currently, casual models using tree-based classifiers are being developed to try determine semi-automatically whether deviations from baselines are due to chance, unusual weather, special events, or accidents.

Publicly available data and information about the first projects is rather limited due to its confidential nature. On the other hand, data for the third project is publicly available from the web site [16].

9. CONCLUSION

In this note, we have introduced a framework consisting of four steps that can help identify and ameliorate data and information quality problems for complex, distributed systems.

Our assumption is that the data is event based and heterogeneous. In a preliminary step, we divide the data into more homogeneous cells or segments and aggregate the data into feature or summary vectors attached to entities of interest.

1. The first component statistically analyzes each segment and produces a baseline.

2. The second component monitors the event stream in real time and compares computed quantities of each interest in each segment to historical baselines. Deviations result in request for an investigation (an alert). This component detects whether something has happened.

3. The third component is a root cause analysis which seeks to identify the root cause of each alert. This involves subject matter experts. This component determines why something has happened and, if so, what its impact is.

4. The fourth component employs formal models [7] to reduce the likelihood that similar problems will happen in the future.

This framework has been applied in several different domains. In this paper, we discussed two of these: understanding data and information quality problems for payments card transactions and for highway traffic data.

10. REFERENCES

[1] Alan Agresti, An Introduction to Categorical Data Analysis, John Wiley and Sons, Inc., New York, 1996.

[2] M. Basseville and I. V. Nikiforov. Detection of Abrupt Changes: Theory and Application. Prentice Hall, 1993

[3] Tamraparni Dasu and Theodore Johnson, Exploratory Data Mining and Data Cleaning, Wiley, 2003.

[4] W. Edwards Deming, Elementary Principles of the Statistical Control of Quality: A Series of Lectures, JUSE, Tokyo, 1952.

[5] Data Mining Group, The Predictive Model Markup Language, Version 3.0, retrieved from www.dmg.org on March 20, 2005.

[6] DOD Guidelines on Data Quality Management (Summary), retrieved from tricare.osd.mil/rm/documents/fa/DoDGuidelinesOnDataQualityManagement.pdf on March 20, 2004.

[7] David S. Frankel, Model Driven Architecture, Wiley Publishing Inc., Indianapolis, 2003.

[8] Glenn W. Goodman Jr., Taming the River of Data: New Software Tools Fuse Intelligence From Many Sources, Defense News, March 14, 2005.

[9] Robert L. Grossman, H. Bodek, D. Northcutt, and H. V. Poor, Data Mining and Tree-based Optimization, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, E. Simoudis, J. Han and U. Fayyad, editors, AAAI Press, Menlo Park, California, 1996, pp 323-326.

[10] Robert Grossman, Mark Hornick, and Gregor Meyer, Data Mining Standards Initiatives, Communications of the ACM, Volume 45, Number 8, 2002, pages 59-61

[11] Robert L. Grossman and R. G. Larson, An Algebraic Approach to Data Mining: Some Examples, Proceedings of the 2002 IEEE International Conference on Data Mining, IEEE Computer Society, Los Alamitos, California, 2002, pages 613-616.

[12] Robert L. Grossman, Alert Management Systems: A Quick Introduction, in Managing Cyber Threats: Issues, Approaches and Challenges, edited by Vipin Kumar, Jaideep Srivastava, Aleksandar Lazarevic, Kluwer Academic Publisher, 2004.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[14] Yang W. Lee, Diane M. Strong, Beverly K. Kahn, Richard Y. Wang, AIMQ: A Methodology for Information Quality Assessment, Information and Management, December 2002, Volume 40, Issue 2, pages 133–146.

[15] Ken Orr, Data Quality and Systems, Communications of the ACM, Volume 41, Number 2, 1998, pages 66–71.

[16] Pantheon Gateway Testbed, retrieved from highway.ncdm.uic.edu on March 20, 2005. (A SVG plug in for your browser is required to see the map.)

[17] Leo L. Pipino, Yang W. Lee and Richard Y. Wang, Data Quality Assessment, Communications of the ACM, Volume 45, Number 4, 2002, pages 211–218.

[18] Thomas C. Redman, Data Quality: The Field Guide, Digital Press, Boston, 2001.

[19] James J. Rooney and Lee N. Vanden Heuvel, Root Cause Analysis for Beginners, Quality Prgoress, 2004, pages 45–53.

[20] D. M. Strong, Y.W. Lee and R.Y. Wang, Data Quality in Context, Communications of the ACM, Volume 40, Number 5, 1997, pages 1030-110.

[21] Shawn Turner, Defining and Measuring Traffic Data Quality, Proceedings of the Traffic Data Quality Workshop, Washington, DC, December 31, 2002.

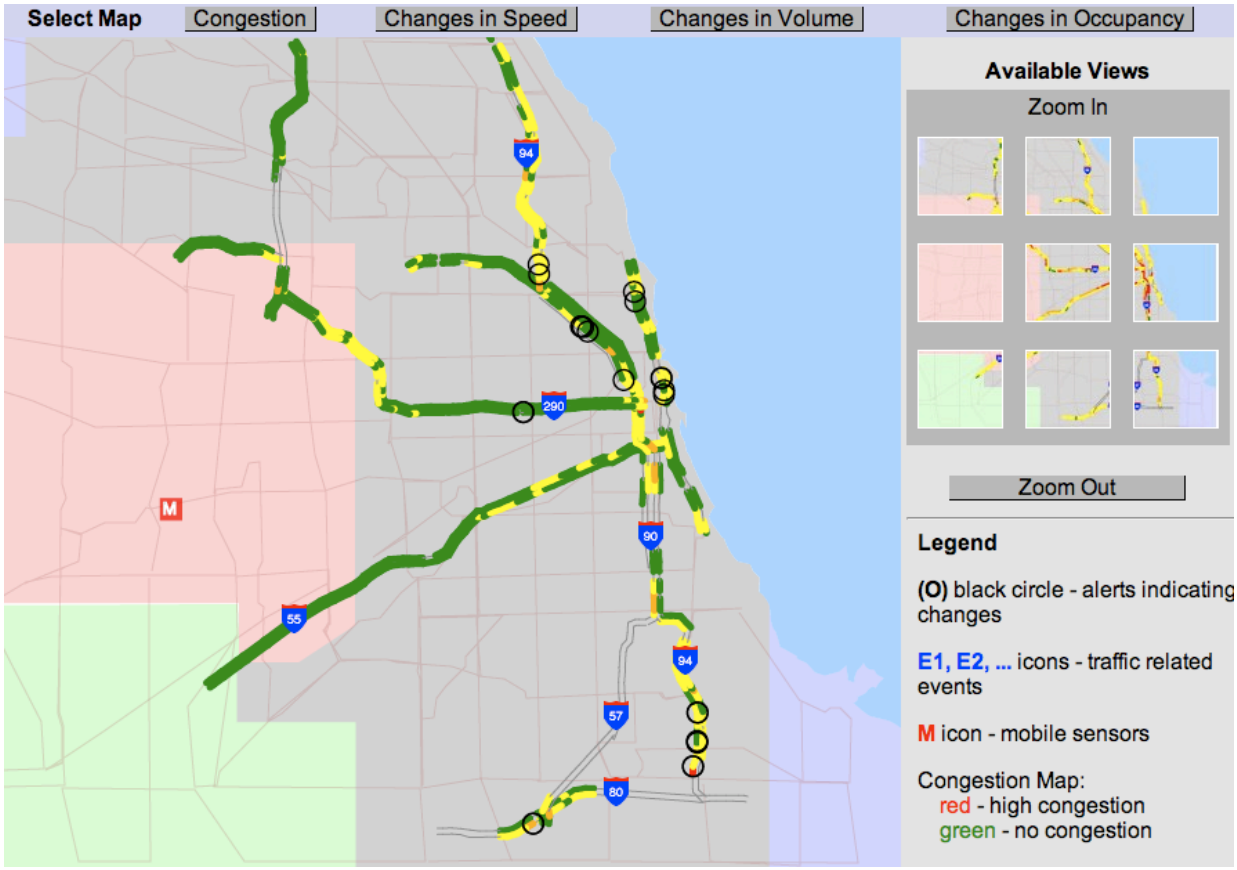

Figure 1: We have applied the framework described here to detect real time deviations from baselines from multi-modal highway data collected from over 800 highway traffic sensors in the greater Chicago region.