A Methodology for Establishing Information Quality Baselines for Complex, Distributed Systems*

Joseph Bugajski
Visa International

Robert L. Grossman†
Open Data Partners

Eric Sumner
Open Data Partners

Tao Zhang
Bearing Point

August 31, 2005 (version 6.1)

This is a draft of the article: Joseph Bugajski, Robert Grossman, Eric Sumner, Tao Zhang, A Methodology for Establishing Information Quality Baselines for Complex, Distributed Systems, 10th International Conference on Information Quality (ICIQ), 2005.

Abstract: We introduce a methodology for improving information quality for complex, distributed event based systems and apply this methodology to an electronic payments system. The methodology consists of five integrated activities: 1) Exploratory data analysis to identify key features of the data. 2) Developing analytical models that detect statistically significant changes from baselines for data fields and attributes derived from data fields. 3) Monitoring baselines and sending alerts when operational data or data derived from operational data deviates from baselines. 4) Root cause analysis to determine why a statistically significant change has occurred and its impact. These models focus on the reasons for change. 5) Developing formal business and technical reference models so that information quality problems are less likely to occur in the future.

Key Words: Data Quality, Information Quality, Baselines, Alert Management System, Change Detection

1 Introduction

In this article, we introduce a framework for analyzing, monitoring, investigating, and ameliorating the information quality of event based data. Here is a motivating example concerning the processing of electronic payments.

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

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

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. Information carried by the data in the transaction that is of poor quality 1) increases the rate of improper declines and improper approvals, 2) increases processing and back office expenses, and reduces the effectiveness of risk management processes. Examples of information quality problems include invalid, incomplete or inconsistent information encoded by data field values in the payments card transaction. These inaccuracies yield poor quality information during computation of aggregates, and summary reports derived from the payments card transaction fields.

There are several challenges that this and related examples present:

1. The data rates are high and the data sizes are large.
2. The data is produced and processed by several different *parties* and this sometimes introduces data and information quality problems.
3. Each party that processes the data usually employs several different *systems and processes* and these different systems and processes sometimes introduce data and information quality problems.
4. 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 exploring, monitoring, analyzing, and ameliorating the information quality for systems with these types of challenges. We also summarize some lessons learned from initial implementations of this framework. The framework has five components: data exploration, building baselines, monitoring baselines, root cause analysis, and amelioration. These are described in more detail in the next section.

We contrast our approach with data quality regimes that detect data errors in event (transaction) records and replace the offending values with those defined as suitable for subsequent computations of aggregates and reports. Although such cleansing improves information presentation, removal of data errors and replacement of these values into event data prohibits measurement of outcomes likely attributable to data errors independently of the business rules that likely "misinterpret" the original intent of the transaction or event.

Although data exploration, causal analysis and amelioration are components for several different data and information quality methodologies [6], [15], [16], as best as we can tell from reading the literature, our paper makes the following contributions:

1. Most data and information quality frameworks [15], [18], [16] do not carefully distinguish between transaction or event level data and summary or feature level data that is derived and 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, [21]). 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, which is the most widely deployed 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.

A preliminary version of this paper appeared in [13].

2 Case Study: Baselines for Payments Card Transactions

We have applied the framework introduced here to several examples, including those involving payments card transactions, highway traffic data, and multi-modal sensor data. In this section, we provide some background on payments card transactions 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 purchases an item at a merchant using a payments card, which is identified by an account number.

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 a financial payment system, such as those operated by Visa and MasterCard. The transaction is processed by the acquiring bank and passed to the payment system.

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

5. The issuing bank processes the transaction 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.

6. For each of these cases, the path is then reversed and the transaction is passed from the issuing bank to the payment system, from the payment system to the acquiring bank, and from the acquiring bank to the merchant.

Data and information quality problems arise from the several different legacy environments involved. Detailed knowledge of legacy data formats is essential to move trillions of dollars of commerce without losing billions of dollars in the process. For instance, Automatic Clearning House (ACH) transactions retain a 80-column format. Credit card authorization messages follow a bit-mapped format (ISO 8583) that differs little from specifications designed 20 years ago. Even newer XML-based payment message standards use conditions that may only be understood by a handful

of payment message insiders. Each payment system independently of the others developed payment message formats because each expected that little information would be transferred between systems. Indeed, many payment systems continue to write legacy specifications,

Here is a question that we would eventually like to answer with this data: Is the number of payment declines from this merchant correlated with data errors arising from transaction record transcription problems?

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 hotels on Fridays at 9 am in June not on a holiday weekend), and to establish appropriate data and information metrics and baselines for each such segment. With this knowledge, understanding data and information quality problems becomes much easier.

3 Overview of Framework

In this section, we provide a high level overview of the baselines-based framework we are proposing for improving information quality.

Here are our assumptions:

1. First, we assume that data is presented to us as a stream of *events* and that these are events are associated with one or more entities of interest. In our running example, the events are payment transactions and the entities of interest include the payments cardholder, the merchant, the acquirer, the issuer, and the payments system.

2. Second, we assume that the transactions and *entities* can be aggregated by one or more dimensions, as is standard in data warehousing. In our running example, dimensions include time, geo-spatial region, type of transaction, etc.

3. Third, we assume that the analysis is based upon *features* that compute derived and aggregated quantities for the events and entities.

This is an example of event based data analysis. Although event based data analysis is becoming more and more common, the framework presented here is the first one we are aware that applies event based data analysis to the problem of data and information quality.

The framework consists of five components:

Data Exploration. The first component of the framework is a data exploration environment that provides an overview of the data. Visualization is a especially important activity.

Building Baselines. The second 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 restricting to appropriate ranges of values along each dimension), the data becomes homogeneous enough to analyze. In this context, we call each such cell a *segment* (as in segmented modeling).

Monitoring Baselines. The third 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.

Root Cause Analysis. The fourth 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 [14]. 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 fifth 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.

4 An Informal Description of Baselines

Note: The examples in this section are hypothetical and only used for the purposes of illustrating how to define baselines.

We begin with an informal description of baselines based upon a simplified example. In this simplified example, assume that one of the data fields being monitored is Point of Sale (POS) Entry Mode and it can assume any of the following (hypothetical) values: 00, 01, 02, 03, and 04. Over an observation period of a week, assume that the frequency of these values for a certain Acquirer is given by the first table in Figure 1. Later, during the monitoring, assume that distribution is instead given by the right hand table in this figure. Although the two first values account for over 97% of the distribution, notice that a new value (05) is now observed. The observed distribution in Figure 2 is similar, except instead of a new observed value, the value 04 is six times more likely in

Value	%
00	76.94
01	21.60
02	0.99
03	0.27
04	0.20
Total	100.00

Value	%
00	76.94
01	21.47
02	0.90
03	0.25
04	0.20
05	0.24
Total	100.00

Figure 1: The distribution on the left is the baseline distribution. The distribution on the right is the observed distribution. In this example, the difference between the two distributions is the presence of a new value (05).

the observed distribution compared to the baseline distribution, although in both cases the values 02, 03 and 04 still as a whole contribute less than 3% of the distribution.

Although this is a simple and familiar idea, in our experience it becomes an effective tool for monitoring data quality in the following manner:

1. Instead of building and monitoring a single baseline, we build and monitor many baselines, say 10^3 to 10^6 baselines. For example, in one application of the methodology described here, we built a baseline for each hour, for each day, and for each location [17]. This generated about $24 \times 7 \times 300$ or over $50,000$ different baselines.

2. Instead of periodic monitoring of baselines, we monitor baselines event by event, using change detection algorithms [2] to decide when there is statistically significant change between the baseline distribution and the monitored distribution.

3. Instead of monitoring single variate baselines, we monitor multivariate baselines. For example, we can look at pairs of values from two fields, triples of values from three fields, etc. A simple example for pairs of values is given in Figure 3.

The tables in Figures 1, 2, and 3 are all examples of multi-variate count distribution tables, which are simply tables showing the distribution of values v_i; pairs of values v_{i_1,i_2}; triples of values v_{i_1,i_2,i_3}, etc. We close this section by recalling a common summary measure for any single or multi-variate count distribution table. Given such a table, we can define the associated entropy

$$H_{i_1,\ldots,i_k} = \sum p_{i_1,\ldots,i_k} \log p_{i_1,\ldots,i_k},$$

where $p_{i_1,\ldots,i_k}$ is the percentage that the value $v_{i_1,\ldots,i_k}$ occurs in the table.

5 A More Formal Description of Baseline Models

As part of the this work, we developed a more formal description of baselines using the Predictive Model Markup Language (PMML) [5]. This is currently under review by the PMML Working Group for inclusion in the standard. Because the methodology described here can generate thousands of different baselines, it became apparent that having a good formalism and interchange format for representing baselines was important.

Value	%
00	76.94
01	21.60
02	0.99
03	0.27
04	0.20
Total	100.00

Value	%
00	76.94
01	20.67
02	0.90
03	0.25
04	1.24
Total	100.00

Figure 2: The distribution on the left is the baseline distribution. The distribution on the right is the observed distribution. In this example, the value 04 is over 6x more likely in the observed distribution, although the two dominant values 00 and 01 still account for over 97% of the distribution.

Value	%
00, -	0.13
00, blank	0.21
06, -	0.01
06, blank	0.01
etc.	etc.
Total	100.00

Value	%
00, -	0.13
00, blank	0.63
06, -	0.01
06, blank	0.11
etc.	etc.
Total	100.00

Figure 3: The distribution on the left is the baseline distribution. The distribution on the right is the observed distribution. In this example, the distribution measures pairs of values, (PIN Entry Mode and PIN Entry Capability) In this example, the pair of values (00, blank) is three times more likely in the observed distributed as in the baseline distribution.

For completeness, we briefly review some of terms used in PMML. A *data attribute or field* is simply an attribute present in the data itself, while a *derived attribute or field* is an attribute derived from the data or aggregations of the data. A *mining attribute or field* is a data or derived attribute that is an input to a statistical or data mining model. A special case of a mining attribute is a *predictive attribute or field*, which is an output of a statistical or data mining model. Finally an *output attribute or field* is a field or table of fields that can accompany the predictive attribute as the output of the model.

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. Any data or derived fields that are used as inputs to a PMML defined statistical or data mining model are examples of mining attributes. An example of a predictive attribute is a fraud score prediciting the likelihood that the transaction will be declined.

Using this terminology, we can now define baselines more formally: A *baseline* is defined by the following:

1. All baselines must specify one or more field values. These are formally defined as PMML mining attributes.

2. Some baselines also specify an outcome field. This is formally defined as a PMML predictive attribute.

3. Some baselines also specify a *condition* expressed in terms of field values. This is formally defined as a PMML derived field. Examples of derived fields that were used include the entropy for the multi-variate tables of counts defined in the section above.

4. All baselines also specify a relationship or model involving 1, and, possibly 2 and 3. This is formally defined as a PMML summary statistic or PMML model. Examples include a) univariate statistics summarizing the distribution of the fields; b) 2 x 2, or, more generally, k x j, contingency tables; c) multi-variate count distribution model (as in the section above); d) regression tree; e) change detection model; etc.

5. Some baselines also specify an output function, such as dollars. This is formally defined as a PMML outcome field.

6. All baselines also specify a test to determine whether the observed condition is statistically different than the baseline or control group (e.g. p value). This is formally defined in the proposal under review as the PMML TestField and TestCondition.

A baseline can be defined for different segments using PMML's segment selection mechanism [5]. For example, a baseline may be defined for each cell using one or more of the following dimensions:

1. time (e.g. daily, weekly, monthly, etc.)

2. business entity, (e.g. merchant, acquirer)

3. country, region, etc.

To summarize, a typical baseline is specified by identifying: a) the cell, b) one or more field attributes, and c) the outcome attribute. Given these, we can construct a model relating the

outcome attribute to the field attributes, collect data for the baseline, and then use a test to determine whether the observed data is statistically different than the model.

To keep it simple, it may be helpful to think of the model as being captured by an XML file (the PMML file). We then collect data and use a statistical test to determine whether the new model is statistically different than the baseline model. If so, we issue an alert. Note that the test to determine whether the observed data is statistically different than the baseline data, as well as related information, is also in the standard.

Example - Change Detection Models Change detection models is 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 variance representing normal behavior and another representing 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]:

$$Z_0 = 0.$$

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

An alert is issued where the Z_n exceeds a threshold.

This example is easily included in the formalized described above.

Example - Contingency Table. Another simple type of baseline model is provided by contingency tables. Recall that contingency tables capture the relation of two categorical variables.

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]. Defining binary indicator variables is a common 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.

Outcome 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.

6 Status and Lessons Learned

To date, we have undertaken several projects using this methodology. Here is a brief description of two of them.

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

Table 1: A simple example of a 2 x 2 contingency table.

Payments card transactions. At Visa International, we have begun to apply this methodology to analyze the adverse effects on business of poor quality data, so called data interoperability problems. Payment data arrives at Visa from more than 24 million merchant locations worldwide, with total annual purchase volume of USD $3.5 trillion, after that data has been processed through risk management rules set by 22,000 individual member banks. These rules determine if a payment authorization request from a merchant either is approved or rejected by the paying bank. Our interests lie with those payment authorizations that are improperly rejected; in jargon, declined or referred. Incorrect, inaccurate, or inconsistent semantic content in authorization request data may cause risk management rules to fire improperly leading the paying bank to issue a decline response to the merchant's payment request. When valid payments are declined, the value of that sale may be lost by Visa to its competitors. Similarly, if the risk management rules fail to fire properly, an invalid approval response may be sent to the merchant. This can result in subsequent exception item processing where charges must be reversed, merchandise returned, formal dispute arbitrated, or a determination that the original transactions were fraudulent. Because VisaNet includes so many different access points and because its data rates are very high, data quality errors leading to inappropriate declines or approvals may enter the system through any of an extremely large number of sources. Thus, it is impractical to define and repair "typical" data quality failures. To isolate data quality problems, we are employing the baseline techniques to reveal patterns of data quality errors and also to determine which sources of poor data quality are causing excessive purchase value losses or exception processing costs. Two examples illustrate the value of these data quality statistical measurement techniques.

In this first example, we defined baseline cells by geography, business identification, type of payment product, type of payment process, and IT support organization. The outcome attribute was the expected value of all declined authorizations. Derived fields related to incomplete, inaccurate and inconsistent records were used. In this way, we discovered USD $2 billion of lost purchase volume due to poor information quality. It is important to note that this represents a small fraction (about 0.06%) of the total transaction dollar volume. An analysis including a code walkthrough revealed that the problem was a coding error in a program consisting of more than 2 million lines. The time required to find the problem and isolate the source was under two weeks. In the second example, there was a very high rate of authorization declines originating at one particular merchant. The baseline rate for declines from similar sources was between 2% and 8% of transactions whereas the suspect case had rates between 70% and 100% of transaction volume. Further manual investigation revealed that the merchant was likely producing fraudulent purchase transactions. The benefits of the baseline approach having been found we are now engaged in perfecting the processes to increase accuracy and timeliness of trouble alert reporting to Visa operating units throughout the world.

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 [17] 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, 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 project is rather limited due to its confidential nature. On the other hand, data for the third project is publicly available from the web site [17].

7 Related Work

There is now quite a bit of research in the field of data and information quality, and several books [19] 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, [15] or [19]).

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 [20], contingency tables [1], discriminant analysis, regression, and classification and regression trees [14].

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. [21] 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 [15], [18], [16], [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 [15], [18], [16], [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.

8 Conclusion

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

Our assumption is that we have a high volume of event based data that is highly 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 explores the data to identify important characteristics and structures in the data.

2. The second component statistically analyzes each segment and produces a baseline.

3. The third 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.

4. The fourth 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.

5. The fifth 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 have focused on how baselines can be built and monitored and illustrated the methodology with a case study involving payments card transactions.

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 June 21, 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] Joseph Bugajski, Robert L. Grossman, Eric Sumner and Zhao Tang, An Event Based Framework for Improving Information Quality That Integrates Baseline Models, Causal Models and Formal Reference Models, Second International ACM SIGMOD Workshop on Information Quality in Information Systems (IQIS 2005), June 17th, 2005, Baltimore, Maryland.

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

[15] 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.

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

[17] 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.)

[18] 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.

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

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

[21] 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.

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

[23] William E. Winkler and Bor-Chung Chen, Extending the Fellegi-Holt Model of Statistical Data Editing, US Census Research Report Series, Number 2002-02, February, 1 2002.