AN ALERT MANAGEMENT APPROACH TO DATA QUALITY: LESSONS LEARNED FROM THE VISA DATA AUTHORITY PROGRAM

(Completed Research Paper)

Joseph Bugajski
Visa International
JBugajski@visa.com

Robert L. Grossman[1]
Open Data Group
rlg1@opendatagroup.com

Abstract: We introduce an end-to-end framework for data quality that integrates business strategy, data quality models, and supporting investigative and governance processes. We also describe a supporting IT architecture for this framework. Finally, we describe how this framework was implemented at Visa International Service Association ("Visa") and some of the lessons learned during its use over the past three years.

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

INTRODUCTION

There is now a relatively mature model, which is usually called the dimensional model of data quality, for characterizing data quality problems in terms of dimensions, such as accuracy, completeness, consistency, timeliness, etc [3]. On the other hand, there is not yet an end-to-end framework that includes the necessary strategic, IT, investigative and governance processes that are required to turn the monitoring of data quality dimensions into a successful data quality program.

In practice, the limiting factor for many data quality programs is the time and costs required to validate data quality problems, to identify root causes, and to ameliorate the problems identified. Because of the time and costs involved, the framework we introduce directly manages data quality alerts and associated investigative processes. We call this approach an alert management approach for data quality.

In short, the challenge is that dimensional models of data quality do not adequately integrate the business objectives and investigative processes that are part of most successful data quality programs.

We believe that this paper makes the following three contributions:

1. We introduce an approach for data quality that integrates business strategy, data quality models, and the supporting investigative and governance processes.
2. We describe an IT architecture that supports this framework.
3. We describe some of the lessons learned from a data quality program at Visa International Service Association ("Visa") that utilizes this framework.

[1] Robert L. Grossman is also a faculty member at the University of Illinois at Chicago.

AN ALERT MANAGEMENT APPROACH FOR DATA QUALITY

An alert management approach for data quality uses statistical and rules-based models to screen event data, update profiles that contain statistical summaries of business entities of interest, and generate alerts that are then investigated by analysts to monitor data quality.

The alert management approach for data quality we introduce in this paper consists of five main steps. We now describe these steps in general and then for concreteness discuss them in the context of one of the case studies described below. In this case study, a Canadian bank had a coding error in their card payment processing software that for certain transactions incorrectly coded the country where the transaction took place. This resulted in some of these transactions being declined.

The first step is to identify an appropriate business problem or opportunity. In this step, we also need to define an appropriate measure that can be used to quantify progress. In the example, the appropriate measure is the approval rate for valid transactions. As mentioned, an incorrectly coded country code field can result in valid transactions being declined. When this occurs, the "interoperability of data" is suspect as that data moves from one processing systems in the financial network to another. We define data quality and information quality problems that result in adverse business outcomes to be "Data Interoperability" issues.

The second step is model development. The alert management framework we propose employs three different types of models:

a) The first type of models are statistical models that quantify the relation between fields that can be monitored for data quality and the business measure of interest, which in our example, is the decline rate.
b) The second type of models are rules-based models that are used to apply business rules to the outputs of statistical models in order to adjust the number of alerts and increase their business relevance and value.
c) The third type of models are architecture models, in particular, architecture models that map business requirements into technical specifications and technical specifications to data attributes in the data being monitored.

For the implementation of this framework at Visa, the statistical model we developed was designed to detect correlations between monitored fields and the decline rate. As described in [1], we in fact developed not just one statistical model, but thousands of them, since in this way, we were able to use statistical models for relatively homogeneous subsets of data. This approach to using multiple models is sometimes called model segmentation in the statistical community. The Visa implementation also made use of rules so that the scores produced by the statistical models could reflect specific business requirements, for example, different regions and different sizes of the banks processing the transactions. The Visa implementation also made extensive use of architecture models, since by directly using architecture models that linked data attributes to high leveler constructs, such as face-to-face transactions or e-commerce transactions, reduced the likelihood of data interoperability issues arising in the first place [2].

The third step is to deploy the models developed. One approach to deploying statistical and rule-based models is to use what are sometimes called scoring engines. These are applications that monitor operational data using XML-based description of statistical models [4] so that changes can be quickly detected and acted on. A monitor employing a scoring engine has two important components. The first component is the scoring engine itself that uses the statistical model produced in the prior step to assign scores to operational data. A threshold is then used to identify data most likely to be poor in data quality and relevant to the measure. This results in alerts. The second component applies business rules to further restrict the number of alerts and to increase the likelihood that an alert results in something actionable and having a significant business impact.

The effective use and deployment of architecture models by developers requires that behavior be changed. For this reason, the successful deployment of architecture models is closely tied to governance. For more details about architecture models and data quality, see [2].

By using appropriate standards, the cost to implement and deploy models can be reduced. In other words, the time and effort to go from Step 2 to Step 3 can be reduced. For this reason, standards are an important component of this framework.

The fourth step is a manual process that investigates the alerts. This begins with a quick assessment of the alert to decide whether further investigation is required. If so, the alert is assigned to one or analysts for a more detailed examination.

The final component is the governance and oversight process. Without an effective governance process, it is unlikely that the alerts identified will be acted on and lead to measurable value. Part of the oversight process is the updating of the dashboard as progress is made towards the identified problem or opportunity.

A summary of the process is below. Figure 1 also contains a flow chart describing the process. The process can be remembered using the mnemonic IMDIG from the first letter of the five key steps – Identify, Model, Deploy, Investigate and Govern.

1. Identify the business problem or opportunity.
    1.1. Determine how to measure it.
    1.2. Create a dashboard to track progress using previously agreed measures.
2. Model – build statistical models that monitor the measures identified; build rule-based models; develop architecture models
    2.1. Collect the appropriate data.
    2.2. Analyze the data.
    2.3. Use the data to build a statistical model.
    2.4. Update the statistical model as required.
    2.5. Develop and maintain rules-based models.
    2.6. Develop architecture models that link the business requirements and the underlying data fields.
3. Deploy – implement the models in operational environments.
    3.1. Use the statistical model to score the operational data. Data may be processed event by event; or, if required, by maintaining state information using signatures.
    3.2. Process events/signatures whose scores are above a threshold using rules.
    3.3. Events/signatures whose scores are still above a predetermined threshold are then processed using rules.
    3.4. Events/signatures that pass the rules are then issued as candidate alerts.
    3.5. Monitor any previously identified alerts
    3.6. Deploy architecture models so that data quality and data interoperability problems are less likely to occur in the future.
4. Investigate the candidate alerts.
    4.1. Perform a preliminary investigation to quickly triage alerts.
    4.2. If warranted, perform an in-depth investigation to identify root causes.
    4.3. Ameliorate any issues identified.
    4.4. Establish a value for the alert and report the value using the dashboard.
5. Govern – support the process with appropriate governance and oversight.
    5.1. Obtain alignment of strategic objective, operational oversight, and investigation process.
    5.2. Report upwards on the meeting of strategic objective and the value generated using a dashboard.
    5.3. Refine operational thresholds, rules, and number of statistical models used to adjust alert workflow.
    5.4. Establish a formal reference model describing the program and approach and a process for reviewing and updating the reference model.

Table 1. This table describes the main steps of the IMDIG process.

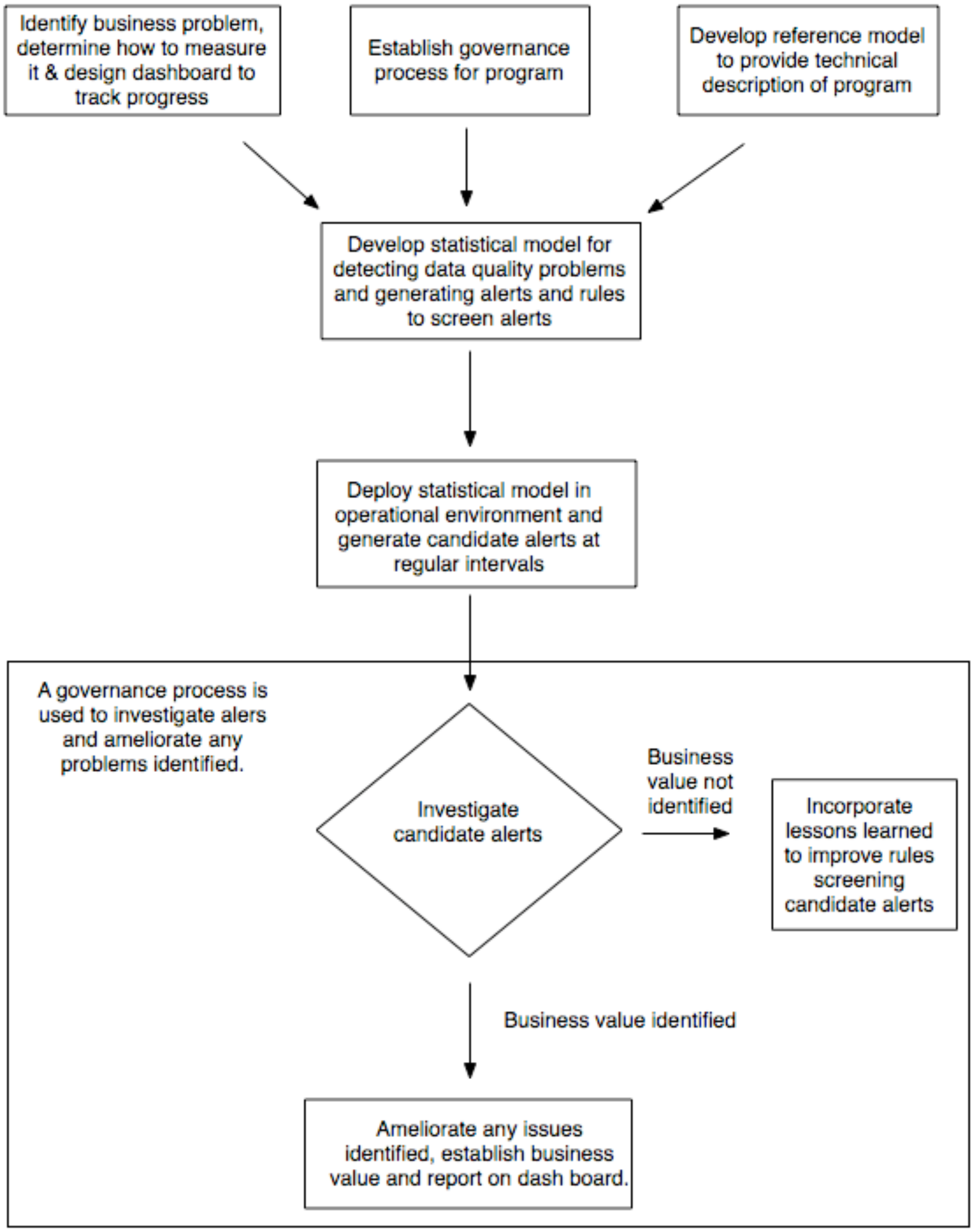
Identify business problem, determine how to measure it & design dashboard to track progress
Establish governance process for program
Develop reference model to provide technical description of program
Develop statistical model for detecting data quality problems and generating alerts and rules to screen alerts
Deploy statistical model in operational environment and generate candidate alerts at regular intervals
A governance process is used to investigate alers and ameliorate any problems identified.
Investigate candidate alerts
Business value not identified
Incorporate lessons learned to improve rules screening candidate alerts
Business value identified
Ameliorate any issues identified, establish business value and report on dash board.

Figure 1. This flowchart provides an overview of the IMDIG process.

An Alert Management Architecture to Support Data Quality

In this section, we describe an IT system architecture to support the IMDIG framework. See Figure 2.

Model Producers and Consumers

The main two system components are a model producer and a model consumer. Briefly, a model producer estimates the parameter of a statistical model and is generally used by someone familiar with data and comfortable working with it. A model consumer is designed so that it can be integrated easily with operational systems and is designed to be operated by someone familiar with IT systems. A model producer and a model consumer communicate though an XML file using a standard called the Predictive Model Markup Language or PMML [4].

Separating the system into two components in this way simplifies deployment since once a model consumer is integrated with an operational system, updating or refreshing statistical models simply requires reading an XML file.

The model producer extracts event data from a project data mart, computes derived attributes and state information from the events (which are sometimes called signature or profiles), persists the signatures, and then uses the information to estimate the parameters of a separate model for each segment.

The model consumer first reads a PMML file specifying the segmentation and the model parameters for each segment. The model consumer than processes a stream of event data and produces a stream of scores. Sometimes a PMML consumer is called a scoring engine since it processes input data using a PMML file to produce a stream of output scores.

Rules Engine

A rules engine is used so that business rules can be used to reduce the number of alerts produced by the Model Consumer. The number of alerts can be controlled in a variety of ways, including:

- by modifying rules or adding new rules;
- by changing the thresholds used in the rules.

As a simple example, a dollar threshold that estimates the approximate business value of the alert can be raised to reduce the number of alerts.

Report Processing Engine

The Report Processing Engine takes the alerts that satisfy all the required rules and arranges them into a report. Another mechanism for managing alerts is to use rules for ordering, formatting and arranging alerts in the reports. For example, alerts can be ordered by the measures used in the dashboard. In our example, we implemented the Report Processing Engine using XSLT so that it was relatively easy to change the format and structure of reports.

Metadata Repository

We stored the various metadata for the project, including the PMML models and rules in a repository. This was important since it was a project requirement that we be able to retrieve the specification for any baseline that had been computed by the project.

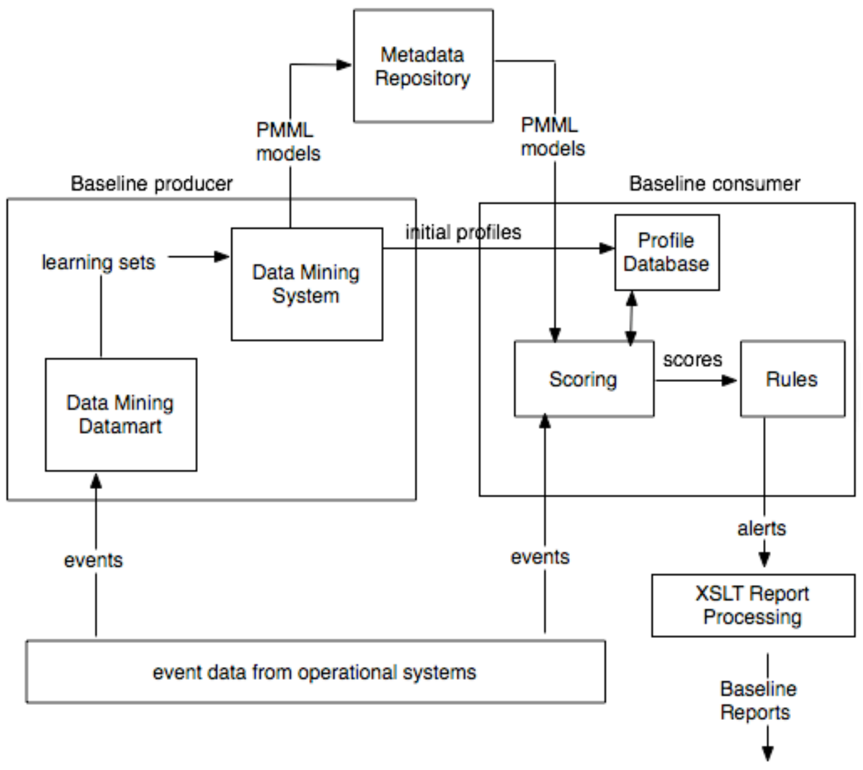

Figure 2. This figure shows a system architecture that supports a IMDIG Program.

The VDA Program at Visa

In this section, we describe the Visa Data Authority (VDA) Program at Visa that is based upon an alert management framework for data quality.

The VDA Program had three phases: architecture, development, and production. Prior to the start of the program three years ago, there was a two-year research effort. One of the main conclusions of this research effort was that data architecture was a primary contributor both to data quality issues and to data interoperability problems. By data architecture in this context we mean the mapping of business requirements into technical specifications, and the mapping of technical specifications to the data elements that eventually appear in the transactional data. The VDA Program approach to data architecture and architectural modeling is described in the paper [2].

Strategic Objective. Critical to the success of the VDA program was identifying its strategic objective and how to measure it. After much discussion, the following objective was agreed upon: identify and ameliorate data quality and data interoperability issues in order to maintain and improve 1) approval rates for valid transactions; 2) disapproval rates for invalid or potentially fraudulent transactions; and 3) correct coding of transactional data. The first two objectives increase the satisfaction of card holders and member banks, while the third objective lowers the overall cost and increases the efficiency of transaction processing. The success of the program was then tracked by a dashboard that monitored the additional dollars processed and the savings resulting from direct actions of the program.

Governance. The Visa Data Interoperability Program was established in 2004 by the global council of the CIO's of Visa's operating units. The program is governed by a global subcommittee of the council of CIOs. This group comprises business executives, business analysts and technical experts who set rules, procedures and processes for the program. Early on, regular meetings were established between the executives in charge of the program and those in operations so that procedures could be developed that assured that data quality and interoperability problems identified in the program were acted upon.

Developing the baseline model. Given the amount of data, our approach was not to develop a single baseline model but rather tens of thousands of them, one for each cell in a multidimensional data cube. For the VDA Alerts described below, we defined a data cube with the following dimensions:

1. The geographical region, specifically the US, Canada, Europe, Latin America, Asia Pacific, Middle East/Africa, and other.
2. The field value or combination of values being monitored.
3. The time period, for example monthly, weekly, daily, hourly, etc.
4. The type of baseline report, for example a report focused on declines or a report describing the mixture of business for a merchant.

Today (June, 2007), for each of 324 field values times 7 regions times 1 time period times 3 report types, we estimate a separate baseline, which gives 324 x 7 x 1 x 3 = 6816. In addition, for 623 field values times 7 regions times 1 time period times 2 report types, we estimate a separate baseline, which gives an additional 623 x 7 x1 x 2 = 8726 separate baseline models. So in total, we are currently estimating 15,542 (=6816+876).

Actually, the description above is a simplified version of what actually takes place. For example, the 6816 baselines mentioned arise from 324 x 7 = 2272 different field values, but the 2272 different field values are not spread uniformly across the 7 regions as indicated, although the total is correct.

Deploying the baseline model. A system supporting data cubes of baseline models was designed and developed using the ideas described above to monitor transactional payment data. For simplicity, call this system the Monitor. The Monitor receives daily samples of tens of millions of authorization messages and clearing transactions from a central ETL facility inside VisaNet. Statistically significant deviations from baselines that are associated with high business value generate what are called Baseline Threshold Alerts.

Investigating Candidate Alerts. Baseline Threshold Alerts are screened by an analyst to produce what are called Baseline Candidate Alerts. Candidate alerts are then analyzed by business analysts and other subject matter experts to understand the issues that led to the candidate alert and to more carefully estimate the business value being lost. If the program team believes that an issue was valid and sufficiently valuable that the cost of repair may be recovered through recaptured revenue or lower processing costs, and, furthermore, they believe that the issue is sufficiently clear that it may be explained accurately to operations and business executives at two or more different firms, they send a Program Alert to the customer relationship manager at the Visa operating region that is closest to the external party that best available evidence indicates is likely the owner of problem. This may be a third party payment processor, an acquiring or issuing bank, or a VisaNet technical group.

The customer relationship manager works with the program analysts to explain the problem identified by the Program Alert to the bank or merchant and to work with them to estimate the cost required to fix the problem. The program team meanwhile reviews measurements to determine when and if the problem is resolved. If the data measurements indicate a resolution, then the business that effected the change is contacted once again to validate recovery of revenue or loss avoidance.

Reference Model. The governance council adopted technical standards for how alerts are generated and business process standards for how alerts are investigated. These rules are recorded in a Reference Model that is maintained by the Program and updated at least twice each year.

Standards. Over time, we developed an XML representation for baseline models and for segmentation. We worked with the PMML Working Group and this work has not contributed to the PMML Baseline and Change Detection Model, which is currently in a RFC status. Over the long term, this should reduce the total costs of the system by enabling the use of third party tools that support this RFC draft standard.

Balancing Manageable vs. Meaningful Alerts. We conclude this section by listing some of the ways that we reduced the number of alerts (so that the alerts become more manageable) or increased the number of alerts (so that the alerts become more meaningful). One of the advantages of the IMDIG framework, is the flexibility provided to adjust the number of alerts.

1. The easiest way to adjust the number of alerts is to adjust the thresholds of the rules in the Rules Engine.

2. Another simple way to adjust the number of alerts is to adjust the number of segments used (in this framework, each segment is associated with one model).

3. The thresholds and parameters of the various statistical models can also be adjusted.

4. The number of alerts can also be managed by changing the rules or adding new rules.

5. Finally, the investigation of alerts often can be effectively controlled by changing the reports, for example, by ordering the alerts in different ways, or highlighting alerts satisfying certain properties.

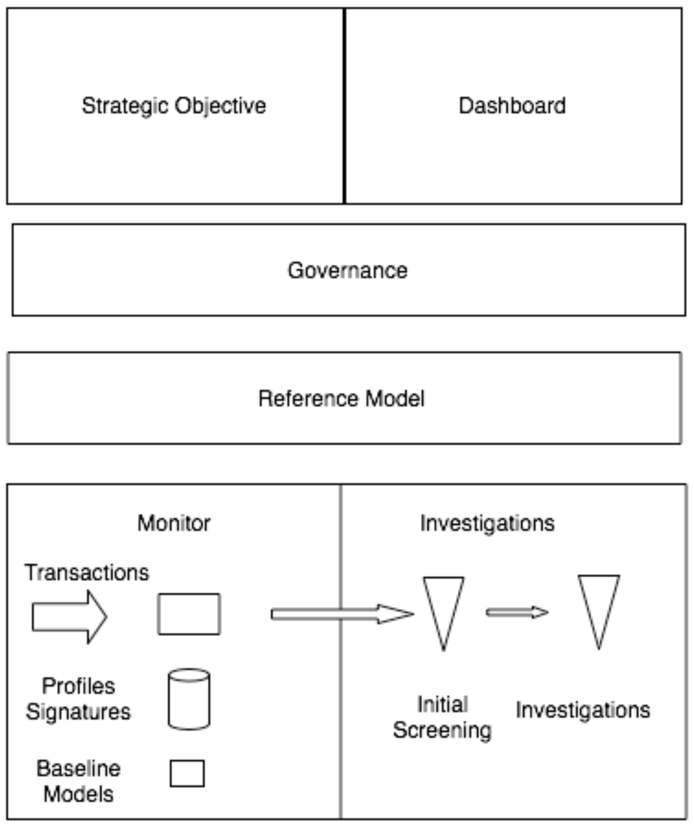

Figure 3. This figure is an overview of the framework used by the VDA Program at Visa.

SOME TYPICAL VDA ALERTS

Overview

Since the program began, Visa and its trading partners have fixed 70 data interoperability issues. The improvement in annual card sales volume realized thereby is about $2 billion, which is nonetheless significant although it is a small number compared with Visa's global annual card sales volume of $4.6 trillion. These "fixed Alerts" comprise 25% of data interoperability issues presently being investigated. In this section, we describe several of these alerts.

Customer Satisfaction Issue : Chip Card Terminal Coding Error

Payment cards with embedded microprocessors ("chip cards") are widely used around the world but standards for chip cards and terminals can be incompatible from one region of the world to another. Chip cards and terminals also differ functionally. If a cardholder uses a chip card at terminal built to an incompatible standard, it usually will not work. The merchant then likely will accept that card for payment by swiping the magnetic stripe on the back of the card across a magnetic strip reading terminal. A more subtle problem occurs when chip cards and terminals follow the same standard but differ in functionality. In this case, the merchant's terminal will read the chip card but the wrong codes will be sent to the cardholder's bank. This will cause the transaction to be declined unless the cardholder or merchant takes extra time and inconvenience to telephone their respective banks. In either case, the payment experience will not meet Visa's standards of excellence, and if the problem were to persist, it could lead to lower total sales using chip cards.

A chip card to chip terminal functionality difference was detected in November 2005 by Visa's VDA Program as a data coding error coming from several merchants serviced by the same US bank. The data analyst who researched the issue noted that most international payments at these merchants were being declined because the chip cards and chip terminals were functionally incompatible. The manager responsible for assisting the merchant's bank enquired about the coding problem and determined that the source of the problem was an ambiguity in a written specification for the chip terminal. A technical letter was sent to all banks that prevented a localized problem from becoming a serious issue.

Lost Revenue Issue : Incorrect Country Code Error

Many risk control systems for electronic payments use time and location information, among a number of other facts, to assess the authenticity of a transaction; e.g., purchase date and time, postal code, city name, country code. For instance, a cardholder who had just used her card at a department store in Edinburgh is not likely to also be using her card at a jewelry store in Beijing. One of these transactions may represent an attempted fraudulent use of the cardholder's information. This case presents a classic dichotomy. The card issuing bank risks losing the entire value of the transaction if they were to accept a payment that later proves to have been inauthentic, but otherwise, it was submitted legitimately by the merchant through their acquiring bank. If the issuer declines the payment then they risk inconveniencing their customer. The only way to be certain that the cardholder made both purchases is to contact the cardholder and enquire about the suspect payment activity. If the bank determines that the location of one of the two payments is invalid, or if reliable supporting information is unavailable, then banks may decide to decline a transaction request. When this happens, and if issuer guessed incorrectly that the transaction was authentic, the cardholder may use another brand of payment card, not complete their purchase, or use cash, thus losing the value of the sale to Visa and the member banks participating in the processing of the transaction.

A Canadian bank had a coding error in their card payment processing software that routinely entered invalid country codes into transactions being accepted by several thousand merchants who banked with them. These errors are quite difficult to detect because every transaction processed by that software has the same error. But, card issuing banks routinely want to know the country where a transaction is occurring in keeping with the rules they established for risk controls. In the absence of valid information, banks often decline such payment requests.

The VDA Program detected the country code error and also showed that the error was accompanied by usually high rates of declined payment authorization requests. These errors came from many merchants who were customers of the same bank whereas similar merchants, who were customers of other banks, did not have the same country code error in their transactions. After the Visa manager contacted the bank, the source of the error was found and the rate at which payments were being approved increased dramatically.

Overpayment Issue : Incorrectly Coded Sales Channel

Banks charge fees to merchants who legitimately accept Visa card payments in return for services provided, additional customer traffic that results from card acceptance, and a guarantee of payment. This last benefit is provided by the card issuing bank that pays for the transaction even if it is found later to have been made fraudulently. In return for accepting this risk, the bank that issued the card receives a small portion of the payment to offset risks of accepting electronic payments. The amount of risk associated with an electronic payment varies according to the nature of the item purchased; e.g., jewelry is fungible where groceries are less so; the location where the card payment was accepted; and sales channel; e.g., store front merchants versus those who do business solely through the Internet. If the payment transaction incorrectly represents any of this information, payments may be declined more often than not, or the merchant or the merchant's bank may pay higher fees to issuing banks than they otherwise would have.

After a Scandinavian bank upgraded their software, many of their transactions were incorrectly identified as higher risk than was actually the case. This resulted in steadily increasing risk offset costs for that bank. The baseline system simultaneously detected a sudden decrease in sales through lower risk channels and a corresponding increase in sales from higher risk sales channels. The data analyst found that the payment data was inconsistent with the true nature of the sales channel for the subject transactions. A meeting with the bank operations team quickly turned this around and kept a problem from growing into a serious issue for the acquirer.

RELATED WORK

Perhaps the most common approach to data quality is to introduce various dimensions of data quality and to measure the adherence of data with respect to these dimensions [3]. The table below contrasts the alert management approach to data quality with the more traditional approach of measuring data along the various dimensions of data quality.

	Dimensions of Data Quality	**Alert Management System Approach to Data Quality**
Strategic Alignment	Not specifically noted	Identify one or more strategic objectives and corresponding measures to determine progress in achieving strategic objectives.
Model	Establish rules to monitor various dimensions of data quality: • Inaccurate • Incomplete • Invalid • Inconsistent • etc.	Correlation Models: • Develop correlation models to measure correlation of events and signatures with identified measures. • Develop rules designed to reduce number of alerts and to prioritize them.
Deployment	Apply rules to operational data and report on percentage of rules that are satisfied.	Create initial set of alerts when scores from models are above a threshold. Create reduced set of alerts by applying rules to initial set of alerts.
Investigative Process	Not specifically noted	Escalate through series of investigations
Oversight & Governance	Not specifically noted	Identify strategic objective, business measures, and dashboard. Set up governance process to oversee.

Table 2. This table compares the approach to data quality described in [3] to the approach proposed here.

SUMMARY AND CONCLUSION

In this paper, we have introduced an alert management approach to data quality called IMDIG. The essential steps in the IMDIG framework are:

Identify – The first step is to identify an appropriate business problem or opportunity related to data quality and an associated measure.

Model – The second step is to build statistical model, rules-based models, and architectural models that relate possible data quality issues to the measure.

Deploy – The next step is to deploy the statistical model, associated rules engine, and architectural models.

Investigate – The fourth step is to investigate the alerts produced by the statistical model and associated rules engine.

Govern – The final component of the framework is to set up an appropriate governance and oversight process.

We described how this framework has been used as a basis for a successful program that identifies and ameliorates data quality and data interoperability problems at Visa.

In future work, we plan to develop algorithms that can set automatically some of the model parameters when more alerts are desired so that the alerts become more meaningful or when fewer alerts are desired so that the alerts become more manageable.

REFERENCES

[1] Joseph Bugajski, Robert L. Grossman, Eric Sumner and Steve Vejcik, Monitoring Data Quality for Very High Volume Transaction Systems, Proceedings of the 11th International Conference on Information Quality, 2006.
[2] Joseph Bugajski and Philippe De Smedt, Assuring Data Interoperability Through the Use of Formal Models of Visa Payment Messages, submitted for publication.
[3] Leo L. Pipino, Yang W. Lee and Richard Y. Wang, Data Quality Assessment, Communications of the ACM, Volume 45, 2002, pages 211-218.
[4] The Predictive Model Markup Language (PMML), www.dmg.org.