The NASD Regulation Advanced Detection System (ADS)*

J. Dale Kirkland, Ted E. Senator, James J. Hayden+,
Tom Dybala, Henry G. Goldberg, Ping Shyr

National Association of Securities Dealers (NASD) Regulation, Inc., 9513 Key West Avenue, Rockville, MD 20850
+SRA International Inc., 2000 15th Street North Arlington, VA 22201
kirkland@nasd.com, senatort@nasd.com, jim_hayden@sra.com

Abstract

The NASD Regulation Advanced Detection System (ADS) monitors trades and quotations in the Nasdaq stock market to identify patterns and practices of behavior of potential regulatory interest. ADS has been in operational use at NASD Regulation since summer 1997 by several groups of analysts, processing approximately 2 million transactions per day, generating over 7000 breaks. More important, it has greatly expanded surveillance coverage to new areas of the market and to many new types of behavior of regulatory concern. ADS combines detection and discovery components in a single system which supports multiple regulatory domains and which share the same market data. ADS makes use of a variety of AI techniques, including visualization, pattern recognition, and data mining, in support of the activities of regulatory analysis, alert and pattern detection, and knowledge discovery.

Introduction

The National Association of Securities Dealers, Inc. (NASD®) has been regulating the securities industry since its founding in 1939. Regulation of securities markets and firms is undertaken by its NASD Regulation, Inc. subsidiary. Our mission of investor protection includes monitoring trading and quotation activities on the Nasdaq Stock Market, the Over the Counter (OTC) Market, and the Third Market, to identify and correct any potential violative activities by over 5500 member firms. Central to this job is the Advanced Detection System, or ADS.

We have been using ADS since the summer of 1997 to provide analysts in the Market Regulation Department with significant leads to potential patterns or practices of regulatory concern, or **breaks**. ADS generates these breaks by integrating and then reviewing all quotation and trade records, almost 2 million on a typical day, for patterns which indicate the occurrence of targeted scenarios.

Since beginning production operations, the system has detected over 7000 breaks, of which more than 10% have merited follow-up actions of various types, a threefold increase in effectiveness compared to previous techniques. More important, it has greatly expanded our surveillance coverage to new areas of the market and to many new types of behavior of regulatory concern.

ADS relies on a rule pattern matcher and a time-sequence pattern matcher. Two- and three-dimensional visualizations allow analysts to see the market context of breaks and temporal relationships of events in large amounts of data. Data mining tools permit discovery of new patterns of potential regulatory interest.

ADS' development is continuing, to remain current with changes in market behavior, to increase its effectiveness, to add additional features, to incorporate additional market data, to cover additional types of potential violations and to keep up with improvements in market structure.

Task Description

Nasdaq is a screen-based dealer market consisting of competing market makers who risk their own capital to provide liquidity. A **market maker** provides quotations for issues in which he makes a market. A **quotation** consists of both a **price** and a **size** (number of shares) at which he is willing to buy or sell, respectively, a particular security, or **issue**. The price at which he is willing to buy is known as a **bid** and at which he is willing to sell as an **ask** or **offer**. The highest bid and lowest ask at a particular time is known as the **inside quote**. Quotations are available to other market makers, to other brokers, and to investors through the distributed computing system that forms the heart of the Nasdaq Stock Market. Trades are executed between market makers (acting as principals for their own account or as agents for customers) and dealers (acting as agents for customers) using one of several automated systems. All trades are reported, shortly after they occur, to a common system, resulting in the well-known stock ticker.

Nasdaq currently has more than 6000 issues on the Nasdaq National Market and the Small Cap Market. There are an average of about 10 market makers per issue. A typical day consists of over 900,00 quote updates, 400,000 inside quote updates, and 800,000 trades on both tiers of the Nasdaq Stock Market and on the OTC Market.

Individual trades or quotes, can often be justified before disciplinary committees. Broad patterns and practices, however, cannot. A key goal of ADS is to detect patterns and practices of violative activity. Another goal of ADS

*The authors of this paper are employees of the National Association of Securities Dealers (NASD) Regulation, Inc. or its contractors. The views expressed herein are those of the authors and do not represent an official policy statement of NASD Regulation, Inc.

was to raise the level of surveillance from issue-based to firm-based patterns and practices.

Our data problem is in part statistical. Statistical techniques are effective at identifying outliers. However, outliers often occur in the context of unusual market activity, while potential violations occur during normal market conditions. Even after a potential concern is identified, an analyst needs to review large amounts of market information to determine if there is a potential explanation for the apparent violation. Prior to ADS, analysts reviewed this information in tabular formats, from which it was difficult to discern relationships. We needed the right mix of statistics, data classification, data visualization, and pattern recognition on a huge database of activity structured in time sequence.

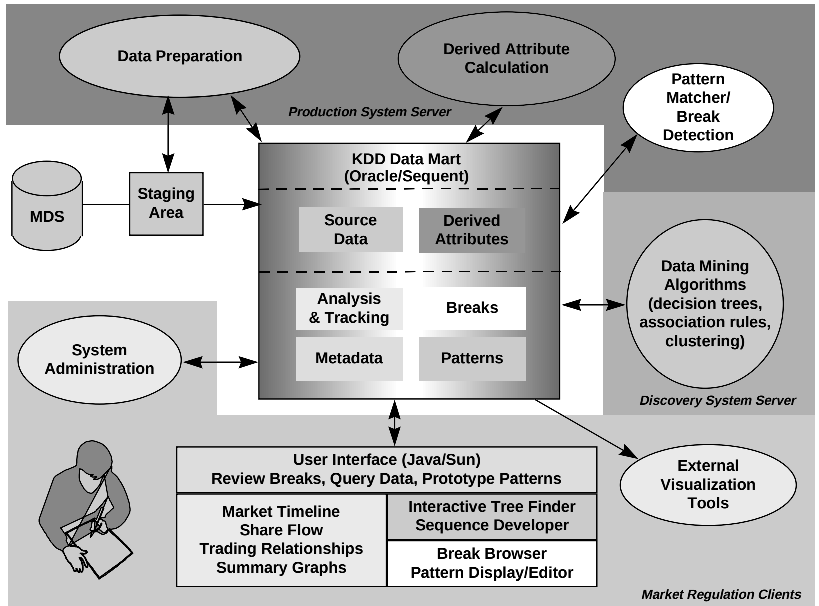

Figure 1 -- ADS Architecture

ADS currently covers three areas, or domains, of potential violations, each with its own user team having a distinct set of business procedures, and needing unique data, knowledge, and tools to perform their function. However, because of the large overlap between necessary and available data and tools, a single system was customized to meet the needs of each team. This approach is allowing us to cost-effectively extend ADS to additional domains as requirements demand and resources permit.

Late Trade Reporting. In order to provide accurate information to the marketplace, trades must be reported within 90 seconds of execution. However, for various reasons, sometimes trades are not reported within this window. The Market Regulation Department is responsible for surveillance of all late reported trades to determine the reason and determine whether or not to initiate regulatory action. Examples which could initiate regulatory action are: A trader delays reporting a large customer trade so the customer doesn't see that there was another trade at a better price at the same time. A market maker has a much higher incidence than his peer firms of reporting trades late.

Market Integrity. The integrity of the Nasdaq Stock Market depends upon free and open price competition between market makers. Some market makers may be dissuaded from competitive pricing by others through a variety of methods of harassment. These methods have been used to "enforce" improper pricing conventions which can result in unfair profits to the market makers at the expense of the customer. Additionally, market makers may coordinate their pricing and trading activity to hide information from other participants and customers, who have a right to it, as a means of influencing prices. The Market Regulation Department is responsible for surveillance of the market with respect to these and any other schemes involving unfair coordination or anti-competitive behavior. Some examples for which the system provides surveillance: After receiving a large customer order to purchase a particular security, a market maker buys the stock from a second market maker, then resells it to the customer at a higher price; the customer could have purchased the stock from the second market maker at the lower price. A market maker stops receiving orders after he narrows his quotes in a security; the orders return when he returns his quotes to a more typical level.

Best Execution. The Best Execution (BE) rule states that the price received by an investor should be as favorable as possible under prevailing market conditions. Such a favorable price is usually achieved when executing a trade within the inside quote. The regulatory enforcement of the BE rule is greatly complicated by market conditions such as relative volatility and liquidity, the size and type of transaction, available communications, accessibility to the primary markets, and quotation sources which may grant exceptions to the rule.

Application Description

This section describes ADS: how it works and what it is. The ADS Architecture is illustrated in figure 1. The key functional modules of ADS are:

- ADS Data Warehouse
- Detection Programs
- Discovery Programs
- User Interface

Data Warehouse

The ADS data warehouse is the heart of the system. It is currently about 240GB. It consists of two sets of tables, referred to as **source** tables and **metadata** tables. The source tables contain market data about trades, quotes, and insides. These tables consist of attributes from systems from which ADS receives information as well as additional attributes and indices that are calculated for use by other components of ADS, as well as summary and profile information about issues and firms. Metadata tables hold set-up and execution control parameters for the break detection and discovery jobs, the rule and sequence patterns that are matched by the detection jobs, the results of the break detection and discovery jobs (breaks and rules), and data necessary for various user interface components.

Data load and preparation programs update the data warehouse daily. They combine information from the source tables and calculate additional attributes deemed useful for detection and discovery. The data sources for ADS present a view of the market as a series of transactions with separate trades and price updates. Matching these transactions to produce a more coherent market state is a major computational effort, but is essential to provide the derived attributes which capture market context.

Detection Programs

ADS Detection programs consist of two pattern matchers: a Rule Matcher and a Time Sequence Matcher, which are run weekly to generate breaks. Breaks are assigned to analysts through distinct automated break assignment modules which reflect each user team's work process.

Sequence Matcher The sequence matcher is a program for finding instances of temporal patterns in databases. It seeks triggering events that, in turn, seek other events in either forward or reverse time sequence and order that could indicate deliberate behavior of concern. The algorithm was independently developed at SRA, drawing upon work in discovery of temporal associations. (see Mannila 1995)

The sequence matcher algorithm works by querying the metadata tables to load a pattern into memory. The sequence matcher algorithm is similar to a regular expression matcher. It maintains a list of potential match states. At each step, a row is fetched and a new state is started for each pattern. Existing states are advanced if they match the constraints on the pattern location they are currently at. When a state reaches the end of a pattern, it is a match. The sequence matcher may be run in forward or reverse mode, fetching rows in increasing or decreasing time order depending on whether the triggering event for the sequence occurs before or after the other necessary conditions. In a single pass, multiple tables may be scanned for several patterns concurrently. The sequence pattern language uses a syntax and precedence similar to the C programming language.

There are three types of inputs to the sequence matcher: target configurations, patterns, and callback functions. A target configuration details the database columns to be processed, along with any row-ordering conditions. Patterns are specified either from the metadata tables or a text file. Callback functions are C++ functions compiled into a dynamically loadable object. They may be time filters or actions. A time filter is a way of coordinating the ordering of rows from heterogeneous tables. Output of the sequence matcher is determined by the action functions that are called. The break generation action function populates the metadata tables with breaks. The break generation callback inserts a new row into a table for every match found. Each matching state has some number of rows that are the instantiation of the pattern.

Rule Matcher The rule matcher fetches trade data and produces breaks based on the detection of repeated instances of pre-defined behaviors, represented formally as rules with an antecedent and an optional consequent. Each rule has two measures of strength, called confidence (fraction of rows satisfying the antecedent where the consequent is also true) and support (fraction of rows in the entire table that this rule holds on), which are used as parameters to ensure that breaks which are generated by the rule correspond to significant patterns of activity. An example pattern would be one that looked for firms showing a high percentage of trades involving more than 10000 shares that are designated late.

The Rule Matcher internally represents each attribute that is mentioned in at least one pattern and uses the attribute names to build the query that will retrieve the targeted trade data. The patterns are represented as conjunctions tests on attributes. Trees are created that contain the pattern conjunctions and counts corresponding to the number of times the conjunction is detected.

Each record retrieved is given an internal representation that facilitates tree traversal based on attribute tests passed. If an attribute has a bind test defined for it a new test is added whenever a new value for the attribute is encountered in the data. The record representation is given to the tree structures and all necessary conjunction counts are updated. The output of Rule Matching results in storing breaks and break-related data in the database.

Discovery Programs

ADS includes parallel and scaleable decision tree and association rule implementations which can be used to discover new rules reflecting changing behaviors in the marketplace.

Association Rules The association rule algorithm is a procedure for generating rules from a table. (see Agrawal, et. al. 1993) Our implementation of the association rules algorithm is written in C++ and performs direct access to an Oracle8 RDBMS through a database class library invoking the Oracle Call Interface (OCI) API. Parallelism is achieved using an implementation of the Message Passing Interface (MPI). The algorithm uses parallelism in several places to divide up tasks. Each process is CPU-intensive and allocates its own memory.

Attribute filters are used to reduce the number of association rules reported by the algorithm. They provide guidelines about how certain attributes make a rule interesting or not interesting. There are four types of attribute filters that add the capability to include or exclude attributes, group attributes in rules, and specify functional dependencies. These filters are instrumental in reducing the number of redundant and definitional rules generated.

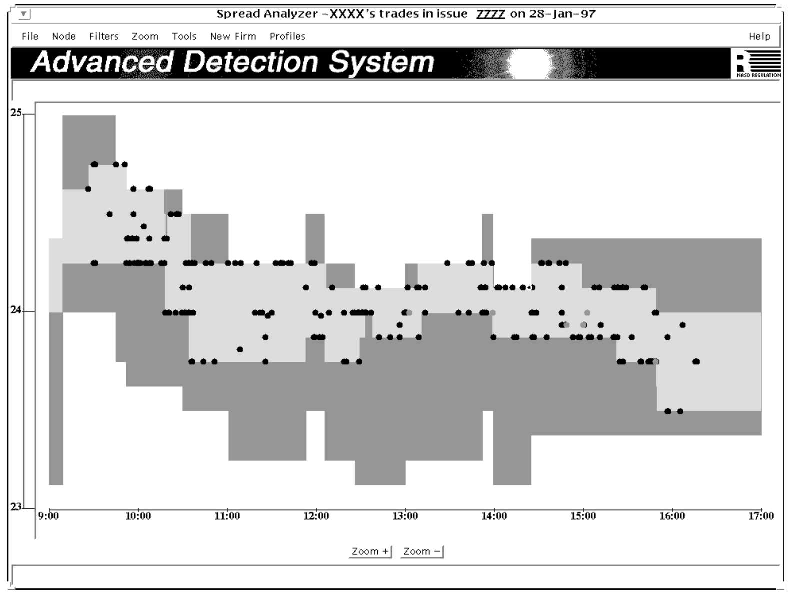

Figure 2 – Market Spread Timeline

Decision Trees The decision tree algorithm is a data mining tool designed to find patterns with respect to a single specified data column called the "dependent attribute." (See Quinlan 1993) The input data consists of a number of "examples" each of which is a vector of attribute/value pairs. The algorithm outputs a set of rules that use the independent attributes to predict or characterize values of the dependent attribute. These rules have a conjunction of independent attributes on the left-hand-side and contain only the dependent attribute on the right-hand-side.

Rules are extracted from a decision tree by tracing the path from a leaf up to the root of the tree. The dependent attribute value assigned to that node become the right-hand-side of the rule, and the independent attributes and values in the nodes on the path to the root become the conditions on the left-hand-side of the rule.

Rules are pruned by dropping conditions from the left-hand-side and seeing if a better rule is produced. If a rule has N conditions in the left-hand-side, then dropping each condition can produce N possible alternative rules. Each of these rules is evaluated, and if any are better than the original rule, then the best alternative is chosen and the process is repeated. A rule's quality is determined by calculated using its normalized estimated net worth. Once each rule is pruned, duplicates are removed.

Rule Management. A run of decision trees or association rules against a data sample of one firm or one security usually produces several rules; thousands of rules may result from runs against all securities and firms, necessitating automated rule management capabilities. Moreover, to follow dynamic behavior of the market the pattern base must be highly adaptive, allowing additions refinements, and deletions. The Rule Management module supports transformations of discovered rules to break detection patterns, similar to the approach described in [Fawcett and Provost 1997]. Rule management comprises three operations: filtering, refinement, and deletion.

During filtering, discovered rules are pruned to a subset which are unique (not repeated among themselves), new (different from existing patterns and rules), have high confidence, and good support in data. Filtering compares the generality of two rules by examining similarity of the rules' conditions and conclusion (Michalski 1983, Tecuci & Duff 1994). The comparison algorithm uses domain specific and domain independent heuristics to capture the meaning of various rule attributes. The filtering process decreases the number of rules from thousands to hundreds, which are still firm or security specific.

During refinement, rules obtained from the filtering process are refined by generalizing their conditions and ranking them against random data samples, to generalize rules by dropping market participant or security specific information, to perform generalization according to the specified refinement heuristics (Dybala at el. 1995), and to rank the obtained rules by testing them against various data sets to refine their support and confidence thresholds. Finally, the best performing rules are selected. Rules which pass the refinement process are unique, general, and different. The result of the refinement process are several or possibly tens of new rules which are presented to the analysts for a review, for a final selection for promotion to active break detection patterns.

User Interface

The ADS user interface consists of screens for break processing and management, tabular displays for viewing detailed trade and quotation information associated with a particular break, two-dimensional graphical displays that allow an analyst to easily visualize market activity at the time of a break, and three-dimensional displays that are useful for viewing large aggregates of information.

The market context around the time of a suspected violation in a security is determined by three things: the trades in that security, the bids and asks of the market makers in that security, and the inside bid and ask. The difference between a bid and an ask price is called the spread. This context is captured visually in the market spread timeline (see Figure 2). A market maker's bid and ask in an issue are plotted against time in one color and the inside bid prices and ask prices in another. Trades are displayed as dots. Details of trades and quotes are available through a drill-down capability. The visual display allows analysts to quickly identify important events such as the inside spread being narrowed, multiple small trades being executed against a market maker, or a market maker buying up a lot of shares in an issue.

The share flow display (Figure 3) shows a group of trades in order by execution time. Each trade is represented as an arrow from the buyer to the seller marked with the number of shares and price. When a group of trades is displayed by time, patterns of share flow are visible. A firm may collect shares through multiple medium-sized buys, and sell them in one large sale. Shares may pass through multiple firms before reaching their final destination. These patterns of share flows are crucial for analyzing potential violations.

Detecting any potential conventions or regularities in market behavior which might indicate improper coordination by market participants requires the ability to rapidly review large amounts of market data for patterns and anomalies. To provide this ability, Visible Decisions Inc. Discovery software was selected for three-dimensional displays. (See Martin96) Two three-dimensional "landscapes" have been developed. The Pricing landscape (see Figure 4) displays a large amount of summary data regarding quotation activity by multiple market makers in many issues. It highlights the possible pricing conventions in portions of the market. These may be agreements among market makers to quote only in specific price intervals. If these agreements are enforced by peer pressure or harassment, they become anti-competitive and are violative practices. A visual examination of this display permits an analyst to rapidly determine patterns of quotations in an issue and similarities between the quotation behavior of different market makers. Situations where the pricing conventions hold for all but a few market makers are those most likely to result in anti-competitive behavior and warrant the closest review. In addition, new conventions are more easily visible, since the display may be adjusted by various thresholds and filters.

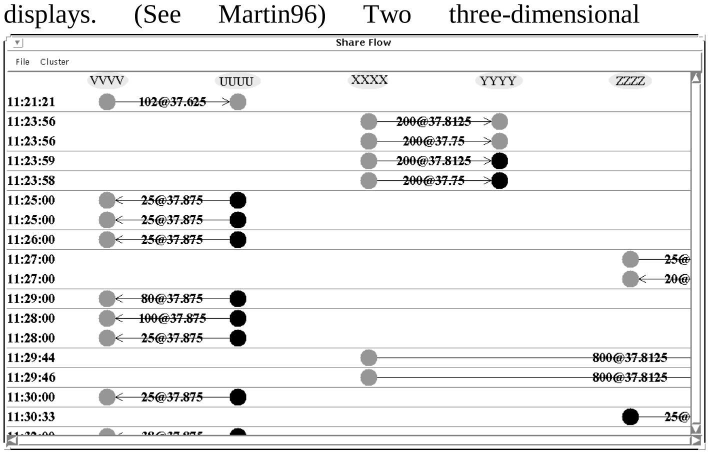

Figure 3 -- Share Flow Display

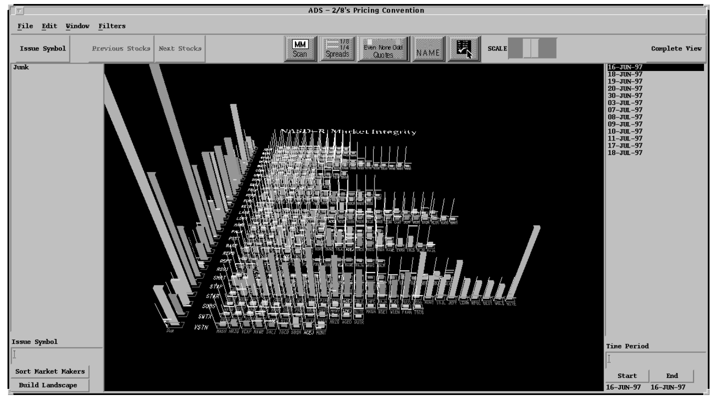

Figure 4 -- Pricing Landscape

The second display, called a Spread landscape (see figure 5), allows an analyst to view the quotation and trading behavior of a set of market makers in a particular issue over a specified time period. This display may be viewed as a generalization of the two-dimensional spread display described above.

Hardware and System Software Environment.

ADS system hardware currently consists of a 12-processor Sequent NUMA-Q host computer acting as the production database server. Scaleability and growth were significant factors in the choice of a server platform. Sun SparcStations serve as user workstations running the ADS client as well as other applications. A second, smaller Sequent NUMA-Q is used as a development and knowledge discovery platform, and a variety of file servers are installed on Sun SparcStations. The operating systems are varieties of Unix. Oracle v.8.0.4 is the RDBMS. The ADS client is written in Java, which has proven to be an effective language for both prototyping and final implementation.

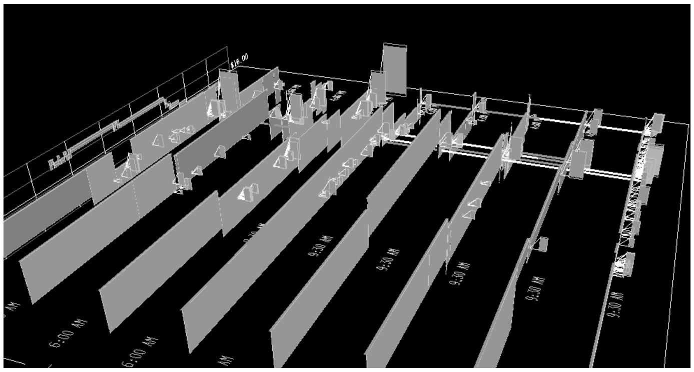

Figure 5 – Spread Landscape

Uses of Artificial Intelligence Technology

ADS integrates data mining (decision trees and association rules), pattern matching (rules and time sequences) and visualization techniques in a single large-scale application, as detailed in previous sections. It represents an advance over previously reported applications because of the large scale of the application, the combination of discovery and detection components with the ability to discover new rules and promote them into the detection system, the explicit representation of rules in the database for use in detection and as a result of discovery, the development of the time sequence pattern matcher, and the direct database access parallel implementations of the detection and discovery components.

Commercial Tool Evaluation

We evaluated off-the-shelf knowledge discovery/data mining tools during the proof-of-concept phase of ADS development. We surveyed more than 20 off-the-shelf KDD products and conducted a detailed evaluation of two of them. But none of them appeared to meet our needs for a system that would do the following, so we developed the components as part of a custom application[1]:

- function in NASD's hardware and software environment
- have an open architecture for integration with other necessary system components, especially the database management system and the to-be-developed workflow management components
- function in a highly dynamic environment
- integrate a comprehensive variety of methods
- provide intermediate results to all components
- scale to handle production volumes of over one million trades and quotes per day; and, most important
- provide an integrated structure for ongoing detection of improper activity combined with analysis to identify new patterns of potential regulatory interest.

[1] SRA has taken the custom application developed for NASD and converted it into a generic KDD toolkit that meets all these requirements and can be applied to other application domains. Details of this product are available at www.knowledgediscovery.com on the web.

Related Applications

The FinCEN AI System (FAIS), described in [Senator, Goldberg et. al. 1995] motivated some of the basic ideas of ADS, although the specific requirements differ. Both are instances of a type of fraud detection system called Break Detection Systems described in [Goldberg and Senator 1997], which also contrasts the two domains along several characteristics. ADS advances over FAIS along several dimensions, in particular, its much larger data volume, its incorporation of automated discovery techniques for identification of new patterns of potential regulatory interest, its explicit representation of multiple domains and user groups, and its use of a time sequence pattern matcher for detection.

[Fawcett and Provost 1997] describe an approach to cellular telephone fraud detection that automatically learns indicators of potential fraud from a large database of transactions. It uses the indicators to create monitors which profile legitimate and anomalous behavior; and combines the output of the monitors to generate alarms. This system employs more automated learning techniques than ADS and it focuses on identification of fraudulent transactions, while ADS looks for multiple types of violations and for patterns and practices of these violations in the context of an integrated application.

Application Use and Payoff

The development of ADS has taken the Market Regulation Department to a new level in monitoring the markets we are charged to monitor and regulate. For a number of years the Department has had systems that point to potential instances of regulatory concern on market data. And we have a history of taking disciplinary or enforcement action on those in which action warranted. But prior to ADS, we have not had a system to help us detect potential violative patterns and practices in quotation and trade data. We depended upon human analysts to recognize patterns and practices as a series of activities became known to them. Now we search proactively through all of our quote and trade data for those pattern or practices of regulatory concern. Individual instances of violative activity can be explained away by a perpetrator, but strong evidence of a pattern or practice is more difficult for a perpetrator to defend.

What is the payback? First, we have seen notable success in the area of late trades, the first of the areas of concern targeted by ADS. We have increased the hit ratio of good breaks, or leads, to overall breaks by a factor of three over the best of our prior approaches. This permits the analysts to spend less time on efforts that expose no regulatory concern. It permits all the analysts, those with much experience and skill and those with less experience and skill, to be effective at a higher level. It enables new analysts to develop a level of sophistication much more quickly -- about half the time. We have been able to bring many more actions in this area than previously. Second, we have been able to establish a new team to monitor concerns in market integrity. We have been able to point to potential anti-competitive, harassment, or collaborative activity in a way not possible previously. This team is now able to proactively surveil the market instead of reacting to customer complaints. Third, we have started a pilot in a third area to monitor firm responsibilities for best execution. We have had this kind of monitoring in the past, but not in the form of detecting patterns and practices. Fourth, we have the visualization tools that permit the analysts to "see" things that our processing may miss. Fifth, we have a system that can be adapted quickly to new market realities, or changes in activity. Sixth, we monitor the data thoroughly. We do not sample. We run tests exhaustively over all data. This has required quick algorithms and quick machines to handle a truly large amount of data.

From June 1997 through February 1998 ADS has generated over 7000 breaks that have resulted in more than 800 follow-up actions of various types (such as requesting records from the securities traders involved or referral to other units of NASD Regulation). This rate, over 10%, is a factor of three increase over previous break detection systems. Another 80% have been closed, yielding valuable experience which has been re-applied to the detection process. The apparently high “false alarm” rate is acceptable because breaks can be closed quickly when no action is warranted, and because the trade-off cost of missing a real violation is extremely high, corresponding to a “zero-tolerance” for major violations. Further, the process of investigating a break, even a false alarm, is a significant deterrent to violative behavior and typically results in market improvements.

A key payback for a surveillance system is the degree of coverage – in terms of the number, type, and detail of potential violations we can monitor and in terms of the amount of market data we can review. Our initial estimates of improved surveillance coverage with ADS compared to manual surveillance is a factor of about 225. We have also seen 75% reductions in the complexity of some surveillance protocols (corresponding to 300% productivity improvements), as well as significant reductions in potential violations in the areas of both late trade reporting and market integrity, corresponding to an improved market for all investors.

Application Development and Deployment

The ADS project team consisted of staff from NASD Regulation, Inc. (Office of Technology Services and Market Regulation Department), NASD Production Services Department, and SRA International, Inc. At its peak, the team consisted of approximately 22 people.

Table 1 lists key ADS development milestones.

ADS began as a proof-of-concept in the area of late trade reporting. The project was initiated and a team assembled in April 1996. Scenarios and corresponding patterns were identified and rule matching and discovery algorithms were developed, resulting in a demonstration that this approach could be effective in improving late trade surveillance and a July 1996 project Steering Committee decision to proceed with a pilot on live data. This pilot was deployed in October 1996. During the summer of 1996, the market integrity area was determined to be of key importance, and the decision made to expand the Late Trade Pilot into ADS during 1997. The time sequence matcher was developed during 1997 for the needs of MI and for some LT scenarios that could not be adequately represented by rules. ADS went into full production on 31 July 1997.

ADS was one of the first production implementations of Oracle 8, undergoing conversion within a month of becoming operational. Because of the success of late trade reporting and market integrity, it was decided to add the best execution domain and work begin in late 1997 to develop patterns and scenarios. Continued improvements in functionality based on feedback from the users and in meeting NASD Production Services standards for a well documented and predictable system was a major emphasis in late 1997 and early 1998. Release 2.0 in spring 1998 formally recognizes the distinction between the ADS application and SRA’s KDD Explorer Toolkit which serves as the underlying discovery and detection engines.

The separate domains came on line at different times. Each domain began as an operational pilot, with live data generating experimental breaks, so the patterns could be tuned appropriately and analyst feedback could be incorporated into the development. The Late Trade domain analysts have been evaluating breaks since October, 1996, as a pilot project. The Market Integrity team began work in July, 1997, with the entire system beginning full accountability in August, 1997. An application of ADS to the enforcement of the Best Execution rule is currently in a pilot stage. The system generates experimental breaks for some types of trades. The breaks are reviewed by analysts who recommend further tuning of patterns. The first production application of ADS to enforcement of the Best Execution rule is expected in early Spring 1998.

Knowledge Maintenance

Because ADS applies to multiple dynamic domains, knowledge maintenance is a key issue. Knowledge maintenance is enabled by processes and tools. Weekly meetings are held with key users in each domain area to review current breaks and pattern performance. At these meetings, new scenarios are discussed and prototype patterns are evaluated for inclusion in the system. Tuning of operational parameters is done at these reviews, so that the quantity of breaks is consistent with the analysts' ability to evaluate them. As break quality improves, thresholds can be adjusted to allow more, marginal breaks, as well as allowing new patterns to be detected.

A combination of manual and automated discovery was anticipated, with the emphasis early in the project on manual specification of detection patterns in order to "jump start" the project and make use of the Market Regulation department's expertise. As ADS matures and we reach the limits of what analysts already know, we expect automated discovery to play an increasing role in shaping the system's knowledge component.

The application includes tools to introduce new and modify existing rule and sequence patterns. The sequence editor provides a graphical structure for modifying the more complex specifications of multi-input, non-deterministic, temporal patterns.

The ADS user interface provides features for managing “experimental” patterns, which are run in production but are not yet validated as producing breaks of regulatory value. The use of experimental patterns allows us to evaluate newly proposed or modified patterns on current data, ensuring they stay current with market conditions.

Finally, we anticipate the regular addition of new domains to ADS, requiring new sets of patterns corresponding to the types of potentially violative behavior of interest.

April 1996	Late Trade Pilot: project initiation
July 1996	LTP Proof-of-Concept
October 1996	Late Trade Pilot (release 1.0)
January 1997	Market Integrity project initiation
June 1997	Late Trade Production / Market Integrity Pilot (release 1.1)
July 31, 1997	Market Integrity Production (release 1.2)
September 1997	Oracle 8 Conversion
January 1998	Best Execution Pilot
January 1998	ADS Release 1.3
April 1998	ADS Release 2.0

Table 1: ADS Development Milestones

Acknowledgments

We thank all our colleagues at NASD and SRA who aided in the development of ADS or contributed to its knowledge bases, especially our Executive Sponsors and our Steering Committee. Most important, we thank Mr. Jim Cangiano, Senior Vice President of Market Regulation, whose vision and support made the project possible.

References

Agrawal, R., Imielinski, T., and Swami, A., "Mining Association Rules Between Sets of Items in Large Databases", In *Proc. ACM SIGMOD Conf. On Management of Data*, pp. 207-216, 1993

Dybala T. 1996. Shared Expertise Model for Building Interactive Learning Agents. Ph.D. Dissertation, George Mason University, Fairfax, Virginia.

Fawcett, Tom And Provost, Foster, “Adaptive Fraud Detection” Data Mining and Knowledge Discovery 1, 291-316 (1997).

Goldberg, Henry G. And Senator, Ted E., “Break Detection Systems” in AI Approaches to Fraud Detection and Risk Management: Collected Papers from the 1997 Workshop, Technical Report WS-97-07, AAAI Press.

Mannila, Heikki, Toivonen, Hannu, and Verkamo, A. Inkeri, “Discovering Frequent Episodes in Sequences,” *Proc. of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, August, 1995.

Martin, James, Beyond pie charts and spreadsheets, *Computerworld*, May 27, 1996.

Michalski R.S. 1983. A Theory and Methodology of Inductive Learning. In Michalski R.S., Carbonell J.G. and Mitchell T.M. (editors) Machine Learning: An Artificial Intelligence Approach, vol. 1 Tioga Publishing Co.

Quinlan, J. Ross, *C4.5 Programs for Machine Learning*, Morgan Kaufamn, San Mateo, CA, 1993.

Senator, Ted E., Goldberg, Henry G., et. al., “The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions,” *Proc. of the Seventh Innovative Applications of Artificial Intelligence Conference*, Montreal, Canada, August 1995.

Tecuci G. and Duff D. 1994. A Framework for Knowledge Refinement through Multistrategy Learning and Knowledge Acquisition. Knowledge Acquisition Journal, vol. 6 No. 2. pp. 137-162.