*Journal of Universal Computer Science, vol. 14, no. 9 (2008), 1391-1410*
*submitted: 30/4/07, accepted: 19/2/08, appeared: 1/5/08 © J.UCS*

# NASDAQ Velocity and Forces: An Interactive Visualization of Activity and Change

Huyen Tue Dao
(University of Maryland, College Park, MD 20742-3360, United States
daotueh@umd.edu)

Adam Bazinet
(University of Maryland, College Park, MD 20742-3360, United States
pknut777@umiacs.umd.edu)

Robin Berthier
(University of Maryland, College Park, MD 20742-3360, United States
robinb@umd.edu)

Ben Shneiderman
(University of Maryland, College Park, MD 20742-3360, United States
ben@cs.umd.edu)

**Abstract:** NASDAQ Market Velocity and Market Forces are two relatively new data products that attempt to capture market sentiment, something that was previously only observable if one was on a trading floor. Given the transient and temporal properties of the data, we were challenged to create a visualization that would highlight the ever-changing qualities of Velocity and Forces. To that end, we developed FireStox, a web application that provides unified representation and filtering solutions to help market researchers observe the behavior of these metrics for one or many companies throughout the course of a trading day.

**Keywords:** Stock Market Visualization, Stock Market Data Analysis, NASDAQ Velocity and Forces, Graphical User Interface, Information Visualization
**Categories:** J.4

## 1 Introduction

Through two of its data products, Market Velocity and Market Forces, the NASDAQ stock market tries to capture trading information in its fully automated system that was provided in physical floors by physical noise and activity. Market Velocity and Market Forces describe the activity level of orders in NASDAQ and the direction of the activity (buy or sell) in order to provide traders a way of recognizing short-term and long-term trends and events in terms of price and volume. These measures are provided as a large data feed generated by NASDAQ throughout the trading day. Some work has been done in visualizing Velocity and Forces with simple line graphs, data listings, and visualizations for display on the NASDAQ Marketsite. However, there is currently lacking a visualization that allows users to see the activity in the market as a whole and visually explore the Velocity and Forces data. The goal of this project is to provide a tool that will allow users to see how Velocity and Forces change throughout the day for a select number of stocks, and to associate trends in

Velocity and Forces with trends in long-established data feeds such as price and volume. We wanted to provide an overview of stocks on the NASDAQ and give users a useful tool to discover events and review the behavior of the market.

### 1.1 Stock Market Activity

Stock markets basically take one of two forms today: physical trading floors and electronic markets. Exchanges that have a physical trading floor, like the New York Stock Exchange (NYSE), have various trading posts at particular locations for trading of a particular company's stocks and use an auction system to determine price. Brokerage firms submit orders via brokers on the trading floor who go to the appropriate trading post to make their buys and sells. At each trading post is a member of the exchange known as a specialist. The specialists facilitate trades by matching buy and sell orders, buying shares if there is no current buyer, and selling shares if there is no current seller. His/her purpose is to help keep the market flowing smoothly and continuously. Even with this high level of face-to-face interaction, markets with physical trading floors are still highly computerized.

Electronic markets, like NASDAQ, are fully-automated systems. In electronic markets, brokers contact dealers, known as market makers, in order to determine the best prices for a particular stock depending on if the broker wanted to buy or sell stocks. The market maker, like the specialist, also responds to current conditions in the market; however, there are several market makers per stock and their goal is profit. If there is predominantly selling occurring, then the market maker lowers the bid price, thus lowering the stock price. If more investors are buying than selling, the market maker raises prices for shares in his own inventory, thus raising the stock price. Market makers are continually adjusting prices depending on the volume traded as well as trends towards buying or selling. The goal of market makers is to maximize the price a seller pays and minimize the price that he/she must pay. The difference between the bid price and the sell price is known as the "spread" and is generally on the order of cents. In this market, the same stock can have different prices under different market makers at the same time.

### 1.2 The NASDAQ Stock Market

The NASDAQ stock market is an American, electronic stock market listing 3,300 companies. The first electronic stock market in the world, NASDAQ currently has the largest trading volume of the American market at around 2 billion shares per day [NASDAQ 07]. While trading in a variety of companies in various sectors, NASDAQ has become known for being an exchange heavy in technology and lists some of the biggest high-tech companies in the United States: Microsoft (MSFT), Apple (AAPL), Google (GOOG), and Intel (INTC), among many others. Being an electronic stock market, NASDAQ executes trades through a computer and telecommunications system. However, it does have a physical presence in the form of the NASDAQ MarketSite in New York's Times Square. MarketSite constantly streams financial information on wall-size displays and contains a TV studio for broadcasting market news.

### 1.3 Market Noise, Velocity, and Forces

While stock markets become increasingly high-tech and many markets start closing their physical floors, there are proponents who argue for the advantages of physical trading [Dwyer 02; GL, 07]. Specifically, they support the high degree of interpersonal interaction as a way of enforcing trading rules and allowing for exchange monitoring which helps maintain the integrity of the market. Further, studies have shown that the sound level on physical trading floors correlates with price changes. In fact, Coval and Shumway [Coval 01] state that such information originating from human interaction cannot be fully replicated by electronic markets. At the same time, markets as a whole have been adding new technology, upgrading their systems and adding more complex data measures in order to capture more information about electronic trades. And while the entirety of information given by human interaction may not be attainable, there is a broad range of important information that can be provided, thus allowing for forecasts to be made.

NASDAQ has tried to compensate for the lack of physically-based information by the creation of two data products called Market Velocity and Market Forces as part of their Market Analytix information service. Market Velocity aims to capture levels of pre-trade activity. It measures the frequency and share volume of orders that have been sent to the trading system, attempting to capture the noise and activity on a physical floor that indicate changes in direction, momentum, or liquidity in a stock. On a physical floor, if several brokers are seen crowding around a single post, frantically trying to place orders, then there is a definite sign of activity in that stock. By looking at how aggressively orders are being made, Market Velocity tries to capture this same increased level of activity. As a complement to Market Velocity, Market Forces utilizes the same information but breaks down orders by whether they are sell orders or buy orders. At a given time, Forces quantifies whether traders are predominantly buying or predominantly selling and the momentum in either direction.

The main data feed that comprises the Velocity and Forces data is an accumulation of buy and sell orders per stock per time of day. The buy and sell order quantity contributes to a running total of Velocity for that stock and time, and so Velocity can be seen as the total volume in orders. Forces are kept as a ratio of buy volume to sell volume for that stock and time. A 21-day running average of previous Velocity values per stock and time of day gives an expected level of activity for comparison. It is important to note that Velocity and Forces measure volume in orders and not trades. Orders are requests for a buy and sell, while trades are completed orders.

### 1.4 Visualizing Velocity and Forces

Market Velocity and Forces are part of the Market Analytix information service and thus generally provided as a data feed; however, there are a few examples of the Velocity and Forces information being used in visualizations. The Experimental Market Information site previously provided graphs showing expected and actual Velocity and separate graphs showing Forces. However, these are no longer being updated with real-time data. The NASDAQ Marketsite currently has visualizations for Market Velocity and Market Forces per company (Figures 1-2).

*Figure 1: A graphic showing Market Velocity for a single company. The velocity meter indicates the level of pre-trade activity compared to the average value.*

*Figure 2: A graphic showing Market Forces for a single company. The dynamic pie chart indicates the ratio between numbers of buy orders versus sell orders.*

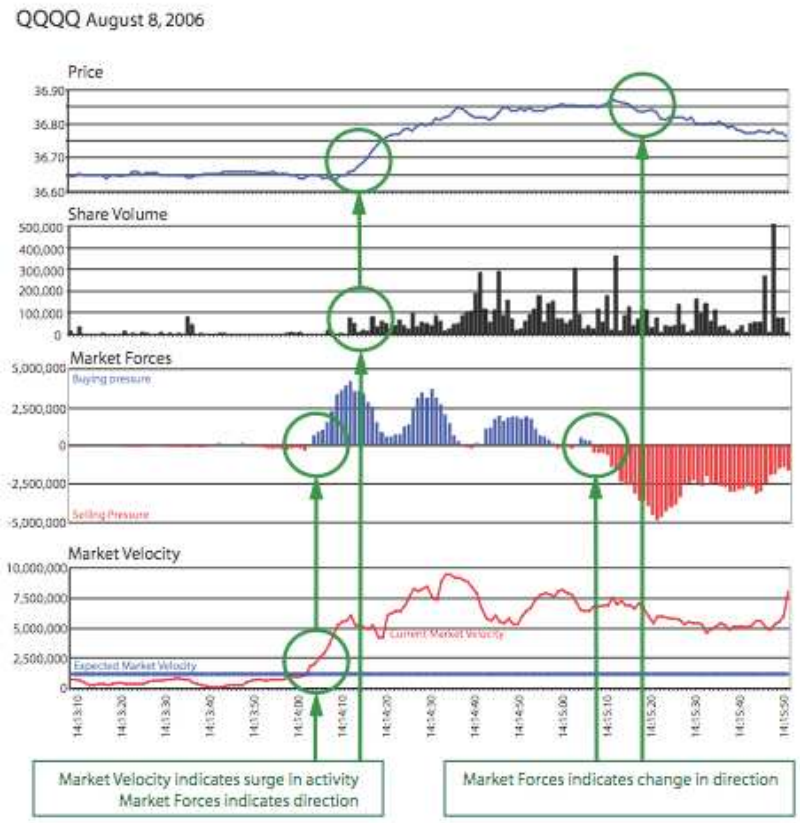


*Figure 3: A chart showing how Forces and Velocity can help to predict changes in price and share volume.*

While Market Velocity and Forces data may predict changes in price and volume (Figure 3), the focus is always per company. The goal of this project is to display Market Velocity and Forces not just for a single company, but for several, to give traders looking at the data an idea of how an index or market as a whole is behaving and to pick out stocks that are starting to increase in activity without having to watch a single stock at a time. To this end we have developed the FireStox visualization. FireStox simultaneously displays the Velocity and Forces for the NASDAQ 100, a stock market index of the largest, non-financial, NASDAQ-listed companies. Details about the development of FireStox can be found in [Section 3], but first we will describe some related work.

## 2 Related Work

Financial data is basically a set of time series. The two standard values measured over time are price and volume. These values are computed for companies, which are themselves organized into industrial sectors. As mentioned in [Roberts 04], financial information is hardly unique and there are many visualization designs that are appropriate to time series. We can classify the visualization techniques applied to financial data into two groups: standard and innovative.

The standard representation is the two-dimensional line plot that every financial portal or newspaper displays. The basic idea is to show time on the X-axis and price or volume on the Y-axis. Price is often represented using a line series and volume using a separated bar chart. Technical analysis of market evolution is entirely based on this type of graph. It is true that these line plots are appropriate for solving problems in low-dimensional time series. But in fact, financial data is highly multi-dimensional. Consequently, standard line plots are not sufficient to solve interesting challenges offered by a large number of dimensions.

That is the reason why in today's financial graphics world, several innovative solutions come out of academic research and in the form of professional software. Most of the innovative works are focused on three-dimensional representations. For example, [Parrish, 00] is extending the standard 2D line plot by adding a third axis to compare the evolution of price and volume between a large number of stocks during the same period of time. The views generated are useful to explore the past evolution of the market, but they do not provide a real-time tool for traders. In [Strausfeld 95] the idea is to have three movable planar intersecting spreadsheets to explore relationships in the data. Besides its attractiveness, this architecture does not offer more to standard line plotting than a third linear dimension. In [Dwyer 02], [HighTower 07], and [GL 07], the idea is to display a navigable three-dimensional map, where columns of different sizes represent stock prices and volumes. The main problem with these solutions is that the 3D view brings occlusion and needs training for users to correctly understand the evolution of the market. [Gravity 07] is trying to enhance the 3D experience by representing a wire-frame sphere onto which a portfolio is displayed with a colored polygon. The goal of this concept is to help users diversify their portfolio. The position of the edges of the polygon represents a diversification metric and so the analyst is directed to make the polygon shape as big as possible within the sphere. Authors illustrate this concept of a Gravity Sphere over the NASDAQ 100 portfolio in [Wyss 06] (Figure 4). The idea is original, but the goal is more focused on risk assessment than market activity representation.

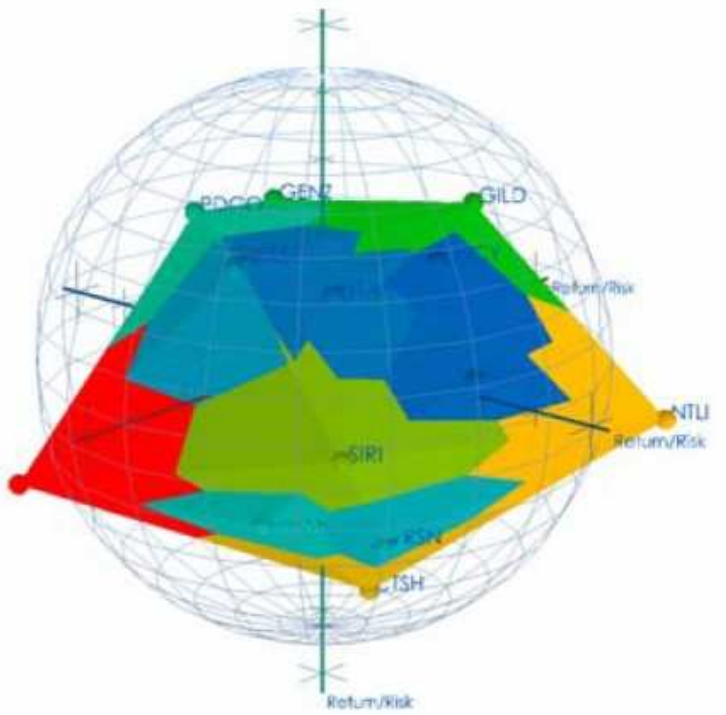


*Figure 4: A portfolio visualized using GSphere. The bigger the polygon within the sphere, the more diversified the portfolio is.*

Finally, to close this paragraph on three-dimensional solutions, it is important to mention the 3D Trading Floor developed by NYSE [Gregory 02]. The 3DTF is a tool to visualize real-time operations based on the actual representation of the physical exchange floor. This architecture was developed to replace in people's mind the traditional market floor representation with a more high-tech view of trading operations. The truth is that nowadays, fewer traders go to the floor and orders are all computerized.

There are also innovative solutions that stick to the two-dimensional approach. In [Wattenberg 99], Martin Wattenberg used the concept of a treemap to display a "map of the market" (Figure 5). It has a strong design that provides users with an immediate overview of the market. The free version of the application lacks sensitive filtering options and it is more an attractive marketing tool than a tactical Web application for traders.

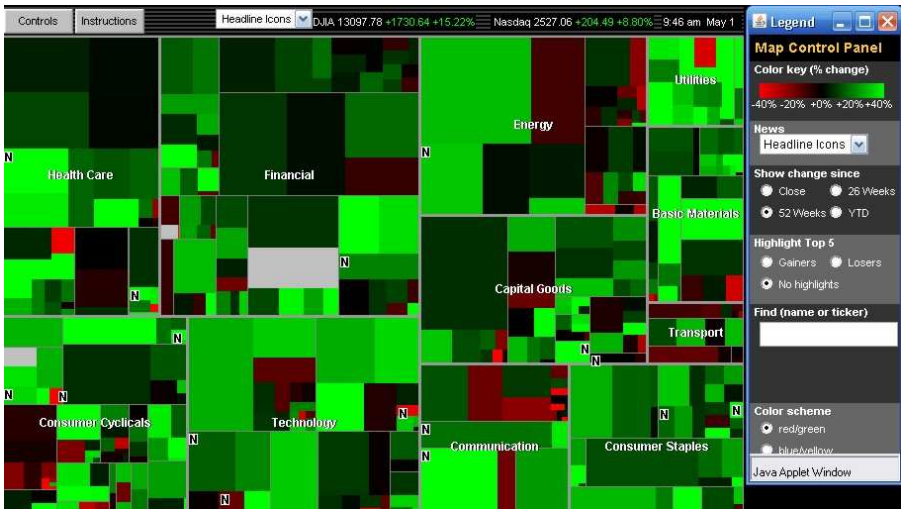


*Figure 5: A screenshot of the Map of the Market showing 6000 stocks. Companies are represented by rectangles and are organized by sector. The size and color of the rectangles indicate the volume and price of the company's shares (http://www.smartmoney.com/marketmap).*

The purpose of [Simunic 03] is to group multiple standard line plots into clusters, in order to assist technical analysis of market evolution. The result is a compact map of thumbnails of line plots grouped by similarities. The distance computed from the similarity metric between each graph is represented by their color brightness. It is possible to zoom from an overview of a large number of stocks to a specific cluster of patterns. Finally, [NeoVision 07] offers a large set of products derived from the concept of a heatmap, where colorful spreadsheets of tickers help users to better spot outliers.

Among the different visualization techniques reviewed in this section, we did not find a solution suitable to display the new highly volatile metrics from NASDAQ, but we did gain insight about the main functionality required in a financial application. In [Merino 06], authors evaluated five visualization techniques over nine related financial tasks. The techniques considered were standard charts, parallel coordinates, heatmaps, recursive patterns and treemaps. The empirical evaluation procedure was to randomly assign one of the nine visualization interfaces to one of the two datasets (historical data and real time data) and a list of tasks to 80 users from various backgrounds. Results were compiled using two metrics: time to completion and ratio of success to failure. The conclusions are that no visualization technique is the best for all tasks. For historical data, charts and recursive patterns seem to be the most efficient whereas for real time data, charts and treemaps are more appropriate.

# 3 Application Design

Given the temporal and highly volatile nature of the Velocity and Forces data, we set out to build an application that would allow users to quickly detect trends and changes in both of these variables for one or more companies simultaneously. We also thought it would be interesting to correlate changes in Velocity and Forces with more traditional indicators such as price and volume. Finally, we envisioned this being a Web application that would be readily available to anyone wishing to monitor the market throughout the course of a trading day.

During our planning and brainstorming stage, there was significant reason to closely examine the *SmartMoney Map of the Market*, which we mentioned previously, as this treemap visualization is one of the most successful in terms of showing multiple variables for many stocks and companies concurrently. The *Map of the Market* treemap is partitioned at the top of its hierarchy by industry sector, which is a natural way to broadly categorize a large number of companies. The two variables it displays, by default, are market capitalization (indicated by the size of a cell) and recent price performance (indicated by cell color, some shade of red or green). The *Map of the Market* application is successful because market capitalization is not a rapidly changing variable, and thus the treemap is relatively stable – companies are not frequently relocated in the overall layout.

We could have chosen to start with a treemap representation of Velocity and Forces data, but we saw this being problematic. Both Velocity and Forces can vary greatly minute-by-minute, and even second-by-second. Even stock price performance data is relatively stable when compared with either Velocity or Forces. Thus, we sought out a more novel visualization that would enable us to present rapidly fluctuating data in a way that still allows users to detect trends and patterns in companies or regions of the market in which they have interest.

## 3.1 Velocity and Forces "Radar"

Some of our early ideas were focused on picturing hundreds or thousands of stocks as points in an "arena" or "universe", much like a network visualization. However, network visualizations are prone to obfuscation and instability, and did not seem well suited for a multivariate, rapidly changing data set. To make the task of tracking such quickly changing data easier for us and for users, we settled on using a two dimensional radial graph, an extension of a pie chart, to represent a relatively small number of stocks. Each wedge in the graph represents a company. As Velocity and Forces change, size and color of the wedges in the graph change correspondingly. Because wedges do not rapidly shift or randomly relocate, it should be easy to find and track particular companies of interest in the core visualization, which we call the "Velocity and Forces radar". We are limited in the number of issues, stocks, or companies we can display simultaneously, so we have chosen the NASDAQ-100 Index as our working data set. This index is composed of 100 of the largest domestic and international non-financial securities listed on the NASDAQ Stock Market as based on market capitalization.

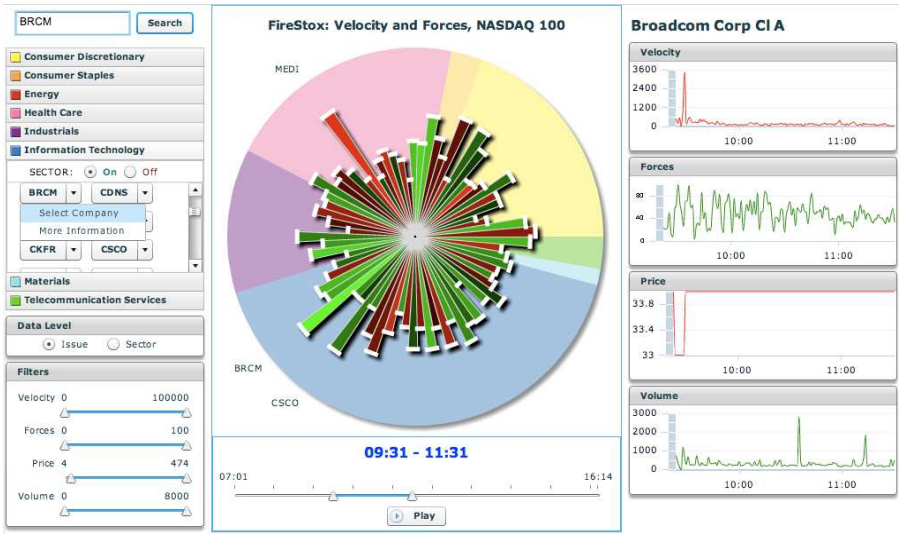


*Figure 6: A screenshot of FireStox. The company browser is on the left, the radar in the center represents Velocity and Forces, and the line graphs on the right show the evolution of values of Velocity, Forces, price and volume.*

The Velocity and Forces radar shows at most 100 companies simultaneously. Each company is represented by a wedge whose length is proportional to the current Velocity of that company. In addition, wedges are colored a shade of red or green depending on current market Forces for that company – a bright red indicates strong selling pressure, whereas a bright green indicates strong buying pressure. The background color in which a company is positioned indicates the sector that company belongs to. Within each sector, companies are ordered alphabetically. Stocks that exhibit a high degree of Velocity are labeled along the outer rim of the graph (Figure 7).

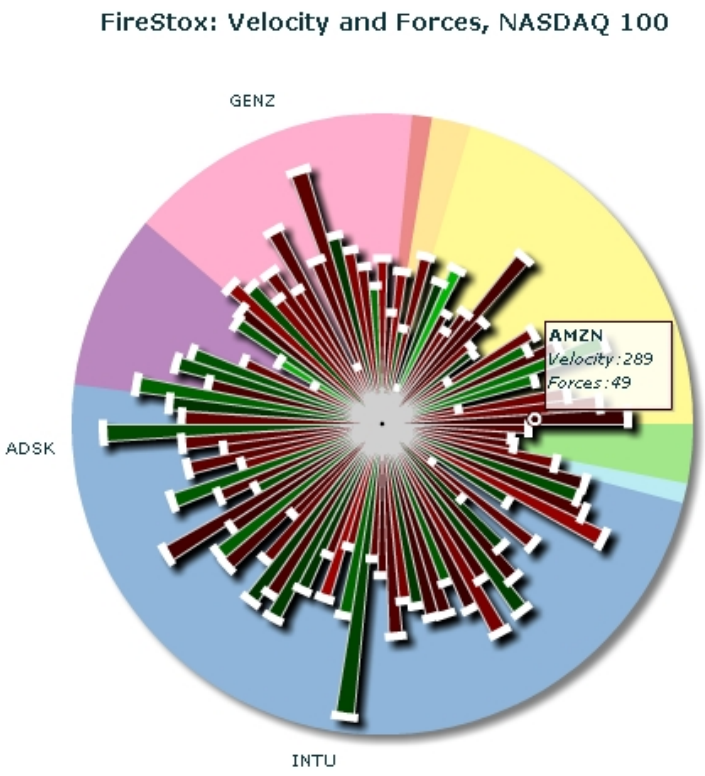


*Figure 7: A screenshot of the Velocity and Forces radar. The size and color of the wedges indicate values of Velocity and Forces, respectively. Companies are organized in sectors, which are represented with different colors in the background.*

### 3.2 Additional Features

Aside from the central visualization, there are a number of other important application features. First, we knew it would be advantageous to organize companies by industrial sector. The NASDAQ-100 companies fall into 8 distinct sectors of varying sizes: for example, Information Technology comprises 48 stocks, while Energy and Materials comprise only one stock each. The index is updated quarterly, however, at which time this distribution of stocks in the NASDAQ-100 is likely to change, so the program reads company and sector information from a database in order to dynamically construct application widgets. We provide the ability to search for a specific stock by ticker or company name, as well as an accordion component to browse companies by sector (Figure 8). We also provide aggregated statistics for each sector in a special mode of FireStox, which we call "sector overview" (Figure 9).

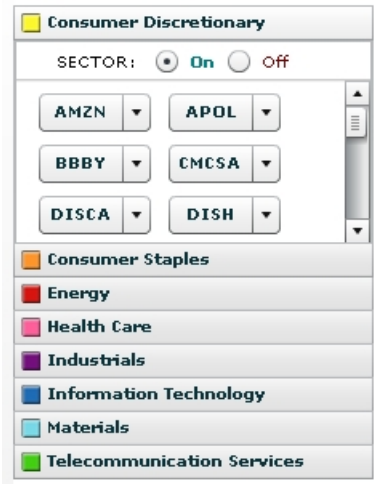


*Figure 8: A screenshot of the sector accordion. By clicking on the title of a sector, users can display the related list of companies.*

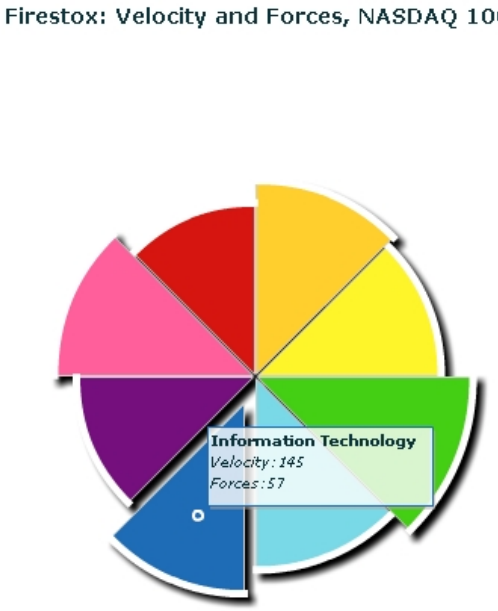


*Figure 9: Screenshot of the aggregated sector data view.*

Taking into account the temporal nature of the data, we felt it was important to allow users to explore the behavior of Velocity and Forces throughout the course of a trading day. To facilitate this process, we offer "replay" as a second major feature. The replay slider allows users to specify a window of time as small as a few minutes or as large as the entire day over which to view changes in Velocity, Forces, price, and volume (Figure 10). The left thumb on the slider bar corresponds to the currently displayed time slice. The right thumb on the slider bar allows users to select an end time. When users click the play button, the visualization will play through the time slices between the times indicated by the left and right thumb in order to show the Velocity and Forces values for the NASDAQ 100 as they change throughout this period. Users can specify a time to the nearest minute. The play button becomes a pause button when the visualization is running, allowing users to stop playback. With such functionality, one can imagine data streaming into the application in real-time throughout the course of a trading day.

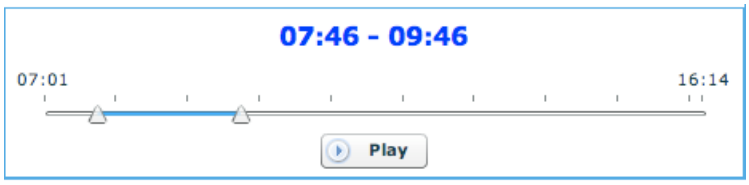


*Figure 10: Screenshot of the replay slider.*

A third feature is a series of charts that line up along the right-hand side of FireStox. From top to bottom, they show change in Velocity and Forces, price and volume for the currently selected window of time (Figure 11). A fourth feature is a set of filters that enable users to select a range for any of the four variables (Figure 12), thus causing only companies that meet the specified criteria to be shown in the Velocity and Forces radar. Users also have the ability to turn on and off entire sectors in the same manner.

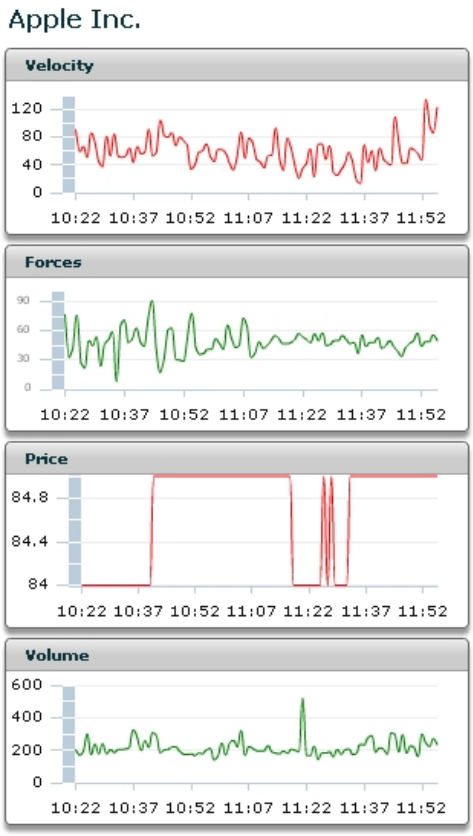


*Figure 11: Screenshot of a portion of the graph panel that shows Velocity/ Forces, price, and volume for Apple Inc. from 10:22 – 11:52.*

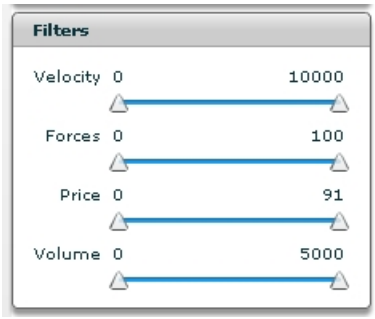


*Figure 12: Screenshot of the panel that allows for filtering of Velocity and Forces data displayed on the main graph.*

# 4 Implementation

We had a number of challenges working with the financial data for this project. There were three main data sources:

1. A spreadsheet containing NASDAQ stock ticker symbols, company names, and sector categorizations, among other things, and a complementary text file listing the companies in the Top-100 index.
2. A data feed comprising buy and sell orders for all NASDAQ issues for one day, February 7th, 2007. The fields of interest in this data feed include company issue, time the order was placed (resolved to the millisecond), transaction volume information, and NASDAQ's calculation of Velocity and Forces for each entry.
3. A separate data feed comprising buy and sell orders for the same day, February 7th, 2007. This feed contains price and volume information in a more human digestible format, and, like the other data feed, contains millions of entries.

Naturally, given this data and the requirements of FireStox, we had to decide which data fields we wanted to use, and on what time scale – it was difficult to envision users looking at data millisecond-by-millisecond. Therefore, we decided that for the 100 companies in question, we would derive four values at each minute of the day: Velocity, Forces, price, and volume. This required aggregating the data into 60-second slices, and prompted discussion about how best to go about this, as different formulas would change the scales of our various metrics. In most cases, our aggregation method took some form of averaging. As we have it, Velocity is expressed on an integer scale of $0 \rightarrow \infty$, where 100 means "100 percent of expected activity". Forces is expressed on a scale of $0 \rightarrow 100$, where values of $0 \rightarrow 49$ indicate degrees of selling pressure, and values of $51 \rightarrow 100$ indicate degrees of buying pressure. We also aggregated our four metrics on a per-sector basis at each time slice to drive the "sector overview" feature. We pre-calculated as much information as we could so this computational burden was not placed on the application.

## 4.1 Architecture

A MySQL 4.1.20 database is where all of our data is stored. An Apache 2.0 HTTP Server hosts the application, and the Web server is capable of executing PHP scripts. FireStox front-end is written with Adobe Flex, a software development kit for Rich Internet Applications based on the Macromedia Flash platform. Our Integrated Development Environment is Adobe Flex Builder 2. Our front-end code compiles into a Flash object that can be executed by any browser with a Flash plug-in. This code makes HTTP requests to the Web server to retrieve data during program initialization and periodically during execution using one of two methods: either going through a PHP script that returns data from MySQL in XML format, or by reading pre-generated XML files directly from the server. The latter is presumed to be faster.

### 4.2 Challenges

We ran into several challenges in developing FireStox. Many of these challenges had to do with the sheer volume of data: the number of entries in the Velocity and Forces data feeds was in the millions, and so we had to do a significant amount of data processing to aggregate it appropriately. Not only that, but we repeated this step multiple times as our understanding of Velocity, Forces, price and volume changed: we were continually asking our colleagues at NASDAQ for clarification of the sometimes cryptic data in the various feeds we were presented with. Another challenge was the behavior of the data itself. Even after aggregating it into one minute time slices, there was still a large amount of variation (and hence visual movement) from one time slice to another. Thus, we were forced to find ways to make the transitions less dramatic and make the behavior of the radar more easily interpretable by users.

Our remaining challenges were mostly in working with Adobe Flex. Flex is an outstanding framework for quickly prototyping and implementing user interfaces. Flex provides a plethora of controls, components, and containers with which to build user interfaces. In addition to familiar controls that one is accustomed to encountering on Web pages, operating systems, and other applications (like text boxes, buttons, checkboxes, and dropdown menus), other more novel widgets are included (like accordions, sliders, and pop-up menu buttons). In addition to these, Flex Charting makes available several types of charts and graphs. FireStox extends the pie chart and uses the line graph, to name two examples. The Flex Builder IDE is intuitive, feature-rich, and includes a design mode that allows users to lay out components graphically and preview changes to their application without invoking the full build process. While for the most part we were incredibly happy with Flex, it was not particularly easy to customize components or implement things exactly how one might have imagined them – there is an inherited look and feel users have to live with. Learning Flex basics is not difficult for experienced programmers, but there is a large number of properties, methods, styles, and events to learn for each component. In this regard, examples go a long way. Another challenge was in understanding the Flex data-binding framework, a unique event-driven method of sharing data between components in the application.

## 5 Evaluation

To evaluate our interface we defined a user test procedure and worked with to two experts from NASDAQ. The procedure consisted of three phases:

1. Learning phase: to explain the goal of the interface and its different components.
2. Practice phase: to allow users to test the interface.
3. Evaluation phase: to get feedback from users with a questionnaire.

Our two expert users work in the NASDAQ Experimental Market Information department. They are interested in innovative and compelling visualization ideas in order to display new data feeds created at NASDAQ. We have met with them twice since the beginning of the project to learn about financial data and to review examples of current visualization tools related to market analysis. We presented our users a

series of 15 positively phrased statements, and asked them to either agree or disagree with them. Each statement was related to a specific function of our interface. We naturally grouped them by component: company selector, Velocity and Forces radar, line graphs and overall.

Figure 13 displays the average expert response to each statement. The statements that received the lowest scores were:

1. "It is easy to find outliers based on Velocity."
2. "It is easy to filter out companies."
3. "This interface is useful to understand the evolution of the market over one day."

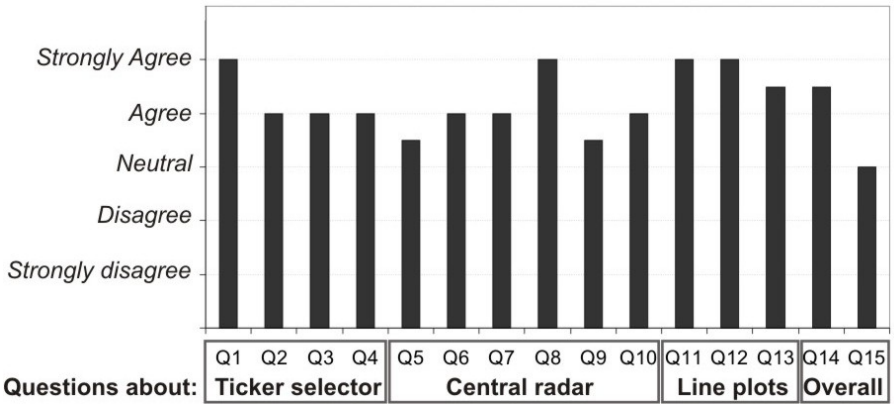


*Figure 13: A chart summarizing the evaluation results from two users. Fifteen questions organized in four categories were graded from 1 (Strongly disagree) to 5 (Strongly agree) by users.*

From these three statements and the comments that users wrote at the end of the evaluation, we identified three different problems in our interface:

1. The size of the wedges in the radar does not usefully represent our velocity scale.
2. The filtering sliders are difficult to use.
3. The interface needs to provide more information to assist users in understanding the evolution of the market over one day.

Consequently, we improved three aspects directly related to these problems. For problem 1, we plan to add a circular marker on the radar to display the 100% Velocity boundary, in order to help users to scale companies according to their Velocity values. For problem 2, we plan to add a status bar to inform users of the filter currently applied. Finally for problem 3, we thought that to help users to understand the evolution of the market over the day, we could link displayed companies with a data feed of related financial news. Nevertheless, we did not have enough time to implement this solution, so we keep it as future work.

## 6 Future Work

The user evaluation provided us interesting directions to follow for future enhancement of this project. These improvements fall into three categories: overall functionality, central radar and line graphs.

Firstly, the overall application would be improved if we could work on more than a single day of market data. Our idea would be to add a calendar widget to offer users the possibility to select the day represented, or even to extend the time window analyzed over several days. Another important feature that is currently missing is the portfolio. We think that it would be important for users to be able to save a portfolio of tickers and to display only stocks they are interested in. Finally we would like to add an "event detection" feature, to highlight important changes in the data, in order to help users easily spot interesting periods of the day.

Secondly, we believe that the central radar could be improved by adding arrows on top of each wedge, to indicate if the Velocity value is increasing, decreasing or idling. We would also like to allow users to tune the radar by configuring the scales and the color palette used to display the Velocity and Forces values.

Thirdly, the line graphs would be enhanced if users could better correlate events across the four time series. We already aligned them on top of each other for this purpose, but a vertical marker that lines up the same time of day across the four charts would help users to better compare the evolution of Velocity, Forces, price and volume. Finally, we think that to be able to select and display more than one company would be an interesting feature to understand the relationship between multiple stocks.

A higher degree of interactivity is also one of the future goals for FireStox. The ability of a user to find and interpret the right information quickly is crucial to the use of FireStox in trading. By allowing users to have more control over what data is presented and how, FireStox can give its users a more visually efficient tool that provides exactly the amount of information that a particular trader would desire. Also by letting users customize the data viewed as well as the visualization, FireStox would become an effective tool for communication between traders as well as lay persons, by paring down undesired data and being able to visually demonstrate trends and events.

These enhancements are being explored and assessed for continued development at NASDAQ. Crucial to the growth of FireStox is being able to create an efficient data-providing infrastructure that will allow users to smoothly explore the huge volumes of data that exist for the millions of trades and thousands of issues. Therefore, the development of this data provider is being examined in conjunction with the above enhancements.

The development of FireStox is not limited to use on a single stock market. The concepts of Velocity and Forces translate to other electronic markets, since they are based on the observation of buy and sell orders which are common to all electronic markets. Furthermore, the idea of watching the volume of buy versus sell orders can be extended to other types of securities, such as warrants for example. However, the analysis of trends for these more long-term securities would be done overtime. A visualization such as FireStox would be best used as a historical tool, rather than in real-time as is the current goal.

## 7 Conclusion

In this paper we introduced an innovative Web application named FireStox, to visualize and explore financial data from the NASDAQ Stock Market. FireStox focuses on representing the evolution of Market Velocity and Market Forces, two new products from NASDAQ that capture trading activity. The temporal and highly volatile nature of this multi-dimensional data made for an interesting visualization challenge. The core solution provided by FireStox is a two-dimensional Velocity and Forces radar. This radar was designed to allow users to quickly detect trends and changes in market evolution for up to 100 companies simultaneously. FireStox also provides features to correlate changes in Velocity and Forces with more traditional indicators such as price and volume. Two experts from NASDAQ evaluated the interface and made suggestions for improvement with respect to the professional trader's expectations.

**Acknowledgements**

We would like to thank Claude Courbois and Jeff Kimsey from the Data Product Department at NASDAQ, for providing us with the problem, the dataset, and feedback on our work. We would also like to thank our classmates for their suggestions on the project.

## References

[Coval 01] Coval, J., Shumway, T.: "Is sound just noise?"; http://www-personal.umich.edu/~shumway/papers.dir/out1.pdf (retrieved 2007/04/28).

[Dwyer 02] Dwyer, T., Eades, P.: "Visualizing a Fund Manager Flow Graph with Columns and Worms"; Proceedings of the 6th International Conference on Information Visualization, IV (2002), 147-158.

[GL, 07] GL Trade SA; http://www.gltrade.com.

[Gravity 07] Gravity Investments, Inc; http://www.gravityinvestements.com.

[Gregory 02] Petroff, G.: "The Virtual Stock Exchange: Lessons learned from the NYSE 3DTF"; Proceedings from CODATA Workshop, July 10-11 (2002).

[HighTower 07] High Tower Software, Inc.; http://www.portfolioimpact.com.

[Merino 06] Merino, C. S., Sips, M., Keim, D. A, Panse, C., Spence, R. "Task-At-Hand Interface for Change Detection in Stock Market Data"; Proceedings of the Working Conference on Advanced Visual Interfaces, ACM Press New York, NY, USA, (2006) 420—427.

[NASDAQ 07] "Get the Facts"; NASDAQ, http://www.nasdaq.com/reference/market_facts.stm (retrieved 2007/04/28).

[NeoVision 07] NeoVision HyperSystems, Inc.; http://www.neovision.com.

[Parrish 00] Parrish, E.: "StockVis: An Internet-Based system for Visualizing Stock Market Data"; Masters Thesis, US Santa Cruz, Department of Computer Science, (2000).

[Roberts 04] Roberts, P. "Information Visualization for Stock Market Ticks: Toward a New Trading Interface." Masters Thesis, MIT, (2004).

[Simunic 03] Simunic, K.: "Visualization of Stock Market Charts"; Proceedings form The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, (2003).

[Strausfeld 95] Strausfeld, L.: "Embodying Virtual Space to Enhance the Understanding of Information"; Master Theses, Massachusetts Institute of Technology, Program in Media Art and Sciences, (1995), 17.

[Wattenberg 99] Wattenberg, M.: "Visualizing the Stock Market"; Conference on Human Factors in Computing Systems, ACM Press New York, NY, USA, (1999), 188-189.

[Wyss 06] Wyss, J., Kerins, M. "Gravity Investments' Optimized NASDAQ 100 Performance Testing"; (2006).