ORDER CENTRIC TRACKING SYSTEM AND PROTOCOL FOR COMMUNICATIONS WITH HANDHELD TRADING UNITS

FIELD OF THE INVENTION:

The present invention is directed to a method and system for allocating, tracking and reporting orders and trades executed in the context of an exchange setting, such as the New York Stock Exchange (NYSE), which method and system supports the use of wireless handheld trading devices used by floor brokers on the exchange floor. The present invention is further directed to a specialized communications language and protocol for communications between the handheld devices and the trading system.

STATEMENT OF RELATED APPLICATIONS:

This application claims priority from pending U.S. Provisional Application Serial No. 60/157,987 entitled "EXTENSIBLE MARKUP LANGUAGE (XML) BASED PROTOCOL", filed October 6, 1999, pending U.S. Patent Application Serial No. 09/413,270, entitled "ORDER CENTRIC TRACKING SYSTEM", filed on October 6, 1999, and pending U.S. Patent Application Serial No. 09/413,150, entitled "HANDHELD TRADING SYSTEM INTERFACE" and also filed on October 6, 1999. The entire contents of these three applications are herein expressly incorporated by reference.

BACKGROUND:

There exist several types of financial markets in which securities, commodities, and other negotiable instruments are traded. An auction market, such as a stock exchange, is one such financial market. In an auction market, buyers and sellers congregate on an exchange floor and announce their respective bid prices (offer to buy) and ask prices (price acceptable to sell). A trade in any particular security will occur at no more than the highest price a buyer is willing to pay and at no less than the lowest price a seller is willing to accept.

Among the players on the floor of an exchange are specialist and floor brokers. A specialist calls out the best bid and ask prices received from the various brokers, ensures that trades are posted, facilitate trades, and acts to ensure liquidity. A floor broker roams the exchange floor and acts as an agent to transact orders on behalf of investors (buyers and sellers).

A typical transaction originates when an order is placed with an off-the-floor trading desk to buy or sell a particular security. The trading desk then conveys the order to an exchange clerk who notes the parameters of the order, including whether the order is a buy or sell order, the symbol of the security, the quantity, the price, any special conditions associated with the order, and the time that the order is placed. Clerks operate from various booths maintained on or near the trading floor by the trading and brokerage establishments.

After appropriate processing, the clerk delivers the order to a floor broker for execution. Traditionally, orders are transcribed onto order slips that are physically delivered to floor brokers by pages or runners. A floor broker executes an order, notes the executed order on a slip of paper, and subsequently returns the notated slip of paper to the clerk via a runner. In addition to buy and sell orders, investors may request a "look" from the floor of the exchange. In response to a "look" request, a broker notes his or her observations with

respect to what is happening in the market for a particular security. The "look" information noted by the broker may vary depending on the particular broker and what he has observed. For example, "look" information may include recent buyer and seller identities, trade sizes and prices, appraisal of market interest, a broker's opinion and any other information that a broker may wish to provide.

There is currently a significant manual component to process an order once the order reaches the floor of an exchange. Typically, an order will be entered into a computerized order processing system of a trading establishment. For example, these orders can be entered by a trader 120 at a listed desk. The order is then routed to an order management system for exchange listed securities. The order is displayed via an order management system application in the trading booth that handles orders for the given security. An order ticket is then automatically printed in the appropriate booth. After the order prints, a clerk takes the ticket and prepares it for handoff, pages a broker, and acknowledges the order in the order management system. The broker, upon being paged, must return to the booth to get the machine-printed ticket and may briefly discusses any special handling instructions with the clerk. Alternately the broker may telephone the booth to get necessary information and write it on a piece of paper. A broker must also update running totals representing how many shares of a particular security to buy or sell, incorporating both pending and new orders. After the broker executes a trade for all or part of the various orders, the broker must convey some or all of the details of the trade to the booth. The broker can convey the information over the phone or write the information on a piece of paper and walk it back to the booth. Alternatively, the broker can send the paper to the booth via an exchange runner. Thus, as will be appreciated, a floor broker must often interrupt their trading by returning to the booth or telephoning the clerk. This introduces inefficiencies into the trading system and can also result in situations where a broker may be

absent during a trading situation which would be particularly advantageous to the broker's clients.

A clerk typically records the verbal execution in an online management system and performs an allocation of a portion of the bought or sold shares of a security among a variety of orders. Contra breakdowns tracking what was traded with whom eventually arrive in the booth on a piece of paper if they were not attached to the verbal communication. This information is generally penned by the broker or by a specialist. The clerk files the contra breakdowns in a special location, to be picked up by a firm runner. The contra breakdowns are then taken to a bank of firm typists located near the exchange floor. The typists enter the information into a firm trading system. Generally, the contra information should be entered within an hour after a trade took place. The typists file the paper containing the verbal and written information. This paper is kept on hand for several days and is then archived. The entered information is both used by the trading firm's systems and the exchange's order reconciliation system (OCS).

It would be useful to have a system capable of achieving greater order processing efficiency. Orders need to be routed more quickly to brokers operating on the floor of the exchange, thereby leading to more timely customer service. In addition it would be useful to capture some of the order information digitally at the point of sale, whereby costly transcription errors can be reduced.

It would be of further use if the order and look information could be directly communicated to the floor brokers by means of a wireless handheld communication device, thus eliminating the need for brokers and/or runners to continually interrupt trading to visit or call booth clerks. In such a wireless system, it would be particularly advantageous if a customized communication protocol were used to communication with the handheld device, which protocol supported all required data communication types in a format which both

simplified parsing and processing of transmitted messages and also reduced required messaging bandwidth.

SUMMARY OF THE INVENTION:

The above and various other deficiencies in the conventional manual trading system, are addressed by the present order centric method and system for tracking orders implemented on a trading floor exchange. The system is configured to automatically route orders to an appropriate booth and a floor broker, via a wireless handheld device, according to a symbol associated with the particular security being traded.

In one aspect of the invention, a set of symbols are allocated for particular securities and various members of that set are assigned to a specified booths and floor broker IDs. An order is entered into a computer and associated with a specified security symbol. The order is transmitted to a computer server. The order is routed through the server to a computerized booth station associated with the booth to which the order symbol had been allocated. In addition, the order is routed through the server to a handheld-device logged into the system and registered with the floor broker ID to which the symbol associated with the order has been allocated.

In addition, a record of an action relating to the order can be sent to the server and logged into a memory at the server. The record can also be routed through the server to the booth station associated with the booth to which the order symbol had been allocated. Typically, multiple booths are utilized with a unique set of symbols allocated to each booth station. The set of symbols allocated to a floor broker ID is a unique subset of the set of symbols associated with a booth.

According to another aspect of the invention, a heartbeat signal from the handheld device to the server within a predetermined time period. In one embodiment, any

communication between the handheld device and the server suffices as a heartbeat. A floor broker can be automatically logged off of the server in the event the server does not receive a predetermined number of heartbeats. For example, the predetermined number of missed heartbeats can be two.

In another aspect, an order is for shares of a security stock described by a symbol and the system calculates an aggregate number of shares of stock for standing orders relating to a particular symbol. Additionally, the system can calculate an aggregate of pending orders that meet a threshold price. The orders can include buy orders or sell orders.

This invention can also include a computerized system for processing an order in a trading exchange. The system can include a computerized booth station and a handheld computing device linked by a computer server. Software operative with the computer server can route an order to a particular booth station according to a security symbol associated with the order. In addition it can route the order to a particular handheld computing device according to the security symbol associated with the order.

In still another aspect, the computerized system can be linked to a computerized order management system and a computerized recording station.

Various communication protocols can be used to communicate with the handheld devices. Preferably, messages are encapsulated in accordance with a specialized extensible markup language based protocol which has been designed to allow easy formatting processing of the various message and data types which are passed to and from the floor brokers. The syntax for the messaging protocol is implemented as an XML document type definition which is can be maintained on one or both of the handheld devices and the server. The protocol includes op-codes to support both look and order processing functionality. In addition, specialized functionality not easily achieved using conventional techniques is also supported. Such functionality includes easily communicating to a broker the last sale price of

a set of securities at a given time, such as the set of securities for which the broker has outstanding orders, and immediately recalling an order previously sent to a broker via a handheld device if it is discovered that the order has become corrupted or otherwise should not be completed.

5

BRIEF DESCRIPTION OF THE FIGURES:

The foregoing and other features of the present invention will be more readily apparent from the following detailed description and drawings of illustrative embodiments of the invention in which:

10 FIG. 1 is an illustration of the high-level architecture of one embodiment of an order centric tracking system;

FIG. 2 is a flow diagram illustrating order processing steps;

FIG. 3 is a flow diagram of a heartbeat synchronization process;

FIG. 4 illustrates a flow of an allocation of traded shares;

15 FIG. 5 is an alternative representation of the architecture of an order centric tracking system according to the invention;

FIG. 6 is a diagram illustrating messaging flow between a handheld device, a server, and other elements of a trading network for sample look and order requests; and

FIGS. 7-20 are example trading-related messages formatted according to a

20 preferred messaging protocol.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS:

Turning to Fig. 1, there is shown illustration of the high-level architecture of one embodiment of a networked computer system 100 for tracking an order executed on an

25 exchange floor. A trader 120 can initiate an order to be executed on the floor of an exchange.

The order is entered into an online management system 130. The online management system 130 can transmit the order to a Handheld Server (HHS) 113 and to one or more computerized booth stations 161-163. The HHS 113 can transmit the order to an appropriate handheld computing device (HHD) 114-116.

The order centric trading system 100 includes a network connecting the computerized Handheld Server 113 and handheld computing devices 114-116. The system can also include computerized booth stations 161-163, computerized trader stations 166, computerized recording stations 150, computerized customer stations 140 and a computerized online management system 130. Each of the computerized devices 114-116 130 140 150 161-166 can include a processor, memory, a user input device, such as a keyboard and/or mouse, and a user output device, such as a video display and/or printer.

The computerized devices 114-116 130 140 150 161-166 can communicate with each other to exchange data. Interactions with the Handheld server 150 and the online management system 130 can proceed as if each was a single entity in the network 100. However, the HHS 113 and the online management system 130 may include multiple processing and database sub-systems, such as cooperative or redundant processing and/or database servers 164-165, that can be geographically dispersed throughout the network 100. A local server 164-165 may be a proxy server or a caching server. The HHS 113 may also include one or more databases 145 storing order related information. Appropriate hardware and software which can be used with suitable modifications as appropriate to implement various aspects of this network as described herein will be known to those of skill in the art.

Referring now to Fig. 2, a trader, customer or other person with access to the Order Management System 130 initiates a trade by entering an order 210 into a network access device such as, for example, a computer. The Order Management System 130 processes the order by logging the order as appropriate and associating it with a broker ID

and a booth according to the symbol of the security involved in the order. The Order Management System 130 then transmits the order to a booth station 212 and also acts to transmit the order to the Handheld server 213, for example, after the order has been processed by the booth clerk. The handheld server in turn transmits the order to a handheld computing device onto which a Broker ID associated with the security symbol is logged.

Each order that arrives at the handheld 114-116 can be accepted or rejected by the broker. If an order is rejected, it can appear in a "ghosted" state until explicitly dismissed by the broker. Brokers can enter executions according to orders received into the handheld computing device 114-116. The information relating to the orders is transmitted to an online management system for exchange-listed securities. The broker can interface with the handheld devices by, e.g., clicking "buttons" and other user interface devices displayed on the screen of the handheld computing device 114-116, to record the symbol, side, price, and quantity of an execution. Contra breakdowns and other relevant information, such as an "as of" time can also be captured on a handheld computing device 114-116. Such information can then be forwarded automatically, or on request, from the handheld device to the trading system. Trading firm personnel, such as booth clerks, can perform allocations of the executions using an online management system. The handheld computing device 114-116 can receive updated leaves based on the clerk's allocation. A typist at a recording station 150 can enter written information using the broker's digital records. A preferred implementation of user interface for a such a handheld trading device is disclosed in co-pending U.S. Patent Application Serial No. 09/413,150, entitled "HANDHELD TRADING SYSTEM INTERFACE", filed on October 6, 2000, the entire contents of which is expressly incorporated by reference.

In a particular embodiment, the handheld computing device 114-116 used on an exchange floor is capable of TCP/IP communication over a wireless network 119. The

wireless network is typically supported by the trading exchange. However, the handheld computing devices 114-116 can also establish a direct TCP/IP socket connection to a handheld server 113 and not be required to use exchange middleware wireless networks 119. Various communication schemes can be used to pass data between the trading system and the handheld devices. A particularly preferred interface protocol implemented as an XML layer over the TCP/IP socket is discussed in detail below.

A broker can execute trades in accordance with outstanding orders that have been transmitted to the handheld computing device 114-116. The order centric system keeps a broker aware of how many shares to buy and sell of a particular security and at what price levels are acceptable. The handheld device 114-116 can be used to assist a broker in this task by maintaining a list of outstanding orders and aggregating the leaves of like orders. The handheld device is configured to permit the broker to record executions on the device 114-116. In one embodiment, order processing functionality can include execution information captured semantically, such as the symbol, side, quantity and price relating to the trade. Information including contra information, time of day, special instructions, and almost any other information relating to an order can also be recorded via the handheld computing device 114-116.

A history log can be provided within the order centric system 100 in which significant events that occur relating to an order can be recorded. The history log can be stored in an electronic storage medium such as a magnetic disc drive or a compact disc (CD). The log can provide a means whereby a broker can review information during the trading day. Tasks can be presented to a user in a manner that will give the user a quick view of what actions have been performed relating to an order or a group of orders. Tasks tracked by the order centric system can be displayed in chronological order, or according to filtering and sorting functionality.

Users can include a trader 120, a booth clerk, a broker, a typist or others with access to the order centric system. In one embodiment, a user can be a customer 140 with remote access to the order centric system. Customers 140 may be given access rights to view orders they have placed. In addition, if desired, customers can be given the ability to track trades placed by others whereby the customer can get a "feel" for the trading environment at any particular time without specifically requesting a floor look.

In one embodiment, task history data is also stored on the handheld computing device 114-116. Data is preferably purged from the handheld computing device 114-116 at the beginning of each new trading day or more frequently as required based on device memory constraints. Purging can also be subject to network failure recovery as discussed in more detail below.

In addition to the general history, a separate database can be maintained on the HHS 113 to store executions that have been entered during the course of the day. The separate database will allow brokers to reconcile executions with the booth in failure recovery situations.

To increase security, the order centric system can encrypt the message stream between a andheld 114-116 and the HHS 113.

The Handheld Server 113 is configured to communication between existing trading firm systems 130, trading exchange systems and the handheld computing devices 114-116. Each handheld 114-116 can establish a communication session with HHS 113 over a wireless network, and HHS 113 will participate in order processing systems on behalf of the handheld computing devices 114-116. HHS 113 can also maintain login session state for the handheld computing devices 114-116. HHS 113 can act as a pass-through, performing protocol conversion between a trading firm's Order Management Architecture and the handheld messaging protocol, one form of which is discussed in detail below.

An order centric system can allow an order to be entered into a computerized order management system. Typically, an order is entered by a trader on the Listed desk of a firm. The order is routed to the order management system for exchange-listed securities. The order can also be displayed in the order management system application in the booth that handles orders for the given security. In one embodiment, the order centric system automatically routes the order to a broker who handles for that particular security. In another embodiment, a clerk or trader initiates routing of the order to the appropriate broker.

In a particular implementation, the order centric system pages the floor broker. Advantageously, no paper ticket needs to be generated. The broker, upon being paged, can retrieve the new order on his handheld 114-116. If the broker accepts the order, the order is added to the list of active orders. The online management system display updates and shows that the designated broker has accepted the order.

The broker can execute a trade for all or part of the order on the exchange floor. The broker can record the symbol, side, price, and quantity by clicking or otherwise operating programmable user interface devices on the screen of the handheld. The broker can also record the contra breakdowns with a freehand image or "digital ink". The image recorded in digital ink can be processed for character recognition or sent as a simple image. When the broker is satisfied with the content of the recording, a "send" option can be selected to transfer the recording to the server. According to one embodiment of the invention, the handheld unit 114-116 is programmed to generate an estimated allocation of the shares of the security traded and update the order leaves to reflect allocation. The execution is transmitted into the online management system.

A clerk, typically located in the booth, can access a display of an execution that has been transmitted to the online management system. The clerk can perform an allocation with the traded securities if appropriate. The image of the contra breakdowns will

also be available to the clerk. Allocations performed by the clerk are in turn transmitted to the HHS 113 and logged. The allocations are also transmitted from the HHS 113 to the floor broker via the handheld 114-116 computing device 114-116. The handheld device 114-116 receives the updated leaves according to the clerk's allocation and the effects of the allocation are displayed. In addition, the allocated execution is also transmitted to the typists with any inked breakdowns whereby they can record the "writtens." The file containing the image of the breakdowns can also be archived. Archives can be accomplished, for example, in an electronic storage medium, such as a disc drive or CD. In one embodiment, the trading firm can utilize an application that permits bulk display and/or printing of the inked breakdown images.

In one embodiment, the order centric trading system can also include a failover procedure. In the event of a primary network failure, the handheld device 114-116 can attempt to connect to a backup server 131. In addition, the order centric trading system 100 can include, in various embodiments, additional features such as the ability to digitally accept all execution information at the point of sale, enhanced messaging between brokers, traders, and clerks, electronic deposit of orders with the specialist, and receipt of analytics, such as market data, statistics, trends or other information useful to accomplishing an educated trade, on a broker's handheld device. In addition the order centric system can operated over an intermediary network system, such as a network system installed in an exchange for communication to and from the floor of an exchange.

A login session in the order centric system can include any actions entered by a broker after they have logged in to a computerized handheld device 114-116. An execution history database can be utilized to record order requests, executions, and other detailed information about a login session. In one embodiment, a history can be cleared whenever a new login session is initiated or more frequently as needed based on available RAM.

An execution history database can be used in recovery situations (such as when a session was abnormally terminated) to reconcile executions entered on the handheld 114-116 with those received by the online management system. Once entered on the handheld 114-116, an Execution can be stored in this execution history database. For example, information about an execution that can be stored in the database can include: HHD Execution ID – an Execution identifier created by the HHD during the creation of execution, Online Mgmt. Execution ID - identifier assigned by the online management system, Security Ticker Symbol Side, Quantity, Price, Timestamp and Status including unsent, sent, confirmed, allocated, failed.

The database entry can remain valid until it is manually removed by the user or until the handheld 114-116 receives allocation data for the execution. In one embodiment, the execution history database is implemented within a Windows CE database as opposed to being stored in volatile application memory to permit the data to exist across application sessions. In another embodiment, the contents of the database are deleted the first time that the handheld 114-116 application is launched each day. In another embodiment, the digital ink image can be discarded to conserve memory after the handheld 114-116 receives a message indicating that the execution has been allocated.

The online management system can communicate with the server in the context of a 'session'. A session uniquely identifies a handheld 114-116 and messages that have been sent to a particular handheld 114-116. Any response or message originating from the handheld 114-116 (except the initial login-request message, when the identifier is not yet available) will include the session identifier so that the server can correctly process the message information. Similarly, all messages arriving from HHS 113 will also contain the session identifier. Both HHS 113 and HHD can compare the session identifier, along with

other tracking information in the header of all messages, to their internal values to help determine if there has been a communications or application error.

Referring now to Fig. 3, the system can utilize a transmitted Heartbeat mechanism to determine if a current session is still active. Upon login, the system can initiate a heartbeat 310 between a handheld computing device 114-116 and the HHS 113. Typically, the online management will send heartbeat information packet to the HHS 113 during idle periods. The HHS 113 can log the time interval of the heartbeat 311. The server can also interpret the receipt of the heartbeat or any other message as an indication that the HHD session is still active. A test can be performed at the completion of a maximum heartbeat interval. The system can test to a heartbeat received within the predetermined time 312. The HHD will likewise interpret the receipt of the server heartbeat or any other message as an indication that the session is still alive 313. A successful heartbeat can loop the process back to initiate a heartbeat and begin the time interval logging. In one embodiment, if a message is not received within the specified timeout period, the online management system will assume that the session has terminated and will notify the user of the failure 314. Information that has not been sent will be lost except for executions stored in the Execution History database.

A Send Data Thread can wait for the SendData event, the Terminate event, or for the wait to timeout. The Terminate event signals that the thread should shut down. Signaling of the SendData event indicates that there is data to send. If the wait timeout occurs waiting for one of these two events, it is an indication that no traffic has been sent to the server. Therefore the thread will send a heartbeat to the server in accordance with the design of connection maintenance.

In one embodiment, a Receive Data Thread serves to block socket reads. The socket can be configured with a read timeout that is set to be equal to twice, or other multiple of, the heartbeat interval. If a socket read fails with a timeout error, wherein no messages

have been received from the server within the required interval, the connection from the handheld 114-116 to the server is assumed to be down. The thread can then call a routine, such as CloseSocket, and the user will be notified of the failure. Communication failures can result from socket termination, network failure, severe network latency, or a server or handheld application error.

In the event that communications are terminated, the User Interface can notify the broker of the failure. The handheld 114-116 can make an automatic attempt to re-establish communications with the server. In addition the broker can manually direct the handheld 114-116 to attempt to re-establish communications with the server.

In addition, a user can be allowed to operate in an "offline mode" which supports limited application functionality. This mode can be enabled, for example, to allow a broker to continue working should a failure occur at a critical moment, such as while executing orders in the crowd. In offline mode, the broker will not be able to send or receive looks or messages or orders. However, the broker will be able to record executions on the handheld 114-116. In this state, the handheld 114-116 essentially functions as a recording device for executions. These executions can be maintained in the Execution History database which the user would eventually need to reconcile with the clerk. e.g., in a manual process.

If the HHS 113 detects a communications failure, it can automatically send a new order that would otherwise be routed to a terminated handheld 114-116 to the booth responsible for the symbol corresponding with the security comprising the order. When the broker logs back in on the same or a different HHD, the broker will then automatically receive all of the active orders that are still assigned to them in the online management system. Orders that were pending can again be displayed as Pending; orders that were accepted will be automatically accepted on the HHD. However, in one embodiment, an order that was accepted on the HHD may return to the pending state if the "Order Accept" message

was lost during the communication failure. Orders that had been sent back to the booth during the communications failure can be "manually" sent back to the broker's HHD from the online management system. An exchange wireless infrastructure can be configured to provide two redundant networks. If a connection cannot be established on a current network, the HHD can prompt the broker to a "fail-over" mode in which the HHD will log into a backup network. The broker can assent or decline to perform a fail-over.

Referring now to Fig. 4, Allocation Estimation is a process for assigning specific quantities of shares that are traded to the orders that are eligible to participate in a trade. When a broker enters an execution 410 into a handheld computing device 114-116, a program in the handheld can estimate the quantity of shares that are allocated to various eligible orders 411 on the handheld 114-116. This allocation will estimate the amount of shares available for execution making up the various remaining orders. Along with the estimation 411, the handheld device 114-116 transmits the order information 412 to the associated booth station 161-163. A clerk can perform a final allocation for an execution in the booth 413. After the clerk finishes the allocations for an execution, the allocation can be transmitted to the HHS 113 and logged 414. In addition, the allocation can be transmitted to the handheld 114-116. When an actual allocation is received, the handheld can "unwind" the estimates by replacing the handheld's 114-116 estimated allocation with the final allocation determined in the booth. The execution quantity can exceed the sum of the remaining quantities of the eligible orders. In such cases the excess allocation quantity can be stored in an execution object. This excess quantity can remain and be factored into the total leaves for the affected security until an execution is unwound with the actual allocation from the booth.

One calculation that can be used for allocation estimates first determines the set of eligible orders and averages allocation qty for each order $\cong$ qty to allocate / no. of eligible orders. The quantity of shares allocated to an order can be limited to a multiple of

100. If the average qty < 100 then average qty = 100. The system can sort the eligible orders by the remaining quantity in ascending order. For each order the system can estimate an allocation = minimum of average estimate and order a remaining quantity if the estimated allocation is less than the average. A recalculation of a new average based on remaining shares can be allocated to remaining eligible orders. If a remaining quantity to allocate is 0 then the allocation routine can stop. Typical trading firm business rules and SEC regulations dictate that Agency orders receive priority over Principal orders. For example, if there are 700 shares to allocate to two orders that differ only in capacity, the Agency order must receive 400 shares and the Principal order 300.

In one embodiment, an estimation allocation can be performed irrespective of whether the handheld 114-116 is able to transmit the execution to the server. Estimates can thus be calculated even if a broker is working in offline mode. In another embodiment, an order will receive an allocation estimate for an execution when the order is for the same security as the execution and the order is for the same side as the execution. Accordingly, Long and Short Exempt sell orders can be eligible to participate in any sell execution, but Short sells can only be eligible for short sell executions. Therefore, if the side is Buy then the execution price should be lesser than or equal to the order price. If the execution is on the sell side then the execution price should be greater than or equal to the order price. A market order can satisfy any execution price.

The order centric system can also track an order timestamp. The order time stamp is the time the order reached the floor. An execution timestamp can be the time of execution. For an order to participate in an allocation, the order time stamp should be earlier than an execution time stamp, indicating that the order reached the floor before the execution was performed.

As noted above, the handheld 114-116 can operate with limited functionality if the handheld computing device 114-116 loses connection to the server. In addition, the application can enter Offline mode if a critical data error occurs. Offline mode can implemented in all layers of an online order application. For example, a Communication Manager can be responsible for detecting a lack of heartbeats and notifying the Data Manager. The Data Manager can disconnect the Communication Manager and notify the User Interface that orders functions and look functions and messaging are unavailable. The User Interface can also notify the broker and take the necessary actions to disable features as appropriate. For example, disabling can include disabling certain windows and/or ignoring user input such as stylus taps.

In one embodiment, a Send button included in an Orders dialog, such as on an Execution Entry page, will remain enabled even if the Orders system is unavailable. As a broker or other user enters and sends an execution while the Orders system is down, the execution can be persisted to the Execution History database and the user can receive a reminder that they must reconcile with the booth.

Failure Recovery can be implemented at Login time for a handheld computing device 114-116. An optional part of a Login-Reply message is a Server-Status element, which optionally contains the Recovery-Orders and Recovery-Executions elements. The Recovery-Orders element contains Order-Request messages for all of the orders that are currently assigned to a user logging in. These orders can be used to populate a Data Manager. This can be useful in the case where a broker logs in after a failure and his/her orders are still assigned to him/her; assuming that there have been no changes to orders in the online management system since the failure, the handheld 114-116 will be able display essentially the same information as when the failure occurred.

A Recovery-Executions message can contain a history of executions that have been entered into the online management system by the broker during the day. The Data Manager can use this data to update and/or reconstruct the Execution History database. The Data Manager can process the executions and bring the state of the Execution History database in line with what is currently in the online management system if that data is more recent. However items that were already in the database that are not present in this message will not be removed. It is considered an error if the status of such an entry is Confirmed or Allocated. The Data Manager can populate its lists with the contents of the Recovery-Orders element. After population is accomplished, the handheld 114-116 can perform an allocation estimation for any executions that have still not been allocated.

In the event there are un-reconciled executions stored on a handheld computing device 114-116, such as in the event of a network failure, the order centric system can reserve the execution history database in the database and refuse further use until the database is reconciled. The handheld computing device 114-116 can be programmed to record the User ID of a broker after a successful logon and compare this value to a User ID associated with an un-reconciled database stored in the handheld device 114-116. If the User ID logged in by the current broker is different from the User ID associated with the un-reconciled database the current will not be allowed access. This will effectively prevent executions from being overwritten.

A handheld computing device 114-116 can enter various states during normal use with an order centric tracking system. Table 1 illustrates specific examples of various handheld computing device 114-116 states and actions a programmable User Interface (UI) may associate with the specific states listed. In addition, Table 1 below illustrates examples of programmable functions, such as a DataManager function, a storage function, and a CommManager function that can be utilized with a handheld computing device in one

embodiment of an order centric tracking system. Programmable actions associated with each state are listed to further exemplify features of this invention.

State	UI	DataManager	Storage	CommManager
Application Startup	Display Application Start and Initial Security Screen	Create Comm Manager and initialize internal data structures.	Initialize storage, ensure all required databases exist.	Initialize connection to HHS 113, either through sockets or queuing software.
User Login	Display Login screen and capture user's credentials	Pass User Credentials to Comm Manager.	N/A	Connect to HHS 113 by sending a logon-request message. Wait for a logon-reply or a timeout, whichever comes first.
Login Successful	Clear connection status, display main application screen	Collect User Profile Information. Inform User Interface about successful login.	Conditionally remove previous storage information. Store Login Response and profile Information in Session History.	Return logon-reply received from HHS 113 to Data Manager.
Login Failure	Display Login failure dialog	Retrieve login response from the Comm Manager. Inform User Interface about the login failure.	N/A	Retrieve login response from HHS 113 and return it to Data Manager.
Order Request Pending	Display order with visual cues to identify as Pending	Process new order request from Comm Manager. Add the new Order to internal data structures. Update Securities list as required. Inform User Interface about arrival of new order request.	Store order information, log to History	Retrieve order request from HHS 113 and forward it to Data Manager.
Order Request Accepted	Update display to show that order is Accepted rather than Pending	Convey the change in order status to Comm Manager.	Mark stored order information as accepted, log to History	Retrieve order-reply message from outbound queue and send to HHS 113
Order Request Rejected	Remove order from UI	Send order-reply (rejected) message to Comm manager	Remove order, log to History	Retrieve order-reply from outbound queue and send to HHS 113
Execution Entry	Display Execution Entry form.	Instruct UI to display Execution Entry form, providing some default values depending on the context. After User has completed the execution, update internal data structure. Perform Leaves calculation. Send the new execution to Comm Manager.	Create an Execution record in the history database and an entry in the Executions database.	Retrieve order-execute from outbound queue and send to HHS 113.
Order change received	Display updated information.	Update Internal Data, pass to UI.	Update order information	N/A
Order Details Selected	Display Order Details form.	Instruct UI to display Order Details form.	Retrieve order data	N/A

Task History Display	Display list of completed tasks	Instruct UI to display task history information	Retrieve completed task information	N/A
Display Unallocated Executions	Display list of sent but unallocated executions	Instruct UI to display information	Retrieve executions information	N/A
User Logoff	Display logoff confirmation dialog	Instruct UI to display logging off status, send the current profile information to Comm Manager, notify it to terminate the session	N/A	Retrieve store-profile and logoff messages from outgoing queue and send to HHS 113
Network Failover	Display user instructions for network failure and device reset	Make modifications to the registry, prepare for reset, and instruct UI to give user instructions.	N/A	N/A
Application Error	Display application error message or show status indicator	Instruct UI to display appropriate message, possibly enter Offline mode.	Record error condition locally	N/A

Table 1

The invention may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention may be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output.

As will be appreciated, there are many ways in which specific high-level commands and data can be transferred between the handheld device ("HHD") client application and a handheld server ("HHS") application. Described below is a preferred high-level communication protocol definition which has been developed and is particularly suitable for use between a HHS and the various HHD clients, such as the Goldman Sachs Windows® CE Handheld Device, on a trading floor, such as the floor of the New York Stock Exchange ("NYSE"). The protocol is specifically designed to provide enhanced order

management features and more streamlined communications as compared to existing, but more general protocols, such as the Financial Information Exchange ("FIX") Protocol which is designed to facilitate the electronic exchange of information related to securities transactions.

Fig. 5 is an alternative representation of the architecture of an order centric tracking system 100 according to the invention, which representation is useful for discussing the system messaging between the handheld devices. As illustrated, this embodiment of the system 100 includes a electronic Floor Order Management system cluster 130 (alternatively refered to as "FOMAD") which is connected between the trader 120 and one or more floor order management systems 161,163 ("FOMS") which are situated in respective trading booths 110, 112 staffed by booth clerks. Also connected to the FOMAD 130 is the Handheld Server System (HHS) 113 which is connected to the various Handheld devices (HHDs) 114, 116 through a wireless network 502 to permit communication with the respective brokers 514, 516.

In addition to placing orders, a trader 120 can also request looks for one or more securities. Although they can be a precursor to an order, looks and orders are generally independent. Thus, while looks can be processed through the FOMAD 130, These looks are generally independent of the orders. Accordingly, and as shown in Fig. 5, a separate Floor Look System (FLS) 504 can be provided and connected directly to the HHS 113 to allow looks to be forwarded to the brokers via their handheld devices.

FIG. 6 is a diagram illustrating sample process and messaging flow between a HHD 114, the HHS 113 and various other elements in the network 100 for sample look and order requests. When first connecting to the system, the HHD initiates a socket connection to the HHS. Once the socket is established, a Logon-Request is sent from the HHD to HHS. The login request is processed and the HHS sends an appropriate login-reply. Provided the

login is successful, the HHD is now active on the system and tagged as being in use by a particular broker.

A look request querying the current market conditions for a given security sent from a Trader 120 through the FLS 504 is processed by the HHS 113 and forwarded to the appropriate broker, e.g., in accordance with a look-up table indicating which brokers are assigned to particular securities. A look-reply indicating whether the trader operating the HHD has accepted or rejected the look request is returned to the Trader via the HHS and FLS. If the look was accepted, a further Look-data message containing the information entered by the trader in response to the look is returned to the Trader via the HHS and FLS.in an encoded graphical format.

Once a decision has been made to initiate a transaction, a order is sent from the trader to the FOMAD system 130 and subsequently forwarded to FOMS system for the booth which specializes in the particular security at issue in the order. Automatically, or under the direction of a booth clerk, the order is forwarded to the HHS which then transmits it to the HHD of the appropriate broker in the form of an Order-Request message. An order reply message is returned to the HHS and thereby to the FOMS and booth clerk indicating whether the order has been accepted by the floor trader. If the order has been rejected, the order can be reassigned to a different trader.

When the trade has been executed, the broker enters the appropriate information into their HHD. This information is conveyed to the FOMS via an Execution message containing details about the trade sent to the HHS. An Execution-Acknowledge message is returned to the HHD indicating the status of the transaction as it is processed by the appropriate electronic order management system.

If the execution is to be split among several orders, the order is allocated, e.g., by the booth clerk, and details of the execution are sent through the FOMS to the HHS. This

information is then forwarded to the HHD in the form of an Execution-Allocation message indicating the collection of orders affected by the identified execution sequence and allocation.

Once the broker has completed their trading session, the session is terminated. This is accomplished by means of a Logoff-Notification message sent from the HHS to the HHD. After logoff, the HHD closes the data socket.

The specific protocol disclosed herein is preferably implemented using an Extensible Markup Language (XML) to allow the protocol to be easily overlayed on a conventional HTTP data communications network. However, the protocol can also be mapped into other data transmission languages. In addition, and as will be appreciated, various lower layer communication protocols will also be utilized in the communication system and intervening hardware elements can be present between the HDD and the HHS, such as infrastructure components of the NYSE. Further, in view of the extensibilty and maintainability of XML based communication, and to improve the ability to debug communication problems, preferably all messages are suitable for transport as plain text, e.g., as base64 encoded data. However, other data encoding techniques can also be used in alternative embodiments.

In a preferred XML implementation, the protocol syntax is specified in an XML Document Type Definition ("DTD"). This permits the an XML parser layer in the communication path to automatically detect and report syntax errors in the messages. One complete implementation of the DTD for the most preferred implementation of the protocol is attached as Appendix A. The general messaging format will be discussed below followed by a detailed description of the specific messaging op-codes and the various semantic sequences of the protocol.

Fig. 7 is an illustration of a sample message. In this and other message examples, the indentation is for readability. In practice, line breaks or extra whitespace characters are not required to be present in the data stream. Within the XML framework, and with reference to the sample message of Fig. 7, each message in the handheld protocol contains an identifying tag, such as "<GSHP>", at its root followed by a standard set of attributes attached directly to the GSHP tag. These attributes include tracking information in the form of a Session ID ("SID"), a Sequence Number ("Seq"), and a Last Received Sequence Number ("LRSeq"). Each sending station is responsible for generating its own sequence numbers, which number is incremented with each outbound message. As viewed from a HHD, the starting sequence number is contained in the Logon-Request, and incremented with every message outbound from that HHD. The LRSeq field contains the last received sequence number. For example, with every outbound message from the HHD to the HHS, the HHD will fill in LRSeq with the sequence number of the last message received from the HHS. The HHS spawns a new series of sequence number for each new session. This enables detection of gaps in the messaging streams by the HHD and the HHS. It should be noted that the Session ID is assigned as part of the Logon-Reply and thus the initial Logon-Request message will have a SID="0".

After the initial GSHP tag, the message can include one or more op-codes, specified as an XML tags in the message. Attributes are associated with the op-code as name-value pairs. Further sub-items can also be included under each op-code. In accordance with the preferred implementation, and to simplify processing, tracking, error recovery, and accountability of messages, only a single op-code is included in the sub-item under the main tag and all attributed defined for the op-code are required elements of the message. Further, no items in the message will have implied values. However, in a less rigorous alternative implementation, multiple op-codes may be transmitted and, in addition, various of the

defined attributes may be omitted, wherein omitted attributes are considered to have a predefined value or a value defined through alternative means.

To further simplify processing of messages between the HHS and HHD, all messages sent using this protocol preferably have the following lightweight envelope immediately preceding the primary XML block:

Description	Code	Length
Header	H	1 char
Underscore	_	1 char
Protocol	GSHP	4 char
Lngth	XXXXXXXX	8 char (left padded with 0's)
CRLF Terminator	<CRLF>	2 char

The envelope header is always a fixed 16 bytes in length, including the trailing CRLF. The length encoded into the header corresponds to the length of the main GSHP message which begins following the CRLF. By specifying the length of the message in this envelope, the HHD and HHS do not need to scan incoming messages to for a terminator character, thus simplifying input processing.

The various op-codes which are supported in a preferred implementation of the XML communication protocol will now be discussed. In general, each op-code is transmitted in an individual message unless otherwise stated. In addition, in a preferred embodiment, the protocol is implemented with the complete op-code set disclosed below. However, as will be appreciated by those of skill in the art, various defined op-codes may be omitted without substantially impacting the overall functionality and usefulness of the system, such as op-codes used generally for handling corrupted messages or which provide brokers with additional flexibility, such as a "do-no-disturb" option. To the extent that a

specific attribute or sub-code has not been explicitly discussed above, it is believed that the definition of such an attribute or sub-code is apparent with reference to the contextual usage and the various specific examples provided.

Many op-codes utilize name-value pairs to associate a name field with one or more data items contained in a value field. To implement this pairing a *Pair* op-code is defined. This op-code can be used within a message but cannot be used as a stand-alone message. For example, it is used within the Logon-Reply and Logoff-Notify messages. The *Pair* op-code includes two attributes, *Name*, which specifies the name at issue, and *Value*, which contains the information associated with the value data. For example, the statement:

<*Pair* Name="Stocks" Value="IBM,DEC,GE,ATT"/>

would associate "Stocks" with the specified stock symbols.

Logon-Request

This message originates in a HHD and is used to initiate a logon procedure. The defined attributes include:

* *UserID*, which specifies a valid Broker User ID;
* *Password*, which is the user's access password; and
* *Force*, which is used to indicate to the system whether or not to force a new logon when the server's state shows that the user is already logged on. Valid values for the *Force* attribute are "Y" or "N".

Also defined are the following optional sub-item: <*Logon-Request-Version* Value="xxx"/>. This sub-item can be used to indicate the version of the XML DTD to use during the new login session. This item simplifies the use of multiple versions of a protocol DTD in the system and simplifies changing from one protocol to another.

A sample Logon-Request message is shown below:

<*Logon-Request* UserID="BryanV" Password="StocksRUs" Force="N"/>

Logon-Reply

This message originates at the HHS and is directed to the HHD in response to a Logon-Request. In the logon reply, the HHS indicates whether the logon was successful or not in the defined attribute *Success*, which contains the value "Y" for a successful logon, or the value "N" for a failed logon. Several sub-items are also defined and which can be used to indicate the heartbeat, various timeout values, a predefined profile for the user logging in, specific special instructions, the present official time, and the status of various hardware systems in the network. Within the status information, various recovery messages can be specified to re-send pending order and execution, etc. to the HHD, which messages may have been dropped or otherwise lost when the broker using the HHD previously logged out. A sample Login-Reply message is illustrated in Fig. 8.

The specifically defined sub-items for this op-code are as follows:

* <*Heartbeat-Period* Seconds="ss" />: Identifies the frequency in seconds of network maintenance messages. The HHD and HHS will each send a heartbeat message if this amount of time elapses with no outbound messages.

* <*Look-Reply-Timeout* Seconds="ss" />: The number of seconds the device should provide visual feedback to the user about a new look request. If the user has not taken action to accept or reject within this period, then the request is passively rejected.

* <*Look-Data-Timeout* Minutes="mm">: The number of minutes the device may "sit on" an accepted look request. In an example scenario, the broker agrees to perform a look, but then takes no action to fulfill that promise. After the specified timeout, the pending look will simply drop off of the handheld, and the accepted request will be passively rejected.

* <*Timeset* Time="UTC"/>: The current server time sync in universal time format.

Preferably, the *Heartbeat-Period*, *Look-Reply-Timeout*, and *Look-Data-Timeout* sub-items are required elements of a Logon-Reply. However, predefined values can

be specified for use in their absence. Preferably, *Timeset* is also a required sub-item. However, alternative mechanisms for coordinating the time between the server and the various handheld units known to those of skill in the art can also be used, such as providing broadcasting a UTC time signal on a predefined radio channel within the trading environment and including an appropriate receiver in each HHD.

The following optional sub-items are also defined:

* *<Profile-Data> ... </>*: This sub-item is an op-code (which can be used within a message, but not as a stand-alone message) contains one or more name/value pairs that describe the custom settings previously specified by the user and stored within the system. has selected. These name/value pairs are described within the Pair op-code.

* *<Sp-Instr>*: This item contains name/value pairs that describe the abbreviations for various special execution instructions defined within the Pair op-code.

* *<Server-Status>*: This is used to specify the status of various hardware elements within the trading system. In addition, various recovery messages can be embedded as sub-items to the *Server-Status* item. This item is specifically discussed in more detail below.

Logoff-Notify

This message originates at the HHD and is used to inform the system that the broker is logging off. There are no required attributes. Preferably, a *Profile-Data* sub-item is defined under which the HHD can supply name/value pairs which describe the custom settings that the broker using the HHD has selected. This profile information can be stored in the system and returned to HHD under the *Profile-Data* item in the Logon-Reply message. Sample Logoff-Notify messages are illustrated in Fig. 9.

Look-Request

This message originates in the HHS (in response to a look requested by a Trader and forwarded to the HHS). A sample Look-Request message is shown in Fig. 10.

The message attributes are *ReqID*, which specifies an assigned Look Request ID number, *Symbol*, which identifies the stock symbol for which the look is being requested, and *FromAddr*, which indicates the address of the Look Request originator. In addition, one or more of the following optional sub-items can be included in the message:

* *<Look-Reply-Timeout* Seconds="ss"/>: Used to override the default timeout issued in the Logon-Reply. If provided, it is in effect for this look only.
* *<Text>* ... </>: A text message provided as part of the look-request.
* *<Priority* value="high" />: Indicates priority of the request. Valid entries include "high" and "normal". If this field is not present, a normal or default priority can be assumed.
* *<Look-Data-Timeout* Minutes="mm" />: The number of minutes that an accepted look request will live before it is passively rejected if the broker does not either fulfill or actively reject the request. This value is used on a per-look-request basis to override the default value that is received in the Logon-Reply message.

Look-Reply

This message originates in the HHD and is used to indicate whether a broker has accepted or declined a specific look request. The following, preferably required, attributes are defined:

* *ReqID*: The Look Request ID.
* *ToAddr*: The address of the recipient.
* *Symbol*: The stock symbol.
* *Accept*: The Accept attribute may be "Y", "N", or "PR". (PR is for Passive-Reject)

No particular sub-items are defined for this op-code. A sample Look-Reply message is shown below:

<Look-Reply ReqID="123" ToAddr="FLS" Symbol="GE" Accept="Y"/>

Look-Data

This message also originates in the HHD and is used to convey the results of a look back to the system, and ultimately to the requesting trader. The particular look request the data is associated with and its routing is specified in the following attributes:

* *ReqID*: The Look Request ID.
* *ToAddr*: The address of the recipient.
* *Symbol*: The stock symbol.

A required *Ink* sub-item is also defined and used to contain the specific look data. To simplify data entry and processing on the HHD, preferably the Look data is entered as graphical data and transmitted in an encoded graphical image format, as opposed to plain-text. The specific image format used in the transmission is specified as an attribute of this sub-item. For example, <*Ink* Format="Stroke-Vector"> ...[data]... </> specifies that the embedded data is in Stroke-Vector format. Alternative formats, such as GIF89a can also be supported. Preferably, the graphical data is base64 encoded digital ink, which format permits the image data to be represented as valid text characters.

A <*Priority* value="xxx" /> sub-item can also be include to indicate the priority of information. Valid entries include "high" and "normal". If this field is not present, then normal priority is assumed. A sample Look-Data message is shown in Fig. 11.

Msg-Request

This op-code is used to transmit a text statement, e.g., from the trader, via the HHS, and directly to the HHD of a broker. The message includes the attributes of *ReqID*, which specifies the Request ID, and *FromAddr*, which identifies the address of the originator. The sub-item <*Text*> ... </> is required and contains the text of the message. An optional *priority* sub-item can also be included. A sample Msg-Request message is shown in Fig. 12.

Msg-Reply

This message is used to respond to a previously received Msg-Request. It can also be used to originate a new message to a specified ID. The required attributes are *ReqID* and *ToAddr*, which identifies the destination address of the message. The following optional sub-items can also be included in the message:

* *<Ink>* ... *</>*: Digital ink of the Msg-Reply
* *<Text>...></>*: Text of the message.
* *<Priority* value="xxx" />: Indicates the priority of the reply.

An example Msg-Reply message is shown in Fig. 13.

DND-Notify

This message originates at the HHD and is used to indicate to the system whether the HHD's "do-not-disturb" switch is turned on or off. Acceptable values are "On" and "Off". An example of this message is:

<DND-Notify Switch="On"/>.

Heartbeat-Notify

This message can originate at either of the HHD and HHS and is used to send a heartbeat signal indicating that the sending device is still active. This message is sent without any associated attributes or sub-items, e.g.:

<Heartbeat-Notify/>

NACK

This message can originate at either of the HHD and HHS and is used for error logging and debugging when a received message is corrupted. The op-code includes a *Reason* attribute which indicates the condition which cause the NACK message to be generated and a single required sub-item, *Original*, which contains the offending text in its entirety. The original text, because it may be malformed, can be bracketed within an

"Escaped Character Data" section, indicating that the parser should not attempt make sense of that particular section. A sample NACK message is shown in Fig. 14.

Set-Profile-Request

This message originates in the HHD and is used to pass a set of name-value pairs which contain user profile information to the system for storage via a required *pair* sub-item. Any number of pairs may be present. If the message is sent with no pairs, it can be interpreted as requesting a deletion of the user's profile. For example:

<*Set-Profile-Request*> <*Pair* Name="Stocks" Value="IBM,DEC,GE,ATT"/> </>

Set-Profile-Reply

This message is returned to the HHD from the HHS in response to a Set-Profile-Request message its single attribute *Success* contains "Y" or "N" as an indication of the status of the Set-Profile-Request message. A successful reply message would be:

<*Set-Profile-Reply* Success="Y"/>

Order-Request

This message is generated in the HHS in response to an order request. The message includes the following attributes:

* *ID*: The FOMAD order ID.

* *Stat*: The hand-held device status, relative to the present order. Possible values include "OK", "CORRECT-PENDING", "CANCEL", or NULL. This generally corresponds with the FOMAD HHD-Status field. A status value other than NULL indicates that the broker had previously accepted the order and somehow lost connection with the server without correctly logging off. (Various other op-codes also include a Status attribute. A set of valid status values used by this and other op-codes is provided in Fig. 15. It should be noted that not all status values are appropriate for a given op-code)

* *Sym*: The securities trading symbol.

* *Price*: The order price. This can be a specific price or an indication to complete the transaction at the current "market" price.. Possible examples are "100 3/16" and "MKT".

* *Qty*: The (original) order quantity.

* *Side*: Indicates the position of the order. Possible values are "BUY" and "SELL".

* *RemQty*: The remaining quantity for the order (where the order may have previously been partially processed).

* *Timestamp*: The FOMAD time of the order.

* *FVer*: The FOMAD version number.

* *Cap*: The Capacity or role of the party on whose behalf the order has been placed.

* *Num*: The order number

* *Booth*: The booth identifier

* *Trader*: The trader identifier

* *Ticket*: The ticket number

In addition to these attributes, one or more other attributes can be defined to pass special instructions, indicate particular order flags, expiration times, or other special instructions. In a preferred embodiment, the following additional attributes are also defined:

* *ExecInst*: The execution instructions

* *Flags*: The order flags

* *TIF*: The time in force

* *SpInst*: Contains special instructions for the completion of the order.

An example Order-Request message is illustrated in Fig. 16.

Order-Reply

This is a message which originates in the HHD and indicates whether the broker has accepted or rejected a specific previous order. The op-code contains the following attributes:

* *ID*: The FOMAD order ID.

* *Stat*: The status of the order. This indicates whether or not the HHD has accepted or rejected the order. Possible values include "OK", "REJECT", "PASSIVE-REJECT", which is indicated when the order has been pending on the HHD for a predetermined period of time without an explicit acceptance or rejection by the broker. Additional status fields can include "CORRECT-ACCEPT", and "CORRECT-REJECT", for use in situations where a previously transmitted order must be corrected. If the broker has not already acted on the order, the correction may be accepted. If order processing has begun, it may be impossible to correct, in which case the correction must be rejected.

In implementations where multiple versions of the FOMAD system are in use, a *FVer*: attribute can be provided to specify the FOMAD version number. An example Order-Reply message is:

<*Order-Reply* ID="123" Stat="OK" FVer="2"/>

Order-Recall

This message originates at the HHS, in response to actions entered by the clerk, trader, or other appropriate personnel. The purpose behind this op-code is to initiate action on the HHD to remove the specified order from the HHD, thus pulling it away from the broker. Appropriate notice to the broker when this action has been taken may be generated by the HHD. The attributes for this op-code can include *ID*, the FOMAD order ID, and, if multiple versions of FOMAD are in use, *Fver*, the FOMAD version number. For example:

<*Order-Recall* ID="123" FVer="2"/>

Execution

This message is generated by the HHD and is used to convey an execution of a transaction, such as the buying or selling of a set of securities. Execution messages can be

generated without specific reference to a given order. The attributes defined for this op-code include:

* *ID*: HHD-generated execution ID.
* *Sym*: The securities trading symbol.
* *Price*: The execution price.
* *Qty*: The execution quantity.
* *Side*: Indicates the position of the execution. Possible values are "BUY" and "SELL".
* *Timestamp*: The timestamp of the execution.

In addition, an *Ink* sub-item, preferably required, is defined for use in transmitting a graphically encoded message written by the broker and containing information about the execution. The *Ink* sub-item is discussed above. An example of an Execution message is shown in Fig. 17.

Execution-Ack:

This message is transmitted from the HHS and indicates to the HHD that the specified Execution message was received and acknowledged. The Execution-Ack op-code includes as attributes one or more identifier fields, which are used to uniquely identify the execution at issue across multiple HHDs (since two separate HHDs could possibly generate the same execution ID number). In preferred embodiment, an *ID* sub-item containing the associated HHD Execution ID, and an *FID* sub-item which represents the FOMAD Execution ID are used. A value of –1 can be included in this field to indicate that a FOMAD transaction error has occurred and a representation of the error code will be placed in a status attribute, *Stat*, which represents the status of the FOMAD transaction. Possible values include "EX_ERR" and NULL. A NULL value indicates a no error condition. A *Fver* attribute for specifying the a FOMAD version number can also be defined if appropriate. For example:

<*Execution-Ack* ID="123" FID="3876402" Stat="EX_OK" FVer="2"/>

Execution-Allocation

As discussed above, when an execution is performed, the results of the execution may be allocated among several pending orders. The allocation information is generated by the FOMS clerk and forwarded to the broker via the HHS. The defined attributes include an *ID*, which represents the HHD Execution ID, and an *FID*, which represents the FOMAD Execution ID.

In addition, a required sub-item <*Allocations*> ... </> is included and used to communicate the collection of orders affected by this allocation in the form of one or more included *Allocation* op-code statements. The *Allocations* op-code can be used within a message but cannot be used as a stand-alone message. It is specifically used within the Execution-Allocation message. The Allocation op-code includes the following attributes:

* *ID*: Represents the HHD Execution ID.
* *QtyAlloc*: The quantity of shares allocated to this order.
* *Qty*: The current leaves for this order.
* *FVer*: The FOMAD version number.

An example of an *Execution-Allocation* message with an *Allocations* block containing sample *Allocation* statements is shown in Fig. 18.

Timeset:

In addition to being a stand-alone message, this op-code can also be included as a sub-item to another message, such as the Logon-Reply message discussed above. The attribute is the official time, such as the time of the Server in UTC format. For example:

<*Timeset* Time="19980106174100"/>

Prices:

This message is sent from the HHS and is used to communicate to the broker the most current price associated with one or more securities. The Prices op-code includes a *Time* attribute which contains a timestamp for the indicated prices. At least one <*Price*> ...

</> sub item must be included to specify the price of a given security. A sample *Prices* message is shown in Fig. 19. Advantageously, this message can be used to quickly send a broker, via their HHD, a snapshot of the last sale price for all securities of relevance to the broker. For example, the system can send price information to a broker on a periodic basis, for example, for all securities that they have specified in their user profile or only those securities for which the broker has outstanding orders or looks.

Server-Status:

This op-code is intended to be used as a sub-item message within another message. As discussed above with respect to the Logon-Reply message, the Server-Status message is used to indicate initial server status and also to pass order and execution recovery messages to the HHD. The defined attributes include:

* *FLS*: Status of the FLS system. Possible values are "UP" or "DOWN".
* *FOMAD*: Status of the FOMAD system. Possible values are "UP" or "DOWN".
* *Already-Logged-On*: Indicates that the user was already logged on to the system.

Additional optional sub-items include:

* *<Recovery-Orders>* ... </>: zero or more order request messages can be embedded in the server status message. Embedded messages are only present when the server status message is included in the Logon Reply message. The *Recovery-Orders* op-code can be used within a message but preferably cannot be used as a stand-alone message and most preferably is used within the Server-Status message only when the Server-Status message is embedded within a Logon-Reply message. This op-code indicates that the current logon is in a state of recovery and that previously accepted orders are being re-sent to the HHD. One or more *<Order-Request>* ... </> messages can be embedded within the Recovery-Orders op-code and used to retransmit included orders.

* *<Recovery-Executions> ... </>*: zero or more Execution messages can be embedded in the server status message. Embedded messages are only present when the server status message is included in the Logon Reply message. As with the *Recovery-Orders* op-code, preferably the Recover-Executions op-code can be used within another message but cannot be used as a stand-alone message and is used within the Server-Status message only when the Server-Status message is embedded within a Logon-Reply message. This op-code indicates that the current logon is in a state of recovery and that previously performed executions are being re-sent to the HHD. The *Recover-Executions* op-code has no defined attributes. One or more *<Execution> ... </>* sub items can be embedded within the Recovery-Executions op-code.

An example of a Server-Status message with an imbedded *Recovery-Orders* and *Recover-Executions* messages is shown in Fig. 20.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the sprit and scope of the invention.

APPENDIX A: Representative XML Document Type Definition for Handheld Protocol

```
<!-- Define what is allowed inside of the GSHP tags -->
<!ELEMENT GSHP (Logon-Request          |
                Logon-Reply            |
                Logoff-Notify          |
                DND-Notify             |
                Heartbeat-Notify       |
                NACK                   |
                Msg-Request            |
                Msg-Reply              |
                Look-Request           |
                Look-Reply             |
                Look-Data              |
                Profile-Data           |
                Profile-Data-Reply     |
                Order-Request          |
                Order-Reply            |
                Order-Recall           |
                Execution              |
                Execution-Ack          |
                Execution-Allocation   |
                Timeset                |
                Server-Status          |
                Prices                 |
                Page-Request           |
               Sp-Instr            |
                       Version)>
<!ATTLIST GSHP
          SID      CDATA                  #REQUIRED
          Seq      CDATA                  #REQUIRED
          LRSeq    CDATA                  #REQUIRED>

<!ELEMENT Logon-Request (Logon-Request-Version?)>
<!ATTLIST Logon-Request
          UserId   CDATA #REQUIRED
          Password CDATA #REQUIRED
          Force    CDATA #REQUIRED>

<!-- If the Logon is unsuccessful, none of the sub-items are
required -->
<!ELEMENT Logon-Reply (  Heartbeat-Period?,
                        Order-Reply-Timeout?,
                          Look-Reply-Timeout?,
                          Look-Data-Timeout?,
                          Profile-Data?,
                          Server-Status?,
                        Sp-Instr?,
                          Timeset?)>
<!ATTLIST Logon-Reply
          Success (Y|N) #REQUIRED>

<!ELEMENT Logoff-Notify (Profile-Data)?>
```

```
<!ELEMENT DND-Notify EMPTY>
<!ATTLIST DND-Notify
          Switch (On|Off) #REQUIRED>

<!ELEMENT Heartbeat-Notify EMPTY>

<!ELEMENT  Look-Request  (Look-Reply-Timeout?,Text?,Priority?,Look-
Data-Timeout?)>
<!ATTLIST Look-Request
          ReqID    CDATA #REQUIRED
          Symbol   CDATA #REQUIRED
          FromAddr CDATA #REQUIRED>

<!ELEMENT Look-Reply EMPTY>
<!ATTLIST Look-Reply
          ReqID    CDATA #REQUIRED
          ToAddr   CDATA #REQUIRED
          Symbol   CDATA #REQUIRED
          Accept   (Y|N|PR) #REQUIRED>

<!ELEMENT Look-Data (Ink,Priority?)>
<!ATTLIST Look-Data
          ReqID    CDATA #REQUIRED
          ToAddr   CDATA #REQUIRED
          Symbol   CDATA #REQUIRED>

<!ELEMENT NACK (Original)>
<!ATTLIST NACK
          Reason  (malformed) #REQUIRED>

<!ELEMENT Profile-Data-Reply EMPTY>
<!ATTLIST Profile-Data-Reply
          Success CDATA #REQUIRED>

<!ELEMENT Msg-Request (Priority?,Text)>
<!ATTLIST Msg-Request
                ReqID    CDATA #REQUIRED
          FromAddr CDATA #REQUIRED>

<!ELEMENT Msg-Reply. (Ink?,Priority?,Text?)>
<!ATTLIST Msg-Reply
          ReqID  CDATA #REQUIRED
          ToAddr CDATA #REQUIRED>

<!ELEMENT Order-Request EMPTY>
<!ATTLIST Order-Request
          ID        CDATA #REQUIRED
          Stat      CDATA #REQUIRED
          Sym       CDATA #REQUIRED
          Price     CDATA #REQUIRED
          Qty       CDATA #REQUIRED
          Side      CDATA #REQUIRED
          RemQty    CDATA #REQUIRED
                Timestamp CDATA #REQUIRED
                FVer      CDATA #REQUIRED
```

```
                Cap         CDATA #REQUIRED
                Num         CDATA #REQUIRED
                Booth       CDATA #REQUIRED
                Trader      CDATA #REQUIRED
                Ticket      CDATA #REQUIRED
                ExecInst    CDATA #REQUIRED
                Flags       CDATA #REQUIRED
                SpInst      CDATA #REQUIRED
                TIF         CDATA #REQUIRED>

<!ELEMENT Order-Reply EMPTY>
<!ATTLIST Order-Reply
          ID      CDATA #REQUIRED
          Stat    CDATA #REQUIRED
                  FVer   CDATA #REQUIRED>

<!ELEMENT Order-Recall EMPTY>
<!ATTLIST Order-Recall
          ID    CDATA #REQUIRED
                FVer CDATA #REQUIRED>

<!ELEMENT Execution (Ink?)>
<!ATTLIST Execution
          ID        CDATA #REQUIRED
                FID    CDATA #REQUIRED
          Sym       CDATA #REQUIRED
          Price     CDATA #REQUIRED
          Qty       CDATA #REQUIRED
          Side      CDATA #REQUIRED
                Stat      CDATA #REQUIRED
                Timestamp CDATA #REQUIRED>

<!ELEMENT Execution-Ack EMPTY>
<!ATTLIST Execution-Ack
          ID    CDATA #REQUIRED
          FID   CDATA #REQUIRED
          Stat  CDATA #REQUIRED
                FVer CDATA #REQUIRED>

<!ELEMENT Execution-Allocation (Allocations+)>
<!ATTLIST Execution-Allocation
          ID    CDATA #REQUIRED
          FID   CDATA #REQUIRED>

<!ELEMENT Server-Status (Recovery-Orders?, Recovery-Executions?)>
<!ATTLIST Server-Status
          FLS               CDATA #REQUIRED
          FOMAD             CDATA #REQUIRED
          Already-Logged-On CDATA #REQUIRED>

<!ELEMENT Page-Request EMPTY>
<!ATTLIST Page-Request
          Look (Y|N) #REQUIRED>

<!ELEMENT Timeset EMPTY>
<!ATTLIST Timeset Time CDATA #REQUIRED>
```

```
<!ELEMENT Prices (Price+)>
<!ATTLIST Prices Time CDATA #REQUIRED>

<!ELEMENT Sp-Instr (Pair+)>

<!-- The following can only be used as sub-items -->

<!ELEMENT Heartbeat-Period EMPTY>
<!ATTLIST Heartbeat-Period Seconds CDATA #REQUIRED>

<!ELEMENT Order-Reply-Timeout EMPTY>
<!ATTLIST Order-Reply-Timeout Seconds CDATA #REQUIRED>

<!ELEMENT Look-Reply-Timeout EMPTY>
<!ATTLIST Look-Reply-Timeout Seconds CDATA #REQUIRED>

<!ELEMENT Look-Data-Timeout EMPTY>
<!ATTLIST Look-Data-Timeout Minutes CDATA #REQUIRED>

<!ELEMENT Logon-Request-Version EMPTY>
<!ATTLIST Logon-Request-Version Value CDATA #REQUIRED>

<!ELEMENT Pair EMPTY>
<!ATTLIST Pair
          Name  CDATA #REQUIRED
          Value CDATA #REQUIRED>

<!ELEMENT Recovery-Orders (Order-Request+)>
<!ELEMENT Recovery-Executions (Execution+)>

<!ELEMENT Allocations (Allocation+)>

<!ELEMENT Allocation EMPTY>
<!ATTLIST Allocation
          ID         CDATA #REQUIRED
          QtyAlloc   CDATA #REQUIRED
          Qty        CDATA #REQUIRED
                 FVer    CDATA #REQUIRED>

<!ELEMENT Priority EMPTY>
<!ATTLIST Priority Value (Normal|High) #REQUIRED>

<!ELEMENT Price EMPTY>
<!ATTLIST Price
          Sym   CDATA #REQUIRED
          Price CDATA #REQUIRED>

<!ELEMENT Profile-Data (Pair+)>

<!ELEMENT Text (#PCDATA)>

<!ELEMENT Ink (#PCDATA)>
<!ATTLIST Ink Format (Stroke-Vector|GIF89a) #REQUIRED>

<!ELEMENT Original (#PCDATA)>
```

```
<! -- MsBroker compares its internal DTD version to the value here -
->
<!ELEMENT Version EMPTY>
<!ATTLIST Version MsBroker (0200) #REQUIRED>
```

APPENDIX B: Glossary of Common Trading Terms

Allocation – The distribution of the shares traded in an execution across the orders that are eligible to participate in the trade.

Break – A failure to reconcile an order that has been filled, either within a member firm or between member firms.

Breakdown – Detailed information about an execution, including quantity and contra party. An execution consists of one or more breakdowns.

Contra – The opposing broker in a trade, e.g., as identified by brokerage and badge number.

Digital Ink - A graphical representation of handwritten information, somewhat akin to a scanned/faxed image. Digital ink may be condensed in a no-loss manner for transmission.

Execution – The act of performing a trade, and/or the information (price and quantity) that describes it.

FOMAD – A particular floor order management architecture which implements a particular OMA.

FOMS – A client application which is part of the order management system that the clerks use in the booths to assist with order management.

HHD – A Hand Held Device, such as the piece of hardware used by brokers to access the wireless network.

HHS – A server application that routes communication between the HHDs and the other system components. The HHS also interfaces with FOMAD on behalf of the HHDs.

Leave –The remaining quantity of an active order.

Look – A set of information, generally written on a small piece of paper or captured digitally on a handheld device, that provides an indication of the current market conditions for a given security. This information may include current ask and bid prices, volume estimates, parties involved in trading, and broker sentiment. The information is written in a commonly understood format. Also referred to as a "Floor Picture."

OMA – Order Management Architecture.

Order – An instruction to trade a security. An order contains all of the information that a broker needs to perform a trade. The term also refers to a representation of this information, such as digital or on a piece of paper.

UTC – Universal Coordinated Time, which refers to the time as set by the World Time Standard. Previously referred to as Greenwich Mean time or GMT.

Verbal – The price and quantity of an execution, as in the "verbal confirmation".

Written – The breakdowns that fully describe an execution, as in the "written confirmation".

CLAIMS:

1. A system for processing an order for a security on the floor of an exchange as shown and described herein.

2. A method for processing an order for a security on the floor of an exchange as shown and described herein.

3. A system for communicating between a handheld trading device and a trading system as shown and described herein.

4. A method for communicating between a handheld trading device and a trading system as shown and described herein.

5. A communications protocol for use in passing messages between a handheld trading device and a handheld server system connected to a trading system as shown and described herein.

6. A computer readable medium containing computer code defining a communications protocol for use in passing messages between a handheld trading device and a handheld server system connected to a trading system as shown and described herein.