Systems and methods are provided for processing mass quote messages and generating market data. A mass quote message (202) is received and individual orders are parsed and processed. Individual market data messages are stored in a market data message buffer (210). After all orders are processed, the contents of the market data message buffer are distributed as a single market data message. A messaging structure (500) allows for adding, re-ordering and/or expanding data, within the printable character set of any language. One or more delimiters are defined and used to delimit data elements within the message structure (500). The data is interpreted based on templates (501) which may be disseminated prior to the sending of messages.
Title: TRADE ENGINE PROCESSING OF MASS QUOTE MESSAGES WITH MARKET DATA MESSAGE FORMAT

Abstract: Systems and methods are provided for processing mass quote messages and generating market data. A mass quote message (202) is received and individual orders are parsed and processed. Individual market data messages are stored in a market data message buffer (210). After all orders are processed, the contents of the market data message buffer are distributed as a single market data message. A messaging structure (500) allows for adding, reordering and/or expanding data, within the printable character set of any language. One or more delimiters are defined and used to delimit data elements within the message structure (500). The data is interpreted based on templates (501) which may be disseminated prior to the sending of messages.
TRADE ENGINE PROCESSING OF MASS QUOTE MESSAGES
WITH MARKET DATA MESSAGE FORMAT

[01] The present application claims the benefit of U.S. Application No. 10/903,826, filed July 30, 2004. The present application also claims the benefit of U.S. Provisional patent application No. 60/517,491, filed November 5, 2003, the entire disclosures of both applications are hereby incorporated by reference.

FIELD OF THE INVENTION

[02] Aspects of the present invention relate to the processing of mass quote messages that include a plurality of orders. More specifically, aspects of the present invention provide systems and methods for processing mass quote messages and distributing market data utilizing a delimited message format.

BACKGROUND

[03] Current financial instrument trading systems allow traders to combine orders for a group of financial instruments into a single order called a mass quote message. For example, a trader may use a mass quote message when transmitting orders for a group of related option contracts, such as an option series belonging to one option class. With conventional trade engines, each individual order is processed separately and results in the creation of market data. The market data produced for each trade is then distributed to traders and other entities.

[04] As the number of orders and traders increases, the distribution of market data messages can strain computer systems and networks that are used to transmit such messages. In addition, the processing of numerous messages and associated overhead consumes bandwidth and processing time. Current messaging structures use fixed length, key-value paired, and/or message mockup structures to communicate content. These current messaging structures are inefficient due to overhead associated with padding extra bytes or use of keys or references which may be contained in the message. Therefore, there is a need in the art for improved systems and methods for processing mass quote messages
and producing and distributing market data messages using an improved electronic message structure.

SUMMARY OF THE INVENTION

[05] The present invention overcomes the problems and limitations of the prior art by providing systems and methods for processing mass quote messages and producing a single market data message. The orders within a mass quote message are processed and result in the production of market data. The market data is stored in one or more market data message buffers. The contents of the market data message buffer(s) is transmitted as a single market data message. In some embodiment of the invention, the mass quote message includes orders spread across different order books.

[06] The mass quote messages may be formatted using a messaging structure that uses delimiters to demarcate data and repeating groups of data. The delimiters may be single-byte control characters that allow data to be extracted using a parsing logic. A parsing algorithm may require that bytes in a message be read in order to locate the delimiters. The message structure may use templates to define message types, data meaning, and data ordering. The template may be disseminated prior to the sending of messages and used as an abstraction so that the meaning of data need not be conveyed in the message. The created messages may present data in the order stipulated by the templates. An application processing the messages may map data to a template and associate it with meaning. The template may be disseminated prior to the sending of messages and used as an abstraction so that the meaning of data need not be conveyed in the message. The use of templates may allow data to be added, deleted, or re-ordered by first distributing an updated template.

[07] In various embodiments, aspects of the present invention can be partially or wholly implemented on a computer-readable medium, for example, by storing computer-executable instructions or modules, or by utilizing computer-readable data structures. Of course, the methods and systems disclosed herein may also include other additional elements, steps, computer-executable instructions, or computer-readable data structures.
[08] The details of these and other embodiments of the present invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[09] The present invention may take physical form in certain parts and steps, embodiments of which will be described in detail in the following description and illustrated in the accompanying drawings that form a part hereof, wherein:

[10] Figure 1 shows a computer network system that may be used to implement aspects of the present invention;

[11] Figure 2 illustrates a system for processing mass quote messages and a distributing corresponding market data in accordance with an embodiment of the invention;

[12] Figure 3 illustrates a computer implemented method for processing mass quote messages in accordance with an embodiment of the invention;

[13] Figure 4 illustrates a computer implemented method of processing mass quote message that includes creating a market data message buffer, in accordance with an embodiment of the invention;

[14] Figure 5 shows an embodiment of a messaging structure in accordance with an embodiment of the invention;

[15] Figure 6 illustrates various delimiters utilized in templates and data messages in accordance with an embodiment of the invention;

[16] Figure 7 illustrates additional delimiters utilized in templates and data messages in accordance with an embodiment of the invention;

[17] Figure 8 illustrates a computer implemented method of creating messaging structure in accordance with an embodiment of the invention; and

[18] Figure 9 illustrates a computer implemented method of processing market data in accordance with an embodiment of the invention.
DETAILED DESCRIPTION

Exemplary Operating Environment

[19] Aspects of the present invention are preferably implemented with computer devices and computer networks that allow users to exchange trading information. An exemplary trading network environment for implementing trading systems and methods is shown in Figure 1. An exchange computer system 100 receives orders and transmits market data related to orders and trades to users. Exchange computer system 100 may be implemented with one or more mainframe, desktop or other computers. A user database 102 includes information identifying traders and other users of exchange computer system 100. Data may include user names and passwords. An account data module 104 may process account information that may be used during trades. A match engine module 106 is included to match bid and offer prices. Match engine module 106 may be implemented with software that executes one or more algorithms for matching bids and offers. A trade database 108 may be included to store information identifying trades and descriptions of trades. In particular, a trade database may store information identifying the time that a trade took place and the contract price. An order book module 110 may be included to compute or otherwise determine current bid and offer prices. A market data module 112 may be included to collect market data and prepare the data for transmission to users. A risk management module 134 may be included to compute and determine a user’s risk utilization in relation to the user’s defined risk thresholds. An order processing module 136 may be included to decompose delta based and bulk order types for processing by order book module 110 and match engine module 106.)

[20] The trading network environment shown in figure 1 includes computer devices 114, 116, 118, 120 and 122. Each computer device includes a central processor that controls the overall operation of the computer and a system bus that connects the central processor to one or more conventional components, such as a network card or modem. Each computer device may also include a variety of interface units and drives for reading and writing data or files. Depending on the type of computer device, a user can interact with the computer with a keyboard, pointing device, microphone, pen device or other input device.
[21] Computer device 114 is shown directly connected to exchange computer system 100. Exchange computer system 100 and computer device 114 may be connected via a T1 line, a common local area network (LAN) or other mechanism for connecting computer devices. Computer device 114 is shown connected to a radio 132. The user of radio 132 may be a trader or exchange employee. The radio user may transmit orders or other information to a user of computer device 114. The user of computer device 114 may then transmit the trade or other information to exchange computer system 100.

[22] Computer devices 116 and 118 are coupled to a LAN 124. LAN 124 may have one or more of the well-known LAN topologies and may use a variety of different protocols, such as Ethernet. Computers 116 and 118 may communicate with each other and other computers and devices connected to LAN 124. Computers and other devices may be connected to LAN 124 via twisted pair wires, coaxial cable, fiber optics or other media. Alternatively, a wireless personal digital assistant device (PDA) 122 may communicate with LAN 124 or the Internet 126 via radio waves. PDA 122 may also communicate with exchange computer system 100 via a conventional wireless hub 128. As used herein, a PDA includes mobile telephones and other wireless devices that communicate with a network via radio waves.

[23] Figure 1 also shows LAN 124 connected to the Internet 126. LAN 124 may include a router to connect LAN 124 to the Internet 126. Computer device 120 is shown connected directly to the Internet 126. The connection may be via a modem, DSL line, satellite dish or any other device for connecting a computer device to the Internet.

[24] One or more market makers 130 may maintain a market by providing constant bid and offer prices for a derivative or security to exchange computer system 100. Exchange computer system 100 may also exchange information with other trade engines, such as trade engine 138. One skilled in the art will appreciate that numerous additional computers and systems may be coupled to exchange computer system 100. Such computers and systems may include clearing, regulatory and fee systems.

[25] The operations of computer devices and systems shown in figure 1 may be controlled by computer-executable instructions stored on computer-readable medium. For example, computer device 116 may include computer-executable instructions for receiving order information from a user and transmitting that order information to exchange computer
system 100. In another example, computer device 118 may include computer-executable instructions for receiving market data from exchange computer system 100 and displaying that information to a user.

[26] Of course, numerous additional servers, computers, handheld devices, personal digital assistants, telephones and other devices may also be connected to exchange computer system 100. Moreover, one skilled in the art will appreciate that the topology shown in figure 1 is merely an example and that the components shown in figure 1 may be connected by numerous alternative topologies.

_mass Quote Message Processing and Market Data Dissemination_

[27] Figure 2 illustrates a system for processing mass quote messages and a distributing corresponding market data in accordance with an embodiment of the invention. A mass quote message 202 contains a plurality of orders for financial instruments. The financial instruments may be derivative products. Derivative products may include options on futures contracts, futures contracts that are functions of or related to other futures contracts, or other financial instruments that have their price related to or derived from an underlying product. In one embodiment of the invention, the orders are for options contracts that belong to a common option class. Orders may also be for baskets, quadrants, other combinations of financial instruments, etc. Mass quote message 202 may be received at an exchange. As used herein, an exchange is a place or system that receives and executes orders. The orders in mass quote message 202 are delivered to appropriate order book objects 204, 206 and 208. Each order book object may correspond to a particular financial instrument, such as an option contract and may track or maintain the order book for that financial instrument. The market data produced by each order may be delivered to a market data message buffer 210 where market data for each order in mass quote message 202 is compiled. When all of the market data has been received, the contents in market data message buffer 210 may be distributed to traders and others tracking market data. The transmission may be via the Internet 210 or other mechanisms connecting traders to an exchange.

[28] Figure 3 illustrates a computer implemented method for processing mass quote messages in accordance with an embodiment of the invention. First, in step 302 a mass quote message is received. The mass quote message includes a plurality of orders for financial
instruments and may be received at an exchange, match engine or any other location or device that processes orders for financial instruments. Next, in step 304 a thread iterates through the mass quote message and passes orders to corresponding order book objects. Step 304 may include sequentially reading all of the orders within the message and passing the orders to corresponding order book objects.

[29] Order book objects process orders and generate market data messages in step 306. The processing of orders and generation of market data messages may utilize conventional methods known to those of skill in the art. Each of the market data messages may be stored in a market data message buffer in step 308. The use of a market data message buffer facilitates compiling market data messages so that such messages can be distributed together in a single message. Finally, when market data has been received for all orders in the mass quote message, the contents in the market data message buffer is transmitted in step 310. Step 310 may include transmitting a market data message across one or computer, audio, video or data networks.

[30] There are a number of alternative methods for determining when a market data message buffer has received market data for all of the orders in a mass quote message. An input thread may determine how many orders are within a mass quote message and pass this number to the market data message buffer. Mass quote messages may also contain a fixed number of orders such that the market data message buffer only needs to count individual market messages. In another alternative embodiment, the last order within a mass quote message may include a flag or code that signifies to the market data message buffer or other module that the order is the last order. In yet another embodiment of the invention, order book objects may transmit a code to the market data message buffer when each one is done processing the orders that they have received. When the market data message buffer receives a code from each order book object, the buffer knows that it has received market data for each of the orders in the mass quote message. Each order book object may utilize its own thread for processing orders.

[31] Aspects of the invention may also use fixed, single and multiple market data message buffers. For example, each order book object may transmit market data messages to a unique market data message buffer. Figure 4 illustrates a computer implemented method of processing mass quote message that includes creating a market data message buffer, in
accordance with an embodiment of the invention. A mass quote message is received in step 402. Next, in step 404, a market data message buffer is created. Step 404 may include determining the required size of the buffer and selecting a buffer from a pool of available buffers or allocating memory for the buffer. In an alternative embodiment, step 404 may include creating a plurality of market data message buffers, such that each order book object that is used to process orders has access to a unique market data message buffer. Of course, two or more order book objects may also share a market data message buffer.

[32] Next, in step 406 the plurality of orders included within the mass quote message are processed by order book objects. Step 406 may include generating market data messages for each of the orders with the mass quote message. Each of the market data messages may be stored in a market data message buffer in step 408.

[33] In step 410 it is determined whether the market data message buffer contains market data for all of the plurality of orders. Step 410 may be performed by using one or more of the processes described above. When all market data has not been received, orders are the processing of orders continues in step 412. When all market data has been received, a single market data message containing the market data stored in the market data buffer is transmitted in step 414.

[34] In embodiments that use multiple market data message buffers, the contents of the buffers may be transmitted as a single market data message after all of the orders have been processed.

[35] The embodiments described above utilize a mass quote message. In alternative embodiments individual orders may be processed and produce market data that is compiled into a single market data message. A variety of criteria may be used to group orders that produce market data that is ultimately compiled into a single market data message. For example, orders received within a predetermined time period or orders for similar financial instruments received within a predetermined time period may be grouped together such that the resulting market data is transmitted within a single market data message.
[36] The embodiments described above also transmit a single market data message after all orders have been processed. In alternative embodiments of the invention a market data message may be transmitted before all of the orders are processed. For example, the message may be transmitted at a predetermined time after receiving market data from the first order. This allows market data to be distributed even when the processing of one or more orders takes an undue length of time. After a first market data message is transmitted, subsequent market data messages may be transmitted upon receipt of additional market data.

*Market Data Messaging Format*

[37] Aspects of the invention further comprise a messaging structure 500 (Figure 5) for communicating electronic data of any nature within the printable character set of any language. Meaning may be associated with actual message content without including any keys in the actual messages or requiring any kind of positional references to data in the messages. The approach supports flat message structures, as well as nested groups of repeatable data to any level of nested depth. A financial data message may comprise a market data message sent from an exchange and/or orders or messages delivered to an exchange.

[38] Figure 5 illustrates one embodiment of messaging structure 500. In Figure 5, a template 501 and a message 503 are illustrated. As shown in Figure 5, the message structure 500 may consist of delimiters to demarcate each attribute so that datum can be easily extracted. Templates such as template 501 may predefine an attribute order so that extracted datum can be associated with meaning. The use of delimiters and templates in messaging structure 500 may enable the messaging structure to be readily extensible.

[39] Character “|” 506 is used to denote a delimiter in an embodiment of the invention. Those skilled in the art will realize that many other additional characters may be utilized to represent a delimiter such as characters “[” 510 and “]” 512 and “}” 514. The delimiters may separate data elements within a given message structure. One or more templates such as template 501 may be defined and disseminated to 1) indicate the number and nature of supported message structures (flat or nested groups of repeatable data), as well as 2) the meaning of the data that may be communicated within a given message structure. The delimiters used may or may not be communicated in the templates, as
well. One or more message structures corresponding to each template may be defined and disseminated, carrying actual or meaningful content.

[40] Figures 6 and 7 illustrate various options for delimiters to be used in messaging structure 500 of the present invention. Those skilled in the art will realize that Figures 6 and 7 are not an exhaustive list of the various delimiters choices but are only an illustrative list. For example, Figures 6 and 7 may contain other options and may map to other options (not shown) using varying multiples of characters to represent the four illustrated delimiters. The use of multiple characters for delimiters may be less efficient but semantically the same as the options illustrated in Figures 6 and 7.

[41] Figures 6 and 7 include a unique delimiters column 602 which may indicate the number of unique characters used to build delimiters. Some of the various options may use multiple consecutive occurrences of one character to form a delimiter. The choice of delimiter characters may not change the format or the messaging structure 500. Considerations such as printability and character set may affect the actual characters and encoding chosen. For each alternative presented, a printable character may be chosen to represent each of the delimiters. Each table in both Figures 6 and 7 includes a “Field Delimiter” column 604, a “Group Start” column 606, a “Group End” column 608, a “Sample” column 610, and a “Notes” column 612. The “Sample” column 610 may include a normalized sample. The sample may be translated into its form for each alternative presented. In addition, Figure 7 also includes a “Group Delimiter” column 614, “Options Fields” column 616, and a “Repeating Group Column” 618.

[42] In many of the alternatives listed in Figures 6 and 7, the “Group End” 608 characters and the “Group Delimiter” 614 characters are the same. A field outside the repeating group in question specifies the number of occurrences of the group. This may allow a parsing algorithm to take advantage of the predictive nature of these tags because they may not be able to depend on the delimiters themselves to uniquely demarcate message structure. In some cases these fields may be necessary in order to parse the message.

[43] The “Options Fields” 616 shown in Figure 7 may be attached at the end of data messages. The “Options Fields” 616 may also be found at the end of repeating groups in data messages or at other places in a message and/or repeating groups in data messages.
The delimiters defined in Figures 6 and 7 may be used to demarcate attributes and repeating groups. The utilization of four delimiters is deemed optimal such as “Field Delimiter” 604, “Group Start” 606 delimiter, “Group Delimiter” 614, and “Group End” 608 delimiter. The use of four delimiters may be optimal because: 1) counts for the number of times a group repeats may not be needed thus saving bytes and simplifying the parsing algorithm; 2) delimiters at the end of repeating groups may be dropped if no data is present; 3) the parsing algorithm to extract datum may be generic for all messages; and 4) any message may be parsed without reference to message types.

Delimiters and templates may be disseminated or communicated by any means that allow them to be incorporated in an electronic system. The message structures then disseminated may be of variable length with data elements shortened or extended in length, as well as included or not present on a real-time per message basis. Message structures, depending on the choice of implementation, may be parsed without prior knowledge of the message content, without references or keys to the content, and/or without fixed positional reference to the message structures.

One or more templates such as template 501 may be defined and disseminated to 1) indicate the number and nature of supported messages (linear or nested groups), as well as 2) indicate the meaning of the data that may be communicated within a given type of message. Templates may allow datum to be associated with meaning by 1) defining the attributes, and 2) indicating the order in which they will appear. The use of delimiters within a given message type may also be communicated in its template.

Once the templates have been disseminated, messages corresponding to each template may be built and disseminated, carrying actual content. The messages may be of variable length with data elements shortened or extended in length, as well as included or not present on a real-time basis. Templates, and thus messages, may be changed on-the-fly so that attributes can be added, deleted, and/or re-order as needed. Template and message integrity may be checked per instance of receipt by validating message size.

In one embodiment of the invention, the messaging format detailed assumes the following: 1) messages are being passed from a sender to one or more receivers; 2) the method of dissemination is variable; 3) two fundamental types of messages are sent, templates and data messages; 4) the data being sent can be represented in key=value
pairs; 5) templates define the order of data in data messages; 6) templates contain keys
and data messages contain associated values; and 7) a protocol that uses this formatting
scheme will provide needed functionality as necessary (such as including a mechanism
by which to match a data message with a template or including a mechanism to verify
message contents).

[49] A message may consist, conceptually, of fields and repeating groups. Repeating groups
may be nested and also consist of fields and repeating groups. In templates, a repeating
group may only exist once. In messages, a repeating group may occur multiple times.
All these occurrences may be consecutive.

[50] As discussed above, messages may consist, structurally, of fields and delimiters. Every
field may be followed by one or more delimiters. Field delimiters may separate fields
within a group or in the message. If the last element of a message is not a repeating
group, it may be followed by a “Field Delimiter” 604; otherwise it may be followed by
the “Group End” 608 delimiter. “Group Start” 606 delimiters may mark the beginning of
a group definition in a template and a repeating group in a data message. “Group
Delimiters” 614 do not exist in templates and separate occurrences of a repeating group
in data messages. “Group End” 608 delimiters may mark the end of a repeating group
and may be placed after the last element of the last occurrence of a repeating group.
Every “Group Start” 606 delimiter may be matched by a “Group End” 608 delimiter later
in the message.

[51] Message structure 500 may perform several functions such as: 1) order attributes which
convey data; 2) it provides a means for extracting specific datum from the overall
message; and/or 3) provide a method for associating the datum with meaning.
Optimizing a message structure, therefore, involves ordering attributes in an efficient
manner while allowing datum to be extracted and associated with meaning in a likewise
efficient manner. An optimized message structure, moreover, may readily allow
attributes to be added, deleted or re-ordered, as well as efficiently handle attributes which
either may not be present or vary in length on a per message basis.

[52] A fully optimized message structure may communicate only meaningful content in a
format that expresses the data in the smallest possible size. A structure for stating price,
for example, may only state the price without overhead. The format for stating the price, moreover, might be optimized by expressing it as a binary rather than string statement.

[53] Message structure 500 orders attributes in a very efficient manner. Message Structure 500 may use delimiters to demarcate each attribute so datum can be readily extracted. Message structure 500 may use templates to pre-define attribute ordering so extracted datum can be associated with meaning.

[54] Message structure 500 may carry data within the printable character set of any language. Message structure 500 supports linear strings of data, as well as nested groups of repeatable data to any level of nested depth. Ordering of attributes may be determined by at least four factors: 1) attributes which are potentially repeatable to optimize efficiency are placed in repeating groups; 2) repeating groups are nested so that common data resides in the outer most group(s) and unique data resides in the inner most group(s); 3) data common to the entire message resides outside any repeating group; 4) attributes which may not appear often are placed at the end of a repeating group or at the end of the message. The last factor may allow delimiters for those attributes without values in a given instance to be dropped.

[55] Template and message integrity may be checked per instance of receipt by validating message size. Messages may be further validated against the structure of its corresponding template.

[56] The advantages of the delimited structure may include: 1) efficient message structure that produces message sizes comparable to or better than an optimized fixed length structure; 2) retains the flexibility of key-value and mockup structures for adding, re-ordering and extending the length of data elements contained in messages; 3) supports linear data strings, as well as nested groups of repeatable data to any level of nested depth. It excels at communicating data via complex nested groups which optimize message size efficiency; 4) attributes can be added, deleted, and/or re-order on-the-fly by defining and disseminating a template and then sending messages corresponding to the given template; 5) attributes can be shortened or extended in length, as well as included or not present on a real-time per message basis; 6) messages can be parsed without prior knowledge of the message type, as well as without references or keys to the content; 7) any number of templates and/or message types can be defined, and any nature of data
within the printable character set of any language can be communicated; and 8) it is easier to optimize both up-front and over time than a fixed length structure.

[57] A further advantage of the delimited structure may include a variable length message structure that consumes no more bytes than: 1) for linear data strings, the number of bytes used for actual data plus one byte per delimiter in a given message; or 2) for nested message structures, the same number of bytes as above plus potentially X number of bytes to close each nested group of data.

[58] Weaknesses of fixed length message structures include the use of padding to accommodate attributes which either may not be present or vary in length on a per message basis. Each attribute must be padded to a fixed length equal to the longest possible value the attribute might convey. This is needed to maintain a consistent overall fixed length structure (or consistent fixed length partitions for repeating groups) so that datum can be extracted using a pre-defined set of positional references. The fixed length structure, therefore, also suffers from not being readily extensible.

[59] Thus, a fixed length structure would optimally state instrument, price and quantity as:

[60] ESZ31075005

[61] and no structure could better optimize the message. It is improbable, however, that a fixed length structure can be as optimized as in the example given above. Because the attributes vary in length on a per message basis, they will have to be padded:

[62] ESZ3-----001075000005.

[63] Thus, the fixed length message size is inflated by a total of 11 unneeded characters in this instance.

[64] If the delimited structure described above is used instead:

[65] ESZ3|107500|5|

[66] ESV3C1060|1040|5|

[67] This structure incurs an overhead of 3 control characters for delimiters and improves upon the fixed length example where padding is added.
From the perspective of optimizing message size, however, the delimited structure may be ultimately deemed equal to the use of an optimized fixed length structure when the structures do not include repeating groups of data. The constraints may include: 1) the more attributes with values that will vary in length as well as the more attributes that will not be present in every instance, the more efficient the delimited structure will be; and 2) the more attributes with values of constant length and always present the more efficient the fixed length structure will be.

The following examples compare messages which communicate date and time, trading system, trading mode, instrument, price, quantity, price variation, order type, and origin:

20031015153015999GOESZ3-----00107500000005+L4 pads with 11 characters

20031015153015999|G|O|ESZ3|107500|5|+|L|4| incurs 9 delimiters

When linear strings are thus compared, the two approaches are generally equal with respect to message size. The delimited structure, however, will perform better than the fixed length structure when repeating groups are introduced. Also, the fixed length structure must always be optimized – attributes efficiently ordered and padding kept to a minimum – if it is to perform well.

Figure 8 illustrates a computer a messaging structure in accordance with an embodiment of the invention. First, in step 802 a template is created that defines the order of the plurality of fields. Next, in step 804 the template may be transmitted though a network as illustrated in Figure 1. The market data to be transmitted may be compiled in step 805. The market data may include a plurality of financial information such as orders, quotes or mass quotes, trades, and statistics. The financial information may include derivate products. Derivative products may include options on futures contracts, futures contracts, future contracts that are functions of or related to other futures contracts, or other financial instruments that have their price related to or derived from an underlying product.

A financial data message may be created in step 806. The financial data message may comprise a market data message sent from an exchange and/or orders or messages delivered to an exchange. The financial data message includes a plurality of fields that have been separated by delimiters. The delimiters may include at least one delimiter that
is used to identify a repeated group of information. Finally, in step 808 the financial data message may be transmitted from an exchange or similar system. The transmission may be across one or more computer, audio, video or data networks.

Figure 9 illustrates a computer implemented method of processing a market data message having a plurality of fields separated by delimiters in accordance with an aspect of the invention. In step 902, a template may be received. The template may define the order of the plurality of fields of the market data message. Next, in step 904 a market data message may also be received. The market data message may include a plurality of fields separated by delimiters. The source of the market data message may be the same as the source of the template. In step 906, the market data message may be parsed to extract the fields separated by the delimiters. The data extracted from the market data message may include market data containing a plurality of orders for financial instruments. Finally, in step 608 the data in the market data message may be interpreted based information such as order information included in the template.

The embodiments described above utilize market data. In alternative embodiments individual orders and quotes may use the same message structure. The messaging structure 500 of the present invention may be used in the processing of market data. The market data may contain a plurality of orders for financial instruments. The financial instruments may be derivative products. Derivative products may include options on futures contracts, futures contracts that are functions of or related to other futures contracts, or other financial instruments that have their price related to or derived from an underlying product. These market data may be received at an exchange that receives and executes orders.

The present invention has been described in terms of preferred and exemplary embodiments thereof. Numerous other embodiments, modifications and variations within the scope and spirit of the invention will occur to persons of ordinary skill in the art from a review of this disclosure. For example, aspects of the invention may be used to process and communicate data other than market data.
CLAIMS

We claim:

1. A computer-readable medium containing computer-executable instructions for performing the steps comprising:

   (a) receiving a mass quote message containing a plurality of orders for financial instruments;

   (b) parsing the mass quotes message and transmitting the plurality of orders to order book objects;

   (c) processing the plurality of orders at the order book objects;

   (d) generating market data for each of the plurality of orders;

   (e) storing the market data in at least one market data message buffer; and

   (f) transmitting a single market data message containing the market data stored in the at least one market data message buffer.

2. The computer-readable medium of Claim 1, wherein the plurality of orders comprises orders for option contracts belonging to an option class.

3. The computer-readable medium of Claim 2, wherein the order book objects correspond to option contracts within an option class.

4. The computer-readable medium of Claim 1, wherein the plurality of orders comprise a basket.

5. A computer implemented method of processing orders for financial instruments, the method comprising:

   (a) receiving a mass quote message containing orders for a plurality of financial instruments;

   (b) selecting an order;

   (c) processing the order selected in (b);
(d) storing in a buffer market data resulting from (c);

(e) repeating (b)-(d) for remaining ones of the plurality of orders; and

(f) transmitting a single market data message containing the market data stored in the buffer.

6. The method of Claim 5, wherein (b) comprises selecting the first order in the plurality of derivative product orders that has not been processed.

7. The method of Claim 5, wherein (f) comprises transmitting the single market data message from an exchange to a plurality of market participants.

8. The method of Claim 5, wherein the plurality of financial instruments comprise option contracts having a plurality of strike prices.

9. The method of Claim 8, wherein the plurality of financial instruments comprise call and put contracts.

10. The method of Claim 5, wherein the plurality of financial instruments comprise a basket.

11. A computer implemented method of processing orders for financial instruments, the method comprising:

(a) receiving a mass quote message containing a plurality of orders for a plurality of financial instruments;

(b) creating a market data message buffer;

(c) processing the plurality of orders;

(d) storing in the market data message buffer market data resulting from (c);

(e) determining when the market data message buffer contains market data for all of the plurality of orders; and

(f) when the market data message buffer contains market data for all of the plurality of orders, transmitting a single market data message containing the market data stored in the market data message buffer.
12. The method of Claim 11, wherein (e) comprises counting instances of market data received at the market data message buffer.

13. The method of Claim 11, wherein (e) determining when market data is received that corresponds to the last of the plurality of orders.

14. The method of Claim 13, wherein (e) comprises identifying a code that is attached to the last order in the plurality of orders.

15. The method of Claim 11, wherein (f) comprises transmitting the single market data message from an exchange to a plurality of market participants.

16. The method of Claim 11, wherein the plurality of financial instruments comprise option contracts having a plurality of strike prices.

17. The method of Claim 11, wherein the plurality of financial instruments comprise call and put contracts.

18. The method of Claim 11, wherein (c) comprises processing the orders with a plurality of threads.

19. The method of Claim 11, wherein the plurality of financial instruments comprise a basket.

20. A computer implemented method of processing orders for financial instruments, the method comprising:

   (a) receiving a mass quote message containing a plurality of orders for a plurality of financial instruments, wherein the orders are arranged sequentially from a first order to a last order;

   (b) processing the plurality of orders;

   (c) storing in a market data message buffer market data resulting from (b);

   (d) determining when the market data message buffer contains market data corresponding to the last order; and
(e) when the market data message buffer contains market data corresponding to the last order, transmitting a single market data message containing the market data stored in the market data message buffer.

21. The method of Claim 20, further including:

(f) after the single market data message is transmitted, transmitting market data messages as additional market data is received.

22. The method of Claim 20, wherein (d) comprises identifying a code that is attached to the last order.

23. The method of Claim 20, wherein (e) comprises transmitting the single market data message from an exchange to a plurality of market participants.

24. The method of Claim 20, wherein the plurality of financial instruments comprise option contracts having a plurality of strike prices.

25. The method of Claim 20, wherein the plurality of financial instruments comprise call and put contracts.

26. The method of Claim 20, wherein (b) comprises processing the orders with a plurality of threads.

27. The method of Claim 20, wherein the plurality of financial instruments comprise a basket.

28. A method of processing a market data message sent from an exchange, the method comprising:

(a) receiving a template, the template defining a plurality of fields;

(b) receiving the market data message having a plurality of fields separated by delimiters, the template received from the same source as the market data message;

(c) parsing the market data message to extract the fields separated by the delimiters; and

(d) interpreting the plurality of fields based on their order and information included in the template.
29. The method of processing a market data message sent from an exchange of Claim 28, wherein the market data message comprises market data.

30. The method of processing a market data message sent from an exchange of Claim 28 further including the step of:

(e) storing the parsed fields from the market data message.

31. The method of processing a market data message sent from an exchange of Claim 29, wherein the market data comprises a plurality of orders for financial instruments.

32. The method of processing a market data message sent from an exchange of Claim 29, wherein the market data comprises quotes and mass quotes.

33. The method of processing a market data message sent from an exchange of Claim 29, wherein the market data comprises trades.

34. The method of processing a market data message sent from an exchange of Claim 29, wherein the market data comprises statistics.

35. The method of processing a market data message sent from an exchange of Claim 31, wherein the orders are selected from the group consisting of securities, fixed income, interest rates, agricultural, and industrial commodities.

36. A computer implemented method of processing market data messages having a plurality of fields separated by delimiters, the method comprising:

(a) receiving a template;

(b) receiving a market data message having fields separated by delimiters;

(c) processing the market data message, the processing including parsing the market data message;

(d) interpreting the plurality of fields based on their order and information provided in the template; and

(e) storing the market data parsed from the market data message.

37. A method of creating a messaging structure, the method comprising:
(a) creating a template, the template defining an order of a plurality of fields;

(b) transmitting the template;

(c) compiling market data;

(d) creating a financial data message; and

(e) transmitting the financial data message.

38. The method of Claim 37, wherein the financial data message comprises a market data message.
Figure 2
302. Receive a mass quote message

304. Iterate through the mass quote message and pass orders to corresponding order book objects

306. Process orders at order book objects and generate market data messages

308. Store market data messages in a market data message buffer

310. When market data has been received for all orders in the mass quote message, transmit the contents in the market data message buffer

Figure 3
402. Receive a mass quote message containing a plurality of orders for a plurality of derivative products

404. Create a market data message buffer

408. Process the plurality of orders

408. Store market data in the market data message buffer

410. Does the market data message buffer contain market data for all of the plurality of orders

412. Continue processing orders

414. Transmit a single market data message containing the market data stored in the market data message buffer

Figure 4
<table>
<thead>
<tr>
<th>Unique Delimiters</th>
<th>Field Delimiter</th>
<th>Group Start</th>
<th>Group End</th>
<th>Sample</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>A</td>
</tr>
</tbody>
</table>

Figure 6
<table>
<thead>
<tr>
<th>Unique Delimiters</th>
<th>Field Delimiter</th>
<th>Group Start</th>
<th>Group Delimiter</th>
<th>Group End</th>
<th>Sample</th>
<th>Optional Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>602</td>
<td>604</td>
<td>606</td>
<td>614</td>
<td>608</td>
<td>610</td>
<td>Fields are optional at the end of a message. Optional fields in the middle of a message can be indicated by two consecutive field delimiters or a protocol specific null value indicator.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repeating Group Occurrence Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req’d = required Rec’d = recommended</td>
</tr>
</tbody>
</table>

| Notes | Will need template in order to parse if repeating groups exist in the message. |

| 1      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|        |   |   |   |   | Fields are optional at the end of a message. Optional fields in the middle of a message can be indicated by two consecutive field delimiters or a protocol specific null value indicator. |

| Req’d | Will need template in order to parse if repeating groups exist in the message. |

| 2      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| 2      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|        |   |   |   |   | Fields are optional at the end of a message. Optional fields in the middle of a message can be indicated by two consecutive field delimiters or a protocol specific null value indicator. |

| Req’d | Will need template in order to parse if repeating groups exist in the message. |

| 2      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| 3      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|        |   |   |   |   | Fields are optional at the end of a message. Optional fields in the middle of a message can be indicated by two consecutive field delimiters or a protocol specific null value indicator. |

| Req’d | Will need template in order to parse if repeating groups exist in the message. |

| 4      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| 4      | 1 | 1 | 1 | 1 | A|B|C|D|E|F|G|H|I|J |
|        |   |   |   |   | Fields are optional at the end of a message. Optional fields in the middle of a message can be indicated by two consecutive field delimiters or a protocol specific null value indicator. |

| Notes | There are many variations of this. All that changes is which character is used as a multiple and to indicate which delimiter. See the template options table for a similar example. |

**Figure 7**
Create a template defining the order of the plurality of fields

Transmit template

Compile market data

Create a financial data message including a plurality of fields

Transmit the financial data from an exchange

Figure 8
902
Receive a template defining the order of a plurality of fields

904
Receive a market data message having a plurality of fields

906
Parse the market data message

908
Process the data contained in the market data message

Figure 9