An algorithmic trading system in which the trading algorithms 3, the operation of which is variable based on values input by a user of various trading algorithm parameters 2, 9, are encoded and stored within a file structure is described. A method of updating the trading algorithms and trading algorithm parameters is also described.

Figure 1
Releases [02487]: Buy 10,000 BAE Systems; Unreleased: 10,000

<table>
<thead>
<tr>
<th>Counterparty</th>
<th>System</th>
<th>Quantity</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston Shaw</td>
<td>FIX 4.2</td>
<td>10,000</td>
<td>Vwap</td>
</tr>
</tbody>
</table>

Standard algorithms:
Float, Tex, Vwap

Custom algorithms:
MyVwap1, MyVwap2

Figure 1
Algorithmic Trading

Stanley Shaw

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>09:30</td>
</tr>
<tr>
<td>End Time</td>
<td>16:00</td>
</tr>
<tr>
<td>Max Volume</td>
<td>50</td>
</tr>
</tbody>
</table>

Save As  OK  Cancel

Figure 2

Algorithm: Vwap
Start Time: 9:00
End Time: 18:00
MaxPctVolume: 50

Figure 3
<table>
<thead>
<tr>
<th>Counterparty</th>
<th>System</th>
<th>Quantity</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABN Amro</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bankers Trust</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bear Stearns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bell</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carnegie</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chase Manhattan – Can sft</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE Shaw</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit Suisse First Boston</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston Shaw</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tex</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vwap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyVwap1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyVwap2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Morgan Stanley – Can sft</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4
Figure 5

Figure 6
<table>
<thead>
<tr>
<th>Name</th>
<th>Order</th>
<th>Name</th>
<th>Status</th>
<th>Original Qty</th>
<th>Counterparty</th>
<th>System</th>
<th>Wave %</th>
<th>V...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABN Amro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bankers Trust</td>
<td>O2183</td>
<td>Reut...</td>
<td>Prop...</td>
<td>5,000</td>
<td>Manhattan</td>
<td>Teleph</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Bear Stearns</td>
<td>O2864</td>
<td>Banc...</td>
<td>Prop...</td>
<td>3,000</td>
<td>Manhattan</td>
<td>Teleph</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Bell</td>
<td>O2838</td>
<td>Barn...</td>
<td>Prop...</td>
<td>5,000</td>
<td>Manhattan</td>
<td>Teleph</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>Carnegie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chase Manhattan – Can sft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE Shaw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit Suisse First Boston</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boston Shaw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vwap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyVwap1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyVwap2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Morgan Stanley – Can sft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7**
Your algorithm selection is invalid for the following order numbers:
O2183, O2836
Please make a different selection or reduce your List

Figure 8
### Orders

<table>
<thead>
<tr>
<th>Res...</th>
<th>Order</th>
<th>Mgr</th>
<th>Dlr</th>
<th>Orig...</th>
<th>Mkt</th>
<th>B/S 2</th>
<th>B/S</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O140</td>
<td>JG</td>
<td></td>
<td>10,000</td>
<td>DE</td>
<td>B</td>
<td></td>
<td>Bayerische Motoren Wer...</td>
</tr>
<tr>
<td></td>
<td>O160</td>
<td>JG</td>
<td></td>
<td>10,000</td>
<td>GB</td>
<td>B</td>
<td></td>
<td>Barclays</td>
</tr>
<tr>
<td></td>
<td>O534</td>
<td>JG</td>
<td></td>
<td>10,000</td>
<td>US</td>
<td>B</td>
<td></td>
<td>Bank of New York</td>
</tr>
<tr>
<td></td>
<td>O1051</td>
<td>JG</td>
<td></td>
<td>10,000</td>
<td>SG</td>
<td>B</td>
<td></td>
<td>Singapore Airlines</td>
</tr>
<tr>
<td></td>
<td>O1084</td>
<td>JG</td>
<td></td>
<td>20,000</td>
<td>ES</td>
<td>B</td>
<td></td>
<td>Telefonica</td>
</tr>
</tbody>
</table>

### Releases

<table>
<thead>
<tr>
<th>C</th>
<th>Limit</th>
<th>TIF</th>
<th>TIF Date</th>
<th>Cpty</th>
<th>Quantity</th>
<th>Executed Qty</th>
<th>Executed Val</th>
<th>Avg Price</th>
<th>H...</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>W</td>
<td>07 May</td>
<td>PS</td>
<td>2,000</td>
<td>2,000</td>
<td>9,140</td>
<td>4.57</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>W</td>
<td>07 May</td>
<td>CMB</td>
<td>3,000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9
ALGORITHMIC TRADING

This invention relates to automated execution algorithms, in particular those used by buy-side trading desks in financial trading situations.

Increasingly, brokers are offering automated execution algorithms ("Algorithmic Trading") to their clients' buy-side trading desks. These algorithms may use large amounts of historical data to aid the prediction of trading patterns, and complex real-time data to assess current trading conditions. For example, stock price correlations between many thousands of stocks need to be re-calculated on a regular basis. The trading algorithms are intended to deliver price benefits and trading efficiencies, combined with different statistical analysis techniques and complex scheduling patterns, to avoid other market participants identifying trading patterns.

There are many possible approaches to algorithmic trading, for example using a "Benchmark" algorithm such as VWAP (Volume Weighted Average Price) (available from CSFB). Each counterparty with which a broker trades will offer and support a different set of algorithms. When requesting an algorithmic trade, buy-side traders have the option of controlling various parameters, such as participation rates, time and pacing controls, start and end times, aggression settings, price thresholds and limits and others.

Algorithmic trading instructions are often passed from the buy-side trader to the counterparty via FIX (Financial Information eXchange) and the various algorithm parameters are defined in the FIX custom tags. FIX contains a delimited string of data that is a dedicated communication between brokers (or other sell-side personnel) and buy-side personnel, and is a messaging protocol for the exchange of financial information. FIX comprises a Session layer and an Application layer. The Session layer handles the actual connection between two counterparties, typically an Asset Manager (known as Buy Side) and Investment Bank (known as Sell Side). The Application layer deals with orders (for example BUY 100 IBM @ 105.4), executions (used to signal status or a fill, for example, partial fill 50 IBM @ 105.4), and allocation (once the trade is complete, an
allocation to an account). The Application layer also encompasses amendments to orders, corrections to fills, cancellations, etc.

Versions 4.0 and 4.2 of FIX are a tag=value message format, for example, 
10=5|35=A|22=2|55=IBM.L|102=105.4, with the message structure having a header, body and tail.

Known software applications dealing with algorithmic trading have the FIX destinations for each counterparty/algorithm/parameter combination that a user wishes to use hard coded into the software itself. However, these algorithms are not supported at the trader front end of the system. The main disadvantage of this is that if an additional algorithm is introduced by a counterparty, or required by a user, the entire software system must be updated as there is no means by which the additional algorithms and FIX destinations can be added to existing software. This is time consuming and costly, and results in there being a period of time where required algorithms cannot be used.

There is therefore a need to produce software which enables a trader to indicate for a release, or set of releases, that they wish to trade with a selected counterparty using a specific algorithm, and which can be updated when the algorithms issued by a particular counterparty are changed, or when a new counterparty that the trader wishes to trade with issues an algorithm or set of algorithms. The present invention aims to address that need.

The present invention provides an algorithmic trading system comprising a file structure for storing encoded trading algorithms selectable by a user, each encoded trading algorithm having at least one encoded trading algorithm parameter associated therewith to vary the operation of the algorithm on input of algorithm parameter values, input means to allow a user to select an encoded trading algorithm from the file structure, to input trading information and to enter algorithm parameter values into the system, for the selected encoded trading algorithm, and communication means to allow the trading information and algorithm parameter values to be communicated to a third party in accordance with the selected encoded trading algorithm.
The invention also provides a method of updating the algorithmic data available to an algorithmic trading system, comprising the steps of encoding the algorithmic data, storing the encoded algorithmic data in a file structure, retrieving the algorithmic data from the file structure and uploading it into the algorithmic trading system.

The invention further provides an algorithmic trading method comprising the steps of encoding algorithmic data, selecting an encoded trading algorithm via a user input device, the operation of the encoded trading algorithm being variable based on algorithm parameter values input to at least one algorithm parameter associated with the encoded trading algorithm, storing the algorithmic data in a file structure, selecting an encoded trading algorithm via a user input device accessing the encoded trading algorithm from the user input device, inputting algorithm parameters values for the selected encoded trading algorithm and inputting trading information, and communicating the trading information and algorithm parameter values, in accordance with the algorithm parameters, to a third party.

The invention also provides a computer program product, for updating the algorithmic data available to an algorithmic trading system, wherein the algorithmic data comprises at least one trading algorithm associated with at least one trading algorithm parameter, the operation of the algorithm variable by algorithm parameter values input by a user, having program code stored thereon which when executed on a computer causes the computer to perform the steps of encoding the algorithmic data, storing the encoded algorithmic data in a file structure, retrieving the algorithmic data from the file structure and uploading it into the algorithmic trading system.

The invention yet further provides a computer program product for an algorithmic trading system, having program code stored thereon which when executed on a computer causes the computer to perform the steps of encoding algorithmic data, storing the algorithmic data in a file structure, selecting an encoded trading algorithm via a user input device, accessing the encoded trading algorithm and at least one associated encoded algorithm parameter, the operation of the algorithm variable by algorithm parameter values input by a user, from the user input
device, inputting algorithm parameter values for the selected encoded trading
algorithm and inputting trading information, and communicating the trading
information and algorithm parameter values, in accordance with the algorithm
parameters, to a third party.

The invention yet further provides a system for arranging algorithms to create an
algorithmic trading system, the system comprising input means for receiving
trading algorithms, the operation of each trading algorithm variable by one or
more trading algorithm parameters, input means for receiving one or more trading
algorithm parameters, and a file structure, arranged to store the trading
algorithms and trading algorithm parameters in a manner accessible to a user to
create an algorithmic trading system.

Preferably, the algorithmic data comprises at least one trading algorithm and/or at
least one trading algorithm parameter.

Preferably, the algorithmic data is encoded in XML.

Preferably, the file structure is a database.

The invention gives the advantages that encoding the algorithmic data and
storing it in a file structure enables the algorithmic data to be updated, for
example, by the addition or removal of trading algorithms or trading algorithm
parameters, without the need to alter the software or structure of the algorithmic
trading system.

In addition, the encoding and storing of the algorithm data enables the user to
name, store and retrieve, under a user-defined file name, frequently used
algorithm parameters or algorithm parameter values.

Embodiments of the invention will now be described by way of example only, and
with reference to the accompanying drawings in which:

Figure 1 is a screen shot of a release screen;
Figure 2 is an example of an algorithm trading window;
Figure 3 is an illustration of a tool tip used in connection with the screen of Figure 1;
Figure 4 shows a custom algorithm maintenance menu;
Figure 5 illustrates saving an algorithm with a public/private visibility;
Figure 6 shows a delete prompt;
Figure 7 is a screen show of a multi release edit window;
Figure 8 shows a warning prompt for invalid algorithm selection;
Figure 9 is a screenshot of a trade screen;
Figure 10 is a screen shot of an algorithm permission tree; and
Figure 11 is a screen show of a custom algorithm permissions tree.

As discussed above, brokers and banks are constantly introducing new trading algorithms with different parameters. In order to accommodate such changes, embodiments of the present invention provide a dynamic software system for algorithmic trading, in contrast to the static, hard-coded systems currently available. This is achieved by implementing the trading algorithms in a generic manner in three stages:

1. For each trading algorithm, the parameters available for that trading algorithm, the associated trading algorithm parameters and the format is defined in an XML (eXtensible Markup Language) schema, and stored in a database. The XML syntax describing the trading algorithms also includes the graphical user interface (GUI) control to be used for each parameter associated with the trading algorithm and their available values.

2. The screen display used by the trader to choose the trading algorithm and set the parameters is soft coded to read the XML schema and display the appropriate trading algorithms and parameters dependent upon the selected counterparty.

3. The FIX custom tags used to set the various parameters for each counterparty and trading algorithm are configured in the data transformation XML that is used by adapter software that allows the FIX data to be transmitted.
FIX data may be written using FIXML, an XML version of FIX. The tag=value
structure of the original FIX format is retained, but the use of XML enables a
more descriptive structure:

```xml
<fixmessage>
  <header><fixversion>4.2</fixversion></header>
  <body><qty>100</qty><side>B</side><symbol>IBM.1</symbol></body>
</fixmessage>
```
which represents trading of IBM data as on page 1.

The software system may be part of a buy-side trading system, such as that
The adapter software may be the LatentZero FIX Out Adapter, also provided
by LatentZero Limited.

The advantage of coding the trading algorithms into an XML schema, stored
in a database and accessed by the trading system, is that only the XML files
in the database and the FIX custom tags need to be updated to support new
trading algorithms. Although the invention is described in terms of accessing
algorithmic data (trading algorithms and algorithm parameters) from a
database, it is also possible to store the data in any suitable file structure
format. Such a file structure format may be part of the system, remote from
the system or networked to the system.

Various features of the invention necessary to support algorithmic trading will
now be described.

Releasing Algorithmic Trades

When releasing orders to be executed as algorithmic trades, the trading
algorithms available for trading are determined by the counterparty. Each
counterparty supports a number of trading algorithms and requires a different
set of parameters. When making a trading algorithm release, users must
select the counterparty, trading algorithm and the parameters associated with
the selected trading algorithm. Custom trading algorithms enable traders to
store frequently used pre-defined parameter values under custom trading algorithm names. Each trading algorithm will have at least one algorithm parameter associated with it. In order to use the trading algorithm, a user will need to insert algorithm parameter values into the various prompts. Of these parameter values, some will be required (the trade cannot take place without the information) and some will be optional. If the optional values are not provided by the user, the system uses a pre-saved set of default values. Other trading information must also be provided by the user, such as quantities.

The release screen (the screen display seen by a user which details the sale being made and the options available) is shown in Figure 1. Figure 1, in common with Figures 2, 4, 7, 9, 10 includes generic representations of dealer/broker names.

A counterparty list control 1 includes both standard trading algorithms and custom trading algorithms for each counterparty. Custom trading algorithms are algorithm parameter combinations saved by the user under a specific name for future reuse. Custom trading algorithms are described in more detail below. The counterparty list control 1 in Figure 1 shows that the counterparty "Boston Shaw" offers three types of trading algorithms: "Float", "Tex" and "VWAP", and two custom trading algorithms: "My VWAP1" and My VWAP2". The following items are shown in a tree structure under each counterparty:

standard trading algorithms supported by the counterparty, and nested under each trading algorithm:

custom trading algorithms defined as private by the current user;
custom trading algorithms defined as public.

Only trading algorithms defined as valid for the base type and market of the selected order are displayed. These restrictions are specified in the XML trading algorithm definition.

The counterparty control list 1 shows the trading algorithms in alphabetical order. The trading algorithms supported by each counterparty are saved in a table "trading_algorithm". Custom trading algorithms available to each user
are stored in a table "trading_algorithm_custom". Both of these tables are described in more detail below.

Figure 1 also shows a release strip 2. If the counterparty selected (via the counterparty list control or using a counterparty walker (not shown)) supports algorithmic trading, the trading algorithm is inserted into the strip in trading algorithm column 3.

If a trading algorithm node was selected in the counterparty list control 1, the trading algorithm is shown in the column 3. The counterparty column 4 will be populated with the counterparty name and the system column 5 will be prepopulated with the value corresponding to the FIX version used by the counterparty (for example, FIX4.2). Users are able to change this value from the list of valid systems for the selected counterparty.

If the counterparty node was selected (in the counterparty list control 1 or counterparty walker (not shown)) then the trading algorithm column 3 is blank. A drop-down combination box lists the trading algorithms for that counterparty as defined for the counterparty list control and in the same order. The combination box also includes a "blank" option to allow a non-trading algorithm release.

If a subsequent release is made to another counterparty that does not support algorithmic trading, the trading algorithm column 3 will still be displayed with the text <disabled>.

In subsequent releases, until a counterparty is selected, the trading algorithm column 3 will be set to blank, with no other option in its combination box.

When a standard trading algorithm is selected in the counterparty list control 1 by the user, default values (already stored or entered into the system) are assigned to the trading algorithm parameters. These default parameters are part of the XML trading algorithm description, stored in the database. If the user selects a custom trading algorithm from the counterparty list control 1, the trading algorithm name appears in the trading algorithm column 3 in the
release strip 2, and the parameter values are set to those stored in connection with that particular custom trading algorithm.

The parameters of either standard or custom trading algorithms are editable in the same manner. A clickable button 6, shown as "a triangle" in trading algorithm column 3 in Figure 1 invokes a trading algorithm parameters control. This is shown in more detail in Figure 2.

When the user hovers the cursor over the trading algorithm field for a particular chosen trading algorithm release, a tool tip 7 is visible, as shown in Figure 3. This tool tip 7 shows the parameter names and values associated with that trading algorithm, based on the data entered into the trading algorithm parameters control in Figure 2. The tool tip is represented by a coloured triangle visible in the upper right corner of trading algorithm column 3.

The selected trading algorithm may support different order types, (for example, Limit orders), which are defined in XML for each trading algorithm, and stored in the database. If this is the case, the supported order types available are shown in the condition field 8 in the release strip 2. If the trading algorithm does not support different order types, then the condition field 8, limit 9 and stop 10 fields are blank and non-editable.

It is also possible for the user to select a trading algorithm where there are values already set at the condition and limit fields. If the trading algorithm supports the condition already set, then the condition and limit fields are left unchanged. If the trading algorithm supports different condition types, but does not support the condition currently set, the condition and limit fields are cleared down and are editable (such that a supported condition can be selected). If the trading algorithm selected does not support different order types, the condition and limit fields are cleared down and are non-editable.

Once an algorithmic release is created on the system, users are able to change the value of the trading algorithm from a combination box control. Valid trading algorithm control values in the combination box control follow the
same rules as described in the counterparty list control 1. For example, if the entry in the "System" box is FIX, users can only modify the trading algorithm or its parameters if the selected counterparty supports Cancel/Replacement messages.

This functionality is indicated in the field "counterparty_params.order_replace_supported". If the counterparty does not support these message types, the parameters are read-only fields.

Algorithmic Parameters Control.

Figure 2 shows an example screen shot of the algorithmic parameter control window 11. The example shown is for a VWAP trading algorithm.

The title region 12 shows the counterparty name followed by the trading algorithm name. The trading algorithm name is in a drop down combination box that contains the names of all of the trading algorithms supported by the displayed counterparty. If the trading algorithm selected in the combination box is changed by the user, the input parameters and default values displayed in the window will change according to the trading algorithm selected by the user.

The window 11 displays two columns: a parameter name column 13 and a value column 14. The parameter column 13 displays the parameter names relevant for the trading algorithm selected, which are part of the XML definition. The value column 14 shows default values as defined for each trading algorithm. In the case of custom trading algorithms the values column 14 shows the saved parameter values. Users have "save", "save as" and "cancel" options on the parameter input window 11:

- selecting "save" causes the parameters input into the window to be accepted. Once the button is pressed, the input process is complete and the window 11 is closed.
- Selecting "save as": allows users to save the parameter values entered under a different trading algorithm name.
Selecting "cancel" closes the window once the button is pressed. The process resumes its previous state, as the user has interrupted this particular release.

Custom Trading Algorithms

Figure 4 shows a custom trading algorithm maintenance menu 15. This menu may be invoked from the counterparty list control 1 using the right-click control on an existing trading algorithm on the list. A custom trading algorithm has associated with it the options save as, delete and edit, whereas only a save as option is available with standard trading algorithms. A custom trading algorithm comprises a set of algorithm parameters for a particular trading algorithm, saved under a user-defined name in the database, and retrievable from the database.

When users select the save as option, the "Algorithm Trading" window 11, which allows a user to update trading algorithm details within the window is displayed. The user is then prompted to enter the name of a trading algorithm. If the user has permission to create public trading algorithms (those available to any other user) then the "Public" box is clickable to mark the trading algorithm public as shown in Figure 5. If this permission is denied or the "Public" box is not ticked, the window shows the text "Private". The trading algorithm is saved in a table "trading_algorithm_custom".

If a user types a trading algorithm name that already exists under his user ID or under the public domain, the system will inform the user that a trading algorithm with that same name already exists. The system simultaneously prompts the user to enter a different name or cancel an action. A duplicate name situation will arise in either of the following case:

If public status = "private" and an entry with the same trading algorithm name already exists in the table "trading_algorithm_custom" for the selected counterparty and current user, or the name already exists as "public" of the same counterparty; if the current user selects the privacy status = "public"
then an entry with the same already exist in the table "trading_algorithm_custom" will select a counterparty under any user ID.

When users select the option "delete", the custom trading algorithm from which the maintenance menu was invoked will be deleted. This system will prompt the user for conformational cancellation of the delete action. An example of a window used for this prompt is shown in Figure 6. If the user confirms the action, the entry corresponding to the user ID, counterparty in trading algorithm will be deleted from the "trading_algorithm_custom" table.

When a user selects the option "edit", the "algorithm trading" window will be presented with pre-populated parameter values as to find and accustomed trading algorithm selected. The options available are those of the "algorithm trading" window, as illustrated in figure 5, except the "save as" option. An "OK" option result from the saving of the trading algorithm details.

Multi-release Editor

Traders can use the multi-release screen shown in Figure 7 to release a group of orders as a list order. An example of such screen is shown in Figure 7. The multi-release screen incorporates the trading algorithms in the counterparty list control 1 and also shows an "trading algorithm" column that contains details of the trading algorithms available. Both "copy/paste" and "copy to all" functionalities are available on this column, as well as on the other columns visible on the multi-release screen. If these two functions are used when copying from the counterparty column on a multi-release strip/grid containing trading algorithms, the trading algorithms, if any, in the destination rows are set to blank. However when copying from a trading algorithm on a multi-release strip/grid, data is only copied into rows in the grid where the counterparty is the same, and the trading algorithm restrictions are valid. If this is not the case, the destination trading algorithm is set to blank.

However when releasing orders as a Market List, normal orders and trading algorithms cannot be mixed, as there is only one FIX destination. The dropdown trading algorithm combination box in the multi-release strip/grid can
only contain options that are valid on the entire set of orders displayed. If a
user selects a trading algorithm from the counterparty list control 1 it is not
valid for all the orders in the list, the user will receive a warning message and
will be prompted to make a different selection or to reduce the list. This would
be the situation where some of the orders do not pass all of the trading
algorithm restrictions. An example of the window display to the user in this
case is shown in Figure 8.

Release Reviewer

The release reviewer as shown in Figure 9, is used to review details of the
existing releases, and also contains a column relating to the trading algorithm
available.

Algorithm Permissions

The user permission for each trading algorithm is managed from a central
administration application. This may be an existing application such as the
LZ Administration application, as available from LatentZero Limited. A trading
algorithms node appears in the user permissions pane on the user
maintenance screen. This node may be placed in alphabetical order in the
users permissions tree structure. The node has two sub-nodes; “trading
algorithms” and “customisation”.

The first sub-node trading algorithm in its expanded state will show the list of
all available standard trading algorithms nested under their corresponding
counterparty names and the current permissions of the selected user. A
trading algorithm is selected when its corresponding tick box is ticked. The
second sub-node customisation, has two sections detailing the current user
permissions accustom trading algorithms; public and private. Both public and
private sections have permission options for the actions: create, amend and
delete. Figure 10 illustrates the trading algorithm trading permissions tree.
The window is split into four panes: a user list 16 a trading algorithm list 17, a
trading desk list 18 and a user information input panel 19. In order to use the
administration application, and access the trading algorithm permissions, the
user must provide ID and a password. The trading algorithm permission tree is used for managing both standard and customer trading algorithms.

A custom trading algorithm administration tree structure is shown in Figure 11. This tree structure comprises the list of all available trading algorithms nested under their corresponding counterparty names (from the data table trading-algorithm table, discussed below) and associated custom trading algorithms (from the data table trading-algorithm-custom table as discussed below).

If a user right-clicks on a trading algorithm, a context menu 20 is displayed, containing 3 options, as appropriate for the selected user permission of the selected trading algorithm:

- "create Like/Edit": authorised users are able to create new custom trading algorithms from an existing standard or custom trading algorithm. The right hand side of the screen displays the trading algorithm parameter control 21. Users can update parameter values (except for the trading algorithm name) and click the "save" button on the administration tree structure.

- "Delete": if permissions allow, the selected trading algorithm may be deleted.

Restrictions

The order flow within the system is restricted. Each trading algorithm is only available on specified base types and markets, which are defined in the trading algorithm XML description. Traders can only save standard trading algorithms with other names in order to create custom trading algorithm for their own use. However standard trading algorithms cannot be deleted, changed or renamed. A standard trading algorithm is one defined by the offering counterparty. Traders can delete, change or rename the trading algorithms defined by them, but not those defined by other traders. This is a case even if the trading algorithms are defined as public domain. The "copy/paste" and "copy to all" option at the multi-release screen is not available for the field trading algorithm. In market lists, a change of counterparty in a release element sets the trading algorithm value to blank,
wherever the trading algorithm field is set to a different value. A change of counterparty/trading algorithm in one of the components from the counterparty list control 1 will result in the same pair being assigned to the remaining list components. Market lists also cannot mix trading algorithms and non trading algorithms.

The data table "trading-algorithm" table contains details linking counterparties with supported trading algorithms, as shown in Table 1 below:

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterparty_id</td>
<td>VarChar(16)</td>
<td>Foreign key</td>
</tr>
<tr>
<td>Algorithms</td>
<td>VarChar(4000)</td>
<td>XML description of trading algorithms supported by counterparty</td>
</tr>
</tbody>
</table>

Table 1

An example of this table populated for two counterparties is shown in Table 2:
<table>
<thead>
<tr>
<th>Counterparty_id</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broker1</td>
<td><code>&lt;izCounterparty name=&quot;Broker1&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;parameter name=&quot;Risk&quot; displayname=&quot;Risk&quot; type=&quot;char&quot; format=&quot;{LOW, MEDIUM, HIGH}&quot; default=&quot;MEDIUM&quot; control=&quot;combo box&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;parameter name=&quot;Expiry&quot; displayname=&quot;Expiry&quot; type=&quot;time&quot; format=&quot;UTC&quot; control=&quot;date/time control&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;restrictions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;markets&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>      &lt;market name=&quot;xx&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/markets&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;basetypes&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;!-- Security --&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;basetype id=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;!-- Derivatives --&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;basetype id=&quot;4&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/basetypes&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;conditions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;condition name=&quot;M&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;condition name=&quot;L&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/conditions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/restrictions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;algorithm name=&quot;MTC_VWAP&quot; value=&quot;1&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>      &lt;parameter name=&quot;Risk&quot; required=&quot;Y&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>      &lt;parameter name=&quot;Expiry&quot; required=&quot;Y&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;exclusions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>      &lt;markets&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>        &lt;market name=&quot;UK&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/markets&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/exclusions&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/algorithm&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;algorithm name=&quot;PRICE&quot; value=&quot;2&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>      &lt;parameter name=&quot;Riskr&quot; required=&quot;Y&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/parameters&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/algorithm&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/izCounterparty&gt;</code></td>
</tr>
</tbody>
</table>
<izCounterparty name="Broker2">
  <parameters>
    <parameter name="start" displayname="Start time" type="time" format="UTC" default="current" control="date/time"/>
    <parameter name="end" displayname="End time" type="time" format="UTC" default="16:00" control="date/time"/>
    <dependency minervaattribute="N" name="Starttime">
      <condition>=="/>
    </dependency>
  </parameters>
  <restrictions>
    <markets>
      <market name="US"/>
    </markets>
    <conditions>
      <condition name="M"/>
      <condition name="L"/>
    </conditions>
    <basetypes>
      <!-- Security -->
      <basetype id="1"/>
    </basetypes>
  </restrictions>
  <algorithm name="VWAP" value="1">
    <parameters>
      <parameter name="start" required="N"/>
      <parameter name="end" required="N"/>
    </parameters>
  </algorithm>
  <algorithm name="Automatic5" value="2">
    <parameters>
      <parameter name="aggr" required="N"/>
    </parameters>
  </algorithm>
  <algorithm>
    <algorithm name="aggress" value="3">
      <parameters/>
    </algorithm>
  </algorithm>
</izCounterparty>
Table 2

The data table "trading_algorithm_custom" table contains details of the user that created the trading algorithm, the values assigned for each parameter and the privacy status assigned:
<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_id</td>
<td>VarChar(10)</td>
<td>Composite key</td>
</tr>
<tr>
<td>Counterparty_id</td>
<td>VarChar(16)</td>
<td>Composite key</td>
</tr>
<tr>
<td>Algorithm_name</td>
<td>VarChar(20)</td>
<td>Composite key</td>
</tr>
<tr>
<td>Custom_name</td>
<td>VarChar(20)</td>
<td>Name assigned to the custom trading algorithm</td>
</tr>
<tr>
<td>Privacy_status</td>
<td>VarChar(10)</td>
<td>“Public” or “Private”</td>
</tr>
<tr>
<td>Algorithm_values</td>
<td>VarChar(600)</td>
<td>XML definition of trading algorithm parameters &amp; values</td>
</tr>
</tbody>
</table>

Table 3

An example of this table populated for two users is shown in Table 4:

<table>
<thead>
<tr>
<th>User_id</th>
<th>Cpy_id</th>
<th>Algorithm_name</th>
<th>Custom_name</th>
<th>Privacy_status</th>
<th>Algorithm_values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tinvestor1</td>
<td>SS</td>
<td>VWAP</td>
<td>MyVWAP1</td>
<td>Private</td>
<td>&lt;parameters&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;Risk Factor&quot; value=&quot;MEDIUM&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;Expire Time&quot; value=&quot;2004052916300000&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;/parameters&gt;</td>
</tr>
<tr>
<td>Tinvestor1</td>
<td>SS</td>
<td>ARRIVAL PRICE</td>
<td>Aggressive1</td>
<td>Public</td>
<td>&lt;parameters&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;Risk Factor&quot; value=&quot;HIGH&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;/parameters&gt;</td>
</tr>
<tr>
<td>Tinvestor2</td>
<td>BS</td>
<td>TWAP</td>
<td>MyTWAP1</td>
<td>Private</td>
<td>&lt;parameters&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;Start time&quot; value=&quot;2004052909300000&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;End time&quot; value=&quot;2004052916150000&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;parameter name=&quot;MaxPctVolume&quot; value=&quot;50&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;/parameters&gt;</td>
</tr>
</tbody>
</table>
XML Schema

The trading algorithm details provided by the sell-side counterparties are defined in the form of a XML schema. As brokers support new trading algorithms, these can be added easily to the schema.

As discussed above, the release table includes a trading algorithm column, which when stored in XML format has the characteristics shown in Table 5 below:

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type &amp; Size</th>
<th>Values</th>
<th>Comment</th>
</tr>
</thead>
</table>
| algorithm | VarChar(200)     | Algorithm name and parameter values         | Example: <algorithm name="Vwap" Starttime="090100" Endtime="150000" MaxPctVolume="50"/>

Table 5

The fix adapters pause and insert the trading algorithm parameters into the correct tags in the release table.

Users may wish to set various counterparty parameters differently depending on whether FIX releases are algorithmic or non-algorithmic. In order to enable this an additional Algorithmic column is added to the counterparty_parameters table.

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type &amp; Size</th>
<th>Values</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm</td>
<td>Char</td>
<td>Y</td>
<td>blank</td>
</tr>
</tbody>
</table>

Table 6

A counterparty parameter where Algorithmic="Y" will override a parameter of the same name where Algorithmic=blank.

A new parameter_name, valid only for trading algorithms, is added to this table:

- Parameter_name: "parameters_amended_supported";
• Possible values: "Y|N";
• Comment: set to "Y" indicates that the counterparty supports trading algorithm parameter amendments.

Algorithmic parameters are shown in Table 7 below:

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Data Type</th>
<th>Values</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;tif_values_supported&quot;</td>
<td>VarChar(100)</td>
<td>List of valid TIF separated by commas</td>
<td>TIFs supported on algorithmic trades</td>
</tr>
<tr>
<td>&quot;order_cancel_replace_supported&quot;</td>
<td>Char</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>&quot;parameters_amendment_supported&quot;</td>
<td>Char</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>&quot;fix_version_id&quot;</td>
<td>VarChar(5)</td>
<td>Fix protocol version</td>
<td>Fix protocol version for counterparty (i.e FIX.4.2)</td>
</tr>
<tr>
<td>&quot;new_order_single_supported_file&quot;</td>
<td>Char</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

Table 7

The XML Schema

Below is a guide to the Trading algorithms XML schema as listed in Table 4.

Counterparty level definitions:

Parameter default values:

The "parameters" tag following a counterparty name describes all trading algorithm parameters for that counterparty. The "default" tag in the parameters section will define the default values. For instance, default="MEDIUM".

Parameter GUI control:
The "control" tag in the parameters section will define the parameter GUI control such as combo box or time control.

Separators used in Format tag:

Multiple values will be separated by commas, for instance format="Low, Medium, High". An ellipsis will indicate a range of values, for instance format="1.9".

Dependencies:

Certain parameter values may be dependent on other parameters or Minerva order or release attributes, for instance, the parameter "EndTime" must be greater than parameter "StartTime". Dependencies are expressed with 3 different tags grouped under the "dependency" tag:

"Minervaattribute": Set to 'Y' indicates a dependency on a Minerva attribute when used in conjunction with the Minerva application available from LatentZero Limited. Set to 'N' indicates a dependency on another parameter;

"Name": name of the parameter or attribute on which the parameter value is dependent;

"Condition": expresses the nature of dependency with standard relational operators. Multiple operators will be separated by ";" (e.g. "&lt;=" indicates "less than" or "equal to").

Counterparty support restrictions

The "restrictions" tag describes combinations of order and release attributes restricting the algorithmic support by any given counterparty. For instance, the example in table 2 indicates that Broker 2 supports trading algorithms only in US and UK markets (<restrictions><markets><market name="US"/></markets><markets><market name="UK"/></markets>).

There are three types of restrictions defined by the corresponding tag names:

Market: specifies instances of orders.market_id where the counterparty offers trading algorithms (e.g. "US") ("*" will mean all markets);
**Base types**: specifies instances of orders.Minerva_instrument_type for which the counterparty offers trading algorithms (e.g. "1") ("**" will mean all types);

**Condition**: specifies instances of releases.condition for which the counterparty offers trading algorithms (e.g. "M").

Algorithm level definitions

Algorithm exclusions

The "exclusions" tag describes combinations of order and release attributes for which algorithm level settings will override counterparty level settings. For instance, the example in table 2 indicates that Broker 1 offers the algorithm "MTC_VWAP" in all markets except the UK (<restrictions><markets><market name="**"/></markets><restrictions>...<exclusions><markets><market name="UK"/></markets><exclusions>).

There are three types of restrictions defined by the corresponding tag names:

**Market**: specifies instances of order.market_id where the counterparty does not offer a specific trading algorithm (e.g. "UK") ("**" will mean all markets);

**Base types**: specifies instances of order.Minerva_instrument_type for which the counterparty does not offer a specific trading algorithm (e.g. "1") ("**" will mean all types);

**Condition**: specifies instances of releases.condition for which the counterparty does not offer a specific trading algorithm (e.g. "L").

Required parameters:

When parameters are mandatory for a trading algorithm the "required" tag will be set to "Y".

Various other modifications are possible and will occur to those skilled in the art without departing from the scope of the invention, which is defined by the following claims.
CLAIMS:

1. An algorithmic trading system comprising:
   a file structure for storing encoded trading algorithms selectable by a user,
   each encoded trading algorithm having at least one encoded trading algorithm
   parameter associated therewith to vary the operation of the algorithm on input of
   algorithm parameter values;
   input means to allow a user to select an encoded trading algorithm from
   the file structure, to input trading information and to enter algorithm parameter
   values into the system, for the selected encoded trading algorithm; and
   communication means to allow the trading information and algorithm
   parameter values to be communicated to a third party in accordance with the
   selected encoded trading algorithm.

2. The algorithmic trading system of claim 1, wherein the file structure is a
   database.

3. The algorithmic trading system of claim 1 or 2, wherein the input means
   comprises at least one of a keyboard, mouse or tracker ball.

4. The algorithmic trading system of claim 1, 2, or 3, wherein the input
   means further comprises a visual display for displaying the trading information
   and algorithm parameter values.

5. The algorithmic trading system of any preceding claim, wherein the
   communication means is a wireless network.

6. The algorithmic trading system of any of claims 1 to 4, wherein the
   communications means is a wired network.

7. The algorithmic trading system of claim 6, wherein the wired network
   comprises a telecommunications line.
8. The algorithmic trading system of any preceding claim, wherein trading information and algorithm parameter values are communicated to a third party using a FIX (financial information exchange) messaging protocol.

9. The algorithmic trading system of any preceding claim, wherein the file structure is remote from the trading system.

10. The algorithmic trading system of any preceding claim, wherein the encoded trading algorithm parameters associated with each encoded trading algorithm are stored in the file structure.

11. The algorithmic trading system of any preceding claim wherein the trading algorithms and trading algorithm parameters are encoded in XML (extensible mark-up language).

12. The algorithmic trading system of claim 11, wherein the file structure is updatable.

13. The algorithmic trading system of claim 12, wherein the file structure is updated by the addition or removal of trading algorithms.

14. The algorithmic trading system of claim 12 or 13, wherein the file structure is updated by the addition or removal of trading algorithm parameters.

15. The algorithmic trading system of any preceding claim, wherein the algorithm parameter values are required values or optional values.

16. The algorithmic trading system of claim 15, wherein required values and optional values are stored in and retrieved from the file structure.

17. The algorithmic trading system of claim 16, wherein the required values are allocated a user-defined file name.

18. The algorithmic trading system of claim 16 or 17, wherein the optional values are allocated a user-defined file name.
19. The algorithmic trading system of any of claims 16, 17 or 18, wherein the algorithm parameters are publicly available.

20. The algorithmic trading system of any of claims 16, 17 or 18, wherein the algorithm parameters can only be accessed by a registered user.

21. The algorithmic trading system of any of claims 15 to 20, wherein when an optional value is not input by the user, a default value is provided.

22. The algorithmic trading system of claim 21, wherein the default values are stored in and retrieved from the file structure.

23. A method of updating the algorithmic data available to an algorithmic trading system, comprising the steps of:
   encoding the algorithmic data;
   storing the encoded algorithmic data in a file structure;
   retrieving the algorithmic data from the file structure and uploading it into the algorithmic trading system.

24. The method of claim 23, wherein the algorithmic data comprises at least one trading algorithm.

25. The method of claim 23 or 24, wherein the algorithmic data comprises at least one trading algorithm parameter.

26. The method of any of claims 23 to 25, wherein the file structure is a database.

27. The method of any of claims 23 to 26, wherein the algorithmic data is encoded in XML (extensible mark-up language).

28. An algorithmic trading method comprising the steps of:
   encoding algorithmic data;
   storing the algorithmic data in a file structure;
selecting an encoded trading algorithm via a user input device, the operation of the encoded trading algorithm being variable based on algorithm parameter values input to at least one algorithm parameter associated with the encoded trading algorithm;

accessing the encoded trading algorithm from the user input device;
inputting algorithm parameter values for the selected encoded trading algorithm and inputting trading information; and
communicating the trading information and algorithm parameter values, in accordance with the algorithm parameters, to a third party.

29. The algorithmic trading method of claim 28, wherein the algorithmic data comprises at least one trading algorithm and at least one algorithm parameter.

30. The algorithmic trading method of claim 29, wherein the file structure is a database.

31. The algorithmic trading method of claim 29 or 30, wherein inputting data comprises typing on a keyboard, or selecting data using a mouse or a tracker ball.

32. The algorithmic trading method of any of claims 29, 30, or 31, further comprising displaying the algorithm parameters values for the selected encoded trading algorithm and the trading information.

33. The algorithmic trading method of any of claims 29 to 32, wherein communicating to the third party takes place via a wireless network.

34. The algorithmic trading method of any of claims 29 to 32, wherein communicating to the third party takes place via a wired network.

35. The algorithmic trading method of claim 34, wherein the wired network comprises a telecommunications line.
36. The algorithmic trading method of any of claims 29 to 35, wherein communicating trading information and algorithm parameter values to the third party is via a FIX (financial information exchange) messaging protocol.

37. The algorithmic trading method of any of claims 29 to 36, further comprising accessing the algorithmic data remotely from the trading system.

38. The algorithmic trading method of any of claims 28 to 37, further comprising associating at least one encoded trading algorithm parameter with each encoded trading algorithm.

39. The algorithmic trading method of any of any of claims 28 to 38, wherein the step of encoding the algorithmic data comprises encoding the algorithmic data in XML (extensible mark-up language).

40. The algorithmic trading method of claim 39, further comprising updating the algorithmic data with additional algorithmic data.

41. The algorithmic trading method of claim 40, wherein the file structure is updated by the addition or removal of trading algorithms.

42. The algorithmic trading method of claim 40 or 41, wherein the file structure is updated by the addition or removal trading algorithm parameters.

43. The algorithmic trading method of claim 39, wherein the algorithm parameter values are required values or optional values.

44. The algorithmic trading method of claim 43, further comprising storing the required values and optional values in and retrieving the required and optional values from the file structure.

45. The algorithmic trading method of claim 44, further comprising allocating the required values a user defined file name.
46. The algorithmic trading method of claim 44 or 45, further comprising allocating the optional values a user defined file name.

47. The algorithmic trading method of any of claims 43 to 46, further comprising enabling the algorithm parameters to be publicly available.

48. The algorithmic trading method of any of claims 43 to 46, further comprising enabling the algorithm parameters to be available only to a registered user.

49. The algorithmic trading method of any of claims 43, 44, 45 or 46, further comprising providing a default parameter value when no optional value is input by the user.

50. The algorithmic trading method of claim 49, further comprising storing and retrieving the default values from the file structure.

51. A computer program product, for updating the algorithmic data available to an algorithmic trading system, wherein the algorithmic data comprises at least one trading algorithm associated with at least one trading algorithm parameter, the operation of the algorithm variable by algorithm parameter values input by a user, having program code stored thereon which when executed on a computer causes the computer to perform the steps of:

   encoding the algorithmic data;
   storing the encoded algorithmic data in a file structure;
   retrieving the algorithmic data from the file structure and uploading it into the algorithmic trading system.

52. A computer program product for an algorithmic trading system, having program code stored thereon which when executed on a computer causes the computer to perform the steps of:

   encoding algorithmic data;
   storing the algorithmic data in a file structure;
   selecting an encoded trading algorithm via a user input device;
accessing the encoded trading algorithm and at least one associated encoded algorithm parameter, the operation of the algorithm variable by algorithm parameter values input by a user, from the user input device;
inputting algorithm parameter values for the selected encoded trading algorithm and inputting trading information; and
communicating the trading information and algorithm parameter values, in accordance with the algorithm parameters, to a third party.

53. The computer program product of claim 52, wherein the algorithmic data comprises at least one trading algorithm and at least one algorithm parameter.

54. A system for arranging algorithms to create an algorithmic trading system, the system comprising:
input means for receiving trading algorithms, the operation of each trading algorithm variable by one or more trading algorithm parameters;
input means for receiving one or more trading algorithm parameters; and
a file structure, arranged to store the trading algorithms and trading algorithm parameters in a manner accessible to a user to create an algorithmic trading system.
Application No: GB0417803.4
Claims searched: 23-27, 51 and 54

Examiner: Dr Mark Gainey
Date of search: 22 April 2005

Patents Act 1977
Further Search Report under Section 17

Documents considered to be relevant:

<table>
<thead>
<tr>
<th>Category</th>
<th>Relevant to claims</th>
<th>Identity of document and passage or figure of particular relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>X,Y</td>
<td>X:23-26, 51 and 54 Y:27</td>
<td>WO 01/37166 A2 (ABOVETRADE.COM) see abstract</td>
</tr>
<tr>
<td>Y</td>
<td>27</td>
<td>WO2003/36540 A (THE ASHTON TECHNOLOGY GROUP) see abstract and figure 1</td>
</tr>
</tbody>
</table>

Categories:

<table>
<thead>
<tr>
<th>X</th>
<th>Document indicating lack of novelty or inventive step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Document indicating lack of inventive step if combined with one or more other documents of same category.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Member of the same patent family</td>
</tr>
<tr>
<td>A</td>
<td>Document indicating technological background and/or state of the art.</td>
</tr>
<tr>
<td>P</td>
<td>Document published on or after the declared priority date but before the filing date of this invention.</td>
</tr>
<tr>
<td>E</td>
<td>Patent document published on or after, but with priority date earlier than, the filing date of this application.</td>
</tr>
</tbody>
</table>

Field of Search:
Search of GB, EP, WO & US patent documents classified in the following areas of the UKC:
G4A

Worldwide search of patent documents classified in the following areas of the IPC:
G06F

The following online and other databases have been used in the preparation of this search report:
EPODOC, WPI, Selected Internet
Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

<table>
<thead>
<tr>
<th>Category</th>
<th>Relevant to claims</th>
<th>Identity of document and passage or figure of particular relevance</th>
</tr>
</thead>
</table>
| X        | 1-8,15,28-36 and 52 at least | WO2003/36540 A
THE ASHTON GROUP see abstract, figure 1, p.7 1.20-p.8 1.6, p.12 1.14- p.13 1.4, p.14 11.3-12. |
| X        | 1-8,15,28-36 and 52 at least | US2003/0233306 A1
MADHAVAN et al. see abstract, figure 1 paragraphs 6,7,10 and 16. |
| X        | 1-8,15,28-36 and 52 at least | WO 01/37166 A2
ABOVETRADE.COM, see abstract and figure 1. |

Categories:

| X | Document indicating lack of novelty or inventive step |
| Y | Document indicating lack of inventive step if combined with one or more other documents of same category. |
| & | Member of the same patent family |
| A | Document indicating technological background and/or state of the art. |
| P | Document published on or after the declared priority date but before the filing date of this invention. |
| E | Patent document published on or after, but with priority date earlier than, the filing date of this application. |

Field of Search:
Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^w:

G4A
Worldwide search of patent documents classified in the following areas of the IPC^7:
G06F
The following online and other databases have been used in the preparation of this search report:
EPODOC, JAPIO, WPI, Selected Internet