CONFIGURABLE ANONYMOUS TRADING SYSTEM

TECHNICAL FIELD

The present invention relates to a computer trading system for providing an electronic broking service for tradable items such as foreign exchange and financial instruments generally. In particular, the invention relates to a computer trading system having a plurality of order input devices such as trader terminals connected to a network for submission and matching of bids, offers, buy and sell orders.

BACKGROUND TO THE INVENTION

An anonymous trading system is known, for example, in EP-A-0,399,850, EP-A-0,406,026 and EP-A-0,411,748 which disclose an automated matching system for anonymous trading of foreign currencies (or other financial instruments). In this system, a single host computer maintains a central database of all trading instruments available for trade, credit information and bids and offers which have been submitted by terminals connected to the host via a computer network. The host computer uses information in its central database to match bids and offers and buy and sell orders based on matching criteria which include a counter party credit limit.

The counter party credit limits are set at each trading floor, and are stored at the host computer, which then establishes a gross counter party credit limit for each possible pair of counter-parties. The gross counter party credit limit is the minimum amount of the remaining credit from a first party to a second party, and the second party to the first party. The various trader terminals

connected to the host computer maintain and display only a restricted subset of the information available at the host computer, such as best bids and offers.

A problem was identified with this system in that the host computer only used the credit information to check that a deal could proceed after a potential match had been identified. A trader thus could not know whether he had credit with a potential counter party prior to attempting to trade. This problem was identified and a solution provided in the system disclosed in US-A-5,375,055, the contents of which are incorporated herein by reference.

In the system disclosed in US-A-5,375,055 a credit matrix is derived and stored at a plurality of regional nodes of a distributed network, with each regional node distributing market information to a set of trader terminals to which the regional node is connected via an access node. The regional node is known as a Market Distributor and provides dealable price information to the trader terminals connected via the access node known as a Market Access Node. The actual matching of bids, offers, buy and sell commands is provided by separate nodes known as Arbitrators.

In the field of online trading systems, there are different systems providing the facility to match and trade financial instruments, including foreign exchange and future instruments, as well as shares. In each case, there are specific requirements of the system to handle the varying nature of the item traded. As a result, each existing system is only able to trade one category of items. We have appreciated that this has limitations in that a new system must be developed each time a new item is desired to be traded. In the financial world in

particular this can hold back the development of new systems.

We have appreciated that a better approach is required. In particular, we have appreciated that a trading platform could be developed to trade many different item types, but would need to be of a very different architecture compared to existing trading system designs.

SUMMARY OF THE INVENTION

Accordingly, there is provided a computer trading platform adaptable for trading a plurality of different tradable items, comprising a plurality of subsystems, each subsystem having an interface to a tradable item specific plug-in, the platform providing: an order management subsystem for receiving and distributing quotes and for matching quotes and orders in accordance with criteria provided by an order management plug-in, a deal execution subsystem for confirming and recording deals matched by the order management subsystem, a market view subsystem for providing market views to traders, and a credit subsystem for managing credit utilisation, wherein each subsystem is communicable with an appropriate plug-in defining rules respectively for matching , deal execution, market view and credit thereby creating a trading system.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described, by way of example only, in which:

Figure 1: is a diagram of the main components of the physical architecture of a trading system;

Figure 2: is a diagram of the main function components of a Broker node; and

Figure 3: is a diagram of the interrelation of subsystems and plug-ins.

DESCRIPTION OF A PREFERRED EMBODIMENT

The embodiment described is a broking platform which is adaptable to provide a trading system for many different items. A tradable item is any item that can be traded including, but not limited to, financial instruments such as foreign exchange, and also shares and commodities.

The purpose of splitting the Broking Platform into application and core is to allow the core to be developed as a stable, re-usable framework that can be used to create applications more easily. Maintenance effort is decreased because much of the system is contained in a reusable package that is maintained as a single effort for all applications. Application specific business functionality is contained in small distinct components that can be analysed, understood, and maintained. Dividing the core and application into components allows parallel design, development, and testing of important pieces of the Broking Platform which in turn provides for a more robust and maintainable product with decreased time to delivery. Another benefit of this effort is that applications may be designed and constructed in parallel with Broking Platform core development efforts, because the core is separated from the application by well-defined interfaces.

The embodiment will first be described in relation to one possible physical architecture, before turning to the key

aspects of the modular design. The physical architecture of the broking platform is shown in figure 1.

The computer trading system embodying the invention is a distributed system. In this document distributed is taken to mean that the functions performed by the network are distributed amongst a plurality of nodes. A key aspect of the embodying network is that each of the nodes performs equal functions of both price distribution and order matching services.

The purpose of the embodying system is to allow traders to enter quotes and orders which are then matched within the system. The system provides a platform for trading at least the following instruments: FX Spot, FRA, and Forwards and also FX Forwards, CFDs, short-dated government and/or central bank paper, commercial bills, CDs, inter-bank deposits, commercial paper, reports, interest-rate futures, swaps, options money markets, metals, call money, government bonds and other short term interest rate instruments. These are all referred to as financial instruments.

A brief overview of the embodying network will first be provided with reference to figure 1, followed by a more detailed discussion of the embodiment with reference to the subsequent figures. The computer trading system of figure 1 comprises a plurality of trading agents 10 each connected to at least one of a plurality of broker nodes 12. The broker nodes 12 are [type of computer and software to be discussed in brief]. The trading agents 10 are [type of computer and software to be discussed in brief]. Each trading agent is the means by which traders access the trading system.

A Broker node 12 provides the basic order matching and price distribution services. The Brokers are arranged in a structure called a Clique Tree which enables faster communications routing, following very specific but simple rules. The Clique Tree is a network structure where individual nodes are grouped into Cliques, and the Cliques are then arranged into a tree structure. Each Broker can be linked logically to a number of Brokers, which are referred to as its neighbor Brokers. Communication between Brokers is on an equal level, with no "up" or "down" direction in the network.

The Trading Agent node provides services related to a specific trading floor or group of traders. The Trading Agent is the node that provides the interface to a trading floor. Each Trading Agent is connected to at least one Broker to access the network. Any number of Trading Agents can be connected to one or more Brokers in the network. There are no "root" or "leaf" Brokers; all Brokers are equal and can accept Trading Agent connections. A plurality of trader terminals are connected to each trader agent.

While Trading Agents must be connected to at least one Broker, they themselves are not members of the Clique Tree, but remain outside the structure. A Trading Agent connected to multiple Brokers will receive multiple sets of market prices. Even though the price information from different Brokers can be substantially the same, the information may be received at different intervals. A Trading Agent will send a given trading order to only one Broker.

The Brokers are equal to each other, and perform the same functions. The arrangement of the network or their position in it is transparent to the brokers. They only need to know about their neighbors. Each Broker has: knowledge of all orders in the market, and is able to match orders as soon as they are submitted. Each Broker also maintains a full list of orders in the market and is therefore able to customize market views as needed by the Trading Agents and is able to react faster to market information as soon as it is received.

The modular nature of the embodying systems will now be described in relation to the various subsystems which provide the key aspects of the design with reference to Figure 2. A subsystem is a component existing within the embodiment that encapsulates a major area of functionality whose services are exposed by one or more interfaces. Each interface is presented by a subsystem agent, and may be invoked directly by local procedure or method invocation. An agent may implement the services provided by the subsystem and/or serve as a proxy to utilise subsystem implementation external to the agent or subsystem implementation distributed over the network. Subsystems are customised for specific applications via Plug-Ins. The term plug-in means a functional component that can be added to the basic platform to provide specific functionality required by a chosen tradable item.

The subsystems in the present embodiment are an order management subsystem 14, a deal execution subsystem 16, a market view subsystem 18, a market inventory subsystem 20, a credit subsystem 22 and a market definition subsystem 24. Each of these will be described further later.

The trading platform comprises the subsystems and associated plug-ins, and also common objects, application specific objects and systems services.

Plug-Ins are used to provide application specific functionality to a subsystem. This is accomplished via the application's implementation of a plug-in interface, which is defined as part of the subsystem specification. Plug-Ins may use any of the services provided by their associated subsystem or any other subsystem as required via the subsystems' agents. Plug-Ins may also use system services.

Common objects consist of the set of all objects that are used in the definition of subsystem service interfaces. The broking platform defines the type of a common object, but it may be implemented or extended as required by the application via inheritance as given by the common object component specifications. Common objects may contain one or more application specific objects.

Application specific objects contain state and behaviour required by a given trading system application. Such objects are created by the broking platform client or by plug-ins. The broking platform provides transport and delivery of the application specific objects via common objects. Application specific objects may also be used directly by a subsystem agent's interface.

System services provide general services to subsystems, common objects, and plug-ins. System services are distinguished from Subsystems because system services do not have any application specific behaviour (no Plug-Ins) and are provided completely by the broking platform. System services are not generally related to specific

business functionality and typically provide services of a technical nature. System services are categorised into groups of like services.

The interaction of key components in the embodiment is shown in figure 3. The example shows two subsystems, subsystem A 30 and subsystem B 32. A plug-in for subsystem B 34 provides application specific services to subsystem B 32. In addition, subsystem B 32 utilizes the services of Subsystem A 30 via a local instance of the Subsystem A Agent 36. The application specific functionality contained in Subsystem B's Plug-In also makes use of the services of Subsystem A through agent 40. The services of Subsystem A are invoked through a method interface using Common Objects that contain Application Specific Objects communicated between A and B 38,42. Subsystems and their Plug-Ins as well as Common Objects may invoke System Services 44 as required to provide general services such as network transport or event logging services.

To create a trading system, therefore, a set of plug-ins are selected with chosen functionality to interface with the subsystems shown in Figure 2. The plug-ins interact with the various subsystems using agents and objects as described.

In addition to the six main subsystems shown in Figure 2, a further subsystem called the Broking Platform Client provides a set of services referred to as the Broking Platform API, which gives application clients access to key services provided by other subsystems and system services.

The Broking Platform Client Subsystem, also referred to as the BPC, presents the services of the Broking Platform Core to the application client. The collection of services provided through the BPC is referred to as the Broking Platform API (application programmers interface). By using the BPC, the application client can utilize the services of the core using a simple well-defined interface.

The Broking Platform Client Subsystem provides services to submit an order as specified by an order parameter. Subsequent changes in order status or deals executed against the order are communicated via an interface. An order identifier for the newly submitted order is returned. A service is provided to allow the client to confirm or deny a proposed deal, to interrupt an order and to allow the client to establish a session with the BP Core. The trader identity is designated by a TraderId with authentication credentials contained in Identity. The BPC also allows the client to request product element information based on product element data. The BPC also provides a service to allow the client to request a subscription to a market view for a given product. Market views are given to the client via an interface.

The client subsystem also provides application Specific Objects within the client subsystem, including an object which allows the application client to provide application-specific order submission details, an object which allows the application client to provide application-specific market view criteria details to specify the construction and delivery of market views, and an object which provides product element specific details to the application client for each product.

The order management Subsystem 14 provides the broking service for traders. It provides for submission and distribution of orders, matching orders to propose deals. New deals are submitted to the deal execution subsystem for deal completion. Order Management Subsystem is responsible for manipulating orders based on market events when applicable. The entire life cycle of orders is managed by the Order Management Subsystem based on the trading rules of the system and the features of orders.

The plug-in interface of the order provides the management system instrument-specific operation for handling order submission, allows the application to perform validation on a newly submitted order, and creates new deals given a new order. If no matches are available, then the order must be inserted into the order book, activated for priced distribution as required, and distributed to other brokers. The Application Specific Objects of the Order Management System contain any application specific order submission details.

The Deal Execution Subsystem 16 is as follows. The Deal Execution Subsystem is responsible for routing a pending deal back through the system, back to the originating trader workstation to allow order confirmation. New deals are provided by the order management subsystem and are executed by this subsystem. This subsystem is responsible for updating the market on all passive brokers, verifying credit, confirming orders, recording deals, ticket printing and any other activity resulting from the execution or completion of a deal. This subsystem works with the order management subsystem in order to obtain the appropriate response interface to allow correct client notification.

The Deal Execution Subsystem is responsible for creating a deal object and initiating deal execution, and for performing any application specific deal initiation activities. Application Specific Objects contain any application specific order submission details. This information originates from the client's original submit order request. Other objects may be used by the various plug-in methods to provide application specific behaviour.

The credit subsystem 22, is responsible for the managing utilization against credit allocations. This subsystem is also responsible for notification of credit observers whenever a credit allocation changes state, between an out-of-credit and a credit-available state. This subsystem is responsible for confirming and/or adjusting deal amounts based on available credit. The credit subsystem is also responsible for certain administrative functions, such as allowing administration of credit allocations and credit recipients.

The Credit Subsystem Interface provides a service which is responsible for checking for credit compatibility for two counterparties for two orders. If the parties are not credit compatible, a false is returned and deal amount is zero. If the parties are credit compatible, then the amount desired is used to compute a credit amount. If the entire amount desired exceeds available credit, then a deal amount based on available credit is computed. If this amount is less than the minimum deal amount, then these parties are considered to be credit incompatible for these two orders and a false value is returned with a deal amount of zero. Otherwise true is returned and the deal amount reflects the amount that is reserved for a new deal.

The Plug-In Interface of the Credit Subsystem is responsible for calculating a credit amount based on a proposed deal amount and the original orders against which the deal would be performed. A return amount of zero implies that the counterparties are not credit compatible for the given orders. The interface is also responsible for calculating a deal amount based on a credit amount. The deal amount returned is based on the two orders for which the deal would be performed. A return amount of zero implies that the counterparties are not credit compatible for the given orders.

The Application Specific Objects of the Credit Subsystem contain any application specific order submission details, and provides objects which may be used by the various plug-in methods to provide application specific behaviour.

The Market Inventory 20, subsystem manages orders and deals for an instrument. Orders and deals for different product elements are organized into order books and deal books. Market Inventory is comprised of collection of order books and deal books for various instruments. This sub-system is responsible for maintaining a collection of order books and providing controlled and secured access to the order books and deal books to other sub-systems.

The Market Inventory Subsystem Interface includes a method which allows the other sub-systems to add an order to an order book for a specific instrument. This method call will invoke the Order plug-in which will position the order in the order book in the right order and will perform pre-order insertion and post-order insertion processing. A further method calculates an aggregate for

an order for a specific product element, and adds a deal to a deal book for a specific instrument.

The Plug-in Interface of the market inventory subsystem does the following:

1. work with order
2. finds position (Order Book ordering)
3. pre-order insertion (Order, OrderBook)
4. inserts order in inactive state
5. post-order insertion

These functions are application specific as the rules for Order Book ordering, pre-order insertion and post-order insertion processing may differ among the applications.

The market definition sub-system 24 provides facilities for accessing and administering all market, instrument, product, and product element data. The responsibilities include providing a means to a user to enter information, to modify the data, and to provide the information to Broking Platform system. It defines what instruments and product elements are traded, the trading method used (types of orders supported, type of order operations supported, rules for deal initiation and execution etc.), market level and instrument level trading rules, and who (institutions/trading floors) can trade. It defines and handles market meta data and market operations such as open, close, suspend, enable etc.

The Subsystem Interface of the market definition subsystem includes a method which results in validation of an order submitted into the Broking Platform system. It in turn invokes an application plug-in as order validation rules may be different for different applications. The Plug-in Interface of the market definition subsystem validates an

order based on validation rules as defined for and by the application.

In order to understand where the market is at, traders must have a view of the market. The Market View 18, subsystem provides them with this information. The Market View subsystem is responsible for providing the ability to subscribe to market views, for creating market views, and for insuring market views are distributed to the subscribers. A method is provided which allows a subsystem to request a subscription for a market view. Since market views are inherently application specific, this method will call the application's MarketView plug-in to create an initial market view.

CLAIMS

1. A computer trading platform adaptable for trading a plurality of different tradable items, comprising a plurality of subsystems, each subsystem having an interface to a tradable item specific plug-in, the platform providing:
an order management subsystem for receiving and distributing quotes and for matching quotes and orders in accordance with criteria provided by an order management plug-in,
a deal execution subsystem for confirming and recording deals matched by the order management subsystem,
a market view subsystem for providing market views to traders, and
a credit subsystem for managing credit utilisation,
wherein each subsystem is communicable with an appropriate plug-in defining rules respectively for matching , deal execution, market view and credit thereby creating a trading system.

2. A computer trading platform according to claim 1, wherein the order management subsystem includes means for matching orders based on tradable item specific criteria.

3. A computer trading platform according to claim 1 or 2, wherein the order management includes an interface for receiving tradable item specific criteria.

4. A computer trading platform according to claim 1, 2 or 3, wherein the order management subsystem maintains an order book of quotes.



5. A computer trading platform according to any preceding claim, wherein the credit subsystem comprises means for checking the credit compatibility of parties to a potential match.

6. A computer trading platform according to claim 5, wherein the means for checking credit compatibility comprises an interface for passing the proposed deal amount to a credit plug-in, and for receiving a credit utilization from the credit plug-in.