McDermott
Will & Emery

Boston Brussels Chicago Dallas Düsseldorf Frankfurt Houston London Los Angeles Miami
Milan Munich New York Orange County Paris San Francisco Seoul Silicon Valley Washington, DC

Strategic alliance with MWE China Law Offices (Shanghai)

Ahsan Shaikh
Attorney at Law
ashaikh@mwe.com
+1 312 984 7750

October 25, 2018

VIA FEDERAL EXPRESS

Gain Capital
Attn: Glenn H. Stevens
135 US Highway 202/206, Suite 11
Bedminster, NJ 07921

Re: Notice of OANDA Corporation Patent Rights
Client Matter No. 096803.0017

Dear Mr. Stevens:

We represent OANDA Corporation ("OANDA") in connection with certain intellectual property matters, including patent matters. As you know, OANDA is highly respected in the financial sector, providing leading and innovative solutions in the areas of currency data, trading, and exchange rate services for a wide range of organizations and investors.

At considerable expense, OANDA has pursued and obtained certain intellectual property protection for its innovative products, including numerous patents, trademarks, trade dress, copyrights, and other intellectual property related to its business. For years, OANDA has owned various patents related to its unique solutions to providing currency trading services, including the following:

Patent Reference No.	**Title**
U.S. Patent No. 7,146,336	Currency trading systems, methods, and software
U.S. Patent No. 7,496,534	Methods for trade decision making
U.S. Patent No. 8,392,311	Currency trading systems, methods, and software

We have attached copies of the above-listed patents for your convenience.

As we understand it, Gain Capital currently sells, offers for sale, uses, or imports currency trading products or services in the United States that may infringe OANDA's patents, including one or more of those listed above.

By way of example only, claim 1 of U.S. Patent No. 8,392,311 recites:

> 1. A method of trading currencies over a computer network connecting a trading system server and at least one trading client system, comprising the steps of
>
> (i) at the trading system server, determining and dynamically maintaining a plurality of current exchange rates, each current exchange rate relating to a pair of currencies and including a first price to buy a first currency of the pair with respect to a second currency of the pair and a second price to sell the first currency of the pair with respect to the second currency of the pair;
>
> (ii) transmitting data from the trading system server to a trading client system, the transmitted data representing at least one current exchange rate at the time of the transmission;
>
> (iii) at the trading client system, displaying the first and second prices for each received current exchange rate to a user,
>
> (iv) at the trading client system, accepting input from the user identifying a pair of currencies the user desires to trade, an amount of at least one currency of the pair desired to be traded and a requested trade price at which it is desired to effect the trade;
>
> (v) transmitting the accepted input from the trading client system to the trading system server;
>
> (vi) at the trading system server, comparing the requested trade price to the respective first price or second price of the corresponding current exchange rate at that time and, if the respective first price or second price of the corresponding current exchange rate at that time is equal to or better than the requested trade price, effecting the trade at the corresponding respective current exchange rate first price or second price and if the corresponding current exchange rate is worse than the requested trade price, refusing the trade; and
>
> (vii) transmitting from the trading system server to the trading client system an indication of whether the trade was refused or transacted and, if transacted, an indication of the price the trade was transacted at.

To the extent Gain Capital sells or offers to sell products or services that are covered by OANDA's patents, Gain Captial may be liable for patent infringement, directly or indirectly, under one or more subsections of 35 U.S.C. § 271. Though it is likely that you were already aware of OANDA and its patent portfolio, to the extent you had not been, we hereby put you on notice of the potential for infringement and Gain Capital's liability thereunder.

To the extent the sale or use of Gain Capital's products or services infringe one or more of OANDA's patents, those acts would allow OANDA to seek numerous equitable and legal remedies against Gain Capital, including, without limitation, an injunction, statutory damages, enhanced damages for willful infringement, attorney's fees, and any other remedies available under applicable law.

To avoid a potential dispute, we recommend that Gain Capital review OANDA's patents and discuss with us a potential license. To the extent Gain Capital wishes to discuss such a license, please contact me directly by November 9, 2018.

Sincerely,

Ahsan Shaikh

Enclosures:

- U.S. Patent No. 7,146,336
- U.S. Patent No. 7,496,534
- U.S. Patent No. 8,392,311

US007146336B2

(12) **United States Patent**
Olsen et al.

(10) **Patent No.:** **US 7,146,336 B2**
(45) **Date of Patent:** **Dec. 5, 2006**

(54) **CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE**

(75) Inventors: **Richard B. Olsen**, Zurich (CH); **Michael Stumm**, Toronto (CA)

(73) Assignee: **Oanda Corporation**, Toronto (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 673 days.

(21) Appl. No.: **09/858,610**

(22) Filed: **May 16, 2001**

(65) **Prior Publication Data**

US 2002/0156718 A1 Oct. 24, 2002

Related U.S. Application Data

(60) Provisional application No. 60/274,174, filed on Mar. 8, 2001.

(51) **Int. Cl.**
G06Q 40/00 (2006.01)

(52) **U.S. Cl.** **705/37**; 705/35

(58) **Field of Classification Search** 705/35, 705/37, 1

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,787,402 A * 7/1998 Potter et al. 705/37
5,916,307 A * 6/1999 Piskiel et al. 709/314
6,199,046 B1 * 3/2001 Heinzle et al. 705/1
2001/0044771 A1 * 11/2001 Usher et al. 705/37
2001/0049651 A1 * 12/2001 Selleck 705/37
2001/0056398 A1 * 12/2001 Scheirer 705/38
2002/0023053 A1 * 2/2002 Szoc et al. 705/39
2002/0087455 A1 * 7/2002 Tsagarakis et al. 705/37
2003/0115128 A1 * 6/2003 Lange et al. 705/37
2003/0159109 A1 * 8/2003 Rossmann et al. 715/513

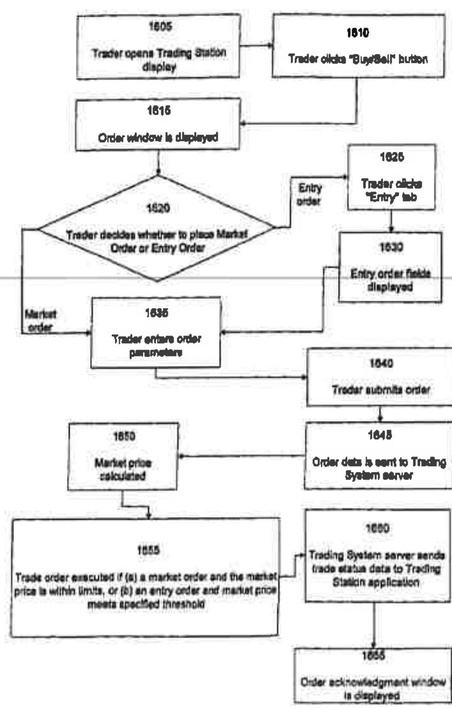
* cited by examiner

Primary Examiner—Alain L. Bashore
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.

11 Claims, 14 Drawing Sheets



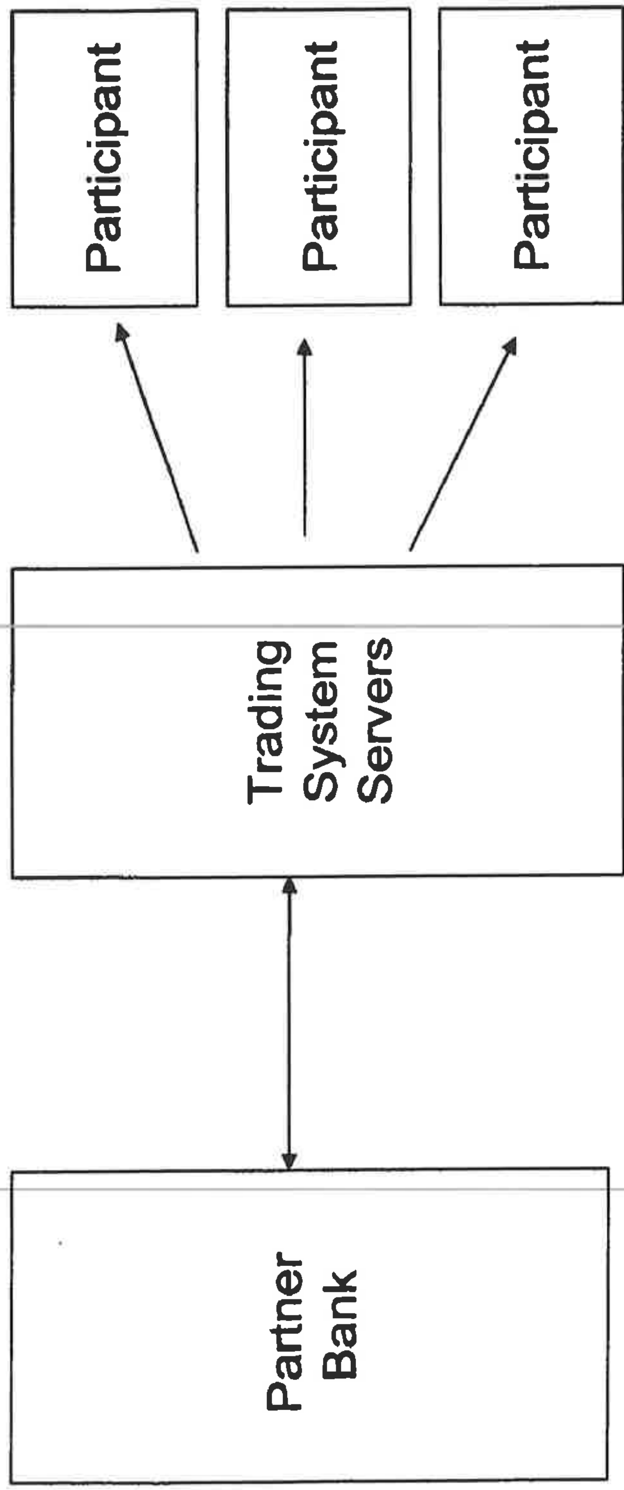

Participant
Participant
Participant
Trading System Servers
Partner Bank

Fig. 1

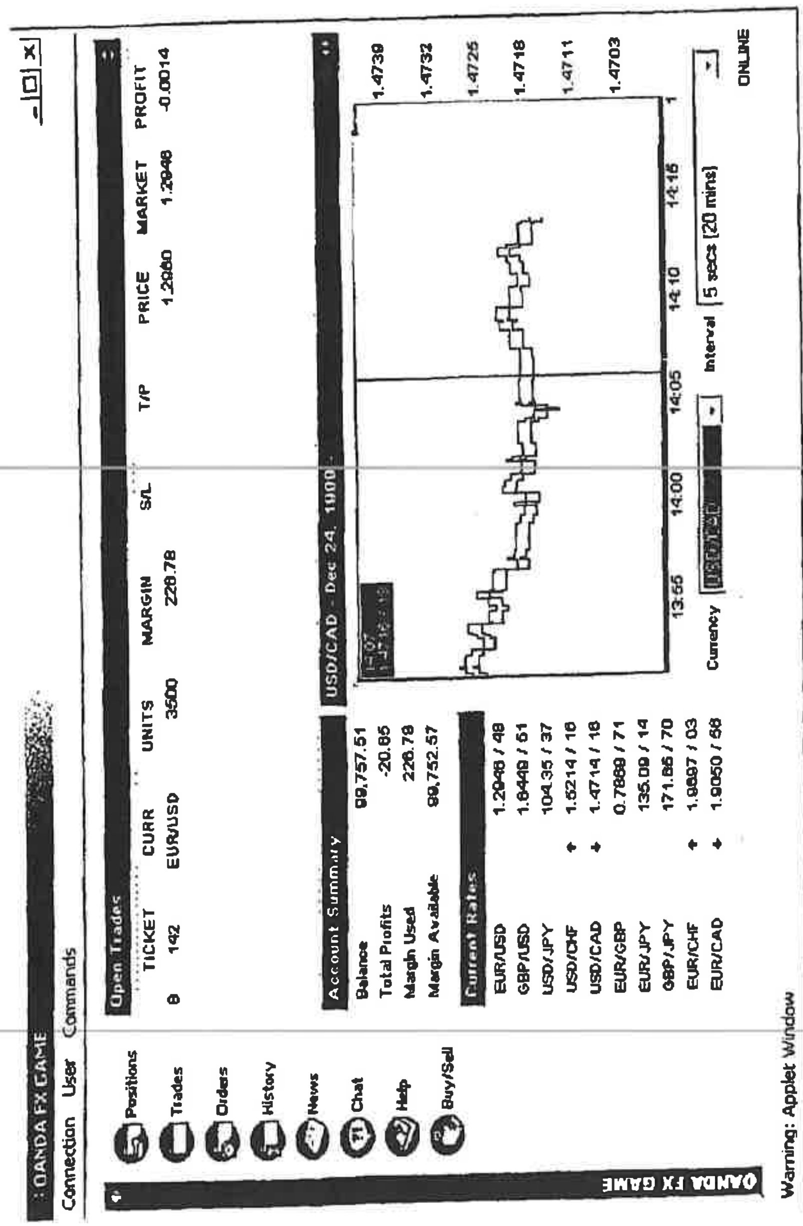

OANDA FX GAME
Connection User Commands
Positions
Trades
Orders
History
News
Chat
Help
Buy/Sell
Open Trades
TICKET CURR UNITS MARGIN S/L T/P PRICE MARKET PROFIT
192 EUR/USD 3500 228.78 1.2960 1.2948 -0.0014
Account Summary
Balance 99,757.51
Total Profits -20.85
Margin Used 228.78
Margin Available 99,752.57
Current Rates
EUR/USD 1.2948 / 48
GBP/USD 1.6449 / 51
USD/JPY 104.35 / 37
USD/CHF 1.5214 / 16
USD/CAD 1.4714 / 18
EUR/GBP 0.7869 / 71
EUR/JPY 135.09 / 14
GBP/JPY 171.86 / 70
EUR/CHF 1.9697 / 03
EUR/CAD 1.9050 / 56
USD/CAD - Dec 24, 1900
1.4739
1.4732
1.4725
1.4718
1.4711
1.4703
13:55
14:00
14:05
14:10
14:15
Currency
USD/CAD
Interval
5 secs (20 mins)
ONLINE
Warning: Applet Window

Fig. 2

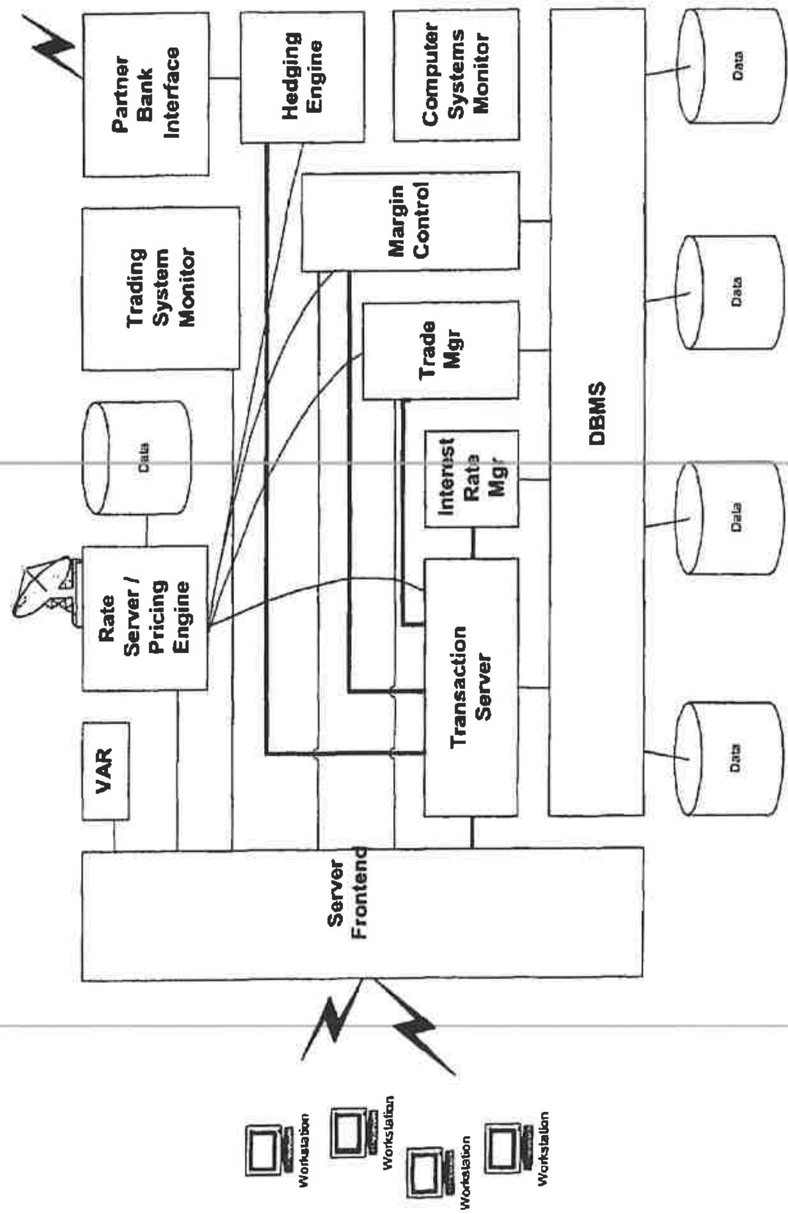

Partner Bank Interface
Hedging Engine
Computer Systems Monitor
Trading System Monitor
Margin Control
Trade Mgr
Interest Rate Mgr
DBMS
Data
Data
Data
Data
Data
Rate Server / Pricing Engine
VAR
Transaction Server
Server Frontend
Workstation
Workstation
Workstation
Workstation

Fig. 3

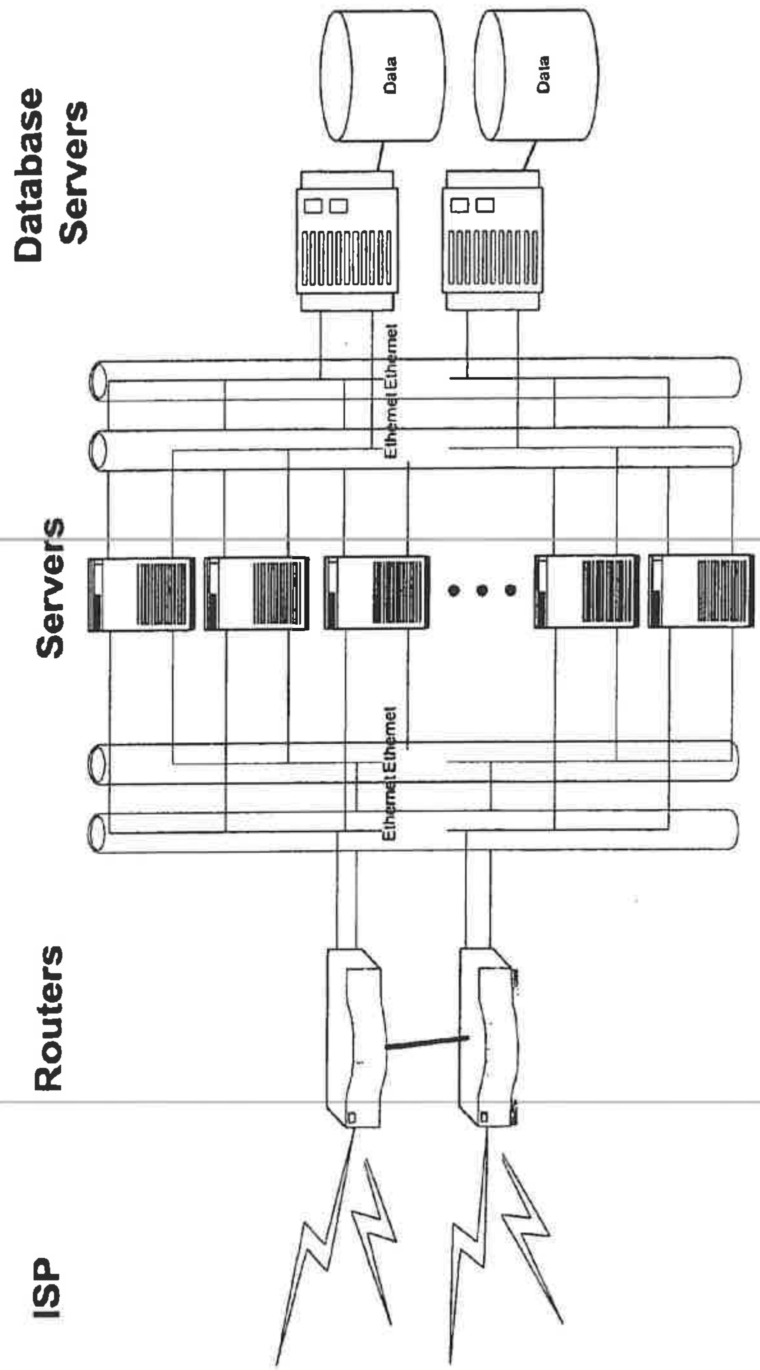

Database Servers
Data
Data
Ethernet Ethernet
Servers
Ethernet Ethernet
Routers
ISP

Fig. 4

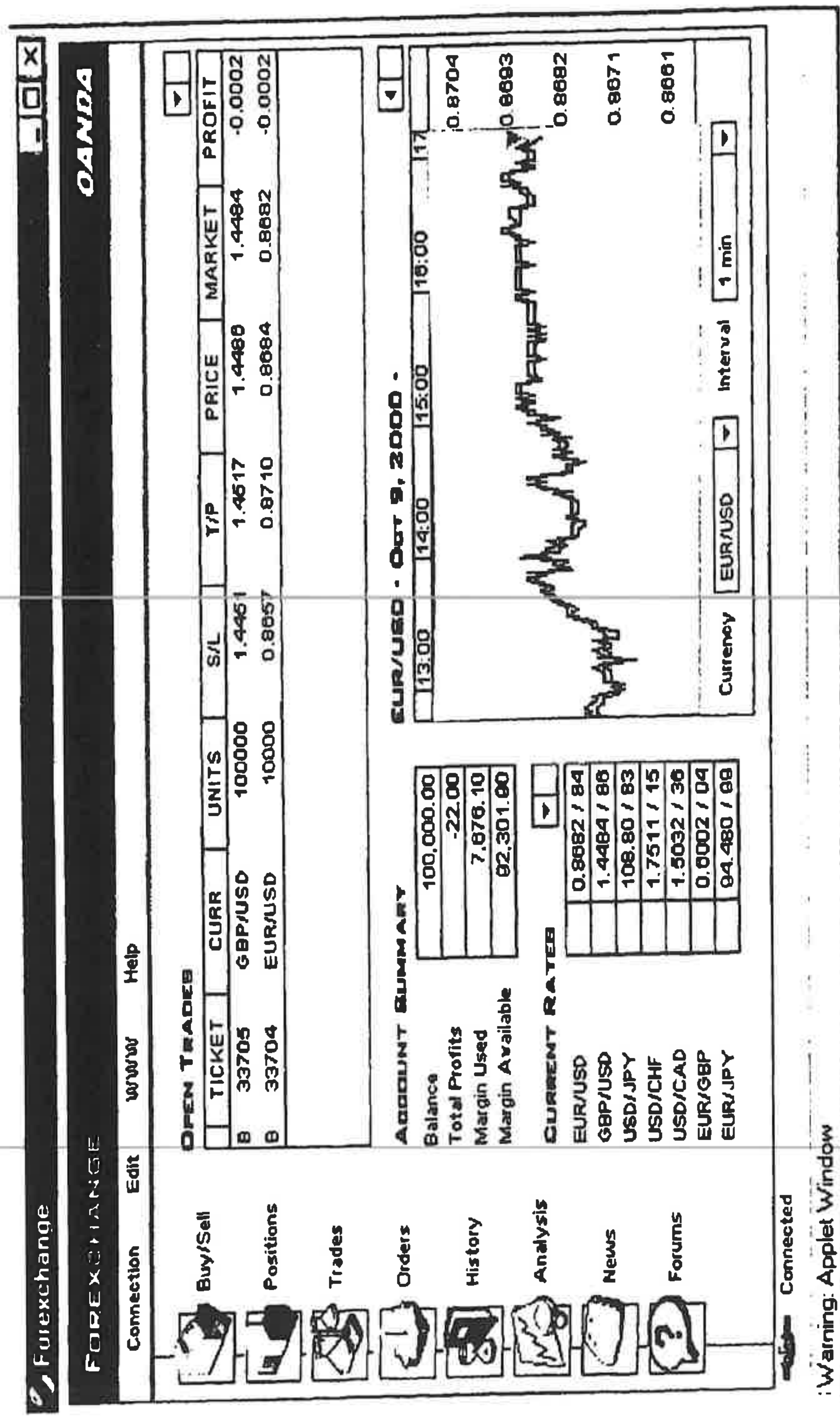

Forexchange
FOREXCHANGE
OANDA
Connection Edit WWW Help
Buy/Sell
Positions
Trades
Orders
History
Analysis
News
Forums
OPEN TRADES
TICKET CURR UNITS S/L T/P PRICE MARKET PROFIT
B 33705 GBP/USD 100000 1.4451 1.4517 1.4488 1.4484 -0.0002
B 33704 EUR/USD 10000 0.8657 0.8710 0.8684 0.8682 -0.0002
ACCOUNT SUMMARY
Balance 100,000.00
Total Profits -22.00
Margin Used 7,676.10
Margin Available 92,301.90
CURRENT RATES
EUR/USD 0.8682 / 84
GBP/USD 1.4484 / 86
USD/JPY 108.80 / 83
USD/CHF 1.7511 / 15
USD/CAD 1.5032 / 36
EUR/GBP 0.6002 / 04
EUR/JPY 94.480 / 89
EUR/USD - OCT 9, 2000 -
13:00
14:00
15:00
16:00
17
0.8704
0.8693
0.8682
0.8671
0.8661
Currency EUR/USD
Interval 1 min
Connected
Warning: Applet Window

Fig. 5

ACCOUNT SUMMARY

Balance	100,000.00
Total Profits	-22.00
Margin Used	7,676.10
Margin Available	92,301.90

Fig. 6

OPEN TRADES

	TICKET	CURR	UNITS	S/L	T/P	PRICE	MARKET	PROFIT
S	33760	USD/CHF	100000			1.7415	1.7413	0.0002
S	33759	EUR/USD	65000	0.8715	0.8668	0.8689	0.8692	-0.0003

Fig. 7

OPEN POSITIONS

	CURR	UNITS	AVG. PRICE	MARKET	PROFIT
S	USD/CHF	100000	1.7415	1.7407	0.0008
S	EUR/USD	65000	0.8689	0.8697	-0.0008

Fig. 8

OPEN ORDERS

	ORDER	CURR	UNITS	S/L	T/P	PRICE	MARKET	DURATION
B	245	EUR/JPY	200000	93.203	93.789	93.512	93.509	Undefined

Fig. 9

RECENT HISTORY

TRANSACTION	TYPE	CURR	UNITS	PRICE	DATE/TIME
58958	Buy Market	EUR/USD	10000	0.8684	Oct 9, 2000, 16:42
58959	Buy Market	GBP/USD	100000	1.4486	Oct 9, 2000, 16:43
57001	Take Profit	GBP/USD	100000	1.4604	Oct 11, 2000, 12:38
57021	Buy Market	EUR/USD	25000	0.8688	Oct 11, 2000, 15:04
57022	Sell Market	EUR/USD	25000	0.8689	Oct 11, 2000, 15:07

Fig. 10

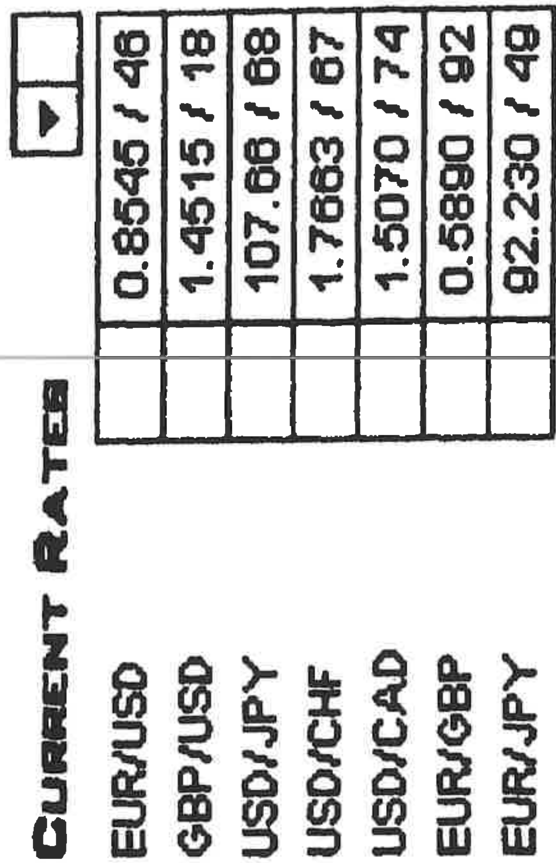

CURRENT RATES
EUR/USD 0.8545 / 46
GBP/USD 1.4515 / 18
USD/JPY 107.66 / 68
USD/CHF 1.7663 / 67
USD/CAD 1.5070 / 74
EUR/GBP 0.5890 / 92
EUR/JPY 92.230 / 49

Fig. 11

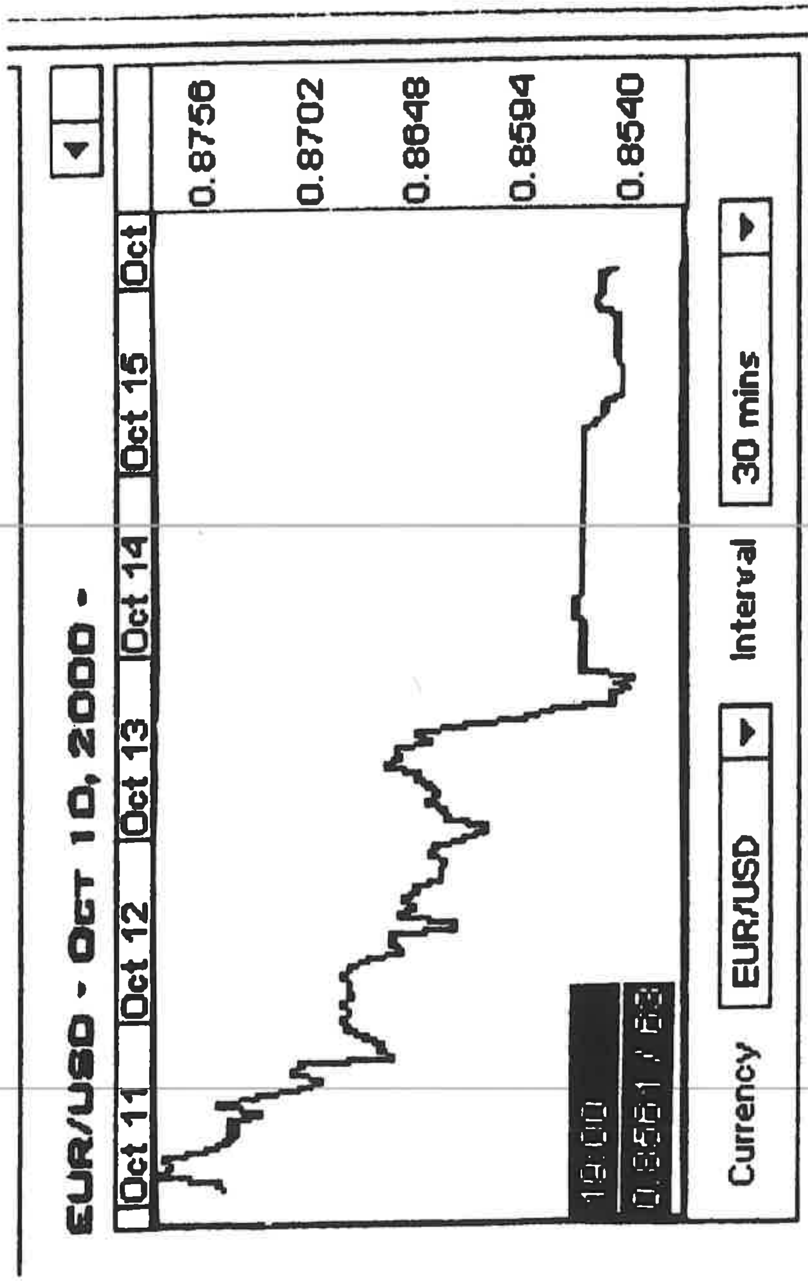

EUR/USD - OCT 10, 2000 -
Oct 11
Oct 12
Oct 13
Oct 14
Oct 15
Oct
0.8756
0.8702
0.8648
0.8594
0.8540
19:00
Currency
EUR/USD
Interval
30 mins

Fig. 12

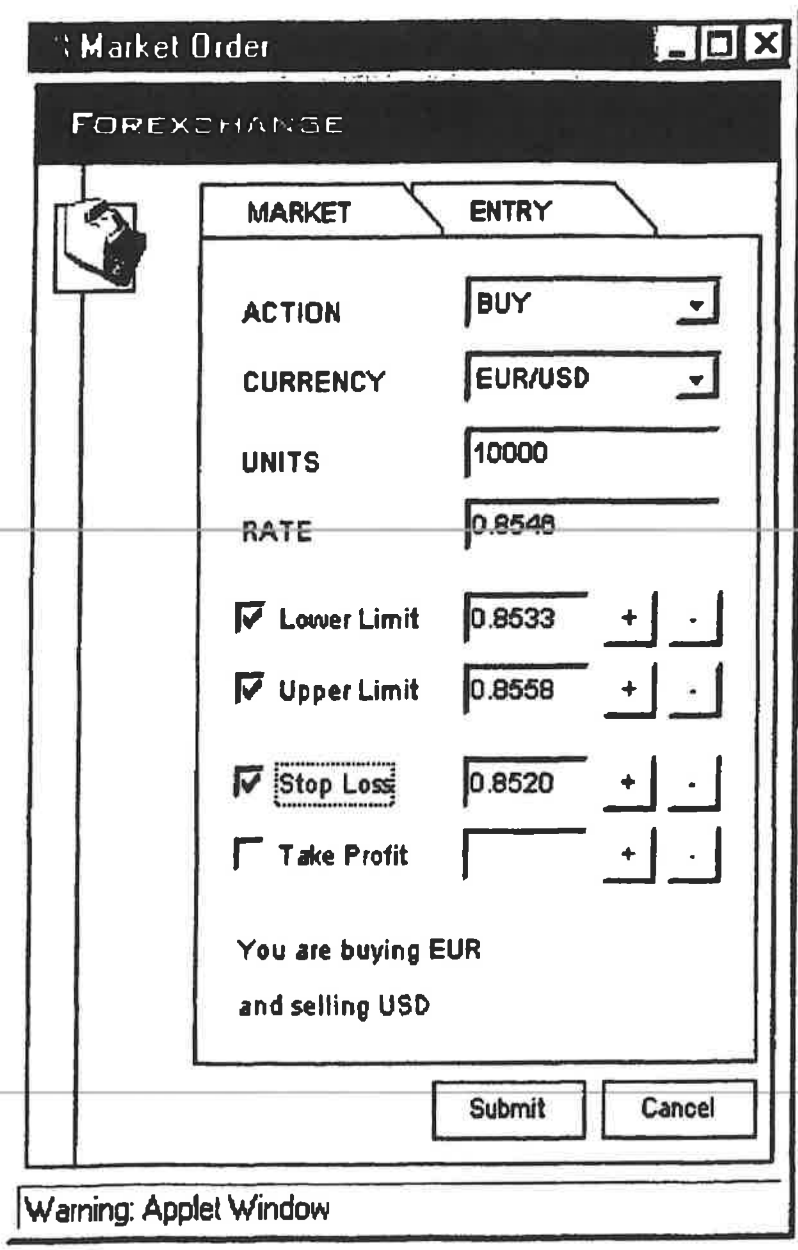

Market Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
10000
RATE
0.8548
Lower Limit
0.8533
Upper Limit
0.8558
Stop Loss
0.8520
Take Profit
+
-
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 13

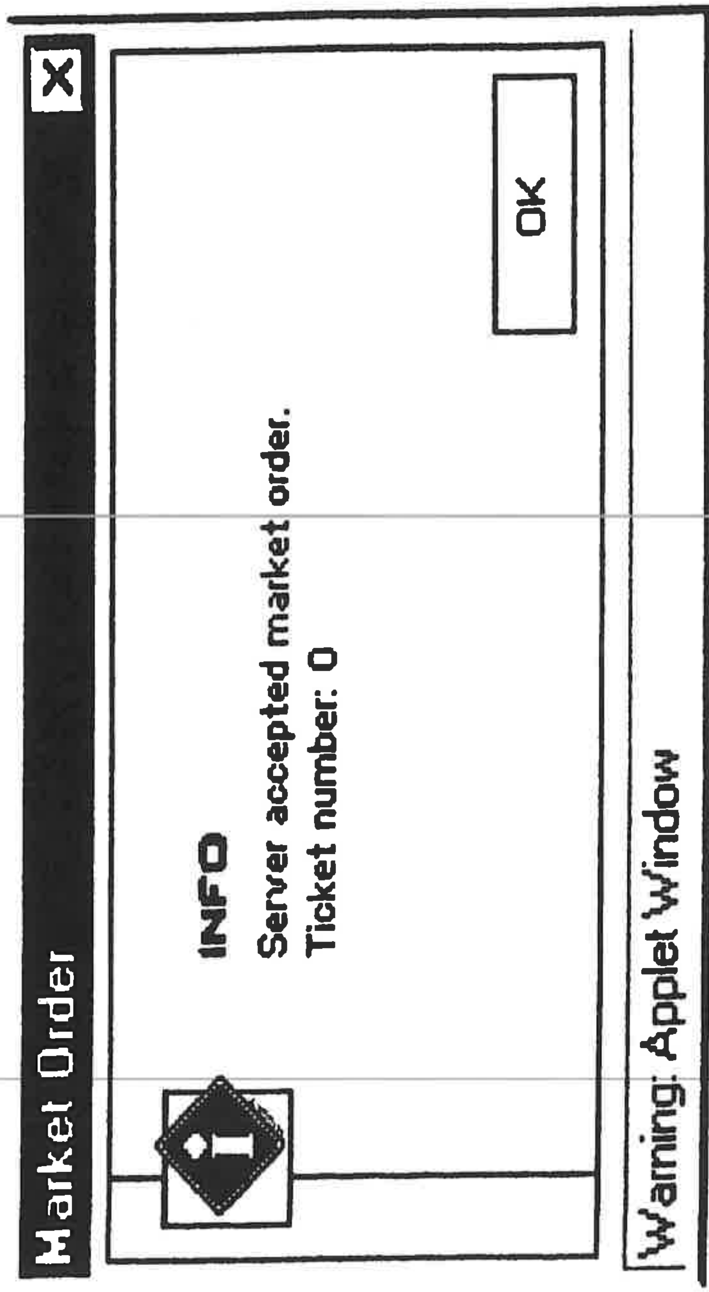

Market Order
INFO
Server accepted market order.
Ticket number: 0
OK
Warning: Applet Window

Fig. 14

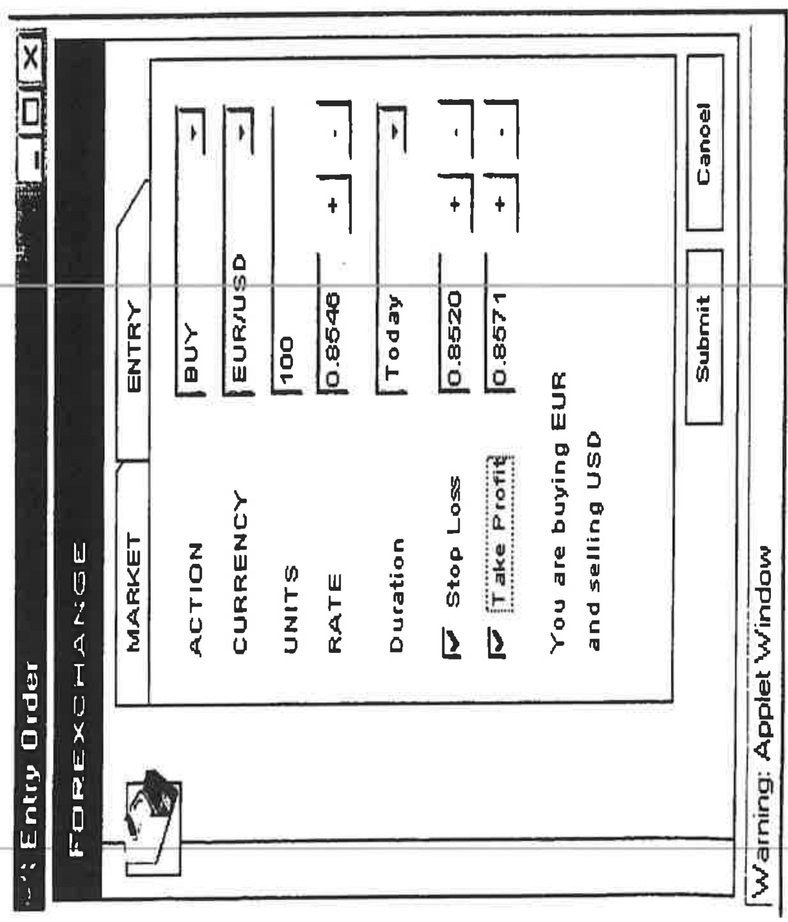

Entry Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
100
RATE
0.8546
Duration
Today
Stop Loss
0.8520
Take Profit
0.8571
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 15

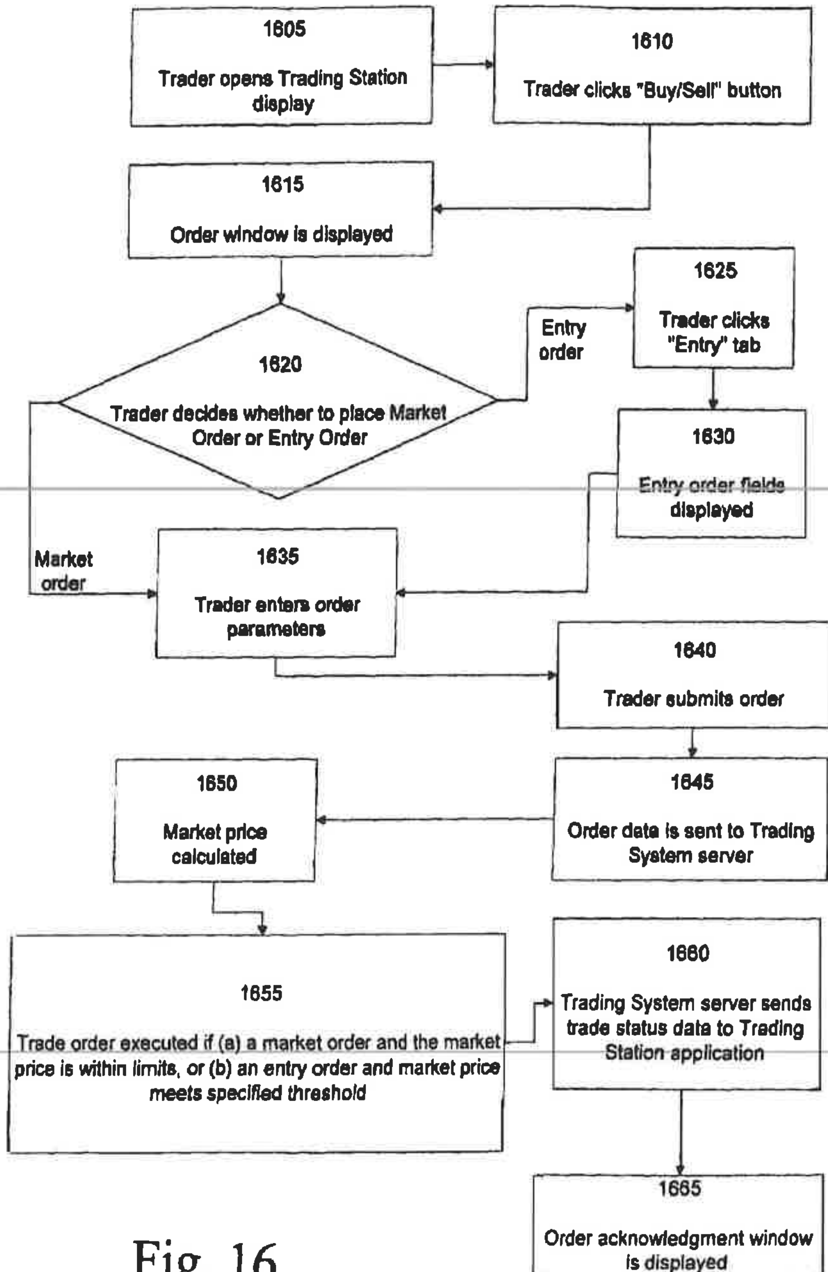

1605
Trader opens Trading Station display
1610
Trader clicks "Buy/Sell" button
1615
Order window is displayed
1620
Trader decides whether to place Market Order or Entry Order
Entry order
1625
Trader clicks "Entry" tab
1630
Entry order fields displayed
Market order
1635
Trader enters order parameters
1640
Trader submits order
1645
Order data is sent to Trading System server
1650
Market price calculated
1655
Trade order executed if (a) a market order and the market price is within limits, or (b) an entry order and market price meets specified threshold
1660
Trading System server sends trade status data to Trading Station application
1665
Order acknowledgment window is displayed

Fig. 16

CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority to U.S. provisional application No. 60/274,174, filed Mar. 8, 2001, and incorporates the entire contents thereof herein by reference.

FIELD OF THE INVENTION

The present invention is related to currency trading; more particularly, the invention is related to trading currency over a computer network.

BACKGROUND

In a traditional on-line currency market, a trade occurs through three steps: (1) the trader specifies to a dealer the currency pair and the amount that he would to trade (but does not specify whether he would like to buy or sell); (2) the dealer specifies to the trader both a bid and an ask price and gives the trader several seconds to respond (the dealer not knowing whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or specifies whether he is buying or selling (his response must occur within a time frame of a few seconds).

But performing such a three-way handshake over the Internet is somewhat impractical because of Internet delays: the trader might not actually have a few seconds to respond before the dealer withdraws the offer. Thus, there is a need for a system and method of on-line currency trading that is based on a trading model that is superior to the three-way handshake described above.

Another problem is that many corporations have firewalls that restrict access to the corporate network, and that typically restrict access to the Internet (and to well-known services such as email, the World Wide Web, etc.) from within the corporation. This inhibits the ability of on-line trading systems to access information from and transfer information to users behind corporate firewalls.

SUMMARY

The present invention overcomes the above-described and other disadvantages of previous currency trading systems and methods. In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. Each of these components is described in detail below in the Detailed Description section.

In another aspect, the present invention comprises methods for trading currency over a computer network. In one embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises limits within which the currency trade will be acceptable to the end user; (c) calculating a market exchange rate for the currency trade order; and (d) executing the order if the market exchange rate is within the specified limits.

In another embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises a threshold exchange rate; (c) calculating a market exchange rate for the received currency trade order; and (d) executing the order (1) if the market exchange rate is or goes above the threshold exchange rate and the order is a sell order, and (2) if the market exchange rate is or goes below the threshold exchange rate and the order is a buy order.

In a further embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises limits within which the currency trade will be acceptable; and (c) if a market exchange rate is within the specified limits, receiving information from the trading system server indicating that the currency trade order was executed.

In another embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises a threshold exchange rate; and (c) if (1) the applicable market exchange rate is or goes above the threshold exchange rate and the order is a sell order, or (2) the applicable market exchange rate is or becomes below the threshold exchange rate and the order is a buy order, receiving information from the trading system server indicating that the currency trade order was executed.

In another aspect, the present invention comprises software for currency trading over a computer network. In one embodiment, preferred software comprises: (a) software for receiving data over a computer network from a trading system server; (b) software for displaying a first graphical user interface display that: (i) displays continuously updated currency exchange rates in real-time based on data received from the trading system server; and (ii) displays action buttons, including a buy/sell button; (c) software for displaying, in response to a user clicking the buy/sell action button, a buy/sell window display that: (i) comprises trade order parameter fields; and (ii) accepts trade order data entered into the trade order parameter fields by a user; and (d) software for transmitting said trade order data to said trading system server over said computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts parties involved in a preferred embodiment.

FIG. **2** depicts a graphical user interface of a preferred embodiment.

FIG. **3** depicts modules of a preferred trading system server.

FIG. **4** depicts hardware components of a preferred embodiment.

FIG. **5** depicts a graphical user interface of a preferred embodiment.

FIG. **6** depicts an account summary table display.

FIG. **7** depicts an open trades table display.

FIG. **8** depicts an open positions table display.

FIG. **9** depicts an open orders table display.

FIG. **10** depicts a transactions table display.

FIG. **11** depicts a currency rates table display.

FIG. **12** depicts a currency exchange rate graph display.

FIG. **13** depicts a buy/sell pop-up window display.

FIG. **14** depicts an acknowledgment window display.

FIG. **15** depicts an entry order display.

FIG. **16** depicts steps of a method of a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The Currency Trading System of a preferred embodiment (hereinafter "Trading System") of the present invention allows traders to trade currencies over a computer network. Preferably, this computer network is the Internet, and the subsequent description herein is primarily in terms of the Internet. However, those skilled in the art will recognize that the following description also applies to other computer networks. Traders interface to the system using ordinary Web browsers running feature-rich Java applets; they obtain real-time data feeds of current exchange rates, they can analyze past exchange rates using graphical tools, they can review their current portfolio and past trades, and they can place buy and sell orders in the real-time market. Businesses interface to the system using an API. Innovative features that set the Trading System apart from the competition are: (i) extremely low spreads on the order of a few basis points, (ii) the ability to trade very small amounts as low as $1, and (iii) 24 hour a day, 7 days a week availability. This system has the potential to revolutionize the way currency trading is done and to open up currency trading to a new, large market segment of investors and speculators for whom currency trading is not feasible today. Moreover, it allows businesses to address their currency exchange requirements in the most cost-effective and efficient way.

A description of the preferred server infrastructure used in the Trading System follows. We first give a brief introduction of the system as a whole.

The Trading System involves three components (see FIG. **1**): (1) traders that are distributed around the world; (2) Trading System servers; and (3) "Partners" consisting of the financial institution(s) through which real currency exchange trades are executed, and from which real-time data feeds are obtained.

Traders communicate with Trading System servers over a secure, encrypted Internet connection to review their accounts, to monitor currency exchange market conditions, or to initiate currency exchange trades. The Trading System servers are preferably connected to the partners' back-office systems, using direct, private lines.

A trader trades with the Trading System similar to the way she currently trades with a broker, except that the trading is over the Web, occurs 24 hours a day, 7 days a week, and allows very small trades with very low spreads. Moreover, an initial deposit, which may be as low as $20, can be charged to a credit card to get started. Alternatively, the trader can transfer initial funds directly to the Partner bank to be credited to her Trading System account.

The end user interface to the Trading System is a Web page that can be displayed on any standard Java-enabled browser. The Web page (one version is shown in FIG. **2**; a second, preferred version is shown in FIG. **5**) depicts a summary of the trader's current position, recent trading activities, profit/loss performance of the portfolio, and a graphical display of recent past performance of the currencies the trader has positions in, as well as real-time exchange rates.

As discussed above, in a traditional on-line currency market a trade occurs through three steps: (1) a trader specifies a currency pair and an amount he would like to trade (and does not specify whether he would like to buy or sell); (2) a dealer provides both a bid and an ask price to the trader and gives the trader several seconds to respond (the dealer does not know whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or accepts the offer and specifies whether he is buying or selling, within the response period set by the dealer.

In the Internet domain, this type of three-way handshake is problematic. The timing constraints are difficult to implement because of frequent delays in transmission over the Internet. To overcome this problem, the present invention uses a "two-way handshake," in which: (1) a trader specifies in her trade order: (a) a currency pair; (b) a desired amount to trade; (c) whether she wishes to buy or sell; and (optionally) (d) upper and lower limits on an acceptable exchange rate; and (2) a dealer (in this case, a preferred Trading System) executes the trade using the most current "market rates" (as calculated by the system). However, the system only executes the order if the calculated market rate lies above any specified lower limit and below any specified upper limit. Note that this method does not require the use of timing constraints, and thus avoids the Internet-implementation disadvantage of previous methods.

In the present invention, trades can be initiated by the trader at the push of a button. A trading request form pops up with fields properly initialized so as to minimize the number of keystrokes required. A trader may elect to execute a trade right away, in which case the buyer of a currency will buy at the current exchange rate market offer price. Conversely, a trader can sell a currency at the current bid price. A range of automatic trading options is available, including setting bid/offer prices with a certain duration and "all-or-nothing" rules. Furthermore, the trader can limit her risks by placing stop-loss orders that are executed automatically. Similarly, she can lock in profits, by issuing take-profit orders.

All communication between the trader's browser and the Trading System server occurs through the Internet, preferably using the strongest available encryption (e.g., 128 bit keys). Moreover, the trader must authenticate herself using private passwords or certification keys obtained from certification authorities, such as Verisign or Entrust.

A request for a market trade preferably proceeds as follows: the trader, at a push of a button, obtains a trade order ticket in a popup window on the browser with key fields pre-initialized (see FIG. **13**). When the trade order is issued, again by the push of a button, a message is sent to a Trading System server, where the market price is calculated based on such factors as market data, size of the transaction, time of day, the Trading System's current exposure, and predictions on market direction. The trade order is executed using this market price. (The trader can specify limits, so that the trade occurs only if the price falls within these limits.) As such, the Trading System operates as a market maker. A message is then sent back to the trader with specific trade details, which is displayed in a popup window (see FIG. **14**) on the trader's browser together with a transaction id (for future reference). Moreover, an open orders table (see FIG. **9**) and current portfolio summary table (not shown) is updated to reflect the change.

Alternatively, the trader can issue in a similar manner an entry order (see FIG. **15**) that requests a trade be made when the currency exchange rate reaches a specified threshold. The trader may specify how long the entry order will be valid.

Referring to the attached figures, a preferred embodiment comprises a method of trading that in turn comprises the following steps (see FIG. **16**): At step **1605**, a trader desiring to trade opens a Trading Station display, and at step **1610** clicks a "Buy/Sell" button **510** on the Trading Station display (see FIG. **5**). At step **1615** an order window is displayed (see FIG. **13**). At step **1620** the trader decides whether to place a market order or an entry order. If a market order, then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **13**). If an entry order, then the trader proceeds to step **1625** and clicks an "Entry" tab **1320** (see FIG. **13**). At step **1630** entry order fields are displayed (see FIG. **15**). Then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **15**).

Once order parameters are entered at step **1635**, the trader submits the order by clicking a "Submit" button **1310** (see FIG. **13**) if the order is a market order, or clicking a "Submit" button **1510** (see FIG. **15**) if the order is an entry order. At step **1645** data describing the order is sent by the Trading Station application to a Trading System server, where the data is stored. At step **1650** a current market price for the currency the trader desires to purchase is calculated. At step **1655** the trader's order is executed if (a) the trader's order is a market order and the calculated market price is within the limits set by the trader in the market order form at step **1635**; or (b) order is an entry order and the calculated market price meets the threshold(s) specified in the Entry order form at step **1635**.

At step **1660** the Trading System server sends trade status data to the trader's Trading Station application. This data includes an indication that the order has been executed, if that is the case, and at any rate includes an indication that the order has been received. At step **1665**, the Trading Station application displays an order acknowledgment window (see FIG. **14**) that displays order status information.

Over time, the Trading System will accumulate an imbalance in its currency portfolio and, at times, it will need to neutralize its risk exposure to adverse currency fluctuations. The Trading System Pricing Engine can influence its exposure by setting its price quotes accordingly. Moreover, it can close out its positions periodically or take hedging positions by executing larger trades through its Partners. Preferably the Trading System's positions are managed based on state-of-the-art trading models. Preferred trading models are described in U.S. patent application Ser. No. 09/855,633, filed May 14, 2001, the contents of which are incorporated herein by reference, as well as U.S. provisional application No. 60/274,174, filed Mar. 8, 2001.

The Trading System servers preferably operate 24 hours a day, 7 days a week. These servers interface with the traders over the Internet on the one hand and on the other hand with the Partner's back-office operations. Using standard, state-of-the-art database technology, it maintains the accounts of all traders and executes trades issued by the traders. The Trading System thus plays the role of a market maker in that it internally aggregates all trades and only occasionally balances its internal positions by trading larger sums through the Partner. These larger trades are issued to the Partner in an automated way. The Trading System also takes hedging positions so as to minimize risks on the unbalanced portions of the traders' account aggregates.

Partner's Role

The Partner maintains all actual funds. It is the source and target of all fund transfers to and from customers; it maintains the aggregate accounts; and it executes all trades issued automatically by the Trading System servers. From a legal point of view, all funds must be maintained in money market instruments. Hence, the Partner will maintain a Long and a Short money market fund for each currency supported.

Overview of Currency Trading System Server Internal Design

The Trading System server architecture is designed to be:

(1) modular, in order to: (a) simplify development (time-to-market); (b) easily allow extensions and modifications; (c) ensure correctness and robustness, and (d) keep it maintainable;

(2) efficient, in order to provide fast response times to online users and minimize the computational and networking resources required to support the service;

(3) scalable, in order to support (with suitable computing infrastructure) a large number of online users and high transaction volumes; and

(4) fault resilient, so that any individual failure of a computing node or network connection does not interrupt service.

The Trading System server software preferably runs exclusively on Unix platforms, and is composed of the following modules, each with a distinct set of responsibilities (see FIG. **3**):

(1) Database (DBMS) **310**. This is the heart of the server. It keeps track of customer profile information, all customer accounts and all transactions, and ensures that atomicity, consistency, independence, and durability ("ACID") properties are maintained. The database is the reference point for all information kept by the system.

The database is preferably a standard commercially-available SQL database, configured for full replication for reliability, availability, and improved performance. The preferred embodiment is based on IBM's DB-2 product line, but Oracle, for example, could also easily be used.

(2) Server Front-end **315**. The server front-end **315** is responsible for all communication with the Web-based clients. It supports both persistent and non-persistent connections to the traders. The persistent connects are used primarily for periodic (i.e., every few seconds) transmission of the latest currency rates so that the traders can update the currency graphs and tables in real-time. Using persistent connections significantly reduces protocol processing overhead and reduces network bandwidth requirements. Non-persistent connections are used for all transaction-oriented requests, such as orders, transaction history requests, logins, etc. All transaction-oriented communication between the trader browsers and the Server Front-end occurs fully encrypted, while rate information is transmitted in unencrypted form for efficiency reasons.

Traders preferably communicate with the server using a request-response type of protocol. The Server Front-end **315** interprets each request it receives and, for each, takes appropriate action. For login requests, it sets up appropriate data structures so that all future requests can be serviced in the most efficient way. It also sets up encryption keys for secure communication, and logs the start of a new session with the Transaction Server. For rate requests, it returns the requested rates it obtains from the rate server. For orders, it executes the orders by issuing appropriate requests to the transaction server after checking the margin requirements, the availability of funds, and using rates as determined by the pricing engine. For stop-loss/take-profit and fixed-price orders (that may get executed in the future), the Trade Manager **365** is also informed. For each trade that gets executed, the Hedging Engine **340** and Margin Control **350** modules are informed, so that they always have an up-to-date snapshot of the state. For transaction history, the appropriate information is returned to the client after obtaining it from the Transaction Server **355**.

The Server Front-end **315** also encapsulates a standard Web server (a la Apache), that services other trader requests that entail formatted text; this includes all of the Help pages, large transaction history requests, and server monitoring information. The Server Front-end **315** also acts as a Firewall.

Internally, the Server Front-end **315** is structured as a set of threads that service one request after another. The threads allow concurrent request servicing so that many requests can be serviced in parallel.

(3) Rate Server/Pricing Engine **325**. The Rate Server obtains currency exchange rate information from a variety of external rate sources and stores it locally. The Pricing Engine computes the currency exchange rates that the traders see and that are used for trading. These are computed from the currency exchange rates obtained from the external rate sources, the directional movement and volatility of the market, the current Trading System exposure and a number of other parameters. The computed rates are made available to the other modules of the system, and are also stored on persistent media. Various methods of calculating such rates are known to those skilled in the art. A preferred method is described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al.

Traders can request historical rate data so that they can graphically display the movements of any pair of currencies. The Rate Server serves such requests and preferably has several years of currency exchange rates available for this purpose.

For fast response time, the Rate Server caches in memory all of the frequently and recently requested rates so as to minimize the number of disk accesses required.

(4) VAR (Value at Risk) Server **320**. This server obtains and serves Value at Risk information. Various methods of calculating VAR are known in the art. A preferred method is disclosed in U.S. Provisional Patent Application No. 60/200,742, filed May 1, 2000, to Müller.

(5) Transaction Server **355**. This server encapsulates all transaction functionality and communicates the transactions to the Database **310** server (which runs on a separate host) after validating the transactions. The Transaction Server **355** also updates all other modules that need to be informed of new transactions. Finally, the Transaction Server **355** informs the currently online traders when a transaction (that may have been issued by a stop-loss, take-profit, or limit order daemon or by the Margin Control Manager) takes place.

(6) Interest Rate Manager **360**. The Interest Rate Manager **360** periodically (for example, every few minutes, every few seconds, or tick-by-tick) goes through all trader accounts to compute the interest rate due or owed based on the instruments currently in the portfolio, each resulting in a transaction of the trader account. The portfolio information is obtained through the Transaction Server **355**. The interest rates used are obtained from external sources, and the history of interest rates are stored on persistent storage. Because real-time (or near real-time) information is used, the Interest Rate Manager is capable of calculating, paying out, and collecting interest by the second. Interest calculation formulas are known to those skilled in the art, and any appropriate formula can be used in the Interest Rate Manager without departing from the scope of the invention. An example is the formula

$$A = P\left(1 + \frac{r}{m}\right)^{mt}$$

where P is the principal, r is the annual interest rate, t is the time (in years) over which interest is earned, m is the number of times per year that interest is compounded, and A is the amount owed (principal plus interest). The interest earned during time t is A−P. Thus, for example, if the annual interest rate is 3%, and the interest is compounded daily, then the interest I_k earned over each time period $T_k = t_k - t_{k-1}$, where each t_k is a clock-time (i.e., a particular day-hour-minute-second-fraction-of-a-second) to the nearest second (thus T_k is in seconds), is calculated according to the formula $I_k = A_k - P_k$, where P_k is the principal (the amount earning interest, not the "original" principal) at time t_{k-1}, and

$$A_k = P_k\left(1 + \frac{0.03}{365}\right)^{\frac{365 T_k}{31,536,000}}.$$

Since there are 31,536,000 seconds per year,

$\frac{T_k}{31,536,000}$ is

the time in years over which the interest is being calculated. Similar formulas can be used when t_k is given to the nearest tenth, hundredth, or other fraction of a second. If interest is compounded continuously, those skilled in the art will recognize how to apply the well-known formula $A = Pe^{rt}$ appropriately. Thus, to calculate interest on a tick-by-tick basis, the above formulas can be used, with T_k representing time between ticks.

(7) Trade Manager **365**. The Trade Manager **365** continuously checks whether a trade should be executed on behalf of a trader, and if so executes the trade by interacting with the transaction server. The Trade Manager **365** consists of multiple subcomponents: (a) a stop-loss daemon continuously checks to see whether stop-loss orders should be executed and, if so, executes them through the Transaction Server **355**; (b) a take-profit daemon continuously checks to see whether take-profit orders should be executed and, if so, executes them through the Transaction Server **355**; and (c) a limit-order daemon continuously checks to see whether a limit order should be executed and, if so, executes it through the Transaction Server **355**.

The daemons continuously monitor the current rates to determine whether action is required. Moreover, each of the daemons caches in memory all of the orders that it may need to execute. They keep the orders suitably sorted so that they can take fast action when necessary; for example, the stop-loss daemon sorts the orders in descending order of stop-loss price, the take-profit in ascending order of take-profit price.

(8) Margin Control Manager **350**. This module continuously monitors the margin requirements of all trader accounts. When necessary, the Margin Control Manager **350** will liquidate some (or all) of a trader's holdings. It caches in memory all of the information necessary for this computation, sorted in decreasing order of risk, so that it can take

swift action when necessary. Holdings are liquidated through the Transaction Server **355**, when necessary.

(9) Trading System Monitor **330**. This module continuously monitors the current state of the Trading System. Among others, state parameters include: (a) current Trading System currency positions; (b) current margin situation; (c) a summary of stop-loss, take-profit, and limit orders that exist; (d) the number of currently online users; (e) the number, size and type of trades executed per second; and (f) a summary of the account positions held by the users.

This information is made available (a) to the Pricing Engine **325** (where it is used to set the currency exchange rates made available to the traders), (b) to the Hedging Engine **340** so that it can determine when to issue trades with the Partner Bankend Bank, and (c) to system operators and Trading System financial engineers in real time via a feature-rich Web interface. Moreover, this information is logged on persistent storage for later, off-line analysis.

(10) Hedging Engine **340**. This module continuously monitors current Trading System currency positions, the positions held in the trader accounts, recent trading activity, and the market direction and volatility to determine when to issue a trade with the backend Partner Bank. Various methods of performing such calculations are known to those skilled in the art. The Hedging Engine **340** preferably uses the hedging tool described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al., the contents of which are incorporated herein by reference.

(11) Partner Bank Interface **335**. This module communicates directly with the backend Partner Bank to issue trades and obtain account information.

(12) Computer Systems Monitor **345**. This module continuously monitors the operation and state of the computer systems on which the Trading System is running. Besides error conditions, such metrics as memory, processor, disk, and network utilization; paging activity; the number of packets sent over the various networks; the number of transactions; and the number of processes and threads are of interest. This information is made available to system operators in real time via a feature-rich Web interface and local consoles. In addition, it is stored on persistent storage for later, off-line analysis.

The Server modules described above are structured so that they can run independently as separate processes that can be independently mapped onto an arbitrary computer within a cluster. Moreover, each of the modules can run in replicated form, providing both fault tolerance and increased throughput.

Preferred Physical Organization of the Trading System Server

A Trading System Server of a preferred embodiment runs on a hardware base consisting of a cluster of hosts and disk farms connected by networks. All of the hardware components are preferably replicated for fault tolerance, as depicted in FIG. **4**.

The cluster is connected to multiple ISPs so that if one ISP goes down, traders can still communicate with the server. The ISPs are connected to the Server through a pair of routers **410** that monitor each other; if one of them goes down, then the other will automatically take over.

For security reasons, the Database **310** is on a separate back-end network; this way, it is not connected directly to the Internet and can only be accessed by the Transaction Server **355**. The Database **310** is setup in a dual configuration, so that the system can continue operating with a single database failure. All disks are mirrored, again, so that any single disk failure will not result in a loss of data.

All of the other server processes run on a cluster of servers **420**, connected to the Internet routers **410** on the one side, and connected to the backend database **430** on the other side. A virtually unlimited number of servers can be used in the cluster, allowing the system to scale up to support a large number of users. The servers can be partitioned by functionality, allowing specialized servers to be used, optimized for the particular functionality. For example, the Rate Servers **325** need minimal CPU power, and only a limited amount of memory. They also can be replicated easily without the introduction of any complexity or overhead. Hence, smaller, less costly hardware can be used for this purpose.

After login, traders typically communicate with a particular server in "sessions" for performance reasons. Using sessions improves cache locality, resulting in far fewer database accesses, and it allows the cost of creating session encryption keys to be amortized over many communication actions. For load balancing purposes, the trader software is directed to henceforth communicate with the least loaded server at the time when the trader first logs in. In case of severe load imbalances, individual traders are redirected to new, less loaded servers. If any of the servers crashes, then the client software that was communicating with the crashed server will detect the failure and automatically (transparently to the user) go through a new login procedure.

User Interface Description

Overview

The following is a description of a preferred user interface of a preferred Trading System. The main user interface display is called a "Trading Station," and it is used for all interactions with the trading system by a trader, such as analyzing changes in currency exchange rates, reviewing the trader's current currency positions, reviewing the trader's past transactions, or issuing buy and sell requests. The key features of the Trading Station are that: (1) it runs on any of the popular Web browsers connected to the Internet; (2) it displays continuously updated currency exchange rates in real-time; (3) it displays all pertinent information in one window; and (4) all interaction with the server occurs over fully-encrypted Internet connections.

System Requirements

The User Interface is preferably implemented in Java so as to run on any browser with JDK1.2 support, which includes all Netscape Navigators versions 4.2 and up as well as Microsoft's Internet Explorer versions 5.0 and up. The Trading Station is preferably supported for Windows 95, Windows 98, Windows 2000, Windows NT, Linux, Sun Solaris, and other Unix-based operating systems.

If operated from behind a firewall, then the Trading Station may operate properly only if the firewall allows HTTP requests to Port **90**. Many corporations have firewalls that restrict access to the corporate network to well-known services such as email. Typically this restriction is accomplished by restricting the ports that may be used. For example, Port **80** is typically used for http (Web-based) traffic. Some firewalls inspect traffic going through Port **80** to ensure that the port is being used only for Web-based traffic. This is problematic for trading systems that do not use http messages—it causes users behind corporate firewalls to be inaccessible. However, a preferred embodiment of the present invention overcomes this obstacle by prefacing Trading System messages with standard http headers to make them appear to be http requests and responses, even though they are not.

Log In Procedure

In order to log in, a trader must be a registered user. Registering is preferably free and can be accomplished by clicking on a "new users" link on a login page. Logging in requires a user to provide a user-ID and password. If a trader forgets her password, she can click on the "forgot your password" link and fill in the information requested; her password will be then be emailed to her.

If user-ID and password are entered correctly, a small window appears indicating that the Trading Station is being loaded. After a short time, a larger window appears with the Trading Station Graphical User Interface shown in FIG. **5**. Once the Trading Station is properly loaded, the contents of the small window is changed to include a number of useful links. It is important that this small window not be closed while the Trading Station is to remain in operation, although it may be minimized so as not to be in the trader's way. (This small window is necessary due to the limitations of typical Java implementations on most browsers that would otherwise not allow a trader to continue browsing the Web while the User Interface is active.)

Main Window of Trading Station

The Trading Station user interface is shown in FIG. **5**. It can be resized to a convenient size, by using the standard resizing mechanisms supported by the trader's operating system's windowing system.

The Trading Station is preferably partitioned into a number of components that each serve a different purpose:

(A) Action buttons: a vertical panel located on the left hand side of the Trading Station contains a set of action buttons that allow a user to perform the most common operations.

(B) Menus: a set of pull-down menus across the top allows a user to invoke additional operations.

(C) Account summary: an area in the middle of the Trading Station that gives a summary of the user's account.

(D) Table: an area located across the top of the Trading Station that is used to display various information in tabular format. The information displayed depends on the most recently clicked Action Button. It might display currently held instruments, current open positions, or a history of recent transactions.

(E) Currency rates: an area at the bottom left that displays various currency rates. These rates are continuously updated in real time.

(F) Graph: located at the bottom-right corner, graphs display currency rates over time. The graphs are also updated in real-time as new rates become available.

Subsequent sections describe each of these components in detail.

Action Buttons

The Trading Station preferably has the following action buttons in a panel on the left side. Clicking the appropriate button will invoke the described operation:

Buy/Sell: Pops up a Buy/Sell window, from which a trader can issue trade requests. (See the description of the buy/sell window (FIG. **13**) for more information.)

Positions: Displays the currently open positions in a table. (See the description of the Open Positions Table (FIG. **8**) for the contents of the table.)

Trades: Displays the currently open trades in a table. (See the description of the Open Trades Table (FIG. **7**) for the contents of the table.)

Orders: Displays the open orders (that may be executed some time in the future) in a table. (See the description of the Open Orders Table (FIG. **9**) for the contents of the table.)

History: Displays a recent history of the trader's transactions in the table. (See the description of the History Table (FIG. **10**) for contents of the table.)

Analysis: Pops up a new browser window with access to a number of analysis tools that might help in making trading decisions.

News: Pops up a new browser window with the latest currency news.

Forums: Pops up a new browser window with access to a number of forums (sometimes known as newsgroups) that allow a trader to participate in discussions with other traders and currency trading experts.

Pull-down Menus

There are preferably 5 pull-down menus (not shown), each offering different operations or services:

Connection Menu: (1) Disconnect: disconnects the Trading Station from the Trading System server. The Trading Station will remain open, but currency rates will no longer be updated, and transactions will not be possible. (2) Connect: connect the Trading Station to the server, so the trader is back on line. (3) Quit: quit and exit this application.

Account Menu: (1) Transaction history: pop up a new browser window to display an extensive list of all transactions that occurred on a trader's account. See the Transaction History section (relating to FIG. **10**) for a description of what is displayed. (2) Clear account balance and P/L: for those who have incurred large losses on their account, this operation allows a trader to start over again with a cleared P/L and new funds in the account. This feature is primarily useful when an account on the Trading System is used as a game—i.e., no real money changes hands. (3) Add funds to account: for a game account, add funds to the account for a real money-account, transfer money into the account from the trader's credit card or obtain instructions on how to wire transfer money into the trader's account. (4) Buy/Sell: issue a trade or market order (see FIG. **13**). (5) Open positions: display the open positions in a table (see FIG. **8**). (6) Open trades: display all open trades in a table (see FIG. **7**). (7) Open orders: display all open orders in a table (see FIG. **9**). (8) Recent transaction history: display the most recent transactions in a table (see FIG. **10**).

Commands Menu: (1) Change passwords. (2) Graph: specify the currency pair to be displayed in the graph.

Information Menu: (1) Interest rates: display interest rate information in a separate browser window. (2) Market News: display up-to-date currency market news in a separate browser window. (3) Analysis tools: use an analysis tool in a separate window. (4) Forum: participate in various forums related to currency trading. (5) Rankings: see a list of the most successful currency traders using the Trading System.

Help Menu: (1) Documentation: links to descriptive documents. (2) About: display software version number and credits. (3) Debug: display debugging information in a new window.

Account Summary

The account summary display (see FIG. **6** for an example) is a small table that provides a summary of the trader's account status. It preferably shows: (1) Account Balance: the amount of the trader's cash holding in the trader's account. (2) Realized P/L: the amount of profit or loss the trader has incurred with the trader's trading activity to date. (3) Unrealized P/L: the amount of profit or loss that the trader holds with the trader's current open positions. If the trader clears all of his open positions, then this amount would be added to the Realized P/L amount. (4) Margin Used: the amount of the trader's account balance and unrealized P/L tied up for margin purposes. (5) Margin Available: the amount of the trader's account balance and unrealized P/L available as margin for new trading transactions.

This information is preferably continuously updated in real-time to take current market conditions into account. Moreover, the information is always shown in the trader's home currency.

Tables

The table area of the Trading Station shows different information depending on the last Action Button-selected. It can include: (1) open trades; (2) open positions; (3) open orders; and (4) transaction history. The default is open positions.

How the information in the table is displayed can be controlled in two ways: (1) Scroll bars are used to scroll the table up or down, allowing a trader to see information that is hidden from view. (2) Sorting can be achieved by clicking on a column header, which causes the table to be sorted so that the column is in increasing or decreasing order. Clicking once sorts the column in increasing order; clicking again sorts it in decreasing order.

For all tables except Transaction history, clicking on a row of the table will cause a pop-up window to appear, offering further actions for that open trade, position, or order.

Open Trades Table

The open trades table (see FIG. 7) shows a list of the trader's currently open trades. The table preferably has 9 columns, described from left to right (not all are depicted in FIG. 7):

(1) Short/Long: Indicates whether the position is short or long.

(2) Ticket Number: a number that uniquely identifies an open trade. A trader can use this number as a reference for inquiries to the Trading System or its operators, or to search for particular entries in the transaction history table.

(3) Currency pair: the pair of currencies involved in this trade. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(4) Units: the number of transacted units for this trade, expressed in the base currency.

(5) S/L: the trader's stop-loss for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: the exchange rate obtained when the trade got executed.

(8) Market: the current exchange rate for this currency pair.

(9) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an open trade will cause a pop-up window to appear offering two operations: (1) Close trade. (2) Modify trade. This is used to modify the S/L or the T/P limits.

Open Positions Table

The Open Positions Table (see FIG. 8) displays a list of the trader's open positions. It is similar to the Open Trades table, except that all trades of the same currency pair are aggregated into one line.

The table preferably has 6 columns, described from left to right (not all are shown in FIG. 8):

(1) Short/Long: Indicates whether the position is short or long.

(2) Currency pair: the pair of currencies the position refers to. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(3) Units: the number of units held in this position, expressed in the base currency.

(4) Rate: the average exchange rate obtained for the trades in this position.

(5) Market: the current exchange rate for this currency pair.

(6) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column displayed in increasing or decreasing order. Clicking on a row with an open position will cause a pop-up window to appear offering the option to close the position.

Open Orders Table

The Open Orders Table (see FIG. 9) shows a list of the trader's currently open orders. An open order is a request that a particular trade should be made automatically when the exchange rate of the specified currency pair crosses a specified threshold.

The table preferably has 9 columns, described from left to right (not all are shown in FIG. 9).

(1) Short/Long: indicates whether the position is short or long.

(2) Order ID: a number that uniquely identifies the order. A trader can use this number as a reference for inquiries to the Trading System.

(3) Currency pair: the pair of currencies to be traded.

(4) Units: the number of units to be traded, expressed in the base currency.

(5) S/L: the stop-loss for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: specifies that the trade should be executed as soon as the exchange rate for the specified currency pair crosses this value.

(8) Market: the current exchange rate for this currency pair.

(9) Duration: specifies the amount of time an order should stand, until it is automatically canceled.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an order will cause a pop-up window to appear offering two operations: (1) Cancel order. (2) Modify order. This is used to modify the exchange rate threshold at which the trade is to be executed, or the S/L or T/P limits.

Transactions Table

The Transactions (or Transaction History) Table (see FIG. **10**) shows a list of the most recent transactions on the account. For access to a full list of past transactions, a user selects the Information pull-down menu and then selects Transaction History.

The Transaction History Table preferably has 6 columns, described from left to right:

(1) Transaction ID: uniquely identifies the transaction.

(2) Type: identifies the type of transaction.

(3) Currency pair: the pair of currencies associated with the transaction.

(4) Units: the number of units to traded in the transaction, expressed in the base currency.

(5) Price: the currency exchange rate applied when buying or selling a currency pair.

(6) Date/Time: the date and time of the transaction.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order.

Currency Rates

The Currency Rates Table (see FIG. **11**) shows the current exchange rate for the currency pairs supported by the Trading System. They are preferably updated in real time, approximately every 5 seconds. When there is a significant exchange rate movement for a currency pair, up/down indicators show the direction of the rate change in order to alert a trader, should a trader not currently have the currency pair displayed in the graph.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a currency pair's ask price will pop up a buy window for that currency pair. Clicking on a currency pair's bid price will pop up a sell window for that currency pair.

Graphs

Graphs (see FIG. **12**) show how currency exchange rates change over a period of time, ranging from minutes to months. All graphs are updated in real-time, as new currency rates arrive.

At any given time, the difference between the lower boundary and the upper boundary of the curve represents the difference between the bid and the ask price, and the difference may vary over time depending on market conditions. Thus, the top part of the curve indicates the ask price, and the bottom of the curve indicates the bid price.

As a mouse cursor **1220** is moved over the graph, a sub-area **1230** in the graph shows precise exchange rate information for the target currency pair corresponding to the time instance represented by the position of the mouse cursor.

The graph may also display Buy or Sell widgets that indicate at which point in time a trader bought or sold a currency pair. Downward pointing red arrows indicate a sold currency pair (where a trader is hoping the rates will go down after that point), and upward pointing green arrows indicate a bought currency pair (where a trader is hoping the rates will go up after that point).

A trader can adjust what is shown in the graph: (1) The currency pair displayed is selected using the pull-down menu **1240** at the bottom left. (2) The granularity of the graph is selected using the pull-down menu **1250** at the bottom right of the graph. Selecting a fine granularity, such as 5 seconds (where each point on the horizontal axis represents 5 seconds of time), will display a relatively short time interval (less than an hour, in this case). Selecting a larger granularity, such as one day, will display longer-tern trends (9 months of exchange rate information in this case).

Scroll buttons **1260** at the top right of the graph area allow a trader to shift the time interval shown to the left or to the right (backward or forward in time). Clicking on the graph with the mouse will hide the Buy/Sell widgets. Clicking again will cause them to reappear.

Buy/Sell Window

A Buy/Sell pop-up window (see FIG. **13**) allows a trader to issue buy or sell orders. The window can be caused to pop up either by: (1) clicking on the Buy/Sell action button (see FIG. **5**); (2) clicking on the bid or ask price in the Currency Rates Table (see FIG. **11**); or (3) clicking on an existing trade, position, or order in the Table area of the Trading Station display (see FIG. **5**).

Two types of orders are supported: (1) Market Orders are orders that are transacted immediately based on market exchange rates. (2) Entry Orders are orders that are executed when the exchange rate crosses a certain threshold.

The type of order can be selected by clicking on the appropriate tab in the Buy/Sell Window (see FIG. **13**). Market order comes up as the default order.

Issuing a Market Order. To issue a market order with the Buy/Sell Window and the Market Tab selected, a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) If the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph. (B) If the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. (C) If a trade or position was clicked in the Table area, the currency pair corresponding to the trade or position. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) Lower Limit: the order will result in a trade only if a price is obtained that does not lie below this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(5) Upper Limit: the order will result in a trade only if a price is obtained that does not lie above this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

The RATE field is set by the Trading Station and corresponds to the most recent exchange rate for the selected currency pair.

To issue the order, a Submit button **1310** must be selected. If the order is successful, and a trade occurs, then an acknowledgment window (see FIG. **14**) pops up with a Ticket number that can be used for future reference. Moreover, the Open Trades Table (see FIG. **7**) will be updated to reflect the new trade, as will the Open Positions Table (see FIG. **8**) and the Transaction History Table (see FIG. **10**).

Several issues are important to note:

(1) If an order is successful and a trade occurs, then the exchange rate obtained for the trade will correspond to the most current exchange rate maintained at the Trading System servers and not necessarily the rate displayed in the Buy/Sell window.

(2) An order without Lower and Upper Limits will always result in a trade.

(3) An order with both Lower and Upper Limits will result in a trade if and only if the exchange rate for the potential trade lies between the two limits.

Issuing an Entry Order. To issue an entry order with the Buy/Sell Window and the Entry Tab selected (see FIG. **15**), a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) if the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph; (B) if the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) RATE: the order will result in a trade as soon as the exchange rate for the selected currency pair crosses the given value; that is, for buy orders, if the rate goes below this value, and for sell orders if the rate goes above the given value

(5) Duration: this value is used to limit the amount of time an outstanding order will remain effective. By default, the order remains effective indefinitely. However, the duration can be set to the end of the day or for an hour.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

To issue the order, a Submit button **1510** must be selected. This results in an acknowledgment window (see FIG. **14**) popping up with a Ticket number that can be used for future reference. Moreover, the Open Orders Table (see FIG. **9**) will be updated to reflect the new order. Note that a trader can modify the parameters of an open order (including the rate representing the trade threshold, or the S/L and T/P) by clicking on the order in the Open Orders Table.

What is claimed is:

1. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising an interest rate manager in communication with said transaction server and said database, wherein said interest rate manager is operative to calculate, pay out, and collect interest on a tick-by-tick basis.

2. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a stop-loss daemon that (a) continuously checks whether stop-loss orders should be executed, and (b) if a stop-loss order should be executed, executes it through said transaction server.

3. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a take-profit daemon that (a) continuously checks whether take-profit orders should be executed, and (b) if a take-profit order should be executed, executes it through said transaction server.

4. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; and further comprising a trade manager in communication with said transaction server and said database, wherein said trade manager comprises a limit-order daemon that (a) continuously checks whether limit orders should be executed, and (b) if a limit order should be executed, executes it through said transaction server.

5. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server, wherein said pricing engine is operable to compute currency exchange rates based on: (a) data obtained from external rate sources; and (b) market directional movement and volatility.

6. A system as in claim 5, wherein said pricing engine is further operable to compute currency exchange rates based on positions held by said system.

7. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; further comprising a hedging engine in communication with said transaction server, wherein said hedging engine is operable to perform at least two of the following calculations: (a) calculate a total amount of home currency appearing in all open positions; (b) calculate an out-of-equilibrium exposure; (c) calculate a new potential net exposure; (d) calculate an equilibrium position; (e) calculate boundaries of possible exposures; (f) calculate values for a pair of quoting functions; and (g) calculate an average price and an average spread.

8. A system as in claim 6, wherein said positions are managed based on one or more trading models.

9. A system as in claim 8, wherein at least one of said one or more trading models comprises: (a) a price collector component; (b) a price filter component; (c) a price database component; (d) a gearing calculator component; (e) a deal acceptor component; and (f) a book-keeper component.

10. A system as in claim 8, wherein at least one of said one or more trading models comprises: (a) a price collector component; (b) a price filter component; (c) a price database component; (d) a gearing calculator component; (e) a deal acceptor component; (f) an opportunity catcher component; and (g) a book-keeper component.

11. A system for trading currencies over a computer network, comprising:

(a) a server front-end in communication with said computer network;

(b) a database;

(c) a transaction server in communication with said server front-end and with said database;

(d) a rate server in communication with said server front-end; and

(e) a pricing engine in communication with said rate server; further comprising a margin control manager in communication with said transaction server and said database, wherein said margin control manager is operable to monitor on a tick-by-tick basis margin requirements of accounts and on said tick-by-tick basis liquidate holdings as needed to maintain specified margins.

* * * * *

US007496534B2

(12) **United States Patent**
Olsen et al.

(10) **Patent No.: US 7,496,534 B2**
(45) **Date of Patent: Feb. 24, 2009**

(54) **METHODS FOR TRADE DECISION MAKING**

(76) Inventors: **Richard B. Olsen**, Wehrenbachhalde 46, 8053 Zurich (CH); **Michel M. Dacorogna**, Zollikerstrasse 246, Zurich (CH); **Olivier V. Pictet**, 9 chemin des Tulipiers, 1208 Geneva (CH); **Ulrich A. Müller**, Seefeldstrasse 239, 8008 Zurich (CH); **Rakhal D. Davé**, Ottenbergstrasse 81, 8049 Zurich (CH); **Lars A. Jaeger**, Krebsgasse 7, 8001 Zurich (CH)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1470 days.

(21) Appl. No.: **09/855,633**

(22) Filed: **May 14, 2001**
(Under 37 CFR 1.47)

(65) **Prior Publication Data**

US 2002/0184134 A1 Dec. 5, 2002

Related U.S. Application Data

(60) Provisional application No. 60/274,174, filed on Mar. 8, 2001.

(51) **Int. Cl.**
G06Q 40/00 (2006.01)

(52) **U.S. Cl.** .... **705/37**; 705/35; 705/36 R

(58) **Field of Classification Search** .... 705/35, 705/36 R, 37
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,926,822	A *	7/1999	Garman	715/503
6,012,042	A *	1/2000	Black et al.	705/36 R
6,016,483	A *	1/2000	Rickard et al.	705/36 R
6,061,662	A *	5/2000	Makivic	705/36 R
6,195,103	B1 *	2/2001	Stewart	345/440
6,317,728	B1 *	11/2001	Kane	705/36 R
6,360,210	B1 *	3/2002	Wallman	705/36 R
6,832,210	B1 *	12/2004	Li	705/36 R

OTHER PUBLICATIONS

Brooks, Chris. "A Double-Threshold Garch Model For The French Franc/Deutschmark Exchange Rate", Journal of Forecasting. Chichester: Mar. 2001. vol. 20, Iss. 2; p. 135.*

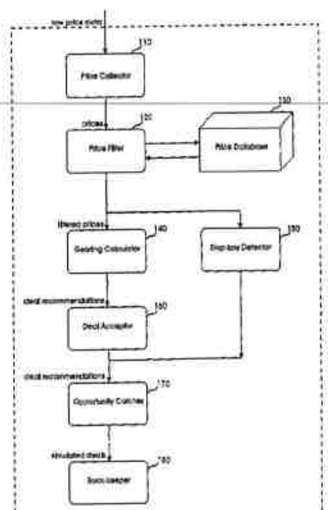
* cited by examiner

Primary Examiner—Harish T. Dass
Assistant Examiner—Elda Milef
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

A method of trading assets on a market including: (1) receiving price data for an asset over one or more computer networks; (2) receiving current system position information; (3) storing the received asset price data and current system position information in a computer-readable medium; (4) calculating trade recommendation information form each of one or more trading sub-models, wherein each sub-model is based on a different time horizon, the calculation based on the received asset price data; and (5) calculating a trade recommendation regarding the asset based on the trade recommendation information from each of the trading sub-models. Each sub-model preferably includes: (1) a price collector component; (2) a price filter component; (3) a price database component; (4) a gearing calculator component; (5) a deal acceptor component; and (6) a book-keeper component.

12 Claims, 8 Drawing Sheets



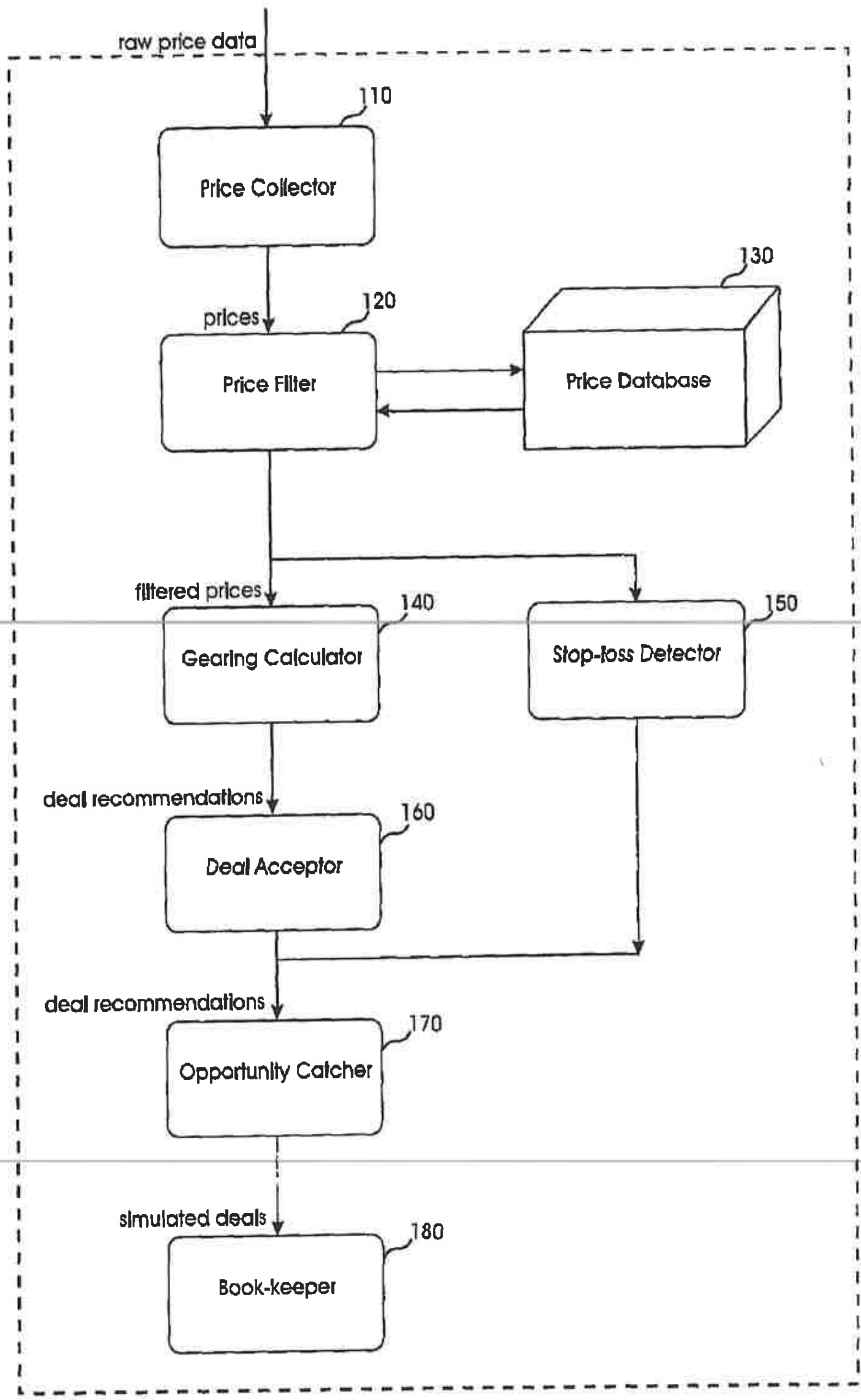

raw price data
110
Price Collector
prices
120
Price Filter
130
Price Database
filtered prices
140
Gearing Calculator
150
Stop-loss Detector
deal recommendations
160
Deal Acceptor
deal recommendations
170
Opportunity Catcher
simulated deals
180
Book-keeper

FIG. 1

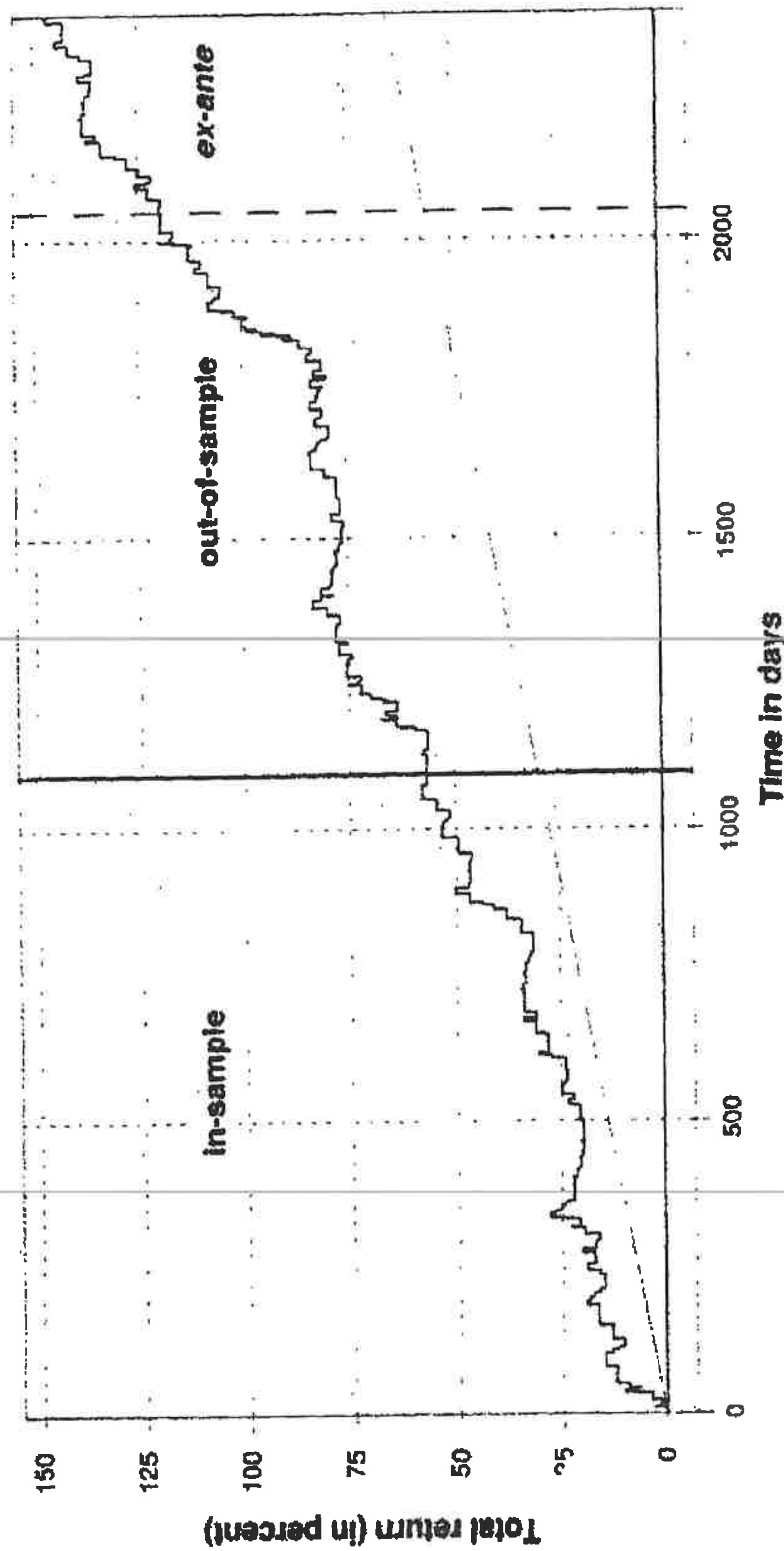

ex-ante
out-of-sample
in-sample
Time in days
Total return (in percent)
0
500
1000
1500
2000
150
125
100
75
50
25
0

FIG. 2

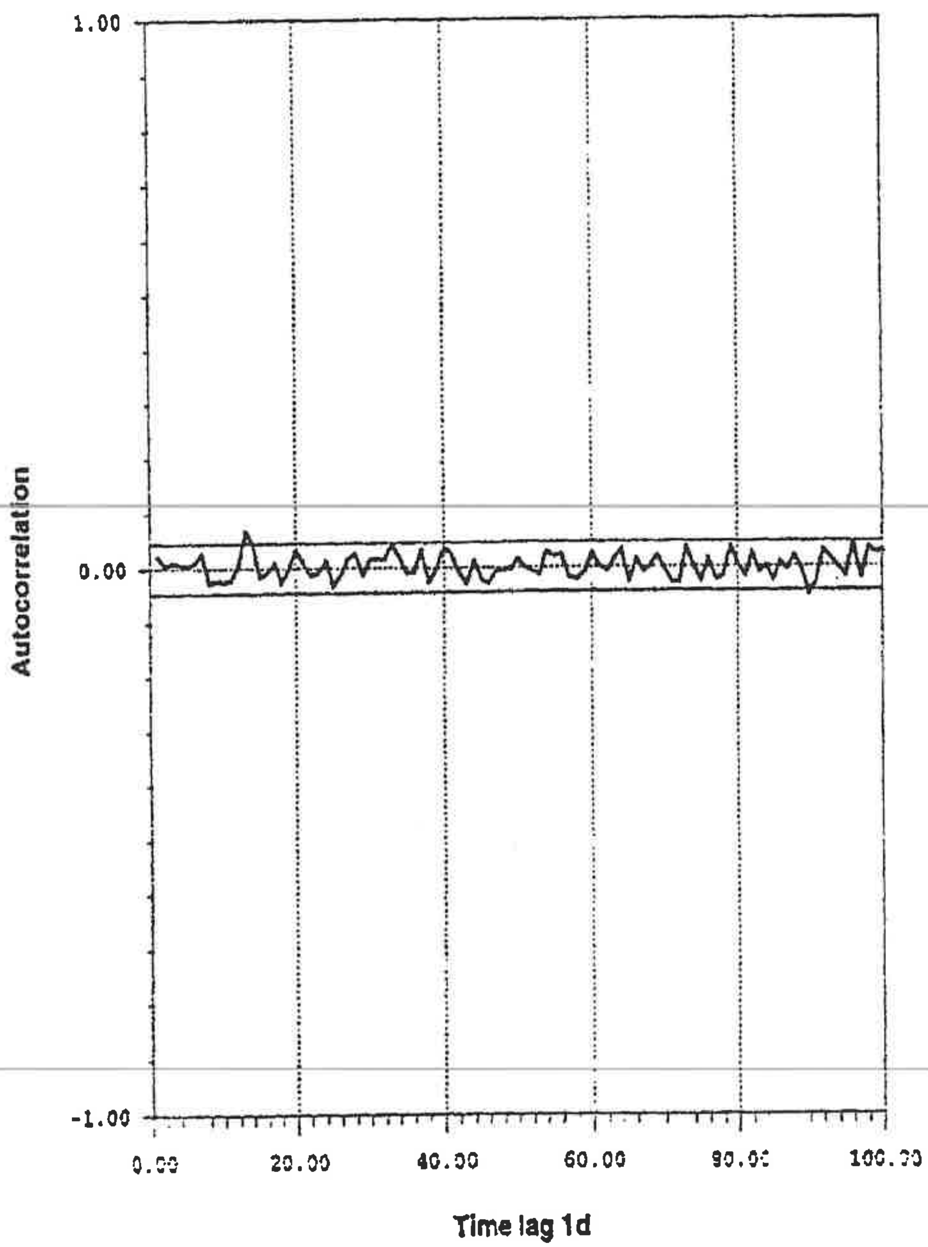
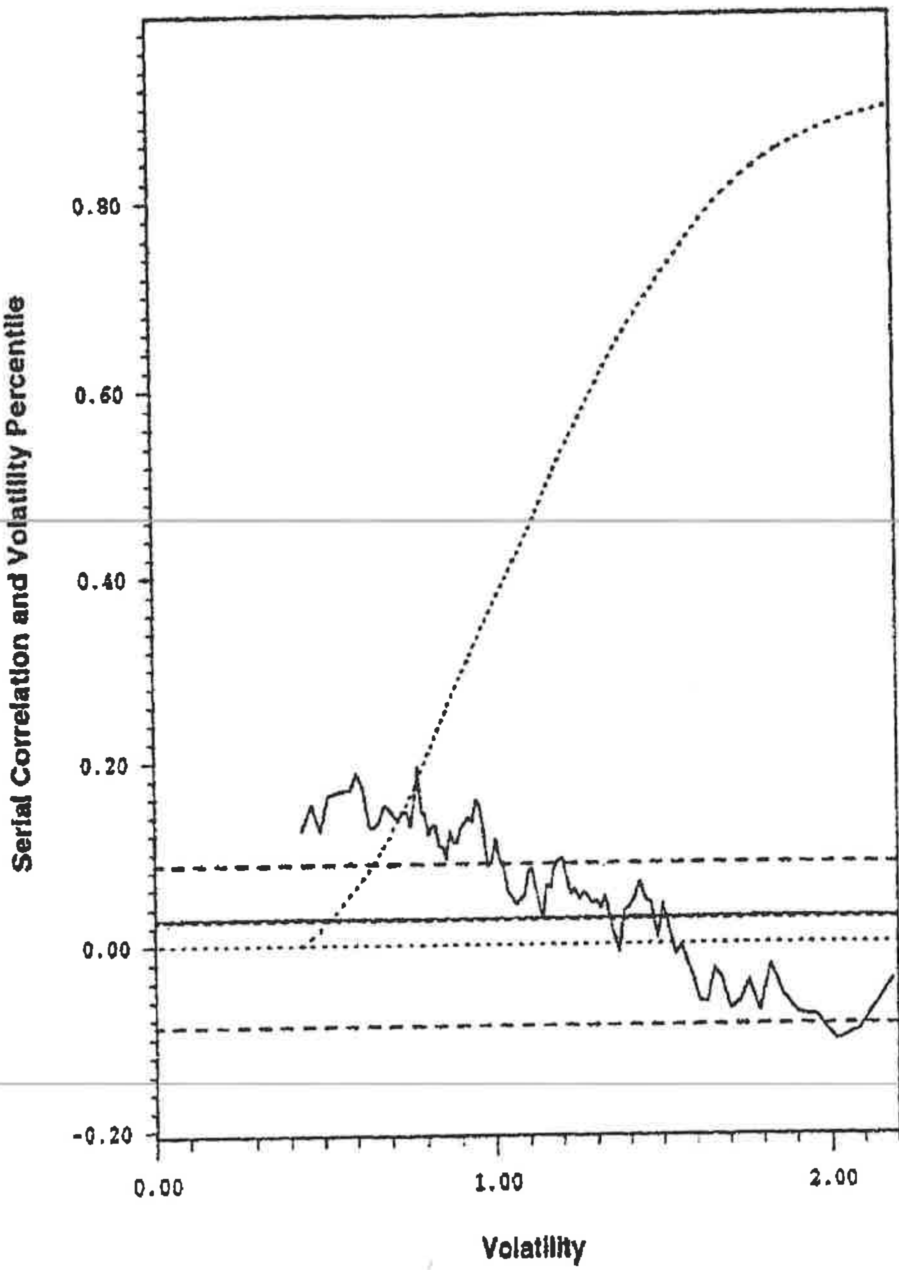

1.00
0.00
-1.00
Autocorrelation
0.00
20.00
40.00
60.00
80.00
100.00
Time lag 1d

FIG. 3



Serial Correlation and Volatility Percentile
0.80
0.60
0.40
0.20
0.00
-0.20
0.00
1.00
2.00
Volatility

FIG. 4

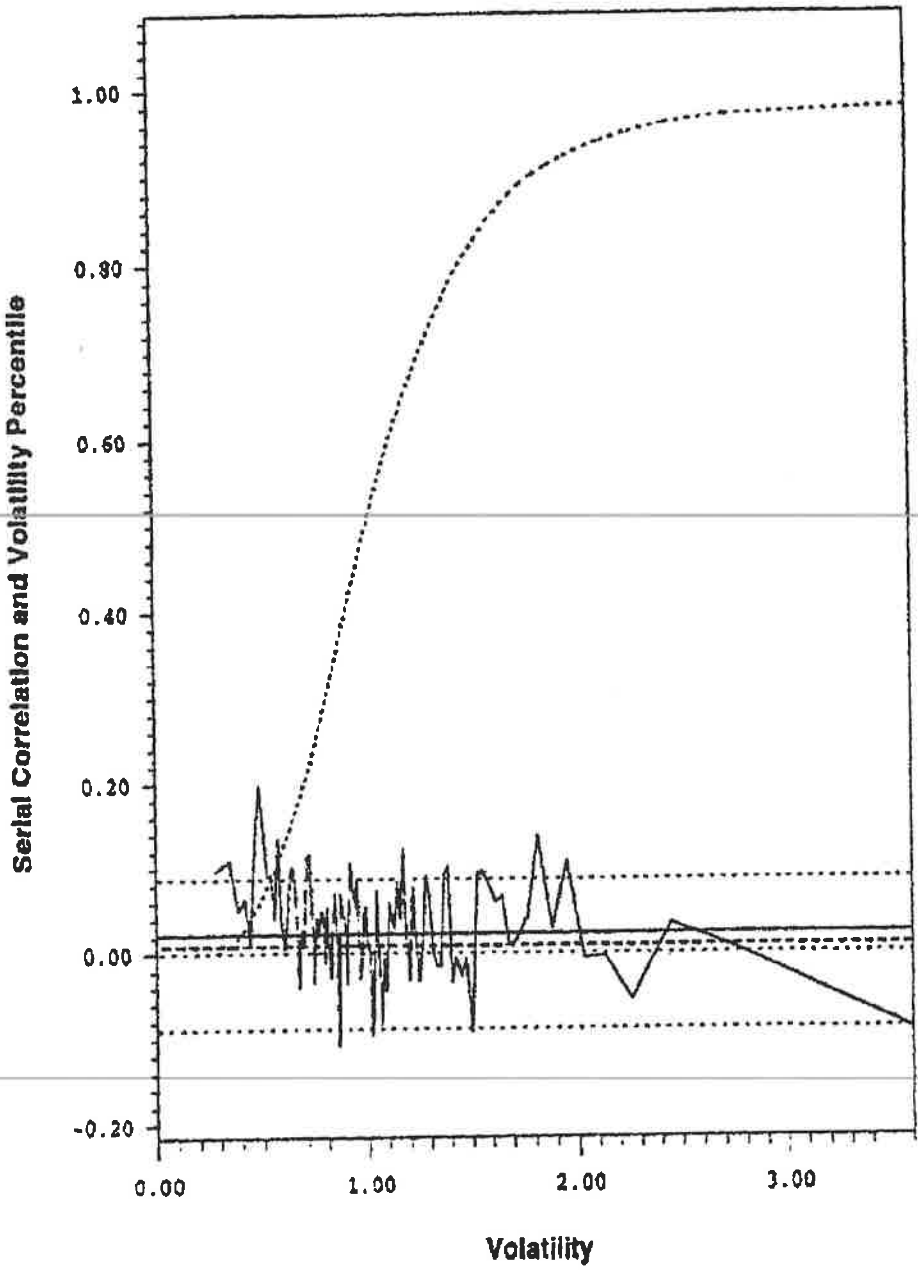

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
Serial Correlation and Volatility Percentile
0.00
1.00
2.00
3.00
Volatility

FIG. 5

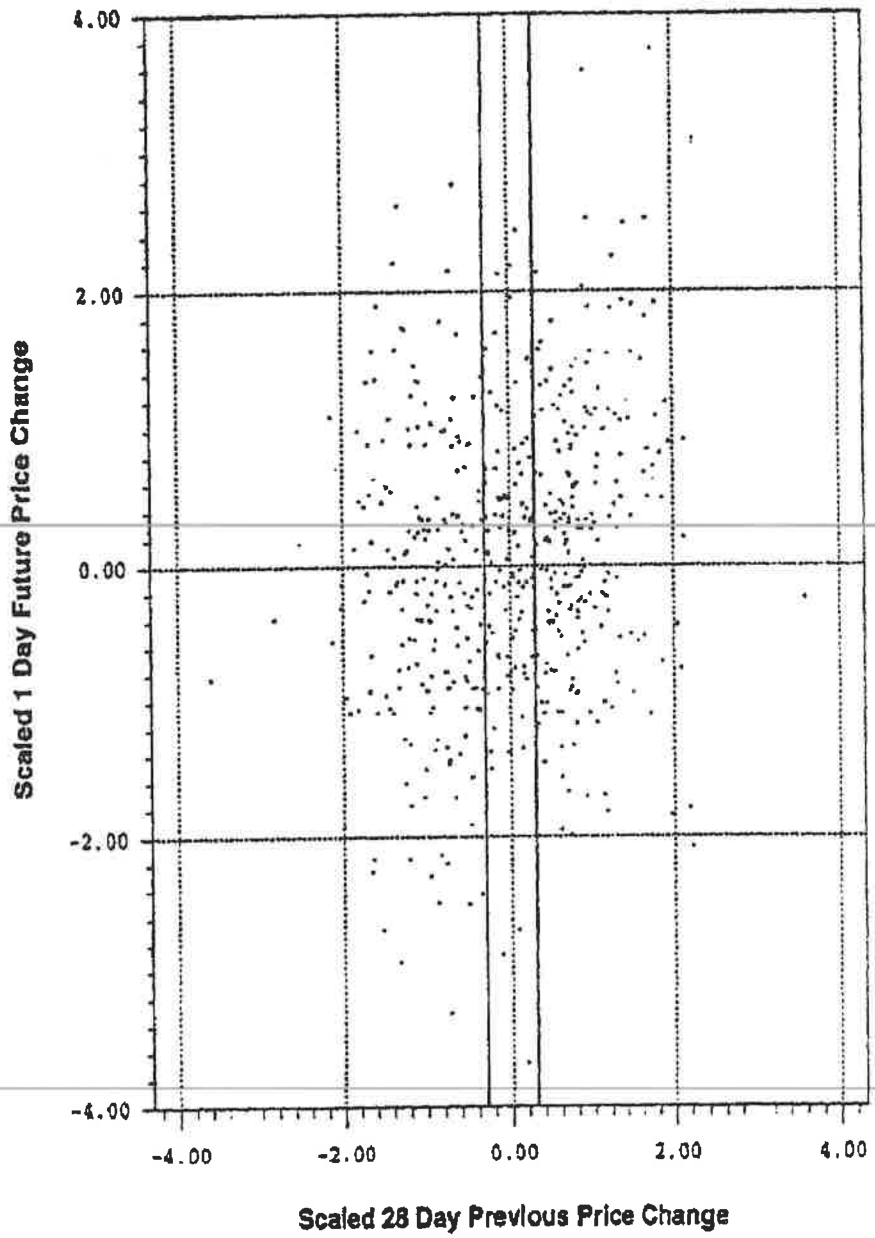

4.00
2.00
0.00
-2.00
-4.00
Scaled 1 Day Future Price Change
-4.00
-2.00
0.00
2.00
4.00
Scaled 28 Day Previous Price Change

FIG. 6

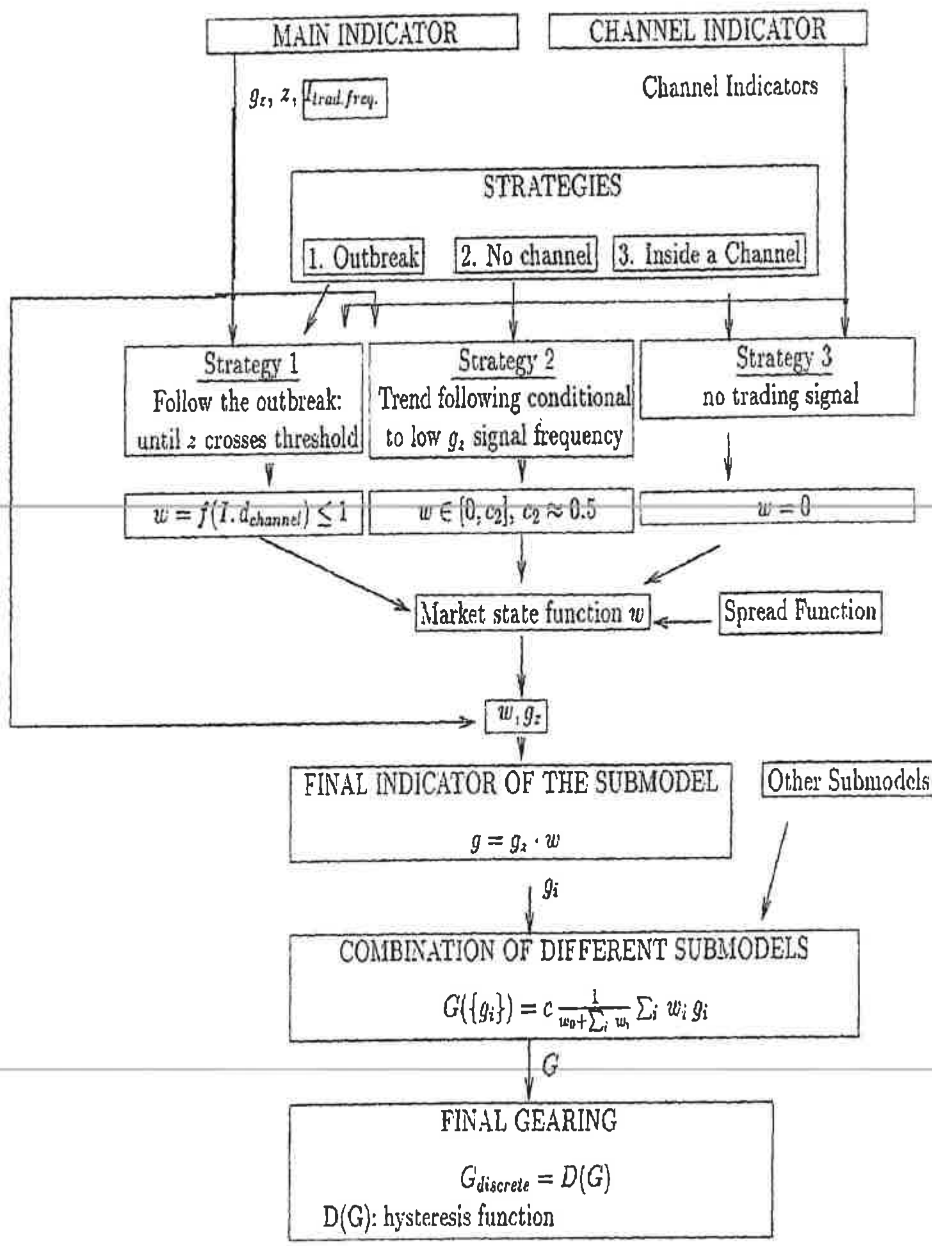

MAIN INDICATOR
CHANNEL INDICATOR
g_z, z, $I_{trad.freq.}$
Channel Indicators
STRATEGIES
1. Outbreak
2. No channel
3. Inside a Channel
Strategy 1
Follow the outbreak:
until z crosses threshold
Strategy 2
Trend following conditional
to low g_z signal frequency
Strategy 3
no trading signal
$w = f(I, d_{channel}) \leq 1$
$w \in [0, c_2]$, $c_2 \approx 0.5$
$w = 0$
Market state function w
Spread Function
w, g_z
FINAL INDICATOR OF THE SUBMODEL
$g = g_z \cdot w$
Other Submodels
g_i
COMBINATION OF DIFFERENT SUBMODELS
$G(\{g_i\}) = c \frac{1}{w_0 + \sum_i w_i} \sum_i w_i g_i$
G
FINAL GEARING
$G_{discrete} = D(G)$
D(G): hysteresis function

FIG. 7

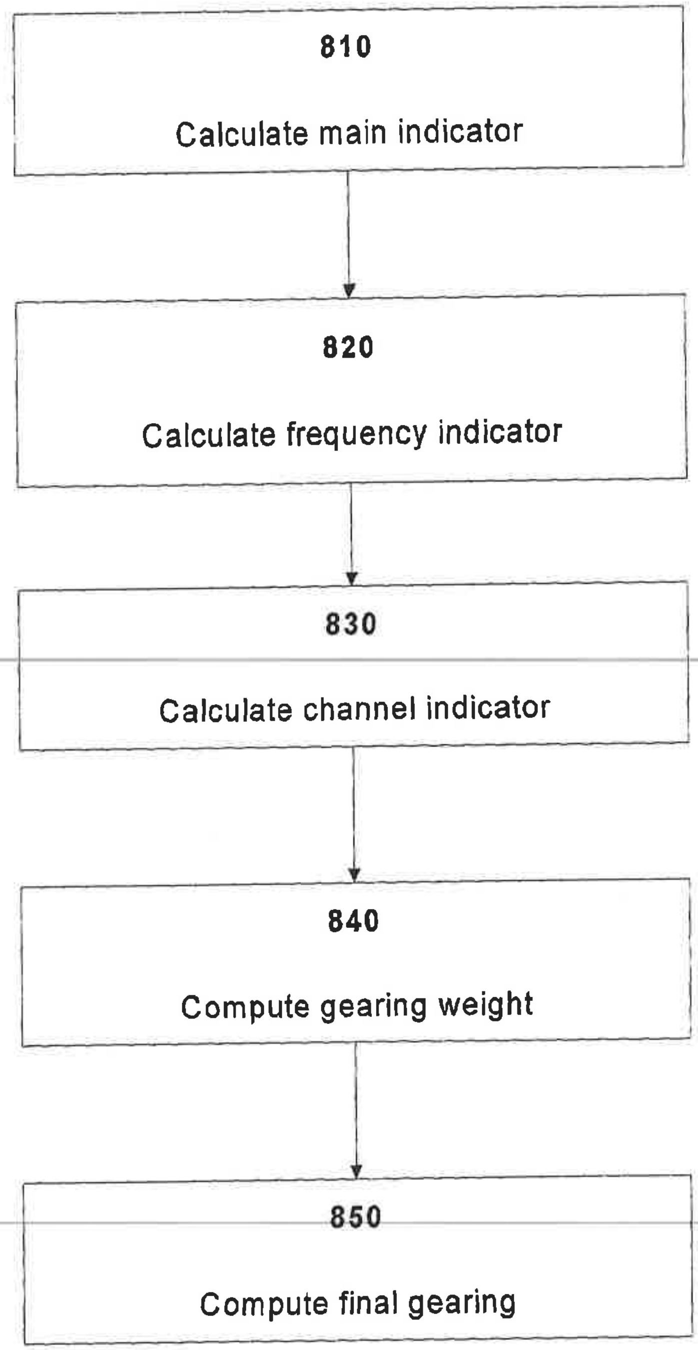

810
Calculate main indicator
820
Calculate frequency indicator
830
Calculate channel indicator
840
Compute gearing weight
850
Compute final gearing

FIG. 8

METHODS FOR TRADE DECISION MAKING

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application No. 60/274,174, filed Mar. 8, 2001. The contents of the above application are incorporated herein in their entirety by reference.

FIELD OF THE INVENTION

The subject invention is directed to trading on financial markets, in particular to methods of using real-time trading models to trade on foreign exchange markets.

BACKGROUND

An exchange rate is the price at which one national currency can be exchanged for another. The most common currency value notion is the bilateral exchange rate (or simply the foreign exchange (FX) rate) quoted by an FX trader or reported by a quote vendor. This is a nominal exchange rate because it is the number of units of a currency offered in exchange for one unit of another (for example, 1.5 German marks for 1 U.S. dollar). The spot exchange rate is a particular example of a nominal bilateral exchange rate, where the transaction takes place immediately. Another example is the forward exchange rate, where the price is agreed now but the transaction takes place in the future. An FX price is composed of two quantities: a price at which a bidder offers to buy (termed the bid price) and a price at which a seller offers to buy (the ask price).

The bid and ask prices of major financial institutions are conveyed to dealers' screens by quote vendors such as Reuters, Telerate, or Knight Ridder. Deals are typically negotiated via telephone. The FX market operates globally and around the clock. Any market maker may submit new bid/ask prices at any time, and many institutions have branches worldwide, so that they can participate in continuous FX trading.

Although the FX market operates continuously, individual traders or institutions generally participate in this market for only part of each day. There is thus a need for trading models that take take local business hours and holidays into account.

There is a need for trading models that offer real-time analysis of FX-rate movements and generate explicit trading recommendations. A clear distinction should be made between a price change forecast and an actual trading recommendation. A trading recommendation naturally includes some kind of price change forecast, but must also account for the specific risk profile of the dealer or user of the respective trading model. Another distinction is that a trading model must take into account its past trading history, while a price forecast is not biased by a position the trading model might be in. A trading model thus preferably goes beyond predicting a price change: it should decide whether a certain action has to be taken. This decision is subject to the specific risk profile, the trading history, and institutional constraints such as business hours. These different parameters can be integrated into the decision-making process and are important for constructing practical models for professional traders.

There is a further need for models that follow the FX market and imitate it as closely as possible. These models should be based on data from continuous collection and treatment of FX quotes by market makers around the clock (up to 5000 non-equally-spaced prices per day for the German

mark, or the euro, against the U.S. dollar). In order to imitate real-world trading accurately, the models also should take into account transaction costs in their return computation, generally avoid trading outside market working hours, and avoid trading too rapidly.

SUMMARY

A preferred embodiment of the subject invention comprises a system and method for operating a set of real-time trading models. An over-all system that utilizes the subject invention includes mechanisms for data collection, data validation, generation of trading recommendations, communication of these recommendations to user-agents, and graphical presentation of these recommendations on the user-agents.

A preferred trading model is based on a set of indicator computations combined with a collection of rules. The former are functions of the price history. The latter determine the applicability of the indicator computations in generating trading recommendations.

A preferred embodiment comprises a method of trading assets on a market, comprising the steps of: (1) receiving price data for an asset over one or more computer networks; (2) receiving current system position information; (3) storing the received asset price data and current system position information in a computer-readable medium; (4) calculating trade recommendation information from each of one or more trading sub-models, wherein each sub-model is based on a different time horizon, the calculation based on the received asset price data; and (5) calculating a trade recommendation regarding the asset based on the trade recommendation information from each of the trading sub-models. In this embodiment, each sub-model has its own indicator.

A further preferred embodiment comprises a method of trading assets on a market, comprising the steps of: (1) receiving price data for an asset over one or more computer networks; (2) receiving current system position information; (3) storing the received asset price data and current system position information in a computer-readable medium; (4) calculating trade recommendation information from each of one or more trading sub-models, wherein each sub-model is based on a different time of day, the calculation based on the received asset price data; and (5) calculating a trade recommendation regarding the asset based on the trade recommendation information from each of the trading sub-models. In one version of this embodiment, 24 sub-models are used. In another version, each sub-model further comprises a pair of sub-models, each pair consisting of a trend-following sub-model and a contrarian sub-model.

A further preferred embodiment comprises a method of trading assets on a market, comprising the steps of: (1) receiving price data for an asset over one or more computer networks; (2) receiving current system position information; (3) storing the received asset price data and current system position information in a computer-readable medium; (4) calculating trade recommendation information from each of one or more trading sub-models, wherein each sub-model is based on the view and actions of one group of traders in the asset and is further based on its own time horizon, the calculation based on the received asset price data; and (5) calculating a trade recommendation regarding the asset based on the trade recommendation information from each of the trading sub-models. In this embodiment, the step of calculating trade recommendation information from each of one or more trading sub-models comprises, for each sub-model, the following steps: (1) calculating a suitable main indicator for the sub-model, which after application of a nonlinear discretization

function provides a first raw gearing; (2) calculating a frequency indicator; (3) calculating a channel indicator; (4) computing a weight to be applied to the raw gearing; and (5) computing a final gearing by multiplying the weight times the raw gearing and then discretizing the product.

In each of these embodiments, each sub-model preferably comprises: (1) a price collector component; (2) a price filter component; (3) a price database component; (4) a gearing calculator component; (5) a deal acceptor component; and (6) a book-keeper component. This overall structure and dataflow of a preferred trading model is depicted in FIG. **1**.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts a data flow and structure diagram of preferred trading model software.

FIG. **2** depicts an example of a total return curve over three periods: in-sample, out-of-sample, and ex ante.

FIG. **3** depicts the USD-DEM autocorrelation function of the price changes (with sign) with lags in units of 1 day in physical time up to a lag of 100 days.

FIG. **4** depicts the USD-DEM serial correlation conditional to volatility.

FIG. **5** depicts the intra-day USD-DEM serial correlation conditional to volatility.

FIG. **6** depicts a scatter plot of δx_{last} along the x-axis against δx_{next} along the y-axis for the bin corresponding to the maximum positive correlation in FIG. **4**.

FIG. **7** depicts a preferred structure of a customized trading model: sub-model and main model.

FIG. **8** depicts sub-model-related steps of a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

At the most general level, a preferred trading model comprises a set of indicator computations combined with a collection of rules. A detailed description of typical indicators is provided below; indicator computations provide an analysis of past price movements. The indicators are mapped into actual trading positions by applying various rules. For instance, a model may enter a long position if an indicator exceeds a certain threshold. Other rules determine whether a deal may be made at all, and the timing of a recommendation. Thus, indicator computations are based on price history, and a collection of rules determines the applicability of the indicator computations to the generation of trading recommendations.

Referring to FIG. **1**, in computer software that comprises preferred trading model software, a price collector **110** collects price quotes from data feeds received over a computer network. A price filter **120** receives the collected price quotes and filters them in real-time, storing filtered price quotes in a price database **130** and sending them to a gearing calculator **140**. A gearing calculator **140** specifies the recommended gearing (exposure size), based on indicator computations that depend on the received and filtered price quote data, trading rules that depend on past dealing history, current position, and other quantities such as current unrealized return of an open position. Software of preferred embodiments uses the following gearings: −1, −0.5, 0, +0.5, and +1, corresponding to short with all of the available capital, short with half of the available capital, neutral, long with half of the available capital, and long with all of the available capital, respectively. The position is re-evaluated each time a new price quote arrives.

A deal acceptor **160** validates the recommendations of the gearing calculator **140** under the following conditions: (1) no deal may take place within fifteen minutes of a deal already having occurred, and (2) a stringent deal filter determines whether a given price is suitable for dealing. A stop-loss detector **150** triggers if the market moves in the wrong direction and passes a stop-loss price, causing a deal-to-neutral position. The stop-loss price can be adjusted dynamically.

An opportunity catcher **170** searches for a realistic price at which to execute the deal, preferably in a time window of 2-3 minutes (depending on the currency). Then it executes a simulated deal and provides signals to a human dealer. A Book-keeper **180** software module evaluates the trading model statistics.

Imitating the real world requires a system that collects all available price quotes and that reacts in real-time to price movements. In a preferred embodiment of the subject invention, trading models use mainly Reuters data but other data suppliers provide similar information in their FX quotes. A preferred embodiment uses price quotes that have been collected, validated, and stored in a database. Each record or tick in the database contains the following items: (1) the time t_j in GMT (Greenwich Mean Time) at which the price quote has been recorded in the database; (2) the bid price p_j^{bid}; (3) the ask price p_j^{ask}; (4) the name of the bank that issued the price quote; (5) the location of the issuing bank; and (6) the result of a validation filter.

The index j identifies individual database records. Reuters pages provide the bid price p_j^{bid} as a complete number, usually with five digits; p_j^{ask} is given with only the last two digits and is recomputed in full by software of a preferred embodiment. The granularity of the time stamps t_j is 2 seconds. The validation filter information is preferably computed by the filtering algorithm described in the paper, Müller, U. A., Dacorogna, M. M., Olsen, R. B., Pictet, O. V., Schwarz, M., Morgenegg, C., *Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis*, Journal of Banking and Finance, 14:1189 (1990) (hereinafter Müller 1990).

In a preferred embodiment, trading models are each preferably associated with a local market that is identified with a corresponding geographical region. In turn, this region is associated with generally accepted office hours and public vacation days. The local market is assumed to be open at any time during office hours, which do not occur during a weekend or a public holiday. Disclosed trading models support the Zürich, London and Frankfurt markets, but it is straightforward to extend this set. Typical open hours for a model are between 8:00 and 17:30 local time, the exact times depending on the particular local market.

Except for closing an open position if the price hits a stop loss limit (described below)), a preferred model does not deal outside open hours or during holidays. Note: in this description, we speak of a trading model "dealing," or "entering a new position," but preferred trading models do not deal directly (that is, without human intervention), but instead provide trade recommendations to a human dealer.

A preferred embodiment of the subject invention comprises a system and method for operating a set of real-time trading models. These include data collection, data validation, generation of trading recommendations, communication of these recommendations to user-agents residing at customers and graphical presentation of these on the user-agents.

The overall structure and data-flow of a preferred trading model is depicted in FIG. **1**.

Indicator computations form a central part of a preferred trading model, providing an analysis of past price movements. These indicators are mapped into actual trading positions by applying various rules. For instance, a model may enter a long position if an indicator exceeds a certain threshold. Other rules determine whether a deal may be made at all. Among various factors, these rules take the timing of the recommendation into consideration.

A complete trading model thus consists of a set of indicator computations combined with a collection of rules. The former are functions of price history. The latter determine the applicability of the indicator computations in generating trading recommendations.

The most significant item displayed to a user-agent is the model's present position or trading recommendation. Preferred trading models make gearing recommendations of S−1.0, S−0.5 (short positions), 0 (neutral), L+0.5 and L+1.0 (long positions). A gearing of S−0.5 means "go short with half the available capital." A gearing of S−1.0 means "go short with all of the available capital."

Software of a preferred embodiment is not constructed as a single huge program with all the required functionality residing in that one entity. Rather it is constructed as a collection of separate programs. In computer-speak, the system is said to be a distributed system because the various programs run in parallel on several computers.

Each such program preferably performs essentially one primary function. For instance, a program termed the GTrader takes in collected and validated price data and uses this data to produce trading model recommendations. (Several trading models may reside independently of one another within one GTrader invocation). Other programs in the system include a price collector for receiving data from the quote-vendors and a price database manager for storing this data for future use. Splitting up the system into several logical components confers several benefits:

One benefit is that a failure in one program is localized and does not affect other programs in the system. This increases the overall reliability. For example, if a GTrader program should crash—hopefully, a rare occurrence but one which may happen as a result of hardware failure—collection of raw price data remains unimpaired.

A second benefit is that if a computer should fail, the programs running on that computer may be transferred to another machine. Again, this promotes reliability. A third benefit is that the performance of the system may be enhanced by adding extra hardware and moving some existing programs onto the new hardware. Finally, a fourth benefit is that having separate programs perform logically distinct operations helps simplify the design and structure of the system.

A gearing calculator lies at the heart of a preferred trading model. The gearing calculator provides the trading model with its intelligence and the ability to capitalize on movements in the FX markets. The gearing calculator also provides the trading model with particular properties. These include the frequency of dealing and the circumstances under which positions may be entered.

In contrast, the other trading model components form a shell around the gearing calculator, providing it with price data, detecting whether the stop-loss limit is hit, and examining the trading recommendations made by the gearing calculator. The gearing calculator re-evaluates its position every time a new price tick is received from the quote-vendors. (As previously noted, a filter preferably validates each price quote beforehand, to eliminate outliers and other implausible data).

The gearing calculator preferably comprises (a) a set of indicators that are produced from the input price data (discussed in detail below); and (b) trading rules that are functions of the past dealing history, the current position, and other quantities (e.g., the current unrealized return of an open position).

As the name suggests, an indicator provides a measure of whether a new position should be entered. Indicators are preferably analyzed by rules inside the gearing calculator, to determine whether such a change of position should in fact take place. In the simplest form, an indicator crossing a predefined threshold may cause a rule to be activated that in turn causes such a change in position to occur. Thus the relative values of the indicators signify internal trading recommendations that are subsequently refined through the application of various rules. Other, more complicated rules may modify the indicators' basic recommendations in additional ways:

1. By inhibiting a recommendation produced by the indicators. For example, if the price movements since the previous deal are too small in either direction, an indicator recommendation to reverse the present gearing is suppressed.
2. By changing the indicator thresholds themselves. While the model is in a neutral position, such a rule may increase the threshold values.
3. By choosing an opposite position to the one hinted at by the indicators. That is, by entering a "contrarian" position. (A contrarian position is long when prices are declining or short when prices are climbing).
4. By imposing new stop-loss values. A smaller stop-loss may be established if the recommended position is contrarian. The stop-loss is reduced because a contrarian strategy is considered risky.

The relative values of the indicators determine the gearing that should be used. That is, not only do the indicators determine the position to be entered (after subsequent transformation by the rules), they also determine the magnitude of the gearing to be used. As stated previously, preferred models employ half-integral gearings in the range −1.0 to 1.0.

Other trading model components supporting the gearing calculator preferably do so through a standardized interface. Newly developed gearing calculators can thus be placed into an existing trading model environment. In addition to embodiments discussed below, at least one preferred embodiment comprises three separate gearing calculators, each working with seven currencies (USD/DEM, USD/CHF, GBP/USD, USD/JPY, USD/ITL, USD/FRF and USD/NLG). These gearing calculators differ in their respective trading frequencies, risk profiles and other properties. Developing a new gearing calculator is thus akin to developing a new trading class while preserving the overall trading model structure. This allows one to customize existing gearing calculators or to experiment with new ones.

The fact that the gearing calculator's indicators and rules suggest entering a new position does not necessarily mean that the model will make such a recommendation. Whether it does or not depends on various secondary rules that then take effect.

These rules are preferably comprised in a deal acceptor, which determines whether the deal proposed by the indicators is allowed to be made. The primary constraint is the timing of the proposed deal. First, no deal other than a stop-loss deal (discussed below) may take place within fifteen minutes of a deal already having occurred. This is to prevent overloading a human dealer who may be following the models. Second, the gearing calculator may make a recommendation to enter a new trading position, but this recommendation can be followed only if the local market is open.

The quality of the most recent price imposes another constraint. A stringent deal-filter determines if a given price is suitable for dealing. This is to ensure that recommended deals are made only with genuine prices rather than extraneous data. The deal acceptor permits a new deal only with a price passing the deal-filter.

If the gearing calculator suggests entering a new position but the deal acceptor decrees otherwise, the suggestion of the gearing calculator is ignored. Eventually, when timing and other factors are right, the gearing calculator will suggest entering a new position and the deal acceptor will approve.

Besides being passed on to a gearing calculator, the filtered price quotes are preferably also sent to a stop-loss detector.

The stop-loss detector triggers if the market moves in the "wrong" direction. If the model enters a trading position because it anticipates that the market will move in a certain direction, but in fact the market then moves the other way, the stop-loss price may be hit.

A preferred trading model defines a stop-loss price when a position is entered. If the current price—that is, the most recent price—moves below the stop-loss price (for a long position) or above the stop-loss price (for a short position), the stop-loss is said to be hit. Hitting the stop-loss causes a deal to a neutral position to be made. In effect, the stop-loss prevents excessive loss of capital should the market go the wrong way.

The stop-loss price may change when a new position is entered or as the current price changes (see below). The current stop-loss price is displayed to a user-agent.

A stop-loss deal may occur at any time, even outside market hours. The assumption is that a position that is kept open outside market hours is handled by a colleague in the American or Far East markets who will deal appropriately if the stop-loss is hit. Should this happen, preferably no further change in position occurs until the local market opens once again.

The concept of stop-profit is associated with that of stop-loss. The stop-loss price starts to move in parallel with the current price once a trading model has achieved a potential profit (3% or slightly less) since entering the latest position. In other words, being in a situation whereby the model could realize such a gain by immediately entering a neutral position causes the stop-loss price to start moving. The difference between the stop-loss and current prices is kept constant as long as the current price continues moving in a direction that increases the potential profit of the open position. Thus, the stop-loss price moves as a ratchet in parallel with the current price. The stop price is allowed to move only during open hours. It is never adjusted when the market is closed.

A preferred model then enters a neutral position if it detects prices slipping backwards. This allows a model to save any profit it has generated rather than lose it when the market abruptly turns. This one-directional movement of the stop-loss price allows the model to capitalize on a price trend.

A preferred trading model may thus make a deal recommendation in two distinct ways. One, the gearing calculator may make a recommendation that is subsequently authorized by the deal acceptor. Two, hitting the stop-loss price activates the stop-loss detector.

Whichever way a deal comes about, the opportunity catcher is activated. The opportunity catcher manifests itself on a user-agent display as an eye-catching signal for the FX dealer to buy or sell according to the recommendation.

While a dealer is actively dealing, the opportunity catcher in the trading model collects the best price with which to deal, either the highest bid price if going from a longer position to a shorter one or the lowest ask price if going from a shorter position to a longer one. This search for the best price lasts for two or three minutes depending on the currency, the assumption being that a quoted price has a useful lifetime of about two or three minutes even if it is superseded by later quotes.

After the two- or three-minute search period, a second signal appears on the user-agent display signifying that the trading model has made a simulated deal using the best price found by the opportunity catcher. The FX dealer then concludes his deal-making activities and waits until the trading model produces another recommendation.

Preferably, the opportunity catcher is not activated for a stop-loss deal occurring outside market hours. In such a case the trading model deals directly. A human trader following the model should then make a corresponding deal for himself as quickly as possible.

In a preferred embodiment, a Book-keeper makes simulated deals on behalf of the trading model. It keeps track of all deals that have been made and evaluates statistics demonstrating the performance of the trading model.

An important variable used by a Book-keeper is the average price $\bar{p}$ paid for achieving the current gearing. After a new deal with index t, this quantity depends on the type of transaction as follows:

$$\bar{p}_t \equiv \begin{cases} \bar{p}_{t-1} & \text{if } |g_t| < |g_{t-1}| \wedge g_t g_{t-1} > 0 \\ g_t\left[\frac{g_t - g_{t-1}}{p_t} + \frac{g_{t-1}}{\bar{p}_{t-1}}\right]^{-1} & \text{if } |g_t| > |g_{t-1}| \wedge g_t g_{t-1} > 0 \\ p_t & \text{if } g_t g_{t-1} < 0 \vee g_{t-1} = 0 \\ \text{undefined} & \text{if } g_t = 0 \end{cases} \quad (1)$$

where g_{t-1} and g_t are the previous and current gearings, respectively, p_t is the current transaction price, and $\bar{p}_{t-1}$ the average price before the deal. In the initial case, when the current gearing is neutral, the average price $\bar{p}$ is not yet defined.

The average price $\bar{p}$ is needed to compute a quantity central to a preferred trading model—the return of a deal:

$$r_t \equiv (g_{t-1} - g_t)\left(\frac{p_t}{\bar{p}_{t-1}} - 1\right) \quad (2)$$

There are deals with no returns: those starting from a neutral gearing, $g_{t-1}=0$, and those increasing the absolute value of the gearing while keeping its sign. In these cases, eq. (2) does not apply (whereas eq. (1) applies to all deals).

Both of the above equations allow the computation of a set of other quantities that are important for the different trading rules and for the performance evaluation of the models and which are computed in the book-keeper. These are the following:

the current-return r_c. This is the unrealized return of a transaction when the current position is off the equilibrium ($g_t \neq 0$). If p_c is the current market price required for going back to neutral, generalizing eq. 2 yields the current-return,

$$r_c \equiv g_t\left(\frac{p_c}{\bar{p}_t} - 1\right) \quad (3)$$

the maximum return when open is the maximum value of r_c from a transaction t to a transaction t+1 reached during opening hours, the minimum return when open is the minimum value of r_c from a transaction t to a transaction t+1 reached during opening hours,

the total return is a measure of the overall success of a trading model over a period T and is simply equal to,

$$R_T \equiv \sum_{i=1}^{n} r_i \tag{4}$$

where n is the total number of transactions that generated a return during the period T. The total return expresses the amount of gains made by a trader who would always invest up to his initial capital or credit limit in his home currency.

the cumulated return is another measure of the overall success of a trading model wherein the trader always reinvests up to his current capital including gains or losses,

$$C_T \equiv \prod_{i=1}^{n} (1 + r_i) - 1 \tag{5}$$

This quantity is slightly more erratic than the total return.

the maximum drawdown D_T over a certain period $T = t_E - t_0$. It is defined as

$$D_T \equiv \max(R_{t_a} - R_{t_b} | t_0 \leqq t_a \leqq t_b \leqq t_E) \tag{6}$$

where R_{t_a} and R_{t_b} are the total returns of the periods from t_0 to t_a and t_b respectively.

the profit over loss ratio gives an idea of the type of strategy used by the model. Its definition is:

$$\frac{P_T}{L_T} \equiv \frac{N_T(r_i \mid r_i > 0)}{N_T(r_i \mid r_i < 0)} \tag{7}$$

where N_T is a function that gives the number of elements of a particular set of variables under certain conditions during a period T.

The book-keeper can also compute other quantities, but they are not central to the invention. Another book-keeper function is to save the trading history of a particular model so it can be retrieved at any time by a user-agent.

In a preferred embodiment, status and simulation messages are issued as appropriate to keep the FX dealer informed of the trading models' states and of any likely deals.

Status messages typically give advance warning of the market opening or closing, or notification if the market is already closed. These messages also indicate whether the supply of raw price data from the commercial market quoters to the trading models has been disrupted for any reason.

Simulation messages give advance notification of a probable deal. A deal is considered to be probable if the market continues its present trend for half an hour or so.

Preferred trading models are event-driven. This means that each model is preferably implemented in terms of defined actions that are triggered when some external event occurs. An external event might be the receipt of a new price from the commercial quote-vendors, or it might be a timer event such as a local market opening or closing.

This event-driven characteristic of the trading models suggests an implementation in terms of a data-flow paradigm. Reception of a new price or a timer event causes some defined actions to be triggered and these, in turn, cause other actions to be triggered. An action may, for example, be a calculation with a new input price, generation of a new trading recommendation, or the transfer of computed data to user-agents. This process continues recursively, one action triggering others, until all dependent actions are executed.

Preferred trading models are therefore preferably implemented in a computer language (such as C++) that supports such a data-flow paradigm.

An input stream to software of a preferred embodiment provides price data and other input in the form of variables being updated. Dependent expressions are then recursively evaluated before new price data is subsequently provided.

In a preferred trading model environment, rules provide a way of ensuring that the separate components such as the opportunity catcher, deal-acceptor, simulation, and so on do not interfere with one another. For example, simulation is disabled while the opportunity catcher is running. Such separation of activities is implemented in terms of meta-rules. These are rules that monitor the activity of the other rules and expressions within the trading model code. Depending on what is happening inside the trading models, the meta-rules set priorities of other rules thereby enabling or disabling other components. In short, inclusion of meta-rules prevents the various trading model components from chaotically disrupting one another.

As noted above, the indicators form a pivotal component of the gearing calculator and hence of the dealing decision process. An indicator is a function only of time and the price history. It summarizes relevant information of the past price movements in the form of a single variable.

The indicator definition needs a time scale on which the price history is analyzed. The usual physical time scale t is inappropriate. For example, the market perceives a price history differently when it covers two working days in one case and two weekend days (Saturday and Sunday) in another, even if the price curves against t have the same form in both cases. A business time scale, related to business activity, is a better choice for analyzing price history.

Changing the time scale can be viewed as an opportunity to introduce some of the "fundamentals" that are missing in conventional time series analysis. These fundamental economic variables certainly have an influence on price movements, but their individual and combined effects are difficult to isolate and thus to replicate. Their impact can be better seen, however, if one views price movements as indelible "footprints" left by these variables. A time scale based on price movements may therefore be used to capture the tracks left by economic factors.

Conventional business-time scales are quite simple: they omit inactive periods such as weekends and count only the business hours. A subtler "modified business time" scale is achieved by expanding periods of high activity and contracting but not omitting periods of low activity.

The 24 hour FX market indeed has periods of low activity (for example, the noon break in East Asia) and high activity (for example, when the European and American business hours overlap). In (Müller 1990), a modified business time scale called σ-time was introduced, modeling the daily and weekly fluctuations of FX activity. This σ-time is used in preferred embodiments of the subject invention as the time scale for indicator computations.

The σ-time model considers business activity in terms of statistical means of absolute price movements (a volatility measure). Hourly or daily FX transaction volume figures would be another alternative for measuring business activity, but such data is generally unavailable. The discussion in Dacorogna, M. M., Müller, U. A., Nagler, R. J., Olsen, R. B., & Pictet, O. V., *A geographical model for the daily and weekly seasonal volatility in the FX market*, Journal of International Money and Finance, 12(4), 413-438 (1993) (Dacorogna 1993) argues for using price movements in a business time scale.

FX activity exhibits distinct daily and weekly seasonal heteroskedasticity, which can be attributed to the changing presence of traders on the FX markets. The σ-time model of a preferred embodiment defines the activity a(t) as the sum of three activity functions $a_k(t)$, corresponding to three sub-markets of the worldwide FX market that are geographically centered in East Asia, Europe and North America. More precisely, the $a_k(t)$ are functions of the time of day and of the day of the week (working day or weekend day). The σ-time is the integral of a(t) over physical time t.

When the three sub-markets model is used, hourly absolute price changes lose their seasonality when analyzed against σ-time instead of t.

In addition to the daily and weekly seasonal heteroskedasticity, FX prices also exhibit non-seasonal heteroskedasticity in the form of clusters of high or low volatility. The σ-time model of a preferred embodiment does not take these into account. Therefore, σ has a regular pattern with a period of 1 week, and may also be determined for the future. It is calibrated so that one full week in σ-time corresponds to one week in physical time t. Hence, it can be measured in the same units as t: seconds, minutes, hours, days, weeks or years. Further details on the modified business time σ can be found in (Dacorogna 1993).

In a preferred embodiment, one trading model indicator is defined as a momentum—the difference of the current logarithmic middle price x and its moving average (MA) computed in σ-time:

$$m_x(\Delta\sigma_r, \sigma) = x - MA_x(\Delta\sigma_r, \sigma), \tag{8}$$

where $\Delta\sigma_r$ is the MA range, a measure related to the length of past time covered by the MA. The moving average is a useful tool for summarizing the past behavior of a time series at any time point. An alternative definition (not used herein) of a momentum would be the difference of two moving averages with different ranges.

A special moving average operator (described below) is used in a preferred embodiment. Among the MAs, the exponentially weighted moving average (EMA) plays an important role. Its weighting function declines exponentially with the time difference of past observations from the present. Sequential EMAs along a time series are straight-forward to calculate with the help of a recursion formula; this is more efficient than the computation of differently weighted MAs.

Although moving averages are well known for homogeneous time series (for instance, see GRANGER, C. W. J & NEWBOLD, P., FORECASTING ECONOMIC TIME SERIES (1977)), their formalism is much less developed for continuous time series, to which preferred embodiments of the subject invention are addressed. Tick-by-tick data is not equally spaced over time (neither t nor σ), but can be interpolated in the time intervals between the price quotes. A preferred embodiment uses linear interpolation in σ-time as the interpolation method. In an alternative embodiments, the previous quote is taken as a function value for the whole interval.

Through interpolation, a continuous time function x(σ) is arrived at, for which the EMA is an integral:

$$EMA_x(\Delta\vartheta_r, \vartheta_c) \equiv \frac{1}{\Delta\vartheta_r}\int_{-\infty}^{\vartheta_c} e^{\frac{\vartheta_c - \vartheta}{\Delta\vartheta_r}} x(\vartheta)\,d\vartheta, \tag{9}$$

where c is the index of the current time series element and the range $\Delta\sigma_r$ is the center of gravity of the EMA weighting function. This EMA can be computed efficiently with a recursion formula:

$$EMA_x(\Delta\sigma_r, \sigma_c) = \mu EMA_x(\Delta\sigma_r, \sigma_{c-1}) + (1-\mu)x_c + (\mu - \nu)\Delta x_c, \tag{10}$$

where Δx_c is defined as $x_c - x_{c-1}$. For a time series representing a function with all types of interpolation,

$$\mu = e^{-\alpha} = e^{-\frac{\Delta\vartheta_c}{\Delta\vartheta_r}}, \tag{11}$$

where the time interval $\Delta\sigma_c$ is defined as $\sigma_c - \sigma_{c-1}$; and

$$\nu = \frac{1 - e^{-\alpha}}{\alpha} = \frac{1-\mu}{\alpha} \tag{12}$$

This is valid under the assumption of linear interpolation between subsequent series elements. If the function value within such an interval is assumed to be constant and equal to the preceding series element, then the recursion formula also applies by setting ν=1 instead of equal to the right-hand side of eq. (12).

Together with the recursion formulae, the initial EMA value must also be specified. There is usually no information before the first series element x_1, so it thus becomes the natural choice for the initialization:

$$EMA_x(\Delta\sigma_r, \sigma_1) = x_1. \tag{13}$$

The error made by this initialization declines by a factor $e^{-(\sigma_c - \sigma_1)/\Delta\sigma_r}$. A part of the data must therefore be reserved to build up the EMAs, so that this error can be made small.

The exponential weighting function with its steeply increasing form gives a strong weight to the recent past, which may contain noisy short-term price movement structures. These can lead to inaccurate trading signals. For this reason other, less-peaked MA weighting functions are often more successful.

Different MA weighting functions can be obtained by repeated application of the EMA operator. The EMA operator yields a result of the same mathematical nature as its input: a time series. Because of this property, the EMA operator can be applied iteratively. The result at each stage is a moving average (MA) with a weighting function more complicated than a simple exponential function:

$$EMA_x^{(n)} = EMA(EMA_x^{(n-1)}(\Delta\sigma_r, \sigma_c)) \tag{14}$$

where $EMA_x^{(0)}(\Delta\sigma_r, \sigma_c) = x_c$. The order of the EMA operator (how many times the simple EMA operator is applied) determines the kind of weighting applied to the past prices.

In a preferred embodiment, the moving average is the mean of EMAs of different orders:

$$EMA_x^{(m,n)} \equiv \frac{1}{n-m+1} \sum_{i=m}^{n} EMA_x^{(i)}(\Delta\vartheta_r, \vartheta) \tag{15}$$

This MA has a weighting function with the center of gravity at $(m+n)\Delta\sigma_r/2$. The weighting function has the following properties: (1) a low weight for the most recent past; (2) a wide, plateau-like weight maximum at time points around the center of gravity (no sharp peak); and (3) a rapidly but smoothly declining weight for more distant times.

The indicator is a momentum in the sense of eq. (8):

$$I_x = \frac{1}{s(\Delta\vartheta_r, m, n)} (x - EMA_x^{(m,n)}), \tag{16}$$

where $s(\Delta\sigma_r, m, n)$ is a scaling factor.

Trading signals are given when the indicator I_x crosses over certain threshold values. There are two types of trading signals. The first type are trend signals. These are recommendations to follow the trend indicated by I_x. Good threshold values are somewhat low so that trends can be detected relatively early. A "buy" signal is given when I_x exceeds such a value, a "sell" signal when I_x exceeds the corresponding negative value.

The second type of trading signals are overbought/oversold signals. These are contrarian instructions—to stop following an old trend, or to take the opposite position. Good threshold values for these signals are high, at levels that are only exceptionally reached by I_x. "Overbought" refers to high positive values, "oversold" to high negative ones.

Preferred trading models use several threshold values of both types. Different threshold values are used in different situations, depending on the current position of the trading model and other internal variables of the decision-making process. Some trading signals recommend taking a new position with full gearing, others a new position with only half gearing, again depending on certain internal conditions of the decision-making process. In many cases, the decision-making process does not follow a trading signal because another decision rule has precedence, as discussed above.

Trading models used in a preferred embodiment are designed for use in a real-time service. However, for testing purposes, for tuning the trading models and for producing statistics, a more stable, reproducible environment is required. Thus, preferred trading models can be driven from a database of historical price data. That is, the defined actions are triggered as though they were being fed from a real-time stream. The trick here is to use the historical price database to form a stream of combined price and timer events. In other words, the timer events are simulated from the historical prices.

This interchangeability between historical and real-time modes is invaluable for the trading model developer. Once a new gearing calculator has been produced, for example, the developer can test it using historical price data. Without making any code changes, he can then switch the trading model into real-time mode so that it then reacts to new price data as they become available.

Optimizing trading models to have minimum overfitting is a difficult task. Overfitting means building the indicators to fit a set of past data so well that they are no longer of general value: instead of modeling the principles underlying the price movement, they model the specific price moves observed during a particular time period. Such a model usually exhibits a totally different behavior or may fail to perform out-of-sample; that is, with data that was not used for the optimization process. To avoid overfitting during optimization, a good measure of the trading model performance is needed, and robust optimization and testing procedures must be used. A strict division of the available historical data used in the different phases of optimization and testing is also important.

The first step is to define a value describing the trading model performance in order to minimize the risk of overfitting in the in-sample period and also to be able to compare different trading models against one other. The performance of a trading model cannot be judged by its total return only; the relevant issue is the overall risk profile. We need a measure of the trading model performance that takes into account the following requirements: (a) the best total return; (b) a smooth, almost linear increase of the total return over time; (c) a small clustering of losses; and (d) no bias towards low frequency trading models.

A measure frequently used to evaluate portfolio models is the Sharpe index. Unfortunately, this measure neglects the risk due to unrealized losses while the model stays in one position. It also introduces a bias in favor of models with a low dealing frequency. Furthermore the Sharpe index is numerically unstable for small variance of returns and cannot consider the clustering of profit-and-loss trades. This measure clearly does not fulfill the above requirements. In the description below, we present a measure of the trading model performance that is risk-sensitive and indicates the stability of returns.

As a basis of our risk-sensitive performance measure, a new trading model return variable R is defined to be the sum of the total return R (eq. (4)) and the non-realized current return r_c (eq. (3)). The variable R is more continuous over time than R and reflects the additional risk due to non-realized returns. Its change over a test time interval Δt is termed $X_{\Delta t}$:

$$X_{\Delta t} = \tilde{R}(t) - \tilde{R}(t-\Delta t), \text{ where } \tilde{R}(t) = R(t) + r(t) \tag{17}$$

We can make N independent observations of $X_{\Delta t}$ within a total test period ΔT, where

$$\Delta t = \frac{\Delta T}{N} \tag{18}$$

A risk-sensitive performance measure of the trading model comparable to the average return can be derived from the utility function formalism of Keeney, R. L., & Raiffa, H., Decision with Multiple Objectives: Preferences and Other Tradeoffs (1976) (hereinafter Keeney 1976). In a preferred embodiment, we assume that the variable $X_{\Delta t}$ is stochastic and follows a Gaussian random walk about its mean value $\overline{X}_{\Delta t}$ and that the risk aversion C is constant with respect to $X_{\Delta t}$. The resulting utility $u(X_{\Delta t})$ of an observation is $-\exp(-CX_{\Delta t})$, with an expectation value of $\bar{u} = u(\overline{X}_{\Delta t}) \exp(C^2\sigma^2_{\Delta t}/2)$, where $\sigma^2_{\Delta t}$ is the variance of $X_{\Delta t}$. This expectation of the utility can be transformed back to a variable termed the effective return: $X_{eff,\Delta t} = -\log(-\bar{u})/C$. The following definition is obtained:

$$X_{eff,\Delta t} = \overline{X}_{\Delta t} - \frac{C\sigma^2_{\Delta t}}{2} \tag{19}$$

The risk term $C\sigma^2_{\Delta t}/2$ can be regarded as a risk premium deducted from the original return. The variance is computed from the sample:

$$\sigma^2_{\Delta t} = \frac{N}{N-1}\left(\overline{X^2_{\Delta t}} - \overline{X}^2_{\Delta t}\right) \tag{20}$$

For a particular time horizon Δt, the variable $X_{eff,\Delta t}$ is our new measure of trading model performance including risk. Unlike the Sharpe index, this measure is numerically stable and can differentiate between two trading models with a straight line behavior ($\sigma^2_{\Delta t}=0$) by choosing the one with the best average return. Moreover, the definition given in eq. (19) gives the book-keeping details of a trade a smaller role than the Sharpe index because $X_{\Delta t}$ also contains elements of the current return and is computed at regular time intervals Δt.

The measure $X_{eff,\Delta t}$ still depends on the size of the time interval Δt. It is hard to compare $X_{eff,\Delta t}$ values for different intervals. The usual way to enable comparisons between different intervals is through annualization: multiplication by the annualization factor, $A_{\Delta t}$=1 year/Δt,

$$X_{eff,ann,\Delta t} = A_{\Delta t} X_{eff,\Delta t} = \overline{X} - \frac{C}{2} A_{\Delta t} \sigma^2_{\Delta t} \tag{21}$$

where $\overline{X}$ is the annualized return, no longer dependent on Δt. In the second term of the last form of eq. (21), we find the factor $A_{\Delta t}\sigma^2_{\Delta t}$. This factor has a constant expectation, independent of Δt, if the return function R(t) is assumed to follow a Gaussian random walk in addition to a linear drift. For such a return function (our reference case), we introduce the condition that the expectation of $X_{eff,ann,\Delta t}$ must not depend on Δt. This condition is fulfilled only if the risk aversion C is constant, that is, independent of Δt. Annualized effective returns $X_{eff,ann}$, computed for different intervals Δt by eq. (21) with a constant C value, can therefore be directly compared.

This measure, though annualized by eq. (21), still has a risk term associated with Δt and is insensitive to changes occurring with much longer or much shorter horizons. To achieve a measure that simultaneously considers a wide range of horizons, we introduce a weighted average of several $X_{eff,ann}$ computed with n different time horizons Δt_i, and thus take advantage of the fact that annualized $X_{eff,ann}$ can be directly compared:

$$X_{eff} = \frac{\sum_{i=1}^{n} w_i X_{eff,ann,\Delta t_i}}{\sum_{i=1}^{n} w_i} \tag{22}$$

where the weights w_i can be chosen according to the relative importance of the time horizons Δt_i and may differ for trading models with different trading frequencies. Substituting $X_{eff,ann}$ by its expression (eq. 21), X_{eff} becomes

$$X_{eff} = \overline{X} - \frac{C}{2}\sum_{i=1}^{n} \frac{w_i A_i \frac{\sigma_i^2}{\Delta t_i}}{\sum_{i=1}^{n} w_i} \tag{23}$$

where the variance σ_i^2 is computed with eq. 20 for the time horizon Δt_i, and A_i is the corresponding annualization factor (=1 year/Δt_i). Because $\sigma_i^2 \geqq 0$, we have $X_{eff} \leqq \overline{X}$. By empirically balancing risk and return of some test trading models, we found values between 0.08 and 0.15 to be reasonable for C.

By adopting this new measure, we depart from the formal utility function theory defined by (Keeney 1976). That theory is based on the additivity of utilities, but in eq. (22), we average effective returns (which are non-linear functions of utilities). Nevertheless, we choose this definition because we do not see the utility of each horizon as a component of a meta-utility but rather as representing a typical segment of the market. If one of these segments endures a bad phase, its influence on the overall outcome need not be over-proportional, which would be the case if we kept the formalism of additive utilities.

In the discussion of eq. (21), we have shown that the risk aversion C has no systematic dependence on the horizon Δt_i. However, dealers using a trading model might perceive differently the risks of various horizons. We might introduce special C values for individual horizons according to their trading preferences. However, we can achieve an equivalent effect by changing the weights w_i, which are already differentiated for each horizon. These weights reflect the importance of the horizons in terms of the risk sensitivity associated with each horizon.

Specifically, in a preferred performance measure, a standard weighting function is used which determines the weights w_i and thus the relative importance of the different horizons:

$$w(\Delta t) = \frac{1}{2 + \left(\log \frac{\Delta t}{90 \text{ days}}\right)^2} \tag{24}$$

The weight maximum is set to a 90 days horizon in order to give sufficient importance to the short horizons compared with the long ones. This weighting function is designed to be applied to horizons Δt_i in a roughly geometric sequence.

An approximately geometric sequence of n horizons Δt_i is chosen with the following construction. Once a testing period (full sample of size ΔT) has been established, it is divided by 4. If this division results in a time horizon longer than 2 years, then the result is divided by 2 and so on, until a horizon At_1 strictly shorter than 2 years is reached. We limit this highest horizon because dealers usually close their books after one year and are less sensitive to return clustering on longer horizons. The next horizon is obtained by a division by 2 of the previous one and so on until a last horizon between 5 to 10 days is reached; this shortest horizon is then forced to be Δt_n=7 days. All horizons are truncated to full days. If there is no integer multiple of a resulting Δt_i that exactly covers the full sample, then the first analyzed interval at the start of the full sample is extended accordingly. The exact Δt_i values to be inserted in eqs. (23) and (24) are the results of eq. (18).

TABLE 1

Δt_i	594	297	148	74	37	18	7
w_i	0.086	0.139	0.212	0.234	0.171	0.104	0.056
σ_i	21.92%	11.80%	7.81%	5.37%	3.64%	2.41%	1.47%
$\overline{X}_{\Delta t_i}$	36.83%	18.42%	9.21%	4.60%	2.30%	1.12%	0.43%
$X_{eff,ann,\Delta t_i}$	7.88%	14.09%	15.15%	15.56%	16.12%	16.75%	16.99%

Typical results for the performance measure according to each horizon. The horizons Δt_i are given in days, the weights are normalized to one.

To illustrate the effective return computation, Table 1 displays typical results for a preferred trading model for the German mark against the U.S. dollar, tested on six and a half years (March 1986 to September 1992) of data. The analyzed horizons Δt_i, the weights w_i; the variances σ_i^2, the average returns $\overline{X}_{\Delta t_i}$, and the annualized effective returns $X_{eff,ann,\Delta t_i}$ (see eq. (21)) are presented in Table 1. The average yearly return of this run is $\overline{X}$=22.65% and the effective yearly return (computed according to eq. 23 with C=0.10) is X_{eff}=14.91%. The yearly return is preferably reduced by a "risk premium" of about a third of the original value.

In order to optimize and test trading models, available historical data is preferably split into three different periods. The first period is used to build up the indicators, the second one is for the optimization of the trading model parameters, and the third for selecting the best trading models. The continuously increasing data set collected from the quote-vendors is reserved for real-time (ex ante) testing.

The build-up period generally contains more than ten years of daily data, which is used to update long-term indicators. A few weeks of tick-by-tick data is also needed to build up short-term indicators. In the disclosed example, the end of that period is 1 Mar. 1986 for the USD/DEM, USD/JPY, GBP/USD and USD/CHF major exchange rates and 1 Dec. 1986 for the USD/NLG, USD/FRF and USD/ITL minor exchange rates.

To optimize the trading model parameters, a three-year in-sample period starting just after the end of the build-up period is used. We assume that three years of tick-by-tick data is sufficient to optimize the trading models. The end of the latter period is 1 Mar. 1989 for the major exchange rates and 1 Dec. 1989 for the minor ones.

The in-sample performance refers to the period for which the model is developed and optimized. Preferably, different formulae for computing the indicators, different trading strategies, and different parameters are tested until the model is found to achieve the optimal return. This process involves thousands of simulation runs. To select the best parameter set, choose the solution in the parameter space that corresponds to the larger effective return. If different solutions with comparable returns are found, choose the solution that is not too sensitive to a small change of the parameter set. In some cases, a period of three years in-sample data is insufficient because some volatility clusters can be as large as the optimization period.

Because extending the available historical tick-by-tick data is not possible, further restrictions are enforced on the choice of possible parameters. A solution is considered only if a similar cluster of good results exists for different FX rates at about the same place in the parameter space. This implies that various parameters must be scaled with the volatility of the individual FX rates in order to make them comparable between different rates.

The result of the in-sample optimization process is a small set of diverse trading models sharing a high effective return. Only these models are analyzed in the out-of-sample performance test. The out-of-sample period covers the remainder of the historical data, from the end of the in-sample period until the end of September 1991. A trading model is considered to be valid if the in-sample and out-of-sample periods yield a similar quality. Otherwise, the model is rejected. To avoid overfitting problems, the out-of-sample results are never used to select between similar models.

Since these models were developed we can now test them on a significant ex ante period. (We term ex ante period a period of time that has never been used in conjunction with model development).

A private investor may use a margin account to invest a multiple of his equity in the market. Such a leverage factor has an overriding impact on the total return. But the higher the leverage the higher too the risk of large draw-downs. Working with a high leverage factor may lead to margin calls during periods of draw-down, thus jeopardizing an investor's trading strategy. We have therefore developed our models without leverage. This should be borne in mind while evaluating the results.

The most important performance measure is the effective return, as defined in eq. (23). While the total return may mask a considerable risk introduced by high volatility of return, the effective return is risk-sensitive: the higher the volatility of return the lower the effective return. In other words, high effective returns indicate highly stable returns.

The maximum draw-down, defined by eq. (6), is also a significant performance indicator for a trading model user. It is the largest loss from a local maximum to a minimum of the total return curve within the test period. The maximum draw-down is a measure of risk, similar to the difference between the total return and the effective return.

Another important performance measure is the profit/loss ratio, defined in eq. (7) to be the ratio between the total number of profitable deals and the total number of losing deals. The preferred models let profits run and cut losses early; they are almost always in an open position, and go neutral only in case of stop-loss or save profit. Such a strategy is bound to result in a lower profit/loss ratio than would be achieved by less active models.

The first quality test of a new trading model is to compare the in-sample and out-of-sample performances. The performances in the two periods must be of nearly equal quality for it to merit further consideration. To receive sufficient statistical information, a model must be tested for an extended out-of-sample period and for many different time series. If the average out-of-sample performance is significantly smaller, it may mean that the model was overfitted. However, a much lower average volatility in the out-of-sample period can explain a reduced performance of the model, in that there are fewer opportunities for profitable trading. The performance difference, however, must not be too large.

An example of a total return curve over all three periods, in-sample, out-of-sample, and ex ante, is given in FIG. 2. The results of Table 2 (see end of this description) belong to the same example.

Table 2 shows the in-sample period (1.3.86 to 1.3.89 for USD/DEM, USD/JPY, GBP/USD and USD/CHF, and 1.12.86 to 12.3.89 for USD/NLG, USD/FRF and USD/ITL), the out-of-sample period (1.3.89 to 1.10.91 and 1.12.89 to 1.10.91 respectively) and the ex ante period (1.10.91 to 4.9.92). The performance indicators shown are: the annualized return ($\bar{X}$), the risk-sensitive return ($X_{\Delta T}$eff), the maximum draw-down (D) over the sample, and the ratio of the number of profitable deals to the number of deals that resulted in a loss (P/L).

In Table 2 we compare the in-sample, out-of-sample and ex ante performances of the preferred trading models. We observe that the average values of these quantities are similar to one another. This indicates the overall stability of our trading models and that the achieved profits do not result from model overfitting.

The individual ex ante values do, however, vary. We attribute this variance to the relative short ex ante testing period. The USD/JPY model has also performed less adequately than the others. This is because of the relatively low volatility in USD/JPY trading over the ex ante testing period.

In table 3, the performance of preferred trading models is compared to that of a more conventional 20-day moving-average model. Both models share the same environment; they have the same opening hours, book-keeping rules and so on. They differ in the indicators and the rules on how to use these. The 20-day moving-average model uses the difference between the current logarithmic middle price and a conventional 20-day moving-average as its indicator, rather than eq. (16). In this simple model, the indicator threshold level is not varied and there are no overbought/oversold signals.

The preferred trading models surpass the 20-day moving-average model in all respects: they produce much higher total returns and effective returns and, except for one rate, much lower drawdowns.

When investing in equity, it is common practice to spread the risk over different stocks. The same strategy may also be applied for currency trading. For example, the volatility may suddenly decline in a particular market and it may then not be possible to make any real money in this market for some period of time. We have tested such portfolios with a stable and equal capital distribution among the different models. The results (see Table 4) show clearly that the maximum drawdown and the drawdown period are significantly lower than those of the individual models. In line with portfolio theory, the portfolios with the lowest values (a little above 5% for the maximum drawdown and 22 (21) days for the drawdown period) are the ones comprising the most diversified currencies. At the same time these portfolios generate an impressive annualized return. Compared to the individual models, therefore, a portfolio achieves a more stable return with lower risk.

A second preferred embodiment comprises a trading model, designated herein as Trading Model 60, or TM60, that uses different sub-models, each with its own time horizon, thus modeling the views and actions of different trader groups or market components.

In this second embodiment, preferred software supports general trading model functions such as opening hours, stop-loss and profit objective, and internal model performance bookkeeping including realistic transaction costs. TM60 is a genuine intra-day real-time model that analyzes every tick available from the market to construct its trading signals.

Trading Model 60 is a multi-horizon model. It is using three different sub-models simultaneously, each with its own time horizon. The idea behind this model structure is the hypothesis of a heterogeneous market with trader groups that differ in their views and actions in several respects, most notably in their time horizons. The trading model eventually constructs its recommendation based on the view of all these components together. A preferred TM60 embodiment has three market components; the centers of gravity of the past analyzed by the model indicators are of 1 day, 4 days, and 16 days (measured in σ-time, not just working days, as explained below). The trading model eventually constructs its recommendation based on the view of all these components together.

Each sub-model has its own indicator. The indicator is constructed from the variable x, which is here a very short-term (range: a few seconds) exponential moving average (EMA) of the logarithmic middle price. This very short-term smoothing eliminates a large part of the short-term noise that can be attributed to the different buying/selling intentions of the price quote suppliers.

The indicator I is a "momentum", that is the difference between x and a specially weighted moving average of x:

$$I = x - \mathrm{EMA}_x^{(2,4)}$$

where $\mathrm{EMA}^{(2,4)}$ is computed through the repeated application of the EMA operator, as defined and explained above (see equation (15)). The superiority of the $\mathrm{EMA}^{(2,4)}$ which is neither a plain moving average nor an exponentially weighted one has been established.

The time scale chosen for the EMA computations is σ-time as defined above, in order to prevent the intra-daily seasonality of volatility from negatively affecting the trading behavior. The depth of the past considered in the indicator computation corresponds to the time horizon of the sub-model. A normalized indicator is then computed, that is the indicator I is divided by the square root of its own long-term moving variance.

This normalized indicator is the basis for constructing the sub-model signal. The signal of the sub-model flips its sign when the indicator crosses a break level. The break level is a small positive value for positive (buy) signals and the inverse negative value for negative (sell) signals. The absolute value of the break level is not constant: after each signal flipping, it is raised by a fixed amount, the "break penalty." Afterwards, it is fading with time, until a minimum level is reached. This mechanism ensures a certain adaptivity to market situations: the large number of signal flippings in certain sideways markets is reduced by the raising of the break level.

As the microstructure of a market changes over the long term it is quite possible that the numerical values of the indicators may drift to a lower or higher numerical territory. Because models tend to function on the basis of thresholds, this drift effect can cause models to become overactive or under-active. A preferred method of guarding against this is to normalize the raw indicators over their long-term average and to assign thresholds based on the statistical properties of the indicator distribution instead of absolute values of the raw indicators. For example, a raw indicator might be said to have crossed a threshold when it increases beyond a certain percentile of its long-term distribution. This effect is achieved by normalization.

The sign of the signal is determined by the indicator breaks explained above. The magnitude of the signal is determined by other factors:

- Signal freshness: the signal is strong when the indicator just crosses the break level; it is weak when the indicator starts to move back to the opposite side.
- Overbought/oversold: the signal is modulated to become weak when the current trend (since the last indicator break) has become exceptionally strong.
- Spread correction: if the bid-ask spread is large (non-negligible) as compared to the (non-normalized) indicator RMS value, the signal is weakened; following this signal might lead to too high transaction costs as compared to the profit. This effect favors long-term sub-models over short-term ones.
- Market condition: the signal is modulated to be weak when the market is a "noisy" sideways market. This is measured through a ratio of recent volatilities: the volatility of short-term price moves measured on a fine time grid, divided by the volatility of long-term price moves measured on a coarse time grid. A low ratio means a trending market with little noise. Statistical studies have shown that not only volatilities are persistent (clustered), but also their ratios.

The modulation of the signal by these factors leads to a more flexible model behavior and a certain adaptivity to current market states and behaviors. The resulting sub-model signal summarizes the "opinion" of the sub-model or the corresponding market component.

The joint TM60 signal is simply the sum of all sub-model signals. TM60 produces a trading recommendation (e.g., "go long +1.0", "go neutral", "go short −0.5") whenever the joint signal crosses over certain fixed threshold values. If the trading model environment forbids the production of a trading recommendation (e.g., overnight, outside the trading hours), there will be no trading recommendation, even if the level of the joint TM60 signal would imply one.

The stop-loss and stop-profit mechanism of the trading model environment over-rides the trading recommendations of the normal TM60 algorithm. They constitute a "safety net"; normal trading recommendations are much more frequent than those from a stop-loss or stop-profit. The profit objective value (3%) is similar to those of other trading models discussed above. The stop-loss value (1.8%) and especially the moving loss limit (the moving stop-loss after reaching the profit objective, 0.1%) are smaller than those of other models discussed above, leading to faster execution. These values are the same for all FX rates in the TM60.

A perfectly adaptive trading model that would constantly adapt its behavior according to some built-in analysis would be able to adapt to any FX rate, minor or major, EMS or non-EMS, and even to any financial price series. For such a model, no rate-specific model optimization would be needed; all FX rates would share the same, hard-wired parameters.

Trading Model 60 is adaptive to a certain degree but not perfectly. Therefore, it comes in three versions that share the algorithm but differ in some parameters:

1. A version with a high transaction frequency for freely floating rates with high volatility and liquidity and low bid-ask spreads. Typical application: USD/DEM, DEM/JPY.
2. A version with a stronger spread correction and other parameters leading to a lower transaction frequency. Typical application: minor rates and European DEM cross rates such as DEM/SEK.
3. A version with longer horizons of the sub-models (4 days, 15.6 days and 61 days instead of 1 day, 4 days and 16 days), also with low transaction frequency, for all (not too minor) rates against the USD, except USD/DEM. Typical application: AUD/USD, USD/CHF.

Preferably, careful optimization and testing procedures are maintained, so as to minimize the "overfitting" of models to particular rates and time periods. For the same reason, all FX rates have a TM60 parameter choice strictly following one of these three versions, even if it is possible to get a higher historical performance through another parameter fit.

FIG. **3** depicts the USD-DEM autocorrelation function of the price changes (with sign) with lags in units of 1 day in physical time up to a lag of 100 days. The horizontal lines represent the 95% confidence interval of a Gaussian random process. Typical auto-correlation structure for price changes (with sign) for FX rates is shown. Note that the correlation of the price change with sign is well within the significance range. This correlation becomes significant only when analyzed conditional to volatility.

A third preferred embodiment comprises a trading model, designated herein as model 70, that is based on the observation that although serial price changes are known to be completely uncorrelated (see FIG. **3**), one can see significant serial correlation if this is examined conditional to volatility. Work in this direction was first presented by Blake LeBaron in his paper LeBaron, B., *Some relations between volatility and serial correlations in stock market returns*, 65 Journal of Business 199 (1992). Model 70 uses triplets, $(vol(M), \delta x_{last}(H), \delta x_{next}(H))$, each triplet being calculated at regular intervals of a basic grid interval t, in physical or σ time. Here δx_{last} and δx_{next} are price changes (with sign) over some horizon H (measured as an integer multiple of t) and vol is the volatility measured as a mean of the absolute log price change; the mean being taken over the last M (for memory) consecutive observations of log price change over the basic grid interval t.

FIG. **4** depicts the USD-DEM serial correlation conditional to volatility. The volatility is measured as the mean daily price change over 15 days and the correlation (solid curve) is measured between the previous 28 day price change against the future 1 day price change. The corresponding unconditional correlation (horizontal solid line) is positive but barely above its 95% significance level (short-dashed line). The conditional correlation for low volatility is clearly well above its significance level (long-dashed curve) as well as the unconditional correlation. Results are based on daily interpolated prices at 18:00 GMT from 1973 to 1993 (excluding weekends and holidays). Each volatility bin in the analysis was chosen to contain 500 pairs. The dotted curve gives the percentile level of the mean volatility of each bin. Typical conditional correlation seen for FX rates is shown. Note how the conditional correlation is clearly positive at low volatility and negative at high volatility and in both cases higher than the unconditional correlation.

FIG. **5** depicts the intra-day USD-DEM serial correlation conditional to volatility. The volatility is measured as the mean over 12 consecutive price changes of 1.5σ hours, and the correlation (solid curve) is measured between the price change over the last 36 σ hours and the price change over the next 1.5σ hours. The corresponding unconditional correlation (horizontal solid line) is barely but significantly positive (compare with the short-dashed line representing the 95% significance level). Results are based on data from 1973 to 1993. Each volatility bin in the analysis was chosen to contain 500 pairs. The dotted curve gives the percentile level of the mean volatility of each bin. Typical conditional intra-day correlation seen for FX rates is shown. Note that the clear signal of a volatility conditional non-linearity as seen in FIG. 4 is absent.

The collection of triplets is then sorted by volatility and the subset of first n price change pairs of the sorted list are then examined for their correlation. This exercise is repeated for every consecutive set of n pairs in the sorted list. As we glide down the list, the mean volatility of the subset of n triplets increases. The relation between the mean volatility of a bin and the corresponding price change correlation is shown graphically in FIG. 4. The main message of FIG. 4 is that there is significant positive correlation in serial price changes in the low volatility regime, and marginal anti-correlation in the extreme high volatility regime.

Note that FIG. 4 corresponds to a series with a basic grid interval of 24 hours in business time (skipping weekends and holidays). Such a series is characterized by a specific hour of the day at which the price changes are reported (the price change being with respect to the same hour the previous working day). FIG. 5 shows the corresponding analysis for a basic grid interval of 1.5 hours in σ time. Such a series is not characterized by any specific hour of the day. One postulate for the poor signal in FIG. 5 is that market microstructure changes with the proportion of traders of different horizons. Every time one of the three international markets (America, Asia and Europe) opens, it brings its own mix of trading profiles (particularly trading horizons), and every time it leaves, it depletes the market correspondingly. This effect can significantly distort the volatility definition used for sorting the triplets. Volatilities which may be regarded as low under certain market mixes may be numerically high when compared to other mixes of traders. Neighboring entries in the sorted data therefore will not uniformly correspond to equivalent market conditions with regard to volatility and this will destroy the signal. On the other hand FIG. 4, based on a series characterized by a specific hour of the day, always refers to a price change corresponding to one complete cycle of microstructure fluctuations.

The results above indicate the existence of a lucrative non-linear signal that may be exploited by trading models. The statistical research could have been continued in its pure form and refined in many ways. In particular, investigations could have been extended to search a function space of volatility definitions that would most accentuate the effect. An even more ambitious front would have been to find volatility definitions corrected for intra-day microstructure fluctuations so that the effect (now missing in FIG. 5) would also begin to show in the intra-day regime.

FIG. 4 implies that there is a trend in the low-volatility regime and a direction change in the extreme high-volatility regime. This motivates the creation of 2 kinds of trading models, one with a trend-following strategy and one with a contrarian strategy. From the statistical investigation we know that, for the best effect, the models must be based on a basic grid size of 24 hours of business time (and therefore characterized by a specific hour of the day, as is FIG. 4). Such models will necessarily suffer from one bad side effect: they would not be truly high-frequency data models and would therefore give trading recommendations at the same hour of each day.

The "trading model incarnation" of the effect captured in FIG. 4 is to have a model with the following attributes:

1. A volatility V(M), measured as the mean daily price change over M (memory) days. (Normalized by the scaling law price change for 1 day).
2. A direction D(H), measured by the price change (with sign) over H (horizon) days. (Normalized by the scaling law price change over H days).
3. A volatility threshold value v, below which a volatility is regarded as low in a trend-following model and above which volatility is regarded as high in the contrarian model.
4. A direction threshold value d, which the absolute value of the direction measure must exceed in order to regard the sign of the direction measure as a solid indicator of direction.

FIG. 6 is a scatter plot of δx_{last} along the x-axis against δx_{next} along the y-axis for the bin corresponding to the maximum positive correlation in FIG. 4. It is clear that the distribution of points is biased in favor of the odd quadrants (as a positively correlated set with null mean should). Note however that this effect can be drastically accentuated if the points within the vertical lines are ignored. Item 4 above is motivated by this observation.

Referring to FIG. 6, as expected for a positively-correlated set of numeric pairs, the scatter is biased toward the odd quadrants. The bias can be accentuated by further eliminating data within the vertical lines. Model 70 exploits this effect through its parameter d, which represents the width between the vertical lines.

The trend-following model must have $V(M)<\upsilon$, and the contrarian model must have $V(M)>\upsilon$ in order to take a position. In addition, both models must satisfy the condition $|D(H)|>d$ (in keeping with item 4 in the list above) to take a position. The position taken is the same as the sign of D(H) for the trend-following model and opposite to the sign of D(H) for the contrarian model (where + is long and − is short).

The simple structure of the model makes it an ideal tool for investigating market components in terms of memory M and more significantly the horizon H. The actual parameter space investigated for these models is as follows:

1. Volatility Memory M ranging from 1 to 30 days (30 values).
2. Direction Horizon H ranging from 1 to 30 days (30 values).
3. Volatility Threshold υ ranging from 0.1 to 3 times the 1 day scaling law in steps of 0.1 (30 values).
4. Direction threshold d ranging from 0 to 1.0 times the H day scaling law in steps of 0.1 (11 values).
5. The total parameter space contains 297000 points.

As already mentioned, the models are investigated with a 24-hour business time series characterized by data at a specific hour of the day. There could, however, be subtle but significant volatility differences in series that measure price differences from open-to-open as compared to price differences from noon-to-noon. By introducing one additional parameter, the time of day corresponding to the series studied, we can use the trading model to study the evolution of the most relevant horizons through the day. Aside from the latter academic aspect, such a study is also motivated by the temporal delocalization issue to be addressed below. The additional parameter, referred to as update-hour, takes values from 0 to 23 in increments of 1 (24 values), and represents the hour of the day characterizing the series studied. A complete database of results for a given currency pair would therefore contain approximately 297,000×24×2=14 million results—approximately 7 million each for the trend-following and contrarian models. With such a large database of results for each currency, there is an extreme risk of over-fitting. The description below presents procedures developed to reduce this risk and also shows evidence that the procedures work as expected.

"Temporal delocalization" refers to the problem that any single model as just described, corresponding to a certain update hour, will only give a recommendation once in 24 hours. The problem is to be able to use the relatively clean information content in the daily business time series, but at the same time to have a final model that is flexible enough to take action at any hour of the day. The solution to this problem is to regard the models at different hours to be simply sub-models that are ingredients of a final model, which we shall refer to hereafter as the grand model. The grand model is thus a portfolio of models that are each updated individually by the daily business time series corresponding to every hour of the day. The grand model then acts at any hour, on the hour, based on some voting scheme applied to the most recently updated sub-models within the past T ($0 \leqq T \leqq 23$) hours. For example, if p_h (+1, 0 or −1) represents the recommended position of the sub-model updated at hour h then the grand model gets into a long or short position at hour h if

$$P_h \equiv \sum_{i=1}^{T} p_{h-i+1}$$

is such that $|P_h| > S_{in}$, with the position being long if the sum is positive and short if negative. The model gets out of this position and becomes neutral if $|P_h| \leqq S_{out}$. Of course we must have the strength parameters S_{in} and S_{out} (thresholds determined during voting rule optimization) such that $S_{in} \geqq S_{out}$, $0 \leqq S_{in} < T$, $0 \leqq S_{out} < T$. (Note that if $S_{in} \geqq T$, the grand model will never take a position). Some other variations are described below.

Note that the input sub-models have to be optimized for update hours at every hour of the day, including hours when the market may actually be closed. During optimization of the sub-models, the environment must be specially modified so that deals can be executed even during closed market hours. In the grand model, deals will never be executed when the market is closed (barring stop losses) but the positions of the models corresponding to the hours when the market is closed will be held in memory since, as seen in the above equations, the grand model uses the sum of sub-model recommendations as an indicator.

In addition to the above strategy, which achieves temporal delocalization up to a one hour grid (besides also increasing robustness as a welcome side effect, by pooling the recommendations of multiple models), there is an additional strategy for further delocalization to the continuous time level. This is achieved by a deal delay mechanism that works in a way similar to the moving stop loss. The deal delay mechanism will prevent the immediate execution of a deal on the recommendation of the grand model. Instead, the grand model will be forced to wait until a better price for the deal becomes available. The wait will continue until a price is obtained that is worse than the best price so far (bid or ask depending on the recommended position) by a configurable ratio referred to as the max-loss (say 0.1%). On some occasions this will be a price worse than the price that would have been obtained if the deal was executed immediately; but the gamble is that on most occasions it will be a better price. This algorithm is configured to work only during open hours when deals are possible and only for a duration of 1 hour between sub-model updates. After a sub-model update (on the hour) the deal-delay mechanism is reset based on the latest grand model recommendation.

The contrarian and the trend-following models are preferably used together. The same algorithm that allows for the pooled opinions of the models at different times of the day can be extended to include the "vote" of multiple models updated at the same hour. Thus a grand model can be used that has not just 24 models but 48 models with 2 synchronous sub-model recommendations (one trend-following and one contrarian) at every hour. It is expected that by and large one of these will be neutral when the other is in a long or short position. Thus, the 2 models complement each other.

As already described, during optimization of sub-models a database is preferably generated (with perhaps 14 million results) for a given exchange rate. To conserve disk space, preferably only those results are entered in the database that correspond to a positive-definite value of X_{eff}. Each result consists of a single line that, besides specifying the parameters, also specifies the X_{eff}, the neutral fraction n (as a percentage of the in-sample during which the model was neutral), and the mean annual return of the model. Due to this large database, it is not a straightforward process to select the best or set of best sub-models that would constitute the grand model. In fact, the selection procedure must itself be an algorithm that produces a final candidate for testing in the out-of-sample. This is described below, after we introduce a new performance measure, the effective-X_{eff}.

The effective-X_{eff}, or fxf[v], is defined by:

$$fxf[v] = \begin{cases} \dfrac{X_{eff}}{1 - \dfrac{n}{100}} & \text{if } n < v \\ 0 & \text{otherwise} \end{cases}$$

The measure fxf[v] has the advantage of evaluating the performance of the model in relation to the amount of time it is exposed to risk. This is a preferred measure in view of the final usage of this (daily) model only as a component in the grand model. The goal is to find a set of sub-models that are individually conservative and act selectively (having large neutral fractions), but when placed together in the grand model complement each other, causing the grand model to have a much lower neutral fraction. The cutoff v is necessary so that models with very large neutral fractions (due to very few trades) would not be rated as very high simply on the basis of a few trades, since this would not be statistically robust.

Effective-X_{eff} is preferable to X_{eff} cases where models are largely neutral but under specific market conditions make money. Because such models are lartely neutral, their absolute return can be lower than that of models that are more often in position. The effective-X_{eff} scales the X_{eff} measure with respect to the amount of time it actually was in position. Through the effective-X_{eff} measure models with drastically different neutral fractions become incomparable.

There are many schemes one could use to transfer the database of results for a given currency (up to 14 million results) into a single grand model. A straightforward scheme is to construct a grand model by choosing the best trend-following and best contrarian model at every hour, based on some performance measure such as X_{eff} or fxf[50]. A more robust selection scheme would be to use as the best model at hour h the model that has the highest mean value of a performance measure (e.g., X_{eff} or fxf[50]) over some of the neighboring hours around h. Such a selection scheme is characterized by a distance d in hours over which the mean should be taken. If the index i represents a specific set of the 4 principle parameters of the model and j represents the update hour of the model with parameters i, then the performance measure of this model may be denoted as χ_{ij}. With the latter definition and parameter d we can determine a parameter set i at hour h for which $\overline{\chi}_{ih}$, defined by

$$\overline{\chi}_{ih} = \begin{cases} \sum_{j=h-d}^{h+d} \chi_{ij} & \text{if } \chi_{ij} > 0 \forall j \\ 0 & \text{otherwise} \end{cases}$$

is maximum. Note that the effect of this scheme as d increases is to prevent rapid variation of sub-models over consecutive hours. The seasonality of the market is in effect at cross purposes to this scheme, so a compromise must be made so that sufficient intra-day market seasonality is captured while robustness of the choice of sub-models is increased.

It is convenient at this stage to formalize the selection schemes by referring to them as $\Theta=\Theta(\chi, d)$, where χ is the choice of performance measure and d is the distance parameter. A preferred embodiment uses 16 selection schemes as possible candidates for choosing the final model. These are: $\Theta(X_{eff}, d)$ and $\Theta(\text{fxf}[v], d)$ for $d \in \{0, 1, 2, 3\}$ and $v \in \{25, 50, 75\}$. These selection schemes are to be separately applied to each database (of 7 million results) for the trend-following and contrarian models for a given currency. The result of each of the (automated) selection schemes is a set of (up to 24) sub-models, denoted by S(rate, strategy, χ, d), when the scheme $\Theta(\chi, d)$ is applied to the currency "rate" and to a model of specific strategy (trend-following or contrarian). The (up to 24) sub-models in a given set each correspond to the recommended components, based on the selection scheme used, for each hour of the day in the grand model. (Note that 24 models may not be obtained in all cases, since the selection algorithm might not find a sub-model that has a positive χ_{ij} in all neighbouring hours over which $\overline{\chi}_{ih}$ is computed).

In general the grand model can be constructed from these sets as the union

S(rate, trend-following, χ_1, d_1) $\cup$ S(rate, contrarian, χ_2, d_2).

Note the subscripts on χ and d. Since we have 16 choices of the pairs $\{\chi, d\}$, there could be 256 choices of the grand model. To limit this number we define:

S(rate, mix, χ, d)=S(rate, trend-following, χ, d) $\cup$ S(rate, contrarian, χ, d)

as the grand model comprising up to 1 trend-following and 1 contrarian model at every hour, obtained by using the same selection scheme $\Theta(\chi, d)$ for choosing the trend-following and contrarian sub-models.

Preferred voting schemes are now described. A scheme based on the sum of sub-model recommendations was mentioned above. The recommendations of all, say N, models within the last T hours are summed. If the current position is neutral, then the grand model recommends a long (+) or short (−) position same as the sign of the sum, provided that the ratio of the sum to N exceeds S_{in}. If the model is already in a position (not neutral), then the model becomes neutral if the above ratio falls to S_{out} or less. The motivation for S_{out} to be distinct from (and less than) S_{in} is to inhibit the grand model from changing position rapidly. The smaller S_{out} is from S_{in} the more difficult is the return to neutral after taking a position. If the successive sub-models (every hour) are not too different in their parameters this would be automatic, because the sub-model at the next hour is then more likely to have the same recommendation as the current one. Thus the choice of

S_{in} is far more crucial than the choice of S_{out}. The logical move is to first study the case $S_{in}=S_{out}$ for various values and then finally set S_{out} marginally lower that S_{in}. We shall refer to this voting scheme as scheme 1.

Another voting scheme is to allow ½ gearing. It is similar to the above scheme except that the ratio of the sum of model recommendations of all N models in the past T hours to the total number of models is taken literally as the gearing. S_{in} and S_{out} are still used as before. If the position is currently neutral and the ratio exceeds S_{in}, the environment will set the gearing to ½, −½, 1 or −1 depending on which of the these values is closest to the ratio. We refer to this as voting scheme 2.

The above two methods are fully democratic among the quorum of sub-models and make no distinction between trend-following and contrarian models. A third, layering, scheme is different in that it introduces a hierarchy among the synchronous models. As mentioned above, the final grand model will preferably have one trend-following and one contrarian model acting synchronously at every hour. These models are largely likely to be uncorrelated by construction since one (the trend-following) takes a position when the volatility is below a certain threshold, and the other (the contrarian) takes a position when it is above a certain threshold. Nevertheless, because of differences in volatility memory between the two models, it is occasionally possible for them to act simultaneously. A layering scheme introduces a hierarchy whereby the non-neutral recommendation of the trend-following sub-model can have a veto-like effect, overruling the recommendation of the synchronous contrarian sub-model, or vice versa. In other words, the subordinate sub-model can only act if the one with precedence is neutral. If, as stated, the trend-following sub-model has precedence, this scheme is referred to as voting scheme 3t, and if the contrarian sub-model has precedence, it is referred to as scheme 3c.

We now describe a preferred optimization strategy for this embodiment (model 70). We choose the in-sample region to be alternating bands of 546 days from 01.01.87 onwards. This gives us 3 bands for optimization. The preferred strategy for optimization of model 70 is to first create a database of solutions for the trend-following and contrarian models (for all hours of the day for a given currency) using only the first two in-sample bands.

The database is then searched by the 16 candidate selection procedures to arrive at the set of sub-models S(rate, trend-following, χ_i, d_j) and S(rate, contrarian, χ_i, d_j), where $i \in [1,4]$ and $j \in [0\text{-}3]$, and where $d_j=j$, χ_1=xeff, χ_2=fxf[25], χ_3=fxf[50], and χ_2=fxf[75]. In addition to the selection of the sub-models, the high-frequency grand model also needs to be specified according to the following 5 high-frequency parameters: (1) the parameter T, which sets the number of hours in the past that play a role in determining the current position; (2) the strength parameter S_{in}; (3) the strength parameter S_{out}; (4) the voting scheme (1,2,3t and 3c); and (5) the max-loss parameter.

Note that here the selection procedure itself becomes a meta-parameter. The selection procedure (determined by χ and d) and the above set of 5 parameters shall be referred to as the high-frequency parameters, since they are the parameters that determine how individual sub-models (acting only once daily) are to be converted into a high-frequency grand model with temporal delocalization.

For the same choice of χ and d, the choice of sub-models can differ between currency rates. For robustness, the strategy is to find χ, d, and the above set of 5 parameters (the set of high-frequency parameters) such that the mean value of the X_{eff} over all currency rates is maximized by their choices in the third in-sample band. Thus, the sub-models are themselves currency-rate-specific, but the high-frequency parameters (including the selection scheme) are common between currency rates. It is reasonable to expect that the degree of overfitting is diminished by (1) a judicious choice of high-frequency parameters; (2) by optimizing them in an in-sample band that has never been exposed to the ingredient sub-models; and (3) choosing these to be globally uniform over all exchange rates. It cannot be over-emphasized that the sub-models are chosen by the selection schemes based only on their individual performance in the first two in-sample bands. There is preferably no optimization to pick the models so that they work well together. (Such a scheme would be drastically prone to overfitting).

Preferably, individual sub-models are optimized on a first in-sample band, and the grand model parameters (including voting scheme) are optimized on a second in-sample band. This allows the grand model parameters, which govern how all of the sub-model recommendations will be used to arrive at a final trading signal, to be optimized on a band that is out-of-sample for the sub-models. This in turn allows the grand model parameters to be optimized in a manner that "buys insurance" against possible over-fitting of the sub-models on their in-sample.

There is one complication that must be discussed. The optimal selection procedure for the trend-following models may not be the same as the one that works best for the contrarian models. Thus the above strategy must be applied to separately construct purely trend-following and purely contrarian high-frequency models. The X_{eff} for these are then separately averaged over currency rates (in the third in-sample band) to determine the best choice of selection procedure for the trend-following sub-models and the possibly distinct best choice for the contrarian models. These two selection procedures are then applied to construct the mixed model, and re-optimized in the third in-sample band to determine the remaining 5 high frequency parameters enumerated above—again by averaging the performance over different exchange rates as usual. We shall refer to models constructed by this procedure as hybrid models.

In addition to the strategy of the last paragraph, we can also test the effect of choosing the set S(rate, mix, χ, d), defined earlier, although it does not allow separate selection procedures for the trend-following and contrarian models. We shall refer to models constructed by this procedure as mixed models.

A database of results for the USD-DEM, USD-JPY and GBP-USD has been completed for the trend-following as well as contrarian models. The results presented below reflect tests in the third in-sample band but by limiting 3 of the 5 high frequency parameters enumerated above in the following way:

1. T is fixed (T=23). This is a common-sense choice since it seems reasonable that the more sub-models used to pool their opinions, the better the model must behave. Some test runs allowing this parameter to vary have confirmed that this is reasonable, although the strictly best models seemed to be clustered around T=20.
2. S_{out} is fixed (S_{out}=S_{in}). Again a common-sense choice. It is expected that performance should marginally improve if S_{out} is marginally less than S_{in}, but the more critical value is definitely S_{in}.
3. The parameter max-loss is fixed (max-loss=0). This essentially disables the deal delay mechanism. The max-loss is the least critical of the parameter set. The value can be set as the last step of the optimization process.

All of the above parameters are preferably allowed to vary at a later stage of the optimization process. The first step is to identify the best choice of selection procedure and the S_{in} parameter. The results presented below, for the 3 databases ready so far (USD-DEM, USD-JPY and GBP-USD), are with the above-enumerated parameter-space limitations.

Table 5 evolves as optimization proceeds over more rates. If the displayed state of Table 1 (having studied only 3 rates) is any indication of its future evolution, it seems reasonable to expect that the best selection procedure for the trend-following sub-models will differ from that for the contrarian sub-models. The mixed model results in Table 5 are for the case where both the trend-following and contrarian models are selected by the same scheme $\Theta(\chi, d)$. After the databases for many more currencies are prepared, one final step is preferably to test a mixed model constructed, not with the same selection scheme for the two strategies, but with the different and optimal selection schemes for the two strategies.

Also note in Table 5 that the strength parameter for the mixed model is lower than that for the single strategy models. This is to be expected since the synchronous trend-following and contrarian sub-models are unlikely to sum to a value greater than 1 (by design, as discussed earlier). Thus the sum of sub-model positions will on an average not change much (compared to the single strategy models), but the total number of models is doubled, thus requiring the strength S_{in} to be reduced. For this reason, the parameter S_{in} for the purely trend-following and purely contrarian models is varied from 0.0 to 0.6, in steps of 0.1, but the same parameter for the hybrid and mixed models is varied from 0.0 to 3.0, in steps of 0.05.

As mentioned above, the selection scheme must strike a balance between allowing intra-day seasonality (by not taking d very high) and robustness (by not taking d very low). Tables 6 and 7 show the evolution of the horizon H (in days), which is the most critical of the 4 sub-model parameters, for each exchange rate, using the best selection scheme for the purely trend-following and purely contrarian grand models as determined by the current status of Table 5.

The rather gradual variation of the horizons from hour to hour and the fact that only a few relevant horizons stand out in Tables 6 and 7 provide testimony to the reasonableness of the selection schemes and their ability to thwart over-fitting.

In the final round, a new database of results must be constructed for each currency rate, wherein the sub-models are exposed to all three in-sample bands. Then the candidate model for the real out-of-sample test, for each currency rate, is the model constructed with these more robust sub-models (by virtue of being exposed to a larger in-sample) by applying the selection procedure and high-frequency parameters gleaned from the tests and criteria in the third in-sample band discussed above.

A fourth preferred embodiment comprises customized trading models. The basic motivation for the trading model of this embodiment, referred to herein as a customized trading model, is that the FX and other markets can change their nature and behavior over the years. The performance of trading models can vary with different market states. Financial markets are known to exhibit distinct and persistent patterns and certain clusters of behavior. In 1995 and 1996, for example, FX markets were characterized by sideways movements with few large trends, while volatility was still high when measured at high resolution. Institutional changes, such as the development of the European Monetary System, can also cause behavior shifts. These changes can affect the performance of a trading model (TM) optimized during periods with particular behavior (e.g., large trends). Thus, a preferred embodiment comprises trading models with a variety of different strategies, according to different market states indicated by volatilities of different time horizons and other nonlinear indicators related to volatility.

This embodiment preferably uses indicators that observe and predict the characteristics of the market. Different strategies for different types of markets are then followed. In other words, specifically designed market state indicators, functions of the present state of the market, are used that help a model to adapt its strategy to the market situation. These adaptive strategies account particularly for the situation of sideways markets, where some other trading models underperform significantly. In sideways markets, short-term noise, rather than large movements, dominates. If a trend-following model is applied, this leads to frequent loss-making transactions. Detecting these states and adjusting the strategy correspondingly can avoid this. Adaptivity of the model refers also to structural breaks of the market, where old market states are disrupted. In this case, the model should react quickly and "forget" about the old market state.

To address these problems, a preferred embodiment implements different sub-models with individual time horizons, thus modeling the views and actions of different trader groups or market components. These different components range from intra-day horizons to time horizons of about one month.

Preferably, from a user's perspective, this trading model (TM) is seen as just another real-time trading model. With respect to the other embodiments described above, there are no additional architectural or functional requirements for the customizations.

Whereas previously-described TMs and the instant customized trading model can be operated in the same technical way, they differ in other respects: the so-called customizations. Some aspects, such as the choice of the active day-times of the model, are discussed only briefly below, since they are common to the general trading model architecture described above. The most important customization is described in detail below.

The trading model architecture is preferably constructed using C++. It has very similar features and properties as the trading model technology described above with respect to the other preferred embodiments. FIG. 1 provides a graphic illustration of this structure.

As in other preferred embodiments, the gearing calculator constitutes the heart of the instant trading model and supplies its characteristic features. The model-specific indicators and a raw gearing of the model are calculated. The result is then processed in the other parts of the architecture. The expression "gearing" refers to the direction and size of the position taken in the market; a negative gearing means a short position.

Financial markets are known to be constituted of different types of traders whose behaviors differ in several respects: the markets are heterogeneous. The most important difference between market components is related to the time horizon: short-term traders observe and cause minute-by-minute price movements, whereas some long-term investors rebalance their positions only in weeks or months.

A trading model algorithm based on a deep understanding of market mechanisms should account for this heterogeneity. An efficient way of dealing with heterogeneity is through the construction of different sub-models, where each of these has its own time horizon. Each sub-model incorporates the view and the actions of one trader group in its analysis and arrives at its own trading recommendations. These recommendations are then weighted together in order to obtain the overall model gearing.

Each sub-model of a preferred embodiment has the same structure, and most parameters are identical among different sub-models (some of them are simple power law functions of the time horizon). This is algorithmically advantageous and makes the optimization process easier and more resistant against overfitting than letting all parameters be optimized independently. The essential parameter of an individual sub-model is, of course, its time horizon.

The algorithm used for the customized trading model has some less specific elements that are described above with respect to other embodiments, such as the price filter, the stop-loss detector, the opportunity catcher (picking a realistic execution price), and others. These elements are adopted by the customized trading model, with an appropriate choice of (optimized) parameters.

Price filtering (checking incoming quotes from the data suppliers for outliers) is necessary to avoid trading signals that result from misquoted prices. A description of a basic approach to filtering is given in Dacorogna, M. M., Müller, U. A., Nagler, R. J., Olsen, R. B., and Pictet, O. V., *A geographical model for the daily and weekly seasonal volatility in the FX market*, 12 Journal of International Money and Finance 413 (1993) (hereinafter Dacorogna 1993). A preferred price filter is based on that approach.

A stop loss detector triggers if the market moves in disfavor of the taken position by passing the stop-loss price, and causes a deal to enter a neutral position. This can happen also overnight, when markets are closed. Overnight stop-loss transactions can be executed by a trader located in another time zone.

A special case is profit taking. A stop price is set as soon as a profit objective is reached. This stop price is moving in parallel with the level of the best price achieved so far during the open periods of the market.

The deal acceptor validates the recommendations of the gearing calculator using the following rules: (1) no deal is allowed to take place within fifteen minutes of another deal having occurred; (2) a deal filter (even more stringent than the normal quote filter) determines if a given price is suitable for trading; and (3) trading can only take place during the open hours of the specific market of the model (except for stop-loss trading, which is supported around the clock).

An opportunity catcher searches for a realistic price to be taken for the internal book-keeping of the model. This happens in a time window of 2-3 minutes (depending on the currency), where the median of the applicable price quotes (bid or ask) is taken. A signal is sent to a human user before this time window (to tell him to execute the deal) and afterwards (to inform him of the price used for internal book-keeping).

The book-keeper evaluates trading model statistics such as current (unrealized) return, total return, cumulated return, maximum drawdown, and profit-over-loss ratio. FIG. 7 depicts a preferred structure of the customized trading model: sub-model and main model.

The final gearing as well as the trading recommendation and some other useful results are displayed on the screen of a user's computer.

We now describe in detail the structure of a single sub-model, analyzing the dynamics of one specific time horizon. Then we describe the integration of all different sub-models in the main gearing calculator.

A visual idea of the overall structure as proposed here can be obtained from FIG. 7. As indicated in FIG. 7 the combination of different sub-models at the end leads to the final gearing recommendation of the gearing calculator (see also eq. 54).

A sub-model comprises different parts that serve different tasks: generating raw gearings, analyzing the market state, generating strategies depending on raw gearings and market state, and finally giving a recommendation for the trading model position. Within this structure two different market state indicators, a "channel indicator" (which is indicated on the upper right in FIG. **7** and a "frequency indicator" $I_{trad.freq.}$ (defined through the main indicator and boxed underneath in the upper left in FIG. **7**) are implemented. The different indicators and strategies for the single sub-models are defined below.

Sub-models can be seen as complete trading models, but the final goal is to combine sub-models of different time horizons into one main model. The different sub-models share the same structure and algorithm, but the time horizons (and some dependent parameters) are different.

The structure of one sub-model can be essentially divided into five parts (see FIG. **8**):

1. The calculation at step **810** of a suitable main indicator for the specific sub-model, which after the application of a simple nonlinear discretization function provides a first raw gearing (either 1 or −1).
2. The calculation at step **820** of a first market indicator, the frequency indicator $I_{trad,freq.}$, which describes the current degree of noisiness in the market (and therefore whether trend-following should be avoided).
3. The calculation at step **830** of a second market indicator, the channel indicator $I_{channel}$ which describes to what degree the current and most recent market situation can be characterized by channel configurations.
4. Computing at step **840** a weight w to be applied to the raw gearing at the end. This weight represents the strategy chosen in view of the market indicators. This is a key feature in making the trading model adaptive. This function w is also modulated by a "spread function" that accounts for the loss due to bid-ask spreads (each "round-trip" transaction leads to a loss of one spread, which may affect the performance in case of frequent transactions of the sub-model).
5. Computing at step **850** the resulting gearing as the product of the raw gearing computed in part 1 and the weight derived from the market indicators. This gearing is later combined with the gearings of other sub-models. The combination is then discretized to provide the final, discrete gearing recommendation of the model.

The trading model works in real-time, with tick-by-tick intra-day data. Therefore, it preferably has a way to deal with the strong intra-daily seasonality of volatility. Price movements around 4:00 GMT, when volatility tends to be low, should be treated differently from price movements of the same size around 14:00 GMT, when volatility is usually high.

The time scale σ has been developed for the deseasonalization of volatility; its computation is explained in detail above and in (Dacorogna 1993). In all the indicator computations presented in the following description, we use σ-time rather than physical time as the basic time scale.

A main indicator z is created with the help of exponential moving averages (EMA). These are computed by an iterative application of the EMA-operator to the (not necessarily regularly-spaced) time series x(t). Depending on the number of iterations, we obtain a moving average based on a specific kernel function. We define:

$$z = z(t) = \frac{\chi(t)}{\sqrt{EMA[\Delta t_{\sigma}, i', j'; \chi^2](t)}}, \tag{25}$$

with a scaling function in the denominator, where

$$\chi(t) = EMA[\Delta t_{short}, i'', j''; x](t) - (1+\epsilon)EMA[\Delta t_{model}, i, j; x](t) + \epsilon EMA[\Delta t_{long}, i, j; x](t) \tag{26}$$

and x denotes the mean logarithmic value of bid and ask prices:

$$x = \frac{\log p_{bid} + \log p_{ask}}{2}. \tag{27}$$

The time t is meant to be the σ-time, as explained above. The parameters i and j determine the form of the kernel of the moving average: $1 \leqq i \leqq j$. Choosing i>1 results in a kernel whose rectangular-like form starts after a lag, rather than immediately. See U.S. Provisional Pat. Application 60/200,743, filed May 1, 2000, to Zumbach & Müller, the contents of which are incorporated herein by reference; see also Zumbach, G. O. & Müller, U. A., *Operators on Inhomogeneous Time Series*, International Journal of Theoretical and Applied Finance (2000.

The EMA-function EMA[Δt_{short}, i, j; x](t) serves the purpose of obtaining steadier data (avoiding "bid-ask bouncing"); the time horizon Δt_{short} is therefore chosen to be in the range of seconds. The specific time horizon of the given sub-model is given by Δt_{model}. The main indicator (with positive small ϵ) leaves space for accounting weakly for possible curvature effects. This is done by including a component corresponding to the (discretized) second derivative which can be done with the help of a moving average EMA[Δt_{long}, i, j; x](t) with longer memory Δt_{long}. Preferably $\Delta t_{long} = 2\Delta t_{model}$ is chosen.

Concerning the scaling function $\sqrt{EMA[\Delta t_{\sigma}, i', j'; \chi^2](t)}$, a specific choice of Δt_{σ} and (i', j') is necessary. We choose Δt_{σ} to be multiple of Δt_{model}, (i.e., χ is scaled with its own long-term moving average). For the iteration numbers i' and j' we choose i'=1 and J'=4, which corresponds to a kernel function that has a plateau for values close to zero and decays slowly for larger values (the center of gravity is given by Δt_{σ}). Although this indicator scaling provides some adaptivity, the overall ability of the entire sub-model to adapt to different market 'states is not yet reached by this scaling function. For this purpose, we need the market state indicators described below.

The indicator z serves as the argument of the following iteratively defined discrete gearing function: Given a gearing g at the previous quote (at $t_{prev.}$), the gearing at the current quote (at t_{now}) is defined to be

$$g(t_{now}) = \begin{Bmatrix} +1 & \text{if } z > z_0 \\ g(t_{prev.}) & \text{if } |z| < z_0 \\ -1 & \text{if } z < -z_0 \end{Bmatrix} \tag{28}$$

where the initialization is given by $g(t_{start})=0$. This defines the first raw gearing of the main indicator. The threshold value z_0 is a fixed parameter of the model (to be optimized), constant over time.

In order to account for different conditions of the markets (where a certain gearing g must be interpreted differently) the gearing g is weighted with a market state indicator function w=w(market, z) such that a modulated gearing g_m is finally obtained:

$$g_m = w(\text{market}, z) \cdot g \tag{29}$$

The function w, which will be defined in detail below, is a function of several different market state indicators. The gearing values g_m of the different sub-models then serve as the basis for a last discretization step that provides the final gearing of the overall model (see below, eq. 54).

The market indicator function w(market, z) accounts for a wide range of market states. Here we choose w to be non-negative, w ∈ [0,1]; the model avoids contrarian strategies (positions opposite to the gearing recommendation g(z)). The weight function w(market, z) and the strategy of the model depend on two different market state indicators, a channel indicator and a noisy market indicator.

One market state indicator is designed to detect sideways markets through the identification and characterization of what we call channel configurations (a term that originates from technical chart analysis). Channel configurations are the result of a market consensus among traders. Here, we use a specially-designed channel indicator describing the degree to which a recent price development can be described as a channel by counting the number (and amplitude) of oscillations of prices within a given time interval and a range of prices. This channel indicator recognizes not only static price ranges; it adjusts to an overall linear trend of the channel as well as a possible narrowing ("triangle formation").

For the definition of the channel indicator $I_{channel}$ we proceed as follows: We choose a time horizon $\Delta T_{channel}$ to specify the length of possible channels (other sub-models have channels of different lengths). This length $\Delta T_{channel}$ is a high multiple of the time horizon Δt_{model} of the main model indicator (a parameter that needs to be optimized). The channel extends over the time interval C of size $\Delta T_{channel}$. At a given time t we define the interval

$$C = [t - \Delta T_{channel}, t] \tag{30}$$

which covers the immediate past until time t.

In the following the variable x refers to the logarithmic price in a smoothed form: a moving average of the original x from eq. 27, with a memory (center of gravity of the kernel) of $\Delta t = 1/120 \cdot \Delta T_{channel}$. This smoothing leads to neglecting extremely fine details of the price history details in the channel determination—a desired effect that reflects also the view of market practitioners. The usage of MAs instead of original prices also leads to a higher degree of statistical stability for the definition of the channel. Within the interval C we store 120 different past values of the moving average (with a certain overlap of the kernel functions) that we name $x_k = x(t_k)$, $t_k \in C$, indexed by k={1, . . . , N} (where N=120).

We make different particular choices for the time horizon $\Delta T_{channel}$ in eq. 30. In order to adapt the channel indicator to particular situations where a market just enters into a channel, we choose different time horizons $j\Delta T_{channel}$ that are certain fractions of $\Delta T_{channel}$ (e.g., j={0.5, 0.6, . . . , 0.9, 1}), and choose j such that the corresponding time interval C shows the the most convincing channel features (measured by a variable introduced below; see eq. 38).

The following steps lead to the definition of the market state indicator:

1. De-trending: We evaluate a possible trend of the price movements in the interval C regression and obtain the slope m for the trend of price movements in the interval C as the offset constant a. This trend is then subtracted such that we obtain the residuals

$$r(t_k) = x(t_k) - (a + m \cdot t_k). \tag{31}$$

2. Narrowing of the channel interval: As a subsequent step we perform a second linear regression, this time on the absolute values $|r(t_k)|$ of the residuals of eq. 31, in order to account for possible channel narrowings ("triangle formations"). This leads to the slope s and the offset b. In case of a negative slope s the residuals $r(t_k)$ are resealed with the inverse of the resulting (decreasing) linear function to give $\tilde{r}(t_k)$. We require s=0 if the linear regression results in a positive slope, as we are not interested in the unusual configuration of a widening channel:

$$\tilde{r}(t_k) = \frac{r(t_k) \cdot b}{b + s \cdot t_k}, \; s \le 0. \tag{32}$$

A further constraint that we impose is a lower limit for s such that the channel width at the end of the interval C be smaller than 30% of the channel size at the beginning of C.

3. Defining the channel frequency: We next define the interval $[\min_{ch}, \max_{ch}]$ by choosing $\min_{ch}$ ($\max_{ch}$) as the minimum (maximum) of the price residuals $\tilde{r}(t_k)$ for $t_k \in C$:

$$\min_{ch} = \min_{t_k \in C} [\tilde{r}(t_k)] \quad \max_{ch} = \max_{t_k \in C} [\tilde{r}(t_k)] \tag{33}$$

The value of $(\max_{ch} - \min_{ch})$ defines a sort of width of the residual channel. Instead of using this value for further calculation, we define another width ΔP of the channel that has a certain narrowing term in the case of triangle formations (s<0):

$$\Delta P = (\max_{ch} - \min_{ch}) + s \cdot \Delta T_{channel} \tag{34}$$

We now track the number of times the residual function $\tilde{r}(t_k)$ approaches $\min_{ch}$ and $\max_{ch}$ (alternating between the two borders). This number is given by the following expression

$$Q = Q(\Delta T_{channel}, r_c) = N(\{k \mid m_k \ne m_{k-1}, 2 < k \le N\}) \tag{35}$$

where

$$m_k = \begin{cases} -1 & \text{if } (\tilde{r}(t_k) - \min_{ch}) < r_c \\ +1 & \text{if} (\max_{ch} - \tilde{r}(t_k)) < r_c \\ m_{k-1} & \text{otherwise (0 in the case } k = 1) \end{cases} \tag{36}$$

with a tolerance level r_c. Together with this number Q we obtain a sequence of time intervals separated by the time points t_q of the crossings over r_c that lead to a count in eq. 35. Within all these Q intervals, we find local minima and maxima of $\tilde{r}(t_k)$, which are used to define our modified oscillation frequency. This is defined as the sum of absolute differences between subsequent local extrema (scaled by the range of global extrema):

$$F = F(\Delta T_{channel}, r_c) = \frac{\sum_{q=1}^{Q} [\max\{|\tilde{r}(t_k)| | t_k \in [t_{q-1}, t_q]\} + \max\{|\tilde{r}(t_k)| | t_k \in [t_q, t_{q+1}]\}\}}{\max_{ch} - \min_{ch}} - 1 \quad (37)$$

Instead of choosing just one tolerance level r_c of an approach to the top or bottom of the channel, we take several r_c values in the range of 20%-40% of the channel width. The average value of F over this range of tolerance levels r_c is calculated. This average is more stable and less sensitive to random effects than an F based on only one r_c level. An offset of −1 is introduced to account for the minimum number of counts (there must be one count at least, even if there are no oscillations).

4. Definition of the channel indicator: In order to obtain a final indicator for the strength of a channel configuration in the market, the function F is scaled with a factor containing the width ΔP, the slope of the trend m, and the time interval $j\Delta T_{channel}$:

$$I'(t, j\Delta T_{channel}, r_c, \mu, \nu) = \frac{\left(\frac{\Delta P}{v(\Delta T_L)}\right)^{\mu j^{\nu}}}{\sqrt{1 + \left(\frac{mj\Delta T_{channel}}{\Delta P}\right)^2}} F(t, j\Delta T_{channel}, r_c) \quad (38)$$

where the second factor F is more important than the only moderately-varying first factor. The variable $v(\Delta T_L)$ is a long-term moving average of the volatility measured with a memory of a multiple of the channel length. This definition is aimed at making the scaling factor adaptive to structural changes of the FX rate (e.g., a substantial volatility increase). The indicator I' receives a high value when the price alternates often between the borders of the considered interval, which corresponds exactly to what we want to describe as a channel. The wider this channel (factor

$$\left(\frac{\Delta P}{v(\Delta T_L)}\right)^{\mu}$$

with an exponent μ=0.3), the longer the channel (factor $(j\Delta T_{channel})^{\nu}$ with ν=0.5) and the smaller the slope as compared to the width

$$\left(factor\ 1 \Big/ \sqrt{1 + \left(\frac{m\Delta T_{channel}}{\Delta P}\right)^2}\right),$$

the higher the indicator value.

We emphasize here again that we choose channels of different lengths in order to make the channel indicator more flexible, especially in situations where the market just entered a new channel (which is short at the beginning). The final indicator $I_{channel}$ is then given as the maximum among these channels:

$$I_{channel}(\Delta T_{channel}, r_c, \mu) = \max_{j=\{1, \ldots 5\}} (I'(j\Delta T_{channel}, r_c, \mu)) \quad (39)$$

The channel indicator $I_{channel}$ is found to be correlated to volatility (and the ratio or difference of coarse and fine volatility). All these quantities measure, in one way or another, the intensity of price changes. An advantage of the channel indicator lies in its ability to quickly catch the disruption of an old market state (channel break). These breaks may occur very suddenly; purely volatility-based indicators are unable to catch them with the same speed.

Another measure of the market state is introduced by a second market state indicator, one that measures directly the "noisiness" of the market. In sideways markets (which are not always described by a clear channel configuration) the short-term noise dominates the large movements and long-term trends. This can lead to frequent, loss-making transactions of a trend-following model. A noise indicator aims at detecting these situations, and enables the incorporation of corresponding features into the definition of the market state function w. The indicator is similar to a direction change frequency as defined in Guillaume, D. M, Dacorogna, M. M., Davé, R. D., Müller, U. A., Olsen, R. B., and Pictet, O. V., *From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets*, 1 Finance and Stochastics 95 (1997), but is directly associated with the main indicator z itself. The definition of the noise indicator is as follows:

$$I_{trad.freq.} = \mathrm{EMA}[\Delta t_{trad.freq.}, i, j; \delta](t) \quad (40)$$

with a moving average with a specific kernel and a long memory $\Delta t_{trad.freq.}$ that is another parameter in the optimization of the model. The function δ is defined as follows:

$$\delta = \begin{cases} \frac{1}{t_{now} - t_{prev.}} & \text{if } g(t_{now}) \neq g(t_{prev.}) \\ 0 & \text{otherwise} \end{cases}, \quad (41)$$

where t_{now} and $t_{prev.}$ denote the times of the current and previous quote, respectively, and g is defined by eq. 28. This definition of $I_{trad.freq.}$ essentially corresponds to the trading frequency of an unconditional trend-following model, that is, a model that strictly follows the trading rule specified in eq. 28.

This indicator essentially tells us when trend-following is dangerous: a high value of $I_{trad.freq.}$ occurs in markets with a high degree of noise. An unconditional trend-following model makes many transactions that lead to subsequent losses. Accordingly, a trend-following strategy should only be followed in market situations with a low value of the indicator $I_{trad.freq.}$. The indicator defined in eq. 40 has a very important feature: a high autopredictive power (high autocorrelation). Indeed, our studies have shown that $I_{trad.freq.}$ possesses significant autocorrelation extending to lags of one order of magnitude longer than the memory $\Delta t_{trad.freq.}$ of the moving average. By being based on the main indicator z, $I_{trad.freq.}$ also has the advantage of indicating exactly the type of noise that is most damaging for trading.

The two market state indicators—the channel indicator and the noise indicator—are now used to define the market state function.

When defining an adaptive weight function w for the main indicator we first want to differentiate three market states.

1. The price is located within a clear channel (high value of $I_{channel}$).
2. A break out of a clear channel occurs (high value of $I_{channel}$, and the current price clearly moves out of the boundaries of the recent channel).
3. There is no channel present (low value of $I_{channel}$). In this case we differentiate between market states with a high value of the noise indicator $I_{trad.freq.}$ and those with a low trading frequency.

These different market situations must be treated differently for the definition of a final gearing—each one of them leads to a different form of the weight function w.

Before defining the specific strategy, we must set up rules to differentiate the different configurations of the market with respect to the channel indicator. In order to define a price to be "inside a channel" or in a "channel break," we must refer to a specific channel of the recent past. One way to do this is to choose the last computed channel without accounting for any other recent channel computed at a previous time. If we want to account for possible different relevant channels in the recent past, we must construct a "channel memory" in order to find the most significant channel of the recent channels. The "quality of a channel" at time t is measured by the indicator $I_{channel}(t)$. For our model the rule for selection of the most relevant channel at a time t is as follows: Given the current indicator value $I_{channel}(t)$, the most significant channel $I_{channel}^{best}(t)$ at the current time t is chosen as

$$I^{best}_{channel}(t) = \max\left(I^{best}_{channel}(t_{before})e^{-\frac{t-t_{before}}{t_{decay}}},\ I_{channel}(t) \right) \tag{42}$$

where t_{decay} is a suitably-chosen decay constant. The channel boundaries of recent channels are extrapolated with the linear trend only, and not with the narrowing trend.

The "inside a channel" as well as the "channel break" situation at time t is conditional to a considerable channel formation, characterized by a high value of $I_{channel}^{best}(t)$. We choose a threshold tr_c (which becomes an important parameter of the model), below which we do not consider the price to be in a channel. This parameter is also exposed to a certain "hysteresis": it acquires a higher value if the recent market configuration has not been detected to be a channel, whereas it is lowered if the market has already been in a channel before.

In order to detect a current price to be inside a channel state at time t, the following formal conditions have to be satisfied:

$$(I_{channel}^{best})(t)>tr_c)\,\vartheta\,(\min_{ch}-tol_{channel}\Delta P<x(t)<\max_{ch}+tol_{channel}\Delta P) \tag{43}$$

where the parameter "$tol_{channel}$" defines a tolerance level that has to be exceeded before we detect an outbreak.

The situation of a channel break differs from the "inside the channel" situation only insofar as the price moves outside the channel borders. In order to detect a channel outbreak at point t, the following formal conditions have to be satisfied:

$$(I_{channel}^{best}(t)>tr_c)\,\vartheta$$

$$[(x(t)-\max_{ch})>tol_{channel}\Delta P \vee (-x)(t)+\min_{ch})>tol_{channel}\Delta P] \tag{44}$$

Once the market is in a channel-break situation, it remains there as long as the main indicator z keeps its sign, irrespective of eq. 44.

The situation of no channel presence in the market can be simply characterized by

$$I_{channel}^{best}(t)<tr_c \tag{45}$$

The noise indicator $I_{trad.freq.}$ is only considered in the no-channel situation: $I_{channel}^{best}(t)<tr_c$, eq. 45. Here, we differentiate between two cases for the definition of the market state function w: $I_{trad.freq.}$ exceeds a threshold parameter tr_{noise} or is lower than this threshold:

$$1.\ I_{trad.freq.}<tr_{noise} \tag{46}$$

$$2.\ I_{trad.freq.}>tr_{noise} \tag{47}$$

Being inside a channel can be a dangerous condition for a trend-following model (as well as a high value of $I_{trad.freq.}$). The model is likely to recommend frequent transactions mostly at the wrong time—just before the trend to be followed is turning. A price development inside a detected channel should not be followed. As a consequence, a preferred (sub)model does not give any trading recommendation in this case, in other words:

$$w=0 \tag{48}$$

There will be neither trend-following (w>0) nor contrarian strategies (w<0) in this case. The expression "contrarian" refers to the given time horizon. Evidently, the global model can take a position "contrarian" to the raw gearing g_c of a single time horizon, in the case that other horizons interpret the market differently. There is the possibility of adapting a very restricted trend-following strategy for the case that the price moves from a channel border with a main indicator indicating a trend towards the other side of the channel. This strategy component can be systematically tested independently. First tests did not yield a convincing improvement of the results, though, when this additional strategy is applied.

A break of a channel configuration constitutes a substantial or sometimes even structural change in the market and should be specifically accounted for in the trading model. The price development occurs outside a former "market consensus" as formulated by eq. 44. The model should react quickly—old assumptions on certain turning point behavior inside the channel lose their validity. We distinguish between whether a break occurs in the same direction as the trend of the channel or against that trend. The latter case should be considered with more caution. We define the strategy function as follows:

$$w' = \left\{ \begin{matrix} 1 & \text{if } mg > 0 \\ (1-K_c) & \text{otherwise} \end{matrix} \right\},\ K_c > 0 \tag{49}$$

The final w is obtained by a modulation with a function F_1(distance, $I_{channel}^{best}$) depending on the distance of the price from the channel border and on the channel indicator value:

$$w=w'F_1(\text{distance}, I_{channel}^{best}),\ 0\leqq F_1\leqq 1 \tag{50}$$

When the distance from the channel is small, right after the break, F_1 is also small, but it rapidly grows and saturates at a higher level with increasing distance from the channel. This saturation level is 1 for clear channels with high $I_{channel}^{best}$ and lower in case of lower $I_{channel}^{best}$.

In case there is no channel present in the market ($I_{channel}^{best}(t)<tr_c$), the noise indicator $I_{trad.\ Freq.}$ comes into play. The market state function is here defined as follows:

$$w = \left\{ \begin{array}{ll} w_{trend} & \text{if } I_{trad.\ freq.} < tr_{noise} \\ 0 & \text{otherwise} \end{array} \right\} \tag{51}$$

where

$$w_{trend}=c_2F_2(I_{channel}),\ 0\leqq F_2\leqq 1,\ 0<c_2<1 \tag{52}$$

The parameters tr_{noise} and c_2 enter into the model. As before with the case of a channel-break, the final w is scaled with a function $0\leqq F_2\leqq 1$, depending on the channel indicator, in order to give a smooth transition between channel and out-of-channel situations.

It should be mentioned that the presented threshold values are all subject to a "hysteresis" modulation In other words, their value depends on the previous state of the market. This avoids random oscillation between different market states and makes them more inert.

Each sub-model has a gearing recommendation g_m resulting from eq. 29. A preferred trading model algorithm integrates these gearing recommendations to produce one final gearing recommendation G.

The set of sub-models and their horizons has to be chosen. A factor of about 2 between neighboring sub-model horizons is preferably selected. The longest main indicator has a horizon of 32 days, its channel a length of a few months. We choose the shortest main indicator to have a horizon of 3 hours in σ-time (corresponding to an hour or less in physical time during the most active trading periods). However, a sub-model with a main indicator horizon of about 12 hours makes little sense as there is an empirically observable gap between genuine intra-day traders and traders with overnight positions. Altogether 8 sub-models are preferably employed.

The goal is to optimize the risk-corrected return ensuing from the gearing recommendations G. G is changing over time and can be seen as a time series. Noisy oscillations of G should be avoided as every move of G implies a transaction and thus transaction costs. Therefore, the formula for G should dampen the noise originating from the varying g_m values of the many sub-models.

The following sub-model integration formula is chosen to produce gearing recommendations G:

$$G_m = c\frac{1}{1+\sum_{i=1}^{n} w_i}\sum_{i=1}^{n} g_{m,i} \tag{53}$$

where n is the total number of sub-models (n=8 in this embodiment). G_m mainly reflects those sub-models with a clear view. Sub-models in a channel or sideways (noisy) market (where the value $I_{trad\ freq.}$ is large) have w and thus g_m values close to zero; they hardly contribute to G_m—they are "switched off". The constant c is a scaling constant that is chosen such that the following discretization can be performed with threshold values between −1 and 1.

G_m is a volatile quantity changing its value at every tick. It has to be converted to a discrete gearing value G with as few noisy oscillations as possible. The discrete gearing values are preferably either −1, −½, 0, ½, 1 (short, half-short, neutral, half-long, long). A finer gearing scheme can be implemented in order to lower the risk and, in thin markets, to keep trade sizes small and avoid slippage. The following formula for G is related to the above 5-position scheme and can be easily adapted to other schemes:

$$G = \begin{cases} 1 & \text{if } G_m \geq 1 \\ \max\left(\frac{1}{2}, G_{old}\right) & \text{if } 1 > G_m \geq \gamma_1 \\ \max(0, G_{old}) & \text{if } \gamma_1 > G_m \geq \gamma_2 \\ G_{old} & \text{if } \gamma_2 > G_m < -\gamma_2 \\ \min(0, G_{old}) & \text{if } -\gamma_2 \geq G_m < -\gamma_1 \\ \min\left(-\frac{1}{2}, G_{old}\right) & \text{if } -\gamma_1 \geq G_m < -\gamma_0 \\ -1 & \text{if } -\gamma_0 \geq G_m \end{cases} \tag{54}$$

where G_{old} is the G value before computing the new G_m. This formula acts like a hysteresis with a dampening effect (or a dragging anchor): G moves only if G_m substantially deviates from the old value G_r. The threshold values have the following relation:

$$1>\gamma_1>\gamma_2>0 \tag{55}$$

Many parameters of the algorithm are "weak" parameters and are not subject to systematic parameter optimizations. They are chosen according to the logics of the algorithm and from separate studies, and then they are frozen.

Only a few essential parameters such as the main threshold values are systematically optimized. Limiting the overall number of parameters to be optimized decreases the danger of overfitting—fitting random features of the specific in-sample period rather than real market structures. The parameter values are as much as possible optimized for all time horizons and FX rates together; different treatments of these should follow from the adaptivity of the algorithm rather than from separately optimizing parameters. The most relevant parameters for optimization are:

the channel length $\Delta T_{channel}$ (see eq. 39)
thresholds for $I_{trad.freq.}$ (see eq. 46), $I_{channel}$ (see eq. 45), and the main indicator z (z_0, see eq. 28)
the memory $\Delta t_{trad.freq.}$ (see eq. 40)
the channel tolerance $tol_{channel}$ (see eq. 43)
the channel memory t_{decay} (see eq. 42).

An important additional parameter is the scaling constant c in eq. 56.

The full sample (the full time period with available raw data) is preferably divided into alternative pieces of a length of about 15 months: in-sample, out-of-sample, in-sample, out-of-sample, ... Only the in-sample periods are used for performance measurement during the whole optimization phase.

After the in-sample optimization, the algorithm and its parameters are frozen. The out-of-sample test then measures the performance of the model in the out-of-sample periods—those that have not been tested yet.

If some FX rates (or other financial instruments) have been left out in the in-sample optimization, their test in the full sample can also be considered as a valid out-of-sample test.

It is to be expected that the model performance is not as good out-of-sample as in-sample. The out-of-sample performance is a much better (though not perfect) measure of the true performance in the future than the in-sample performance.

While the subject invention has been particularly shown and described with reference to preferred embodiments of the system and methods thereof, it will also be understood by those of ordinary skill in the art that various changes, variations, and modifications in form, details, and implementation may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. For example, although the description of preferred embodiments has focused on a currency trading system, it will be apparent to those skilled in the art that the subject invention is also applicable to securities markets, options markets, and other markets. Also, because U.S. Provisional Application No. 60/274,174, filed Mar. 8, 2001, has been incorporated herein in its entirety by reference, the trading models disclosed therein, in particular in Chapter 11 (pages 286-338), are part of this disclosure, and are intended to be encompassed by the appended claims.

TABLE 2

FX rate	Sample	$\bar{X}$	$X_\Delta$eff	D	P/L
USD/DEM	in sample	19.1%	13.5%	8.4%	0.48
	out of sample	26.9%	19.0%	7.6%	0.41
	ex ante	28.8%	22.2%	3.9%	0.69
USD/JPY	in sample	15.1%	9.7%	7.5%	0.61
	out of sample	13.4%	8.7%	8.2%	0.70
	ex ante	−0.2%	−1.1%	4.2%	0.57
GBP/USD	in sample	14.6%	8.9%	11.1%	0.43
	out of sample	20.9%	14.9%	7.5%	0.46
	ex ante	13.7%	11.8%	7.1%	0.38
USD/CHF	in sample	18.4%	14.4%	6.9%	0.69
	out of sample	12.5%	6.2%	12.0%	0.51
	ex ante	22.7%	15.7%	3.2%	0.68
USD/FRF	in sample	19.3%	13.0%	10.3%	0.48
	out of sample	17.4%	12.4%	6.3%	0.48
	ex ante	9.1%	6.5%	5.7%	0.49
USD/NLG	in sample	20.3%	16.6%	7.6%	0.63
	out of sample	19.6%	11.1%	8.4%	0.53
	ex ante	28.9%	20.4%	4.2%	0.74

TABLE 2-continued

FX rate	Sample	$\bar{X}$	$X_\Delta$eff	D	P/L
USD/ITL	in sample	11.2%	5.8%	13.3%	0.46
	out of sample	20.2%	14.9%	6.4%	0.47
	ex ante	18.8%	13.5%	2.9%	0.56
Average	in sample	16.9%	11.7%	9.3%	0.54
	out of sample	18.7%	12.5%	8.1%	0.51
	ex ante	17.4%	12.7%	4.5%	0.59

In-sample period (1.3.86 to 1.3.89 for USD/DEM, USD/JPY, GBP/USD and USD/CHF, and 1.12.86 to 12.3.89 for USD/NLG, USD/FRF and USD/ITL), the out-of-sample period (1.3.89 to 1.10.91 and 1.12.89 to 1.10.91, respectively) and the ex ante period (1.10.91 to 4.9.92).

TABLE 3

FX rate	Model	$\bar{X}$	$X_\Delta$eff	D	P/L
USD/DEM	MA(20)	8.8%	3.3%	14.3%	0.37
	Preferred model	21.9%	14.1%	7.7%	0.41
USD/JPY	MA(20)	−0.7%	−7.1%	39.9%	0.16
	Preferred model	16.2%	12.1%	8.2%	0.66
GBP/USD	MA(20)	8.7%	4.3%	9.3%	0.45
	Preferred model	17.7%	11.4%	10.9%	0.43
USD/CHF	MA(20)	7.9%	1.1%	18.9%	0.35
	Preferred model	17.7%	10.7%	13.0%	0.62
USD/NLG	MA(20)	10.5%	6.3%	12.8%	0.41
	Preferred model	20.1%	14.2%	8.9%	0.57
USD/FRF	MA(20)	8.8%	4.4%	12.7%	0.30
	Preferred model	22.1%	15.9%	8.6%	0.49
USD/ITL	MA(20)	6.6%	2.3%	14.4%	0.34
	Preferred model	15.3%	10.3%	9.2%	0.46

Performance Comparisons with a 20 Day MA Model (between the preferred trading models and a more conventional 20-day moving-average model). The test was conducted from 3.3.86 to 1.10.91 (major rates) and from 3.12.86 to 1.10.91 (minor rates). The same performance indicators as in Table 2 are displayed.

TABLE 4

	Preferred Trading Model			Preferred Trading Model Portfolios		
Exchange rates	Annualized return	Maximum drawdown	Drawdown period	Annualized return	Maximum drawdown	Drawdown period
USD/GBP	17.61%	11.07%	95 days			
USD/DEM	23.06%	8.36%	133 days	18.35%	5.24%	22 days
USD/JPY	14.19%	8.24%	21 days			
USD/DEM	23.06%	8.36%	133 days			
USD/CHF	15.77%	12.00%	142 days	17.73%	5.41%	22 days
USD/JPY	14.19%	8.24%	21 days			
USD/GBP	17.61%	11.07%	95 days			
USD/CHF	15.77%	12.00%	142 days	15.95%	6.26%	21 days
USD/JPY	14.19%	8.24%	21 days			
USD/DEM	23.06%	8.36%	133 days			
USD/CHF	15.77%	12.00%	142 days	18.95%	6.62%	149 days
USD/GBP	17.61%	11.07%	95 days			
USD/FRF	18.51%	10.28%	129 days			
USD/DEM	23.06%	8.36%	133 days	18.28%	6.54%	129 days
USD/JPY	14.19%	8.24%	21 days			

Performance Results of a Portfolio of Trading Models (portfolios of three preferred trading models with a stable, equal distribution of capital among the three models over a test period from Mar. 3, 1986 (Dec. 1, 1986) to Mar. 2, 1992).

TABLE 5

Strategy	Selection Scheme	Voting Scheme	S_{in} S_{out}	X_{eff} USD-DEM	X_{eff} USD-JPY	X_{eff} GBP-USD	X_{eff} Mean
Trend Following	Θ(fxf[50], 2)	2	0.4	1.46	1.61	0.22	1.10
Contrarian	Θ(fxf[75], 1)	1	0.3	2.66	6.93	5.16	4.91

TABLE 5-continued

Strategy	Selection Scheme	Voting Scheme	S_{in} S_{out}	X_{eff} USD-DEM	X_{eff} USD-JPY	X_{eff} GBP-USD	X_{eff} Mean
Hybrid	Θ(fxf[50], 2) Θ(fxf[75], 1)	1	0.15	3.00	9.20	1.69	4.63
Mixed	Θ(fxf[75], 3)	1	0.2	1.28	11.81	0.20	4.43
Model 40				1.80	−3.61	−18.35	−6.72
Model 50				−1.21	−0.34	−10.33	−3.96

Results for the purely trend-following and purely contrarian models as well as for the mixed and hybrid models (constructed with one trend-following and one contrarian sub-model acting synchronously at every hour) in the third in-sample band (to which the sub-models have not been exposed). Indicated in the table are the set high frequency parameters (including the selection scheme) in order to get the highest mean X_{eff} over the 3 rates studied so far. The individual performances for each rate that contribute to the mean are also included. (It should be noted that the best results for the individual rates are of course different). For comparison, the corresponding performance for other preferred models in the third in-sample band (24.12.92 to 22.06.94) are also presented.

TABLE 6

	Trend-Following Horizons using θ(fxf[50], 2)																							
Rate/Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
USD-DEM	28	28	18	28	28	18	28	28	28	28	28	28	28	28	15	15	15	18	28	28	28	28	28	28
USD-JPY	17	17	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	16
GBP-USD	14	14	14	14	16	16	16	16	16	15	15	15	17	14	14	14	14	14	14	14	16	16	16	16

Variation of trend-following sub-model horizons (in days) over the hours of the day using the selection scheme θ(fxf[50],2) for the exchange rates studied.

TABLE 7

	Contrarian Horizons using θ(fxf[75], 1)																							
Rate/Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
USD-DEM	4	5	4	4	4	3	7	3	3	3	3	3	7	6	7	7	7	7	7	3	3	3	3	3
USD-JPY	4	3	3	6	6	4	4	4	4	4	4	4	4	1	5	6	6	6	6	4	6	3	3	3
GBP-USD	3	3	7	7	7	7	3	3	3	3	3	3	3	7	14	13	4	4	3	4	4	4	6	6

Variation of contrarian sub-model horizons (in days) over the hours of the day using the selection scheme 0(fxf[75],1) for the exchange rates studied.

What is claimed is:

1. A method of trading assets on a market, comprising the steps of:

(a) receiving price data for an asset over one or more computer networks;

(b) receiving current system position information;

(c) storing said received asset price data and said current system position information in a computer-readable medium;

(d) calculating trade recommendation information from each of a plurality of trading sub-models; wherein each sub-model is based on a different time of day, said calculation based on said received asset price data;

(e) calculating a trade recommendation regarding said asset based on said trade recommendation information from each of said trading sub-models.

2. A method as in claim 1, wherein 24 sub-models are used.

3. A method as in claim 1, further comprising the step of evaluating the performance of said sub-models using a risk-sensitive performance measure.

4. A method as in claim 3, wherein said risk-sensitive performance measure is used to optimize the performance of the trading models.

5. A method as in claim 3, wherein said risk-sensitive performance measure is an effective return.

6. A method as in claim 1, wherein each sub-model comprises:

(a) a price collector component;

(b) a price filter component;

(c) a price database component;

(d) a gearing calculator component;

(e) a deal acceptor component; and

(f) a book-keeper component.

7. A method as in claim 1, wherein the step of calculating a trade recommendation regarding said asset based on said trade recommendation information from each of said trading sub-models is performed by summing weighted trade recommendations of the sub-models.

8. A method as in claim 1, wherein the step of calculating a trade recommendation regarding said asset based on said trade recommendation information from each of said trading sub-models is performed by summing weighted trade recommendations of N sub-models within the last T hours, where N and T are positive integers.

9. A method as in claim 1, wherein the step of calculating a trade recommendation regarding said asset based on said trade recommendation information from each of said trading sub-models is based on a ratio obtained by summing weighted trade recommendations of N sub-models within the

last T hours, where N and T are positive integers and dividing that sum by the total number of sub-models.

10. A method as in claim **1**, wherein each sub-model is based on a triplet comprising price change and volatility data calculated at regular intervals of a basic grid interval.

11. A method as in claim **10**, wherein the volatility is measured as a mean of absolute log price change.

12. A method as in claim **11**, wherein the mean is taken over the last M consecutive observations of log price change over the basic grid interval, where M is a positive integer.

* * * * *

US008392311B2

(12) United States Patent
Olsen et al.

(10) **Patent No.: US 8,392,311 B2**
(45) **Date of Patent: *Mar. 5, 2013**

(54) **CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE**

(75) Inventors: **Richard B. Olsen**, Zurich (CH); **Michael Stumm**, Toronto (CA)

(73) Assignee: **Oanda Corporation** (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1305 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/634,014**

(22) Filed: **Dec. 4, 2006**

(65) **Prior Publication Data**

US 2007/0078755 A1 Apr. 5, 2007

Related U.S. Application Data

(63) Continuation of application No. 09/858,610, filed on May 16, 2001, now Pat. No. 7,146,336.

(60) Provisional application No. 60/274,174, filed on Mar. 8, 2001.

(51) **Int. Cl.**
G06Q 40/00 (2012.01)

(52) **U.S. Cl.** **705/37**

(58) **Field of Classification Search** 705/37
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,787,402	A *	7/1998	Potter et al.	705/37
H2064	H *	5/2003	Buchalter	705/37
H0002064	H *	5/2003	Buchalter	705/37
2001/0049651	A1 *	12/2001	Selleck	705/37
2002/0023053	A1 *	2/2002	Szoc et al.	705/39

OTHER PUBLICATIONS

Cagan, Penny. Financial risk management sources: A walk on the wild side; EContent 22.6 (Dec. 1999): 16-25.*

By Darren McDermott; Corporate Performance 1999 Review: 1st quarter—Latest Pro; Wall Street Journal [New York, N.Y] May 3, 1999: C1.*

Turton, Jonathon. It is time rethink your netting; Corporate Finance 175 (Jun. 1999): 32-34.*

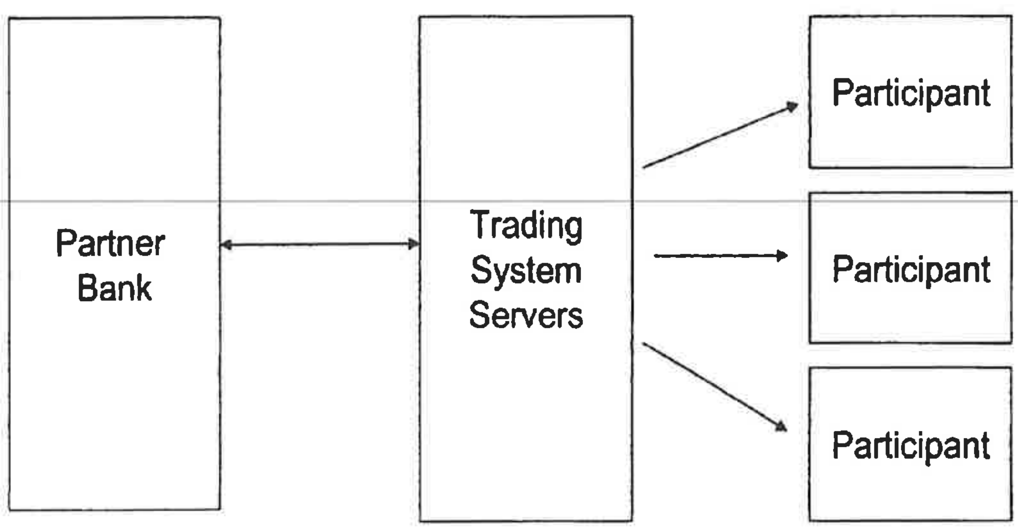
* cited by examiner

Primary Examiner — Hani M Kazimi
Assistant Examiner — Hatem M Ali
(74) *Attorney, Agent, or Firm* — Ward & Zinna, LLC

(57) **ABSTRACT**

In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. In another aspect, the present invention comprises methods for trading currency over a computer network. In another aspect, the present invention comprises software for currency trading over a computer network.

7 Claims, 14 Drawing Sheets



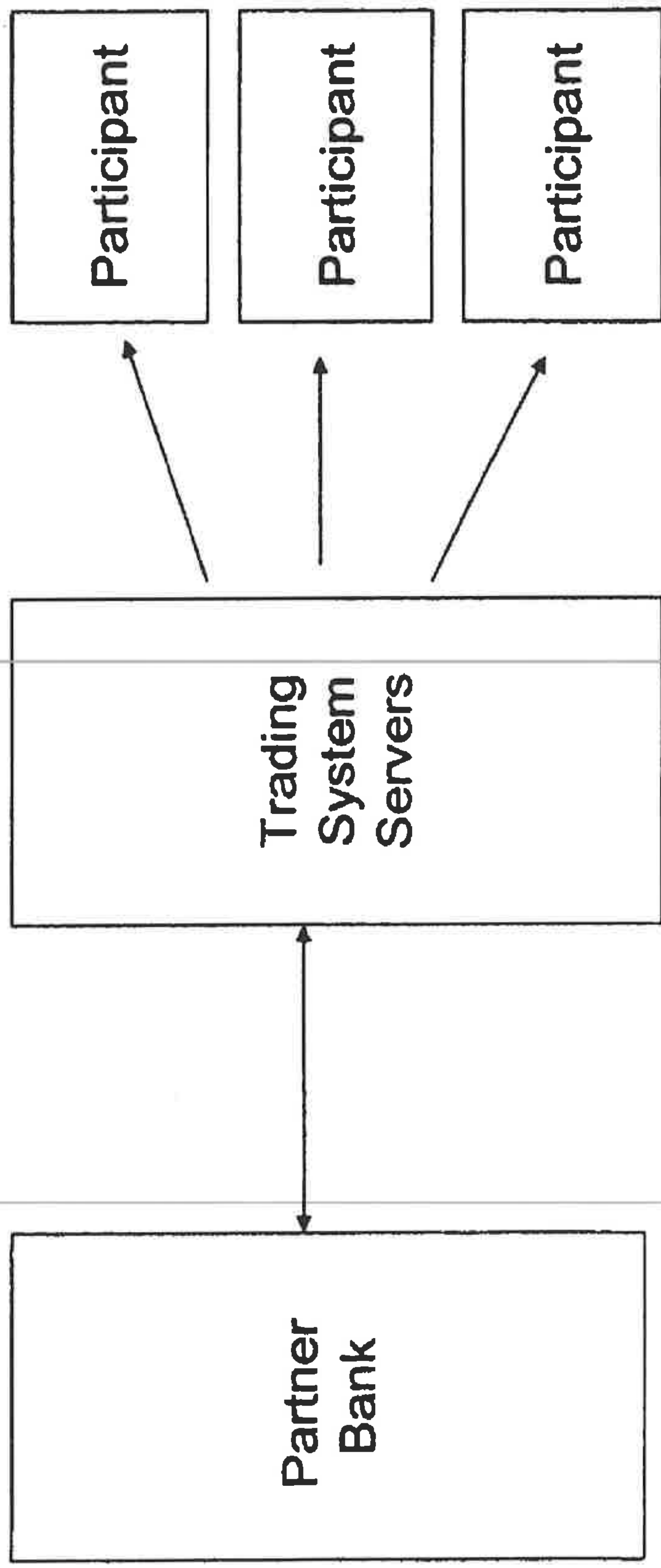

Participant
Participant
Participant
Trading System Servers
Partner Bank

Fig. 1

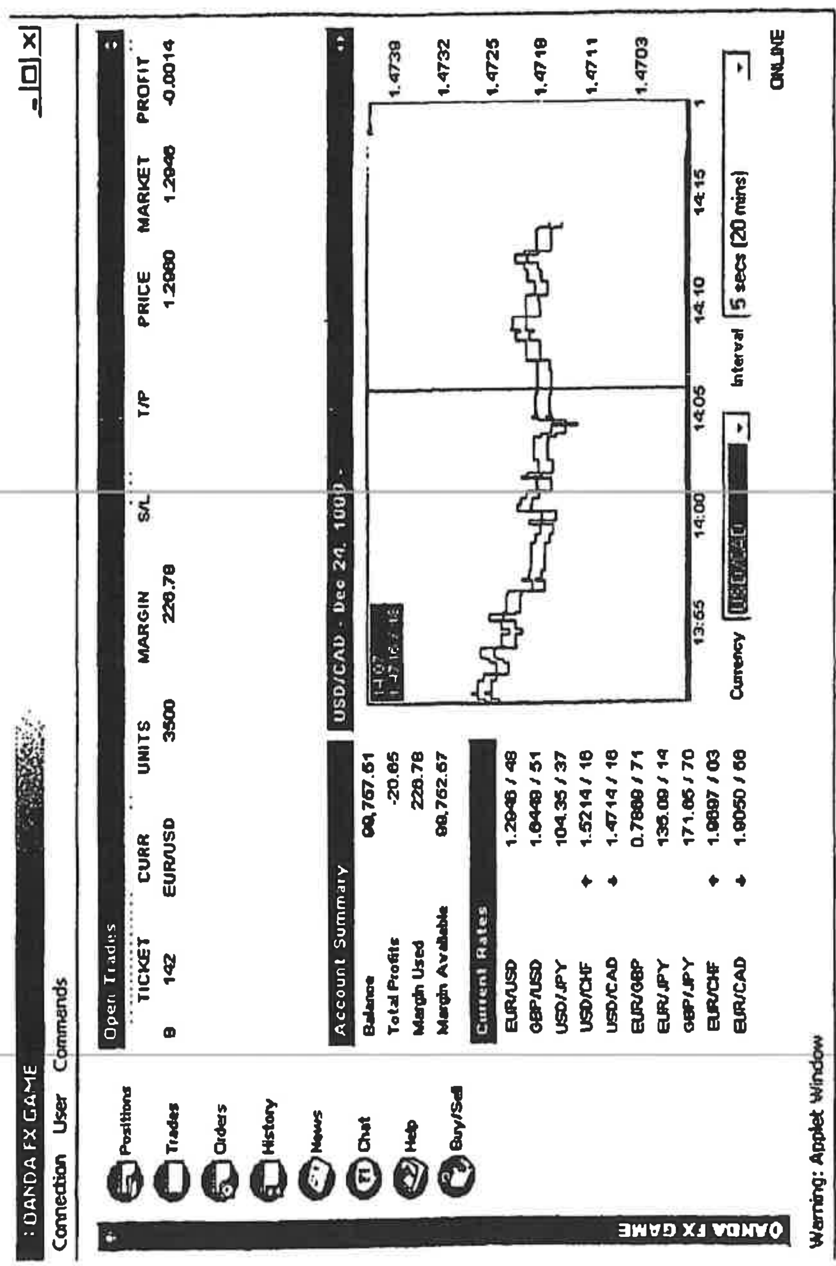

OANDA FX GAME
Connection User Commands
Positions
Trades
Orders
History
News
Chat
Help
Buy/Sell
Open Trades
TICKET CURR UNITS MARGIN S/L T/P PRICE MARKET PROFIT
142 EUR/USD 3500 228.78 1.2980 1.2946 -0.0014
Account Summary
Balance 99,757.51
Total Profits -20.65
Margin Used 228.78
Margin Available 99,752.57
Current Rates
EUR/USD 1.2946 / 48
GBP/USD 1.6448 / 51
USD/JPY 104.35 / 37
USD/CHF 1.5214 / 16
USD/CAD 1.4714 / 16
EUR/GBP 0.7868 / 71
EUR/JPY 135.09 / 14
GBP/JPY 171.65 / 70
EUR/CHF
EUR/CAD
USD/CAD
1.4739
1.4732
1.4725
1.4718
1.4711
1.4703
13:55
14:00
14:05
14:10
14:15
Currency
Interval 5 secs (20 mins)
ONLINE
Warning: Applet Window

Fig. 2

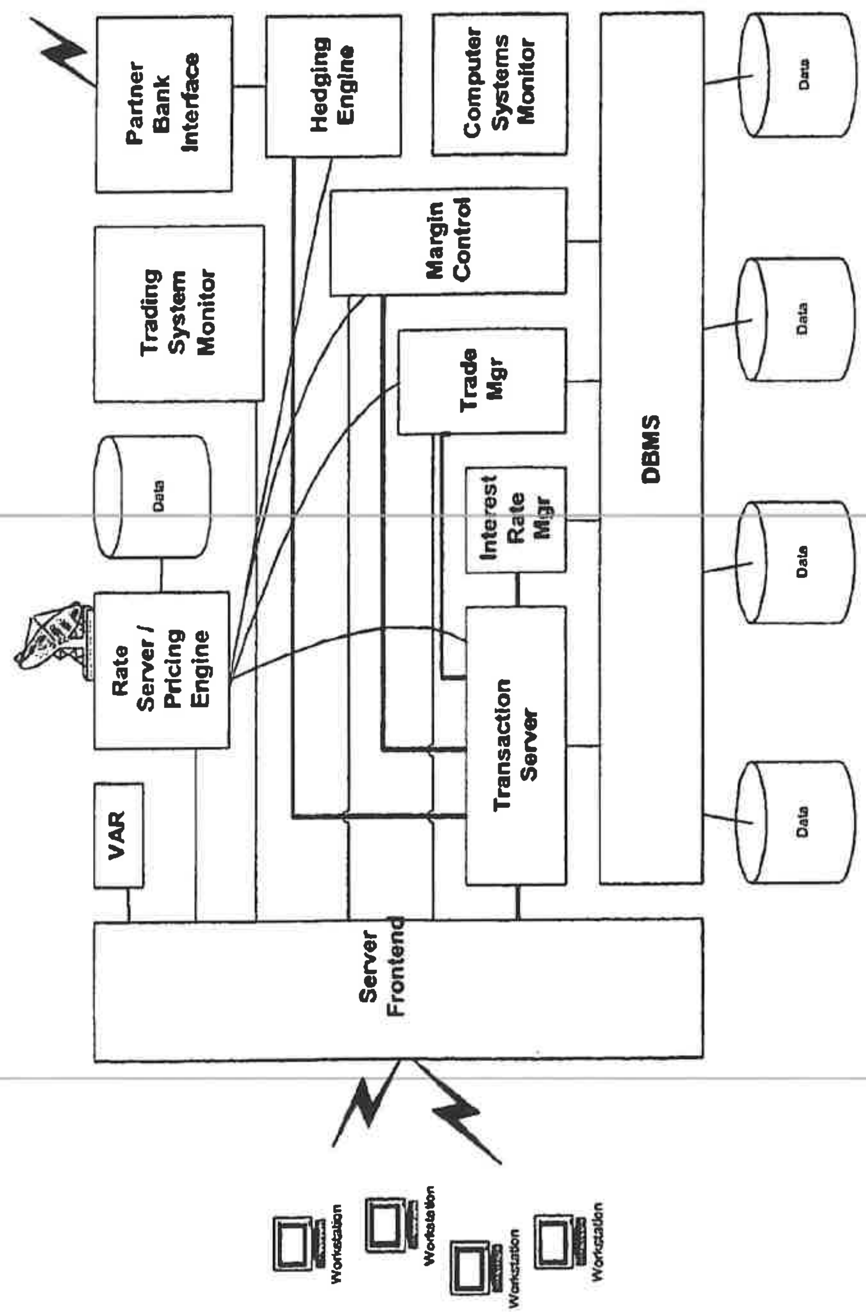

Partner Bank Interface
Hedging Engine
Computer Systems Monitor
Trading System Monitor
Margin Control
Trade Mgr
Data
Interest Rate Mgr
DBMS
Rate Server / Pricing Engine
VAR
Transaction Server
Data
Data
Data
Data
Server Frontend
Workstation
Workstation
Workstation
Workstation

Fig. 3

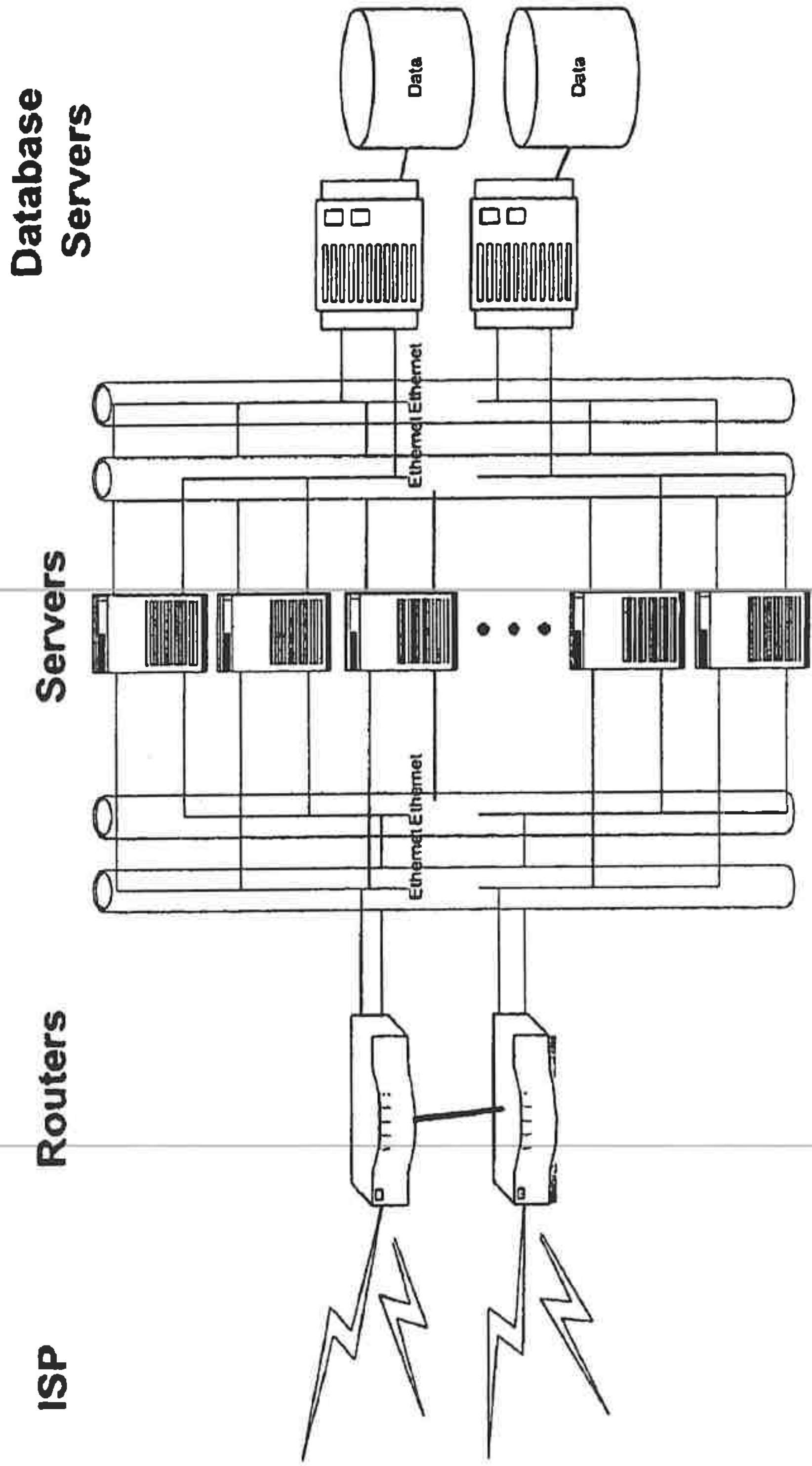
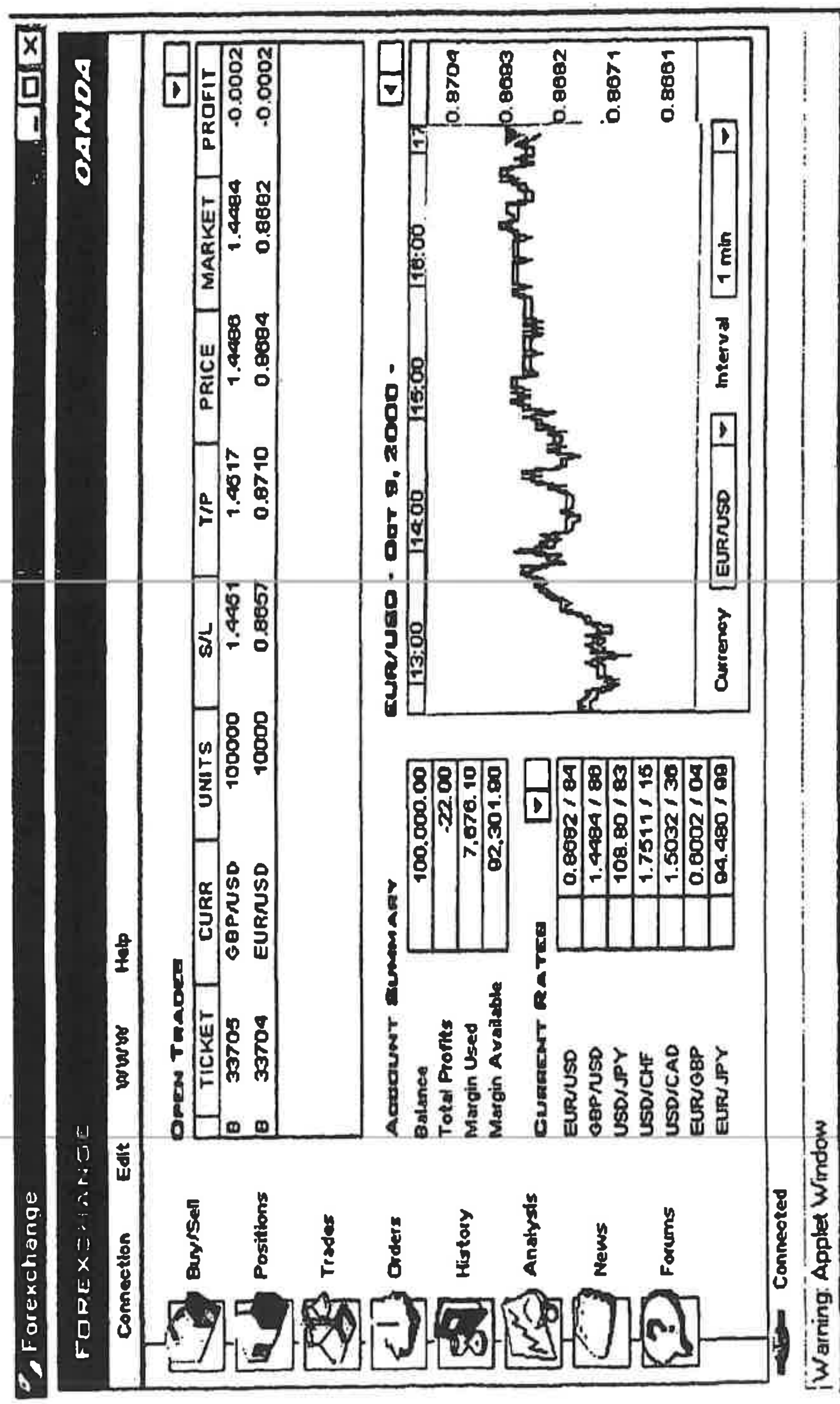

Database Servers
Data
Data
Ethernet Ethernet
Servers
Ethernet Ethernet
Routers
ISP

Fig. 4



Forexchange
FOREXCHANGE
OANDA
Connection
Edit
WWW
Help
Buy/Sell
Positions
Trades
Orders
History
Analysis
News
Forums
Connected
Warning: Applet Window
OPEN TRADES
TICKET
CURR
UNITS
S/L
T/P
PRICE
MARKET
PROFIT
B 33705 GBP/USD 100000 1.4461 1.4517 1.4486 1.4484 -0.0002
B 33704 EUR/USD 10000 0.8657 0.8710 0.8684 0.8682 -0.0002
ACCOUNT SUMMARY
Balance 100,000.00
Total Profits -22.00
Margin Used 7,676.10
Margin Available 92,301.90
CURRENT RATES
EUR/USD 0.8682 / 84
GBP/USD 1.4484 / 88
USD/JPY 108.80 / 83
USD/CHF 1.7511 / 15
USD/CAD 1.5032 / 36
EUR/GBP 0.6002 / 04
EUR/JPY 94.480 / 99
EUR/USD - OCT 9, 2000 -
13:00
14:00
15:00
16:00
0.8704
0.8693
0.8682
0.8671
0.8661
Currency
EUR/USD
Interval
1 min

Fig. 5

ACCOUNT SUMMARY

Balance	100,000.00
Total Profits	-22.00
Margin Used	7,676.10
Margin Available	92,301.90

Fig. 6

OPEN TRADES

	TICKET	CURR	UNITS	S/L	T/P	PRICE	MARKET	PROFIT
S	33760	USD/CHF	100000			1.7415	1.7413	0.0002
S	33759	EUR/USD	65000	0.8715	0.8668	0.8689	0.8692	-0.0003

Fig. 7

OPEN POSITIONS

	CURR	UNITS	AVG. PRICE	MARKET	PROFIT
S	USD/CHF	100000	1.7415	1.7407	0.0008
S	EUR/USD	65000	0.8889	0.8897	-0.0008

Fig. 8

OPEN ORDERS

	ORDER	CURR	UNITS	S/L	T/P	PRICE	MARKET	DURATION
B	246	EUR/JPY	200000	93.203	93.789	93.512	93.509	Undefined

Fig. 9

RECENT HISTORY

TRANSACTION	TYPE	CURR	UNITS	PRICE	DATE/TIME
56958	Buy Market	EUR/USD	10000	0.8684	Oct 9, 2000, 16:42
56959	Buy Market	GBP/USD	100000	1.4486	Oct 9, 2000, 16:43
57001	Take Profit	GBP/USD	100000	1.4604	Oct 11, 2000, 12:38
57021	Buy Market	EUR/USD	25000	0.8688	Oct 11, 2000, 15:04
57022	Sell Market	EUR/USD	25000	0.8689	Oct 11, 2000, 15:07

Fig. 10

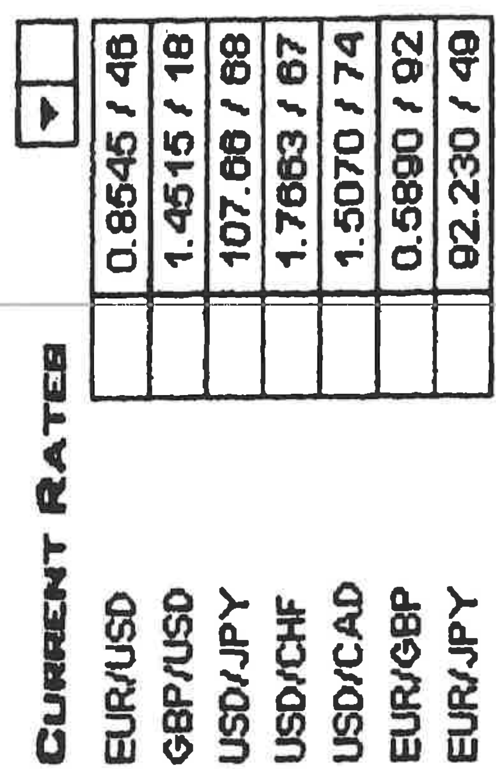

CURRENT RATES
EUR/USD 0.8545 / 48
GBP/USD 1.4515 / 18
USD/JPY 107.66 / 68
USD/CHF 1.7663 / 67
USD/CAD 1.5070 / 74
EUR/GBP 0.5890 / 92
EUR/JPY 92.230 / 49

Fig. 11

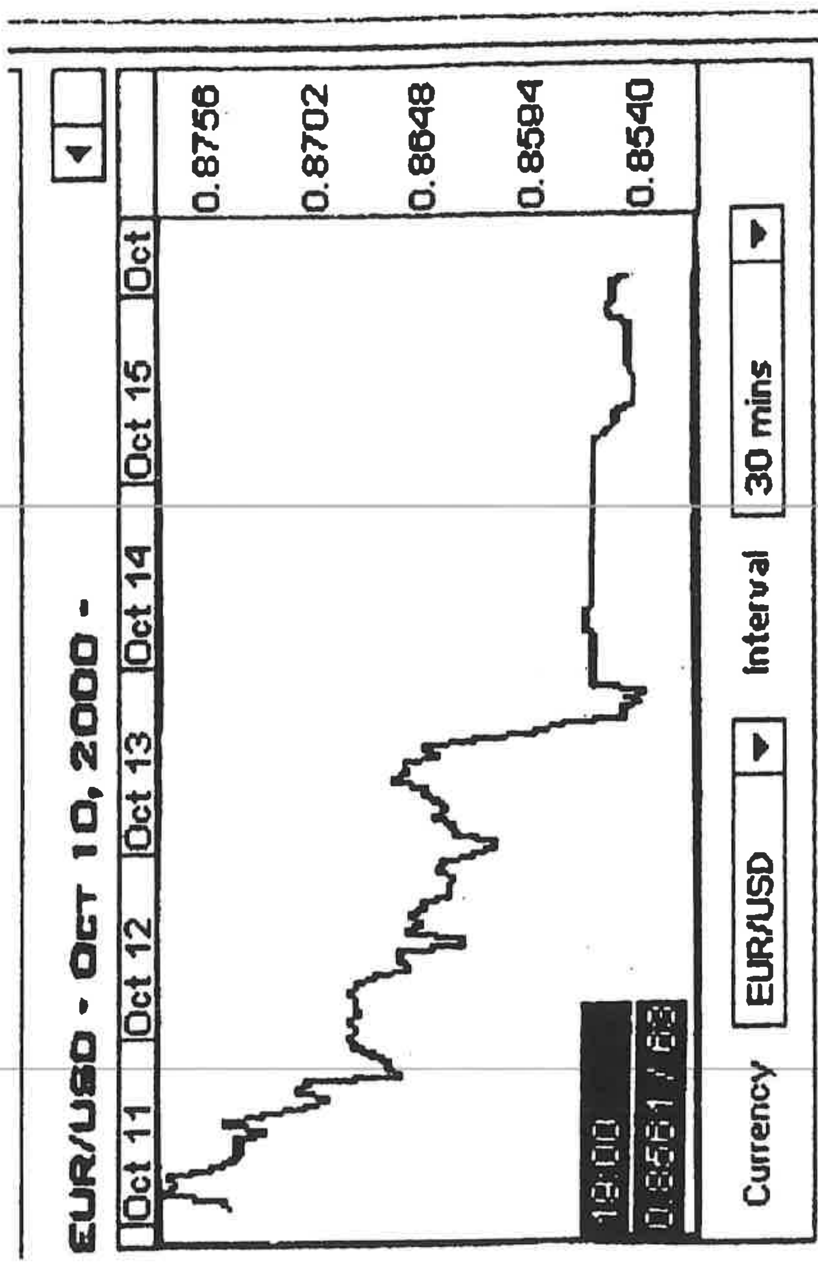

EUR/USD - OCT 10, 2000 -
Oct 11
Oct 12
Oct 13
Oct 14
Oct 15
Oct
0.8756
0.8702
0.8648
0.8594
0.8540
19:00
0.8561 / 63
Currency
EUR/USD
Interval
30 mins

Fig. 12

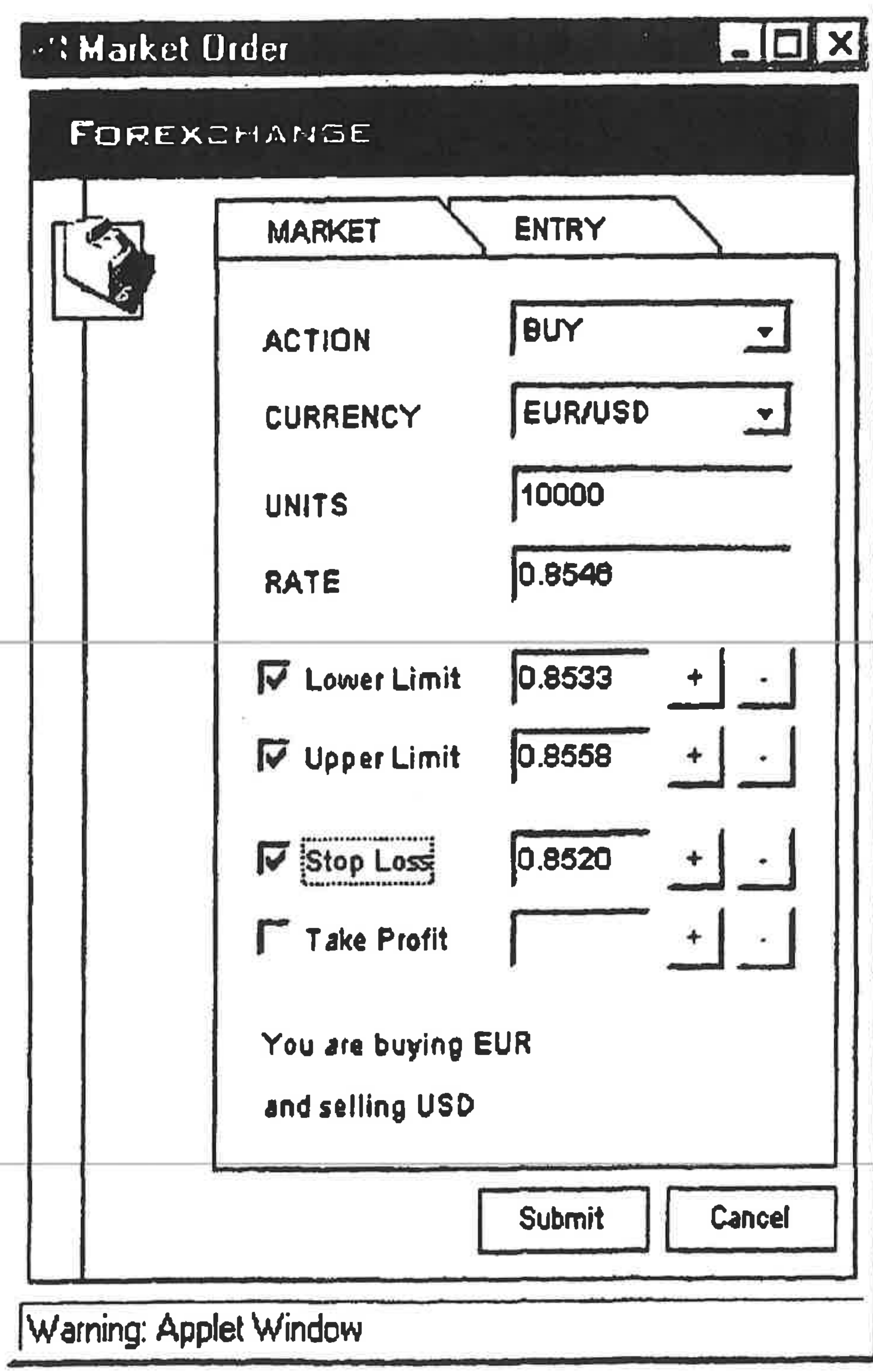

Market Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
10000
RATE
0.8546
Lower Limit
0.8533
Upper Limit
0.8558
Stop Loss
0.8520
Take Profit
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 13

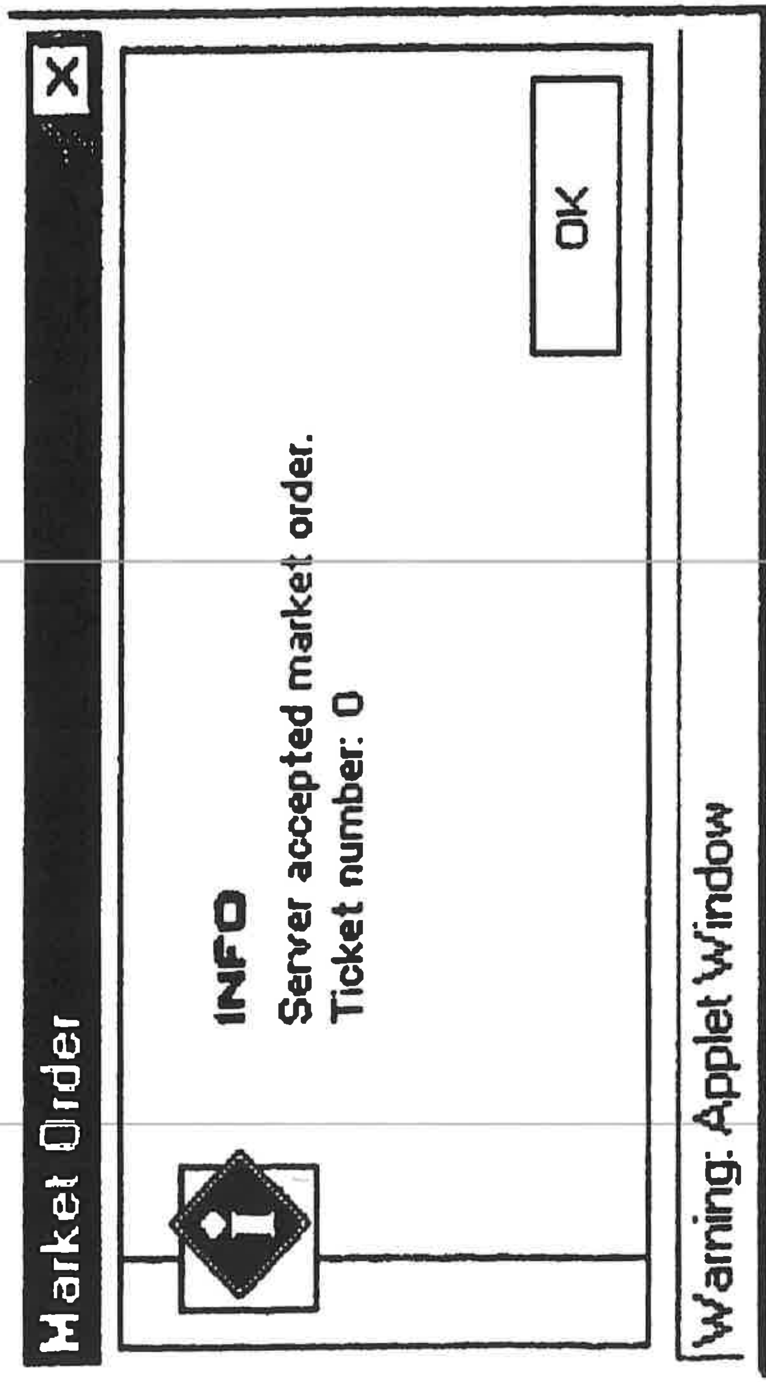

Market Order
INFO
Server accepted market order.
Ticket number: 0
OK
Warning: Applet Window

Fig. 14

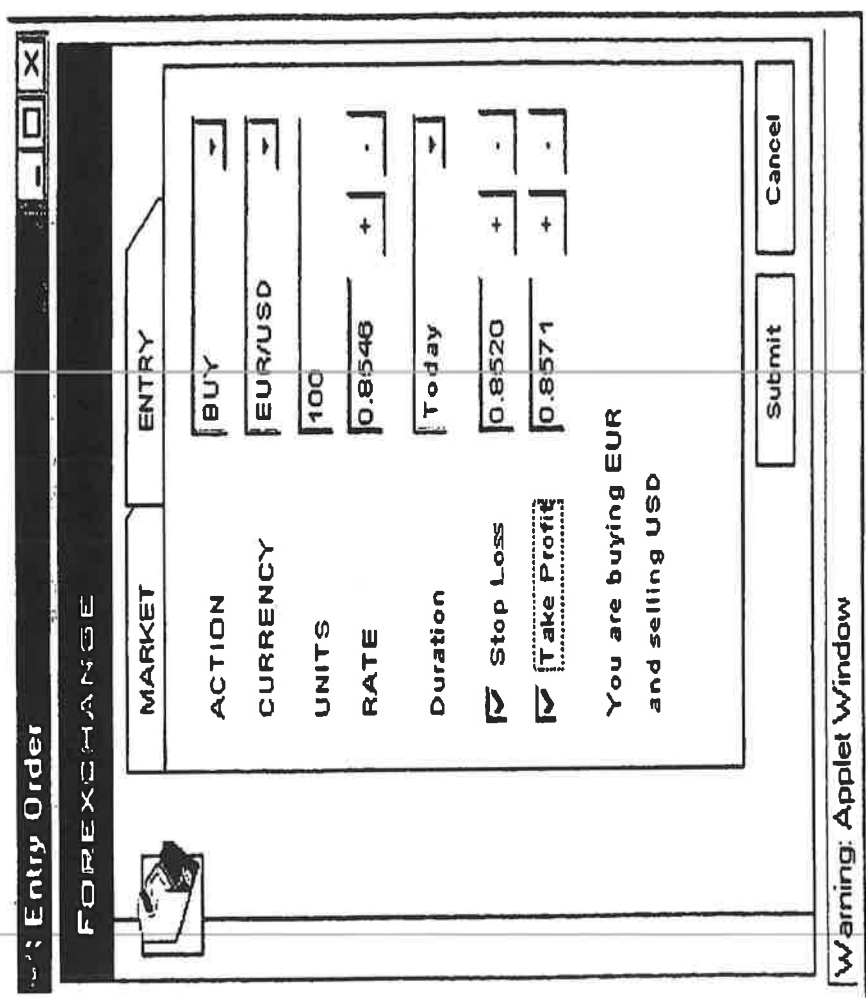

Entry Order
FOREXCHANGE
MARKET
ENTRY
ACTION
BUY
CURRENCY
EUR/USD
UNITS
100
RATE
0.8546
Duration
Today
Stop Loss
0.8520
Take Profit
0.8571
You are buying EUR
and selling USD
Submit
Cancel
Warning: Applet Window

Fig. 15

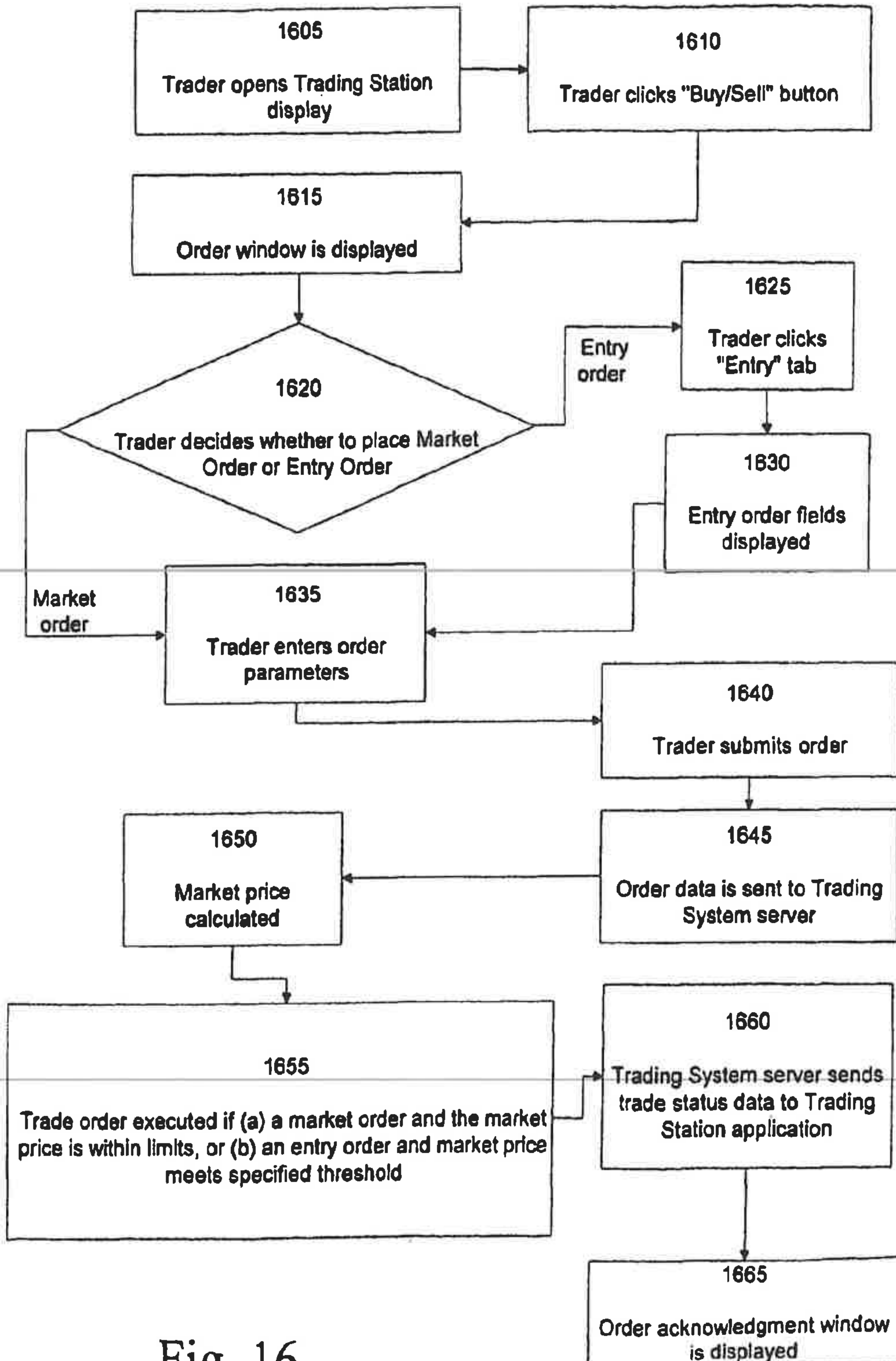

1605
Trader opens Trading Station display
1610
Trader clicks "Buy/Sell" button
1615
Order window is displayed
1620
Trader decides whether to place Market Order or Entry Order
Entry order
1625
Trader clicks "Entry" tab
1630
Entry order fields displayed
Market order
1635
Trader enters order parameters
1640
Trader submits order
1645
Order data is sent to Trading System server
1650
Market price calculated
1655
Trade order executed if (a) a market order and the market price is within limits, or (b) an entry order and market price meets specified threshold
1660
Trading System server sends trade status data to Trading Station application
1665
Order acknowledgment window is displayed

Fig. 16

CURRENCY TRADING SYSTEM, METHODS, AND SOFTWARE

CROSS-REFERENCE TO RELATED APPLICATION

The present application is a continuation of application Ser. No. 09/858,610 filed May 16, 2001 now U.S. Pat. No. 7,146,336 which claims priority to U.S. provisional application No. 60/274,174, filed Mar. 8, 2001, and incorporates the entire contents thereof herein by reference.

FIELD OF THE INVENTION

The present invention is related to currency trading; more particularly, the invention is related to trading currency over a computer network.

BACKGROUND

In a traditional on-line currency market, a trade occurs through three steps: (1) the trader specifies to a dealer the currency pair and the amount that he would to trade (but does not specify whether he would like to buy or sell); (2) the dealer specifies to the trader both a bid and an ask price and gives the trader several seconds to respond (the dealer not knowing whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or specifies whether he is buying or selling (his response must occur within a time frame of a few seconds).

But performing such a three-way handshake over the Internet is somewhat impractical because of Internet delays: the trader might not actually have a few seconds to respond before the dealer withdraws the offer. Thus, there is a need for a system and method of on-line currency trading that is based on a trading model that is superior to the three-way handshake described above.

Another problem is that many corporations have firewalls that restrict access to the corporate network, and that typically restrict access to the Internet (and to well-known services such as email, the World Wide Web, etc.) from within the corporation. This inhibits the ability of on-line trading systems to access information from and transfer information to users behind corporate firewalls.

SUMMARY

The present invention overcomes the above-described and other disadvantages of previous currency trading systems and methods. In one aspect, the present invention comprises a system for trading currencies over a computer network. A preferred embodiment comprises: (a) a server front-end; (b) at least one database; (c) a transaction server; (d) a rate server; (e) a pricing engine; (f) an interest rate manager; (g) a trade manager; (h) a value at risk server; (i) a margin control manager; (j) a trading system monitor; and (k) a hedging engine. Each of these components is described in detail below in the Detailed Description section.

In another aspect, the present invention comprises methods for trading currency over a computer network. In one embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises limits within which the currency trade will be acceptable to the end user;

(c) calculating a market exchange rate for the currency trade order; and (d) executing the order if the market exchange rate is within the specified limits.

In another embodiment, a preferred method comprises: (a) transmitting currency market information over a computer network to an end user; (b) receiving a currency trade order from the end user, wherein the currency trade order comprises a threshold exchange rate; (c) calculating a market exchange rate for the received currency trade order; and (d) executing the order (1) if the market exchange rate is or goes above the threshold exchange rate and the order is a sell order, and (2) if the market exchange rate is or goes below the threshold exchange rate and the order is a buy order.

In a further embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises limits within which the currency trade will be acceptable; and (c) if a market exchange rate is within the specified limits, receiving information from the trading system server indicating that the currency trade order was executed.

In another embodiment, a preferred method comprises: (a) receiving currency market information over a computer network from a trading system server; (b) transmitting a currency trade order to the trading system server, wherein the currency trade order comprises a threshold exchange rate; and (c) if (1) the applicable market exchange rate is or goes above the threshold exchange rate and the order is a sell order, or (2) the applicable market exchange rate is or becomes below the threshold exchange rate and the order is a buy order, receiving information from the trading system server indicating that the currency trade order was executed.

In another aspect, the present invention comprises software for currency trading over a computer network. In one embodiment, preferred software comprises: (a) software for receiving data over a computer network from a trading system server; (b) software for displaying a first graphical user interface display that: (i) displays continuously updated currency exchange rates in real-time based on data received from the trading system server; and (ii) displays action buttons, including a buy/sell button; (c) software for displaying, in response to a user clicking the buy/sell action button, a buy/sell window display that: (i) comprises trade order parameter fields; and (ii) accepts trade order data entered into the trade order parameter fields by a user; and (d) software for transmitting said trade order data to said trading system server over said computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts parties involved in a preferred embodiment.

FIG. **2** depicts a graphical user interface of a preferred embodiment.

FIG. **3** depicts modules of a preferred trading system server.

FIG. **4** depicts hardware components of a preferred embodiment.

FIG. **5** depicts a graphical user interface of a preferred embodiment.

FIG. **6** depicts an account summary table display.

FIG. **7** depicts an open trades table display.

FIG. **8** depicts an open positions table display.

FIG. **9** depicts an open orders table display.

FIG. **10** depicts a transactions table display.

FIG. **11** depicts a currency rates table display.

FIG. **12** depicts a currency exchange rate graph display.

FIG. 13 depicts a buy/sell pop-up window display.

FIG. 14 depicts an acknowledgment window display.

FIG. 15 depicts an entry order display.

FIG. 16 depicts steps of a method of a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The Currency Trading System of a preferred embodiment (hereinafter "Trading System") of the present invention allows traders to trade currencies over a computer network. Preferably, this computer network is the Internet, and the subsequent description herein is primarily in terms of the Internet. However, those skilled in the art will recognize that the following description also applies to other computer networks. Traders interface to the system using ordinary Web browsers running feature-rich Java applets; they obtain real-time data feeds of current exchange rates, they can analyze past exchange rates using graphical tools, they can review their current portfolio and past trades, and they can place buy and sell orders in the real-time market. Businesses interface to the system using an API. Innovative features that set the Trading System apart from the competition are: (i) extremely low spreads on the order of a few basis points, (ii) the ability to trade very small amounts as low as $1, and (iii) 24 hour a day, 7 days a week availability. This system has the potential to revolutionize the way currency trading is done and to open up currency trading to a new, large market segment of investors and speculators for whom currency trading is not feasible today. Moreover, it allows businesses to address their currency exchange requirements in the most cost-effective and efficient way.

A description of the preferred server infrastructure used in the Trading System follows. We first give a brief introduction of the system as a whole.

The Trading System involves three components (see FIG. 1): (1) traders that are distributed around the world; (2) Trading System servers; and (3) "Partners" consisting of the financial institution(s) through which real currency exchange trades are executed, and from which real-time data feeds are obtained.

Traders communicate with Trading System servers over a secure, encrypted Internet connection to review their accounts, to monitor currency exchange market conditions, or to initiate currency exchange trades. The Trading System servers are preferably connected to the partners' back-office systems, using direct, private lines.

A trader trades with the Trading System similar to the way she currently trades with a broker, except that the trading is over the Web, occurs 24 hours a day, 7 days a week, and allows very small trades with very low spreads. Moreover, an initial deposit, which may be as low as $20, can be charged to a credit card to get started. Alternatively, the trader can transfer initial funds directly to the Partner bank to be credited to her Trading System account.

The end user interface to the Trading System is a Web page that can be displayed on any standard Java-enabled browser. The Web page (one version is shown in FIG. 2; a second, preferred version is shown in FIG. 5) depicts a summary of the trader's current position, recent trading activities, profit/loss performance of the portfolio, and a graphical display of recent past performance of the currencies the trader has positions in, as well as real-time exchange rates.

As discussed above, in a traditional on-line currency market a trade occurs through three steps: (1) a trader specifies a currency pair and an amount he would like to trade (and does not specify whether he would like to buy or sell); (2) a dealer provides both a bid and an ask price to the trader and gives the trader several seconds to respond (the dealer does not know whether the trader will buy, sell, or reject the offer); and (3) the trader either rejects the offer or accepts the offer and specifies whether he is buying or selling, within the response period set by the dealer.

In the Internet domain, this type of three-way handshake is problematic. The timing constraints are difficult to implement because of frequent delays in transmission over the Internet. To overcome this problem, the present invention uses a "two-way handshake," in which: (1) a trader specifies in her trade order: (a) a currency pair; (b) a desired amount to trade; (c) whether she wishes to buy or sell; and (optionally) (d) upper and lower limits on an acceptable exchange rate; and (2) a dealer (in this case, a preferred Trading System) executes the trade using the most current "market rates" (as calculated by the system). However, the system only executes the order if the calculated market rate lies above any specified lower limit and below any specified upper limit. Note that this method does not require the use of timing constraints, and thus avoids the Internet-implementation disadvantage of previous methods.

In the present invention, trades can be initiated by the trader at the push of a button. A trading request form pops up with fields properly initialized so as to minimize the number of keystrokes required. A trader may elect to execute a trade right away, in which case the buyer of a currency will buy at the current exchange rate market offer price. Conversely, a trader can sell a currency at the current bid price. A range of automatic trading options is available, including setting bid/offer prices with a certain duration and "all-or-nothing" rules. Furthermore, the trader can limit her risks by placing stop-loss orders that are executed automatically. Similarly, she can lock in profits, by issuing take-profit orders.

All communication between the trader's browser and the Trading System server occurs through the Internet, preferably using the strongest available encryption (e.g., 128 bit keys). Moreover, the trader must authenticate herself using private passwords or certification keys obtained from certification authorities, such as Verisign or Entrust.

A request for a market trade preferably proceeds as follows: the trader, at a push of a button, obtains a trade order ticket in a popup window on the browser with key fields pre-initialized (see FIG. 13). When the trade order is issued, again by the push of a button, a message is sent to a Trading System server, where the market price is calculated based on such factors as market data, size of the transaction, time of day, the Trading System's current exposure, and predictions on market direction. The trade order is executed using this market price. (The trader can specify limits, so that the trade occurs only if the price falls within these limits.) As such, the Trading System operates as a market maker. A message is then sent back to the trader with specific trade details, which is displayed in a popup window (see FIG. 14) on the trader's browser together with a transaction id (for future reference). Moreover, an open orders table (see FIG. 9) and current portfolio summary table (not shown) is updated to reflect the change.

Alternatively, the trader can issue in a similar manner an entry order (see FIG. 15) that requests a trade be made when the currency exchange rate reaches a specified threshold. The trader may specify how long the entry order will be valid.

Referring to the attached figures, a preferred embodiment comprises a method of trading that in turn comprises the following steps (see FIG. 16): At step 1605, a trader desiring to trade opens a Trading Station display, and at step 1610 clicks a "Buy/Sell" button **510** on the Trading Station display (see FIG. **5**). At step **1615** an order window is displayed (see FIG. **13**). At step **1620** the trader decides whether to place a market order or an entry order. If a market order, then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **13**). If an entry order, then the trader proceeds to step **1625** and clicks an "Entry" tab **1320** (see FIG. **13**). At step **1630** entry order fields are displayed (see FIG. **15**). Then the trader proceeds to step **1635** and enters desired order parameters (as shown in FIG. **15**).

Once order parameters are entered at step **1635** the trader submits the order by clicking a "Submit" button **1310** (see FIG. **13**) if the order is a market order, or clicking a "Submit" button **1510** (see FIG. **15**) if the order is an entry order. At step **1645** data describing the order is sent by the Trading Station application to a Trading System server, where the data is stored. At step **1650** a current market price for the currency the trader desires to purchase is calculated. At step **1655** the trader's order is executed if (a) the trader's order is a market order and the calculated market price is within the limits set by the trader in the market order form at step **1635**; or (b) order is an entry order and the calculated market price meets the threshold(s) specified in the Entry order form at step **1635**.

At step **1660** the Trading System server sends trade status data to the trader's Trading Station application. This data includes an indication that the order has been executed, if that is the case, and at any rate includes an indication that the order has been received. At step **1665** the Trading Station application displays an order acknowledgment window (see FIG. **14**) that displays order status information.

Over time, the Trading System will accumulate an imbalance in its currency portfolio and, at times, it will need to neutralize its risk exposure to adverse currency fluctuations. The Trading System Pricing Engine can influence its exposure by setting its price quotes accordingly. Moreover, it can close out its positions periodically or take hedging positions by executing larger trades through its Partners. Preferably the Trading System's positions are managed based on state-of-the-art trading models. Preferred trading models are described in U.S. patent application no. [METHODS FOR TRADE DECISION MAKING, to Olsen et al.], filed May 14, 2001, the contents of which are incorporated herein by reference, as well as in U.S. provisional application No. 60/274,174, filed Mar. 8, 2001.

The Trading System servers preferably operate 24 hours a day, 7 days a week. These servers interface with the traders over the Internet on the one hand and on the other hand with the Partner's back-office operations. Using standard, state-of-the-art database technology, it maintains the accounts of all traders and executes trades issued by the traders. The Trading System thus plays the role of a market maker in that it internally aggregates all trades and only occasionally balances its internal positions by trading larger sums through the Partner. These larger trades are issued to the Partner in an automated way. The Trading System also takes hedging positions so as to minimize risks on the unbalanced portions of the traders' account aggregates.

Partner's Role

The Partner maintains all actual funds. It is the source and target of all fund transfers to and from customers; it maintains the aggregate accounts; and it executes all trades issued automatically by the Trading System servers. From a legal point of view, all funds must be maintained in money market instruments. Hence, the Partner will maintain a Long and a Short money market fund for each currency supported.

Overview of Currency Trading System Server Internal Design

The Trading System server architecture is designed to be:

(1) modular, in order to: (a) simplify development (time-to-market); (b) easily allow extensions and modifications; (c) ensure correctness and robustness, and (d) keep it maintainable;

(2) efficient, in order to provide fast response times to online users and minimize the computational and networking resources required to support the service;

(3) scalable, in order to support (with suitable computing infrastructure) a large number of online users and high transaction volumes; and

(4) fault resilient, so that any individual failure of a computing node or network connection does not interrupt service.

The Trading System server software preferably runs exclusively on Unix platforms, and is composed of the following modules, each with a distinct set of responsibilities (see FIG. **3**):

(1) Database (DBMS) **310**. This is the heart of the server. It keeps track of customer profile information, all customer accounts and all transactions, and ensures that atomicity, consistency, independence, and durability ("ACID") properties are maintained. The database is the reference point for all information kept by the system.

The database is preferably a standard commercially-available SQL database, configured for full replication for reliability, availability, and improved performance. The preferred embodiment is based on IBM's DB-2 product line, but Oracle, for example, could also easily be used.

(2) Server Front-end **315**. The server front-end **315** is responsible for all communication with the Web-based clients. It supports both persistent and non-persistent connections to the traders. The persistent connects are used primarily for periodic (i.e., every few seconds) transmission of the latest currency rates so that the traders can update the currency graphs and tables in real-time. Using persistent connections significantly reduces protocol processing overhead and reduces network bandwidth requirements. Non-persistent. connections are used for all transaction-oriented requests, such as orders, transaction history requests, logins, etc. All transaction-oriented communication between the trader browsers and the Server Front-end occurs fully encrypted, while rate information is transmitted in unencrypted form for efficiency reasons.

Traders preferably communicate with the server using a request-response type of protocol. The Server Front-end **315** interprets each request it receives and, for each, takes appropriate action. For login requests, it sets up appropriate data structures so that all future requests can be serviced in the most efficient way. It also sets up encryption keys for secure communication, and logs the start of a new session with the Transaction Server. For rate requests, it returns the requested rates it obtains from the rate server. For orders, it executes the orders by issuing appropriate requests to the transaction server after checking the margin requirements, the availability of funds, and using rates as determined by the pricing engine. For stop-loss/take-profit and fixed-price orders (that may get executed in the future), the Trade Manager **365** is also informed. For each trade that gets executed, the Hedging Engine **340** and Margin Control **350** modules are informed, so that they always have an up-to-date snapshot of the state. For transaction history, the appropriate information is returned to the client after obtaining it from the Transaction Server **355**.

The Server Front-end **315** also encapsulates a standard Web server (a la Apache), that services other trader requests that entail formatted text; this includes all of the Help pages, large transaction history requests, and server monitoring information. The Server Front-end **315** also acts as a Firewall.

Internally, the Server Front-end **315** is structured as a set of threads that service one request after another. The threads allow concurrent request servicing so that many requests can be serviced in parallel.

(3) Rate Server/Pricing Engine **325**. The Rate Server obtains currency exchange rate information from a variety of external rate sources and stores it locally. The Pricing Engine computes the currency exchange rates that the traders see and that are used for trading. These are computed from the currency exchange rates obtained from the external rate sources, the directional movement and volatility of the market, the current Trading System exposure and a number of other parameters. The computed rates are made available to the other modules of the system, and are also stored on persistent media. Various methods of calculating such rates are known to those skilled in the art. A preferred method is described in U.S. patent application Ser. No. 09/764,366 filed Jan. 18, 2001, to Müller et al.

Traders can request historical rate data so that they can graphically display the movements of any pair of currencies. The Rate Server serves such requests and preferably has several years of currency exchange rates available for this purpose.

For fast response time, the Rate Server caches in memory all of the frequently and recently requested rates so as to minimize the number of disk accesses required.

(4) VAR (Value at Risk) server **320**. This server obtains and serves Value at Risk information. Various methods of calculating VAR are known in the art. A preferred method is disclosed in U.S. Provisional Patent Application No. 60/200,742, filed May 1, 2000, to Müller.

(5) Transaction Server **355**. This server encapsulates all transaction functionality and communicates the transactions to the Database **310** server (which runs on a separate host) after validating the transactions. The Transaction Server **355** also updates all other modules that need to be informed of new transactions. Finally, the Transaction Server **355** informs the currently online traders when a transaction (that may have been issued by a stop-loss, take-profit, or limit order daemon or by the Margin Control Manager) takes place.

(6) Interest Rate Manager **360**. The Interest Rate Manager **360** periodically (for example, every few minutes, every few seconds, or tick-by-tick) goes through all trader accounts to compute the interest rate due or owed based on the instruments currently in the portfolio, each resulting in a transaction of the trader account. The portfolio information is obtained through the Transaction Server **355**. The interest rates used are obtained from external sources, and the history of interest rates are stored on persistent storage. Because real-time (or near real-time) information is used, the Interest Rate Manager is capable of calculating, paying out, and collecting interest by the second. Interest calculation formulas are known to those skilled in the art, and any appropriate formula can be used in the Interest Rate Manager without departing from the scope of the invention. An example is the formula

$$A = P\left(1 + \frac{r}{m}\right)^{mt}$$

where P is the principal, r is the annual interest rate, t is the time (in years) over which interest is earned, m is the number of times per year that interest is compounded, and A is the

amount owed (principal plus interest). The interest earned during time t is A−P. Thus, for example, if the annual interest rate is 3%, and the interest is compounded daily, then the interest I_k earned over each time period $T_k=t_k-t_{k-1}$, where each t_k is a clock-time (i.e., a particular day-hour-minute-second-fraction-of-a-second) to the nearest second (thus T_k is in seconds), is calculated according to the formula $I_k=A_k-P_k$, where P_k is the principal (the amount earning interest, not the "original" principal) at time t_{k-1}, and

$$A_k = P_k\left(1 + \frac{0.03}{365}\right)^{\frac{365T_k}{31,536,000}}.$$

Since there are 31,536,000 seconds per year,

$$\frac{T_k}{31,536,000}$$

is the time in years over which the interest is being calculated. Similar formulas can be used when t_k is given to the nearest tenth, hundredth, or other fraction of a second. If interest is compounded continuously, those skilled in the art will recognize how to apply the well-known formula $A=Pe^{rt}$ appropriately. Thus, to calculate interest on a tick-by-tick basis, the above formulas can be used, with T_k representing time between ticks.

(7) Trade Manager **365**. The Trade Manager **365** continuously checks whether a trade should be executed on behalf of a trader, and if so executes the trade by interacting with the transaction server. The Trade Manager **365** consists of multiple subcomponents: (a) a stop-loss daemon continuously checks to see whether stop-loss orders should be executed and, if so, executes them through the Transaction Server **355**; (b) a take-profit daemon continuously checks to see whether take-profit orders should be executed and, if so, executes them through the Transaction Server **355**; and (c) a limit-order daemon continuously checks to see whether a limit order should be executed and, if so, executes it through the Transaction Server **355**.

The daemons continuously monitor the current rates to determine whether action is required. Moreover, each of the daemons caches in memory all of the orders that it may need to execute. They keep the orders suitably sorted so that they can take fast action when necessary; for example, the stop-loss daemon sorts the orders in descending order of stop-loss price, the take-profit in ascending order of take-profit price.

(8) Margin Control Manager **350**. This module continuously monitors the margin requirements of all trader accounts. When necessary, the Margin Control Manager **350** will liquidate some (or all) of a trader's holdings. It caches in memory all of the information necessary for this computation, sorted in decreasing order of risk, so that it can take swift action when necessary. Holdings are liquidated through the Transaction Server **355**, when necessary.

(9) Trading System Monitor **330**. This module continuously monitors the current state of the Trading System. Among others, state parameters include: (a) current Trading System currency positions; (b) current margin situation; (c) a summary of stop-loss, take-profit, and limit orders that exist; (d) the number of currently online users; (e) the number, size and type of trades executed per second; and (f) a summary of the account positions held by the users.

This information is made available (a) to the Pricing Engine **325** (where it is-used to set the currency exchange rates made available to the traders), (b) to the Hedging Engine **340** so that it can determine when to issue trades with the Partner Bankend Bank, and (c) to system operators and Trading System financial engineers in real time via a feature-rich Web interface. Moreover, this information is logged on persistent storage for later, off-line analysis.

(10) Hedging Engine **340**. This module continuously monitors current Trading System currency positions, the positions held in the trader accounts, recent trading activity, and the market direction and volatility to determine when to issue a trade with the backend Partner Bank. Various methods of performing such calculations are known to those skilled in the art. The Hedging Engine **340** preferably uses the hedging tool described in U.S. patent application Ser. No. 09/764,366, filed Jan. 18, 2001, to Müller et al., the contents of which are incorporated herein by reference.

(11) Partner Bank Interface **335**. This module communicates directly with the backend Partner Bank to issue trades and obtain account information.

(12) Computer Systems Monitor **345**. This module continuously monitors the operation and state of the computer systems on which the Trading System is running. Besides error conditions, such metrics as memory, processor, disk, and network utilization; paging activity; the number of packets sent over the various networks; the number of transactions; and the number of processes and threads are of interest. This information is made available to system operators in real time via a feature-rich Web interface and local consoles. In addition, it is stored on persistent storage for later, off-line analysis.

The Server modules described above are structured so that they can run independently as separate processes that can be independently mapped onto an arbitrary computer within a cluster. Moreover, each of the modules can run in replicated form, providing both fault tolerance and increased throughput.

Preferred Physical Organization of the Trading System Server

A Trading System Server of a preferred embodiment runs on a hardware base consisting of a cluster of hosts and disk farms connected by networks. All of the hardware components are preferably replicated for fault tolerance, as depicted in FIG. **4**.

The cluster is connected to multiple ISPs so that if one ISP goes down, traders can still communicate with the server. The ISPs are connected to the Server through a pair of routers **410** that monitor each other; if one of them goes down, then the other will automatically take over.

For security reasons, the Database **310** is on a separate back-end network; this way, it is not connected directly to the Internet and can only be accessed by the Transaction Server **355**. The Database **310** is setup in a dual configuration, so that the system can continue operating with a single database failure. All disks are mirrored, again, so that any single disk failure will not result in a loss of data.

All of the other server processes run on a cluster of servers **420**, connected to the Internet routers **410** on the one side, and connected to the backend database **430** on the other side. A virtually unlimited number of servers can be used in the cluster, allowing the system to scale up to support a large number of users. The servers can be partitioned by functionality, allowing specialized servers to be used, optimized for the particular functionality. For example, the Rate Servers **325** need minimal CPU power, and only a limited amount of memory. They also can be replicated easily without the introduction of any complexity or overhead. Hence, smaller, less costly hardware can be used for this purpose.

After login, traders typically communicate with a particular server in "sessions" for performance reasons. Using sessions improves cache locality, resulting in far fewer database accesses, and it allows the cost of creating session encryption keys to be amortized over many communication actions. For load balancing purposes, the trader software is directed to henceforth communicate with the least loaded server at the time when the trader first logs in. In case of severe load imbalances, individual traders are redirected to new, less loaded servers. If any of the servers crashes, then the client software that was communicating with the crashed server will detect the failure and automatically (transparently to the user) go through a new login procedure.

User Interface Description

Overview

The following is a description of a preferred user interface of a preferred Trading System. The main user interface display is called a "Trading Station," and it is used for all interactions with the trading system by a trader, such as analyzing changes in currency exchange rates, reviewing the trader's current currency positions, reviewing the trader's past transactions, or issuing buy and sell requests. The key features of the Trading Station are that: (1) it runs on any of the popular Web browsers connected to the Internet; (2) it displays continuously updated currency exchange rates in real-time; (3) it displays all pertinent information in one window; and (4) all interaction with the server occurs over fully-encrypted Internet connections.

System Requirements

The User Interface is preferably implemented in Java so as to run on any browser with JDK 1.2 support, which includes all Netscape Navigators versions 4.2 and up as well as Microsoft's Internet Explorer versions 5.0 and up. The Trading Station is preferably supported for Windows 95, Windows 98, Windows 2000, Windows NT, Linux, Sun Solaris, and other Unix-based operating systems.

If operated from behind a firewall, then the Trading Station may operate properly only if the firewall allows HTTP requests to Port **90**. Many corporations have firewalls that restrict access to the corporate network to well-known services such as email. Typically this restriction is accomplished by restricting the ports that may be used. For example, Port **80** is typically used for http (Web-based) traffic. Some firewalls inspect traffic going through Port **80** to ensure that the port is being used only for Web-based traffic. This is problematic for trading systems that do not use http messages—it causes users behind corporate firewalls to be inaccessible. However, a preferred embodiment of the present invention overcomes this obstacle by prefacing Trading System messages with standard http headers to make them appear to be http requests and responses, even though they are not.

Log in Procedure

In order to log in, a trader must be a registered user. Registering is preferably free and can be accomplished by clicking on a "new users" link on a login page. Logging in requires a user to provide a user-ID and password. If a trader forgets her password, she can click on the "forgot your password" link and fill in the information requested; her password will be then be emailed to her.

If user-ID and password are entered correctly, a small window appears indicating that the Trading Station is being loaded. After a short time, a larger window appears with the Trading Station Graphical User Interface shown in FIG. **5**. Once the Trading Station is properly loaded, the contents of the small window is changed to include a number of useful links. It is important that this small window not be closed while the Trading Station is to remain in operation, although it may be minimized so as not to be in the trader's way. (This small window is necessary due to the limitations of typical Java implementations on most browsers that would otherwise not allow a trader to continue browsing the Web while the User Interface is active.)

Main Window of Trading Station

The Trading Station user interface is shown in FIG. **5**. It can be resized to a convenient size, by using the standard resizing mechanisms supported by the trader's operating system's windowing system.

The Trading Station is preferably partitioned into a number of components that each serve a different purpose:

(A) Action buttons: a vertical panel located on the left hand side of the Trading Station contains a set of action buttons that allow a user to perform the most common operations.

(B) Menus: a set of pull-down menus across the top allows a user to invoke additional operations.

(C) Account summary: an area in the middle of the Trading Station that gives a summary of the user's account.

(D) Table: an area located across the top of the Trading Station that is used to display various information in tabular format. The information displayed depends on the most recently clicked Action Button. It might display currently held instruments, current open positions, or a history of recent transactions.

(E) Currency rates: an area at the bottom left that displays various currency rates. These rates are continuously updated in real time.

(F) Graph: located at the bottom-right corner, graphs display currency rates over time. The graphs are also updated in real-time as new rates become available.

Subsequent sections describe each of these components in detail.

Action Buttons

The Trading Station preferably has the following action buttons in a panel on the left side. Clicking the appropriate button will-invoke the described operation:

Buy/Sell: Pops up a Buy/Sell window, from which a trader can issue trade requests. (See the description of the buy/sell window (FIG. **13**) for more information.)

Positions: Displays the currently open positions in a table. (See the description of the Open Positions Table (FIG. **8**) for the contents of the table.)

Trades: Displays the currently open trades in a table. (See the description of the Open Trades Table (FIG. **7**) for the contents of the table.)

Orders: Displays the open orders (that may be executed some time in the future) in a table. (See the description of the Open Orders Table (FIG. **9**) for the contents of the table.)

History: Displays a recent history of the trader's transactions in the table. (See the description of the History Table (FIG. **10**) for contents of the table.)

Analysis: Pops up a new browser window with access to a number of analysis tools that might help in making trading decisions.

News: Pops up a new browser window with the latest currency news.

Forums: Pops up a new browser window with access to a number of forums (sometimes known as newsgroups) that allow a trader to participate in discussions with other traders and currency trading experts.

Pull-Down Menus

There are preferably 5 pull-down menus (not shown), each offering different operations or services:

Connection Menu: (1) Disconnect: disconnects the Trading Station from the Trading System server. The Trading Station will remain open, but currency rates will no longer be updated, and transactions will not be possible. (2) Connect: connect the Trading Station to the server, so the trader is back on line. (3) Quit: quit and exit this application.

Account Menu: (1) Transaction history: pop up a new browser window to display an extensive list of all transactions that occurred on a trader's account. See the Transaction History section (relating to FIG. **10**) for a description of what is displayed. (2) Clear account balance and P/L: for those who have incurred large losses on their account, this operation allows a trader to start over again with a cleared P/L and new funds in the account. This feature is primarily useful when an account on the Trading System is used as a game—i.e., no real money changes hands. (3) Add funds to account: for a game account, add funds to the account; for a real-money account, transfer money into the account from the trader's credit card or obtain instructions on how to wire transfer money into the trader's account. (4) Buy/Sell: issue a trade or market order (see FIG. **13**). (5) Open positions: display the open positions in a table (see FIG. **8**). (6) Open trades: display all open trades in a table (see FIG. **7**). (7) Open orders: display all open orders in a table (see FIG. **9**). (8) Recent transaction history: display the most recent transactions in a table (see FIG. **10**).

Commands Menu: (1) Change passwords. (2) Graph: specify the currency pair to be displayed in the graph.

Information Menu: (1) Interest rates: display interest rate information in a separate browser window. (2) Market News: display up-to-date currency market news in a separate browser window. (3) Analysis tools: use an analysis tool in a separate window. (4) Forum: participate in various forums related to currency trading. (5) Rankings: see a list of the most successful currency traders using the Trading System.

Help Menu: (1) Documentation: links to descriptive documents. (2) About: display software version number and credits. (3) Debug: display debugging information in a new window.

Account Summary

The account summary display (see FIG. **6** for an example) is a small table that provides a summary of the trader's account status. It preferably shows: (1) Account Balance: the amount of the trader's cash holding in the trader's account. (2) Realized P/L: the amount of profit or loss the trader has incurred with the trader's trading activity to date. (3) Unrealized P/L: the amount of profit or loss that the trader holds with the trader's current open positions. If the trader clears all of his open positions, then this amount would be added to the Realized P/L amount. (4) Margin Used: the amount of the trader's account balance and unrealized P/L tied up for margin purposes.(5) Margin Available: the amount of the trader's account balance and unrealized P/L available as margin for new trading transactions.

This information is preferably continuously updated in real-time to take current market conditions into account. Moreover, the information is always shown in the trader's home currency.

Tables

The table area of the Trading Station shows different information depending on the last Action Button selected. If can include: (1) open trades; (2) open positions; (3) open orders; and (4) transaction history. The default is open positions.

How the information in the table is displayed can be controlled in two ways: (1) Scroll bars are used to scroll the table up or down, allowing a trader to see information that is hidden from view. (2) Sorting can be achieved by clicking on a column header, which causes the table to be sorted so that the column is in increasing or decreasing order. Clicking once sorts the column in increasing order; clicking again sorts it in decreasing order.

For all tables except Transaction history, clicking on a row of the table will cause a pop-up window to appear, offering further actions for that open trade, position, or order.

Open Trades Table

The open trades table (see FIG. 7) shows a list of the trader's currently open trades. The table preferably has 9 columns, described from left to right (not all are depicted in FIG. 7): (1) Short/Long: Indicates whether the position is short or long.

(2) Ticket Number: a number that uniquely identifies an open trade. A trader can use this number as a reference for inquiries to the Trading System or its operators, or to search for particular entries in the transaction history table.

(3) Currency pair: the pair of currencies involved in this trade. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(4) Units: the number of transacted units for this trade, expressed in the base currency.

(5) S/L: the trader's stop-loss for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: the exchange rate obtained when the trade got executed.

(8) Market: the current exchange rate for this currency pair.

(9) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an open trade will cause a pop-up window to appear offering two operations: (1) Close trade. (2) Modify trade. This is used to modify the S/L or the T/P limits.

Open Positions Table

The Open Positions Table (see FIG. 8) displays a list of the trader's open positions. It is similar to the Open Trades table, except that all trades of the same currency pair are aggregated into one line.

The table preferably has 6 columns, described from left to right (not all are shown in FIG. 8):

(1) Short/Long: Indicates whether the position is short or long.

(2) Currency pair: the pair of currencies the position refers to. The first currency of the pair is referred to as the base currency, while the second one is referred to as the quote currency.

(3) Units: the number of units held in this position, expressed in the base currency.

(4) Rate: the average exchange rate obtained for the trades in this position.

(5) Market: the current exchange-rate for this currency pair.

(6) Profit: the unrealized profit (when positive) or loss (when negative) expressed in base currency and on a per unit basis.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column displayed in increasing or decreasing order. Clicking on a row with an open position will cause a pop-up window to appear offering the option to close the position.

Open Orders Table

The Open Orders Table (see FIG. 9) shows a list of the trader's currently open orders. An open order is a request that a particular trade should be made automatically when the exchange rate of the specified currency pair crosses a specified threshold.

The table preferably has 9 columns, described from left to right (not all are shown in FIG. 9).

(1) Short/Long: indicates whether the position is short or long.

(2) Order ID: a number that uniquely identifies the order. A trader can use this number as a reference for inquiries to the Trading System.

(3) Currency pair: the pair of currencies to be traded.

(4) Units: the number of units to be traded, expressed in the base currency.

(5) S/L: the stop-loss for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the S/L value. A stop-loss limit is used to limit the loss a trader may incur with this trade.

(6) T/P: the trader's take-profit for this trade. This trade, once executed, will be closed automatically as soon as the currency exchange rate for this currency pair crosses the T/P value. A take-profit limit is used to realize the trader's profit as soon as it reaches the T/P value.

(7) Rate: specifies that the trade should be executed as soon as the exchange rate for the specified currency pair crosses this value.

(8) Market: the current exchange rate for this currency pair.

(9) Duration: specifies the amount of time an order should stand, until it is automatically canceled.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order. Clicking on a row with an order will cause a pop-up window to appear offering two operations: (1) Cancel order. (2) Modify order. This is used to modify the exchange rate threshold at which the trade is to be executed, or the S/L or T/P limits.

Transactions Table

The Transactions (or Transaction History) Table (see FIG. 10) shows a list of the most recent transactions on the account. For access to a full list of past transactions, a user selects the Information pull-down menu and then selects Transaction History.

The Transaction History Table preferably has 6 columns, described from left to right:

(1) Transaction ID: uniquely identifies the transaction.

(2) Type: identifies the type of transaction.

(3) Currency pair the pair of currencies associated with the transaction.

(4) Units: the number of units to traded in the transaction, expressed in the base currency.

(5) Price: the currency exchange rate applied when buying or selling a currency pair.

(6) Date/Time: the date and time of the transaction.

Clicking on the scroll buttons. will cause the table to scroll up or down. Clicking on a column header will sort the table so that the contents of the column are displayed in increasing or decreasing order.

Currency Rates

The Currency Rates Table (see FIG. 11) shows the current exchange rate for the currency pairs supported by the Trading System. They are preferably updated in real time, approximately every 5 seconds. When there is a significant exchange rate movement for a currency pair, up/down indicators show the direction of the rate change in order to alert a trader, should a trader not currently have the currency pair displayed in the graph.

Clicking on the scroll buttons will cause the table to scroll up or down. Clicking on a currency pair's ask price will pop up a buy window for that currency pair. Clicking on a currency pair's bid price will pop up a sell window for that currency pair.

Graphs

Graphs (see FIG. 12) show how currency exchange rates change over a period of time, ranging from minutes to months. All graphs are updated in real-time, as new currency rates arrive.

At any given time, the difference between the lower boundary and the upper boundary of the curve represents the difference between the bid and the ask price, and the difference may vary over time depending on market conditions. Thus, the top part of the curve indicates the ask price, and the bottom of the curve indicates the bid price.

As a mouse cursor **1220** is moved over the graph, a sub-area **1230** in the graph shows precise exchange rate information for the target currency pair corresponding to the time instance represented by the position of the mouse cursor.

The graph may also display Buy or Sell widgets that indicate at which point in time a trader bought or sold a currency pair. Downward pointing red arrows indicate a sold currency pair (where a trader is hoping the rates will go down after that point), and upward pointing green arrows indicate a bought currency pair (where a trader is hoping the rates will go up after that point).

A trader can adjust what is shown in the graph: (1) The currency pair displayed is selected using the pull-down menu **1240** at the bottom left. (2) The granularity of the graph is selected using the pull-down menu **1250** at the bottom right of the graph. Selecting a fine granularity, such as 5 seconds (where each point on the horizontal axis represents 5 seconds of time), will display a relatively short time interval (less than an hour, in this case). Selecting a larger granularity, such as one day, will display longer-term trends (9 months of exchange rate information in this case).

Scroll buttons **1260** at the top right of the graph area allow a trader to shift the time interval shown to the left or to the right (backward or forward in time). Clicking on the graph with the mouse will hide the Buy/Sell widgets. Clicking again will cause them to reappear.

Buy/Sell Window

A Buy/Sell pop-up window (see FIG. 13) allows a trader to issue buy or sell orders. The window can be caused to pop up either by: (1) clicking on the Buy/Sell action button (see FIG. 5); (2) clicking on the bid or ask price in the Currency Rates Table (see FIG. 11); or (3) clicking on an existing trade, position, or order in the Table area of the Trading Station display (see FIG. 5).

Two types of orders are supported: (1) Market Orders are orders that are transacted immediately based on market exchange rates. (2) Entry Orders are orders that are executed when the exchange rate crosses a certain threshold.

The type of order can be selected by clicking on the appropriate tab in the Buy/Sell Window (see FIG. 13). Market order comes up as the default order.

Issuing a Market Order. To issue a market order with the Buy/Sell Window and the Market Tab selected, a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) If the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph. (B) If the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. (C) If a trade or position was clicked in the Table area, the currency pair corresponding to the trade or position. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) Lower Limit: the order will result in a trade only if a price is obtained that does not lie below this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(5) Upper Limit: the order will result in a trade only if a price is obtained that does not lie above this limit. By default, no limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

The RATE field is set by the Trading Station and corresponds to the most recent exchange rate for the selected currency pair.

To issue the order, a Submit button **1310** must be selected. If the order is successful, and a trade occurs, then an acknowledgment window (see FIG. 14) pops up with a Ticket number that can be used for future reference. Moreover, the Open Trades Table (see FIG. 7) will be updated to reflect the new trade, as will the Open positions-ions Table (see FIG. 8) and the Transaction History Table (see FIG. 10).

Several issues are important to note:

(1) If an order is successful and a trade occurs, then the exchange rate obtained for the trade will correspond to the most current exchange rate maintained at the Trading System servers and not necessarily the rate displayed in the Buy/Sell window.

(2) An order without Lower and Upper Limits will always result in a trade.

(3) An order with both Lower and Upper Limits will result in a trade if and only if the exchange rate for the potential trade lies between the two limits.

Issuing an Entry Order. To issue an entry order with the Buy/Sell Window and the Entry Tab selected (see FIG. **15**), a number of fields must be filled out (although most of the fields are pre-initialized with reasonable values):

(1) ACTION: choose between buy and sell.

(2) CURRENCY: choose the currency pair the trader wishes to buy or sell. By default, this field will be initialized as follows: (A) if the Buy/Sell button was used to obtain the window, the currency pair currently shown in the graph; (B) if the bid or ask price was clicked to obtain the window, the currency pair for which the price was clicked. The pull-down menu can be used to select another currency pair.

(3) UNITS: the number of units of the currency pair the trader wishes to buy or sell, with units expressed in terms of the base currency.

(4) RATE: the order will result in a trade as soon as the exchange rate for the selected currency pair crosses the given value; that is, for buy orders, if the rate goes below this value, and for sell orders if the rate goes above the given value

(5) Duration: this value is used to limit the amount of time an outstanding order will remain effective. By default, the order remains effective indefinitely. However, the duration can be set to the end of the day or for an hour.

(6) Stop Loss: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no stop-loss limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

(7) Take Profit: if the order results in a trade, then the stop-loss value given will be associated with the trade. By default, no take profit limit is selected, but one can be set by checking the check box. If the check box is checked, then the field is automatically initialized with a reasonable value; however, the value can be changed either by modifying the number directly or by using the +/− buttons to increase or decrease the value, respectively.

To issue the order, a Submit button **1510** must be selected. This results in an acknowledgment window (see FIG. **14**) popping up with a Ticket number that can be used for future reference. Moreover, the Open Orders Table (see FIG. **9**) will be updated to reflect the new order. Note that a trader can modify the parameters of an open order (including the rate representing the trade threshold, or the S/L and T/P) by clicking on the order in the Open Orders Table.

What is claimed is:

1. A method of trading currencies over a computer network connecting a trading system server and at least one trading client system, comprising the steps of:

(i) at the trading system server, determining and dynamically maintaining a plurality of current exchange rates, each current exchange rate relating to a pair of currencies and including a first price to buy a first currency of the pair with respect to a second currency of the pair and a second price to sell the first currency of the pair with respect to the second currency of the pair;

(ii) transmitting data from the trading system server to a trading client system, the transmitted data representing at least one current exchange rate at the time of the transmission;

(iii) at the trading client system, displaying the first and second prices for each received current exchange rate to a user;

(iv) at the trading client system, accepting input from the user identifying a pair of currencies the user desires to trade, an amount of at least one currency of the pair desired to be traded and a requested trade price at which it is desired to effect the trade;

(v) transmitting the accepted input from the trading client system to the trading system server;

(vi) at the trading system server, comparing the requested trade price to the respective first price or second price of the corresponding current exchange rate at that time and, if the respective first price or second price of the corresponding current exchange rate at that time is equal to or better than the requested trade price, effecting the trade at the corresponding respective current exchange rate first price or second price and if the corresponding current exchange rate is worse than the requested trade price, refusing the trade; and

(vii) transmitting from the trading system server to the trading client system an indication of whether the trade was refused or transacted and, if transacted, an indication of the price the trade was transacted at.

2. The method of claim **1** wherein the requested trade price is derived from a respective one of the first price or second price of the received current exchange rate and a user input limit value defining a maximum acceptable difference between the respective one of the first price or second price of the received current exchange rate received at the trading client system and the respective one of the first price or second price of the corresponding current exchange rate determined at the trading client system at which the trade can be effected.

3. The method of claim **2** wherein the user can input a first limit value to define a maximum acceptable difference between the first price of the current exchange rate received at the trading client system and the first price of the corresponding current exchange rate determined at the trading client system and can input a second limit value to define a maximum acceptable difference between the second price of the current exchange rate received at the trading system and the second price of the corresponding current exchange rate determined at the trading client system and the requested trade price is derived from the first price or second price of the current exchange rate received at the trading client system and the corresponding one of the first limit value and second limit value.

4. The method of claim **2** wherein step (iv) comprises the steps of:

(a) the user selecting one of the first price and second price of the current exchange rate displayed at the trading client system;

(b) displaying to the user a set of input fields to define a desired trade, the input fields including an identification of the pair of currencies the user desires to trade, the amount of the currencies desired to be traded, the selected first price or second price of the current exchange rate received at the trading client system and a limit value, and where the input fields to identify the pair of currencies and the first price or second price are populated with appropriate values determined from the user's selection of the one of the first price or second price;

(c) receiving from the user input to the input field defining the desired amount of currency to be traded; and

(d) determining the requested trade price from the selected one of the first price and second price and the limit value.

5. The method of claim 4 wherein in step (b) the displayed set of input fields includes: a first limit value to define a maximum acceptable difference between the first price of the current exchange rate received at the trading client system and the first price of the corresponding current exchange rate determined at the trading client system; and a second limit value to define a maximum acceptable difference between the second price of the current exchange rate received at the trading system and the second price of the corresponding current exchange rate determined at the trading client system and in step (d) the requested trade price is derived from the selected first price or second price and the corresponding one of the first limit value and second limit value.

6. The method of claim 2 wherein, when the limit value is zero, the requested trade price is the current corresponding first price or second price of the current exchange rate at the trading server.

7. A method of trading currencies over a computer network connecting a trading system server and at least one trading client system, comprising the steps of:

(i) at the trading system server, determining and dynamically maintaining a plurality of current exchange rates, each current exchange rate relating to a pair of currencies and including a first price to buy a first currency of the pair with respect to a second currency of the pair and a second price to sell the first currency of the pair with respect to the second currency of the pair;

(ii) transmitting data from the trading system server to a trading client system, the transmitted data representing at least one current exchange rate at the time of the transmission;

(iii) receiving at the trading system server input from a user of the trading client system identifying a pair of currencies the user desires to trade, an amount of at least one currency of the pair desired to be traded and a requested trade price at which it is desired to effect the trade;

(iv) at the trading system server, comparing the requested trade price to the respective first price or second price of the corresponding current exchange rate at that time and, if the respective first price or second price of the corresponding current exchange rate at that time is equal to or better than the requested trade price, effecting the trade at the corresponding respective current exchange rate first price or second price and if the corresponding current exchange rate is worse than the requested trade price, refusing the trade; and

(v) transmitting from the trading system server to the trading client system an indication of whether the trade was refused or transacted and, if transacted, an indication of the price the trade was transacted at.

* * * * *