STOCK EXCHANGE TRADING SYSTEM

The present invention relates to a stock exchange trading system, as well as to a method for operating a stock exchange trading system.

For years now, stock market trading has been carried out almost exclusively in a computerised manner. Thereby, the traders sit at a computer terminal, and this terminal is connected to a computer-implemented trading platform via a computer network. Such a trading platform as its key component comprises at least one so-called "matcher" (trading platform data processing device, i.e. an order accounting and implementation unit), in which orders are stored in an order book or are carried out. The matcher is connected on the one hand - generally via at least one switch - to the traders, and on the other hand to a backoffice (order post-processing system), in which the execution of the order is completed and the respective financial transaction is processed for example.

If a trader is interested in a stock trade, then he places an order at the trading platform. This order comprises the title concerned, information as to whether the title should be bought or sold (buy/sell), the desired quantity as well as the price. The trading platform manages the order book. An incoming order is examined as to whether there is a matching, stored order. If not, then the order is stored in the order book and acknowledgment of receipt is given to the ordering party (client). If yes, then the trade is executed, registered in the sales ledger and both parties are informed. Information concerning the accomplished trade, in an anonymous form (i.e. without information as to the identity of two involved traders), then also goes to all traders (broadcast). Of course, it can also be the case that an order is only able to be partly executed and then a residual order remains in the order book after carrying out the trade; whenever an execution of an order is mentioned in this text, this is also to include such a partial execution of the order. Any other order variants (amongst other things: market orders, accept orders, fill-or-kill orders, iceberg orders) are likewise assisted, apart from the mentioned order type with the attributes buy/sell, quantity and price.

Three types of communication primarily result from this communication pattern for electronic stock exchange trading:

a) trader -> trading platform (point -> point)

b) trading platform -> trader (point - > point)

c) trading platform -> all traders (broadcast)

A network for rapid electronic trading should assist these communication patterns. It is particularly with the communication a) (trader -> trading system) that it is a question of fairness. For this reason, it should not be possible for one of the traders to be able manipulate this procedure and gain advantages on account of this. Moreover, those traders who obtain information on a price change first of all are, at an advantage in such a system. Fairness should also be ensured here. This is not possible with the application of traditional networks – even in the field of super-computers - without special complicated measures. Moreover, one should also ensure that the system cannot become overstretched in such networks. High-availability communication systems should also be available.

Added to this is the fact that the decision making on the part of the trader is effected with extensive computer assistance; in some cases it is even a computer program which makes the effective decision on the part of the trader (of course in accordance with the instructions of the trader). Here, one speaks of algotrading.

The network speed, the processing speed in the trading system and the processing speed in the trader system are crucially important with this application. Many endeavours have therefore been made in the stock trading business, in order to improve these various speeds.

Standard networks from the world of supercomputers have already been suggested for certain applications, in order to improve the speed of communication networks, due to the fact that such standard networks are counted as belonging to the quickest networks worldwide.

Such supercomputer networks on the one hand are optimised for point-to-point communication and on the other hand for multicast / broadcast. Point-to-point communication is particularly important to supercomputers, since in many applications, many "equally authorised" computers operate on a single task, and a communication with an "arbitrary" (set by the task) partner is necessary. A message is sent to a group which is defined in advance, in the case of a multicast or broadcast. Thereby, in the supercomputer world, it is assumed that all partners receive the message, and if a single computer cannot receive the message, then as a rule the computation usually has to be stopped, since it no longer makes sense to compute further given the absence of individual computers.

It is evident that with the application "stock exchange", the information flows in a "tree-like" manner: from the traders to the trading platform and back again, and with the broadcast, from the trading platform to all traders.

A network from the application "supercomputer" therefore has characteristics other than those required in the application "stock exchange". For example, in the case of a stock [exchange] trading application, it would be disruptive if, as is envisaged with supercomputer networks, another trader computer is inundated with information due to an error or a wanted action, to the extent that it can no longer operate properly. In the case of a broadcast, the complete system can come to a standstill and stop, if an individual one of the trader computers does not accept the broadcast notification, be it due to a technical problem or deliberately.

It would therefore be advantageous if the disadvantages of a supercomputer network could be eliminated, without thereby having to forego the advantage of the high speed.

Moreover, it can be advantageous if a network for this application of "stock exchange" were to be scalable, and almost any number of trader computers were able to be connected. Thereby, it can also make sense to apply several trading platform data processing devices (matchers) for reasons of performance.

Yet a further problem with the application "stock exchange" is the re-synchronisation of an individual trader data processing device after a local interruption, e.g. due to a technical problem. According to the state of the art, this synchronisation is effected by way of an independent second channel, by way of the trader data processing device being supplied by this channel with information on the already effected trading activities, until it is at the same "level of knowledge" as the other trader data processing devices. With high-performance systems, it can occur that the re-synchronisation requires hours or is even no longer possible or that the trade even has to be interrupted for a certain time, in order to permit a complete synchronisation, due to these high quantities of information.

An important complex of issues relates to those concerning the processing of orders. According to the state of the art, the traders are connected to the matcher via a network switch (switch), and common protocols, in particular TCP/IP protocols are applied. Common switches have buffers (buffer memories) for incoming messages, and these are processed and transferred according to a defined sequence. Thus on the one hand it is the arrival time and on the other hand it is also the port at which the related message arrives and the condition, in which the switch currently is, which are relevant concerning the question as to which of two incoming messages are transferred first of all. It is due to this situation that a trader can inadvertently or even deliberately exert an influence on the processing of the orders on another trader, amongst other things by way of very many messages being sent within a short time and the hardware of the switch thus being brought to the limits of its capacity.

A further challenge is the intelligent receiving of information on the part of the trader. With a high-performance system, there can be times, at which the trader data processing device is no longer capable of processing and sorting the flood of information inundating it.

Finally, most of these systems are not designed as highly-availability systems. This characteristic can only be realised at a later stage at a very high expense/effort.

It is the object of the present invention, to at least partly eliminate the disadvantages outlined above, and to provide a system and a method which meet the demands concerning speeds, fairness, functional ability, and this being the case during and after the failure of components.

According to a **first** aspect of the invention, a stock exchange trading system is provided which comprises:

- a trading platform and
- a plurality of trader data processing devices;
- wherein each trader data processing device is capable of submitting orders in the form of electronic messages to the trading platform, and the trading platform is capable of storing these orders in an order book, or, given a matching order already stored in the order book, of at least partly carrying them out;
- wherein each trader data processing device comprises a time generator and is configured to provide each order with a timestamp which represents the point in time of the submission of the order;
- and wherein the system is configured such that the trading platform processes incoming orders strictly in a sequence corresponding to the timestamp.

A "trader data processing device" can be an integrated computer. However, it does not need to be a unit in the context of a "personal computer". It can also be divided onto several units which are spatially separated as the case may be, wherein the same applies to other data processing devices which are described in this application.

Thus one departs from the previous approach of working through the orders according to their arrival (first come, first served) and changes to a time-stamping system. This means that an order which is received at later stage under certain circumstances is executed before an order which had already arrived beforehand. This, under certain circumstances, also entails waiting for a certain time interval after the arrival of an order and before its implementation.

The trading system can be constructed in a tree-like hierarchical manner with several levels, and this applies to all aspects of the invention, wherein the network nodes of the levels are

formed in each case by switches, each with at least one uplink connection and several downlink connections.

According to an embodiment, at least one trader data processing device comprises a network interface (in particular a network card), via which the trader data processing device is connected to the trading platform, and the time generator is integrated into the network interface.

In particular, one can envisage the time generator of all trader data processing devices being identical and being provided by the system operator. For example, one can envisage each trader obtaining a network interface which is provided by the system operator and which is identical for all traders, and only being able to communicate with the trading platform via this.

The trading platform as is known per se comprises a matcher as its key component, i.e. a trading platform data processing device which is configured to manage an order book and to execute the function of the matching (assignment), i.e. bringing together offers and requests, in accordance with predefined rules (matching rules).

In embodiments, the trading system comprises at least one network switch (switch), and the trader data processing devices are connected to the matcher via this switch. The switch can serve as an aggregation switch in the case of a network architecture which results for the application described here.

In this text, a switch is generally to indicate a network node which permits a selective transfer to certain (individual or several, under certain circumstances selectively also to all) connected devices - in the present context, the switch at the minimum permits an incoming order from a trader data processing device to be led further onto one or more trading platform data processing device(s) and not for example also to all other trader data processing devices. With regard to the context used here, a switch is a device ensuring this functionality, independently of whether all elements belonging thereto are arranged physically close to one another (for example in a common housing) or not.

The at least one switch can thereby be considered as part of the trading platform, which does not mean that it is physically present at the location of the matcher. At least if the switch(es) is/are considered at part of the trading platform, the capability of incoming orders being processed by the trading platform strictly in accordance with the sequence specified by the timestamp, is a characteristic of the trading platform itself, i.e. the trading platform is capable of processing the incoming orders strictly in the sequence of the timestamp.

Now one can particularly envisage the sorting of the orders (i.e. the determining of the sequence by way of the timestamps) being effected in this switch, which then also comprises a suitably configured buffer. The switch then always firstly relays that order with the oldest time. The orders are then worked off in the matcher, according to the sequence defined by the timestamps.

Alternatively or supplementary to this, the sorting can also be effected and/or confirmed in a prioritiser which is part of the trading platform data processing device or is serially connected in front of this (for example in a direct manner).

In redundant embodiments according to the third aspect which will hereinafter be dealt with in more detail, such a prioritiser can moreover suppress the second of two identical incoming messages.

From now, it is possible to ensure a fair execution of orders which can be reproduced at all times, in particular independently of network conditions and coincidences, due to the amazingly simple measure according to the invention (provision of the orders with a timestamp). This in turn permits the system to be designed in such a redundant manner, that it is accordingly available without interruption, even if a matcher or a switch fails.

In the system according to the first aspect of the invention, the time generator of the trader data processing devices has an important function. There are various possibilities concerning measures which ensure that all trader data processing devices are with the same time:

- identical time generator hardware provided by the system operator;

- the (for example identical for all) time generator can have a free oscillator or run via the network cycle;

- in both cases, the time generator can be repeatedly (for example periodically) synchronised via the trading platform or via a trustworthy external source (for example GPS);

- the network cycle can be set by the trading platform if the time generator runs via the network cycle.

According to a **second** aspect of the invention which can be combined with the first aspect, a stock exchange trading system is provided which comprises:

- a trading platform; and

- a plurality of trader data processing devices;
- wherein each trader data processing device is capable of submitting orders to the trading platform in the form of electronic messages, and the trading platform is capable of storing these orders in an order book, or, given a matching order already stored in the order book, of at least partly carrying it out, and of acknowledging the storage or the at least partial execution, to the trader data processing device;
- wherein each trader data processing device is configured to monitor the number of unacknowledged messages submitted to the trading platform and is configured such that a predefined maximal number of such unacknowledged electronic messages is not exceed at any time.

Unacknowledged messages are messages for which no acknowledgement of receipt has yet been received.

The maximal number corresponds to a message credit, which is available to each trader data processing device. The number of messages which are underway (i.e. sent, but not yet acknowledged) may not exceed a maximal number. By way of this, one prevents a trader data processing device inadvertently or deliberately inundating the trading system with messages and thus blocking electronic data channels, which would create unequal conditions for the various participants.

Moreover, one can envisage the size of at least one message buffer of the trading system being matched to the maximal number, in a manner such that the buffer cannot overflow.

Such a buffer of the trading system, in particular can be a buffer which according to the first aspect of the invention is applied for the order sorting.

Such a buffer of the trading system for example can be a buffer of a switch. Thereby, it can be a dedicated (i.e. reserved for the downward link to the trader) buffer, whose size then corresponds at least to that of the maximal number of messages. Such a buffer can alternatively also be a central buffer for several traders (i.e. either a central buffer of a switch or a buffer of a switch at a higher network level, for its part connected to several switches).

In embodiments, in which a sorting of arrived messages, in particular according to the first aspect, takes place in a switch, then the preferred criterion "size of the buffer is matched to the maximal number in a manner such that the buffer cannot overflow" applies at least to the buffer/buffers of the switch, in which the/a sorting takes place.

For example, one can envisage sufficient buffers being present in each network node in the case of a tree-like network structure, so as to intermediately store all unanswered messages of the trader data processing devices assigned to the network nodes – with the switch of the lowermost level according to the trader data processing devices connected to the switch, with a switch of the second level according to trader data processing devices connected via a switch of the lowermost level (or possibly directly) to the switch of the second level, etc..

According to the **third** aspect of the invention which in particular can be combined with the first and/or second aspect, the trading platform is designed redundantly, i.e. all components are present in duplicate, so that in principle (at least) two trading platforms result.

Redundancy per se is also known with stock exchange trading systems per se. According to the state of the art, a corresponding, equivalent component is provided in the case of the failure of a system-relevant component – for example of the matcher. If need be, this is used instead of the failed component, wherein switch-on and synchronisation procedures are necessary for this. The same applies to the transition back to the standard component, when this is capable of functioning again. A time, during which the system does not function, results in both cases, and the fair treatment of orders in the case of such a failure at a location in the system which is not known beforehand, is also quite a challenge.

According to the third aspect of the invention, the two trading platforms are constructed identically and applied on equal terms during normal operation. All transactions are thus implemented on both trading platforms. The trading platforms are therefore configured such that with simultaneous operation, it is ensured at all times, that both matchers (or with a plurality of matchers per trading platform, the matchers corresponding to one another) have identical order books at all times. This for example can be particularly ensured, by way of the orders being sorted on the basis of the timestamp, according to the first aspect of the invention, and/or, as the case may be, a credit system being applied, according to the second aspect. Inasmuch as random components play a part (for example with exactly identical timestamps in two orders), the pseudo random number generators of the two trading platforms are designed identically and operate with the same starting values.

The switch or the system of switches which is used for connection to the trader data processing devices is preferably also present in a redundant manner and thus at least in duplicate. The same of course also applies to the connections between the switches and connections from these to the trading platform data processing device or trader data processing device. One can envisage the switches of the two parallel systems not being connected to one another, but only the switches of the lowermost level being connected to the trader data processing devices and possibly the switch of the uppermost level being connected to both trading platform data

processing devices, in the case of a system of switches with several levels. Each switch of a lower hierarchy level in each case is then only connected to one switch of the hierarchy level lying above this, and not for example also to the corresponding switch of the second parallel system, although this is not to be completely ruled out.

The switches of the uppermost level (the central switches; general in each case one per system) are connected to both trading platform data processing devices (or if the tasks of the matcher are distributed in a segmented manner onto several matchers, are connected in each case to both trading platform data processing devices of each segment). Alternatively, one can envisage each matcher, on arrival of an order, transferring this order to its counterpart (i.e. the other matcher). In both cases therefore, each order arrives twice at each assigned matcher during normal operation of the trading system without any failure of network components. Optionally, one can yet provide a delay unit in each case, wherein this slightly delays the message arriving via that system of switches, from which a quicker message transmission is expected, due to it being more direct.

A prioritiser – for example of the above described type, or also without the function of the sorting according to timestamp - can also be present, and this, with embodiments which also correspond to the first aspect, carries out a sorting according to the timestamp (inasmuch as this has not already been effected in a switch) and which in particular suppresses identical messages.

With embodiments of the third aspect, a unit which further processes the first valid message (for example transfers this) and suppresses identical copies can (also) be present on the part of each trader data processing device.

With trading platform data processing devices or with the segmentation into several trading platform data processing devices with different competences, one can optionally envisage the two trading platform data processing devices corresponding to one another having a direct "private" connection, via which they can also be synchronised with one another.

The invention also relates to a method for operating an electronic trading system according to the first, second and/or third aspect of the invention.

According to all aspects of the invention, one can envisage the trader data processing device keeping copies of the parts of the order book which are of interest to the respective traders. The fourth aspect of the invention which is described hereinafter is also of particular interest with regard to this context.

According to a **fourth aspect**, a method is suggested, in order to synchronise a unit (a trader or matcher for example) with other units of the system, for example in the case that this unit has failed or has been temporarily separated from the system, due to a malfunction. It is necessary for certain condition data - for example the order book, if this is open - to be available to all participants, for a fair stock exchange trading operation, and it is only the respectively informed participant who can participate in the markets.

According to this fourth aspect, condition data is continuously sent in a packet-wise manner to all participants (broadcast mode) during operation, wherein each packet contains a part of the condition data. This is carried out intermittently with the sending of current messages, which is to say up-to-date messages (orders, confirmations, information on orders/transactions, to the market participants), until all data elements have been distributed, whereupon condition data is sent afresh.

"Intermittently" in the context of the fourth aspect of the invention does not necessarily means that condition data and current messages are always sent alternately. In contrast, if for example the system is not used to capacity, then several condition data packets can be sent successively, before a current message is again sent, or possibly also vice versa.

The condition data according to definition is current and correct at the point in time of the sending. A data element for example can comprise the number of offered papers of a certain title at a certain price. If now a message with regard to a data element arrives, this message is processed and the data element updated accordingly, if the data element concerned has already been synchronised, i.e. if for this data element a condition data packet has already been sent. If however a synchronisation has not yet been effected since the last failure of the unit concerned, and the respective data element does not yet comprise data, then the message is ignored.

The method according to the fourth aspect can be used, in order to synchronise the trader data processing devices and/or matchers (in a redundant system). In particular, the application solely for trader data processing devices in combination with another method for the matchers is also an option, apart from the favourable application to both.

In embodiments of the first, second, third and/or fourth aspect, the matcher can be configured as a dedicated logic circuit. A dedicated logic circuit is a hardware component or a group of hardware components, in which the logic functions, by way of which signals or data are processed, are implemented in a predefined manner by way of given circuits which are present in the hardware itself. This is in contrast to main processors of conventional computers (including server-computers) which are designed as "generic", universally useable microprocessors which

work off a sequence of commands which are provided in a memory (as software), wherein it is only this software and not the circuit itself which is for a specific purpose.

Examples of dedicated logic circuits are field programmable gate arrays (FPGAs) (dedicated by the configuration, i.e. designed in an applications-specific manner) and ASICs, whose switching logic is set on manufacture and can no longer be changed.

In the embodiments of the first, second, third and/or fourth aspect, one can likewise envisage all authorisations of the trader being examined in real-time for each order, before executing the order. Alternatively or supplementarily, one can envisage the relevant limits for each trading participant (e.g. the total value of all orders etc. which are present in the book) being tracked and examined in real time before the execution of the order. These two measures which can be realised in each case individually or in combination serve for a possible self-regulation of the markets.

Embodiment examples and principles of the invention are hereinafter explained by way of the Figures. There are shown in:

Fig. 1 a schematic overview of a trading system;

Fig. 2 a trading system with several matchers;

Fig. 3 in a schematic manner, in inner circuitry of a switch S as well as elements of traders and of a trading platform;

Fig. 4 a redundant trading system;

Fig. 5 a redundant trading system with several matchers; and

Fig. 6 the synchronisation after the loss of a participant (for example a trader) after a failure.

Network architecture which is favourable for the rapid communications for stock exchange systems is represented in **Figure 1**. It is thereby the case of a hierarchical network which is optimally adapted to the communication requirements in stock exchange applications. M indicates the matcher. $S_{0,0}$ is the central switch, via which the traders (here indicated as clients C_0, ... C_n) communicate with the matcher. An optional further switch level with decentralised switches $S_{1,0}$, ... $S_{1,m}$ is moreover drawn in the represented embodiment example, wherein at least one client (generally several clients) are connected to each of these decentralised switches. An

expansion to yet further levels and accordingly yet further architecture branching in a tree-like manner is also possible. The approach with the decentralised switches permits a good scalability and may be appropriate when the central switch does not physically have enough interfaces to all traders. It can be advantageous if each trader is connected to the central switch via the same number of levels, since the transfer times are then roughly identical and, with the approach with a sorting according to a timestamp and which is discussed above, the waiting time before a message (for example order) is transferred upwards can be kept shorter.

The switching logic of the at least one switch can optionally be such that it only permits messages from a trader data processing device to the trading platform data processing device, from the trading platform data processing device to a certain trader data processing device and from the trading platform data processing device to all trader data processing devices - which means that the communication pattern resulting in the electronic stock exchange trade is implemented in the switching logic of the network at the very beginning.

A division of the function of the matcher onto several matchers is also possible, and this is represented symbolically in **Figure 2**. Each matcher is then responsible for one or more titles. The transfer to the correct matcher can be effected by the uppermost switch, for example on account of the target address.

The uppermost switch (i.e. the uppermost network node) can moreover optionally have a further output which stores all relevant messages which are transferred by this switch, in an archive A and thus makes this information available for example to the backoffice, for further processing. This option exists independently of whether only one matcher or, as in Figure 2, several matchers are applied. However, this option is particularly favourable with the application of several matchers, since it permits a central archiving. Alternatively, each matcher can also have an individual archive.

The inner circuitry of a switch S is represented symbolically in **Fig. 3**. The construction which is represented in Fig. 3 for example is applied in particular in the switch of the uppermost level, i.e. the uppermost network node. Switches of the levels lying therebelow - inasmuch as they are present - can optionally also be designed accordingly, wherein at least the broadcast function should also be implemented onto the levels possibly lying therebelow.

In Fig. 3, components generally belonging to so-called uplinks (in the direction of the matcher) have a "U" in the reference, and components belonging to downlinks (in the direction of the trader) have a "D".

A message (data packet; in particular with an order) is transmitted from the trader C_0 ... C_n via a transmitter Tx (a transmitter component). Thereby, a conversion for example into a serial data format takes place. This is received in the switch S and again converted in parallel ($KDRx_0$ ... $KDRx_n$) and written into the buffer (UB_0 ... UB_n). Thereby, a dedicated buffer is assigned to each first receiver ($KDRx_0$ ... $KDRx_n$; i.e. to each downlink), in the drawn embodiment example. It would also be possible to provide a central buffering in alternative embodiments.

The first arbiter UA decides on which packet is transferred further and controls the multiplexer UM accordingly. The data is now converted by the first switch transmitter KUTx again into a serial protocol, transmitted and converted again into parallel format by a matcher receiver MRx. The message reaches the matcher M via an optional matcher buffer MB.

If now the matcher M sends a message (for example the acknowledgment of receipt of an order) to a specific trader Co... Cn, then this message is converted via a matcher transmitter MTx again into a serial format, transmitted and received by the second switch receiver KURx and converted again into a parallel format. The second arbiter DA, by way of the address, now decides which trader C_0 ... C_n is to obtain the message, and closes the respective switch DD_0, DD_1 ... or DD_n. The message is subsequently transmitted via a second switch transmitter $KTDx_0$ ... $KDTx_n$ to the respective trader-receiver Rx_0 ... Rx_n and is now available to the trader C_n for further processing.

Apart from the specific communication between the matcher and a specific trader C_0 ... C_n and which has been described above, there is yet the broadcast case: matcher to all traders C_0 ... C_n. This case differs from the specific case in that the second arbiter DA recognises that it is the case of a broadcast message and now closes all switches DD_0...DD_n, so that finally the data packets are transmitted to all traders C_0 ... C_n.

According to the first aspect, each trader is provided with a time generator, here in the form of a precisely synchronised clock CLK_0 ... CLK_n. The time reference for this clock can thereby be set by the matcher or by the uppermost network node (the uppermost switch), by way of this periodically sending out reference time messages, which then reset the local clocks CLK_0 ... CLK_n. The actual clock cycle can thereby be produced by a local oscillator or also by the global network cycle which in turn is set by the matcher M. According to the first aspect, each message from the client C_0 ... C_n is now provided with a timestamp, and the network system ensures that older messages always arrive at the matcher first of all; the network therefore "sorts" the messages on account of the time. This function here is effected by way of the first arbiter UA transferring the older message first of all. The arbiter must await at least the time (waiting time) which arises due to unequal construction manners (e.g. cable lengths, component tolerances), so

that this selection take its course fairly. Only then can the arbiter make its decision. The messages are therefore ordered strictly according to time, on each uplink or at least on the uplink of the uppermost network node, and with this arrive at the trading system M in the correct temporal sequence (also via several switch levels).

Even with such architecture, it may still occur that individual traders C_0 ...C_n send many messages, and a switch S thus become overloaded. This would be manifested in a buffer overflow in one of the buffers UB_0 ... UB_n. A credit system can be introduced according to the second aspect, in order to prevent this: the trading system allots credits to all traders C_0 ... C_n. A credit permits the trader to send an order (or more generally: a message). The number of credits is so large that an individual trader can still operate at full speed in the case of a non-loaded system and that the individual buffers do not overflow even with a highly loaded system. A credit is deducted per message and is only credited when an acknowledgment of receipt has been sent from the matcher to the trader.

A system which is redundant according to the third aspect of the invention is shown in **Fig. 4**. The principle of duplication is illustrated here by way of an embodiment example, with which the matcher is responsible for all titles.

The matcher M is duplicated to M*, as well as the network with the switches Si,j with Si,j*, in Figure 4. One of the two matchers M or M* can thereby set the time (time master, i.e. the network cycle as well as the reference time), and the other matcher then adopts this time. In the case of a failure of one matcher, the other becomes the time master, if it was not already hitherto the time master. Both matchers are constructed in a completely identical manner with regard to the actual processing of orders, and the accounting and execution of the orders take place in both independently of one another.

The received message flows are transferred to the respective other trading system (PS(t) and PS*(t)), in order to ensure that both trading systems are in possession of all information, even with a partial failure of the network. So that their individual message flows are not favoured by the transmission, these are delayed by a delay time TD and TD*. The prioritisers P and P* now ensure that messages with earlier timestamps are taken first of all. Identical messages (which arrive via the two independent networks) are moreover suppressed in the prioritiser P. One can therefore guarantee that identical, temporarily ordered message flows are transmitted to the actual trading systems M and M*.

For reasons of redundancy, in each case two clocks CLK; CLK* can also be present on the part of the trader data processing devices C_0 ... C_n. For example, both matchers can continuously inform over the course of time, and the information of the one matcher M serves

for the synchronisation of the one clock CLK, whilst the other matcher M* synchronises the other clock CLK*. However, only the one time basis is used for the timestamp of the message sent twice, so that the timestamp is completely identical on both channels. For example, one can envisage a certain one of the clocks being responsible for the timestamps, as long as it functions, or one can ascertain as to which is more reliable via suitable algorithms, etc..

The implementation into a system as is represented in Fig. 2, with matchers M_1, ... M_m with different competences is possible without further ado and is represented in **Fig 5**. At least one matcher, preferably all matchers is/are present in duplicate and each are connected to both uppermost network nodes $S_{0,0}$, $S_{0,0}^*$. Inasmuch as an archive A is connected to the uppermost network node, this can also be designed in a redundant manner, i.e. be present in duplicate, as is represented in Fig. 5 by way of the archive A'.

As in the case of Fig. 4 and as corresponds to an often advantageous procedure, in the embodiment according to Fig. 5, the switches S_0, S and S_0', S' of both systems of switches are not connected to one another. The signals run independently of one another in both systems (and in duplicate as described above), but come together in the respective matchers.

The connections C1 ... Cm in Fig. 5 indicate optional "private" connections between the matchers which correspond to one another and which permit a direct data exchange bypassing the network. As an alternative or complementarily to the method according to the fourth aspect, which will be described in more detail hereinafter, they can be used for example for the synchronisation of the matchers corresponding to one another.

A further challenge is the resynchronisation of a trader C_0 ... C_n or of a matcher in the case that such a unit is also to participate again in the current trading affairs after an interruption. A lot of data which is based on the already executed transactions is produced by the complete trading system in normal operation. In the case of a restart, this places very high demands upon all participants, due to the fact that they need to synchronise again. This, as a rule was effected in the past by a "reference picture" of the current status and a tracking of all transactions since the reference picture. However, in this manner it can last a very long time until the unit is synchronised again, if the system is operated at the very limits of its capacity and no longer has any capacity for the tracking.

According to the fourth aspect of the invention, the synchronisation is effected in steps. The functioning manner is explained in **Fig. 6** by way of a simple linear array; more complex data structures can mostly be put together by way of several such arrays.

The array as is represented in Fig. 6 for example can represent the order book for a certain title. The respective orders with the details specifications for instructions, for example sorted according to their temporal sequence are contained in this. The order book in these embodiments consists of different central data structures of a fixed size, in each case with clearly defined data elements for the respective data structure.

The fields which have a grey background in Figure 6 show entries of the array which have already been synchronised afresh after a failure, and those fields with a white background are not yet synchronised. According to the fourth aspect, entries - of the order book and/or of other relevant elements - are continuously synchronised during normal operation, and specifically independently of whether a unit requires a new synchronisation at all, or not. The synchronisation in particular is effected by the matcher or, given a redundant system, inasmuch as the fault does originate from one of the two matchers, by both matchers. The synchronisation follows linearly or logically linearly along the data structure, as is represented.

At least one individual entry can be synchronised for example after each transaction in the trading system (point in time t). If the trading system is not used to capacity, then several entries can also be effected. If now a new transaction (for example a change of an existing order, due to a part having been implemented by a purchase or sale) is effected on an entry which was already renewed (point in time t + 1 in Fig. 6), then the respective entry is directly replaced. If an entry if effected on a part which has not yet been renewed (point in time t + 2) then this is simply ignored. The resynchronisation of the unit which is newly adopted or readopted into the trading system is successively advanced in this manner, whilst the units which are not affected are in no way compromised. The data distribution begins afresh again when all data elements of the trading system have been distributed.

The trading system does not need to be "bothered" at all as to whether of the trader data processing devices C_0 ... C_n or partner trading systems are presently re-synchronising thanks to this mechanism; the trading system simply does its task and arbitrary subsystems can autonomously resynchronise again. A large number of traders can simultaneously resynchronise in an autonomous manner thanks to this procedural manner.

PATENT CLAIMS

1. A computerised stock exchange trading system comprising:
   - a trading platform and
   - a plurality of trader data processing devices;
   - wherein each trader data processing device is capable of submitting orders in the form of electronic messages, to the trading platform, and the trading platform is capable of processing these orders, by way them being stored in an order book, or, given a matching order already stored in the order book, by way of them at least partly being carried out;
   - wherein each trader data processing device comprises a time generator and is configured to provide each order with a timestamp which represents the point in time of the submission of the order;
   - and wherein the trading system is configured such that incoming orders are processed by the trading platform strictly in a sequence corresponding to the timestamp,

   wherein each trader data processing device is configured to monitor the number of unacknowledged messages submitted to the trading platform and is configured such that a predefined maximal number of such unacknowledged electronic messages is not exceeded at any time.

2. A trading system according to claim 1, wherein at least one trader data processing device comprises a network card, via which the trader data processing device is connected to the trading platform, and wherein the time generator is integrated into the network card.

3. A trading system according to claim 1 or 2, wherein the time generators of all trader data processing devices are identical.

4. A trading system according to one of the preceding claims, wherein the time generators of the trader data processing devices run via a network cycle defined by the trading platform.

5. A trading system according to one of the preceding claims, wherein the time generators are continuously repeatedly synchronisable during operation, by way of the trading platform or a trustworthy external source.

6. A trading system according to one of the preceding claims, comprising at least one switch, wherein the switch is connected to several of the trader data processing devices,

comprises a buffer and is configured to store the orders produced by the trader data processing devices, in the buffer and to transfer then in a manner sorted according to the timestamp, so that the order with the oldest time is transferred first of all.

7. A trading system according to claim 6, wherein the switch waits for a certain time before transfer of an order, in order to compensate possible running time differences of different orders.

8. A trading system according to claim 6 or 7, comprising several levels of switches which are linked to one another in a tree-like manner, wherein each switch comprises a buffer and is configured to store orders in the buffer and to transfer them in a manner sorted according to the timestamp.

9. A trading system according to one of the preceding claims, comprising a second trading platform which is capable of processing orders by way of them being stored in a second order book or, given a matching order already stored in the second order book, by way of them at least partly being carried out, wherein the trading system is configured such that all orders in the trading platform and in the second trading platform are processed independently of one another.

10. A trading system according to one of the preceding claims, wherein the trading platform or each trading platform comprises a prioritiser which carries out a sorting according to the timestamp and/or suppresses identical messages arriving in multiple.

11. A trading system according to claim 1, wherein a sorting according to the sequence corresponding to the timestamp is carried out in a unit with a buffer, and wherein the size of the buffer is selected such that it can accommodate the maximum number of orders.

12. A trading system according to claim 1 or 11, wherein the trading platform is structured in a tree-like manner with network nodes, and wherein sufficient buffer memory is present in each network node, in order to intermediately store all unanswered messages of the traders data processing devices assigned to the network node.

13. A trading system according to one of the preceding claims, additionally comprising a second trading platform, wherein each trader data processing device is connected to the first as well as to the second trading platform, so that each order gets to the first as well as to the second trading platform, wherein the first trading platform is capable of storing these orders in a first order book or, given a matching order already stored in the first

order book, of at least partly carrying them out, wherein the second trading platform is capable of storing these orders in a second order book or, given a matching order already stored in the second order book, of at least partly carrying them out, and wherein the system is configured such that orders going into the first trading platform as well as into the second trading platform are processed strictly in a sequence corresponding to the timestamp, so that the first order book and the second order book, with regard to the contents are equal at all times.

14. A method for operating a trading system for stock exchange trading, in particular according to one of the preceding claims, wherein an order is generated by a trader data processing device, is sent to a trading platform and is processed by this, by way of it being stored in an order book or, given a matching order already stored in the order book, by way of it being at least partly carried out, characterised in that the order is provided by the trader data processing device with a timestamp containing real-time information and that orders going into the trading platform are worked off in a manner sorted according to the timestamp
wherein each trader data processing device monitors the number of messages submitted to the trading platform and is configured such that a predefined maximal number of such unacknowledged electronic messages is not exceeded at any time.

15. A computerised stock exchange trading system comprising
- a trading platform; and
- a plurality of trader data processing devices;
- wherein each trader data processing device is capable of submitting orders in the form of electronic messages, to the trading platform, and the trading platform is capable of processing these orders, by way of them being stored in an order book, or, given a matching order already stored in the order book, by way of them being at least partly carried out;
- wherein each trader data processing device is configured to monitor the number of unacknowledged messages submitted to the trading platform and is configured such that a predefined maximal number of such unacknowledged electronic messages is not exceeded at any time.

16. A trading system according to claim 15, wherein the trading platform comprises a switch with at least one buffer, and wherein the size of the buffer or buffers is selected such that the switch can store at least the maximal number of orders.

17. A method for operating a trading platform, in particular according to claim 15 or 16, wherein a plurality of orders is generated by a trader data processing device, is sent to a

trading platform and is processed by this, by way of the said plurality of orders being stored in an order book or, given a matching order already stored in the order book, by way of at least partly carrying it out, characterised in that a predefined credit balance is given to the

trader data processing device, that a credit is deducted on sending each order and is credited again on receiving an acknowledgement of receipt of the order, and that the trader data processing device cannot send orders as long as the credit balance is used up.

18. A trading system, for example according to one of the claims 1 to 13, 15 or 16, comprising

- a first trading platform, a second trading platform; and
- a plurality of trader data processing devices;
- wherein each trader data processing device is capable of generating orders in the form of electronic messages and sending them;
- wherein each trader data processing device is connected to the first as well as second trading platform, so that each order gets to the first as well as to the second trading platform;
- wherein the first trading platform is capable of storing these orders in a first order book or, given a matching order already stored in the first order book, of at least partly carrying them out;
- wherein the second trading platform is capable of storing these orders in a second order book or, given a matching order already stored in the second order book, of at least partly carrying them out;
- and wherein the system is configured such that the first trading platform as well as the second trading platform processes incoming orders according to defined rules which are identical for the first and the second trading platform, so that the first order book and the second order book are identical with regard to contents, at all times.

19. A trading system according to claim 18, wherein the first and the second trading platform in each case comprise a prioritiser which suppresses the messages which are identical to an already arrived message.

20. A trading system according to one of the claims 1-13 or 16-19, wherein a condition of the system can be represented by a number of data elements, and wherein the trading system is configured to send condition data in each case representing the current condition of the data element, in packets, in a continuous manner during operation and intermittently with the sending of messages, to at least all trader data processing devices, at least until a data packet has been sent for each data element.

21. A trading system according to claim 20, wherein the sending of the condition data is continued in a continuously running manner, by way of the sending of condition data for

each data element beginning from afresh in each case after the sending of a data packet for each data element.

22. A trading system according to claim 20 or 21, wherein each trader data processing device is configured to process a message concerning a data element, when condition data for this data element is present, and to reject this message when no condition data for this data element is present.

23. A trading system according to one of the claims 1-13, 15, 16 or 18-21 comprising several levels of switches, wherein each trader data processing device is connected to the trading platform data processing device via an identical number of switches.

24. A method, in order to synchronise a unit of a stock exchange trading system, in particular according to one of the claims 1-13, 15, 16 or 18-22, with other units of the system, wherein condition data is sent in packets to all units in a continuous manner during operation of the stock exchange trading system, wherein each packet contains a part of the condition data, wherein the sending of the condition data is carried out intermittently with the sending of current messages, until all data elements are distributed, whereupon condition data is sent anew.

25. A method according to claim 24, wherein a unit to be synchronised processes a current message when it relates to a data element for which it has already received condition data, and ignores this message when it relates to a data element for which it has still not received condition data.

26. A computer program which can be loaded onto a data processing device or a system of data processing devices and which on implementation permits the data processing device or the system of data processing devices to carry out a method according to one of the claims 14, 17, 24 or 25.

27. A data carrier, comprising a computer program according to claim 26.