(12) UK Patent Application (19) GB (11) 2 354 847 (13) A

(43) Date of A Publication 04.04.2001

(21) Application No 9922812.4

(22) Date of Filing 28.09.1999

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Stephen James Paul Todd
Malcolm D Ayres
Michael George Taylor
Timothy Holloway
Marc-Thomas Schmidt
Simon A Holdsworth

(74) Agent and/or Address for Service
Kevin J Fournier
IBM United Kingdom Limited, Intellectual Property
Department, Hursley Park, WINCHESTER, Hampshire,
SO21 2JN, United Kingdom

(51) INT CL^7
G06F 9/46 17/30

(52) UK CL (Edition S)
G4A AFGDC APX AUDB

(56) Documents Cited
EP 0759591 A1 US 4815030 A

(58) Field of Search
UK CL (Edition R) G4A AFGDC APX AUDB
INT CL^7 G06F

(54) Abstract Title

Publish/subscribe data processing with subscription points for customised message processing

(57) A message broker data processing method includes steps of: receiving published messages 325 on a topic from a publisher application 31; processing 321 the received messages; and distributing the processed messages to subscriber applications 33,34; where the distributing step is carried out via a plurality of subscription point data processing nodes 323,324, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes. The processing 321 may be format conversion.

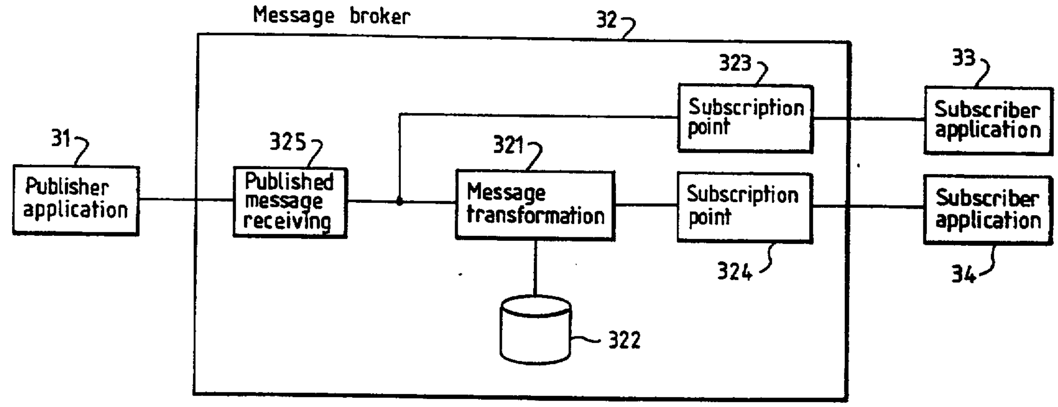

FIG. 3

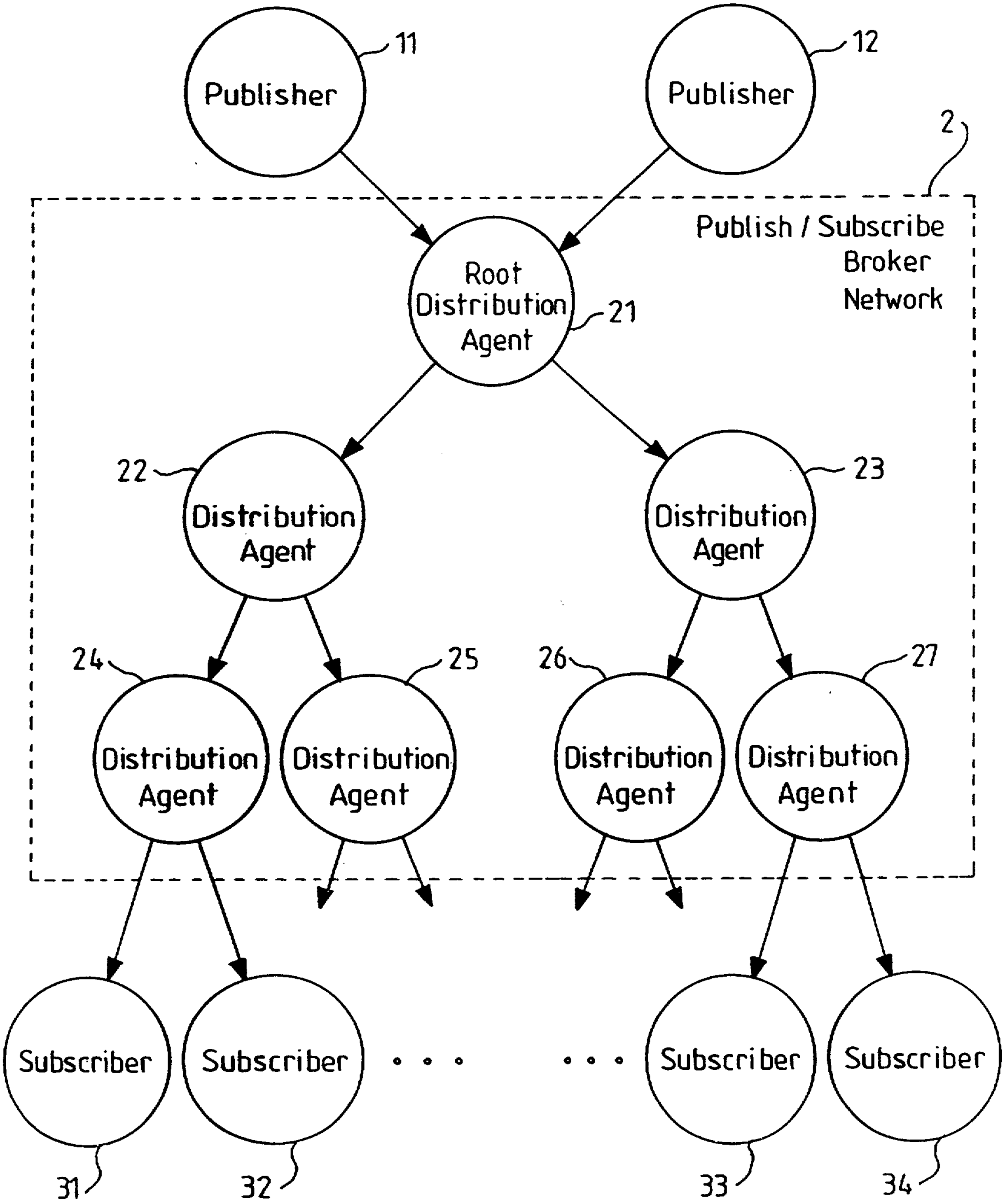
Publisher
11
Publisher
12
2
Publish / Subscribe
Broker
Network
Root
Distribution
Agent
21
22
Distribution
Agent
23
Distribution
Agent
24
Distribution
Agent
25
Distribution
Agent
26
Distribution
Agent
27
Distribution
Agent
Subscriber
Subscriber
Subscriber
Subscriber
31
32
33
34

FIG. 1

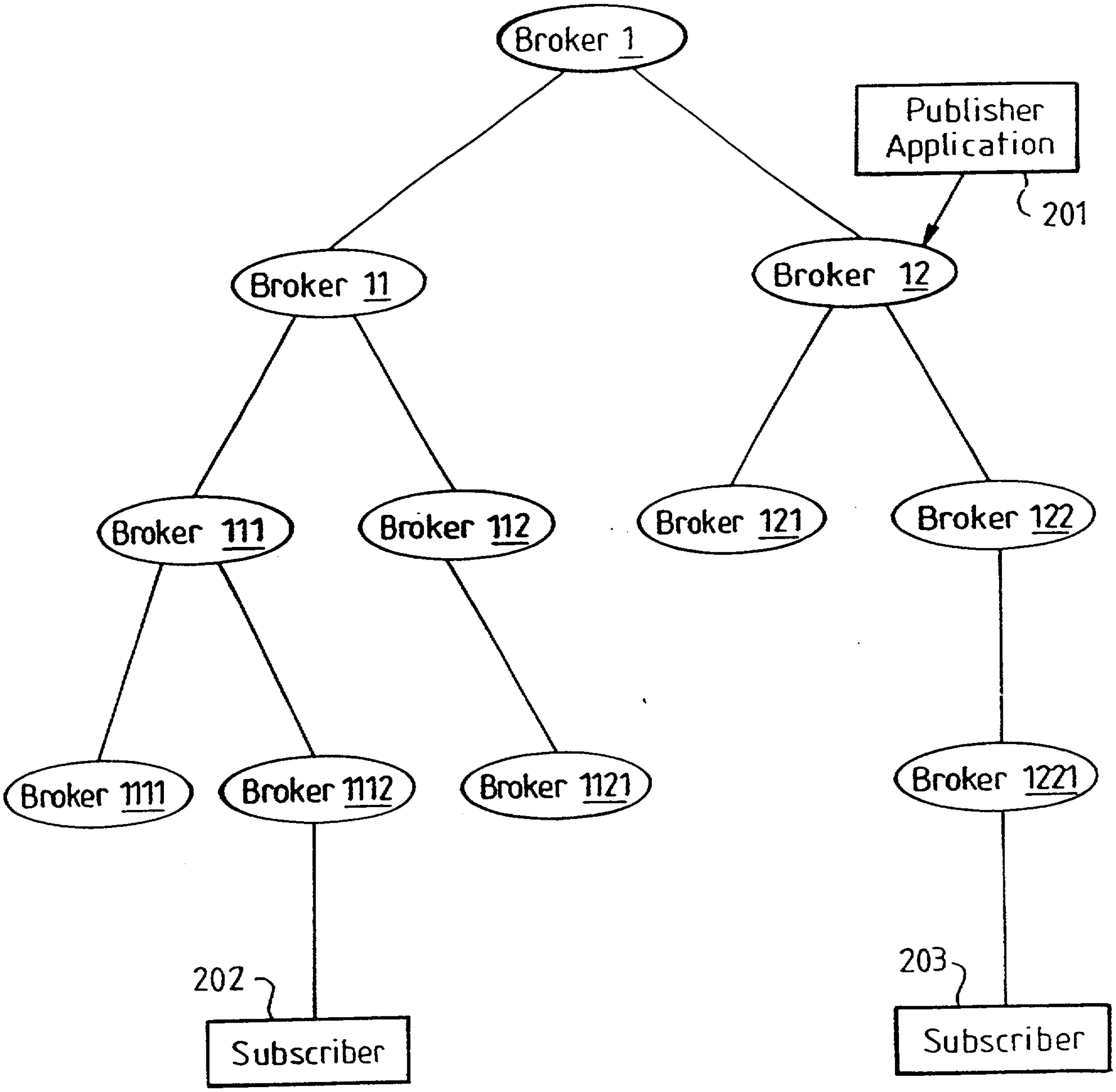
Broker 1
Publisher Application
201
Broker 11
Broker 12
Broker 111
Broker 112
Broker 121
Broker 122
Broker 1111
Broker 1112
Broker 1121
Broker 1221
202
Subscriber
203
Subscriber

FIG. 2

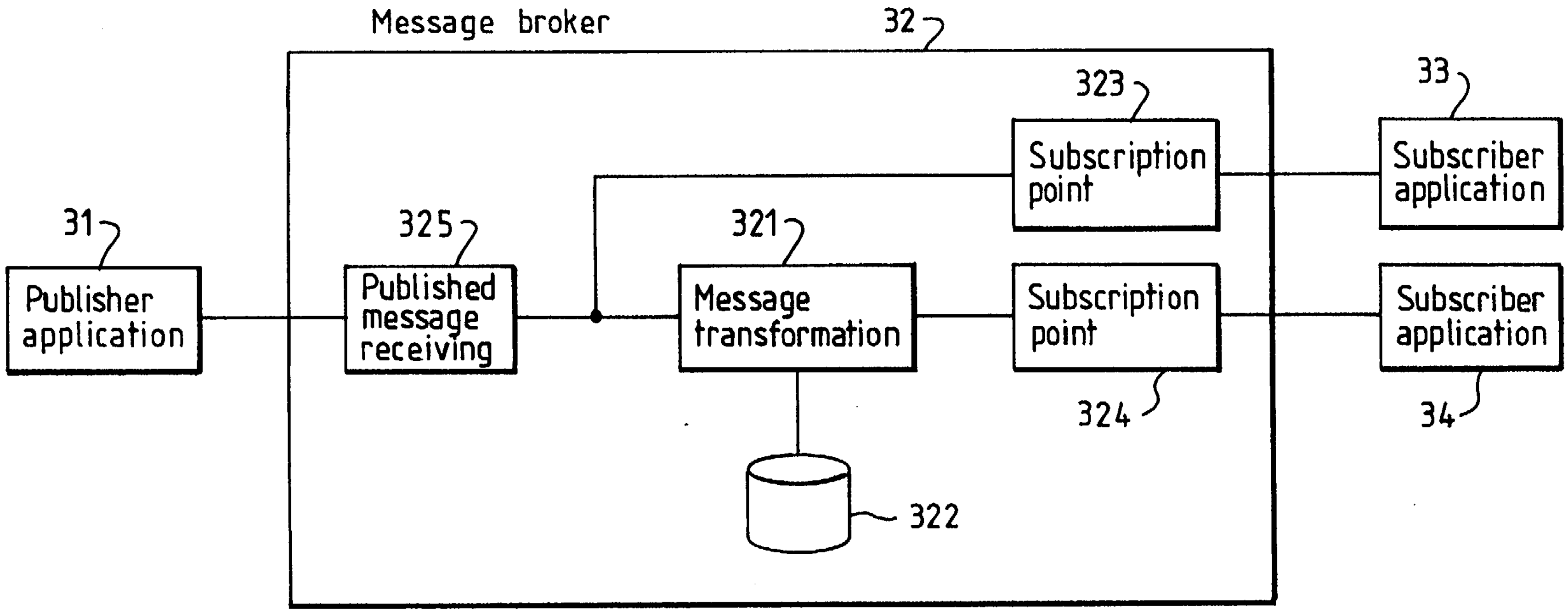
Message broker
32
31
Publisher application
325
Published message receiving
321
Message transformation
322
323
Subscription point
324
Subscription point
33
Subscriber application
34
Subscriber application

FIG. 3

PUBLISH/SUBSCRIBE DATA PROCESSING WITH SUBSCRIPTION POINTS FOR CUSTOMISED MESSAGE PROCESSING

Field of the Invention

The present invention relates to the field of data processing and more specifically to event notification data processing which distributes event messages from suppliers (called, hereinafter, "publishers") of data messages to consumers (called, hereinafter "subscribers") of such messages. While there are many different types of known event notification systems, the subsequent discussion will describe the publish/subscribe event notification system as it is one of the most common.

Background of the Invention

Publish/subscribe data processing systems (and event notification systems in general) have become very popular in recent years as a way of distributing data messages (events) from publishing computers to subscribing computers. The increasing popularity of the Internet, which has connected a wide variety of computers all over the world, has helped to make such publish/subscribe systems even more popular. Using the Internet, a World Wide Web browser application (the term "application" or "process" refers to a software program, or portion thereof, running on a computer) can be used in conjunction with the publisher or subscriber in order to graphically display messages. Such systems are especially useful where data supplied by a publisher is constantly changing and a large number of subscribers needs to be quickly updated with the latest data. Perhaps the best example of where this is useful is in the distribution of stock market data.

In such systems, publisher applications of data messages do not need to know the identity or location of the subscriber applications which will receive the messages. The publishers need only connect to a publish/subscribe distribution agent process, which is included in a group of such processes making up a broker network, and send messages to the distribution agent process, specifying the subject of the message to the distribution agent process. The distribution agent process then distributes the published messages to subscriber applications which have previously indicated to the broker network that they would like to receive data messages on particular subjects. Thus, the subscribers also do not need to know the identity or location of the publishers. The subscribers need only connect to a distribution agent process.

One such publish/subscribe system which is currently in use, and which has been developed by the Transarc Corp. (a wholly owned subsidiary of the assignee of the present patent application, IBM Corp.) is shown in Fig. 1. Publishers 11 and 12 connect to the publish/subscribe broker network 2 and send published messages to broker network 2 which distributes the messages to subscribers 31, 32, 33, 34. Publishers 11 and 12, which are data processing applications which output data messages, connect to broker network 2 using the well known inter-application data connection protocol known as remote procedure call (or RPC)(other well known protocols, such as asynchronous message queuing protocols, can also be used). Each publisher application could be running on a separate machine, alternatively, a single machine could be running a plurality of publisher applications. The broker network 2 is made up of a plurality of distribution agents (21 through 27) which are connected in a hierarchical fashion which will be described below as a "tree structure". These distribution agents, each of which could be running on a separate machine, are data processing applications which distribute data messages through the broker network 2 from publishers to subscribers. Subscriber applications 31, 32, 33 and 34 connect to the broker network 2 via RPC in order to receive published messages.

Publishers 11 and 12 first connect via RPC directly to a root distribution agent 21 which in turn connects via RPC to second level distribution agents 22 and 23 which in turn connect via RPC to third level distribution agents 24, 25, 26 and 27 (also known as "leaf distribution agents" since they are the final distribution agents in the tree structure). Each distribution agent could be running on its own machine, or alternatively, groups of distribution agents could be running on the same machine. The leaf distribution agents connect via RPC to subscriber applications 31 through 34, each of which could be running on its own machine.

In order to allow the broker network 2 to determine which published messages should be sent to which subscribers, publishers provide the root distribution agent 21 with the name of a distribution stream for each published message. A distribution stream (called hereinafter a "stream") is an ordered sequence of messages having a name (e.g., "stock" for a stream of stock market quotes) to distinguish the stream from other streams (this is known as "topic based" publish/subscribe, another well known model is called "content based publish/subscribe which involves matching publishers and subscribers by the content of the messages rather than by the topic). Likewise, subscribers provide the leaf distribution agents 31 through 34 with the name of the streams to which they would like to subscribe. In this way, the broker network 2 keeps track of which subscribers are interested in which streams so that when publishers publish messages to such streams, the messages can be distributed to the corresponding subscribers. Subscribers are also allowed to provide filter expressions to the broker network in order to limit the messages which will be received on a particular stream (e.g., a subscriber 31 interested in only IBM stock quotes could subscribe to the stream "stock" by making an RPC call to leaf distribution agent 24 and include a filter expression stating that only messages on the "stock" stream relating to IBM stock should be sent to subscriber 31).

The above-described publish/subscribe architecture provides the advantage of central co-ordination of all published messages, since all publishers must connect to the same distribution agent (the root) in order to publish a message to the broker network. For example, total ordering of published messages throughout the broker network is greatly facilitated, since the root can easily assign sequence numbers to each published message on a stream. However, this architecture also has the disadvantage of publisher inflexibility, since each publisher is constrained to publishing from the single root distribution agent, even when it would be much easier for a publisher to connect to a closer distribution agent.

In the Fig. 1, a publisher application 11, running on one computer, is, for example, a supplier of live stock market data quotes. That is, publisher application 11 provides frequent messages stating the present value of share prices. In this example, publisher application 11 is publishing messages on a stream called "stock" which has already been configured in the broker network 2. As is well known, when publisher 11 wishes to publish a stock quote message to stream "stock", publisher 11 makes an RPC call to the root distribution agent 11 which is at the top level of the broker network tree structure. In this example, subscriber application 32, running on another computer, has sent a subscription request via an RPC call to leaf distribution agent 24, which is at the bottom level of the tree structure, indicating that subscriber 32 would like to subscribe to stream "stock".

Thus, whenever publisher 11 publishes a data message to stream "stock" the distribution tree structure of broker network 2 channels the message down through the root distribution agent 21, through any intermediary distribution agents (e.g., 22 in the example of Fig. 1) and through the leaf distribution agent 24 to the subscriber 32. This involves a series of RPC calls being made between each successive circle in the diagram of Fig. 1 connecting publisher 11 and subscriber 32 (i.e., 11 to 21, 21 to 22, 22 to 24 and 24 to 32).

Figure 2 shows a different publish/subscribe architecture where publisher applications can publish messages to the broker network by directly communicating with any one of a plurality of distribution agents (brokers). For example, publisher application 201 is shown communicating directly with Broker 12. There is no requirement in this architecture that all publisher applications communicate directly with a top (or root) distribution agent. Publisher application 201 can potentially communicate directly with any of the distribution agents shown in Fig 2, in the described examples below it will be shown communicating directly with Broker 12.

Subscriber applications 202 and 203 would like to receive messages on the stream/topic that publisher application 201 is publishing on. Thus, subscriber applications 202 and 203 communicate directly with Brokers 1112 and 1221, respectively, to provide subscription data thereto informing the broker hierarchy of their desire to receive such published messages. Since the publisher application 201 is allowed to communicate directly with any of a plurality of distribution agents, the subscription data entered by the subscriber applications must be propagated throughout the broker network to each Broker shown in Fig. 2. This way, no matter which distribution agent the publisher application 201 happens to communicate directly with, the published messages will be able to be routed to the subscriber applications 202 and 203.

Publish/subscribe broker systems have commonly been integrated into multi-function message broker systems which are used to inter-connect applications which may be on heterogeneous platforms and may use different message formats. For example, Saga Software of Reston, Virginia (USA) (www.sagasoftware.com) have such a message broker product called "Sagavista" (a trademark of Saga Software). Further, Tibco Software Inc. of Palo Alto, California (USA)(www.tibco.com) also have such a message broker called "TIB/Message Broker" (both "TIB" and "TIB/Message Broker" are trademarks of Tibco). In these multi-function message brokers, a set of pluggable data processing nodes is provided, with each node being dedicated to a specific data processing task, such as message format transformation, publish/subscribe message distribution, and a rules engine for deciding (based on a plurality of predefined rules) where an incoming message should be routed.

In these multi-function message broker products, when a subscriber application registers a subscription request with the broker, the subscriber application sends the subscription request to a publish/subscribe broker node specifying the topic of the desired subscription. The publish/subscribe broker node (usually in cooperation with a plurality of other such publish/subscribe broker nodes) then ensures that any published messages on that topic are sent to the subscriber application. Different subscribers may wish to receive the same published messages but in different message formats. For example, a subscriber in the United States may want to know IBM's stock price per share in US dollars while another subscriber in the United Kingdom may want to know IBM's stock price in UK (British) pounds.

In order to accommodate such format desires of various subscribers, the message broker would have to modify the topic after having performed a format transformation so that a subscriber can subscribe to this modified topic (rather than the original topic that the publisher published on) in order to receive the format-transformed messages. Alternatively, the publishers would have to publish the same messages in different formats(with each format having its own topic), thus doing away with the need for the broker to do the format transformation. Because the topic needs to correspond to the format in both of these cases, this can cause many problems. For example, it is very useful to carry out access control on a topic basis. That is, when deciding which subscribers can have access to which published messages, it is very useful to be able to use the topics of the messages to make such access control decisions. However, when the topics must be different for essentially the same group of messages because of format changes, such access control decisions become much more complex.

It would be clearly desirable to be able to use the same topic for a variety of different message formats in a message broker, but the present state of the art does not allow for this.

Summary of the Invention

According to one aspect, the present invention provides a message broker data processing apparatus including: a unit for receiving published messages on a topic from a publisher application; a unit for processing the received messages; and a unit for distributing the processed messages to subscriber applications; where the unit for distributing includes a plurality of subscription point data processing nodes, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes with the broker apparatus.

According to a second aspect, the present invention provides a data processing method of carrying out the functionality discussed above with respect to the first aspect.

According to a third aspect, the present invention provides a computer readable storage medium having a computer program stored on it which, when executed on a computer, carries out the functionality of data processing method of the second aspect of the invention.

Thus, the present invention provides a message broker having a publish/subscribe capability where subscribers can receive messages in their desired formats without the need for either the broker or the publishers to modify the topic names used by the publisher, broker and subscriber. Instead, the publisher, broker and subscriber can use the same topic name even though the messages sent under this topic will be of differing formats. The presence of multiple subscription points within the broker provides for this ability.

As one advantage of the invention, access control can thus be easily carried out using the topic name. Further, the publisher application does not need to publish the same messages on a plurality of topics in order to accommodate subscribers who want publications in differing formats, thus decoupling the publisher application from having to deal with the varying desires of subscribers. The publisher need only publish messages in the format most convenient to that publisher.

Brief Description of the Drawings

The invention will be better understood by referring to the detailed description of the preferred embodiments which will now be described in conjunction with the following drawing figures:

Figure 1 is a block diagram showing a first architecture of a publish/subscribe data processing system to which the preferred embodiment of the present invention can be advantageously applied;

Figure 2 is a block diagram showing a second architecture of a publish/subscribe data processing system to which the preferred embodiment of the present invention can be advantageously applied; and

Fig. 3 is a block diagram showing an exemplary message broker according to a preferred embodiment of the present invention.

Detailed Description of the Preferred Embodiments

In Fig. 3 a message broker 32 receives published messages on a topic called "IBM stock" from a publisher application 31 (which is an application running at a major stock exchange) and distributes such published messages to subscriber applications 33 and 34 (which are two different stock broking agencies), each of which have previously registered subscriptions to the topic "IBM stock". In this example, the publisher application, broker and subscriber applications are all running on separate machines (and are thus interconnected via a network which is not shown in Fig. 3). In other embodiments, however, two or more of the applications (e.g., the publisher and the broker) could be running on the same machine. Further, as was explained above, the broker 32 is most likely running on a plurality of machines.

When the published messages are received into the message broker 32 at a published message receiving data processing node 325, they are sent to a message transformation data processing node 321 which performs the function of transforming the format of the published messages so that the IBM stock prices, which are originally published in US dollars, are converted to UK pounds. The message transformation node 321 accesses local storage 322 in order to determine the current exchange rate of US dollars to UK pounds (this exchange rate is updated at the beginning of every business day). After having their US dollar amounts converted to UK pounds, the messages are output from the message transformation node 321 and received at a subscription point processing node 324.

A subscription point data processing node (or "subscription point" for short) is an instance of a publish/subscribe matching engine which performs the function of looking at the topics in previously received subscription requests (received from subscribers) and determining whether the topic in an incoming message (just received from a publisher application) matches the topic of any of the previously received subscription requests. For any subscriptions that match, the subscription point data processing node distributes the published message to the subscriber application(s) which had entered the subscription requests.

Back to Fig. 3, the subscription point processing node 324 determines (e.g., by accessing local storage 322) that subscriber application 34 has previously entered a subscription on the topic "IBM stock". Thus, subscription point processing node 324 distributes the published messages to subscriber application 34.

Upon entering the broker 32, published messages from publisher 31 also take another path bypassing the message transformation data processing node 321. Specifically, the published messages also are sent directly to subscription point data processing node 323 (which is separate from subscription point data processing node 324 discussed earlier). Subscription point data processing node 323 then performs a publish/subscribe topic matching operation and determines that subscriber application 33 has previously entered a subscription request to the topic "IBM stock". Thus, subscription point processing node 323 distributes the published messages to subscriber application 33.

When subscriber application 33 registered its subscription to the topic "IBM stock" with the broker 32, subscriber application 33 provided the broker not only with the name of the topic "IBM stock" but also with the identity of subscription point 323 as the desired subscription point in the message flow of data processing nodes within the broker 32. This informs the broker 32 that subscription point 323 should be used to satisfy subscriber application 33's subscription request. Subscriber application 33 has selected subscription point 323 because subscriber application 33 is being run by a stock broker agency located in the United States and thus subscriber application 33 desires to receive publications on topic "IBM stock" with the stock prices quoted in US dollars. Because the message flow through the data processing nodes of message broker 32 for subscription point 323 bypasses the message transformation node 321, the original format (stock prices in US dollars) of the published messages from publisher application 31 is maintained and the subscriber application 33 receives such published messages without message transformation.

On the other hand, when subscriber application 34 registered its subscription to the topic "IBM stock" with the broker 32, subscriber application 34 provided the broker not only with the name of the topic "IBM stock" but also with the identity of subscription point 324 as the desired subscription point in the message flow of data processing nodes within the broker 32. This informs the broker 32 that subscription point 324 should be used to satisfy subscriber application 34's subscription request. Subscriber application 34 has selected subscription point 324 because subscriber application 34 is being run by a stock broker agency located in the United Kingdom and thus subscriber application 34 desires to receive publications on topic "IBM stock" with the stock prices quoted in UK pounds. Because the message flow through the data processing nodes of message broker 32 for subscription point 324 includes the message transformation node 321, the original format (stock prices in US dollars) of the published messages from publisher application 31 is converted into UK pounds and the subscriber application 34 receives such published messages after having had the format of such messages transformed.

Thus, by the use of a plurality (two in Fig. 3) of subscription point data processing nodes in a message broker, subscriber applications can be provided with published messages in a message format selected by the subscriber without having to use different topics (the topic "IBM stock" is the same for both subscription points 323 and 324). This allows access control to be easily carried out on a topic basis. For example, the broker can perform a security measure on both subscriber applications 33 and 34 by simply checking whether the requested topic "IBM stock" of their subscription requests is a topic which has previously been determined as acceptable for subscribers 33 and 34 from a security standpoint.

The use of subscription points in message flows through the broker is not limited to message format transformation. Such subscription points could also be used to implement different filters on subscription topics. For example, message transformation data processing node 321 could be replaced with a topic filtering data processing node which carries out the function of filtering any published messages on topic "IBM stock" where the stock price is greater than 100 US dollars per share. That is, the filtering data processing node examines the stock price in each published message on topic "IBM stock" and only if the stock price is greater than 100 dollars per share is the published message allowed to pass through the filtering node. The published messages that make it through the filtering node are then sent to subscription point data processing node 324 which performs a topic matching function (as described above) and distributes the published messages to subscriber application 34. In this case, subscriber application 34 is being run by a US-based stock broker agency which is only concerned with messages concerning IBM stock if the IBM stock price per share is greater than 100 US dollars.

A system may include multiple brokers each of which provides an equivalent set of multiple subscription points. These brokers are arranged as described earlier for publish/subscribe systems so that a subscriber subscribing to a particular subscription point receives the same set of messages regardless of the broker to which the subscription is made. Each broker subscribes to the subscription points of other brokers on behalf of client subscribers. Resulting interbroker messages all flow multiplexed onto a single communication channel (e.g. MQSeries queue) between two given connected brokers, regardless of the subscription point to which the message belongs. The source broker tags the interbroker message with header information indicating the subscription point to which the message belongs. The receiving broker uses the header tag to publish the messages to all subscribers at the given equivalent subscription point on the receiving broker.

CLAIMS

1. A message broker data processing apparatus comprising:

means for receiving published messages on a topic from a publisher application;

means for processing the received messages; and

means for distributing the processed messages to subscriber applications;

wherein the means for distributing includes a plurality of subscription point data processing nodes, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes with the broker apparatus.

2. The apparatus of claim 1 wherein said apparatus communicates with a subscriber application over the Internet and wherein at least one of the subscriber applications and the publisher application runs in conjunction with a World Wide Web browser.

3. A message broker data processing method comprising steps of:

receiving published messages on a topic from a publisher application;

processing the received messages; and

distributing the processed messages to subscriber applications;

wherein the distributing step is carried out via a plurality of subscription point data processing nodes, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes.

4. A computer program product stored on a computer readable storage medium for, when run on a computer, instructing the computer to carry out the method steps recited in claim 3.

The Patent Office

INVESTOR IN PEOPLE

Application No:	GB 9922812.4	**Examiner:**	Mike Davis
Claims searched:	1-4	**Date of search:**	20 April 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AFGDC, APX, AUDB)

Int Cl (Ed.7): G06F

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	EP 0759591 A1	(IBM) eg abstract	1-4
X	US 4815030	(CROSS ET AL) eg column 4 line 34 to column 6 line 5, and column 8 lines 3-27	1-4

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