

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
23 September 2004 (23.09.2004)

PCT

(10) International Publication Number
WO 2004/081818 A1

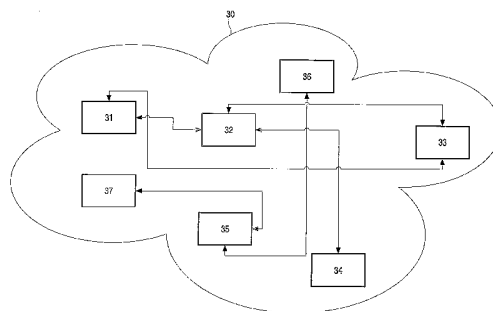
- (51) International Patent Classification⁷: **G06F 17/30**, H04L 29/08
- (21) International Application Number: PCT/IB2004/050215
- (22) International Filing Date: 9 March 2004 (09.03.2004)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 03100591.1 10 March 2003 (10.03.2003) EP
- (71) Applicant (for all designated States except US): **KONINKLIJKE PHILIPS ELECTRONICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **FONTIJN, Wilhelmus, F., J.** [NL/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). **SINITSYN, Alexandre** [BY/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (74) Agent: **GROENENDAAL, Antonius, W., M.**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR OWNERSHIP TRANSFER OF TRANSACTIONS IN PEER-TO-PEER SYSTEMS



(57) Abstract: This invention relates to a method of, a device and a protocol for performing ownership transfer of a change in a peer to peer network (30). The device or a peer can be a car, a garage, a video cassette recorder (VCR), a personal digital assistant (PDA), a mobile phone, a climate system, a television, a lamp, a coffee machine, a radio, a DVD player, a CD player, an information panel, a web tablet, a smart remote, an answering machine, a personal computer. Said method includes the steps of attempting, by means of a first device (31) to find a second device (32) which will accept responsibility for committing said change; transferring, by means of said first device, responsibility of committing the change to said second device and propagating the change to said second device, wherein a global commit status variable is further transferred to said second device and where said global commit status variable is maintained on said second device; setting, by means of said first device, a local commit status variable to "provisional" signifying that the device will act as if the global commit went through, wherein said first device will wait with setting the local commit status variable to 'void', signifying a real commit, until it gets confirmation that the global commit status variable is set to 'void' also, when said first device re-enters said network and checks the status of the global commit status variable; or setting, by means of said first device, the local commit status variable to 'void' signifying a real commit; and propagating, by means of said second device, said change to one or more devices (33, 34) for which said change is relevant, when responsibility of committing the change is received and accepted on said second device.

WO 2004/081818 A1



MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR OWNERSHIP TRANSFER OF TRANSACTIONS IN PEER-TO-PEER SYSTEMS

This invention relates to a method of performing ownership transfer of a change in a peer to peer network.

The change is transferred among various devices in said peer to peer network.

5 The present invention also relates to a computer system for performing the method.

The present invention further relates to a computer program product for performing the method.

Additionally, the present invention relates to a protocol for ownership transfer of a change.

10 The present invention further relates to a device corresponding to a peer, which belongs to said peer to peer network.

The invention further relates to a computer program product comprising code means stored on a computer readable medium for performing such a method.

15

WO 02/39305 discloses information management via delegated control. An information management system utilizes delegated control over a dataset. Said information management system comprises more computers and more software applications interacting to store information. Said delegated control is a transfer, temporary or partially of said dataset from a delegating system (as a so called "delegator") to a delegate system.

20 Committing transactions or delegating control in distributed databases is known to be difficult. In the art, currently there are three options for a basic transaction model:

25 Firstly, the originating part sends the update to a central transaction server and this server is responsible for updating all relevant parts and committing the change.

Secondly, the originating part propagates an update to all parts of the database that are relevant and commits the change as soon as it gets a message from all relevant parts that they received the update.

Thirdly, the commitment is not performed explicitly. There are protocols, e.g. so called “gossip protocols” that just propagate the change and assume a commit (see references below).

David Kempe, Jon M. Kleinberg, and Alan J. Demers. Spatial gossip and resource location protocols. In ACM Symposium on Theory of Computing, pages 163-172, 5 2001.

Alan Demers, Dan Greene, Carl Houser, Wes Irish, and John Larson. Epidemic algorithms for replicated database maintenance. SIGOPS, 22(1):8-32, 1987.

The first option of using the server is a problem since not all distributed 10 database have a central server that controls all transactions, e.g. in P2P (peer to peer) systems. In which case, the first option is out of question.

In addition, some distributed databases have parts that are not always in contact with all relevant parts and/or the central transaction server, i.e. ad-hoc connections, in which case the second option is out of question. As a result, the originating part may not be 15 able commit the change.

In many cases it is not an option not to commit explicitly because it means that there is no certainty about the commit. In this case the third option is out of question.

This leaves the problem of having no adequate reliable/robust transaction model in a peer to peer communication, i.e. it is a problem that a transaction of a change e.g. 20 to a database, a file, etc. – where the database, the file, etc. resides on other peer(s) than that who wishes to have the change performed – in some occasions is not performed. As a consequence, said file, database, etc. is left un-updated – and what is even worse – the requester of said change is not aware of it.

From the art it is known that Peer-to-peer is a communications model in which 25 each party (i.e. each peer) has the same capabilities and either party can initiate a communication session. Other models with which the Peer-to-peer communications model might be contrasted include the *client / server* model and the *master/slave* model, both also known in the art. In some cases, peer-to-peer communications is implemented by giving each communication node both server and client capabilities. In recent usage, peer-to-peer has 30 come to describe applications in which users can use the Internet to exchange files, update databases with each other directly or through a mediating server.

On the Internet, peer-to-peer (referred to as P2P) is a type of transient Internet network that allows a group of computer users (peers) with the same networking program to connect with each other and directly access files from one another's hard drives. Napster and

Gnutella are examples of this kind of peer-to-peer software. Corporations are looking at the advantages of using P2P as a way for employees to share files, update and access common databases, information, etc without the expense involved in maintaining a centralized server and as a way for businesses to exchange information with each other directly.

5 When the Internet P2P is applied, it is known in the art that the user must first download and execute a peer-to-peer networking program, e.g. Gnutella-net is currently one of the most popular of these decentralized P2P programs because it allows users to exchange all types of files. After launching the program, the user enters the IP address of another computer belonging to the network, typically, the Web page where the user obtained the
10 download will list several IP addresses as places to begin. Once the computer finds another network member on-line, it will connect to that user's connection, which has obtained their IP address from a connection of another user, and so on.

 It is further known in the art that users of peers can choose how many member connections to seek at one time and determine which files, databases, information items, etc
15 they wish to share, update or password protect, but still said problems are left unsolved.

 However, said problems are solved by said method of the present invention, when the method comprises the steps as discussed in figure 4.

 It is hereby an advantage of the invention that a method and a protocol, respectively is proposed that enables the delegation of the responsibility to commit a change.

20 The invention further has the advantage that commitment of a change can be effectuated even if the initiator (first device as claimed) of the change is no longer connected.

 In most cases the integrity of the distributed database can be maintained and guaranteed. Further, by not having a central server makes peer to peer network less vulnerable and further enables for any high number of peers communicating, i.e. the network
25 can be scaled up and down and still has said advantages.

 Said system, protocol and device, respectively provides the same advantages and solves the same problem(s) for the same reasons as described previously in relation to the method.

30

 The invention will be explained more fully below in connection with preferred embodiments and with reference to the drawings, in which:

 Fig. 1 shows various ways for a peer in contact with the system transferring responsibility for an update to a peer in constant contact with the system;

Fig. 2 shows responsibility transfer with status change on a commit status variable;

Fig. 3 shows a network of devices;

Fig. 4 shows a method of performing ownership transfer of a change in a peer
5 to peer network, and

Fig. 5 shows commit status variable changes during transaction ownership transfer.

10 Throughout the description of the present invention, transaction is understood as the following:

In computer programming, a transaction usually means a sequence of information exchange and related work (such as a database or a file updating) that is treated as a unit for the purposes of satisfying a request and for ensuring database or file integrity.

15 For a transaction to be completed, and database or file changes to be made permanent, the transaction has to be completed in its entirety. A typical business transaction could be a catalogue merchandise order phoned in by a customer and entered into a computer by a customer representative. The order transaction involves checking an inventory database, confirming that the item is available, placing the order, and confirming that the order has
20 been placed and the expected time of shipment. If this is considered as a single transaction, then all of the steps must be completed before the transaction is successful and the database is actually changed to reflect the new order. If something happens before the transaction is successfully completed, any changes to the database must be kept track of so that they can be undone, e.g. rolled back.

25 Figure 1 shows various ways for a peer in contact with the system transferring responsibility for an update to a peer in constant contact with the system.

In the figure, reference numeral (a) shows a peer in temporary contact with the system transferring responsibility (square) for an update (dark dot) to a peer in constant contact with the system.

30 Reference numeral (b) shows how the second peer accepts and the originator does a provisional commit (white dot).

Reference numeral (c) shows that the acceptor propagates the change to the other relevant peers.

Reference numeral (d) shows how the other peers acknowledge the change and that the change is committed (grey dots).

Reference numeral (e) shows that if the originator comes into contact with the system again it will check the status of the change.

5 Reference numeral (f) shows that the acceptor acknowledges the commit in the system.

Figure 2 shows responsibility transfer with status change on a commit status variable.

This figure illustrates three deviations from figure 1:

- 10
- 1) The original acceptor propagates the responsibility to other peers (a reason could be 3)).
 - 2) The originator assumes a real commit instead of a provisional.
 - 3) The scope of the peers is more limited.

Reference numeral (a) shows a peer in temporary contact with the system transferring responsibility (square) for an update (dark dot) to a peer in constant contact with the system. Reference numeral (b) shows that the second peer accepts and that the originator assumes it can do a real commit (grey dot). Reference numeral (c) shows that the acceptor propagates the change (dark dot) and that the responsibility (square) to other relevant peers it is in contact with. Reference numeral (d) shows that both accepting peers accept responsibility and that the original acceptor does a real commit. Reference numeral (e) shows
15 that the delegated acceptors propagate the update to further peers. Reference numeral (f) shows that the final update receiving peers acknowledge the commit in their system and that the delegated acceptors do the same. The latter also dissolves the update task.
20

With respect to state transition, three types of commits are mentioned, these are correspondingly reflected in a commit status variable having various states, i.e. a real
25 commit state, a provisional commit state or an assumed commit state, however the assumed commit is not a separate state, i.e. it is the same state as a real commit, but reached through a different transition.

The assumed commit state is a real commit without confirmation. The initial state is uncommitted. The final state is committed. This is identical to the real commit state.

30 The difference is that the real commit gets confirmation in-between. Note that the assumed commit state can be dangerous, since it can lead to inconsistencies, i.e. if the update is not committed by other peers, even though the update was assumed to occur. For this reason the state of a provisional commit can be applied. So the difference between the assumed commit state and the provisional commit state is that in the latter case there is still a flag (or a similar

indication) that indicates that the commit (state provisional) was not confirmed. If the confirmation arrives later, this flag can be removed, i.e. the commit state changes into (really) committed, i.e. the state of a real commit. So the state of a provisional commit can be seen as a 'real' commit for which the confirmation is expected to be late.

5 So, regarding commitment, four states exist for the commit status variable. By only looking at commitment there is no difference between 0 and 3, i.e. the database is with 3 in an updated state, but no updates are pending:

(0) no update pending = no state regarding commitment;

(1) un-committed = update pending;

10 (2) provisional commit = update is committed to the database but unconfirmed; and

(3) real commit = update is confirmed (others commit) or assumed and committed.

In state (0) an update request is received, which brings it to state (1). The peer can now wait till it gets a confirmation (real commit = state (3)), assume confirmation (assumed commit = state (3)), or pretend for the moment that it has the confirmation, because
15 it expects it will get it later (provisional commit = state (2)).

State (2) becomes state (3) if (finally) confirmation arrives. State (3) equals state (0), i.e. the updated state which corresponds to that no update is pending.

Figure 3 shows a network of devices. Said network of devices is illustrated by means of reference numeral 30. As will be explained more detailed in the next figure, a first
20 device, reference numeral 31, will do the attempt to find another, i.e. a second device, reference numeral 32, which will accept responsibility for committing a change. As a result, said second device will propagate the change to at least one more device, e.g. reference numerals 33 and 34 assuming said change is relevant for these devices. In the network further devices may be present, e.g. reference numerals 35, 36 and 37. The network is shown for
25 illustrative purposes, any other dynamic or static topology or arrangement of peers or devices may also be applied in the present invention.

Any of said devices may be a car, a garage, a video cassette recorder (VCR), a personal digital assistant (PDA), a mobile phone, a climate system, a television, a lamp, a coffee machine, a radio, a DVD player, a CD player, an information panel, a web tablet, a
30 smart remote, an answering machine or a personal computer. As an example, in principle, the lamp with access to the network may communicate change, e.g. a schedule change to a personal digital assistant (PDA), a web tablet, a smart remote, an answering machine and/or a personal computer, hereby it is assured that the user most likely will receive said schedule change.

The device alternatives as mentioned may be understood as corresponding peers in a peer-to-peer type of transient network similar to the type found on the Internet, that allows a group of computer users (with access to their corresponding peers or devices) with the same or similar networking program or protocol to connect with each other and directly
5 access and/or update files, databases, etc to/from one another's hard drives, memories, etc. A peer-to-peer network is simply a network of peers, the Internet, Gnutella software, computers are all just examples of aspects of specific implementations.

Said state change can be applied in a protocol used for ownership transfer of a change, i.e. the protocol will comprise a commit status variable having various states, i.e. a
10 real commit, and a provisional commit. If the originator of an update request tries to transfer the responsibility for commitment, it can also communicate which state it will be in after the responsibility is accepted by another. The originator can be uncommitted, committed or provisionally committed. The first case, i.e. the originator is uncommitted, is avoided according to the present invention since the acceptor needs to wait for commitment from the
15 originator. The second case, i.e. the originator is committed, committed requires no further action from the acceptor towards the originator. The third case, i.e. the originator is provisionally committed, means that the acceptor needs to keep in mind that the originator may want or need confirmation later.

Note that any type of commit is a state change, i.e. the assumed commit is not
20 a state it is a state transition.

The protocol may be applied in ownership transfer of changes among various devices (similar to peers), e.g. in a peer to peer network or in a similar network without a central server, it can in fact also be applied in a system with a central server but it makes less sense.

25 Said change can be any change to a database and / or to a file. Additionally or alternatively, said change may be a change to any information item, such as a variable, one or more parameters, one or more status flags, a string variable, etc.

In other words, said change may have the effect that text, numerical
information, picture, video, sound and combinations thereof subsequently are updated in a
30 file and/or a database.

The file and/or database may be stored individually or distributed in any device communicating on the peer to peer network or in a similar network.

In the following various practical applications of the invention are shown, the update is similar to said change.

Example 1) Travelling commits:

Johan has forgotten certain drawings in his vault at home. He has no time to get them and asks Hendrik to get them for him. He changes his security settings of his home on his PDA device to allow Hendrik into his garage, his study and to open his vault. 5 However, for security reasons he can not change these settings online. He transfers the responsibility for the update of his settings to his car and gives Hendrik his car keys. Hendrik uses the car of Johan to drive to Johan's home. As Hendrik reaches Johan's home, the car - as another device in the network - transfers the update of the security settings to the garage. The 10 garage propagates the change to the rest of the devices of the home and Hendrik can get what he came for. As Hendrik leaves, security settings, i.e. a new change, are reverted to exclude Hendrik again. The garage - as yet another device in the network - informs the car of this update, i.e. the change. Back with Johan in the office the car updates the settings on Johan's PDA and Johan can trust that all is back to normal again. Hendrik hands him the drawings. 15

Example 2) Fire and forget approach to updating code or parameters on many peers or devices:

Pieter invites Fien to visit him for dinner. Unexpectedly, she says yes. He uses his mobile phone - as yet another device in the network - to prepare his ambient home for 20 dinner with Fien. He has no time to go through all changes required but his answering machine (as another device, etc) accepts responsibility for propagating this information to all other relevant devices. As his PVR (as another device, etc) gets this information it prepares to record the live cricket game Pieter intended to watch. The climate system (as another device, etc) prepares to elevate the temperature from the usual 18 degrees to 19.5 degrees centigrade. 25 The kitchen checks its stocks for a dinner for two with some class. It decides to order in some Cajun food. The messaging service reschedules the appointment with the 24 hours dentist on his corresponding device, e.g. his PDA. As Pieter gets home, he immediately gets informed that all but one of the changes due to dinner with Fien were successful so he can relax in anticipation of the arrival of Fien. There is one hitch, the rescheduling of his dentist 30 appointment did not succeed and he will have to make a new appointment himself. The Cajun food arrives 15 minutes before Fien does so Pieter has time to put it in his own cooking ware.

Other Example:

- A group of persistent peers sort of emulate a central server towards multiple ad-hoc connected devices. At home Wubbo has several persistent peers that together handle his agenda. Any device can always offload an item for the agenda to any of the persistent peers (i.e. devices) and rely on the change being committed.
- Fast offload of updates before a pending shutdown. Carol's smart remote (as another device, etc) could not commit all pending changes before an imminent shutdown due to loss of power. It transfers responsibility to one of the responders (as other devices, etc) in the wall.
- A task may require a sequence of devices each performing a subtask. The responsibility for the task travels with the task.

Figure 4 shows a method of performing ownership transfer of a change in a peer to peer network. Said peer to peer network is similar to the network of devices from the foregoing figure, i.e. the ownership transfer of the change can be performed among devices in said networks.

Prior to the following steps, it is assumed that a change is originated at a device. This can be seen in the next figure, figure 5 which is generally also referred to in the present method description. In figure 5(a), the originator or the first device as claimed (corresponding to reference numeral 31 in figure 3)) attempts to find another, i.e. a second device, reference numeral 32, etc. At this stage only the first device knows about the change because it has not been communicated to any other devices. Two commit status variables are instantiated, which (to begin with) are both maintained by this originator. The first commit status variable has a local scope, signifying the status of the local database (value at this stage: 'pending'). The second commit status variable has a global scope, signifying whether the change has been committed in all relevant peers (value at this stage: 'pending').

The second device was previously called the acceptor since it may accept responsibility for committing said change.

In step 100 (figure 1a, figure 2a, figure 5(b), the first device may attempt to find a second device which will accept responsibility for committing said change.

The attempt may prove not to be a success, i.e. all commit states (commit status variables) remain 'pending' and the first device will have to try again (to find another device to accept responsibility for committing said change) or it may prove to be a success, i.e. the next step.

In step 200 (figure 1b, figure 2b, figure 5(c)), said first device may then transfer responsibility of committing the change to said second device. This means propagating the change to the acceptor (second device). Furthermore, this means that the responsibility for maintaining the global commit status variable is transferred to the second device. The
5 location of this global variable will be on the second device.

In step 300, the originator, i.e. said first device, may set its local commit status variable to 'provisional (figure 1b, figure 2b, figure 5(d2)), which signifies the device will act as if the global commit went through, but it will wait with setting the local commit status variable to 'void' until it gets confirmation that the global commit status variable is set to
10 'void' also. E.g. when said first device re-enters said network and check the status of the global commit status variable.

Normal procedure is to first check whether all parts of a database accept a certain change and then committing it (i.e. really going through with it). If a database is distributed, a device has to propagate the change to all relevant parts and they have to say that
15 they accept the change. They do the latter by sending a message to that effect to the part of the database that is responsible for committing the change (is maintaining the global commit status variable). In a P2P setting this is normally the originator (1st device), according to the present invention it is the acceptor (2nd device). If the acceptor has confirmation from all peers (devices) that should apply the change that they will actually apply the change then the
20 acceptor knows that the change is globally committed, so the global commit variable can be set to void.

The local commit status variable indicates whether the local database has applied the change or not.

The global commit status variable indicates whether all peers have applied the
25 change or not.

Alternatively, in step 310, the local commits status variable may be set to 'void' which signifies a real commit (figure 2b, figure 5(d1)).

Said first device can wait or have to wait a certain amount of time before re-entering the network, and hereby – in the unconnected situation – resources may be freed to
30 other tasks than communicating with the network, i.e. when said first device is a Personal Video Recorder it can dedicate its resources to record a movie.

In step 400, said second device may propagate said change to one or more devices for which said change is relevant (figure 1c, figure 2c, figure 5 (e, f, g, h)). This is the

case when responsibility of committing the change is received and accepted on said second device.

The method may here be successfully finished, i.e. the ownership transfer of the change is here successfully transferred.

5 However, it is possible that said method may further comprise the following two steps.

In step 500, said first device may convert the local commit status variable of a provisional commit into a real commit. This is the case when said first device re-enters said network and receives a message indicating a successful commit from said second device
10 (figure 5 (i)).

Again, said first device may wait a certain amount of time before entering the network. It may have the experience – from former situations spend unconnected – that it will take some time before the second device eventually will provide said successful commit message.

15 The originating device (said first device) may also convert the provisional status to ‘void’ after not being connected for a predetermined amount of time.

In step 600, said first device may receive a message indicating a non successful commit from said second device. That is the case when committing the change globally is not successfully performed by said second device, e.g. because the second device
20 failed to reach all relevant peers or because one or more peers refused to commit the change (e.g. due to locking or conflicting updates). Said message may be received when said first device enters said network the next time.

The method may here be successfully finalized; however it is possible that said method additionally comprises the following step.

25 The values of the variables and the names of the parameters can be changed without departing from the concept of the invention. For example the value of the parameter “global commit status” can be “commit” in stead of “void” to indicate that the commit was a success.

In step 700, said first device may roll back said change. This is the case, for
30 instance, when the local commit status (variable) is the provisional commit and step 600 occurs. The other peers (devices) did not commit so the provisional commit on the originator (first device) needs to be rolled back. If step 600 occurs and the originator has performed previously a real commit then no local commit status variable exists anymore for the change in question and the database is inconsistent. To remedy the situation the originator may

initiate the same change again (i.e. retry) or initiate a new change to counter the first change and to lift the inconsistency.

A computer readable medium may be magnetic tape, optical disc, digital versatile disk (DVD), compact disc (CD record-able or CD write-able), mini-disc, hard disk,
5 floppy disk, smart card, PCMCIA card, etc.

In the claims, any reference signs placed between parentheses shall not be constructed as limiting the claim. The word "comprising" does not exclude the presence of elements or steps other than those listed in a claim. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements.

10 The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In the device claim enumerating several means, several of these means can be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to
15 advantage.

CLAIMS:

1. A method of performing ownership transfer of a change in a peer to peer network (30), said method comprising the steps of:
 - attempting (100), by means of a first device (31) to find a second device (32) which will accept responsibility for committing said change;
 - 5 – transferring (200), by means of said first device, responsibility of committing the change to said second device and propagating the change to said second device, wherein a global commit status variable is further transferred to said second device and where said global commit status variable is maintained on said second device;
 - setting (300), by means of said first device, a local commit status variable to
10 “provisional” signifying that the device will act as if the global commit went through, wherein said first device will wait with setting the local commit status variable to ‘void’, signifying a real commit, until it gets confirmation that the global commit status variable is set to ‘void’ also, when said first device re-enters said network and checks the status of the global commit status variable; or
 - 15 – setting (310), by means of said first device, the local commit status variable to ‘void’ signifying a real commit; and
 - propagating (400), by means of said second device, said change to one or more devices (33, 34) for which said change is relevant, when responsibility of committing the change is received and accepted on said second device.
 - 20
2. A method according to claim 1 further comprising the steps of:
 - converting (500), by means of said first device, the local commit status variable of a provisional commit into a real commit, when said first device re-enters said network and receives a message indicating a successful commit from said second device; and
 - 25 – receiving (600), a message on said first device indicating a non successful commit from said second device if committing the change is not successfully performed by said second device, when said first device re-enters said network.
3. A method according to claim 2 further comprising the step of:

- rolling back (700), by means of said first device, said change when the local commit status variable is the provisional commit and the situation of the receiving step (600) occurs.
- 5 4. A method according to any of claims 1 to 3 characterized in that said change is at least one of a change to a database, a change to a file and a change to an information item.
5. A method according to any of claims 1 to 4 characterized in that any of said devices is one of a car, a garage, video cassette recorder (VCR), a personal digital assistant
10 (PDA), a mobile phone, a climate system, a television, a lamp, a coffee machine, a radio, a DVD player, a CD player, an information panel, a web tablet, a smart remote, an answering machine, a personal computer, or any other electronic device.
6. A protocol for ownership transfer of a change characterized in that the
15 protocol comprises a commit status variable having at least one of the states of a real commit and a provisional commit.
7. A device (31) for performing ownership transfer of a change, said device (31) comprising:
- 20 – means for attempting to find a second device (32) which will accept responsibility for committing said change;
- means for transferring responsibility of committing the change to said second device (32) and propagating the change to said second device (32), and means for transferring a global commit status variable to said second device (32);
- 25 – means for setting a local commit status variable to “provisional” signifying that the device (31) will act as if the global commit went through, when said device (31) waits with setting the local commit status variable to ‘void’, signifying a real commit, until it gets confirmation that the global commit status variable is set to ‘void’ also, when said device (31) re-enters said network and checks the status of the global commit status
30 variable; or
- means for setting the local commit status variable to ‘void’ signifying a real commit; and
- means for propagating said change to one or more devices (33, 34).
8. A device (31) according to claim 7, said device (31) further comprising:

- means for converting the local commit status variable of an provisional commit into a real commit; and
- means for receiving a message indicating a non successful commit.

5 9. A device (31) according to claim 8, said device (31) further comprising:

- means for rolling back said change.

10. A computer system for performing the method according to any one of claims 1 through 5.

10

11. A computer program product comprising program code means stored on a computer readable medium for performing the method of any one of claims 1 through 5 when the computer program is run on a computer.

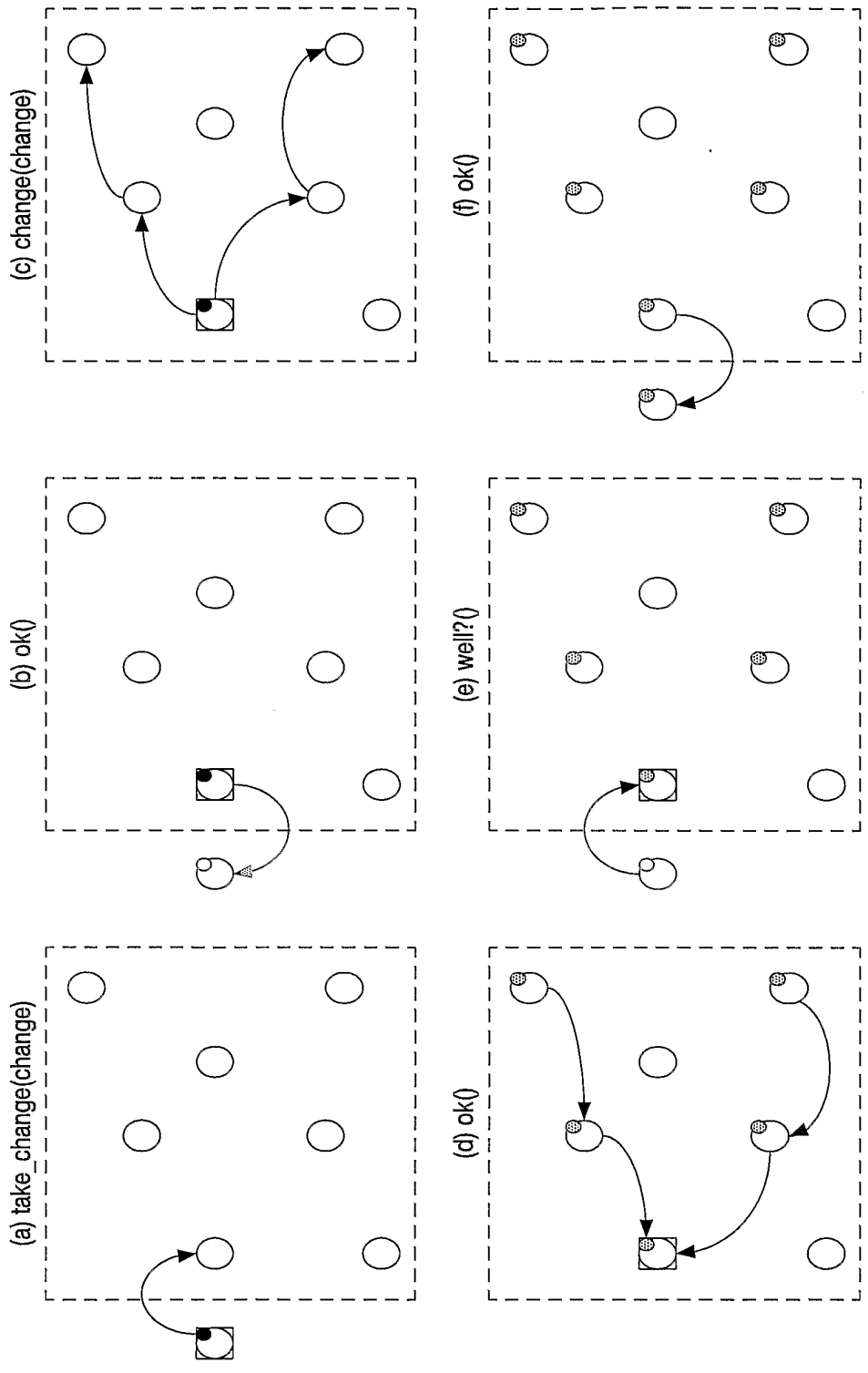


FIG.1

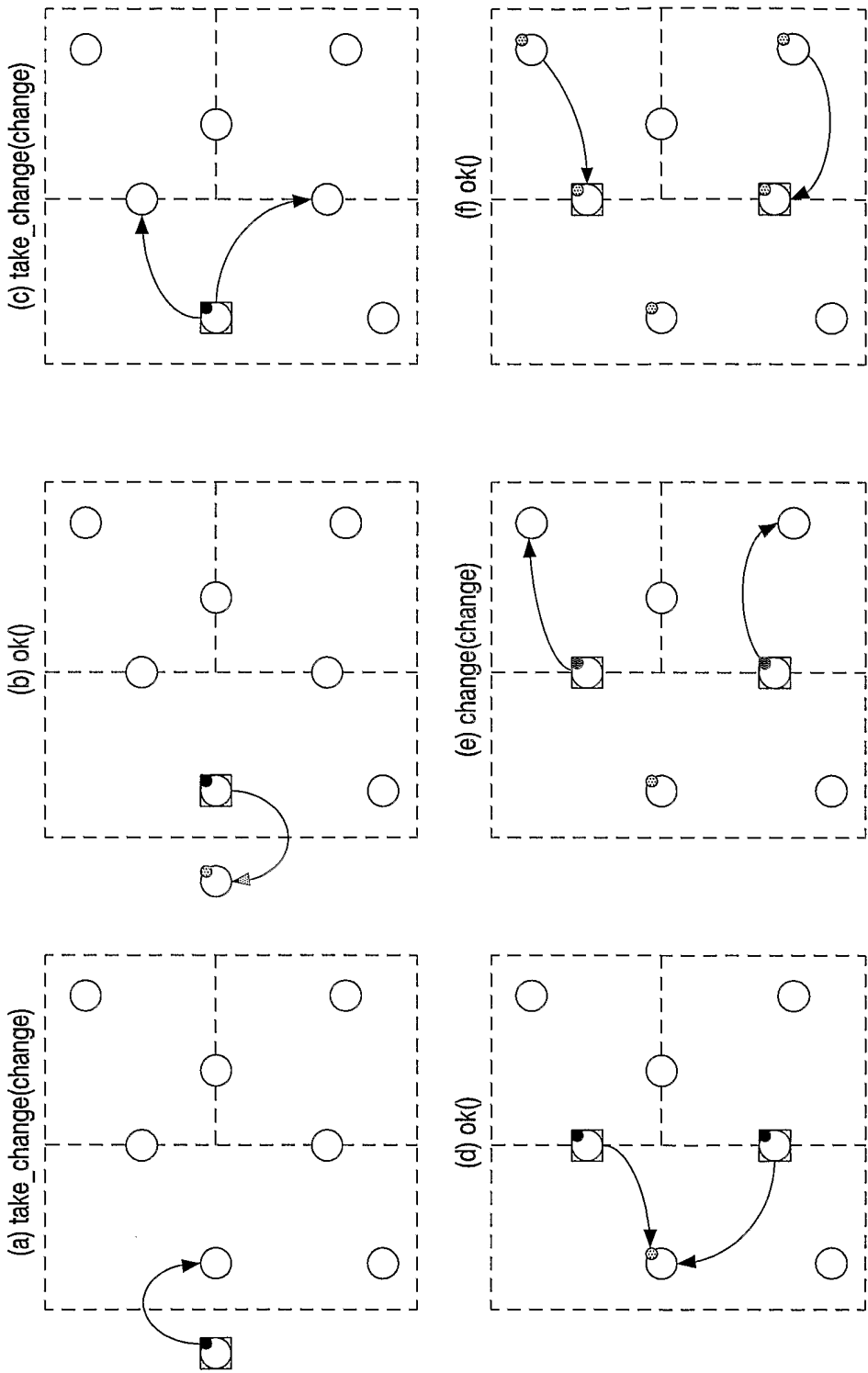


FIG.2

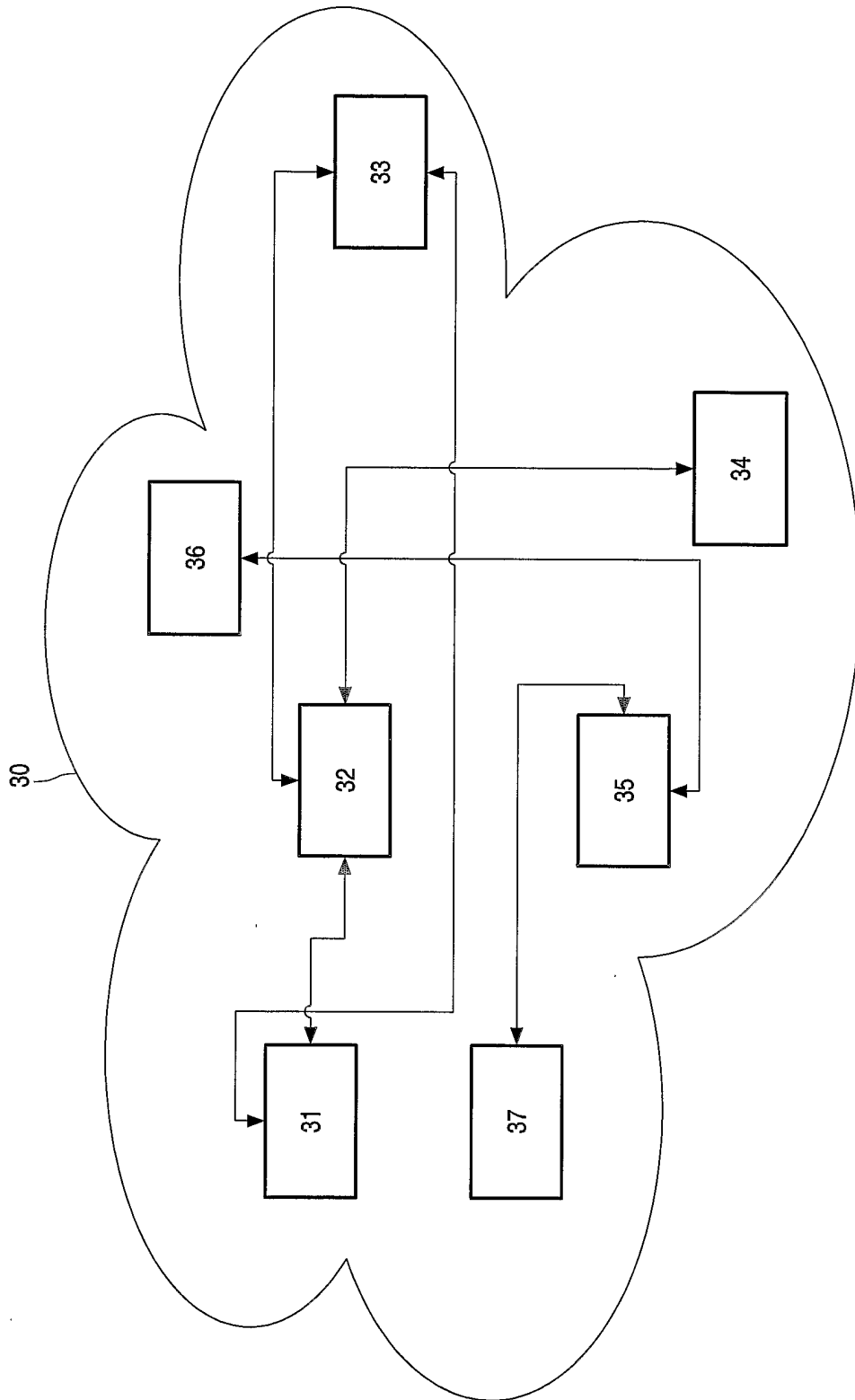


FIG.3

4/5

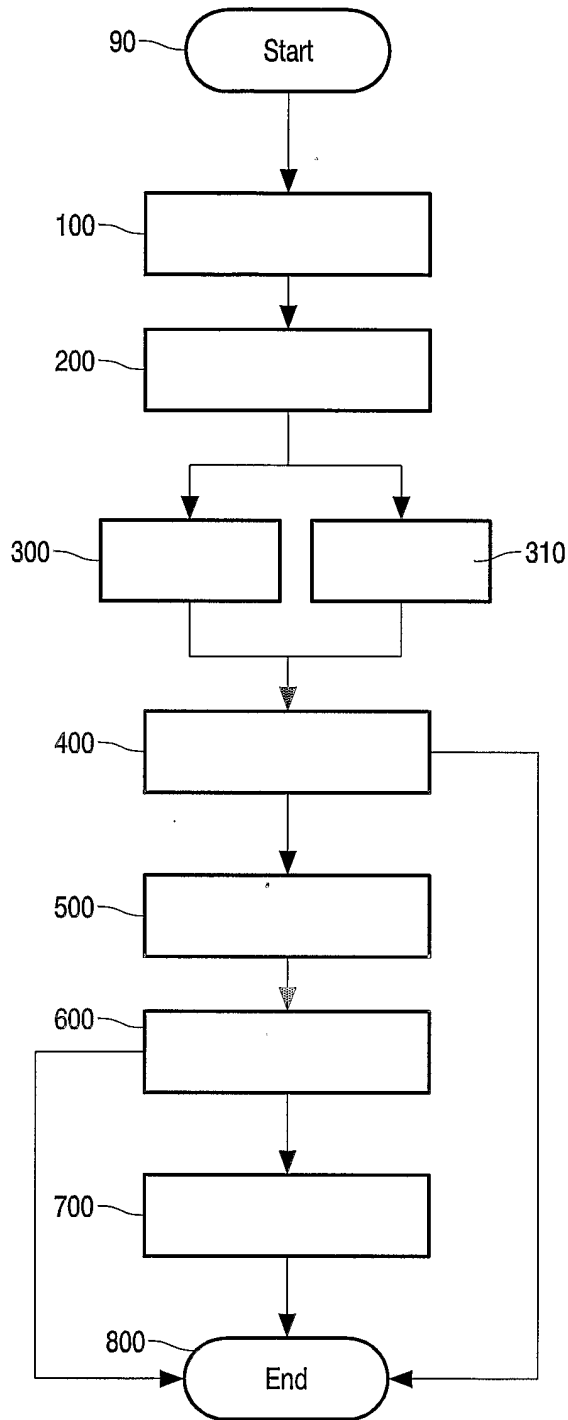


FIG. 4

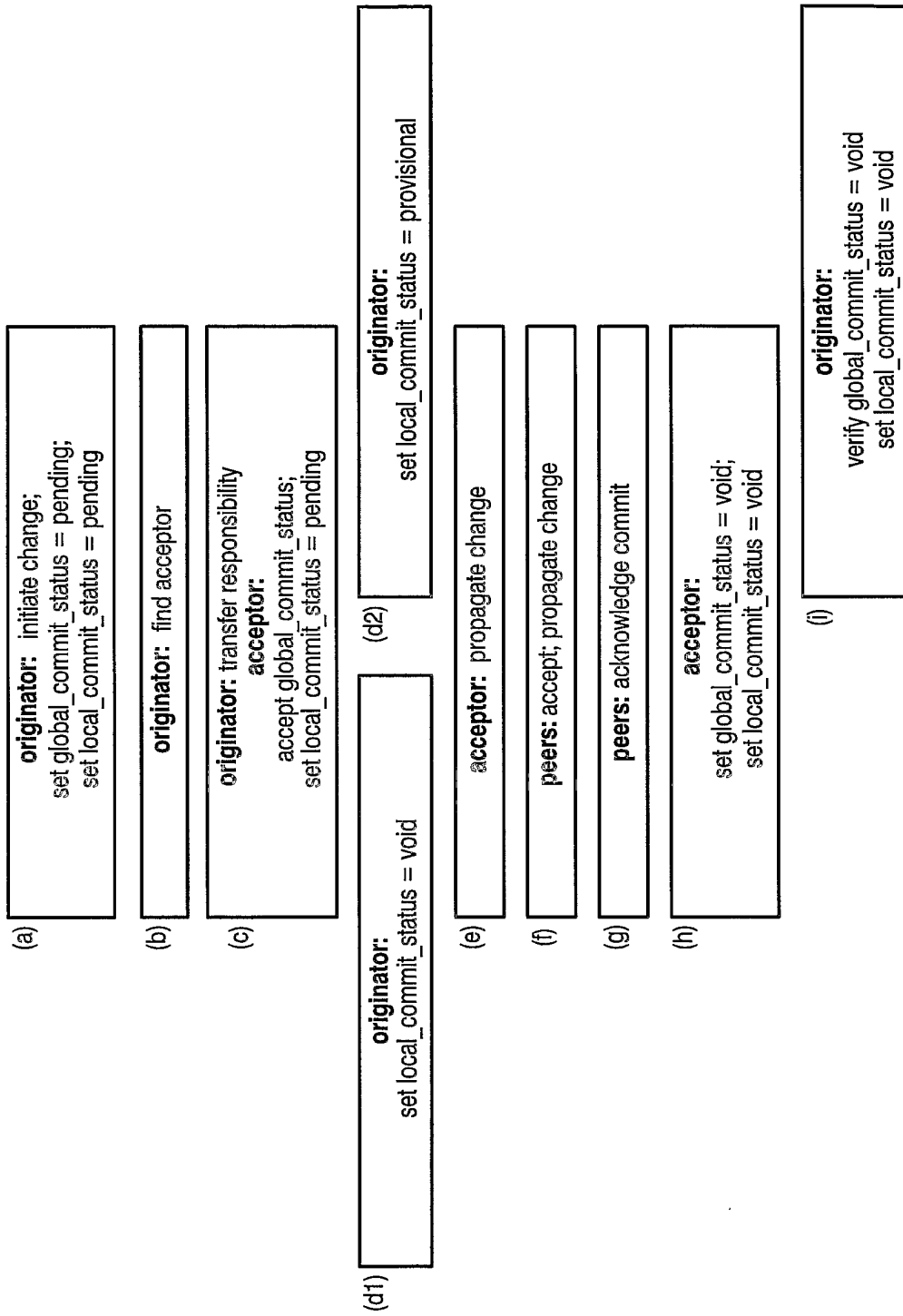


FIG.5

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB2004/050215

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30 H04L29/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 729 733 A (SHARIF-ASKARY JAMSHID) 17 March 1998 (1998-03-17) column 8, line 26 -column 10, line 43 figures 8-12	1-11
Y	US 5 999 931 A (KORTH HENRY F ET AL) 7 December 1999 (1999-12-07) column 5, line 28 -column 10, line 47 column 16, line 62 -column 18, line 11 figures 1-4B	1-11
	--- -/--	

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

° Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

9 July 2004

Date of mailing of the international search report

19/07/2004

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Bengi-Akyuerek, K

INTERNATIONAL SEARCH REPORT

International Application No

PCT/IB2004/050215

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>UEHARA K ET AL: "A framework of customizing transactions in persistent object management for advanced applications"</p> <p>OBJECT-ORIENTATION IN OPERATING SYSTEMS, 1995., FOURTH INTERNATIONAL WORKSHOP ON LUND, SWEDEN 14-15 AUG. 1995, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 14 August 1995 (1995-08-14), pages 84-93, XP010148132</p> <p>ISBN: 0-8186-7115-7</p> <p>page 86, left-hand column, paragraph 2.2</p> <p>-page 87, left-hand column, line 11</p> <p>page 89, left-hand column, paragraph 3.1</p> <p>-page 91, right-hand column, line 9</p> <p>figures 2,4,6</p> <p style="text-align: center;">---</p>	1-11
A	<p>AGRAWAL D ET AL: "Using data migration for heterogeneous databases"</p> <p>INTEROPERABILITY IN MULTIDATABASE SYSTEMS, 1991. IMS '91. PROCEEDINGS., FIRST INTERNATIONAL WORKSHOP ON KYOTO, JAPAN 7-9 APRIL 1991, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 7 April 1991 (1991-04-07), pages 226-229, XP010024617</p> <p>ISBN: 0-8186-2205-9</p> <p>page 227, left-hand column, paragraph 2.2</p> <p>-page 228, left-hand column, line 4</p> <p>figure 1</p> <p style="text-align: center;">-----</p>	1-11

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/IB2004/050215

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5729733	A	17-03-1998	NONE	
US 5999931	A	07-12-1999	NONE	