US 20030172070A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2003/0172070 A1

Sawadsky et al. (43) Pub. Date: Sep. 11, 2003

(54) SYNCHRONOUS PEER-TO-PEER MULTIPOINT DATABASE SYNCHRONIZATION

(76) Inventors: **Nicholas Justin Sawadsky**, Vancouver (CA); **Daniel (Yilin) Shi**, Vancouver (CA); **Andrew Edward Block**, Vancouver (CA); **Michael Anthony Blackstock**, Coquitlam (CA); **Henricus Gerardus Spaay**, Vancouver (CA); **Desiree Paula Rodriguez**, Vancouver (CA)

Correspondence Address:
Robert E. Krebs
BURNS, DOANE, SWECKER & MATHIS, L.L.P.
P.O. Box 1404
Alexandria, VA 22313-1404 (US)

(21) Appl. No.: **10/090,613**

(22) Filed: **Mar. 6, 2002**

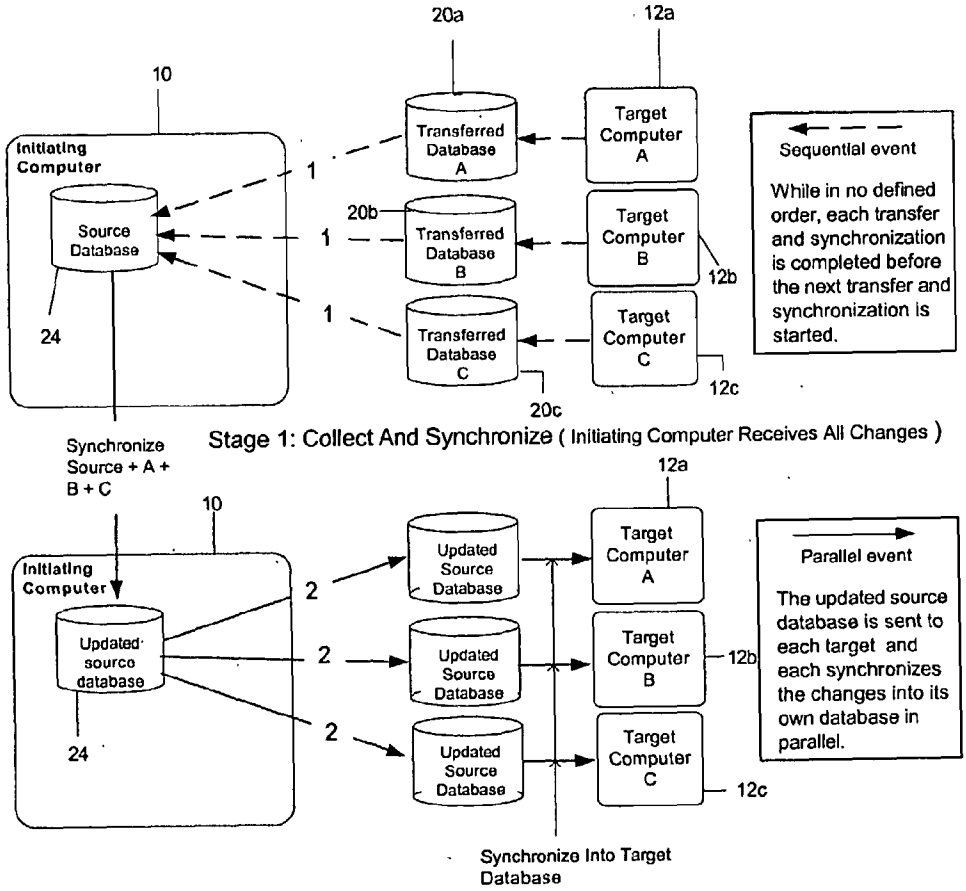
Publication Classification

(51) **Int. Cl.**[7] **G06F 7/00**

(52) **U.S. Cl.** **707/10**

(57) **ABSTRACT**

A method of synchronizing databases between multiple users in a peer-to-peer network is disclosed. The method comprises extracting changes from a source database of one of the users. Next, the changes are compressed and sent to the other users in the peer-to-peer network in parallel. Finally, the changes are decompressed and replicated on the database of each of the other users. In this respect, the databases of the other users will contain all changes from the source database.



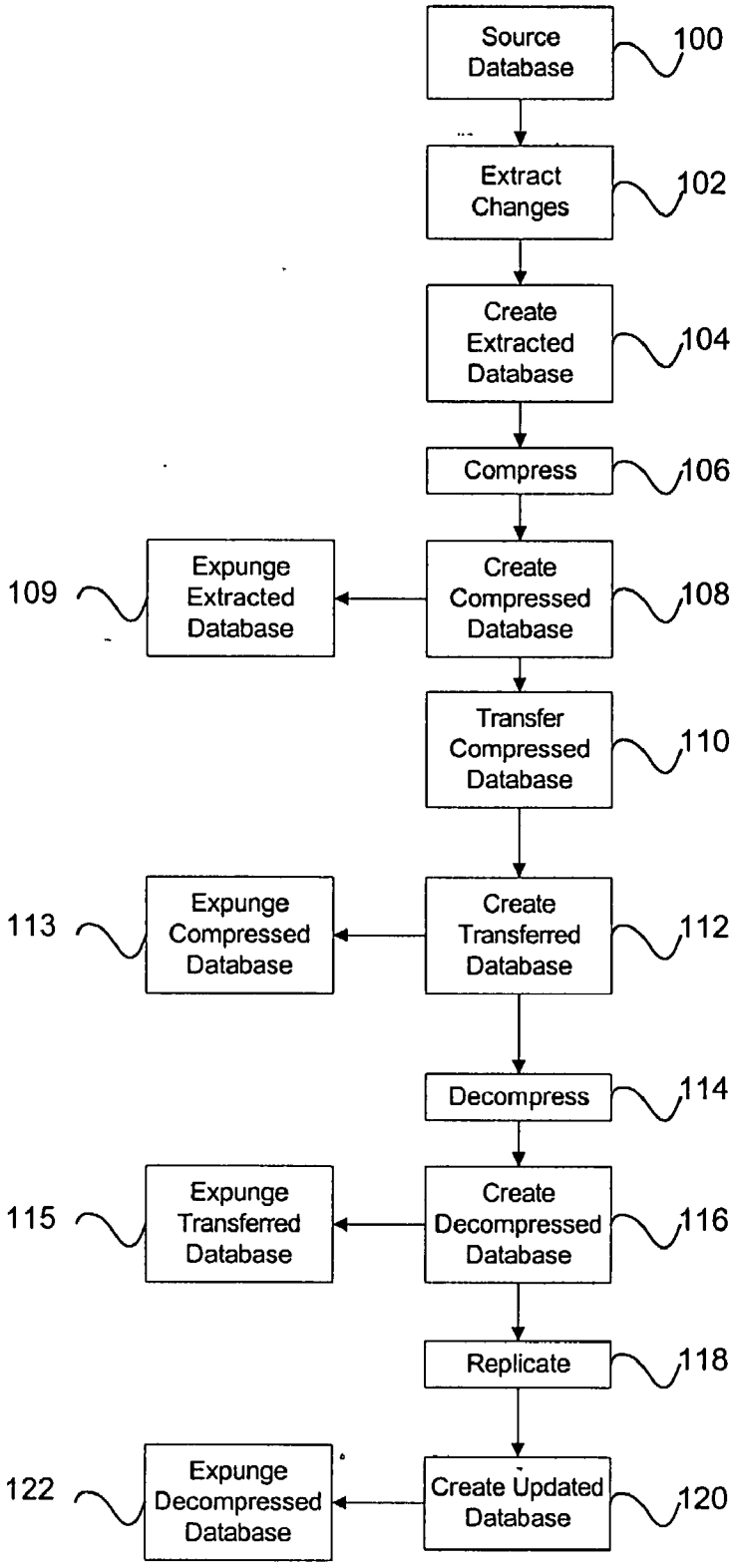

Source Database
100
Extract Changes
102
Create Extracted Database
104
Compress
106
Create Compressed Database
108
Expunge Extracted Database
109
Transfer Compressed Database
110
Create Transferred Database
112
Expunge Compressed Database
113
Decompress
114
Create Decompressed Database
116
Expunge Transferred Database
115
Replicate
118
Create Updated Database
120
Expunge Decompressed Database
122

Figure 1

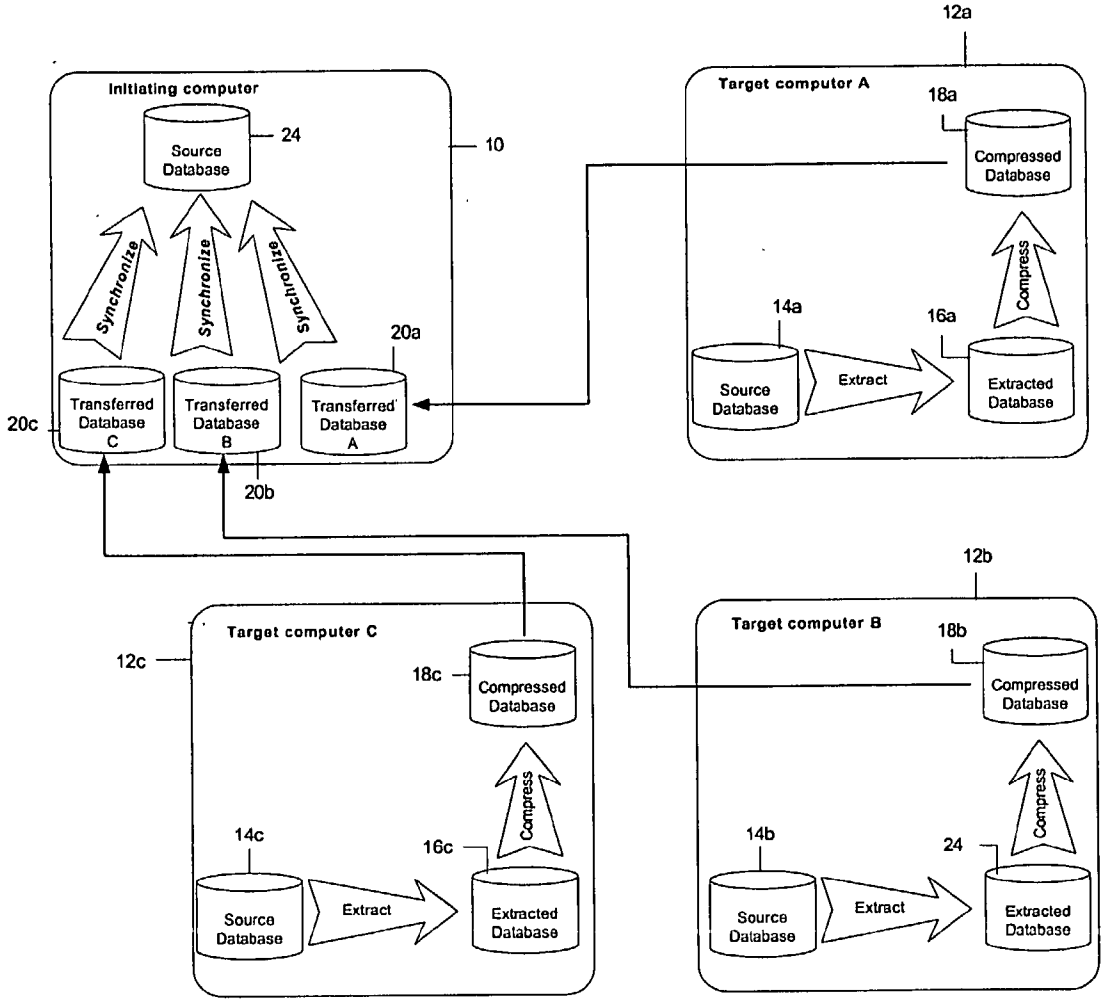

Initiating computer
Source Database
24
10
Synchronize
Synchronize
Synchronize
20a
Transferred Database C
Transferred Database B
Transferred Database A
20c
20b
12a
Target computer A
18a
Compressed Database
Compress
14a
16a
Source Database
Extract
Extracted Database
12b
Target computer B
18b
Compressed Database
Compress
14b
24
Source Database
Extract
Extracted Database
12c
Target computer C
18c
Compressed Database
Compress
14c
16c
Source Database
Extract
Extracted Database

Figure 2

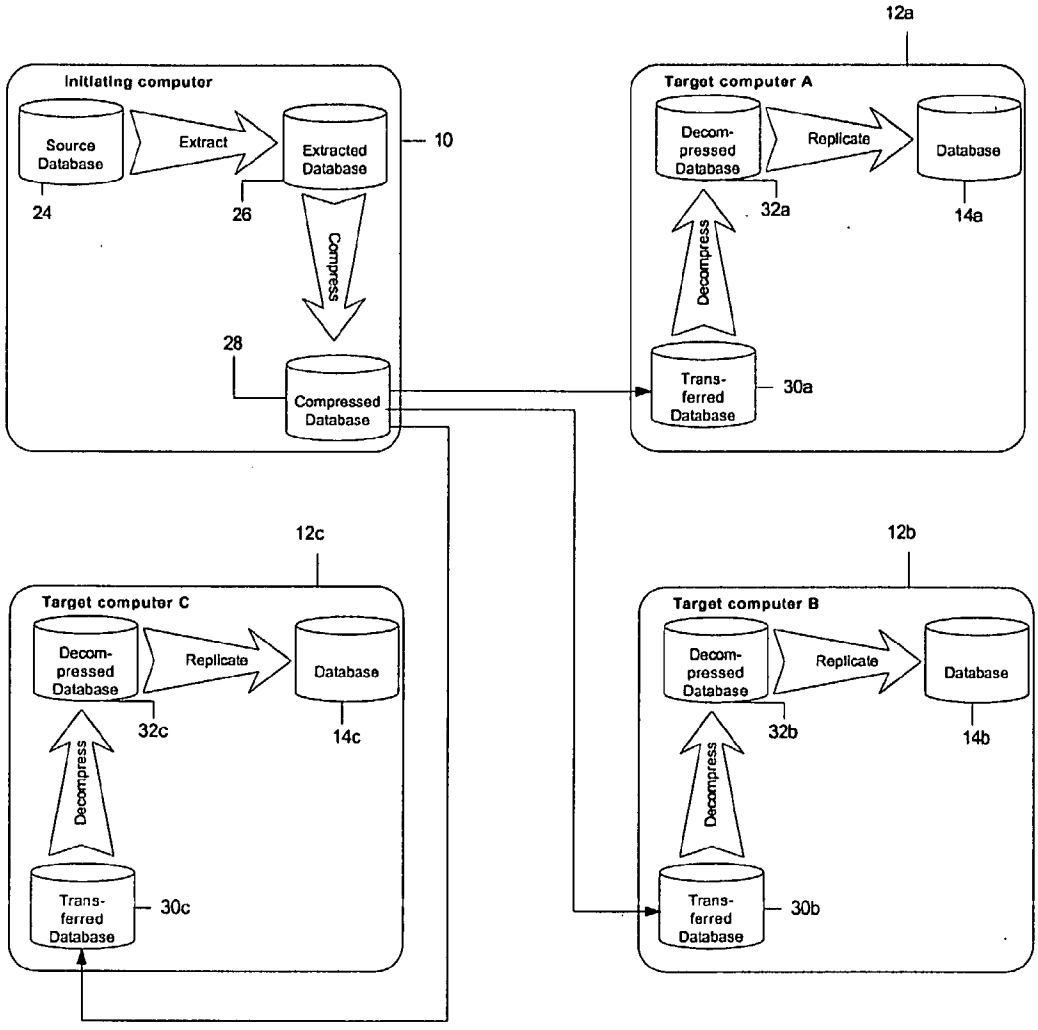

Initiating computor
Source Database
Extract
Extracted Database
10
24
26
Compress
28
Compressed Database
12a
Target computer A
Decom-pressed Database
Replicate
Database
32a
14a
Decompress
Trans-ferred Database
30a
12c
Target computer C
Decom-pressed Database
Replicate
Database
32c
14c
Decompress
Trans-ferred Database
30c
12b
Target computer B
Decom-pressed Database
Replicate
Database
32b
14b
Decompress
Trans-ferred Database
30b

Figure 3

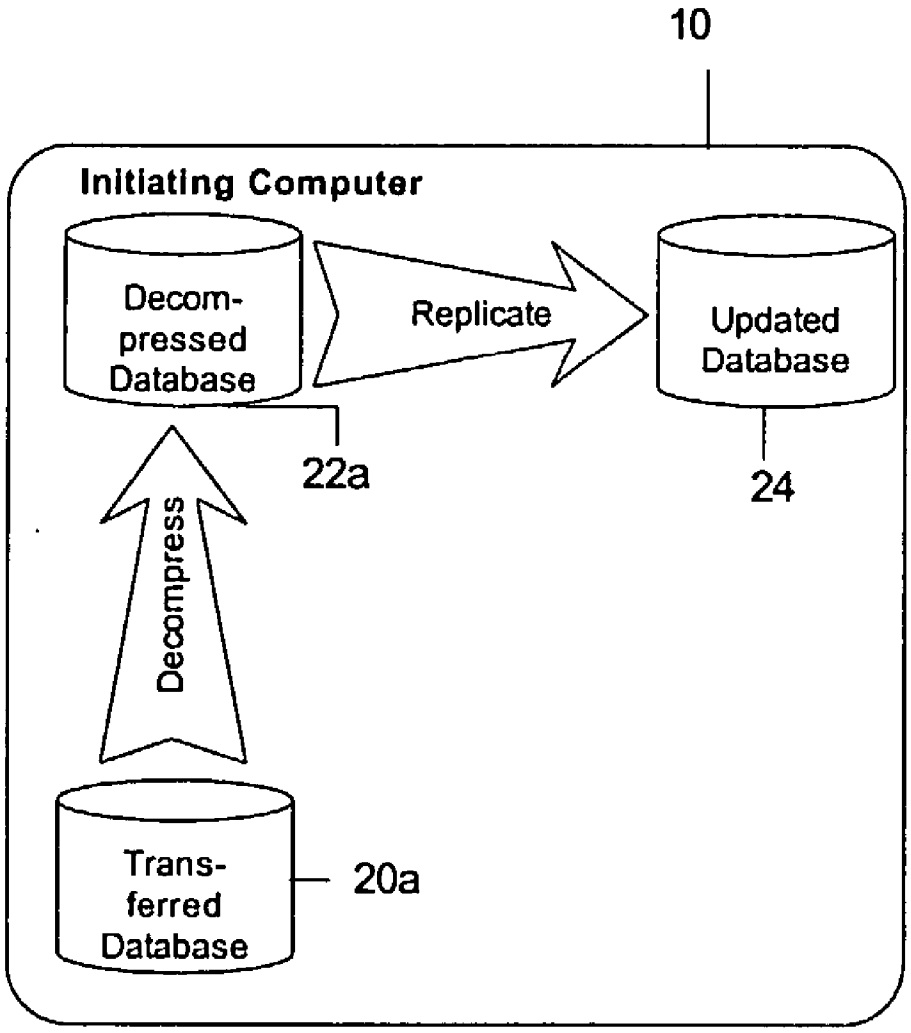
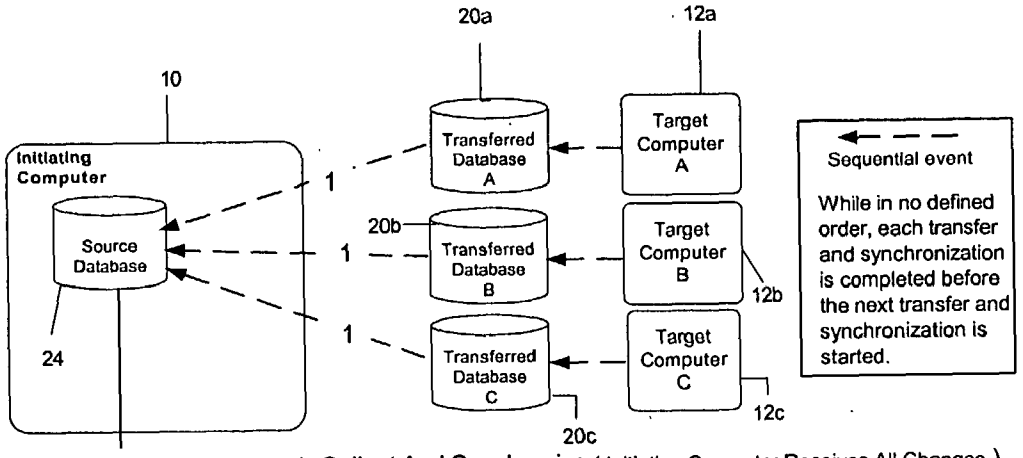
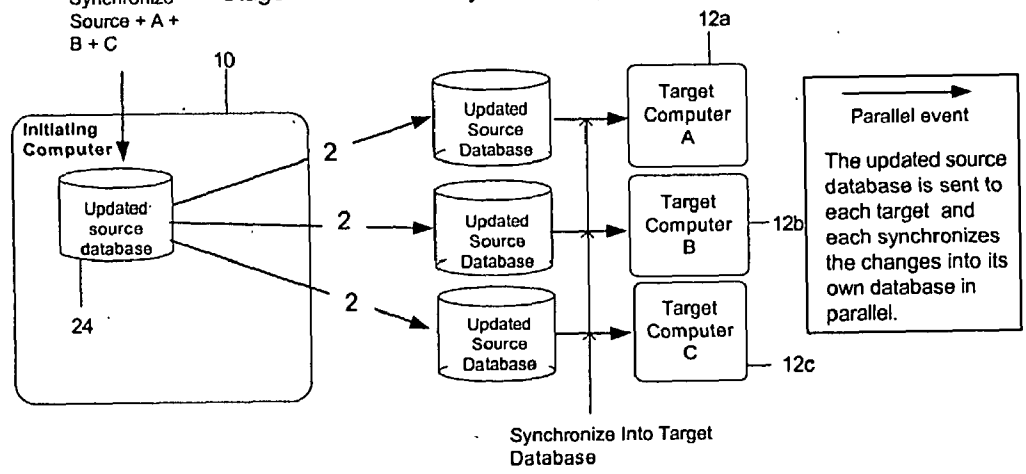

10
Initiating Computer
Decom-
pressed
Database
Replicate
Updated
Database
22a
24
Decompress
Trans-
ferred
Database
20a

Figure 4



Stage 1: Collect And Synchronize (Initiating Computer Receives All Changes)



Stage 2: Distribute New Updated Database To All Targets
(Initiating Computer Sends All Changes)

Figure 5

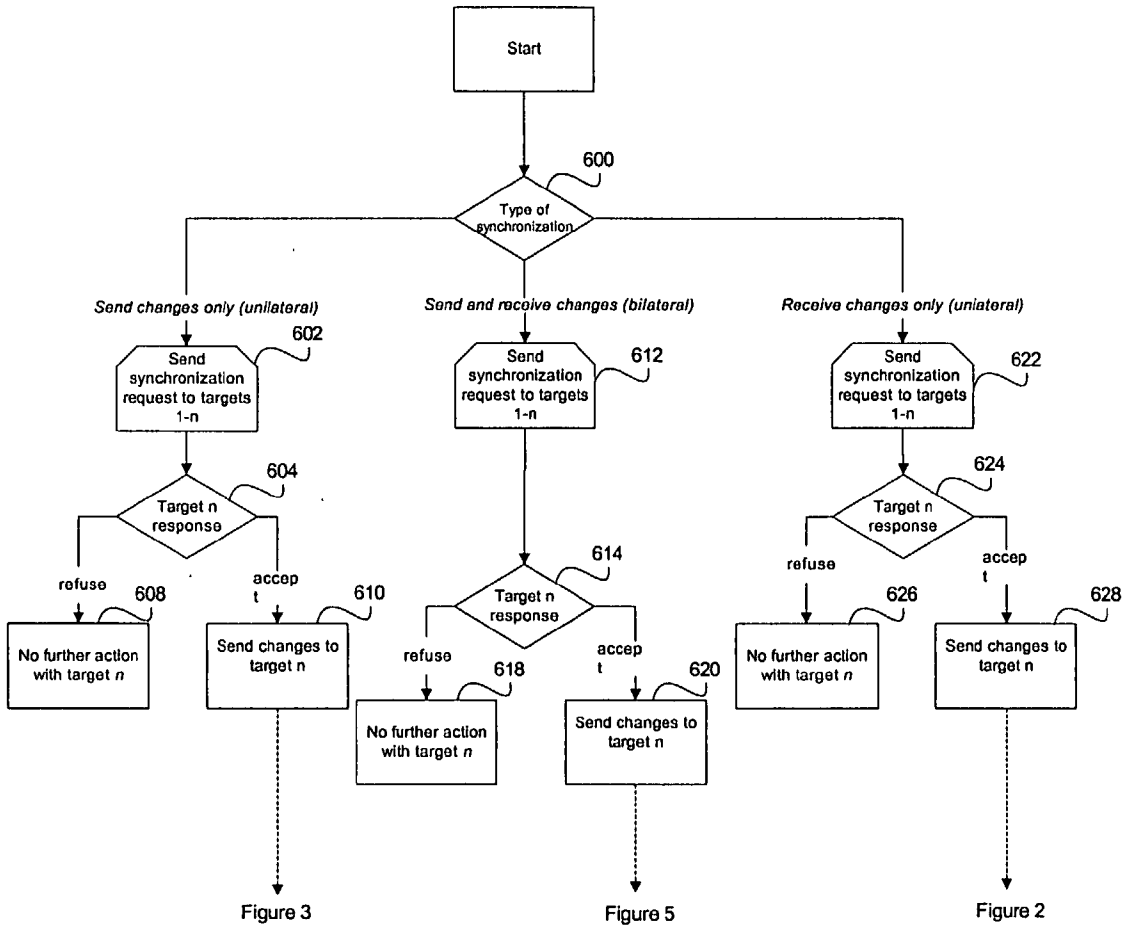

Start
600
Type of synchronization
Send changes only (unilateral)
Send and receive changes (bilateral)
Receive changes only (unilateral)
602
Send synchronization request to targets 1-n
612
Send synchronization request to targets 1-n
622
Send synchronization request to targets 1-n
604
Target n response
614
Target n response
624
Target n response
refuse
accept
608
No further action with target n
610
Send changes to target n
refuse
618
No further action with target n
accept
620
Send changes to target n
refuse
626
No further action with target n
accept
628
Send changes to target n
Figure 3
Figure 5
Figure 2

Figure 6

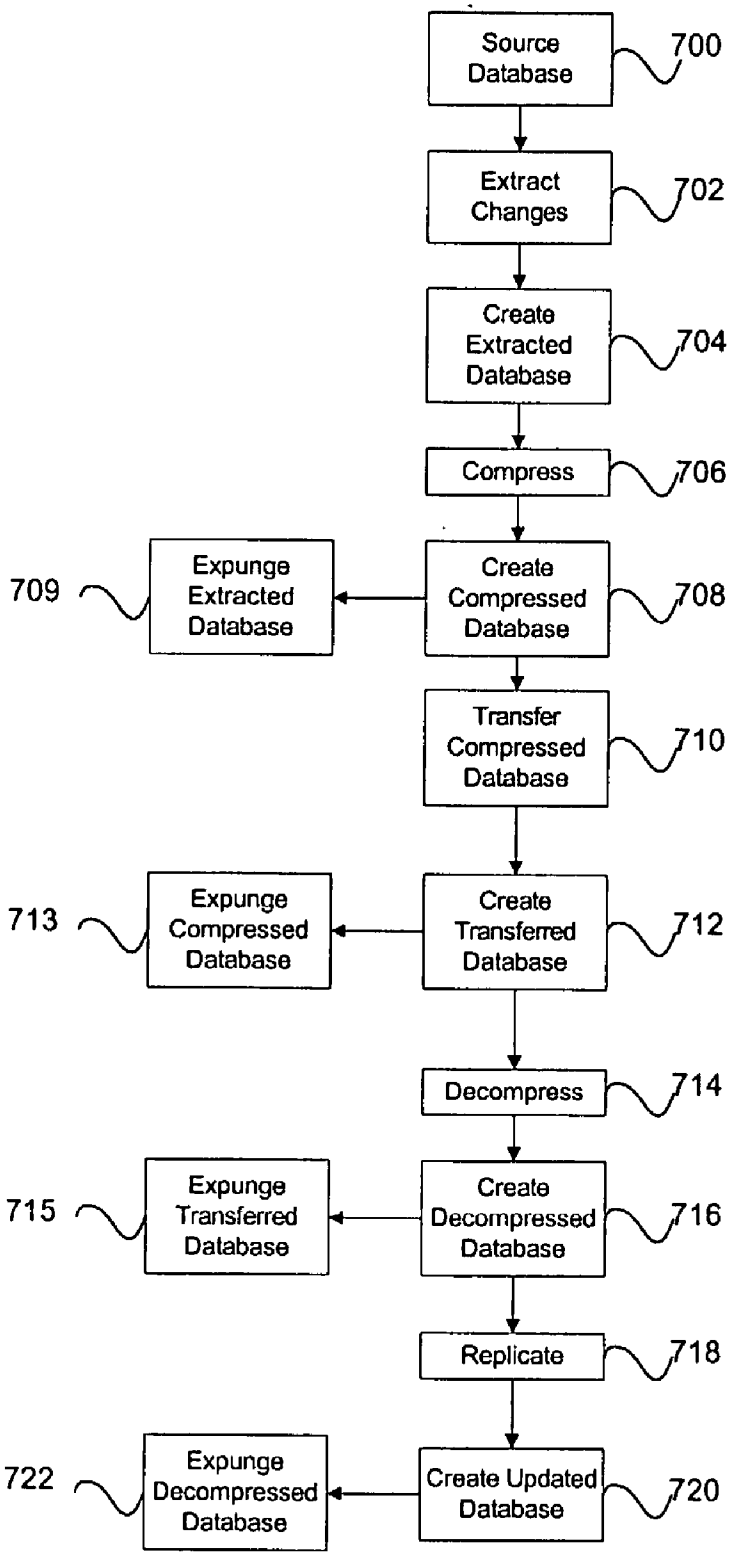

Source Database
700
Extract Changes
702
Create Extracted Database
704
Compress
706
Expunge Extracted Database
709
Create Compressed Database
708
Transfer Compressed Database
710
Expunge Compressed Database
713
Create Transferred Database
712
Decompress
714
Expunge Transferred Database
715
Create Decompressed Database
716
Replicate
718
Expunge Decompressed Database
722
Create Updated Database
720

Figure 7

SYNCHRONOUS PEER-TO-PEER MULTIPOINT DATABASE SYNCHRONIZATION

BACKGROUND OF THE INVENTION

[0001] The present invention generally relates to database synchronization and more particularly to a system and method of database synchronization over a peer-to-peer network.

[0002] With the popularity of handheld computing devices (i.e., PDA's, cell phones, etc. . . .) increasing, there is becoming a greater need and ability to share information between devices. Computing networks can be established between the devices so that collaborative information can be shared. The computing devices can form peer-to-peer networks between one another such that information can be shared without the use of a central server to store a database of information.

[0003] Currently, in the prior art, computers use databases stored on a central database server in order to communicate information. Database systems have replication and synchronization capabilities in order to update information on client systems. These synchronization capabilities are usually restricted to simple two-point exchanges between clients and servers. The synchronization and replication capability require the database server to effect dataset reconciliation between multiple users.

[0004] When multiple users without access to the database server wish to synchronize their databases, it must be done by a series of 2-way exchanges. For instance, information must flow in both directions between users in order to synchronize the dataset. At the very least, this involves 2*n combinations of replications and synchronization. As databases get large, the synchronization and replication procedures between the users becomes tedious and time consuming. Moreover, without the use of a central database server, one user must manage the entire process and ensure that all parties have been included in the synchronization and that data integrity is not lost.

[0005] U.S. Pat. No. 6,295,541, entitled "Systems and Methods for Synchronizing Two or More Datasets" describes a method whereby a reference database is maintained which the other devices synchronize to when available. The patent describes a system whereby devices synchronize to the reference database at different times when they are online such that synchronization occurs serially between devices. Furthermore, only one device is considered the reference database such that only one device controls the synchronization process.

[0006] The present invention addresses the above-mentioned deficiencies in database reconciliation by providing a peer-to-peer method for synchronizing two or more copies of databases without server mediation in real-time. In this respect, the present invention provides a method of synchronization wherein there is no asynchronous storage of interim reference datasets. The present invention provides synchronization in parallel so that several users can simultaneously synchronize individual varying datasets without having to serially exchange and aggregate changes. Furthermore, the present invention allows for any user to initiate synchronization such that one user does not need to be control.

BRIEF SUMMARY OF THE INVENTION

[0007] In accordance with the present invention, there is provided a method of synchronizing databases between multiple users in a peer-to-peer network. The method is initiated by one member of a group of peers. The other group members first extract their changes from their local copy of the database, compress the changes, and send those changes to the initiator. The initiator then decompresses their changes and replicates them into its local database. The initiator then extracts all changes (including those received from other group members) from the local database. Next, the changes are compressed and sent to the other users in the peer-to-peer network. Finally, the changes are decompressed and replicated on the database of each of the other users. In this respect, the databases of each group member will contain all changes from the databases of all the group members.

[0008] The above description describes a bilateral synchronization, in that changes are both sent and received from each peer. A unilateral synchronization is also possible, which consists of either the first half (receiving changes) or the second half (sending changes) of the above description. It will be recognized that the process can be initiated and performed by any user in the peer-to-peer network. In the preferred embodiment of the present invention, the transfer of the changes between the multiple users is done in parallel, although it may possibly be sequential. The transfer can be performed over a wired or wireless network.

[0009] In accordance with the present invention, there is also provided a system for synchronizing databases of multiple users. The system includes a plurality of computers in a peer-to-peer network wherein each computer has a database and software configured to provide synchronization. The software is configured to extract changes from the local database of each of the initiator's peers. The peers' changes are sent to the initiator in parallel as they finish being extracted. Each peer's changes are replicated into the initiator's database. Then all changes (including those received from the peers) are extracted from the initiator's database. Next, the software sends the changes in parallel to the other users of the peer-to-peer network. Finally, the changes are replicated by the software on each database of the other users in order to synchronize the databases.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] These as well as other features of the present invention will become more apparent upon reference to the drawings wherein:

[0011] **FIG. 1** is a flowchart illustrating a method of unilateral database synchronization from a target computer to an initiating computer according to the present invention;

[0012] **FIG. 2** is a diagram illustrating unilateral database replication from multiple target computers to the initiating computer;

[0013] **FIG. 3** is a diagram illustrating unilateral database replication from the initiating computer to multiple target computers;

[0014] **FIG. 4** illustrates the synchronization process on the initiating computer;

[0015] **FIG. 5** is a diagram illustrating bilateral database replication between target computers and the initiating computer;

[0016] **FIG. 6** is a diagram illustrating the selection of database replication techniques by a user; and

[0017] **FIG. 7** is a flowchart illustrating a method of unilateral database synchronization from the initiating computer to the target computer.

DETAILED DESCRIPTION OF THE INVENTION

[0018] Referring now to the drawings wherein the showings are for purposes of illustrating a preferred embodiment of the present invention only, and not for purposes of limiting the same, **FIG. 2** shows unilateral database replication between an initiating computer **10** and target computers **12***a*, **12***b* and **12***c* according to the present invention. Both the initiating computer **10** and the target computers **12***a*, **12***b*, and **12***c* are computing devices having a memory and capable of storing and processing information. For instance the computing devices may be PDA's (i.e., Personal Digital Assistants) which are in electronic communication with one another through a peer-to-peer network. The network may be a wireless network such as Bluetooth, or could be a wired network such as an Ethernet. In either case, the peer-to-peer network provides electronic communication between the initiating computer **10** and the target computers **12** without the use of a server.

[0019] The initiating (source) computer **10** is synchronized with the other computers **12** by the transfer of database information from the target computers **12** to the initiating computer **10**. As seen in **FIG. 2**, each of the target computers **12** transfer information to the initiating computer **10** in the same manner. Accordingly, the description below will be for the transfer of database information from target computer **12***a* to source computer **10**, yet the process is identical for the transfers between target computers **12***b* and **12***c* to source computer **10**. As will be further explained below, the processing and transfers may occur in parallel for each of the target computers **12***a*, **12***b*, and **12***c* to the initiating computer **10**.

[0020] The process for synchronizing the target computers **12** to the initiating computer **10** begins by the initiating computer **10** sending out a synchronization request to the target computers **12**. The synchronization command informs the target computers that the initiating computer **10** wishes to synchronize databases with them. It will be recognized that any computer in the network may be the initiating computer such that it is possible for any computer to start the synchronization process. The term initiating computer designates the computer that wishes the synchronization process to begin.

[0021] Referring to **FIGS. 1 and 2**, in step **100**, the target computer **12***a* has a source database **14***a* that is created in memory (i.e., hardrive, RAM, etc. . . .). The source database **14***a* contains information which is shared between the target computer **12***a* and the other computing devices (i.e. computers **12***b*, **12***c* and **10**). Next, in step **102**, any changes to the source database **14***a* are extracted to create a temporary extracted database **16***a* containing at least all relevant changes to both the target computer **12***a* and the initiating computer **10**. The extracted database **16***a* corresponds to the replica source **14***a* stored on the target computer **12***a*. All changes to the source database **14***a* from a user-specified date and time, or from the last time a synchronization occurred, are extracted and copied to the extracted database **16***a*. Accordingly, as seen in step **104**, the extracted database **16***a* is created by extracting and copying the changes to the source database **14***a*. The extraction and copying is performed using the database API of the target computer **12***a* if such an API is available. In the case where no such API is available, changes are tracked as data is written to each database and extracted by the synchronization system itself.

[0022] Next, the extracted database **16***a* is compressed in step **106** to create a compressed database **18***a* in step **108**. The compressed database **18***a* is created by using well known compression techniques on the extracted database **16***a*. Once the compressed database **18** is created, then the extracted database **16***a* is expunged in step **109**.

[0023] Referring to step **110**, the compressed database **18***a* is transferred to the initiating computer **10**. Specifically, the file of the compressed database **18***a* is transferred to the initiating computer **10** in response to a request using any well known file transfer technique over any type of network, as previously described. Each of the target computers **12** will transfer information to the initiating computer **10** in parallel or in sequence when it is not possible to perform parallel communications. Typically, the compressed database **18***a* is transferred to a temporary transferred database **20***a* that is created on the initiating computer **10** in step **112**. Once the transfer is complete, in step **113**, the compressed database **18***a* on the target computer **12***a* is expunged.

[0024] Once the transferred database **20***a* has been created on the initiating computer **10**, the transferred database **20***a* is synchronized with the source database **24** of the initiating computer **10**. Specifically, in step **114** of **FIG. 1**, the transferred database **20***a* is decompressed on the initiating computer **10**, as seen in **FIG. 4**. The transferred database **20***a* is decompressed using a complementary decompression technique to that of step **106**. Accordingly, in step **116**, a decompressed database **22***a* is created on the initiating computer **10**. Once the decompressed database **22***a* is created, then the transferred database **20***a* is expunged in step **115**.

[0025] The decompressed database **22***a* is then replicated with the source database **24** of the initiating computer **10** in step **118**. This operation is performed using the database API of the initiating computer **10** if such an API is available. In the case where no such API is available, replication is performed by the synchronization system itself. The process of replication causes the changes copied in the extracted database **16***a* to be incorporated into the source database **24** of the initiating computer **10**. Once the decompressed database **22***a* is replicated onto the source database **24**, the updated source database **24** is created in step **120** which has the changes and is identical to the source database **14***a*. Finally, the decompressed database **22***a* is expunged in step **122**.

[0026] Each time a target computer **12** sends changes to the initiating computer **10**, the synchronization process previously described (i.e., transfer, compression, decompression and replication) are completed in full before another synchronization from another target computer **12** is processed.

[0027] The above-described procedure is operative to send changes from the target computers **12** to an initiating com-

puter **10**. This procedure typically occurs when an initializing user wishes to receive changes from the other computers. However, the procedure can also be used if the initializing user wishes to transfer changes to other computers. In that instance, changes from the database of the initiating computer **10** would be transferred to the other computers **12***a*, **12***b*, and **12***c*.

[**0028**] Referring to **FIG. 3, a** diagram for the unilateral synchronization of multiple computers from the initiating computer **10** to target computers **12** is shown. Synchronization between the initiating computer **10** and target computers **12** occurs in parallel. Each target computer **12** has a transferred database, a decompressed database, and a target database that are created during the process of synchronization.

[**0029**] Referring to **FIGS. 3 and 7**, the process for synchronizing the target computers **12** to the initiating computer **10** is similar to the process of synchronization described in **FIG. 1**. For example, the process begins with the initiating computer **10** sending a synchronization request to the target computers **12***a*, **12***b*, and **12***c* informing them that the initiating computer **10** wishes to send them changes to their databases. The synchronization process proceeds according to **FIG. 7** such that changes in the source database **24** of the initiating computer **10** are extracted in step **702** to create an extracted database **26** in step **704**. In step **706**, the extracted database **26** is compressed to create a compressed database **28** in step **708**. Once the compressed database **28** is created, the extracted database **26** is expunged in step **709**. The compressed database **28** is transferred to the target computers **12***a*, **12***b*, and **12***c* in parallel over the computer network.

[**0030**] Each of the target computers **12***a*, **12***b*, and **12***c* receives the compressed database **28** and creates a respective transferred database **30***a*, **30***b*, and **30***c*. For simplicity, the following description will be for target computer **12***a*. However, it will be recognized that the following synchronization process occurs in any of the target computers (i.e., **12***b*, and **12***c*) synchronizing to the initiating computer **10**. After the compressed database **28** is transferred and the transferred database **30***a* is created on the target computer **12***a*, then the compressed database **28** on the initiating computer **10** is expunged in step **713**. In step **714**, the compressed database **30***a* on the target computer **12***a* is decompressed to create a decompressed database **32***a* in step **716**. The transferred database **30***a* is then expunged in step **715**. The changes from the source database **24** are then replicated onto the database **14***a* of target computer **12***a* in step **718**. In this regard, the database **14***a* will be updated with the changes from initiating computer **10**, as seen in step **720**. Finally, the decompressed database **32***a* is expunged from the target computer **12***a*.

[**0031**] The above-described method is concurrently performed on each of the target computers **12***a*, **12***b*, and **12***c* such that synchronization occurs simultaneously. Accordingly, the synchronization process of the present invention is a parallel method whereby each of the target computers **12** can synchronize with the initiating computer **10** quickly.

[**0032**] A bilateral database replication between an initiator and n peers consists of first n unilateral replications conducted in parallel to replicate all of the peers' changes into the initiator's database, followed by another n unilateral replications, again conducted in parallel, to replicate the accumulated changes from the initiator's database into the peers' local databases. Referring to **FIG. 5**, the initiating computer **10** receives changes from each of the target computers **12** in a sequential manner through the process described for **FIG. 2**. While in no defined order, each transfer and synchronization is completed before the next transfer and synchronization is started. For example, the synchronization process from target computer **12***a* to the transferred database **20***a* and synchronization to source database **24** will be completed before the next synchronization process from either target computer **12***b* or **12***c*. Once all of the target computers **12** have synchronized their changes to the source database **24**, then the changes will be sent back to all of the target computers **12** in order to fully synchronize each target computer **12** to one another. Specifically, the source database **24** is now updated with the changes from each of the target computers **12**. The initiating computer **10** will send all of its changes to the to the target computers **12** through a parallel manner as described for **FIG. 3**. The updated source database **24** is sent to each target computer **12** and each target computer **12** synchronizes the changes into it own database **14** in parallel. Accordingly, complete synchronization of all databases (i.e., initiating computer **10** and target computers **12**) occurs in a two-stage process (bi-lateral exchange).

[**0033**] Even though the above-mentioned synchronization process for multiple computers has been described as being between an initiating and target computers, it will be recognized that any computer in the peer-to-peer network can begin the synchronization process. The computer that initiates the process will push and pull the transfer of files as needed. Furthermore, during multi-user synchronization, if any computer (other than the initiator) disappears from the network or cancels the process, the synchronization process can still proceed between the other computers.

[**0034**] Referring to **FIG. 6, a** flowchart showing possible synchronization processes is shown. As previously mentioned, either bilateral or unilateral synchronization between the initiating computer **10** and the target computers **12** can occur. The user determines the type of synchronization process in step **600** of **FIG. 6**. The user can send changes from the initiating computer as described for **FIG. 3**, receive and send changes as described for **FIG. 5**, or receive changes as described for **FIG. 2**. If the user decides to send changes only from the initiating computer **10** to target computers **12** (i.e., unilateral exchange), then in step **602**, synchronization requests are sent from the initiating computer **10** to 1–n target computers **12**. Next, each target computer **12** will respond to the request in step **604**. Specifically, a target computer **12** can refuse the request such that synchronization does not occur. If the target computer **12** refuses, then in step **608**, there is no further action with those target computers **12** which refuse synchronization. However, in step **610**, target computers **12** which accept the synchronization request proceed to receive changes as previously described for **FIG. 3**.

[**0035**] If the user wishes to send and receive changes (i.e., indicating a bilateral exchange), then the initiating computer **10** will send the synchronization request to the 1–n target computers **12** as shown in step **612**. Next, each of the target computers **12** will respond to the request by either refusing or accepting the request. If a target computer **12** refuses the request, then in step **618** no further action is taken with that

target computer **12**. However, if the target computer **12** accepts the request, then the bilateral synchronization process shown in **FIG. 5** commences and the target computer **12** will begin sending changes to the initiating computer **10** and then proceed to receive all changes therefrom, as shown in step **620**.

[**0036**] Referring to **FIG. 6**, if the user wishes just to receive changes only (i.e., unilateral exchange), then the initiating computer **10** will send the synchronization request to the target computers **12** in step **622**. Next, the target computers **12** will respond in step **624**. If a target computer **12** refuses to send changes to the initiating computer **10**, then no further action will occur, as shown in step **626**. However, if the target computer **12** agrees to the synchronization process, then in step **628**, the changes are sent to the initiating computer as described for **FIG. 2**.

[**0037**] Additional modifications and improvements of the present invention may also be apparent to those of ordinary skill in the art. Thus, the particular combination of parts described and illustrated herein is intended to represent only a certain embodiment of the present invention, and not intended to serve as a limitation of alternative devices within the spirit and scope of the invention.

1. A method for peer-to-peer database synchronization between a first computer and a second computer, the method comprising the steps of:

a) extracting changes from a source database of the first computer to generate an extracted database;

b) transferring the extracted database from the first computer to the second computer; and

c) replicating the source database on a target database of the second computer from the extracted database in order to synchronize the target database with the source database.

2. The method of claim 1 further comprising the steps of:

1) compressing the extracted database to generate a compressed extracted database subsequent to step (a); and

2) decompressing the compressed extracted database on the second computer to generate a decompressed extracted database subsequent to step (b).

3. The method of claim 2 further comprising the step of expunging the extracted database from the first computer after the compressed extracted database is generated.

4. The method of claim 2 further comprising the step of expunging the compressed extracted database from the first computer after the transferring from the first computer to the second computer.

5. The method of claim 2 further comprising the step of expunging the extracted database on the second computer after the decompressed database has been generated.

6. The method of claim 2 further comprising the step of expunging the decompressed database on the second computer after it has been replicated on the target database.

7. The method of claim 2 wherein in step (b) the compressed extracted database is transferred over a peer-to-peer network between the first computer and the second computer.

8. The method of claim 7 wherein the peer-to-peer network is a wireless network.

9. The method of claim 7 wherein the peer-to-peer network is a wired network.

10. The method of claim 2 wherein in step (2) the compressed extracted database is decompressed in a manner complementary to the compression of step (1).

11. The method of claim 1 further comprising the steps of:

d) extracting changes from the target database of the target computer to generate an extracted target database;

e) transferring the extracted target database from the second computer to the first computer; and

f) replicating the target database on the source database of the first computer from the extracted target database in order to synchronize the source database with the target database.

12. The method of claim 11 further comprising the steps of:

1) compressing the extracted target database to generate a compressed extracted target database subsequent to step (d); and

2) decompressing the compressed extracted target database on the first computer to generate a decompressed target database subsequent to step (e).

13. A method for synchronizing a database of a first computer with a database of a second computer, the method comprising the steps of extracting changes from the database of the first computer, transferring the changes to the second computer and replicating the changes on the database of the second computer such that the databases on the first computer and the second computer are synchronized.

14. The method of claim 13 further comprising the step of compressing the changes prior to transferring from the first computer to the second computer.

15. The method of claim 13 wherein the changes are transferred over a peer-to-peer network.

16. The method of claim 15 wherein the peer-to-peer network is a wireless network.

17. The method of claim 15 wherein the peer-to-peer network is a wired network.

18. A method for synchronizing databases of multiple users in a peer-to-peer network wherein one of the users is designated an initiating user, the method comprising the steps of:

a) extracting changes from a respective database of each of the users;

b) sending the changes from each of the users to the initiating user;

c) replicating the changes from each of the users onto the database of the initiating user;

d) sending the changes on the database of the initiating user to each of the other users; and

e) replicating the changes on a respective database of each user in order to synchronize the databases of all of the users.

19. The method of claim 18 further comprising compressing the changes prior to sending them and decompressing the changes after being received.

20. The method of claim 18 wherein:

step (a) further comprises creating a transferred database from the changes to each respective database of the users;

step (b) further comprises compressing and sending the transferred database as the changes to the initiating user; and

step (c) further comprises decompressing the transferred database in order to replicate the changes to the database of the initiating user.

21. The method of claim 18 wherein:

step (d) further comprises creating a transferred database from the changes to the database of the initiating user and then compressing and sending the transferred database as the changes to each of the databases of each of the users; and

step (e) further comprises decompressing the transferred database by each of the users in order to replicate the changes of all the users on each of the user's databases.

22. The method of claim 18 wherein in step (d) the changes are transferred in parallel to each of the users.

23. A system for synchronizing databases of multiple users in a peer-to-peer network, the system comprising:

a plurality of computers in a peer-to-peer network, wherein one of the computers is designated an initiating computer, each computer having a database and software configured to:

a) extract changes from a source database of each computer of the users;

b) send changes from each of the user's computers in the peer-to-peer network to the initiating computer; and

c) replicate the changes from each of the databases of the user's computers onto the database of the initiating computer;

d) extract all of the changes from the database of the initiating computer;

e) send the changes to each of the user's computers in the peer-to-peer network; and

f) replicate the changes on a respective database of each of the users in order to synchronize all databases.

24. The system of claim 23 wherein the software is configured to send the changes to the other users in parallel.

* * * * *