(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 October 2010 (14.10.2010)

PCT

(10) International Publication Number
WO 2010/115446 Al

(51) International Patent Classification:
G06Q 10/00 (2006.01) ***H04L 29/06*** (2006.01)

(21) International Application Number:
PCT/EP2009/002774

(22) International Filing Date:
8; April 2009 (08.04.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US):* **FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V.;** Hansastr. 27c, 80686 München (DE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only):* **PFEFFER, Heiko** [DE/DE]; Urbanstrasse 130, 10967 Berlin (DE). **LINNER, David** [DE/DE]; Dunckerstrasse 22, 10437 Berlin (DE). STEGLICH, **Stephan** [DE/DE]; Oldenburger Strasse 22, 1055 1 Berlin (DE).

(74) Agent: GROSS, **Felix;** Maikowski & Ninnemann, Postfach 15 09 20, 10671 Berlin (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available):* ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, 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, HR, HU, IE, IS, **IT,** LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

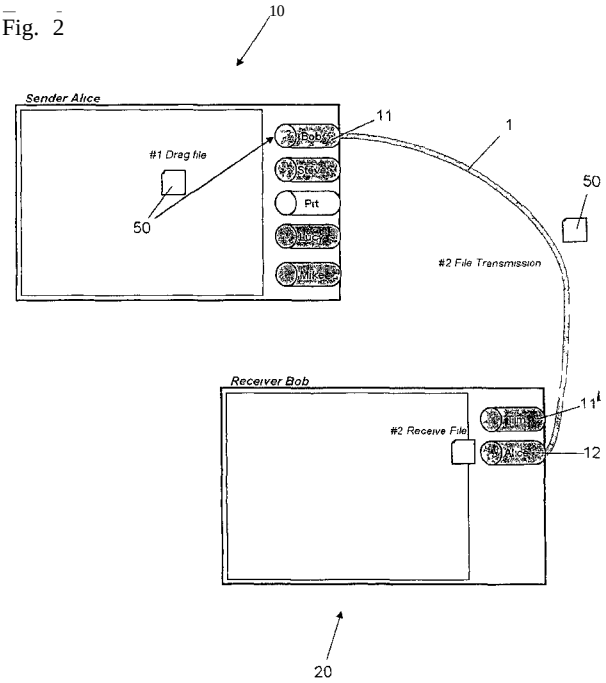
Published:

— *with international search report (Art. 21(3))*

(54) Title: METHOD AND SYSTEM FOR FILE TRANSFER BETWEEN COMPUTER SYSTEMS

(57) Abstract: Method and system for transferring at least one data file (50) from at least one first computer system (10) to at least one second computer sytem (20), wherein a) at least one data transfer link (1) is established between the computer systems (10, 20) b) the at least one data file (50) is sent or received via the at least one data transfer link (1) by associating the at least one data file (50) with at least one end (11, 12, 13, 14, 15, 11') of the at least one data transfer link (1) and c) the at least one file comprises at least one annotation tag (60).

Fig. 2



WO 2010/115446 A1

Method and system for file transfer between computer systems

The invention relates to a method for file transfer according to claim 1 and a system for file transfer accroding to claim 16.

The transfer of a file from one computer system, such as an PC to another computer system require often additional hardware such as USB memory sticks, CDs/DVDs or require complex software. Computer systems in this context mean any system which can handle an file exchange. Therefore, e.g. digital cameras, mobile phones, PDAs, and modern audio-video installations are to be considered as computer systems since they require and / or possess the capability of file exchange.

In the file exchange between PCs Email clients are commonly used to transfer files as an attachment although the main feature of Email is the transmission of text messages from one person to another. Limits in the receiver's mail inbox often forbid the transmission of large files. Other examples for file transfer are FTP clients, which were designed to transfer files to a servers. A FTP server maybe set up on a computer system; however, this proceeding is considered as a technology for more advanced users. Another example are messengers such as Microsoft Messenger or Skype, providing the exchange of data files. However, the connections are slow and the programs were not intentionally built for file transfer but for chatting and Voice-over- IP.

Therefore the requirement of an improved file transfer exists. File transfer in this context also implies the transfer of a media stream, e.g. the tranfer of a large video file which can be viewed before the transfer of the whole file is complete.

The method according to claim 1 and the system according to claim 14 provides such a solution.

In the following different embodiments of the method and systems are described in an exemplary way.

Fig. 1 schematically shows a computer desktop with multiple data transfer links, i.e. BitTubes;

Fig. 2 schematically shows a connection between two computer sytems using a BitTube (Online Transmission mode) ;

Fig. 3 schematically shows a file transfer to a user who is currently offline;

Fig. 4 schematically shows a group file transfer;

Fig. 5 schematically shows an embodiment with a concept of an annotation tag;

Fig. 6 schematically shows a flow chart for ensuring the consistency of a single file when it is edited by multiple users simultaneously;

Fig. 7 schematically shows a file and locking sequence scheme for a file transfer between two users or computer systems;

Fig. 9 showing a flowchart for an embodiment of the assigning of a new ownership for tasks;

Fig. 8 showing a flowchart for an embodiment of an login procedure ;

Fig. 10 showing a flowchart for an embodiment of the releasing of file locks;

Fig. 11 showing a flowchart for an embodiment of the retrieving of the ownership of a task.

In the following several embodiments of methods and systems are described which allow a fast and efficient transfer of files 50 via data transfer links 1. This allows e.g. a file transfer between clients that is directly integrated within the users' computer systems 10, 20. In one embodiment the transmission of files 50 between users or groups of users by just one click is facillitated.

In one embodiment, users can establish data transfer links 1 between their computer systems (e.g. PCs). The data transfer links 1 are referred to as BitTubes 1 in the following. In beforehand, users can agree on such a BitTube 1 between each other, establishing a data channel between the computer systems 10, 20, as it will described more fully in connection with Fig. 2. A data transfer link 1 or BitTube 1 can be a data transfer line which is either permanently present between one sender (the first computer system 10) and at least one recipient (the second computer systems 20) or which is automatically established between sender and recipient or recipients when a certain action is performed by the sender. In those embodiments one difference to e.g. sending an E-mail is that the data transfer via the BitTube can be effected without that the user has to invoke a separate (i.e. separate from the operating system) program for the data transfer.

Fig. 1 schematically shows a desktop of a personal computer, i.e. a computer system 10, 20. The desktop is divided into two areas, the common desktop region 100 as it is known from operating systems and a BitTube region 200. In the BitTube region 200 representations (i.e. the respective ends) of five BitTubes, here termed BitTube ends 11, 12, 13, 14, 15 are shown. The BitTubes ends 11, 12, 13, 14, 15 conntect the computer 10, 20 systems as will be explained below.

These BitTube ends 11, 12, 13, 14, 15 are assigned to different persons or computer systems 10, 20. The BitTube ends 11, 12, 13, 14, 15 symbolize a direct connection between different users or group of users.

In Fig. 2 the connection of two computer systems 10, 20 is shown. The first computer system 10 comprises five BitTube ends 11, 12, 13, 14, 15, the second computer system 20 comprises two BitTube endes 11, 12.

In Fig. 2 it is shown, how a user Bob using the first computer system 10 can transmit a data file 50 (e.g. a document) to a user Alice using the second computer system 20. Both users have established a BitTube 1 between each other. User Alice can drag the data file 50 on the BitTube end 11 assigned to the user Bob on her desktop and the data file 50 automatically appears at the tube end 11 of user Bob. Thereby, any data file 50 can e.g. be transferred by only one click or pointing action. The dragging of the data file 50 to the BitTube end 11 is a form of associating (or coupling) the data file 50 with a particular recipient (a person or a computer system 20) . The drag-and-drop procedure the BitTube end 11, 12, 13, 14, 15 initiates the file transfer. The association can technically achieved by other means, e.g. by typing in a command or certain keystrokes. In the following decription the drag-and-drop embodiment is most often described. The person skilled in the art will recognize that this is just on particular embodiment of the general case of associating a data file 50 with a BitTube end 11, 12, 13, 14, 15.

The data file 50 can be e.g. a wordprocessing document or a spreadsheet, i.e. relatively small files which are transferred more or less instantaneously. The data file 50 can also be a very large file, e.g. a video file which is transferred in a form of streaming to the recipient . The

recipient could start viewing the data file 50 before it is transferred completely to the second computer system 20.

Especially when BitTubes is embedded within the context of a Web application, the data file 50 in the present context is not limited to the applications just mentioned. The data file 50 can e.g. comprise URIs pointing to a specific Web resource or an abstract object describing a resource's state, such as a JSON object or a XML file. Thus, in case a specific object (such as an image) should be transmitted via BitTubes, the transmission is not limited to the transmission of the object (here: image file) itself, but can also be a pointer (URI) or abstract object describing it in a way that it can be accessed or automatically created at the receiver side.

In the first instance, the abstract object could contain a description of a Google Maps view, including the resolution, the center location and further details. In case one user wants to transmit a Google Maps image with a BitTube 1, not the image file itself but an abstract object describing the view (including the center location, overlays, content, routes etc.) might be sufficient to enable the recipient to automatically create and view the Map with the same options and settings as on the sender side. JSON is of course only on example of a data file 50 to represent such an abstract object describing a resource at the sender side, which can be transmitted via BitTubes. In the same context, it might not be necessary to transmitte a complete file but rather a link (e.g. URL/URI) . In general, a data file 50 for the present purpose can contain the necessary information itself (e.g. a word processing file) and /or it comprises a meta information, a pointer and /or a link to the necessary information.

With this general understanding of the term data file 50 it is possible to integrate the method in a Web application. BitTubes 1 can be integrated directly into Web applications.

In the following embodiments of methods and systems are described which allow e.g. the transmission of data files 50 in one click by dragging the file on a special field, termed the BitTube end 11, 12, 13, 14, 15.

Within the following sections, we describe various embodiments of technical solutions. Sections 1.1 to 1.4 outline the general idea and envisioned features of the BitTubes approaches from a usability point of view. Concluding, section 1.5 will discuss the realization of the approach's Backbone, i.e. the actual transmission of the data file 50 between two or multiple users. Here, technologies based on open standards are given as examples .

1.1. Displaying BitTubes

The BitTube ends 11, 12, 13, 14, 15 representing individual ends of BitTubes 1 are integrated on the desktop of the computing system 10, 20 (see Fig. 1 or 2). In case more BitTubes 1 or BitTube endes 11, 12, 13, 14, 15 are defined than displayable on the screen of a user, two arrows appear on top end on bottom of the list of tubes. When a data file 50 is dragged over one of these arrows, the list of BitTube ends 11, 12, 13, 14, 15 scrolls so that the data file 50 can be dropped (i.e. associated with a BitTube 1) on a BitTube end 11, 12, 13, 14, 15 that was not initially visible. This secures the transmission of files 50 by one click even if a large number of possible receivers exists.

1.2. Transmission modes

Four transmission modes can be distinguished. Accordingly, BitTubes end 11, 12, 13, 14, 15 can be marked in different colours on the desktop in order to indicate which transmission mode is currently available.

1.2.1. Online Transmission Mode

In this mode both users are online, i.e. connected by the Web. This could be marked by e.g. a green color of the respective BitTube end 11. When dragging the data file 50 to the BitTube end 11 e.g. as shown in Fig. 2, the data file 50 is directly transmitted to the BitTube end 11' of the respective user. Therefore, a P2P style communication is established, by associating the data file 50 with the BitTube end 11.

1.2.2. Offline Transmission Mode

In this mode, e.g. the receiving user of the second computer system 20 is offline, e.g. not connected to the Web (indicated by the dashed line in Fig. 3). The respective BitTube end 11 on the first computer system 10 would be displayed in red colour on tube region 200. In case the first user sends a data file 50 to the user of the second computer system 20 who is currently offline, the respective data file 50 is cached on a server 70 and delivered to the receiver as soon as the receiver becomes online, i.e. connected to the Web.

1.2.3. Group Transmission Mode

BitTube ends 11, 12, 13, 14, 15 can be assigned to either one specific user or a group of users as shown in Fig. 4. In Fig. 4 a first BitTube end 11 is assigned to a group of second computer systems 20a, 20b, 20c. By dragging and dropping the data file 50 to this BitTube end 11, the data file 50 is automatically transferred to the respective BitTube ends 11', 11'', 11''.

Thereby, both unicast and multicast file transmission are enabled. If a group is partially online, partially offline as defined above, online transmission mode is applied for the

online users, offline transmission mode for the offline users. BitTube ends 11, 12, 13, 14, 15 for groups of users who are partially online and offline can be presented in orange colour on the BitTube end region 200 on the desktop.

1.2.4. Context -Aware Transmission Mode

In this mode it is possible that a user can transmit a data file 50 under specific circumstances, i.e. the transfer depends on certain predetermined conditions. For instance, a data file 50 should be sent to a user at a specific point in time or when the receiver enters a specific location. In this case, the sent data file 50 is e.g. transferred to a server and transmitted to the receiver when the defined rules applied, i.e. when the specified point in time occurs or the receiver reaches the specified location. This mode can work with the online mode (Fig. 2), the offline mode (Fig. 4) and the group mode (Fig. 4).

1.3. Drag modes

Dragging the data file 50 with a left mouse click to the respective BitTube end 11, 12, 13, 14, 15 sends the data file 50 to the corresponding user and leaves the file on the sender's desktop (copy mode) .

Dragging the data file 50 with the right mouse button transmits the data file 50 and deletes it from the sender's desktop (move mode) . Latter mode enables the freeing of disk space with one click, e.g. for big temporary data files that are not required after they have been created by the sender, but are only of use for the requester.

Naturally in alternative embodiments, the choice of the mouse operation or the operation of any other poiting devices can be different.

1.4. Protecting against unintentional file transmission

In order to secure that a data file 50 is not transferred to an unintended recipient or unintended computer system 20 because the sender drags a data file 50 to a BitTube end 11, 12, 13, 14, 15 accidentally, a gauge 16 (see Fig. 1) indicating a count-down (e.g. a progress bar) can appear above the BitTube end 11, 12, 13, 14, 15 of the dragged data file 50 when the mouse pointer with the dragged file is over the BitTube end. After a predefined time span (e.g. 5 seconds) has been elapsed, the tube is freed for transmission. If the user releases the mouse button earlier the file is not transmitted. This delay device improves the security of file transmission.

1.5. Backbone: File Transfer

There are multiple options to transfer a data file 50 from one endpoint in a computer network to another. A possible approach for BitTubes 1 is to utilize a secure point-to-point file transfer protocol atop the standard Internet Protocol (IP) (see http://tools.ietf.org/html/rfc79 and http://tools.ietf.org/html/rfc2460). One embodiment uses the File Transfer Protocol described in http://tools.ietf.org/html/rfc959 .

Control and data channel are secured by the open Transport Layer Security protocol (TLS) described in http://www.ietf.org/rfc/rfc4346.txt .
The bundling of both protocols is referred to as FTP over TLS (FTPS) as decribed in http://tools.ietf.org/html/rfc4217 .

Point-to-point connections for file transfers between two computing systems in an IP network require the initiating endpoint of the connection (client) to know the IP address of the terminating endpoint of the connection (server). Assumed that all BitTubes user have a BitTube 1 user name, a

resolution mechanism from BitTube user names to IP addresses can be used to establish a file transfer connection.

Generally, BitTube 1 users must define there tubes themselves, i.e. define to whom they want to send and from whom they want to accept files. Alternatively this can be arranged with a preconf igured setup.

An intermediate component in the IP network, hosted on a third computing device (beside client and server) with a fixed IP address is required to ensure both of the above two issues. This intermediate component is in the following referred to as BitTubes Broker and basically implements a server for the Hypertext Transfer Protocol (HTTP) as described in http ://www. ietf .org/rf c/rf c2616 .txt .

The access to the BitTubes Broker is again secured through TLS within the HTTP extension HTTP over TLS (HTTPS) as described in http ://www. ietf .org/rf c/rf c2818 .txt .

File transfer originating and file transfer terminating endpoints can access the BitTubes Broker to:

- Manage (create, update, delete) BitTubes 1
- Lookup the IP address of the BitTube 1 terminating endpoint (e.g. the second computer system 20)
- Pre-verifying acceptance of file transfer through reverse resolution of IP address to BitTube user name for a requested data connection

Further intermediary components in the IP network may be used to cope with problems caused by Network Address Translation (NAT) and temporal decoupling of client and server (as indicated in section 1.2.2). A kind of back-to-back FTPS server which acts as relay for file transfer originating endpoint and file transfer terminating endpoint could, for instance, be utilized as solution.

2. Task Annotation Extension

Further possibilities can be realised by task annotations for the files 50 transferred. In the following this is described in more detail.

Data files 50 can be annotated with an annotation tag 60, e.g. notes specifying tasks for the recipients. Annotation tags 60 can automatically trigger some reaction when the data file 50 is received by a computer systems 20. These Annotation Tags 60 can e.g. be stored as proprietyary XML files within a predefined folder of a user's file system. Beside the information on the task itself, the Task Annotation can contain the path to the file it is associated with.

The annotation tags 60 can be reduced to simple, unified instructions. The concept is depicted in Fig. 5.

2.1. Defining Tasks

In Fig. 5 an annotation tag 60 is schematically shown. This annotation tag 60 can be attached and / or embedded with the data file 50 to be transferred. This can be done individually by a user before submitting the data file 50 to a BitTube 1 or it can be done automatically by the BitTube 1 and / or BitTube ends 11, 12, 13, 14, 15 when they determine the sender information, the content information of the data file 50 and / or the recipient information of the data file 50.

The Task element 61 of the annotation tag 60 specifies the task that has to be performed on the data file 50. This information is e.g. automatically displayed or attached to a calendar of a user. Classic operations performed on files 50 (such as documents) can be predefined. For instance, files 50 can be annotated to be read, written, signed, corrected and

so forth. Further tasks can be defined by users and communicated to other possible recipients .

The Processing element 62 of the annotation tag 60 of the task defines the processing order of the task in case multiple recipients are addressed. For instance, it may be allowed that a data file 50 is simultaneously read by all recipients, but can only be signed successively in order to avoid conflicts and lost modifications.

Last, the Acknowledge element 63 of the annotation tag 60 enables the receiver to define whether an acknowledgement of a performed action of one of the recipients should be sent or not. This can be automated.

It is possible to extend the task annotations by further components such as a blank field to insert text messages for the receiver (s) .

Tasks can be customized through task profiles. For instance, a lawer's office may define a task annotation profile featuring the tasks "read" and "sign" and encompass a deadline. Another profile may simply contain a message field. A user annotating a data file 50 can now chose, which type of task should be attached to the file, i.e. either a task containing the "read" and "sign" tasks together with a deadline or a task simply encompassing a message. Thereby, tasks annotations become customizable.

2.2. Defining Access Rights

The annotation tag 60 can be part of a data set attached and / or embedded with the data file 50. Furthermore, the data file 50 can be tagged with access rights 64. Here, it may be possible e.g. to restrict the access to data file 50 to read only. Moreover, it is possible to extend the access rights to parts of the data file 50 itself. Thereby, a whole data file 50 such as a document may be marked as read only while the

recipient may be granted write access to a specific section or part within the data file 50 in order to modify its content. Again the access rights 64 can be set individually by a user or automatically by the BitTube 1 and / or the BitTube ends 11, 12, 13, 14, 15.

2.3. Defining Deadlines

An integrated calendar enables the definition of deadlines for specific tasks. Thus, a recipient has to perform the requested task and send the data file 50 back before the deadline expires. This can be integrated with other communication technologies in order to remind users of approaching deadlines. For instance, an Email, SMS or automatic call may be sent to the user that notifies him or her about the approaching deadline.

2.4. Annotation data

Beside the tasks descriptions, the following information can be stored together with the data file 50 and / or the annotation tag 60:

- Sender of the file
- List of all recipients including their addresses

2.5. Ensuring consistent data

BitTubes 1 provide a data transfer with one click; more precisely, files (or documents) are copied or moved between distributed computing devices. Thus, in case multiple users are addressed with one data file 50, multiple copies of the same data file 50 exist. In this case, it has to be guaranteed that no inconsistencies occur within the distributed files 50 and that every user has the file in the most up to date version. In case the transmitted data file 50

is marked as "read only", this aspect is irrelevant. However, if multiple recipients have to modify the data file 50, the consistence of the data has to be ensured. Note that BitTubes 1 can be based on a Peer-to-Peer approach, thus, the sent data is not residing on a central server where the access to a central file can easily be restricted in case one user is currently accessing it. Instead, a messaging mechanism can be used to restrict the access to files and update them. In that sense, BitTubes specifies a protocol that creates a Client/Server Overlay for P2P infrastructures. Thus, although the single devices are only connected in P2P fashion, i.e. hold a local copiy of a file instead of sharing a global copy on a central server, data consistency and synchronization are ensures as if the files were shared globally.

Assume two users *A* and *B* received a data file 50 from user C where both *A* and *B* should add a personal curriculum vitae at the end of the data file 50. This scenario exposes so-called "hidden writing" threats, since the last person writing to the file and sending it back would overwrite the prior modifications. BitTubes 1 therefore use an access restricting mechanism that is schematically depicted in Fig. 6.

In case a user tries to open a data file 50, it is first checked whether the data file 50 has not yet been opened by another user. Therefore, one user, the so-called owner of a task, holds a semaphore that guarantees the single access of files. A user creating a task is initially the owner of this task. (Within the example above, user C would initially be the owner of the task.) In case a user, who holds the ownership of one or more tasks, logs out of the BitTubes application, the ownership is moved to another user who holds a copy of the file. This procedure ensures even within P2P systems that a file can be accessed with writing access even if all other users are offline.

If it is already opened with writing access for another user, the data file 50 is opened with read only rights (left branch in Fig. 6). Otherwise, the access to the data file 50 is blocked, the copy of the file is updated if necessary and the file can be edited. Afterwards, when the editing is completed and the file is shut down, the modified file is transferred to the owner the replace he old copy and the file lock is released.

Continuing the above mentioned example: User *A* and B have received a data file 50 to add their personal curriculum vitae. The procedure explained in the following also holds for more users, i.e. users *B'*, *B''*, and so forth.

A detailed specification of the access restriction is given in Figure 7. When user B accesses a file, it is checked with the owner of the file whether the file is already opened with read/write rights by another user or not. Therefore, a requestLock (FILE) message is transmitted to the owner containing additional information on the file such as its Hash value and a time stamp indiciating the last point in time the file has been modified. The owner checks whether user B holds an up-to-date copy of the file. If yes, the file is locked and an ACK is sent to confirm the read/write access to the file. If the version is outdated, the new version is transmitted. Afterwards, the lock of the file is checked. In case the file is already in use (with read/write rights) by another user, a NACK is sent back and the file opens in read mode only. Thus, while the user B is accessing the file in read/write mode, all other users trying to open the data file 50 (such as user A in Figure 7) would only receive reading access to their files 50. As soon as the user with the writing access to the data file 50 has closed the data file 50, a new version of the data file 50 is distributed among the other users, i.e. the old file is replaced automatically; afterwards, the lock is released such that other users can access the data file 50 with writing capabilities.

The process of updating files can be timewise displaced. Assume user B has finished editing and user A is online. The owner of the file would then update user A's file such that he/she holds an up-to-date copy. Assume another user C holds a copy of the file, but is not online when user A sends in the update. In this case, user C would have an outdated copy. However, as soon as user C wants to access the file, he/she would get an update of the file before opening it. Although an update of the modified file to all users that are online is not mandatory, since every user is assured to have an up-to-date copy when he/she opens a file with read/write access, it is nevertheless recommended. In case a user is offline, he/she can nevertheless open a file in read only mode. Here, the last version of the file is accessed. When updates are sent to all users that are online every time when a file is modified, the chances are less that a user opens an outdated document when he/she is offline. Moreover, an update of a file can be regarded as a signal for a released lock. For instance, within Figure 7, user A has tried to access the file while user B already held read/write access rights. Therefore, the file was opened in read only mode for user A. However, as soon as user A gets an update of the file, this update also serves as signal for the fact that user B must have released the lock of the file and that user A can now get write access to it.

3. Applications

The following sections describe some applications of embodiments using BitTubes 1.

Depending on the number of BitTubes 1, advertisements (e.g. such as Google Ads) can be displayed directly on the user's desktop. In a further embodiment, the BitTubes 1 might be configured with a special filter to block certain advertisements.

The transmission of files 50 in the offline mode defined in section 1.2.2 can be billed (since a server is required). Here, multiple billing models are possible:

1. Subscription and payment per month/year
2. Payment per transferred volume
3. Pay per data file 50
4. Payment depending on the number of tubes that are open simultaneously

Apart from applications directly related to the BitTubes 1 themselves, they offer other possibilities.

Netbooks are hardware-restricted subnotebooks that were introduced in order to provide cheap mobile notebooks for peoples' everyday lifes. Those restricted subnotebooks are commonly not shipped with hard drives, but only flash memory of e.g. 2GB size. Here, storage capacity is a scare resource. With BitTubes 1, files 50 or data can be sent and automatically deleted afterwards, instead of filling the outbox of an email client. Moreover, the idea of restricted subnotebooks was to provide an easy to use computing device to the users. Here, BitTubes 1 usability is a good fit to provide a novel way for spontaneous, easy and fast data transmission.

But also in business communication the BitTube concept can be applied. Business people can e.g. define a BitTube *myMeeting* during a business meeting. Then, files can be shared with one click, speeding up the sharing of temporary content.

For travelling persons BitTubes 1 are a convenient way to deal with all kinds of multi-media content, including photos, videos, music, short messages and so on. BitTubes 1 enable a fast way to spontaneously transmit files to friends or family. For instance, as soon as Alice has taken a picture

from herself in front of Big Ben, she simply drags the file to her "Family Tube" . The members of this group will either receive the file in a realtime fashion (in case they are online) or as soon as they connect to the Web (currently offline) .

The concept of BitTubes 1 is very accessible so that handicapped or elderly people have no difficulties using it.

In Fig. 8 to 11 flowcharts for embodiments of a method using data transfer links 1 are shown. These representations especially highlights the importance of the file access status and the file ownership management.

Every time a user defines a task for a data file 50 and transmits it to other users via a data transfer link 1, a virtual ownership of the task is created that is assigned to the the creator of the task. A user who holds the ownership of a task acts as a "virtual server" for the respective file, i.e. manages the access rights to the file in a P2P way.

Fig. 8 an overview of the attribution of the "virtual server", involving an file locking test ist described.

The functionality shown in Fig. 8 is best explained from a user (Alice) who is online 801 and who is currently the "virtual server" for a data file 50 which can be handled by a method for transferring the data file 50. Now Alice is about to log out, so that the ownership of the rights to the data file has to be transferred to a diffrent user.

Before Alice logs out, in step 802 the list of users currently online is checked. If the list is empty (step 803), Alice logs out (step 800) , since the task cannot be transferred to somebody else in the system. The next user who logs in, will be prompted, as will be explained in connection with Fig. 9 .

If the there are users online, it is established in step 804 if the data file 50 is locked by some user. If yes, a "release lock" procedure is invoked (step 1000, see Fig. 10).

If the data file 50 is not locked by some user, a new owner is retrieved with a procedure 1100 which is described in more detail in Fig. 11.

In the end, a new owner is defined (step 805) and all users are notified of the new owner (step 806), who serves as the virtual server for the data file 50 to be transferred.

In Fig. 9 the procedure at login is described. This ties in with the description in Fig. 8.

Fig. 9 shows the process which takes place, if a user logs into an application which utilizes data transfer links 1, like e.g BitTubes.

When a user Alice logs in, the BitTube UserName (step 901) and the password (step 902) are provided. Furthermore, communication with the Management Server (step 903) takes place.

After checking the login data (step 904) and a positive test (step 905) the user information is determined (step 906) and a current user status is assigned (step 907).

In a subsequent step 908 permission for contacting other user is sought and the users are notified (step 909).

Following further in Fig. 8, it is established if the current user list on the system is empty (step 807).

In the sequence diagramm in Fig. 10 the release lock process 1000 is described in more detail .

A user Alice is current owner of the data file 50, so that others cannot alter it. Further, there is second user Bob who wants to alter (write to) the file. Further, there are other users who have just write access.

In step 1001 a release lock command is sent to Bob. Bob replies with an OK (step 1002) . The system database is updated with this information (step 1003) and update information is sent to the other users (step 1004) .

In Fig. 11 the retrieve new owner procedure is described in more detail. This is the procedure (see Fig. 8) in which a new owner of the data file 50 can be determined after the data file 50 has been unlocked (see Fig. 10) or the data file 50 was not locked by the previous owner. This is e.g. used in connection with the method for transferring a data file 50 via a data transfer data link 1 .

After getting the current information about the present tasks (step 1101) , it is checked if the list of the present user who is writing to the the file is empty (step 1102) .

If not, the current writer becomes the owner of the task, i.e. this user now becomes the virtual server (step 1103) . This is shown in the embodiment shown in Fig. 10, i.e. the user Bob is given the task after the user Alice logs out.

If the list of the current users with write access is empty, the list of the task users is accessed (step 1104), i.e. the list containing all users that are assigned to the specific task T . Now, all users within the list are evaluated with regard to their capability to overtake the task ownership. If is the end of this list is reached without that a new appropriate owner has been found (step 1105) , the procedure

stops since nobody can take on the task ownership. As long as the list is not yet empty, the next user is chosen. In case the user is currently offline, he/she cannot be assigned the ownership of the task. Moreover, if the currently selected user is equal to the user that originally has held the task ownership, he/she is also not an appropriate choice sind this user has just decided to log out and thereby initialted to reassignment of the task ownership. Thus, in case the selected user is neither offline nor identical to the origin task owner, he/she is selected (step 1106) and assigned the task ownership (step 1107) .

Reference numbers

1 Data transfer link, BitTube

10 First computer system
11 BitTube end
11' BitTube end
11" BitTube end
11'" BitTube end
12 BitTube end

13 BitTube end
14 BitTube end
15 BitTube end
16 Progress bar of delay device

20 Second computer system

30 Cache Server

50 File

60 Annotation tag
61 Task element
62 Processing element
63 Acknowledge element
64 Access right element
65 Deadline element

100 Common desktop region
200 BitTube end region

800-807 process steps in Fig. 8
900-909 process steps in Fig. 9
1000-1004 process steps in Fig. 10
1100-1107 proces steps in Fig. 11

Patent claims

1. Method for transferring at least one data file (50) from at least one first computer system (10) to at least one second computer sytem (20), wherein

a) at least one data transfer link (1) is established between the computer systems (10, 20)

b) the at least one file (50) is sent or received via the at least one data transfer link (1) by associating the at least one file (50) with at least one end (11, 12, 13, 14, 15, 11') of the at least one data transfer link (1) and

c) the at least one file comprises at least one annotation tag (60).

2. Method according to claim 1, wherein the association of the at least one data file (50) with the at least one end (11, 12, 13, 14, 15) of the data transfer link (1) is effected by only one user input, especially one mouse click, a drag-and drop mechanism and / or one movement of the data file (50) in a GUI.

3. Method according to claim 1 or 2, wherein the annotation tag (60) comprises a task element (61) with information about a task to be performed by a user, a processing element (62), an acknowledge element (63), an access right element (64), temporal information, especially a dead line (65), and / or a filed of inputting at least on string, especially text.

4. Method according to at least one of the preceding claims, wherein the at least one data file (50) comprises a link or pointer to information, especially an URL link and / or an abstract object, especially

represented by JSON or XML.

5. Method according to at least one of the preceding claims, wherein the data file (50) is interpretable directly in a WebBrowser to automatically create the representation of the Web resource that was dragged into data transfer link (1) by the sender although not the resource itself, but an abstract object describing its settgin was transmitted.

6. Method according to at least one of the preceeding claims, wherein the annotation tag (60) comprises a command that the at least one data file (50) is automatically forwarded to a second computer system (20) after a predetermiend task has been performed on a first computer system (10).

7. Method according to at least one of the preceeding claims, wherein the access status and / or the file ownership status of the at least one data file (50) received by one computer system (20) is automatically communicated to all other recipients of the at least one data file (50).

8. Method according to claim 7, wherein the access status is set automatically to ,,read only" when a the at least one data file (50) is opened by a recipient.

9. Method according to claim 7 or 8, wherein the file status of the at least one data file (50) is set to ,,write" after a recipient has completed a predetermined operation, especially has closed the at least one data file (50).

10. Method according to at least one of the preceeding claims, wherein the annotation tag (60) comprises „read¬ only" instructions for certain recipients of the at

least one data file (50) and / or for parts of the at least one data file (50) .

11. Method according to at least one of the preceeding claims, wherein the at least one data file (50) is instantaneously transferred or transferred after caching between at least two computer systems (10, 20) .

12 . Method according to at least one of the preceeding claims, wherein the at least one data file (50) is transferred via the data transfer link (1) in dependences of a predetermined rule set .

13 . Method according to at least one of the preceeding claims, wherein the at least one data file (50) is copied or moved to more than one recipient .

14 . Method according to at least one of the preceeding claims, wherein the at least one data file (50) is automatically encrypted, decrypted, compressed, decompressed and / or virus checked when it is associated with one end of the data transfer link (1) .

15. Method according to at least one of the preceeding claims, wherein the data transfer link (1) is coupled to a delay device (16) for delaying the sending of the at least one data file (50) .

16. System for transferring at least one data file (50) from at least one first computer system (10) to at least one second computer system (20) , with

a) at least one data transfer link (1) established between the computer systems (10, 20) ,

b) the at least one data file (50) is sendable or receivable via the at least one data transfer link (1)

by an association of the at least one data file (50) with at least one end (11, 12, 13, 14, 15, 11') of the at least one data transfer link (1) and

c) the at least one file comprising at least one annotation tag (60).

17. System according to claim 16, wherein the at least one end (11, 12, 13, 14, 15) of the data transfer link (1) comprises association means so that the association of the at least one data file (50) with the at least one end (11, 12, 13, 14, 15) of the data transfer link (1) is effected by only one user input, especially one mouse click, a drag-and drop mechanism and / or one movement of the data file (50) in a GUI.

18. System according to claim 16 or 17, wherein the annotation tag (60) comprises a task element (61) with information about a task to be performed by a user, a processing element (62), an acknowledge element (63), an access right element (64) and / or temporal information, especially a dead line (65).

19. System according to at least one of the claims 16 18, wherein the annotation tag (60) comprises a command that the at least one data file (50) is automatically forwarded to a second computer system (20) after a predetermiend task has been performed on a first computer system (10).

20. System according to at least one of the claims 16 to 19, comprising meand for communicating the access status of the at least one data file (50) received by one computer system (20) automatically to all other recipients of the at least one data file (50).

21. System according to at least one of the claims 16

to 20, with at least one caching server (70) for caching the at least one data file (50) between the least two computer systems (10, 20) .

22. System according to at least one of the claims 16 to 21, with at least one means for automatically encrypting, decryptiing, compressing, decompressing and / or virus checking the at least one data file (50) when it is associated with one end of the data transfer link (1) .

23. System according to at least one of the claims 16to 22, with a delay device coupled to the the data transfer link (1) for delaying the sending of the at least one data file (50) .

Fig. 1

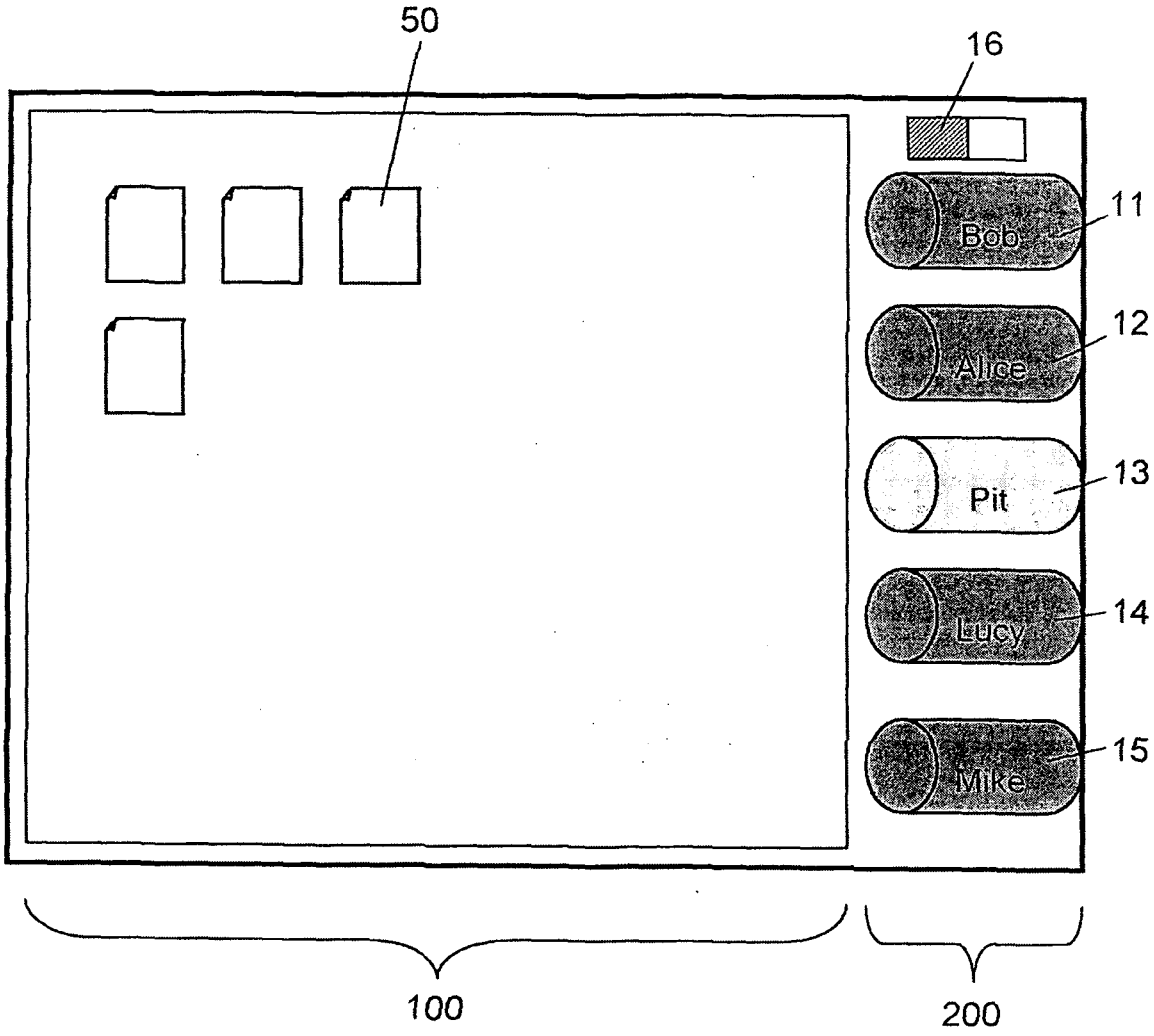


Fig. 2

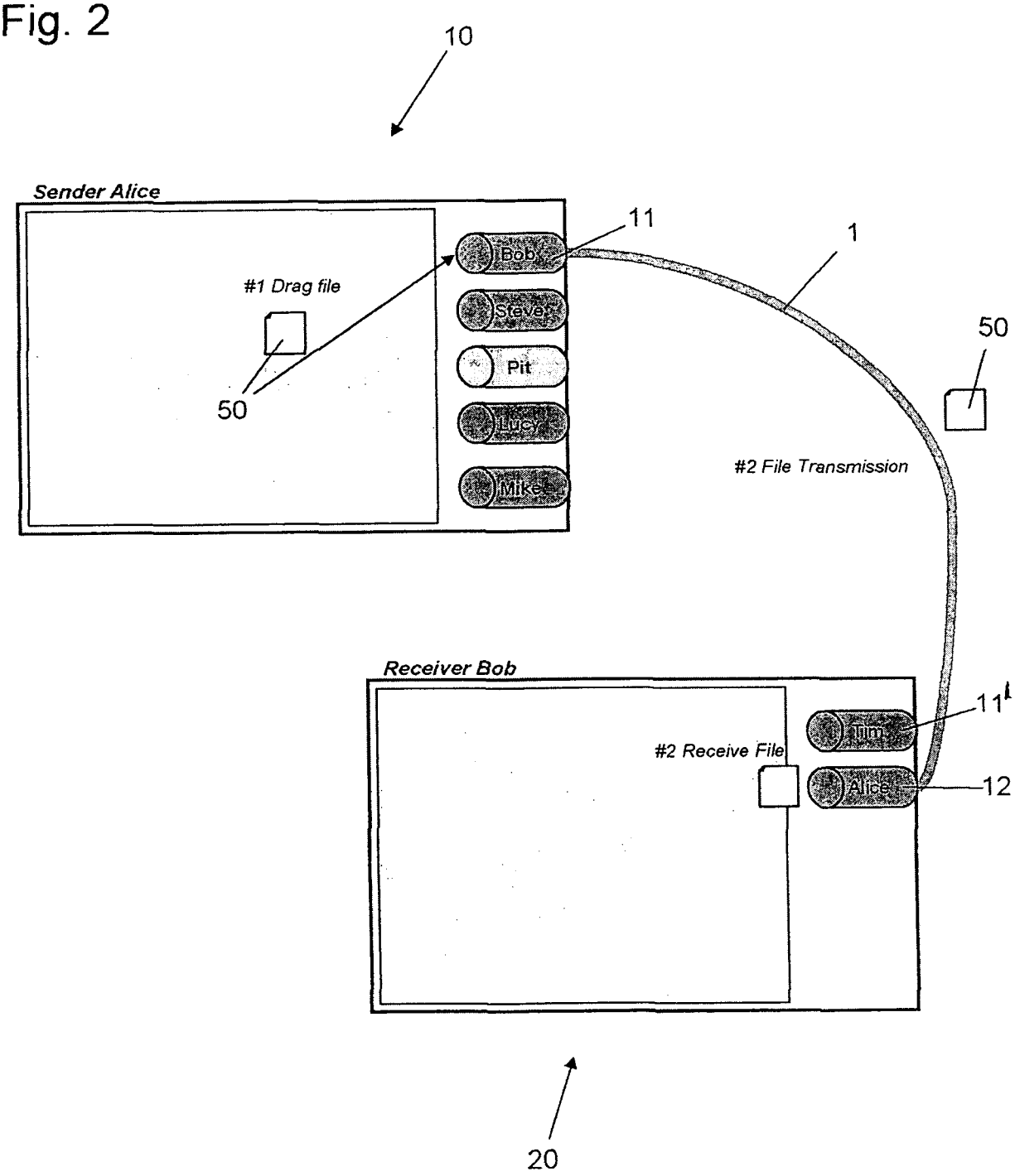
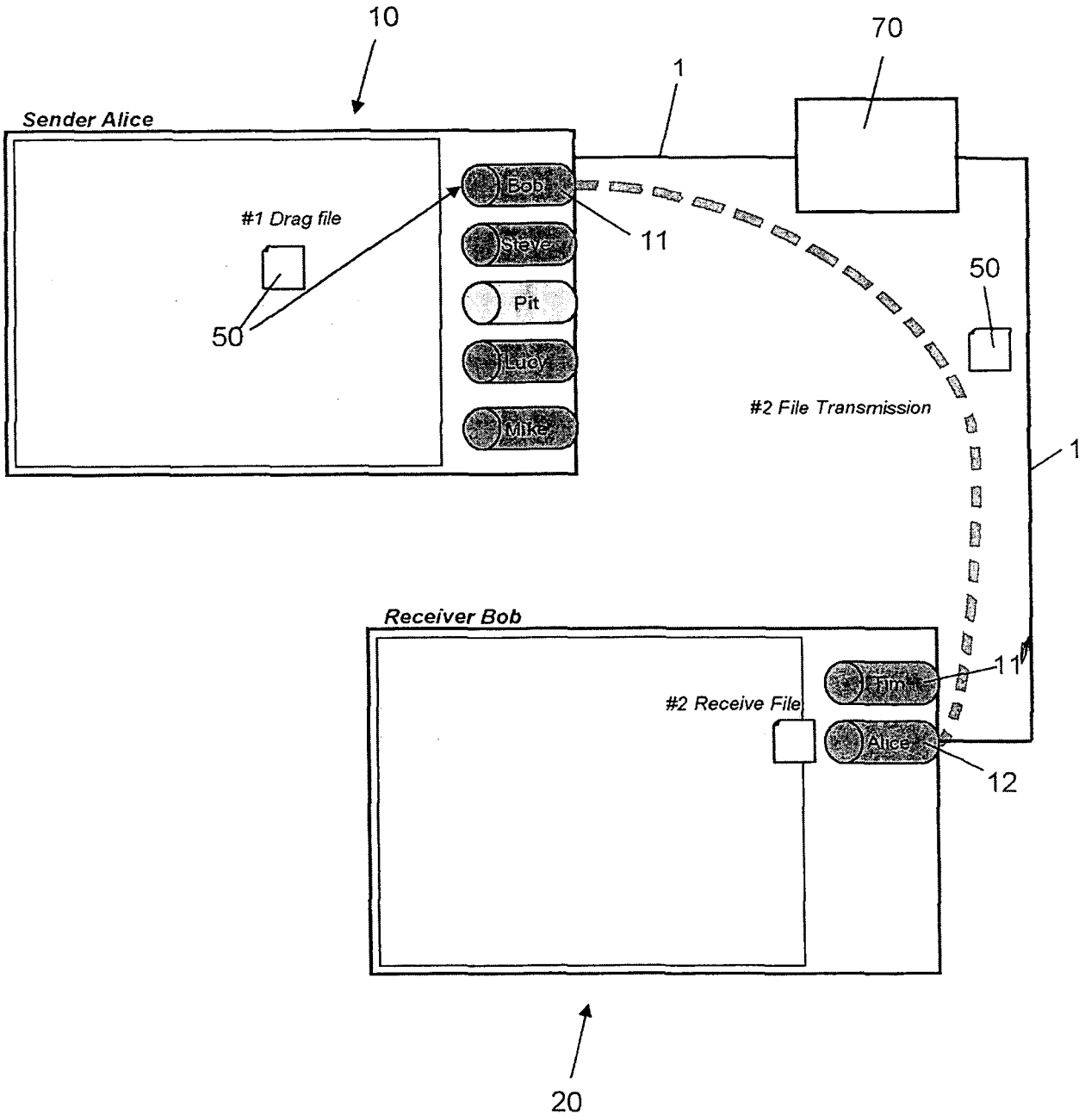

10
Sender Alice
Bob
Steve
Pit
Lucy
Mike
11
1
#1 Drag file
50
50
#2 File Transmission
Receiver Bob
Tim
Alice
11'
12
#2 Receive File
20

Fig. 3


10
70
1
Sender Alice
Bob
Steve
Pit
Lucy
Mike
#1 Drag file
50
11
50
#2 File Transmission
1
Receiver Bob
Tim
Alice
#2 Receive File
11
12
20

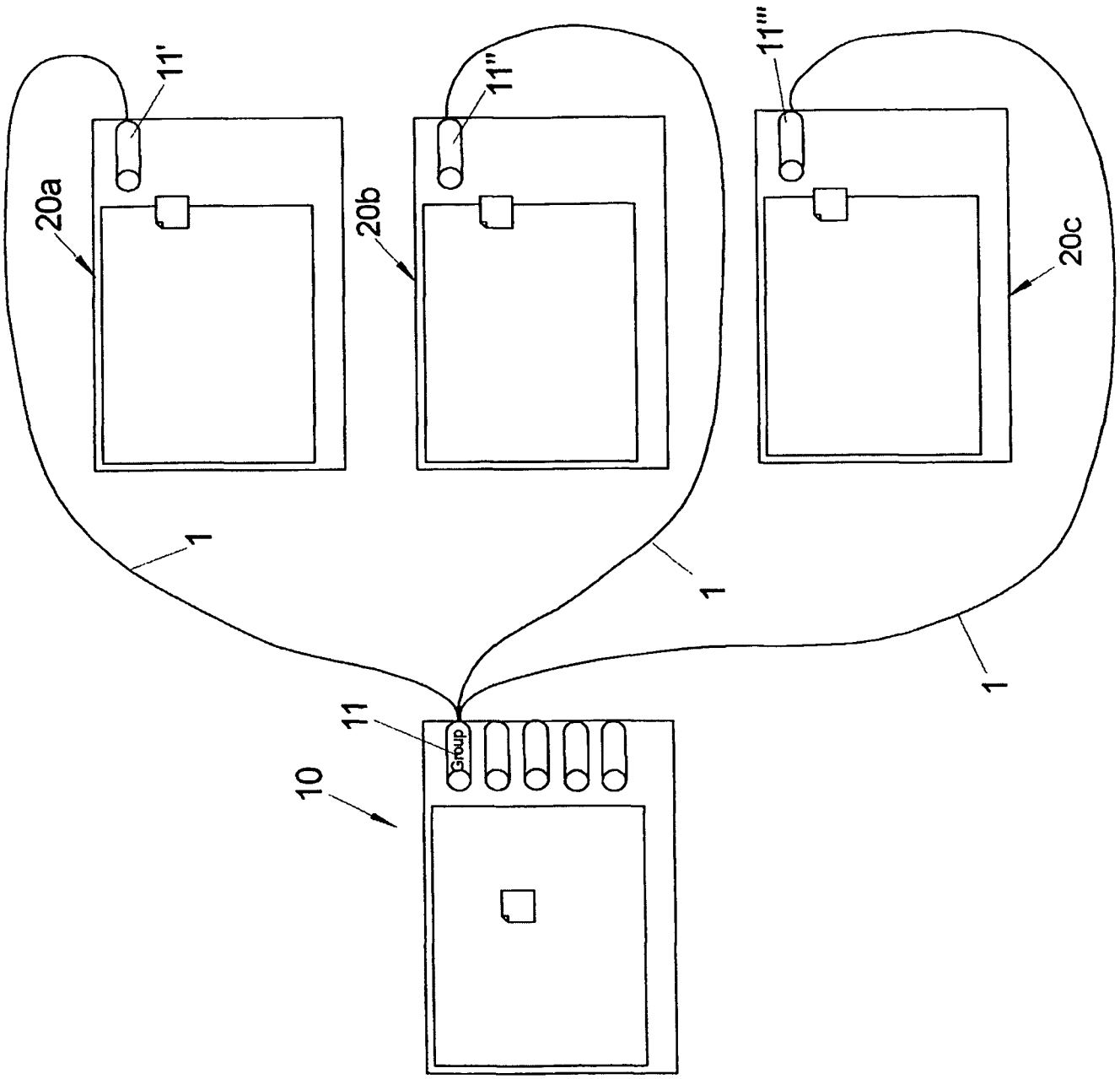


Fig. 4

Fig. 5

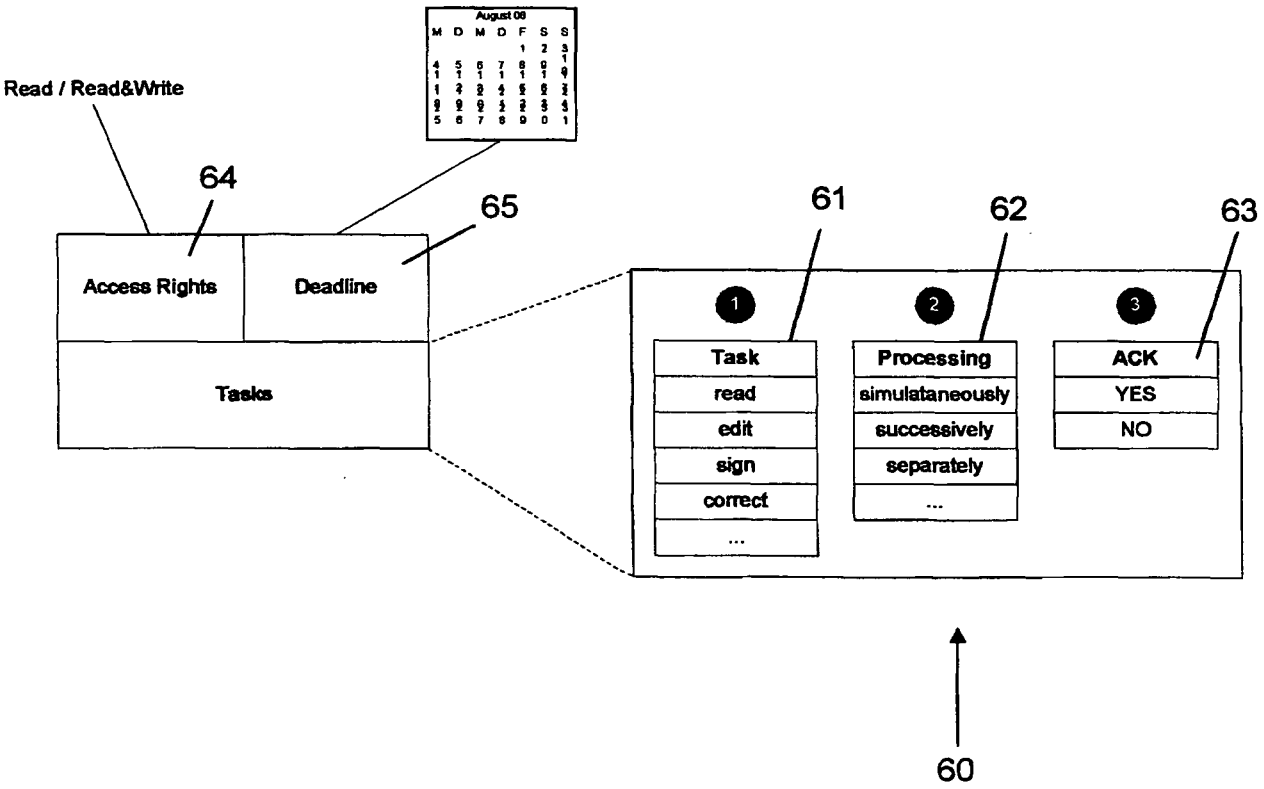

Read / Read&Write
64
65
61
62
63
Access Rights
Deadline
Tasks
1
2
3
Task
read
edit
sign
correct
...
Processing
simulataneously
successively
separately
...
ACK
YES
NO
60

Fig. 6

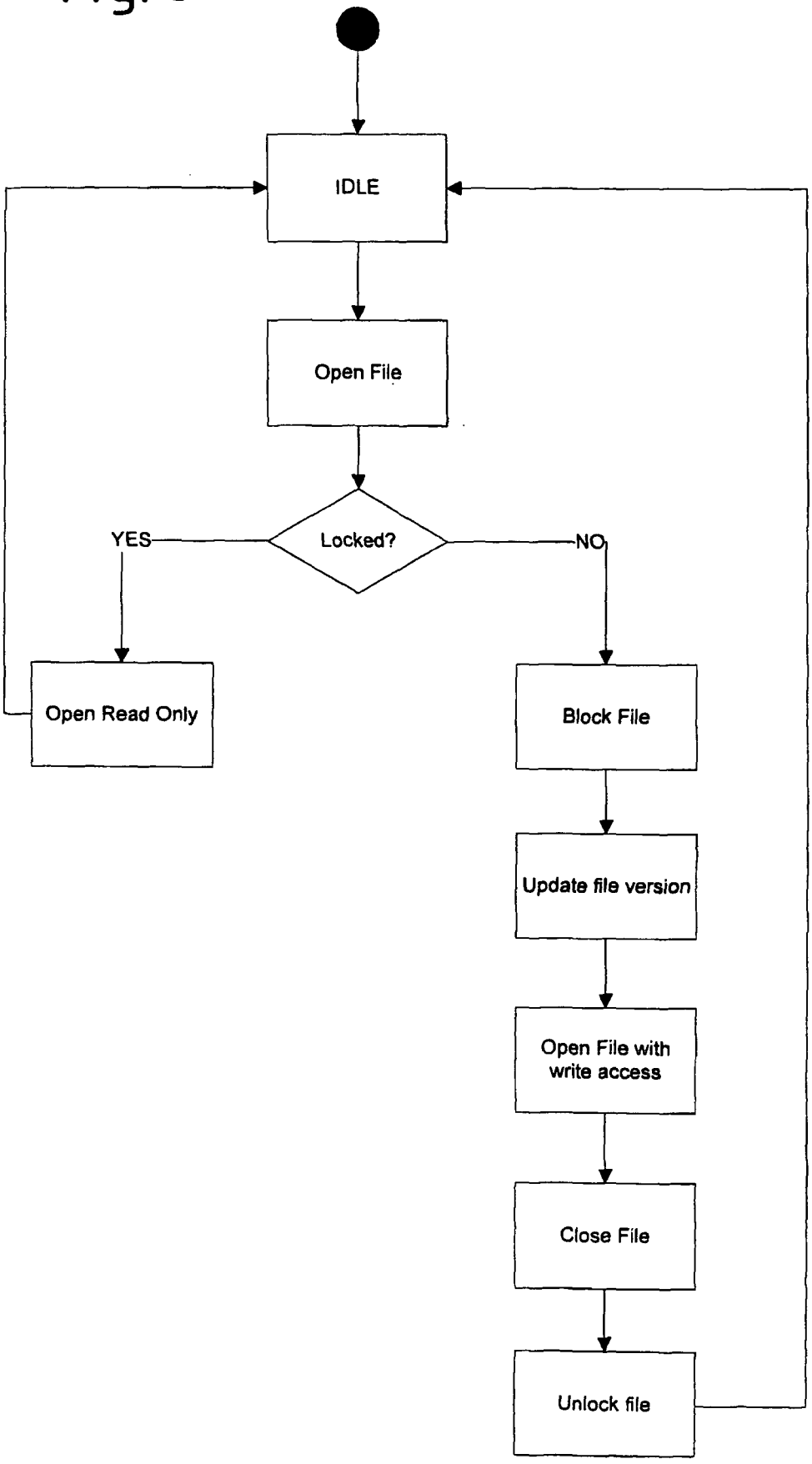
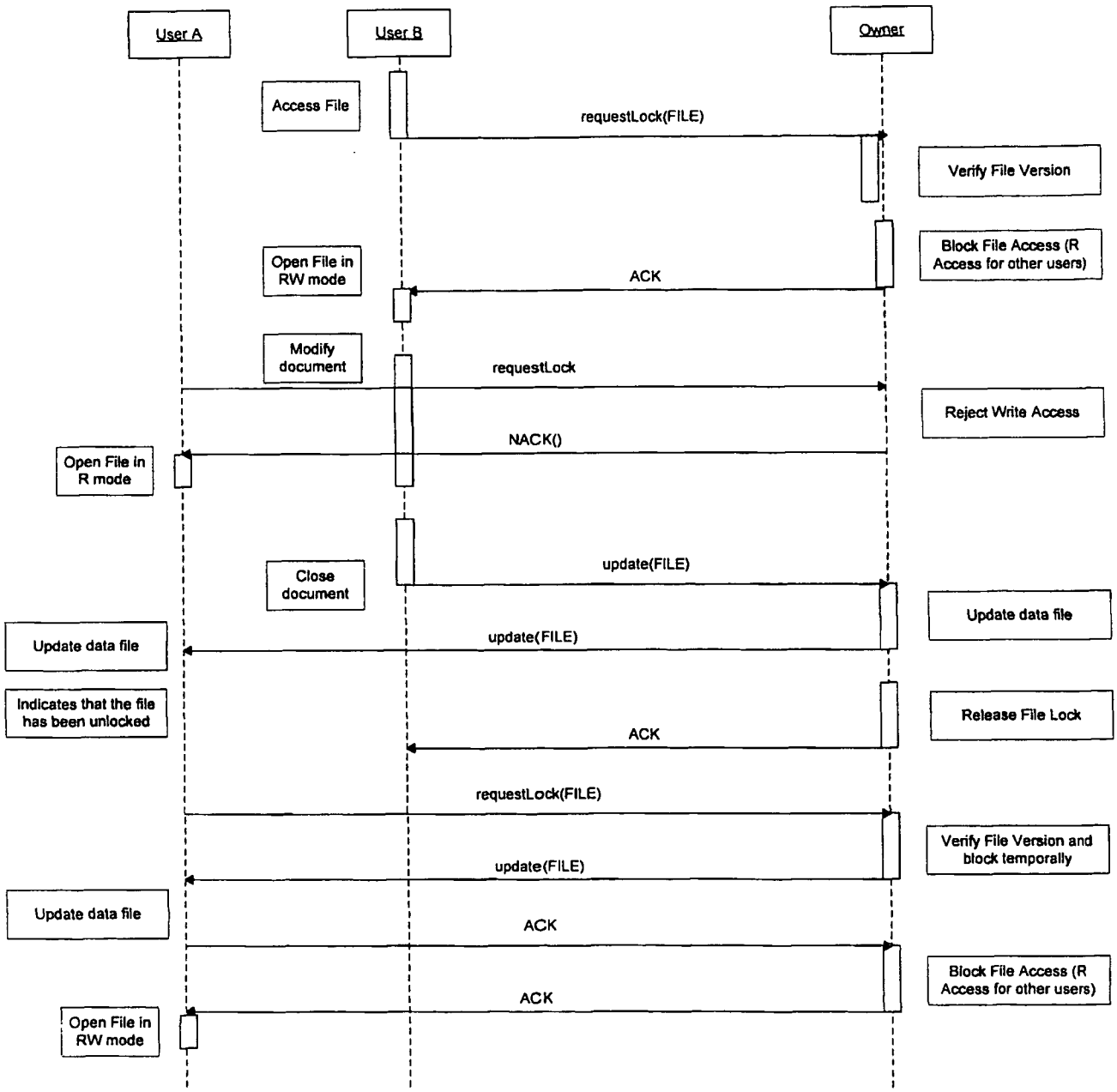

IDLE
Open File
Locked?
YES
NO
Open Read Only
Block File
Update file version
Open File with write access
Close File
Unlock file

Fig. 7


User A
User B
Owner
Access File
requestLock(FILE)
Verify File Version
Block File Access (R Access for other users)
Open File in RW mode
ACK
Modify document
requestLock
Reject Write Access
NACK()
Open File in R mode
Close document
update(FILE)
Update data file
Update data file
update(FILE)
Indicates that the file has been unlocked
Release File Lock
ACK
requestLock(FILE)
Verify File Version and block temporally
update(FILE)
Update data file
ACK
Block File Access (R Access for other users)
ACK
Open File in RW mode

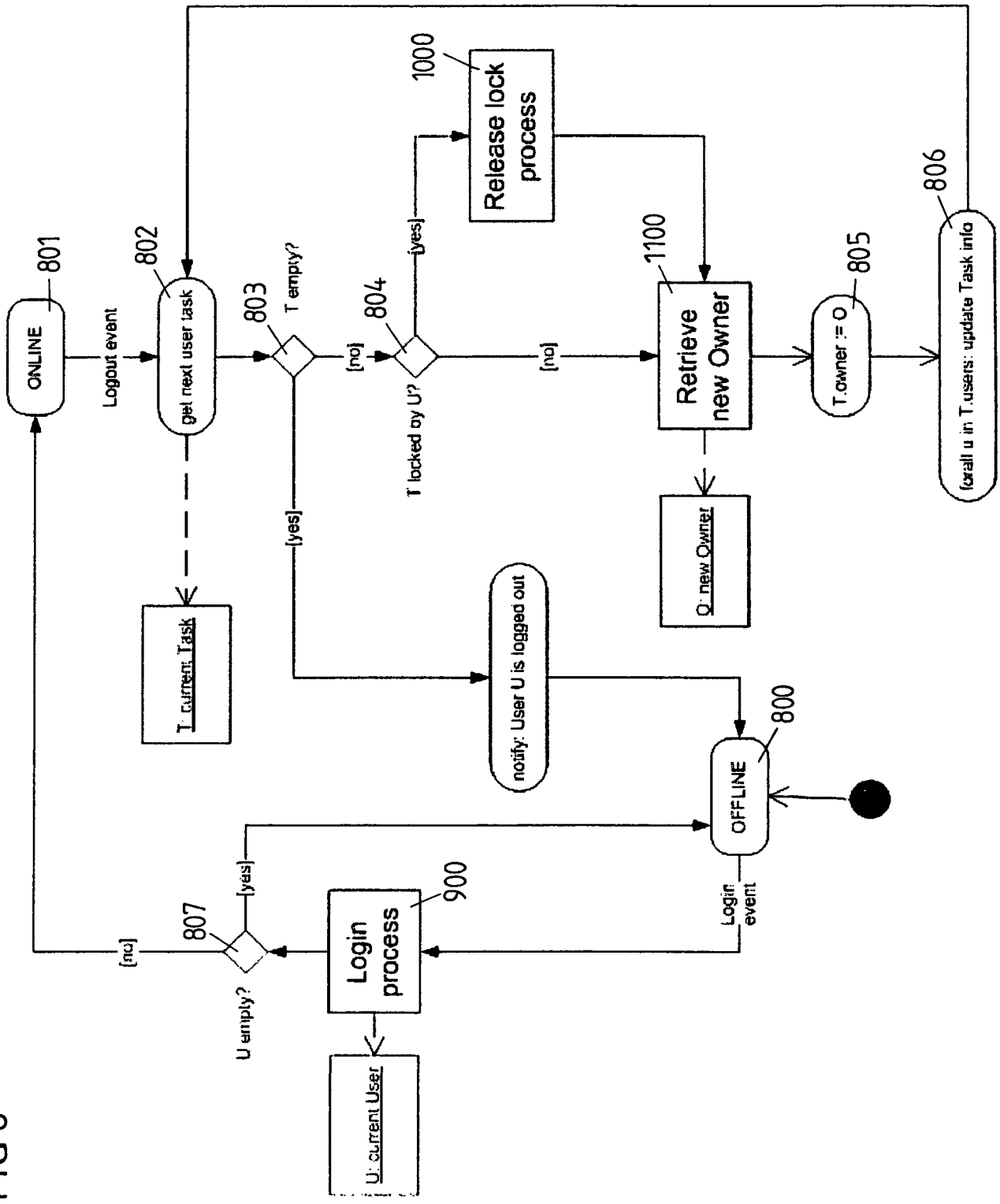

ONLINE
801
Logout event
802
get next user task
803
T empty?
[no]
804
T locked by U?
[yes]
1000
Release lock process
[no]
1100
Retrieve new Owner
805
T.owner := 0
806
forall u in T.users: update Task info
Q: new Owner
T: current Task
[yes]
notify: User U is logged out
OFFLINE
800
Login event
[yes]
807
U empty?
[no]
Login process
900
U: current User

FIG 8

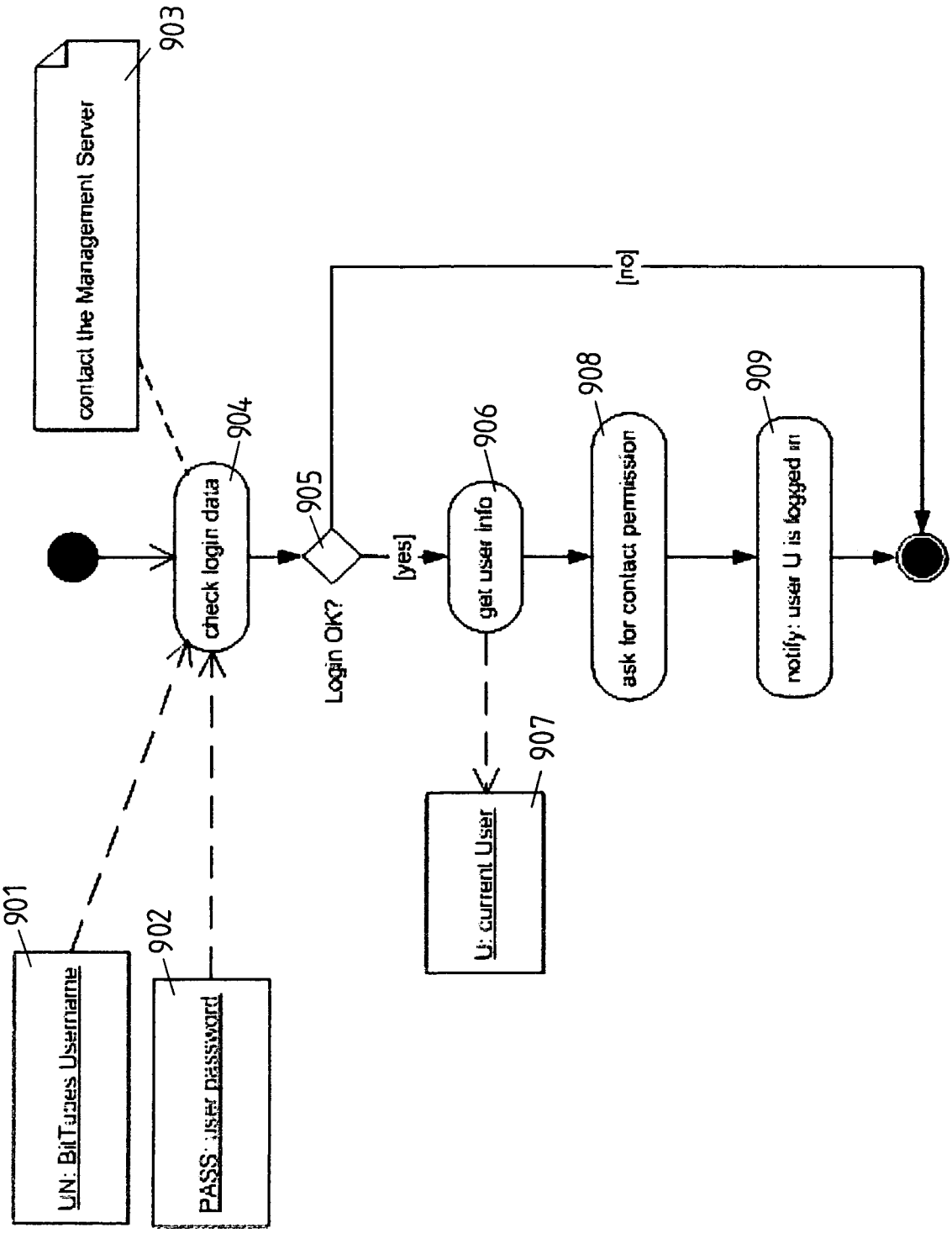
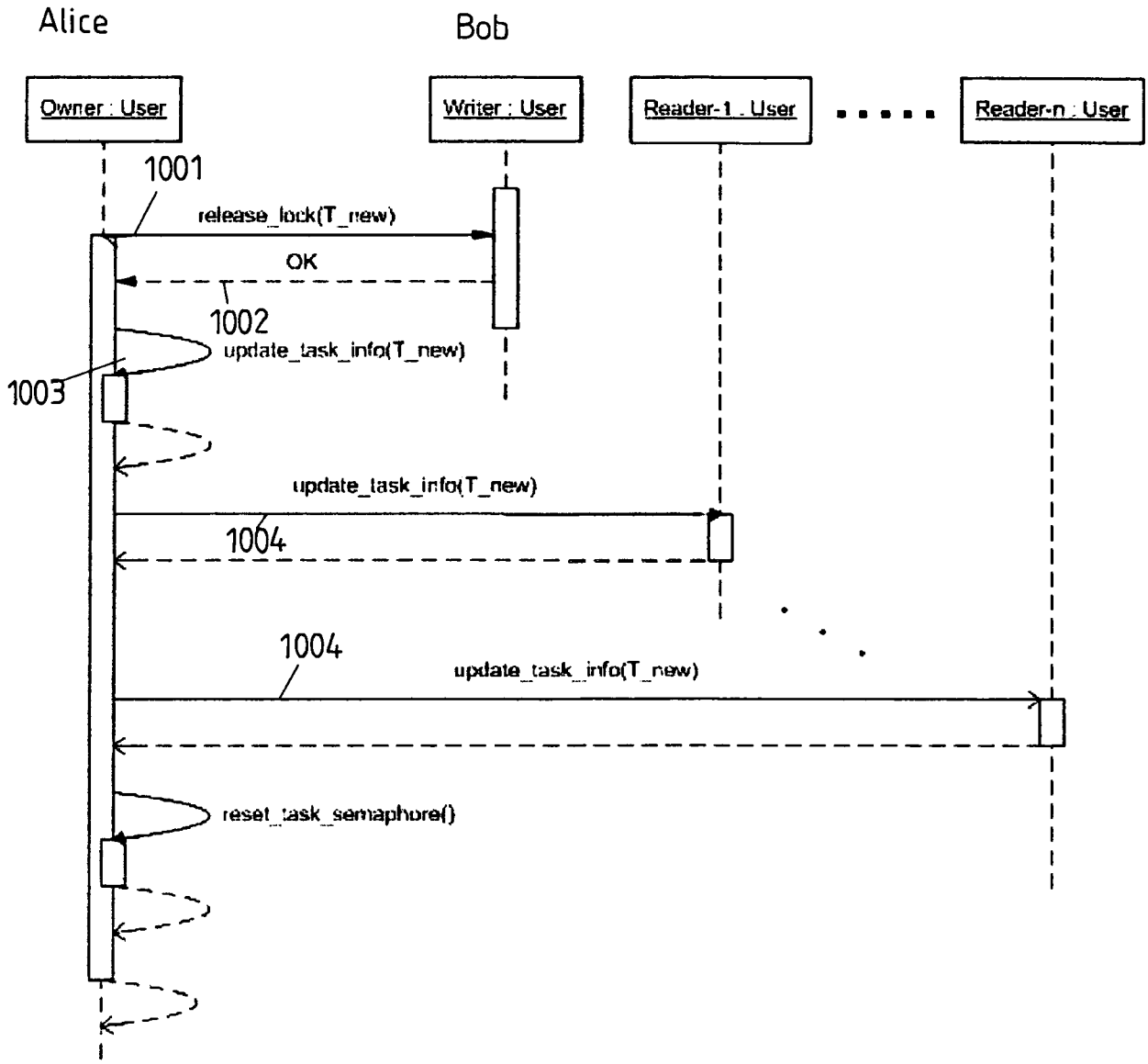

contact the Management Server
903
check login data
904
Login OK?
905
[yes]
[no]
get user info
906
ask for contact permission
908
notify: user U is logged in
909
U: current User
907
UN: BitTunes Username
901
PASS: user password
902

FIG 9

FIG 10


Alice
Bob
Owner : User
Writer : User
Reader-1 : User
Reader-n : User
1001
release_lock(T_new)
OK
1002
update_task_info(T_new)
1003
update_task_info(T_new)
1004
1004
update_task_info(T_new)
reset_task_semaphore()

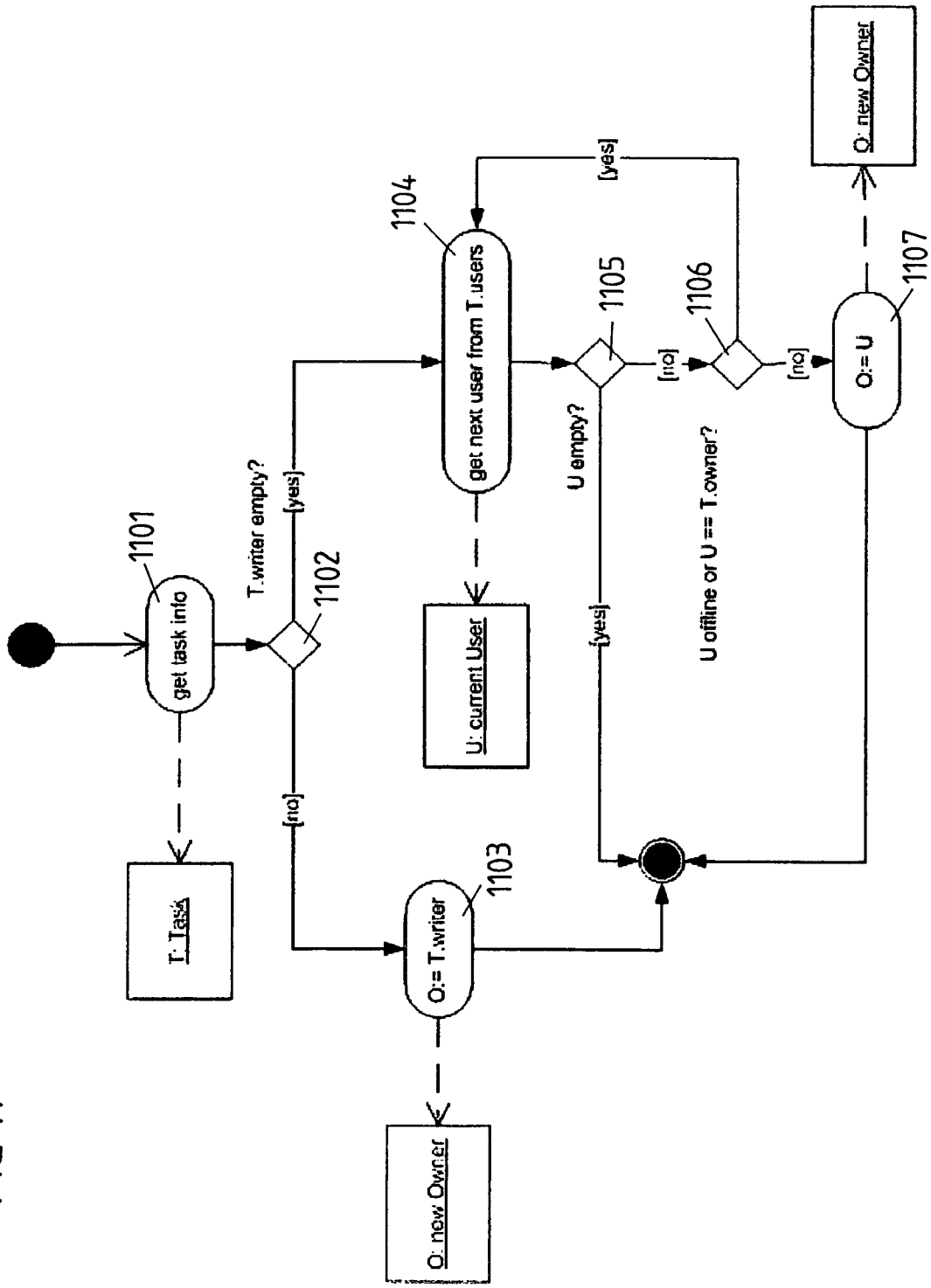

1101
get task info
T: Task
T.writer empty?
[yes]
[no]
1102
1103
O:= T.writer
O: new Owner
1104
get next user from T.users
U: current User
U empty?
[yes]
[no]
1105
U offline or U == T.owner?
[yes]
[no]
1106
O:= U
1107
O: new Owner

FIG 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/002774

A CLASSIFICATION OF SUBJECT MATTER
INV. G06Q10/00 H04L29/06

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

B FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06Q 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

C DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document with indication where appropriate of the relevant passages	Relevant to claim No
X	US 2004/049515 Al (HAFF MAURICE W [US] ET AL) 11 March 2004 (2004-03-11) figures 1,20 paragraphs [0005], [0064], [0068], [0069], [0072], [0073], [0084], [0125], [0130], [0131], [0147]	1-23
X	WO 03/048966 A (WEBXCENTRIC HOLDINGS PTY LTD [AU]; BUSINESS WEBS INTERNAT PTY LTD [AU]) 12 June 2003 (2003-06-12) figures 4,5 page 13, line 17 - page 15, line 29	1-23
X	US 6 230 185 Bl (SALAS PITO [US] ET AL) 8 May 2001 (2001-05-08) figures 1,4,9-12 column 2, line 34 - column 3, line 45 column 11, line 47 - column 13, line 14	1-23

-/--

[X] Further documents are listed in the continuation of Box C

[X] See patent family annex

* Special categories of cited documents

'A' document defining the general state of the art which is not considered to be of particular relevance

[1]E" earlier document but published on or after the international filing date

'L- document which may throw doubts on prio πty cla ιm(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

[1]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 cannol be considered to involve an inventive step when the document is taken alone

[1]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 docu¬ments 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	Date of mailing of the international search report
6 November 2009	24/11/2009

Name and mailing address of the ISA/	Authorized officer
European Patent Office P B 5818 Patentlaan 2 NL - 2280 HV Ri|swi|k Tel (+31-70) 340-2040 Fax (+31-70) 340-3016	Tyszka, Krzysztof

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/002774

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 801 700 A (FERGUSON GREGORY J [US]) 1 September 1998 (1998-09-01) figures 1-3,5 column 3, line 33 - line 49 column 3, line 62 - column 4, line 15 column 5, line 9 - column 6, line 32 column 7, line 1 - line 15	1-23
A	US 2005/144308 A1 (HARASHIMA ICHIRO [JP] ET AL) 30 June 2005 (2005-06-30) figures 1,3,8-10 paragraphs [0017], [0027], [0028], [0030], [0036], [0042], [0067]	1-23

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2009/002774

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004049515 A1	11-03-2004	US 2009205026 A1	13-08-2009
WO 03048966 A	12-06-2003	NONE	
us 6230185 B1	08-05-2001	US 6314408 B1	06-11-2001
		US 7127501 B1	24-10-2006
		US 6233600 B1	15-05-2001
us 5801700 A	01-09-1998	NONE	
us 2005144308 A1	30-06-2005	JP 2005189995 A	14-07-2005