

Internal distribution code:

- (A) [-] Publication in OJ
- (B) [-] To Chairmen and Members
- (C) [-] To Chairmen
- (D) [X] No distribution

**Datasheet for the decision
of 16 October 2019**

Case Number: T 2139/16 - 3.5.07

Application Number: 04003415.9

Publication Number: 1452981

IPC: G06F17/30

Language of the proceedings: EN

Title of invention:

A method to delay locking of server files on edit

Applicant:

Microsoft Technology Licensing, LLC

Headword:

Delayed locking/MICROSOFT TECHNOLOGY LICENSING

Relevant legal provisions:

EPC Art. 56, 123(2)
RPBA Art. 13(1), 13(3)

Keyword:

Amendments - added subject-matter - main request and first
auxiliary request (yes)
Inventive step - second, third and fourth auxiliary requests
(no)
Late-filed request - fifth auxiliary request (not admitted)



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 2139/16 - 3.5.07

D E C I S I O N
of Technical Board of Appeal 3.5.07
of 16 October 2019

Appellant: Microsoft Technology Licensing, LLC
(Applicant) One Microsoft Way
Redmond, WA 98052 (US)

Representative: Grünecker Patent- und Rechtsanwälte
PartG mbB
Leopoldstraße 4
80802 München (DE)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted on 26 April 2016
refusing European patent application No.
04003415.9 pursuant to Article 97(2) EPC

Composition of the Board:

Chairman R. Moufang
Members: R. de Man
M. Jaedicke

Summary of Facts and Submissions

I. The applicant (appellant) appealed against the decision of the Examining Division refusing European patent application No. 04003415.9.

II. The decision was issued on EPO Form 2061 and referred for its reasons to two communications dated 16 October 2015 and 8 April 2016, which dealt with a main request and an auxiliary request, respectively. These communications referred to the following documents:

D1: M. West: "VICE File System Services", Technical Report CMU-ITC-84-020, 7 August 1984, retrieved from <http://reports-archive.adm.cs.cmu.edu/anon/usr0/anon/usr/ftp/itc/CMU-ITC-020.pdf>;

D2: US 5 615 373 A, published on 25 March 1997;

D3: US 5 774 717 A, published on 30 June 1998;

D4: US 6 240 414 B1, published on 29 May 2001.

The Examining Division decided, *inter alia*, that claims 1 and 11 of the main request lacked inventive step over document D1 and that the subject-matter of claims 1 and 6 of the auxiliary request lacked inventive step over document D2.

III. In its statement of grounds of appeal, the appellant resubmitted copies of the claims of the main request and the auxiliary request.

IV. In a communication accompanying the summons to oral proceedings, the Board introduced the following documents:

D5: "ITC File System Design", Technical Report CMU-ITC-83-032, 29 September 1983, retrieved from <http://reports-archive.adm.cs.cmu.edu/anon/usr0/anon/usr/ftp/itc/CMU-ITC-032.pdf>;

D6: D. Whaley: "Re: How does one manage concurrency issues with JDBC?", 14 March 2001, retrieved from <https://www.mail-archive.com/servlet-interest@java.sun.com/msg33860.html>.

It expressed the preliminary view that neither request complied with Articles 84, 123(2) and 56 EPC.

V. In a letter dated 16 September 2019, the appellant relabelled the auxiliary request as the first auxiliary request and filed second, third and fourth auxiliary requests.

VI. During oral proceedings held on 16 October 2019, the appellant filed a fifth auxiliary request. At the end of the oral proceedings, the chairman pronounced the Board's decision.

VII. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the claims of the main request or, in the alternative, one of the first to fifth auxiliary requests.

VIII. Claim 1 of the main request reads as follows:

"A method for delaying locking of a file being opened on a computing device, the method comprising:

receiving (300) a request from a user to open said file;

in response to the request, opening (310) said file for reading in a read-only mode such that the file remains unlocked;

determining (320) that an editing trigger event has occurred with respect to the user and the open file, where determining (320) that an editing trigger event has occurred comprises determining that the user has had said file open for a pre-specified period of time; and

in response to the editing trigger event, obtaining (330) a lock on said file so as to change the file from the read-only mode to a read/write mode."

IX. Claim 1 of the first auxiliary request reads as follows:

"A method for delaying locking of a file being opened, the method comprising:

receiving (410) a request from a user to open said file, where said file is located on a server computer, the user is using a user computer, and said server computer and said user computer are operably connected;

in response to the request, creating a local user copy of said file and opening (420) said local user copy for editing;

determining (430) that an editing trigger event has occurred with respect to the user and the open local user copy, where determining (430) that an editing trigger event has occurred comprises determining that the user has had said local user copy open for a pre-specified period of time; and

in response to the editing trigger event, determining (430) if said file has been changed since said local user copy was made by comparing version information of a current version of said file on said server computer with version information of said local user copy; and

if said file has not been changed, obtaining (330) a lock on said file, and

if said file has been changed, offering one or more conflict resolution options to said user."

X. Claim 1 of the second auxiliary request reads as follows:

"A method for delaying locking of a file being opened on a computing device, the method comprising:

receiving (300) a request from a user to open said file;

in response to the request, opening (310) said file for reading in a read-only mode without locking the file;

determining (320) that an editing trigger event has occurred, where determining (320) that an editing trigger event has occurred comprises determining that the user has had said file open for a pre-specified period of time; and

in response to the editing trigger event, obtaining (330) a lock on said file, wherein said obtaining (330) a lock on said file comprises changing the mode from the read-only mode to a read/write mode for said user."

XI. Claim 1 of the third auxiliary request differs from claim 1 of the second auxiliary request in that its first three paragraphs have been replaced with:

"A method for delaying locking of a file located on a server computer, the method comprising:

receiving (300) a request from a user to open said file, the user using a user computer;

in response to the request, opening (310) said file directly from the server computer in a read-only mode for the user without locking the file;"

Dependent claim 3 reads as follows:

"The method of claim 1, where said step of opening (310) said file comprises creating a user copy of said file."

XII. Claim 1 of the fourth auxiliary request differs from claim 1 of the first auxiliary request in that "with respect to the user and the open local user copy" has been deleted.

XIII. Claim 1 of the fifth auxiliary request reads as follows:

"A method for delaying locking of a file located on a shared server computer, the method comprising:

receiving (300), by the server computer, a request from a user to open said file, the user using a user computer;

in response to the request, opening (310) said file at the server computer in a default read-only mode for the user without locking the file and without denying other users access to the file,

determining (320) that an editing trigger event has occurred, where determining (320) that an editing trigger event has occurred comprises determining that the user has had said file open for a pre-specified period of time; and

in response to the editing trigger event, obtaining (330) a lock on said file, wherein said obtaining (330) a lock on said file comprises changing the mode from the read-only mode to a read/write mode for said user, and locking any other users out of obtaining editing privileges for said file."

XIV. The appellant's arguments, where relevant to the decision, are discussed in detail below.

Reasons for the Decision

1. The appeal complies with the provisions referred to in Rule 101 EPC and is therefore admissible.
2. *The invention*
 - 2.1 The application relates to providing synchronised file access to multiple users.
 - 2.2 Paragraph [0004] of the background section of the application as filed explains that a file opened for reading is generally opened with read-only permission, which allows the user to access the file's data but not

to change it. A file opened for editing is opened with read/write permission, which allows the user not only to access the file's data but also to change it. When a file is opened with read/write permission, the file is locked to ensure that only one user at a time can modify its data.

- 2.3 According to paragraph [0006] of the description, the invention is directed to a "modified file open" action. Initially, the file is opened without being locked, which does not deny other users access to the file. When the user indicates that editing should occur, the file is locked. The claims refer to this as "delaying locking".

Main request and first auxiliary request

3. *Added subject-matter*

- 3.1 The feature of claim 1 of the main request "opening said file for reading in a read-only mode such that the file remains unlocked", which requires that the file was unlocked before it was opened and remains unlocked afterwards, does not have a basis in the application as filed.

The application discloses, in paragraph [0006], that the file is initially opened (for reading) "without locking the file". The same can be inferred from paragraph [0027] ("In this way more than one user may access the file and use it"). But opening a file without locking ensures neither that the file was unlocked when it was opened nor that it remains unlocked.

3.2 Claim 1 of the main request contains the feature "determining that an editing trigger event has occurred with respect to the user and the open file". Likewise, claim 1 of the first auxiliary request contains the feature "determining that an editing trigger event has occurred with respect to the user and the open local user copy".

Whereas "determining that an editing trigger event has occurred" was present in original claim 1, the further limitations "with respect to the user and the open file" and "with respect to the user and the open local user copy" have no apparent basis in the application as filed. The only basis that has been indicated by the appellant, in its letter of 13 August 2007, is original claims 2 and 3, but neither claim discloses these limitations.

3.3 The above objections were raised in the Board's communication and have been addressed by the appellant in its second auxiliary request (amending the main request) and its fourth auxiliary request (amending the first auxiliary request). The appellant has not provided specific counterarguments against the objections.

3.4 The Board therefore concludes that both the main request and the first auxiliary request infringe Article 123(2) EPC.

Second and third auxiliary requests

4. *Claim interpretation*

4.1 Claim 1 of the second auxiliary request is directed to a method "for delaying locking" of a file being opened on a computing device.

First, a request to open the file is received from a user. In response, the file is opened for reading in read-only mode "without locking the file".

When it is determined that an "editing trigger event" has occurred, a lock on the file is obtained, "wherein said obtaining a lock on said file comprises changing the mode from the read-only mode to a read/write mode for said user".

The claim further specifies that the editing trigger event takes the form of the user having had the file open for a pre-specified period of time.

4.2 Claim 1 of the third auxiliary request adds the following to claim 1 of the second auxiliary request:

- the file is located on a server computer;
- the user uses a user computer;
- the file is opened "directly from the server computer" in read-only mode "for the user".

Claim 1 of the third auxiliary request no longer states that the file is opened "for reading", but this is still implied by "in a read-only mode". The claim is hence strictly narrower in scope than claim 1 of the second auxiliary request.

4.3 The appellant argued that the steps of "opening", "determining" and "obtaining" were carried out by the same entity, which, in the third auxiliary request, was the server computer. This entity implemented the "modified file open" action referred to in paragraph [0006] of the description. This was not a regular file open action to open a file in read-only mode but one that also involved (i) determining that an editing trigger event had occurred and (ii) obtaining a lock on the file, comprising changing the access mode from read-only to read/write in response to the event.

4.4 However, the wording of claim 1 in both the second and third auxiliary requests does not rule out the possibility that the initial request received from the user is a regular request to open a file in read-only mode and that, as a separate action in response to an editing trigger event, a lock on the file is obtained and the access mode is changed to read/write.

In this respect, the Board notes that the claims do not define with any precision what kind of "request" is received and by what entity. For example, it could be a request to open a file received directly from the user by a word processor application program, which subsequently instructs the user computer's operating system to open the file. Or the request of claim 1 could be the instruction received by the operating system from the word processor application program. Or it could be a file-open request issued to a remote file server by the user computer's operating system in response to the instruction received from the application program.

Likewise, the claims do not specify the precise kind of mechanism used, in response to the editing trigger

event, to obtain the lock and change the access mode to read/write. It could be the application program that obtains the lock by means of an instruction to the operating system and changes the access mode by means of the same or another instruction. Or it could be the operating system or a remote file server that obtains the lock.

- 4.5 The Board further notes that the description does not support the appellant's position, either. Paragraph [0006], on which the appellant primarily bases its arguments, reads as follows:

"A server is implemented with a modified file open action, which, when a user performs the modified open, initially opens a file without locking the file. When a user indicates (either explicitly or implicitly) that the user is attempting to or intending to open the file, the file can then be locked for editing. In this way, the default action when a user requests a file is to open the file without denying other users access to the file. Then, when the user indicates that editing should occur, the lock for the file is obtained."

This passage does not unambiguously disclose that the "modified file open action" implemented in the server corresponds to a single request received by one entity that causes the entity to open the file in read-only mode, determine that an editing trigger event has occurred, and obtain a lock on the file while changing the access mode from read-only to read/write. In fact, it refers to opening the file in read-only mode as "the default action", which suggests that the operations performed later are part of a separate action.

Nor can the Board identify a passage in the detailed description that supports the appellant's reading of the claim. On the contrary, paragraph [0029] discloses that "[l]ocking of the file only occurs upon an indication that the user intends to edit the file". According to paragraph [0030], this intent to edit may be indicated by the user "through a specialized 'edit' command incorporated into the application, or when the user requests that changes already made be saved". Hence, at least in some embodiments, the "editing trigger event" is determined by a user application program, which then somehow indicates to either the user computer's operating system or the server that the file is to be prepared for editing. Likewise, paragraphs [0032] and [0037] discuss schemes whereby, in response to the editing trigger event, the file on the server is compared with a local copy on the user's computer before being locked. The detailed description is therefore inconsistent with claim 1 being construed to mean that all steps are carried out by a single server entity in response to a single request.

4.6 The Board will therefore interpret claim 1 of both the second and third auxiliary requests as encompassing methods consisting of two separate actions: a first action in which, in response to a user request, a file is opened in read-only mode without the file being locked, and a second action in which, in response to an editing trigger event, the file is locked and the access mode is changed to read/write.

4.7 The feature of claim 1 "wherein said obtaining a lock on said file comprises changing the mode from the read-only mode to a read/write mode for said user" is based on original dependent claim 2. It is supported by paragraphs [0012] ("... is the lock on the file

obtained, making the file read/write for the user") and [0038] ("when a user wishes to modify the file, the lock is then obtained for the user, and the user will have read/write privileges for the file"), but neither passage explains the mechanism by which the act of obtaining the lock results in the - in principle unrelated - act of changing the access mode from read-only to read/write.

The Board will therefore interpret this feature of claim 1 as meaning that both the lock is obtained and the file access mode is changed from read-only to read/write and as leaving open how this is implemented.

- 4.8 As to the feature in claim 1 of the third auxiliary request specifying that the file is opened "directly from the server computer", the appellant argued that this was to be understood as meaning that no local copy of the file was made on the user computer.

Since this interpretation has a basis in the application as filed, e.g. in the sentence in paragraph [0035] stating that "[i]n one embodiment, the file is opened directly from the server in read-only mode; in an alternate embodiment, a copy of the file is made and opened", the Board will adopt it for the purpose of assessing inventive step.

5. *Inventive step*

- 5.1 It is expedient to first assess inventive step for claim 1 of the third auxiliary request, which is strictly narrower in scope than claim 1 of the second auxiliary request (see point 4.2 above).

5.2 In its communication, the Board argued that the subject-matter of claim 1 of the main request lacked inventive step over both document D5 and the acknowledged prior art disclosed in paragraphs [0002] and [0003] of the present application's description. However, in both of these starting points a local copy is made of the file that is opened in read-only mode. The file is therefore not opened "directly from the server computer" as now required by claim 1 of the third auxiliary request (see point 4.8 above).

Nevertheless, the basic argument behind the inventive-step reasoning presented in the Board's communication still applies.

5.3 At the priority date, opening a file shared by multiple users in read-only mode without locking was known. This can be inferred from paragraph [0004] of the background section of the application (see point 2.2 above), which contrasts opening a file in read-only mode with opening the file in read/write mode. In the latter case, the file is locked to prevent conflicting modifications by two or more users.

It is also apparent from paragraph [0004] that locking and opening a file in read/write mode was part of the same prior art.

At the oral proceedings, the appellant contested neither the Board's reading of paragraph [0004] nor the existence of the prior art acknowledged there.

5.4 The subject-matter of claim 1 of the third auxiliary request differs from this prior art in that:
(a) the shared file is located on a server computer, which the user accesses through a user computer;

- (b) the file is opened in read-only mode "directly from the server";
- (c) after the file has been opened in read-only mode, a determination is made that an "editing trigger event" has occurred;
- (d) in response to the editing trigger event, a lock on the file is obtained, "wherein said obtaining a lock comprises changing the mode from the read-only mode to a read/write mode for said user"; and
- (e) determining that an editing trigger event has occurred comprises "determining that the user has had said file open for a pre-specified period of time".

5.5 At the priority date in 2003, it was well known to access shared files stored on a server computer from a user computer in accordance with distinguishing feature (a), for example in the context of a web-based system as described in paragraphs [0002] and [0003] of the background section of the application.

5.6 At the oral proceedings, the appellant did not contest that opening a file in read-only mode "directly from the server" in accordance with feature (b), i.e. without making a local copy (see point 4.8 above), was also known and can therefore not support an inventive step.

5.7 Opening a file in read-only mode if there is currently no intention to edit the file and reopening the file in read/write mode when an intention to edit the file arises is a common scenario and would therefore have been an obvious way of operating a prior-art system with shared files that allowed files to be opened with read-only or read/write permission.

This means that, after the file had been opened in read-only mode, it would have been obvious, in response to an indication that the user now wishes to edit the file (i.e. an "editing trigger event"), to close the file and to lock and reopen it in read/write mode.

Distinguishing features (c) and (d) are implied by this obvious scenario. Indeed, closing a file followed by locking and reopening the file implies both "obtaining the lock" and "changing the file access mode for the user from read-only to read/write mode".

- 5.8 Distinguishing feature (e) expresses that an "intent to edit" is assumed to exist when the user has had the file open for a pre-specified period of time (see paragraph [0030]). This prediction of the user's "intent to edit" can either take the place of or supplement an explicit expression of the user's intent, such as the user inputting an "edit" command.

The Board agrees with the Examining Division that the idea of predicting that a user who has had a file open for a predetermined period of time now has the intention to edit the file is non-technical and that implementing that idea would not require inventive skill.

- 5.9 In sum, the subject-matter of claim 1 of both the second and the third auxiliary requests lacks inventive step (Article 56 EPC).

Fourth auxiliary request

6. *Claim interpretation*

6.1 Claim 1 of the fourth auxiliary request is also directed to a method "for delaying locking" of a file being opened, but it deviates in several respects from claim 1 of the second and third auxiliary requests.

First, a request to open the file, which is located on a server computer, is received from a user who is using a user computer. In response, a "local user copy" of the file is created, and this local user copy is opened "for editing".

When it is determined that an "editing trigger event" has occurred, version information of the current version of the file on the server computer is compared with version information of the local user copy to determine whether the file on the server computer has been changed (by another user) since the local user copy was made.

If the file on the server computer has not been changed, a lock on the file is obtained.

If the file on the server computer has been changed, one or more conflict resolution options are offered to the user.

The claim further specifies that the editing trigger event takes the form of the user having had the local user copy open for a pre-specified period of time.

6.2 In its communication, the Board pointed out that claim 1 allowed for different interpretations in

respect of the feature "opening said local user copy for editing" and the determination of the "editing trigger event".

In a first interpretation, editing the local user copy (or an attempt to edit it) is itself an editing trigger event resulting in (an attempt to) obtain a lock on the file. In this interpretation, the "editing trigger event" refers to editing of the local user copy.

In a second interpretation, the "editing trigger event" refers to editing of the file on the server computer, which means that the local user copy can be edited without the event being triggered.

6.3 Paragraph [0032] of the application refers to "substantial changes" that a user may have made to the local user copy before the file on the server is locked. Paragraph [0037] likewise considers the case that "the user has made changes to the local copy of the file" before the lock is obtained. Both passages support the second interpretation.

Other passages in the description, however, apparently assume that no changes are made to the local copy before the editing trigger event occurs, which is in line with the first interpretation. For example, paragraph [0031] states that when a user A triggers the locking by demonstrating an intent to edit, "it must be ensured that the version of the file currently on the server is the same as that which User A is viewing and intending to edit".

6.4 In view of the Board's finding of lack of inventive step below, it is not necessary to decide whether the above-mentioned ambiguity renders claim 1 unclear.

Instead, the Board will adopt the second interpretation for the purpose of assessing inventive step.

7. *Inventive step*

7.1 The background section of the application discloses that, in a collaboration environment, files located on a server computer can be accessed by multiple users (paragraph [0002]). A user can either create a local copy of the file without altering the original file or open the file on the server for editing (paragraph [0003]). In the latter case, the file is locked, and no other users are allowed to edit the file.

7.2 The subject-matter of claim 1 differs from this acknowledged and uncontested prior art in that:

- (a) the local copy of the file is opened "for editing";
- (b) in response to an "editing trigger event", it is determined whether the file has been changed since the local copy was made by comparing version information;
- (c) if the file has not been changed, a lock is obtained on the file;
- (d) if the file has been changed, one or more conflict-resolution options are offered to the user; and
- (e) determining that an editing trigger event has occurred comprises "determining that the user has had said file open for a pre-specified period of time".

These distinguishing features solve the problem of allowing multiple users to edit the content of the file simultaneously.

7.3 The Board agrees with the Examining Division that, at the priority date of the application, the skilled

person was well aware of the concept of "optimistic locking" or "optimistic concurrency control", whereby users wishing to access a shared resource for modification are granted access to the resource on the "optimistic" assumption that no conflicting accesses will take place, and the file is locked and conflicts are checked for only when the modifications are about to be written back to the resource.

Evidence of this common general knowledge is provided by document D4, which, in column 2, lines 18 to 26, discloses that Microsoft Exchange allows any number of users to open and modify the same record in a distributed information store. When a user attempts to save the modified record, it is determined whether the record has been changed by another user. If it has not been changed, the modified record is saved. If it has been changed, conflict-resolution options are offered to the user. See also the passage in column 5, lines 46 to 56, which confirms that conflicting updates may be detected by comparing version information.

Document D6, which explains the concept of "Optimistic Record Locking" on page 1, provides further evidence of this common general knowledge.

7.4 Starting from the acknowledged prior art and faced with the problem of allowing multiple users to edit the content of the file simultaneously, the skilled person would therefore have considered implementing the "optimistic locking" concept. In the context of the acknowledged prior art, this means that users wishing to edit the file are allowed to access the current version of the file to create a local copy for editing (feature (a)). When the modifications made to the local copy are to be written back to the server, it is

determined, by comparing version information, whether in the meantime the file has been changed by another user (feature (b)). If the file has been changed, conflict-resolution options are offered to the user (feature (d)). If the file has not been changed, the modifications are written back to the server, which involves locking the file and opening it in write-only mode (feature (c)).

- 7.5 Feature (e) expresses that the "intent to edit", which now signals that the user wishes to write the modifications made to the local copy back to the server, is assumed to exist when the user has had the file open for a pre-specified period of time (see paragraph [0030]).

The Board again takes the view that the idea of predicting that a user who has had a file open for a predetermined period of time now has the intention to write modifications back to the server is non-technical and that implementing that idea would not require inventive skill. The appellant did not argue otherwise.

- 7.6 Hence, the subject-matter of claim 1 lacks inventive step over the acknowledged prior art and the common general knowledge as evidenced by documents D4 and D6 (Article 56 EPC).

Fifth auxiliary request

8. *Admission into the appeal proceedings*

- 8.1 The fifth auxiliary request, which amends the third auxiliary request, was filed during the oral proceedings before the Board. Its admission is

therefore at the Board's discretion under Article 13(1) and (3) RPBA.

8.2 The appellant explained that the amendments made in the fifth auxiliary request were intended to clarify that the invention's "modified file open action" was implemented fully at the server computer, as it had already argued in respect of the second and third auxiliary requests (see point 4.3 above). This was now expressed, in particular, by the amended features "receiving, by the server computer, a request from a user to open said file" and "in response to the request, opening said file at the server computer in a default read-only mode".

8.3 Although the claim now specifies that the request to open the file is received "by the server computer" and that the file is opened "at the server computer", these amendments *prima facie* do not imply that the steps of "determining" and "obtaining" are also carried out by the server computer and in response to the same request. Moreover, for the reasons given in point 4.5 above, the Board has serious doubts that the application as filed supports the interpretation put forward by the appellant.

8.4 Since the fifth auxiliary request was filed at a late stage in the proceedings and, at least on a *prima facie* assessment, does not achieve what the appellant intended, the Board exercises its discretion under Article 13(1) and (3) RPBA to not admit the request into the appeal proceedings.

Conclusion

9. Since none of the requests admitted into the appeal proceedings is allowable, the appeal is to be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



I. Aperribay

R. Moufang

Decision electronically authenticated