Internal distribution code:

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

Datasheet for the decision of 27 June 2013

Case Number: T 0540/10 - 3.5.01

Application Number: 05826522.4

Publication Number: 1831803

IPC: G06F17/30

Language of the proceedings: EN

Title of invention:
SYSTEM AND METHOD FOR MANAGING BINARY LARGE OBJECTS

Applicant:
Nuance Communications Austria GmbH

Headword:
Managing BLOBs/NUANCE COMMUNICATIONS

Relevant legal provisions:
EPC 1973 Art. 56

Keyword:
Inventive step - (no)

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 0540/10 - 3.5.01

D E C I S I O N
of Technical Board of Appeal 3.5.01
of 27 June 2013

Appellant: (Applicant)	Nuance Communications Austria GmbH Triester Strasse 64 1101 Wien (AT)
Representative:	Driver, Virginia Rozanne Page White & Farrer Bedford House John Street London WC1N 2BF (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 27 October 2009 refusing European patent application No. 05826522.4 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman: S. Wibergh
Members: P. Scriven
P. Schmitz

Summary of Facts and Submissions

I. The appeal is against the Examining Division's decision to refuse European patent application 05826522.4. The impugned decision cited a lack of clarity (Article 84 EPC) and of inventive step (Article 56 EPC 1973) in the light of document WO 01/48638 (D2).

II. With the statement setting out its grounds of appeal, the appellant requested that the impugned decision be set aside, that a patent be granted on the basis of a newly filed main request or else of a newly filed auxiliary request, and that oral proceedings be held if either of the main and auxiliary requests were not considered allowable.

III. The Board summoned the appellant to oral proceedings, to be held on 27 June 2013. In a communication sent with the summons, the Board raised questions of clarity (Article 84 EPC) and of added subject matter (Article 123(2) EPC).

IV. The appellant responded by letter dated 24 May 2013, in which it argued in favour of clarity, and informed the Board that it would not be represented at oral proceedings. With this letter, the appellant submitted a second auxiliary request, and further requested that the Board consider the patentability of claims 4, 5, 9 and 10 according to the main, first and second auxiliary requests.

V. The appellant's requests, therefore, are that the decision under appeal be set aside and that a patent be granted on the basis of the main or first auxiliary requests, filed with the statement setting out the grounds of appeal, dated 19 February 2010; or the second

auxiliary request, file with the letter dated 24 May 2013; and, as a further precautionary measure, that the Board consider the patentability of claims 4, 5, 9 and 10 of the main and auxiliary requests.

VI. Claims 1, 4 and 5 according to the main request read as follows.

1. A system (10) for managing Binary Large Objects (BLOBs), comprising:
a file subsystem (20) arranged to create, store, modify and/or delete files, wherein said system (10) is arranged for including a specified BLOB (50) associated with a BLOB name in a specified file (40) associated with a file name in said file subsystem (20); and
a database subsystem (30) arranged to ensure that ACID requirements are met,
wherein the system (10) is configured to:
start a database transaction (210) in the database subsystem (30);
retrieve (220) the file name for the specified file (40) comprising said specified BLOB (50);
create (23 0-250) a new file name;
associate (260) the new file name and the BLOB name in the database subsystem;
modify the specified BLOB (50) and write (270) the modified specified BLOB to a new file in the file subsystem (20), where said new file is associated with said new file name;
check (280) if all previous steps were successful; and

if the check (280) shows that the previous steps were successful, commit (290) said database transaction,

wherein said association between said file name and said BLOB name in the database subsystem is bijective, so that any unique file name is associated with one unique BLOB name and any unique BLOB name is associated with one unique file name, and

wherein the database subsystem (30) is arranged to store, modify, rename and/or delete said association between said file name and said BLOB name to maintain said bijective relationship within the database subsystem.

4. A system (10) according to any of the claims 1 to 3, wherein said file name comprises a globally unique identifier (GUID) and a version index.

5. A system according to claim 4, wherein said new file name comprises said globally unique identifier (GUID) and an updated version index.

VII. Claim 1 according to the first auxiliary request reads identically, except that the definition of the file subsystem has the following text added at the end:

... said file subsystem being configured such that ACID requirements are not ensured.

VIII. Claim 1 according to the second auxiliary request reads identically to that according to the main request, except

as shown here:

> ...
> *a database subsystem (30)arranged to ensure that ACID requirements are met, wherein the BLOB has a BLOB name within the database subsystem and the file has a file name within the file subsystem, the file name containing a path to the specified BLOB (50), wherein within the file subsystem all filenames are unique; and within the database subsystem all BLOB names are unique;*
> ...
> *associate (260) the new file name and the BLOB name in the database subsystem so that a relation exists between the new file name and the BLOB name in the database subsystem;*
> ....

IX. The appellant's arguments can be summarized as follows.

The statement in the final paragraph of page 5 of the published application, that "throughout this specification the term 'BLOB' (Binary Large OBject) is meant to cover any data object that due to its size cannot be handled efficiently by the database system," specifically used the term "meant to cover." The passage, therefore, did not define the meaning of the term, but only mentioned some things that should fall within the definition.

The Examining Division was wrong to consider the "BLOB handler" of D2 as a file system. According to D2, the BLOB storage was not separate from the database, but was part of it and under its control.

The BLOBs in D2 were stored in a single file, rather than

in individual files.

Filenames can change over time but BLOB names do not.

Reasons for the Decision

Introduction

1. The application concerns the storage of what are called "binary large objects" ("BLOBs"), and certain requirements which are imposed on the way data are stored. These are the "ACID" requirements: atomicity, consistency, isolation, and durability.

Atomicity is the requirement that a transaction either succeed as a whole, or fail as a whole. It should not be possible, for example, when data are to be modified, that some, but not all, of the changes are made.

Consistency is the requirement that any prescribed rules always hold. If, for example, there is a rule that any data designated as a "letter" must have a non-empty address, then it should not be possible for any transaction to produce a letter that has no address. That is, the data moves only from one valid state to another. Even transient invalid states are forbidden.

Isolation is the requirement that transactions do not interfere with one another. In effect, two transactions cannot both affect the same data at the same time.

Durability is the requirement that once a transaction is finished ("committed") then data will not be lost even in the event of a power outage, crash etc.

2. It is common for databases to enforce the ACID requirements. It is not normal for file systems to do so. However, as the application (for example in paragraphs 2 and 3 of the published application) explains, there are databases that do not enforce the ACID requirement, and file systems that do.

3. The application seeks to obtain the advantages of enforcing the ACID requirements for BLOBs. It would be possible to do that by storing the BLOBs in a database, but that is problematic: *Most standard databases provide some support for transactions involving BLOBs that can have sizes of up to tens or hundreds of MB, but they are not optimized for transactions involving hundreds of MB* (published application, page 2, line 21 ff).

4. The invention addresses the problem by storing the BLOBs not in a database, but in a file system. It proposes a way of combining a database with a file system so that the enforcement of ACID requirements in the former can be applied to BLOBs stored in the latter.

5. The basic idea is that BLOBs each have a unique name, and are stored in files that also each have unique names. There is a bijection between the two sets of names. The bijection might be stored in a table within the database, but that is not essential.

6. The claims of all requests are now limited to the handling of modifications. In brief: a new file name is created, it is associated with the name of the BLOB to be modified, the modified BLOB is written to the new file. Those operations are encapsulated within the

database's transaction structure.

The Main request

7. D2 discloses a system for managing BLOBs. The clarity of the term "BLOB" has been questioned, but there has been no dispute as to whether what D2 discloses falls within the definition.

8. The system disclosed by D2 comprises a database subsystem that enforces the ACID requirements (D2, pages 8, lines 11 - 13). This, too, has not been disputed.

9. The system disclosed by D2 also comprises a file subsystem. That has been disputed. The appellant has argued that the Examining Division was wrong to consider the "BLOB handler" of D2 as a file system. It was part of the database system rather than separate and distinct. The specially configured "contiguous storage" which D2 used for BLOBs was under the control of the database system. The Board does not accept that argument. The term "file system" is broad, and nothing in claim 1, or in the application as a whole, places any limits on what it may be, beyond being a data storage system. The examples of DOS FAT, NTFS, and Linux ext2 (published application, page 2, line 10) are mentioned only as examples of some standard file systems. In the Board's view, the separate, contiguous storage which D2 proposes for storing BLOBs counts as a file system. The appellant has also argued that the BLOBs in D2 are stored in a single file, rather than in individual files. The Board does not agree with that (see point 11, below) but the view that the BLOBs are stored in even a single file seems to suffice: if the BLOBs are contained within a file, there is a file

subsystem. Nor does the Board accept the appellant's conclusions from the observation that, according to D2, the BLOB storage is not separate from the database, and that storage is under the control of the database. That seems to be correct, but it is just as true of the file subsystem according to the invention.

10. The database subsystem disclosed in D2 is configured to start and run transactions, and to commit them only if successful. Otherwise the transaction is "rolled back" and the database remains in the state it was in before the transaction started. That is shown in Figure 4B.

11. When updating, the system disclosed in D2 (see Figure 4B) writes the updated blob data in a new location in the contiguous storage (see block 242). That involves the creation of a new header, and a new record in the allocation control table. If D2 does not call that a "file", that does not mean that it is not a file in the sense of the present application. In the Board's view, it is one. The BLOB is stored as a logical unit, accessed via the record in the allocation control table. The Board sees no difference between that and a file in DOS, accessed via the "file allocation table." Thus, the update transaction disclosed in D2 comprises finding the location of the present BLOB, via the allocation control table, the creation of a new file, and the writing of the modified BLOB to the new file.

12. What D2, apparently, does not disclose is the use of file names. The BLOB files seem to be identified only by an address. Nor does there seem to be any disclosure of BLOB names. It follows, naturally, that there is no association between the two sets of names.

13. However, in the disclosure of D2, there is an association between BLOBs and the files in which they are stored. That is just the nature of storage. When image X is stored in file Y, there is an association between X and Y. That association is necessarily bijective. X is stored in Y (and not anywhere else, although other files may store copies of X), and Y stores X and nothing else (a file stores what it stores).

14. Thus, the invention defined in claim 1 differs from the disclosure of D2 by the file and BLOB names, and by the bijective association between them.

15. The use of textual names to refer to data objects stored in different locations is commonplace. It is the Board's view that it would have been obvious to use them, when implementing D2. Such names are simply more convenient for users. However, the use of textual names need not preserve the bijective relationship between files and their contents. That is because a file, or its contents, can be given more than one name. Thus, the skilled person, implementing the system of D2 with textual file and BLOB names, has a choice. She can choose to maintain bijectivity, or not. Each has its advantages and disadvantages, and in the Board's view, the trade off would be routine. In the present case, however, the aim is to enforce ACID requirements, and that suggests maintaining bijectivity. For example, it is easier to ensure that two processes do not simultaneously try to modify a BLOB, if access is always via the same name. The skilled person, therefore, has a clear incentive to maintain bijectivity when providing textual names.

16. That is sufficient to show that the subject matter defined by claim 1 does not involve an inventive step.

17. The Board, however, also finds that the invention would have been obvious for other reasons. A database record might, for example, point to a BLOB called "Self_Portrait", which is stored in a file called "selfportrait.jpg." In the database, there would be a reference in the record, and a table which links such references to file names. The same reference could occur at many places in the database, but the filename is only stored once, in the table linking BLOB names to file names. A change in file name, to "selfportrait1.jpg" would be easy to administer: it would be enough to change the table entry, and no need to find all the records that refer to "Self_Portrait". None of that is claimed, but it does seem to be what the appellant has in mind when explaining that the file name is made of a globally unique identifier and a version number (claims 4 and 5; application as published, page 10, lines 5 - 14) and that filenames can change over time but BLOB names do not (letter dated 24 May 2013, page 3). The Board does not see such naming conventions as a technical issue. It is simply the act of a user choosing what, to him, are meaningful names. As to the use of separate names for BLOBs, and a table linking them to filenames, the Board regards only the use of a table in the database as technical. That it is used to link different names is not. As non-technical considerations, the use of separate BLOB and file names, and the file naming convention can be seen as constraints that the skilled person would have been required to adhere to, when she implemented the system of D2. In doing so, her only choice would have been in how the association between names is stored, and the use of a table in a database would have been no more

than using a table to do what tables are designed to do. That is, it would have been obvious. From this point of view, not only the system defined by claim 1, but also those defined by claims 4 and 5 would have been obvious.

The first auxiliary request

18. Claim 1 additionally defines the file subsystem as "being configured such that ACID requirements are not ensured." The Board finds that formulation ambiguous: the file subsystem alone might not ensure the requirements, or the system as a whole (due to some property of the file subsystem) does not ensure them. That is not simply academic, because the file system as set out in the application can be accessed via the database (which ensures the ACID requirements are met), but nothing prevents it being accessed directly. For example, if the file system were NTFS, which is one of the examples (published description, page 2, lines 10 - 12), then it would be possible to delete a BLOB directly, without passing through the database system. In the Board's view, then, claim 1 is unclear (Article 84 EPC).

19. However, the argumentation set out above, regarding inventive step for the main request makes no use of any properties of the file subsystem other than its ability to store files (with names). Whether or not it ensures ACID requirements makes no difference. Thus, the Board concludes that there is a lack of inventive step for the same reasons.

The second auxiliary request

20. As compared with the main request, claim 1 here defines the BLOB and file names as being within the database and file subsystems, respectively; the filename as containing a path to the BLOB; and all BLOB and file names as being unique. It also defines that the association between BLOB and file names is such that "a relation exists" between them.

21. As to the last of those points, the word "relation" does not seem to add anything that is not inherent in "association", and the Board considers that it makes no difference.

22. The choice of making all names unique is a matter of convenience for the user. Any storage system needs to identify the different items it stores. It needs to have some way of accessing item A rather than item B. The location at which items are stored provides a unique identifier. As noted in respect of the main request (see point 15., above), the skilled person has a choice, when providing names, to keep the uniqueness property, or not. The reasons already given apply equally to the second auxiliary request.

23. The Board, therefore, concludes that the subject matter defined by claim 1 does not involve an inventive step, so that the second auxiliary request cannot be allowed.

The request that the Board consider the patentability of claims 4, 5, 9 and 10

24. The Board has already considered the patentability of the subject matter defined in claims 4 and 5 according to the main request. See point 17., above. That

reasoning applies equally to claims 9 and 10 according to the main request, and carries over to the auxiliary requests.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

T. Buschek S. Wibergh

Decision electronically authenticated