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

Datasheet for the decision of 28 October 2010

Case Number:	T 0353/07 - 3.5.06
Application Number:	99961787.1
Publication Number:	1147469
IPC:	G06F 9/46
Language of the proceedings:	EN

Title of invention:
Method and apparatus for concurrent DBMS table operations

Applicant:
Computer Associates Think, Inc.

Opponent:
-

Headword:
Concurrent table unloading/COMPUTER ASSOCIATES THINK

Relevant legal provisions:
EPC Art. 123(2)

Relevant legal provisions (EPC 1973):
EPC Art. 84, 56

Keyword:
-

Decisions cited:
-

Catchword:
-

Case Number: T 0353/07 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06 of 28 October 2010

Appellant:

Computer Associates Think, Inc.
One Computer Associates Plaza
Islandia
New York 11749 (US)

Representative:

Dunlop, Hugh Christopher
R.G.C. Jenkins & Co.
26 Caxton Street
London SW1H 0RJ (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 20 September 2006 refusing European patent application No. 99961787.1 pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: D. H. Rees
Members: M. Müller
W. Sekretaruk

Summary of Facts and Submissions

I. The appeal lies against the decision of the examining division to refuse the European patent application 99961787.1. The decision was delivered during oral proceedings on 21 June 2006 and dispatched with letter dated 20 September 2006.

II. The decision refers to a main request as filed with letter of 17 June 2005, a first auxiliary request as filed with letter dated 18 May 2006, and a second auxiliary request as filed during oral proceedings.

III. In the decision, the examining division made use of only one document, namely:

D3: US 5 222 235

and concluded that claim 1 of the main request and the first auxiliary request violated Article 123(2) EPC, while claim 1 of the second auxiliary request violated Article 56 EPC in view of D3 and common knowledge about the best-fit algorithm.

IV. An appeal was filed by fax on 30 November 2006 and the appeal fee was paid on the same day. A statement of grounds of appeal was filed by fax on 30 January 2007. The appellant requested that the decision be set aside and that a patent be granted based on claims 1-20 according to a main request or a first (and only) auxiliary request, both as filed with the statement of grounds of appeal, or that the first auxiliary request be resubmitted to the examining division for further prosecution.

V. With letter dated 28 July 2010, the board summoned the appellant to oral proceedings to be held on 28 October 2010. In an annex to the summons the board expressed the preliminary opinion that the independent claims of both requests showed the required inventive step, Article 56 EPC 1973, but raised a number of additional objections under Article 123(2) EPC and 84 EPC 1973. Remittal to the examining division for further prosecution was indicated as a possible outcome of the appeal.

VI. With fax of 5 October 2010, the appellant filed new claims 1-20 to replace those according to the then main request.

VII. Oral proceedings before the board took place on 28 October 2010 as scheduled. During the oral proceedings, the appellant filed amended claims 1-20 to replace those of the pending auxiliary request and requested remittal to the first instance on the basis of amended main or auxiliary requests.

VIII. Claim 1 according to the main request reads as follows:

"A method for unloading database tables comprising the steps of:

determining a plural number of export directories, each export directory being located on a separate storage device;

launching (410) a number of threads limited to the number of export directories to process the database tables;

unloading the tables to create respective export files, including:

assigning (420) each of the tables to a corresponding one of said threads, using a best fit algorithm in which each thread checks all of the export directories and chooses and claims the smallest available directory that can accommodate the export file; and

unloading (440) the respective database tables by processes of the corresponding threads into the corresponding export directories."

Claim 7 according to the main request reads as follows.

"The method according to claim 1, further comprising loading data into database tables, comprising the steps of:

determining X threads for loading data into database tables;

creating X temporary tables (600), each temporary table corresponding to a set of data stored in one of said export directories;

launching (610) a structured query language process in conjunction with each thread for loading each temporary table;

loading each temporary table with the data stored in the corresponding export directory via the corresponding structured query language loader process."

Claim 11 according to the main request reads as follows.

"A computer readable medium having computer instructions stored thereon that, when loaded into a computer, cause the computer to perform the steps of any of claims 1 to 10."

Claim 12 according to the main request reads as follows.

"An apparatus for unloading database tables maintained in a system, comprising:

a symmetric multiprocessor computer having a plurality of processors, and a plurality of shared storage devices;

means for determining a number of export directories, each export directory being located on a separate storage device;

means for launching (400, 405, 410) a plurality of threads in the plurality of processors limited to the number of export directories to process the database tables;

means for unloading each of the tables to create an export file by assigning (420) each of the tables to a corresponding one of said threads, using a best fit algorithm in which each thread checks all of the export directories and chooses and claims the smallest available directory that can accommodate the export file, and

unloading (430) the respective database tables by processes of the corresponding threads into the corresponding export directories."

Claim 1 according to the auxiliary request reads as follows.

"A parallel processing method for unloading database tables, comprising the steps of:

determining a plural number of export directories, each export directory being located on a separate storage device;

launching (410) a number of threads limited to the number of export directories to process the database tables;

unloading the tables to create respective export files, including:

assigning (420) each of the tables to a corresponding one of said threads, using a best fit algorithm in which each thread checks all of the export directories and chooses and claims the smallest available directory that can accommodate the export file; and

unloading (440) the respective database tables by processes of the corresponding threads into the corresponding export directories."

Claim 7 according to the auxiliary request reads as follows.

"The method according to claim 1, further comprising loading data into database tables, comprising the steps of:

determining X threads for loading data into database tables;

creating X temporary tables (600), each temporary table corresponding to a set of data stored in one of said export directories;

launching (610) a process in conjunction with each thread for loading each temporary table;

loading each temporary table with the data stored in the corresponding export directory via the corresponding process."

Claims 11 and 12 according to the auxiliary request are identical to claims 11 and 12 of the main request, respectively.

Reasons for the Decision

1. The appeal is admissible (see facts and submissions, points I and IV).

Main request

2. Article 123(2) EPC

2.1 The amendments find support in original claim 1 and in the description, p. 15, lines 7-18.

In the decision under appeal, the examining division had not raised any concerns under Article 123(2) against the then second auxiliary request on which both present requests are based. With the summons to oral proceedings, the board had expressed the view that claim 1 of the then main request offended against Article 123(2) EPC because it specified the unloading of only one table and because the "choosing" of an export directory did not appear to imply that the export directory was "claimed" as well. Both objections have been remedied in amended claim 1.

Thus, the board is satisfied that amended claim 1 complies with Article 123(2) EPC.

2.2 In claim 7, the reference to an "SQL*Loader process" (cf. original claim 8) have been replaced by reference to a "structured query language loader process" and a "structured query language process", where the omission of "loader" in the latter appears to be a clerical error.

The original term "SQL*Loader" refers to an Oracle utility (cf. description, p. 10, line 30; p. 21, line 11), i.e. a specific piece of software which has been in use over many years in several different versions. The description discloses that, instead of SQL*Loader, other databases may be used, as well as other programs "capable of reading ... and loading [table] data" (p. 21, lines 10-14). The description does not disclose, however, that such other databases or programs must be based on "structured query language", i.e. SQL.

The board thus finds that the term "structure query language loader process" constitutes an intermediate generalisation of the term "SQL*Loader" which is not directly and unambiguously derivable from the description and thus claim 7 of the main request violates Article 123(2) EPC.

3. Article 84 EPC 1973

3.1 Claim 1 specifies a method of unloading database tables using a number of threads without specifying how the threads are to be executed. As claimed, they could be processed by a single processor which would distribute the available processing time over several threads so that they would advance in turns and appear to make simultaneous progress.

It is the express goal of the invention, reiterated throughout the description, to use "parallel processing techniques to increase the speed" at which database tables are handled (cf. *inter alia*, p. 1, lines 9-10; p. 7, line 15 - p. 8, line 22: esp. p. 7, line 16; p. 10, lines 5-9; p. 11, lines 3-6). It is established terminology in the art that "parallel" processing implies actual simultaneity, and the description conforms with this interpretation when it discloses that "parallel processes run simultaneously" or that "tables [are] to be unloaded/loaded at the same time" (cf. p. 13, lines 14-22). However, the use of multiple threads alone does not achieve a speed-up unless several processors actually operate in parallel.

In the board's view, therefore, the subject matter of claim 1 according to the main request does not achieve the express goal of the invention and is, therefore, not supported by the description as required by Article 84 EPC.

Auxiliary request

4. Articles 123(2) EPC

4.1 The limitation of claim 1 to a "parallel processing method" is supported throughout the original application documents (see references under point 3.2 above).

4.2 The generalisation in claim 7 to use any suitable process for loading tables rather than exclusively the Oracle utility SQL*Loader is disclosed in the original application, too (p. 21, lines 10-14).

4.3 The board is therefore satisfied that the claims according to the auxiliary request conform with Article 123(2) EPC.

5. Article 84 EPC 1973

5.1 Due to the limitation to a "parallel processing method", the board considers that claim 1 is supported by the description as required by Article 84 EPC and that the corresponding objection is overcome (cf. point 3 above).

5.2 In an *obiter dictum*, the decision (p. 10, point 1.4) considered that the term "data base table" was ambiguous between the "data of the table" and the "table structure", thereby suggesting a deficiency under Article 84 EPC 1973. In the board's view, no ambiguity arises.

The description discusses reorganization of a database after it has become fragmented over time. Database table "unloading" is a known process in this context and disclosed as such (cf. descr. p. 4, lines 25-29; p. 6, lines 31-33). Database defragmentation is meant to improve the memory space usage of the tables, but must maintain the table content so that queries to the tables before and after defragmentation produce the same results. In the board's view, therefore, the term "database table" in the context of "unloading" (as used in the preamble of claims 1 and 12) clearly refers to all the data in the table and sufficient structural information so as to enable its rebuild. Which specific information this implies in any particular case will be evident to the person skilled in the art.

5.3 The board is hence satisfied that the claims according to the auxiliary request conform with Article 84 EPC 1973.

6. Articles 54 and 56 EPC 1973

6.1 D3 is apparently the closest piece of available prior art because it discloses database reorganization and concurrent unloading of tables (see abstract, lines 7-8).

6.2 It is common ground that claims 1 and 12 of both requests differ from D3 at least in that D3 teaches concurrent unloading of table partitions as opposed to entire tables and in that D3 does not disclose a best-fit analysis to choose a suitable export directory for unloading a table. The subject matter of claims 1 and 12 is thus new over D3.

6.3 D3 does not use the term "threads" but refers to "tasks" instead (col. 8, lines 31-33 and col. 10, lines 19-21). Neither of these terms is specifically defined in the application under appeal or in D3, respectively, nor does either have, to the board's knowledge, a single, unambiguous and established meaning in the art. Both terms refer to small chunks of work that can be assigned individually to a processor (see e.g. D3, fig. 4). The board considers that it may be left open whether and to what extent "thread" and "task" differ in meaning because, at any rate, threads are a standard concept for the implementation of parallel algorithms such as that of D3. Their use would therefore not appear to be crucial for the assessment of inventive step.

7. The decision under appeal states that "D3 has a clear indication that partitions and tables have the same behaviour" (p. 8, point 2, 2nd par.). The board does not share this view but agrees with the appellant that there is a relevant difference between entire tables and table partitions.

7.1 The passage in D3 that was invoked by the examining division to support their view (col. 4, lines 9-13) contains the statement that the "behavior [of a partitioned table] is as if it were a single table" (col. 4, lines 12-13). When the context of that statement is taken into account, however, it becomes evident that it refers to the fact that different ways of storing a table (col. 4, lines 2-8) should not be visible for the user. Other differences are not excluded by this statement.

7.2 Any one partitioned table has, by construction, partitions of uniform size, whereas entire tables may vary considerably in size. While, in principle, different tables could have partitions of different sizes, D3 appears to assume this not to be the case (cf. e.g. Appendix D, D3, "calculate memory required per open DB2 partition"). Moreover, D3 assumes that the size of partitions is substantially smaller than the amount of memory available per task where it discloses calculation of a "number of partitions to be assigned to each task" (col. 10, lines 19-21).

7.3 In summary, the load balancing algorithm of D3 makes crucial assumptions about relative and absolute sizes of partitions which do not hold for entire tables.

8. D3 mentions that the pertinent database system supports different types of tables, called "simple", "segmented" and "partitioned" (col. 4, lines 2-18). D3 explains that a partitioned table is divided into "small groups", or partitions, which can be located on different storage devices. A segmented table is divided into groups (segments) that reside on the same storage device. These segments are allocated for only one segmented table and cannot have records from other tables mixed in. Simple tables are similar to segmented tables, but records from several simple tables may be mixed within segments.

8.1 D3 discloses concurrent unloading for "partitioned" tables and that the extent of partitioning affects the performance of the unloading process (col. 6, lines 16-18), but does not explain how to speed up unloading of a table that is not partitioned. In view of this, the board considers as an appropriate objective technical problem the adaptation of the method of D3 to other types of tables.

8.2 D3 does not contain any specific hint towards the solution of this problem, even where it discloses general considerations concerning optimal resource utilisation in view of the available system resources (D3, col. 9, lines 34-40).

8.3 Given the focus of D3 on concurrent unloading of *partitions*, the skilled person would, in the board's view, be bound to consider whether and how chunks of simple or segmented tables could be treated like partitions or whether, possibly, a simple or segmented table might be small enough so that it could be handled as a single partition. But considering this, the skilled person would maintain the idea of D3 that the units of parallelization should have a more or less uniform size which should be small compared to the memory available for unloading.

8.4 In this situation, a best-fit analysis would not serve any purpose. Moreover, D3 directly teaches away from a best-fit analysis when it discloses that several partitions are assigned to the same task (col. 10, lines 19-21).

8.5 In the decision under appeal, the examining division referred to the best-fit algorithm as well known (p. 9, penult. par. of sec. II) and argued, *obiter*, that the best-fit algorithm as claimed is used in the way it is normally used without any surprising effect (p. 10, sec. III, point 1.2), but also that it would not have any impact on the speed of the unloading process (sec. III, point 1.3)

The board accepts that the best-fit algorithm is well-known as a heuristic optimization algorithm. It is not apparent to the board however, whether the examining division also considered the best-fit algorithm - in general or as claimed - to be well-known in the context of database processing or for which kinds of operations. The board cannot find any of these stronger allegations self-evident, nor are they implied or hinted at by any of the available prior art documents.

The best-fit algorithm is based on the idea that a large resource should not be wasted on a small item if a small resource could be used instead, because this large resource might be needed later for a larger item. In the given context this means that a small table should not block a large export directory, if a smaller one is available, since this might delay the unloading of a large table. Accordingly, the best-fit algorithm has a clear, if only heuristic, impact on overall speed of the unloading process.

The board tends to agree that, on some level of abstraction, the best-fit algorithm according to the invention is used as it is meant to be used. This implies for example that the skilled person, when instructed to put the invention into practice, would know how to adapt the best-fit algorithm to the claimed case.

However, neither the fact that the best-fit algorithm is well-known *per se,* nor the fact that it is straightforward to adapt the algorithm to the parallel unloading of database tables, is, in the board's view, sufficient to show that it would be obvious for the skilled person to use the best-fit algorithm to modify the method of D3 as claimed.

8.6 The board therefore concludes that, starting from D3 and taking into account common general knowledge, it is not obvious for the skilled person board to perform parallel unloading on the level of entire tables, to process entire tables per thread in association with a specific export directory, and to select a suitable export directory by best-fit analysis.

8.7 Therefore, the subject matter of claims 1, 11 and 12 according to the auxiliary request shows an inventive step as required by Article 56 EPC.

Remittal

9. According to Article 111(1) EPC 1973 the board of appeal may either exercise any power within the competence of the department which was responsible for the decision appealed or remit the case to that department for further prosecution. In the present case, it appears possible to the board that for the subject matter according to the auxiliary request a search under Article 54(3) and (4) EPC remains to be carried out. Therefore, and since the appellant has agreed to this procedure, the board exercises its discretion to remit the case to the examining division for further prosecution.

Order

For these reasons it is decided that:

1. The decision is set aside.

2. The application is remitted to the examining division for further prosecution.

The Registrar: The Chairman:

B. Atienza Vivancos D. H. Rees