BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS	**BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE**	**CHAMBRES DE RECOURS DE L'OFFICE EUROPEEN DES BREVETS**

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

Datasheet for the decision of 6 December 2011

Case Number:	T 0844/07 - 3.5.01
Application Number:	01924878.0
Publication Number:	1287446
IPC:	G06F 17/30
Language of the proceedings:	EN

Title of invention:
Directory searching methods and systems

Applicant:
Computer Associates Think, Inc.

Opponent:
-

Headword:
Directory searching/COMPUTER ASSOCIATES THINK

Relevant legal provisions:
-

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

Keyword:
"Inventive step - decomposing searchable attribute values stored in a first table into components and storing them in a second table (no - routine design depending on circumstances)"

Decisions cited:
-

Catchword:
-

Case Number: T 0844/07 - 3.5.01

DECISION
of the Technical Board of Appeal 3.5.01
of 6 December 2011

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 7 November 2006 refusing European patent application No. 01924878.0 pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: S. Wibergh
Members: W. Chandler
P. Schmitz

Summary of Facts and Submissions

I. This appeal is against the decision of the examining division to refuse the European patent application No. 01924878.0 that is based on a continuation in part of WO-A-96 07 147 (D1), entitled "X.500 System and Methods". It concerns the arrangement of data and tables in an object-oriented database to enable better searching.

II. The examining division decided that the feature of "providing a choice" in the evaluation of the directory search in the independent claims of the main request violated Article 123(2) EPC. The subject-matter of the first auxiliary request, in particular the use of a second (sub-search) table for storing different components of attribute values, was considered not to be novel (Article 54(2) EPC) over D1. The second and third auxiliary requests, adding details of using the second table for searching and providing a third (sub-attribute) table, were found not to be inventive (Article 56 EPC).

III. In the statement setting out the grounds of appeal, the appellant filed a new main request based on the refused second auxiliary request, but containing the feature that was found to violate Article 123(2) EPC, reworded as "a choice is provided". A first auxiliary request was also filed without this feature. A second auxiliary request included the additional feature of the third (sub-attribute) table. A third auxiliary request contained separate independent claims to the "different components", "checksum or fingerprint" and "reverse form" of the data in the second table. The appellant also made an auxiliary request for oral proceedings.

IV. In the communication accompanying the summons to oral proceedings, the Board summarised the issues to be discussed and tended to agree with the examining division about the added-matter and that the independent claims lacked novelty and/or inventive step.

V. In a reply, dated 19 September 2011, the appellant filed a fourth auxiliary request with claims redirected to a "data processing apparatus" and with additional clarifications.

VI. At the oral proceedings, held on 18 October 2011, the appellant filed a fifth auxiliary request based on the fourth auxiliary request and designed to overcome the objection to the "alternate form". At the end of the oral proceedings, the Chairman announced that the decision would be issued in writing.

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

"A computer-implemented method of arranging and searching directory data in a directory services system modelled on X.500 or LDAP, in which the data comprises a plurality of directory objects (20) in a hierarchy, each of the plurality of objects (20) comprising plural attribute values (21-25), the method comprising:

providing a first table (SEARCH, 3) adapted for storing the directory data and having a single row for each attribute value in syntax-normalized form;

characterized by:

providing a second table (SUB-SEARCH, 4) adapted for storing different components or alternate forms of the directory attribute values in or represented by the first table, the second table having a single row (26-30) for each component or alternate form, and each row having an attribute value identifier (AID) for identifying the directory attribute value and a component identifier (CID, 7) for identifying the component or alternate form associated with the row, whereby a choice is provided, in the evaluation of a directory search, between the first table and the second table if there exists one or more appropriate components or alternate forms suitable for evaluating the search; and

executing a directory search using the second table (SUB-SEARCH, 4) and a particular component or alternate form identified by its associated identifier (CID, 7)."

Claim 7 is a corresponding claim to a "directory system".

In comparison with the independent claims of the main request, the auxiliary requests differ as follows:

First auxiliary request: Deletion of the qualification at the end of the first feature of the characterising part, starting with the words, "whereby a choice is provided…".

Second auxiliary request: Addition, as the second feature of the characterising portion, of, "providing a third table (SUBATTR, 6) containing at least one row having a plurality of columns in which the identifier

(CID, 7) and a description of the component or alternate form is provided as a tool or index for searching components of attributes". The end of the last feature is supplemented with, "using the third table (6) if necessary to determine the associated identifier".

Third auxiliary request: Deletion of the "alternate form" alternative in all places in the characterising part. Addition of two further independent claims in each category, corresponding to the existing ones, but with both alternatives replaced by "a checksum or fingerprint" and "a reverse form", respectively.

The single independent claim 1 of the fourth auxiliary request reads as follows:

"Data processing apparatus for providing directory services relating to data objects (20) in a hierarchy, each of the data objects (20) having at least one attribute, each attribute having one or more values and at least one attribute value having a plurality of attribute component values or alternate forms, the apparatus comprising:

receiving means for receiving directory search operation instructions; and

database means adapted to store data and perform directory search operations in accordance with received directory search operation instructions, wherein the database means includes a first table (SEARCH, 3) adapted to store value data for each object, said value data comprising attribute values in syntax-normalized form and, associated with each said attribute value, an attribute identifier (AID) and an entry identifier (EID) identifying the object to which said attribute value relates;

characterized in that said database means comprises:

a second table (SUB-SEARCH, 4) adapted to store component value data each defining a respective attribute component or alternate form of the attribute values in the first table, and each comprising a component identifier (CID, 7), an attribute identifier (AID) identifying the attribute value to which said component value relates and an entry identifier (EID) identifying the object to which said component value relates;

means responsive to said directory search operation instructions for locating and/or returning data objects utilizing the stored component value data in the second table (SUB-SEARCH, 4) when directory search operation instructions include a component identifier (CID, 7) and for locating and/or returning data objects utilizing the stored value data in the first table (SEARCH, 3) when directory search operation instructions do not include a component identifier."

Claim 1 of the fifth auxiliary request corresponds to that of the fourth auxiliary request with the deletion of the "alternate form" alternative in the first feature of the preamble and the first feature of the characterising part. There is an additional independent claim 6 with the alternatives replaced by "reverse form".

VIII. The appellant argued essentially as follows:

Claiming "whereby a choice is provided" did not teach new material. It was not an explicit method step performed by the system, but rather followed from the provision of the search and sub-search table recited in the claims. There was a disclosure of the two tables and explicit examples of how the tables could be used to provide a choice: "each table is provided so that there is a choice for particular search queries" - page 7, lines 21-22. How the choice was executed was a user's decision. See, for example: page 11, line 21 – page 12, line 15; page 14, lines 2-9; and page 15, lines 3-9. The user might be a designer of the directory or a user of the directory inputting search queries. What mattered was that a choice was provided and this was directly and unambiguously derivable from the application as originally filed.

The essential idea of the invention was the realisation that a more efficient search could be achieved by providing further searchable forms of attribute values in a second table, each additional form having a component identifier that could be specified in the search instruction. The additional form could be "components" of the attribute value, or some other form, such as a "reverse" form.

D1 did not disclose or suggest this. The ENTRY table could not be equated with the claimed second table because it was not searched. There were no additional searchable forms and only one table, the SEARCH table, in which searches of attribute values were performed. Dl clearly taught that all components of an object were stored in the SEARCH table. In the example given, the object included the components "surname", "commonName", "title" and "telephone number". There was no need in Dl to further decompose the attributes stored in the SEARCH table and it would not have been obvious to do this and store them in another search table. Thus the examining division had made a leap with full hindsight knowledge of the present application.

IX. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the main request or auxiliary requests 1 to 3 filed with the statement of grounds of appeal dated 19 March 2007, or auxiliary request 4 filed with letter dated 19 September 2011, or auxiliary request 5 filed during the oral proceedings before the Board.

Reasons for the Decision

1. The appeal complies with the requirements referred to in Rule 65(1) EPC 1973 and is therefore admissible.

Document D1

2. D1 explains well that a standard relational database management system (RDBMS) stores data in tables (e.g. the Employee table on page 10, Table 1.1a) where each column represents a particular data type ("name", "surname", "title", "phone") and each row represents an entry in the database (e.g. the entry for "Chris Masters"). This has a problem that if a new data type is to be added (e.g. mobile telephone number), then a new column has to be added which requires changing the

whole table - the approach is not "extensible" in the parlance of D1 (page 10, line 16).

3. In an object-oriented database (e.g. the Employee table on page 11, Table 1.2a), the data is represented as objects (e.g. the person "Chris Masters"). Each object has "attributes" (e.g. "Name", "Surname", "Title", "Phone", etc.) and it is these that are stored as rows in the table with the columns representing information about each attribute ("name", "type", "syntax", "value"), which is the same for all attributes. The advantage is that new attributes can be added without changing the structure of the table (Chris Master's mobile number is simply another row with the type "Mobile"). Since the X.500 standard requires the objects to be hierarchical, the table also contains a column called "parent name", which gives the name of the parent object (Fig 1.3b - e.g. "Chris Masters" works for "Datacraft").

4. D1 explains that such a single extensible, object oriented, hierarchical Property table can be decomposed into smaller separate "Hierarchy", "Object" and "Attribute" tables (page 14). This is a standard database technique to reduce redundant information (e.g. the repeated entries of type, syntax, object and parent names in Fig.1.3b). As its name suggests, the Hierarchy table defines the hierarchy of the objects. The Object table defines the attribute values within each object. The "entry identifier" EID and "attribute identifier" AID are keys that join (connect related information in) the tables. The Object table additionally has a "value identifier" VID to enable an attribute (e.g. "Phone"

defined by AID 20) to have more than one value (page 15).

5. D1 explains (pages 15 to 19) why various other columns are added to the different tables to improve performance. In particular, a "NameNorm" field in the Hierarchy table and a "ValueNorm" field in the Object table give normalised forms of the data in addition to the raw forms (page 19). This is said to be "in order to do comparisons (e.g. search for a particular value)". Table 2.5 (pages 19/20) shows the final structure of these tables.

6. After an explanation of how these tables can be used to implement X.500 services, such as reading attribute values of given entries (page 24) or searching for an entry with a given attribute value (page 27), D1 then describes a further decomposition of the tables to provide further performance improvements based on the needs of the X.500 services (page 32). For example, the "PARENT", "ALIAS" and normalised name "NameNorm" (now called "RDN") data from the Hierarchy table, which are all needed for navigation, are put into the "Directory Information Tree" DIT table. The Object table containing the attribute values is split into a "SEARCH" and an "ENTRY" table. The former contains the normalised attribute data (now called "NORM") and is used for searching for data. The latter contains the raw data ("RAW") and is used to return the actual results of the reads and searches. The Attribute table remains essentially the same.

The application

7. The subject-matter of the invention is common ground, e.g. as set out in the appellant's "overview" in the grounds of appeal. The invention supplements the above-mentioned object or SEARCH table ("first table" in the claims), which contains the attribute values, with an additional "SUB-SEARCH" table ("second table" in the claims) that stores the attribute values in another form. This can be a "component" or various other "alternate" forms.

8. The example given for the data in "component" form relates to an X.509 public key certificate (Figure 4: 20), which is issued by an independent authority to certify that a public key belongs to a particular user. The certificate includes data about the issuing authority 21, the validity of the certificate 22, its serial number 23, the version of X.509 24 and the name of the user whose key is being certified 25. In the SEARCH table all this data is in normalised form in one entry 3. In the SUB-SEARCH table 4, the data is split into its constituent parts 26-30, each with a "Component ID" CID. The description explains at page 12, lines 19 to 23 that the component form makes the search faster.

9. The main example given for data in an "alternate" form relates to a telephone number (Figure 5: 31). In the SEARCH table, it appears in its usual form 32, but in the SUB-SEARCH table it is stored in reverse 33, in this case with a component ID equal to zero. This enables a search for a telephone extension, i.e. a search for numbers with a specific ending, to be

translated into a search for a number *beginning* with a specific number, which is also faster (page 15, lines 3 to 9).

10. With the structure of the invention, a search can use either the SEARCH or the SUB-SEARCH tables. When a directory search instruction includes a component ID, the search is carried out in the SUB-SEARCH table, otherwise it uses the standard SEARCH table.

The requests

11. In the Board's view, the examining division was correct to object to the feature relating to providing a choice as an extension of subject-matter. Moreover, in general agreement with the comments expressed in the Obiter Dicta in section III of the decision, the Board has doubts that the term "alternate form" is entirely clear. Only the fifth auxiliary request does not have either of these problems, which is the reason that it was filed and admitted by the Board. The Board prefers to consider this request first.

12. Claim 1 of this request refers only to the alternative of storing different components. It is common ground that the subject-matter of this claim differs from D1 by the second table that stores components of the attribute values in the first table, each component having a component identifier. A search operation specifying this identifier uses the data in the second table.

13. It is also common ground that this has the effect of improving the efficiency of the search as stated at

page 9, lines 10 to 15 of the application. The examining division formulated the problem as how to improve the data retrieval from an LDAP or X.500 directory represented in a relational database system with complex attributes. According to the minutes of the oral proceedings before the examining division at page 14, the applicant did not consider such a problem to be obvious. However, the Board considers that even if the skilled person did not already know (see below), he would realise in the course of using such a database that searching in complex attribute values is slow and would look for a way to speed this up.

14. The difference in opinion in this case, both during examining and in appeal, essentially boils down to whether the invention is the inventive provision of alternate attribute forms for improved searching, or, as the examining division stated in the decision (page 9, penultimate paragraph), merely a further decomposition along standard lines of complex attributes into an additional table.

15. Concerning the problem of the efficiency of search, the Board considers that the skilled person would know that this is high for exact match searches or "beginning with" searches, a technical reason for the latter being given in D1 at page 38, lines 13 to 20. In D1, the values of the attributes of all the objects are contained in the single object or SEARCH table containing one row for each attribute value. Thus, if an attribute value becomes complex, itself having attributes, such as the example of the certificate given in the description, a search for anything other than the beginning would be inefficient. In the Board's

view, the skilled person would therefore consider storing these sub-attributes or "components" in separate fields. This also follows from the general idea, prevailing in D1 (see for example, page 12, "1.3 Hierarchical"), of organising objects into a hierarchy. Thus, as far as the structure is concerned, the Board agrees with the examining division that the invention is essentially a re-application at a finer level of granularity of the hierarchical relationship between the values in the Hierarchy and Object tables.

16. The Board further agrees with the examining division at page 13, main paragraph and page 14, main paragraph, first two sentences, that it would follow from the general principles of D1 to store these components in a "vertical schema" in which the components would be stored as new attributes in additional rows of a table rather than in additional columns as in a "horizontal schema", which would require modifying the query application each time a column is added to the table. This is essentially how an attribute with more than one different value is handled using the above-mentioned value identifier VID, albeit for semantic rather than search reasons.

17. In the Board's view, the use of a single additional table (second table) for such alternatives or components essentially boils down to a design choice using a standard technique depending on the circumstances. The standard technique is decomposition of complex data into an additional table, as the examining division stated at the last paragraph of page 9 of the decision. The examining division gave at page 14, end of main paragraph, some examples of the

circumstances that would lead to the use of a single additional table (second table) for all these components. Firstly, a further table is needed to avoid "breaking the semantics" of the existing (first) search table and thus avoid changing the existing search system, i.e. the same reason as for not using additional columns. Secondly, too many tables increase the complexity of the search queries. The Board also agrees with the examining division that the provision of identifiers to identify the components is a matter of normal design procedure in this field. The above-mentioned identifier VID is such an identifier for different values of the same attribute. Finally, the Board notes that the use of the second table does not involve any surprising effects.

18. The appellant argued that in D1 there was only ever one "search" table, i.e. the table that the search filter is applied to in order to find the required data. In particular, the DIT table could not be considered as the first table. There was no disclosure or suggestion of searching in a second table. Although the Board accepts that the search for attribute values in D1 is only in the SEARCH table, it in no way teaches away from the use of a second table because the storing of components was not contemplated. As soon as it is, the Board agrees with the examining division that a second table is an obvious possibility for the reasons given above. Moreover, in the Board's view, the search in the DIT table for a normalised name (RDN) in the course of the "tree walk" does teach how such a search would be performed, namely a "SELECT" statement with an appropriate identifier in the hierarchy and an exact match filter (cf. D1, page 24 and application, page 12).

19. Since claim 1 of the fifth auxiliary request is not inventive, claim 6, which relates to storing a "reverse" form of the attribute values in the second table, does not need to be considered. However, this claim only differs by the nature of the alternate form, the key idea of using the second table and a component ID, not judged to be inventive by the Board, remaining the same.

20. Moreover, since claim 1 of the first, third and fourth auxiliary requests is essentially the same or broader, these requests are not allowable either.

21. Claim 1 of the main request is at first sight more restricted than that of the above mentioned requests, additionally including the feature relating to providing a choice. However, even if this is not an extension of subject-matter, but, as maintained by the appellant, merely follows from having two search tables available, then it does not add anything new and cannot contribute to inventive step.

22. Claim 1 of the second auxiliary request adds to claim 1 of the main request the idea of the third (sub-attribute) table. However, the Board agrees with the examining division that this is simply the further application of a routine design measure that does not involve an inventive step; the attributes of the components would need to be identified and stored in the same way as the attributes of the objects themselves, the use of an additional table being an obvious possibility.

23. Since the subject-matter of claim 1 of all requests does not involve an inventive step (Article 56 EPC 1973), it follows that the appeal must be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
C. Eickhoff	S. Wibergh