

Internal distribution code:

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

**Datasheet for the decision
of 28 October 2010**

Case Number: T 1266/08 - 3.5.01

Application Number: 01300672.1

Publication Number: 1130511

IPC: G06F 9/46

Language of the proceedings: EN

Title of invention:

Data transfer and synchronization system

Applicant:

fusionOne, Inc.

Opponent:

-

Headword:

Synchronising databases II/FUSION ONE INC

Relevant legal provisions:

-

Relevant legal provisions (EPC 1973):

EPC Art. 56

Keyword:

"Inventive step - use of universal format to enable synchroniser to work with data from different applications (No - routine design)"

"Inventive step - use of server data store to avoid need for devices to be synchronised to be connected to the system at the same time (No)"

"Inventive step - use of mapping module to enable synchronisation of the same data in different directories (No)"

Decisions cited:

-

Catchword:

-



Case Number: T 1266/08 - 3.5.01

D E C I S I O N
of the Technical Board of Appeal 3.5.01
of 28 October 2010

Appellant: fusionOne, Inc.
55 Almaden Boulevard - Suite 800
San Jose
California 95113 (US)

Representative: Harris, Ian Richard
D Young & Co LLP
120 Holborn
London EC1N 2DY (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 5 February 2008
refusing European patent application
No. 01300672.1 pursuant to Article 97(2) EPC.

Composition of the Board:

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

Summary of Facts and Submissions

- I. This appeal is part of a series of appeals (also including T 1203/08, T 1262/08 and T 1263/08) from related applications that tackle the problems of synchronising information for a user having a PC and various portable devices, such as a laptop computer and a personal digital assistant (PDA), or mobile phone.
- II. The present appeal is against the decision of the examining division to refuse the European patent application No. 01300672.1 according to the state of the file. In the communication forming the basis of the decision, the division considered that the feature of the "data interface" used in claim 1 was vague and unclear (Article 84 EPC 1973). Notwithstanding this objection, it was considered that the synchronising system of claim 1 lacked an inventive step (Article 56 EPC 1973) over US-A-5 926 816 (D5), which the division introduced into the proceedings in the above-mentioned communication, and the skilled person's common general knowledge.
- III. In the statement setting out the grounds of appeal, the appellant filed an amended claim 1 with new features relating to a "field mapping module".
- IV. In the summons, dated 28 June 2010, the Board scheduled oral proceedings for all four related appeals on the 27 and 28 October 2010 with a reserve day on the subsequent day. In the communication, the Board summarised the issues to be discussed and was of the preliminary view that the new features did not add anything inventive.

- V. In a letter, dated 23 September 2010, the representative informed the Board of the appointment of a new representative.
- VI. In a letter, dated the same day, the new representative informed the Board that he was in the process of obtaining instructions from the appellant's US counsel and requested time to prepare written submissions in preparation for the oral proceedings. In a further letter, dated 27 September 2010, the representative requested a postponement of the oral proceedings in order to allow sufficient time to prepare and have written submissions approved.
- VII. The Board did not allow the postponement because the reason was not considered to be a serious substantive reason in the sense of the "Notice of the Vice-President of Directorate-General 3 of the European Patent Office dated 16 July 2007 concerning oral proceedings before the boards of appeal of the EPO" (SE No. 3 OJ EPO 2007, 115) that might justify a change of date. Moreover, the Board pointed out that the summons had already been issued on 28 June 2010 which seemed to have given the appellant enough time for preparation.
- VIII. In a reply, dated 1 October 2010, the appellant filed an auxiliary request with a substantially amended claim 1.
- IX. At the oral proceedings, the appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the further amended auxiliary request as submitted during the oral

proceedings before the Board, this request being the sole request. All other requests were withdrawn. At the end of the oral proceedings, the Chairman announced the decision.

X. Claim 1 of the sole request reads as follows:

"A system comprising a network for synchronizing data between a first system (802-808) and a second system (802-808), comprising:

 a first sync engine (862-868) on the first system interfacing with data on the first system to provide difference information in a first difference transaction, the first sync engine including

 a first application interface (910) for interfacing with data on the first system having a first application data format and for converting between the first application data format and a universal format that has objects and field types,

 a first application object store (920) for storing a copy of a previous state of the data on the first system,

 a first field mapping module (935) for user-remapping of data fields of the data on the first system, and

 a first differencing engine (950) for calculating differences in data between the output of the first application interface and the copy of a previous state of the data in the first application object store for generating the difference transaction for output having first difference information in the universal format including the objects and field types;

 a server data store (850) coupled to the network and in communication with the first and second systems,

the server data store being operable to receive via the network from the first synch engine and to store the first difference information in the first difference transaction in the universal format including the objects and field types; and

a second sync engine (832-838) on the second system coupled to receive from the server data store via the network the first difference information in the first difference transaction in the universal format including the objects and field types and to interface with data in a second application data format on the second system to update the data on the second system, wherein the second synch system includes

a second application interface (910) for interfacing with data in the second application data format on the second system and for converting between the second application data format and the universal format that has objects and field types,

a second application object store (920) for storing a copy of a previous state of the data on the second system, and

a second differencing engine (910) operable to receive the difference transaction in the universal format including the objects and field types from the server data store via the network and to return differenced data in the universal format to the second application interface for conversion to the second application data format."

XI. The appellant argued essentially as follows:

Claim 1 defined a device that synchronised data on two clients via a server data store.

The field mapping module enabled the user to select mappings between fields in the applications on one client with fields in the applications on the other client.

The invention differed fundamentally from D5 in that an application interface at the client converted the data into a universal format. The previous values of the data were also stored in this format in the application object store and the difference was generated and output in this format. There was no translation of the data at the data server, it merely passed the data to the second client.

In D5, the comparison was made in the format in which it was stored, i.e. that of the tables shown in the figures. D5 stated at column 3, lines 31 to 34 that the current and previous versions were "identical copies" of the tables. There was no suggestion of a comparison in a different format, and this would have gone against the teaching of the document.

The invention enabled a modular and more flexible system for synchronising.

If a new device were to have been added to the system of D5, the skilled person would not have considered a universal format, but would have provided a new difference engine operating on the data in the format of the new device.

Reasons for the Decision

1. The appeal complies with the requirements referred to in Rule 99(2) EPC and is therefore admissible.

The application

2. The basic idea of synchronisation as shown in Figure 8 of the published application is that if the user changes application data (822-828) associated with various applications (812-818) in one of his devices (802-808), this change must be propagated to the corresponding application data in the other devices.
3. An important aspect of the invention as defined in the original and present claims is that when application data is to be synchronised, only the items that have changed are transmitted instead of the entire data. This is achieved using a sync engine, such as the generic one shown in Figure 9A in a first system that extracts the application data into an "application object" (AO) 910 and uses a delta module 950 to calculate the difference between the current value of the AO and the value at the time of the last synchronisation as stored in the AO store (AOS) 920 (paragraphs 55 to 56). The differences are output via the storage server 850 to a similar sync engine in the second system, which receives the difference data and converts it into the second system's application data format (paragraphs 66 and 67). Transmitting only data that has changed has the effect of reducing the required bandwidth and thus increasing the speed of the synchronisation (paragraph 38).

4. D5 discloses a general purpose system for synchronising databases. This operates essentially in the same way as the invention, namely by transmitting only modifications detected in the client data since the last synchronisation by comparing the data with a "before-image" (column 2, lines 12 to 22). Thus D5 discloses the main aspect of the invention, which is what led the examining division to refuse the application under Article 56 EPC 1973, correctly, in the Board's view.

Requests in appeal

5. In appeal, the applicant had previously only relied on the newly added feature of the mapping module, citing paragraph 63 of the description for support. In reply to the Board's objections about this support at the oral proceedings, the new representative clarified and restricted the feature to specify that the *user* made the mapping.
6. The representative explained that the other amendments to claim 1 were designed to define the essential differences over D5. In particular, to emphasise the fact that in the invention each system has a specific "application interface" 910 that immediately maps the system's application data into a generic or "universal" format that has "objects" and "field types" (paragraph 54). The data in the AOS and the differences are also stored in this format (paragraph 57). The server data store 850 merely acts as a temporary storage for the difference data, but does not process it (paragraphs 49 and 66).

7. Although the present claims were filed late in the proceedings, the Board admits them because they are clearly a serious attempt to meet the outstanding objections without introducing any further objections and there was a good reason for the lateness, namely that the representative only took over the case at short notice and there was no postponement of oral proceedings (see above).

Inventive step

8. It is common ground that the system of claim 1 differs from that of D5 essentially by the above-mentioned immediate conversion and subsequent processing in the "universal" format that has objects and field types, in that there is a field mapping module for user-remapping of data fields of the data on the first system and in that there is intermediate storage in the data store.
9. The application explains at paragraph 54 that the use of the universal format has the effect of enabling the synchroniser to work with data from different applications using the same basic structure for the difference engine, store and difference packaging, only requiring a specific application interface for each. The representative suggested that this solved the problem of providing a more flexible system. Paragraph 41 gives the effect that arises from the intermediate storage, namely that devices to be synchronised need not be connected to the system at the same time. This could also be seen to fall under the general problem of providing a more flexible system, albeit in the timing of the synchronisation process. In the Board's view, there is no synergy between this effect and that

arising from the choice of the format of the data. Finally, paragraph 63 explains that the mapping feature allows the user to synchronise data between devices where the data is in directories with different names (e.g. "My Documents" on a PC and another directory on the notebook computer). However, it was not argued that the effect of this feature had any synergy with the effects of the other features in the claim. Thus, in the Board's view, the three distinguishing features can be discussed separately.

10. Concerning the problem of providing a more flexible system, in addition to the above-mentioned features, D5 acknowledges at column 9, lines 52 to 57 the problem of working with heterogeneous database products and states that this "requires a general purpose technique". D5 explains that the use of a "before-image" is such a general purpose technique because not all database products have logging capabilities, i.e. enable detection of differences. The present invention appears also to have been shaped by this consideration since the description contemplates at paragraph 59 cases, e.g. for PDAs, where the AOS is not required.

11. D5 also mentions at the end at column 27, lines 47 to 54 that its teaching is also applicable to object-oriented representations of data where a class has the properties equivalent to that of a table in a relational database. Although the appellant played down this aspect of the disclosure as not providing any direct suggestion to use a universal format, the Board considers that the skilled person would be led to this using common general knowledge and routine design skills.

12. This is because when the skilled person reads D5 as a whole, he would realise, if he did not know already, that the disclosed "before-image" technique of synchronising data between different data base systems could be implemented in an object-oriented language. Since the emphasis in object-oriented programming is on re-usability and modularity, he would have this at the back of his mind as well.

13. Thus in the Board's view the problem of providing a flexible synchroniser based on D5 would immediately translate into the practical problem of how to implement the technique of D5 in an object-oriented language.

14. In the Board's view, the examining division was correct to assert that it is conventional programming practice in complex systems that interact with proprietary products such as databases to write a common program that uses the product manufacturers' interfaces (APIs). These are the "traditional" APIs actually mentioned in the application at paragraph 90. Indeed without such interfaces it would scarcely be possible to access the applications' data at all without having recourse to each program's source code, which of course manufacturers are reluctant to release and would involve a lot of work. In short, the API's job is to transfer data to and from the database.

15. Thus in any practical implementation, the skilled person would have to access the data from the different applications using the respective APIs. It is the programmer's job to integrate the data provided by the

APIs into the data structures of his own program; he is not bound to any particular data structure and indeed must design some structure to be able to use the data at all. For a single program designed to be used with different database products, it is self evident that there can only be one such structure and that this would be different from at least some of the product's own internal structures. Thus, in the Board's view as soon as the data has been extracted via the API and stored in the data structure of the synchronising program it is already in a "universal" format in the sense that this format must accommodate, or be extensible to accommodate, all of the data items that may be present in the systems to be synchronised.

16. In the Board's view, the remainder of the features of the first and second sync engines follow immediately from this inevitable use of a common data structure in the synchroniser program. The methods that use the APIs to extract the data would form an "application interface". The data structure used to store the "before-image" would be an "application object store" storing data in the "universal" data format having objects and field types, i.e. the format defined by the program object's representation of the data. The difference engine would calculate differences between the output of the application interface and the copy in the application object store. These differences would still be in the "universal" format and would most obviously be sent to and received from the server in this format.
17. The representative argued that in D5 the difference operation was taken at the level of the structure of

the tables and so there was no need or suggestion to convert to a universal format first. However, in the Board's view, a dogged attempt to stick with the concept of tables does not take full account of the disclosure of D5 or the skilled person's common general knowledge as explained above. In particular, once the decision has been taken to represent the data (i.e. the tables) as objects, the argument must be translated to mean that the difference operation must be taken at the level of the objects. However, as pointed out above it is not practical to consider operating directly on the objects of each database application because these are not generally available to the programmer of a third party system; the APIs are used to access the data.

18. The representative also argued that even if the skilled person were to use APIs to access the applications' data, he would still provide a separate difference engine for each application appropriate to its structure and store the before-image data in the application itself in its own structure. The Board agrees that this is at least a technically meaningful alternative, but its existence does not imply that the other involves an inventive step. The choice of storing the before-image in the application program or in the synchroniser program is a design decision that would depend on the circumstances, such as memory availability. Moreover, in the Board's view, the principle of modularity would lead the skilled person to provide as many common modules as possible so that a single difference engine and store is the more realistic solution.

19. Concerning the mapping feature, the Board considers that it would be a common everyday situation that the files to be synchronised might be in directories with different names, so that the skilled person would have recognised the problem. Moreover, the application itself confirms this when it says in paragraph 62, preceding the explanation of the mapping module, that the module "provides the functionality both necessary for all synchronization programs, and *which users have come to expect*" (Board's emphasis). The implementation by means of a "field mapping module" would be a self-evident aspect of implementation.

20. Concerning the problem relating to the timing of the synchronisation, the Board considers that it is common knowledge to use some sort of server to provide an intermediate storage so that devices need not be continuously connected to the system. This concept is also implicit in the system of D5, which describes at column 1, lines 52 and 54, for example, that "updates performed by either client or server are propagated to the other side when a connection is established and eventually to other clients in the system", i.e. when they are connected to the system.

21. The appellant argued that the invention differed in that the server was "dumb" and merely passed data to the second client, whereas in D5 it translated the data. However, in the Board's view, the claim does not necessarily exclude an additional translation by the server, but merely recites a server that stores the data that is later received by the second system. Moreover, the possibility of passing the data without processing is a direct consequence of using a

"universal format", so that the skilled person would have only considered translating the data at the server if the server computer needed to use the data, e.g. if it itself had applications that needed to be synchronised. The server in D5 does because it is the central source of data for the system and thus needs to capture the data coming from the portable device. The Board cannot see an inventive step in the decision to remove this requirement leaving the server with a mere data transferring role, or equivalently to site the server in any given device.

22. Accordingly, the Board judges that the subject-matter of claim 1 does not involve an inventive step (Article 56 EPC 1973), so that the appeal must be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:

T. Buschek

S. Wibergh