BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS | BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE | CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN 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 23 May 2016

Case Number:	T 2413/10 - 3.5.07
Application Number:	02716869.9
Publication Number:	1390874
IPC:	G06F17/30
Language of the proceedings:	EN

Title of invention:
Synchronization of database data

Applicant:
Nokia Technologies Oy

Headword:
Database synchronization/NOKIA TECHNOLOGIES

Relevant legal provisions:
EPC Art. 56, 84, 123(2)

Keyword:
Amendments - added subject-matter (no)
Claims - clarity after amendment (yes)
Inventive step - (yes)

Decisions cited:

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Catchword:

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

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 2413/10 - 3.5.07

DECISION
# of Technical Board of Appeal 3.5.07
of 23 May 2016

Appellant: (Applicant)	Nokia Technologies Oy Karaportti 3 02610 Espoo (FI)
Representative:	Kolster Oy Ab Iso Roobertinkatu 23 PO Box 148 00121 Helsinki (FI)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 10 June 2010 refusing European patent application No. 02716869.9 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	M. Rognoni
Members:	R. de Man
	F. Blumer

Summary of Facts and Submissions

I. The applicant (appellant) appealed against the decision of the Examining Division refusing European patent application No. 02716869.9.

II. The Examining Division decided that claim 1 as then on file infringed both Article 123(2) EPC and Article 84 EPC. In addition, it considered the subject-matter of claim 1 to lack inventive step over the following document:

D1: "SyncML Sync Protocol, version 1.0",
7 December 2000, retrieved from the Internet at
URL:http://www.syncml.org/docs/
syncml_protocol_v10_20001207.pdf.

III. With the statement of grounds of appeal, the appellant filed a marked-up copy of a set of claims 1 to 13. The appellant indicated that it had amended its claims to overcome some of the grounds for the refusal but also that it preferred to maintain the claims unamended and that it proposed, as an auxiliary request, to amend claim 1 to include a definition marked in the amended claims in parentheses.

IV. With a communication dated 20 October 2015, the Board invited the appellant to clarify its requests. It further indicated that the subject-matter of claim 1 of the marked-up claims appeared to lack inventive step and to include an unclear feature.

V. With a letter dated 17 December 2015, the appellant submitted a main request and auxiliary requests 1 and 2, the main request and auxiliary request 1 corresponding to the marked-up claims of the request filed with the statement setting out the grounds of appeal, respectively without and with the parenthesised features.

VI. The appellant was summoned to oral proceedings. In the course of the oral proceedings held on 23 May 2016, the appellant replaced its substantive requests with a sole main request comprising claims 1 to 10. 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 sole main request.

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

"A method of arranging synchronization of databases, the method comprising the steps of:

establishing a logical data transmission connection for synchronization according to Synchronized Markup Language synchronisation protocol between a first and a second device which synchronize databases,

transmitting a Last anchor, which is stored at least in the first device and describes the latest synchronization event the first and the second device have performed in the databases, during initialization of synchronization from the first device to the second device, and a Next anchor, which is defined by the first device and describes the present synchronization,

storing temporarily said Next anchor in the first device and in the second device;

comparing the Last anchor received in the second device with the Last anchor stored in the second device, performing synchronization in the requested manner if the received Last anchor corresponds to the stored Last anchors,

synchronizing all data units of the databases if the received Last anchor does not correspond to the stored Last anchor;

and

updating the contents of the Last anchors stored in the devices with the contents of said temporarily stored Next anchor in the first device and in the second device if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for, and

not updating the contents of the Last anchors stored in the devices with the contents of said temporarily stored Next anchor in the first device and in the second device if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

Claims 2 to 5 are dependent on claim 1.

Claim 6 reads as follows:

"A synchronization system for synchronizing databases comprising a first device and a second device which perform synchronization and are arranged to

establish a logical data transmission connection between the first device and the second device for performing synchronization according to Synchronized Markup Language synchronisation protocol,

transmit a Last anchor, which is stored at least in the first device and describes the latest synchronization event the first and the second device have performed in the databases, from the first device to the second device during initialization of synchronization and a Next anchor, which is defined by the first device and describes the present synchronization,

store temporarily said Next anchor in the first device and in the second device;

the second device is arranged to compare the Last anchor received with the Last anchor stored in the second device,

the first and the second device are arranged to perform synchronization in the requested manner if the received Last anchor corresponds to the stored Last anchor,

the first and the second device are arranged to synchronize all data units of the databases if the received Last anchor does not correspond to the stored Last anchor;

and

update the contents of the Last anchors stored in the devices with the contents of said temporarily stored Next anchor if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for, and

not to update the contents of the Last anchors stored in the devices with the contents of said temporarily stored Next anchor

if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

Claim 7 reads as follows:

"A telecommunications device comprising

means for establishing a logical data transmission connection to at least one other device for synchronizing databases according to Synchronized Markup Language synchronisation protocol,

means for transmitting a Last anchor and a Next anchor defined by said telecommunications device to the other device during initialization of synchronization, the Last anchor being stored at least in said telecommunications device and describing the latest synchronization event said telecommunications device and the other device have performed in the databases, and said Next anchor describing the present synchronization,

means for storing said Next anchor temporarily; and

means for updating the contents of the Last anchor stored in said telecommunications device with the contents of said temporarily stored Next anchor if synchronization has been performed and after this said logical data transmission connection has been ended properly, and

means for not updating the contents of the Last anchor stored in said telecommunications device with the contents of said temporarily stored Next anchor if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

Claim 8 reads as follows:

"A synchronization device comprising

means for establishing a logical data transmission connection to at least one other device for synchronizing databases according to Synchronized Markup Language synchronisation protocol,

means for receiving a Last anchor and a Next anchor from the other device during initialization of synchronization, the Last anchor describing the latest synchronization event said synchronization device and the other device have performed in the databases, and the Next anchor describing the present synchronization,

means for storing said Next anchor temporarily;

means for comparing the Last anchor received with the Last anchor stored in said synchronization device,

means for performing synchronization in the manner requested by the other device if the received Last anchor corresponds to the stored Last anchor, and

means for synchronizing data units of all databases if the received Last anchor does not correspond to the stored Last anchor;

and

means for updating the contents of the Last anchor with the contents of said temporarily stored Next anchor if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for;

and

means for not updating the contents of the Last anchor with the contents of said temporarily stored Next anchor if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

Claim 9 reads as follows:

"A computer program product loadable into the memory of a telecommunications device and comprising a code which is executable in the telecommunications device making the telecommunications device

establish a logical data transmission connection to at least one other device for synchronizing databases according to Synchronized Markup Language synchronisation protocol,

transmit a Last anchor and a Next anchor defined by said telecommunications device to the other device during initialization of synchronization, the Last anchor being stored at least in said telecommunications device and describing the latest synchronization event said telecommunications device and the other device have performed in the databases, and said Next anchor describing the present synchronization,

store said Next anchor temporarily

and

update the contents of the Last anchor stored in said telecommunications device with the contents of said temporarily stored Next anchor if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for,

not to update the contents of the Last anchor stored in said telecommunications device with the contents of said temporarily stored Next anchor if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

Claim 10 reads as follows:

"A computer program product loadable into the memory of a computer functioning as a synchronization device and comprising a code which is executable in the synchronization device making the synchronization device

establish a logical data transmission connection to at least one other device for synchronizing databases according to Synchronized Markup Language synchronisation protocol,

receive a Last anchor and a Next anchor from the other device during initialization of synchronization, the Last anchor describing the latest synchronization event said synchronization device and the other device have performed in the databases, and the Next anchor describing the present synchronization,

store the Next anchor temporarily;

compare the Last anchor received in the second device with the Last anchor stored in the second device,

perform synchronization in the requested manner if the received Last anchor corresponds to the stored Last anchor,

synchronize all data units of the databases if the received Last anchor does not correspond to the stored Last anchor;

and

update the contents of the Last anchor stored in said synchronization device with the contents of said temporarily stored Next anchor if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for, not to update the contents of the Last anchor stored in said synchronization device with the contents of said temporarily stored Next if the synchronization has not been performed and/or the logical data transmission connection has not been ended properly in a manner after which no transport layer messages are waited for."

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 invention relates to synchronisation of databases storing application data on two devices, for example a portable client terminal and a network server, based on the SyncML ("Synchronized Markup Language") synchronisation protocol.

2.2 Paragraph [0005] of the background section of the published application explains that, in a synchronisation session in accordance with the SyncML protocol, a SyncML client terminal transmits to a SyncML server a message that includes the data that has changed since the last synchronisation. The server analyses the data, makes necessary changes, and transmits a message containing server modifications to the client.

2.3 As explained in paragraphs [0006] and [0007], the SyncML protocol includes a mechanism for verifying, during initialisation of a synchronisation session, whether the previous synchronisation session completed successfully. If it did not complete successfully, the synchronisation process must check all the data in the two databases and not just those fields that have changed since the last session. This is referred to as "slow synchronization".

2.4 This verification mechanism employs "synchronization anchors", which are strings describing synchronisation events in terms of, for example, the date and time. The "Last" anchor represents the last successful synchronisation event and the "Next" anchor represents the present synchronisation event. During initialisation, the client device transmits both its "Last" anchor and its "Next" anchor to the other device. If the client determines that, from its point of view, the present synchronisation completed successfully, it updates its "Last" anchor to the value of the "Next" anchor. Similarly, if the server determines that, from its point of view, synchronisation completed successfully, it stores the received "Next" anchor as the new "Last" anchor. During a subsequent synchronisation, the server verifies whether the previous synchronisation completed successfully from both the client's and the server's point of view by determining whether the stored "Last" anchor matches the received "Last" anchor.

2.5 Paragraph [0008] of the background section explains that the SyncML standard (more precisely version 1.0 of the standard as current at the priority date of the application; see paragraph [0006]) does not accurately define the moment at which the "Last" anchor is to be updated by client and server devices. A device implemented in accordance with the standard may update the "Last" anchor once all SyncML messages required by the SyncML protocol have been exchanged, but it may also first require the underlying network transport connection to have terminated without errors.

In the example depicted in Figure 2, the server updates its "Last" anchor once all SyncML messages have been interchanged, whereas the client waits until the underlying network transport connection terminates without errors. Since in this scenario the network transport connection does not terminate without errors, the client does not update its "Last" anchor. The "Last" anchor received from the client therefore does not match the server's "Last" anchor at the start of the second synchronisation session, resulting in "slow synchronization". Had the server not "unnecessarily" updated its "Last" anchor, but had it, like the client, waited for error-free termination of the network transport connection, neither of the "Last" anchors would have been updated and the second synchronisation session would not have resulted in "slow synchronization" (see the scenario depicted in Figure 6).

2.6 The invention therefore proposes updating, in both the client and the server, the "Last" anchor "if synchronization has been performed and after this the transport layer connection established for synchronization has been ended substantially properly", i.e. once all SyncML messages required by the SyncML protocol have been exchanged and the underlying network transport connection has been terminated without errors. See paragraph [0010].

3. *Clarity and added subject-matter*

Independent claim 1 of the sole main request corresponds essentially to a combination of claims 1 and 2 as originally filed with the following clarifications based on the original description:

- the two devices perform synchronisation in accordance with the SyncML protocol (paragraphs [0015] and [0021]);
- the update identifiers exchanged are "Last" and "Next" anchors (paragraphs [0018] and [0019]);
- the condition for the transport or logical data transmission connection to have "ended substantially properly" is specified as "in a manner after which no transport layer messages are waited for" (paragraph [0023]);
- if the condition for updating the "Last" anchor has not been fulfilled, the "Next" anchor is not updated (paragraph [0029]).

Dependent claims 2 to 5 are based on originally filed dependent claims 3 to 6. Independent claims 6 to 10 correspond to independent claim 1.

The Board is therefore satisfied that the claims comply with Articles 84 and 123(2) EPC.

4. *Inventive step*

4.1 Document D1, describing version 1.0 of the SyncML synchronisation protocol, is referred to in paragraph [0006] of the background section of the application and evidently served as the starting point for the invention. The Board agrees with the Examining Division that it represents the closest prior art, and the appellant has not contested this.

4.2 In sections 1.1, 1.2 and 2.2.1, document D1 discloses the "SyncML Framework" for synchronising databases storing application data of a client device and a server device. The framework employs the SyncML synchronisation protocol on top of a network transport connection such as HTTP, WSP or OBEX (see Figure 1).

4.3 A number of synchronisation types are listed in section 1.3 and discussed in sections 5, 5.5, 6, 6.3, 7, 7.5 and 8. During initialisation of a synchronisation session, the client specifies the desired synchronisation type (see section 4.1, first paragraph). Section 5.5 discusses the "slow sync" synchronisation type, which is a form of two-way synchronisation in which all items in one or more databases are compared with each other on a field-by-field basis. Slow sync is initiated when two devices synchronise with each other for the first time or when either the client or the server indicates a need for it.

4.4 Section 2.2.1 explains that the protocol employs "Last" and "Next" synchronisation anchors, which are sent during initialisation of a synchronisation session. The "Last" anchor describes the last synchronisation event, and the "Next" anchor describes the current synchronisation event from the point of view of the sending device.

4.5 In the example discussed in Section 2.2.1.1 and shown in Figure 3, the sending device is the SyncML client, which transmits a "Last" anchor and a "Next" anchor to the SyncML server. At the start of the first synchronisation session, the server stores a "client sync event" anchor, which matches the value of the "Last" anchor received from the client. Because they match, slow sync is not initiated. After completion of the first session, the server updates its "client sync event" anchor to the contents of the "Next" anchor received from the client.

In this example, the persistent storage of the client is reset between the end of the first synchronisation session and the beginning of the second synchronisation session. The "Last" anchor transmitted by the client to the server at the start of the second synchronisation session (now set to "Empty") consequently does not match the "client sync event anchor" stored in the server (now set to the contents of the "Next" anchor received during the first synchronisation session). As a result, the server initiates slow sync.

4.6 The "client sync event anchor" stored on the server is hence the "Last anchor stored in the second device" of claim 1. Once the synchronisation session is finished (i.e. "synchronization has been performed"), the client and server update their respective "Last" anchors to the contents of the "Next" anchor transmitted from the client to the server during initialisation of the session. That "Next" anchor is hence stored for the duration of the session (and thus "temporarily") at both devices.

Since the "Last" anchor serves to identify the last (successful) synchronisation, the skilled person understands that it is not updated to the contents of the "Next" anchor if the synchronisation session is not completed successfully.

4.7 Document D1 does not specify in detail how the client and server are to determine whether the synchronisation session has been successfully completed. Since the client can only "see" its side of the synchronisation session, the determination of "successful completion" must be made on the basis of the messages it has sent and received (in addition to any internal error conditions), and the same applies to the server. The reader of document D1 understands that, at the least, a synchronisation session is not considered to have been completed until a device has received all SyncML messages as defined by the SyncML protocol.

4.8 Claim 1 defines for both devices participating in a synchronisation session the condition for updating the "Last" anchors for both devices participating in a synchronisation session as "if synchronization has been performed and after this said logical data transmission connection has been ended properly in a manner after which no transport layer messages are waited for". This feature is the only distinguishing feature and indicates more precisely that, from the point of view of a device participating in a synchronisation session, the session is considered to have been successfully completed when not only all expected SyncML messages have been received, but also all transport layer messages.

4.9 The Board notes that, from a technical point of view, a device participating in a SyncML synchronisation session could instead consider the session to have been successfully completed once it has sent the last SyncML message it is supposed to send and has received the last SyncML message it is supposed to receive even if the underlying transport layer connection has not yet been terminated (and possibly will only terminate with a timeout error).

This approach is in fact advantageous if the network connection happens to break down after all SyncML messages have been sent and received but before the last transport layer message has been received: both participating devices will update their "Last" anchors and the next synchronisation session will not trigger a slow sync. On the other hand, if the network connection happens to break down after the last SyncML message has been sent but before it has been received, one device will update its "Last" anchor and the other will not, resulting in an unnecessary slow sync during the next session. In this case, waiting for the transport layer connection to close before updating the "Last" anchor as proposed by claim 1 would reveal to the device sending the last SyncML message that the other device probably did not receive it, and neither device would update its "Last" anchor.

4.10 In other words, multiple implementations for detecting "successful completion" of the synchronisation session are possible, and the claimed one is not necessarily better or worse than others in the sense of being inherently more likely to prevent an unnecessary slow sync.

In the Board's view, the objective technical problem solved by the invention over document D1 is therefore merely that of providing an implementation for detecting "successful completion" of the synchronisation session.

4.11 Faced with this problem, the skilled person would consider any suitable implementation suggested by the prior art or his common general knowledge, and any such implementation would in principle have to be regarded as obvious; in the absence of a specific technical advantage, a particular reason for selecting the claimed implementation would not be necessary (see Case

Law of the Boards of Appeal, 7th edition 2013, I.D. 9.18.7).

However, in the present case no prior art is available suggesting the claimed approach, and the Board is not convinced that the skilled person, starting from document D1, on the basis of his common general knowledge alone would consider transport-layer events when implementing the detection of the "successful completion" of a synchronisation session. Indeed, document D1 is primarily concerned with communication at the (SyncML) session layer.

4.12 The Board therefore considers that the subject-matter of independent claim 1 and of corresponding independent claims 6 to 10 involves an inventive step. By virtue of their dependency on claim 1, the same applies to the subject-matter of claims 2 to 5. The sole main request hence meets the requirements of Articles 52(1) and 56 EPC.

5. Since the sole main request complies with the provisions of the EPC, the appeal is to be allowed.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the department of first instance with the order to grant a patent on the basis of claims 1 to 10, filed as "Main request" during oral proceedings before the Board, and a description and drawings yet to be adapted.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

I. Aperribay M. Rognoni

Decision electronically authenticated