

Internal distribution code:

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

**Datasheet for the decision
of 13 February 2014**

Case Number: T 2055/08 - 3.4.03

Application Number: 03775360.5

Publication Number: 1561193

IPC: G09B5/00

Language of the proceedings: EN

Title of invention:

VERSIONING ELECTRONIC LEARNING OBJECTS

Applicant:

SAP AG

Headword:

Relevant legal provisions:

EPC 1973 Art. 56

Keyword:

Inventive step - mixture of technical and non-technical
features - (yes)

Decisions cited:

Catchword:



**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 2055/08 - 3.4.03

D E C I S I O N
of Technical Board of Appeal 3.4.03
of 13 February 2014

Appellant:
(Applicant)

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf (DE)

Representative:

Müller-Boré & Partner Patentanwälte PartG mbB
Friedenheimer Brücke 21
80639 München (DE)

Decision under appeal:

**Decision of the Examining Division of the
European Patent Office posted on 19 June 2008
refusing European patent application No.
03775360.5 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman: G. Eliasson
Members: V. L. P. Frank
T. Bokor

Summary of Facts and Submissions

- I. This is an appeal against the refusal of European patent application No. 03 775 360 for lack of an inventive step (Article 56 EPC 1973).
- II. At oral proceedings before the board the appellant requested that the decision under appeal be set aside and that a patent be granted with the following documents:

Description

pages 1, 1a, 2, 2a, 22 as filed during the oral proceedings before the Board,
pages 3-21 of the application as published;

Claims

1-7 as filed during the oral proceedings before the Board, titled final version;

Drawings

Figures 1-20, sheets 1/15-15/15 as published.

- III. Claims 1, 3 and 4 of the sole request read as follows:

"1. A method for use in an electronic learning system that manages versioned learning objects in a master repository (130) storing the learning objects that are valid throughout the electronic learning system to be accessed by any user of the electronic learning system and in a local repository (132) for storing a local version of the object to create and edit new learning object versions which can be checked-in to the master repository (130) by storing,
wherein the learning objects comprise at least one referencing learning object and at least one referenced learning object which is referenced by

the referencing learning object,
wherein the referencing learning object has
metadata which stores a reference to the
referenced learning object;
the method comprising:
defining two conflict rules as follows:
- according to a first conflict rule, each
learning object version within the master
repository (130) must not reference, directly or
indirectly, any other learning object in more
than one version; and
- according to a second conflict rule, each local
repository (132) must not contain any learning
object in more than one version;
detecting a version conflict associated with a
learning object by checking for compliance with
both of the two conflict rules during each
creation of a version of the learning object and
each deletion of a version of the learning object;
and
resolving the version conflict by ensuring
compliance with the two conflict rules,
comprising:
- when a checked-in version of a learning object
is edited in a local repository (132):
-- creating a new version of the learning
object,
-- replacing the checked-in version with the new
version, and
-- updating metadata of each object referring to
the learning object with version information for
the new version of the learning object;
- when reverting to an earlier version of a
learning object:
-- deleting the most recent version of the
learning object;

- revising metadata of all objects in the local repository (132) that refer to the most recent version of the learning object to refer to the earlier version of the learning object;
- when checking-in edited versions of existing learning objects from the local repository (132) to the master repository (130), requiring each learning object version to be checked in with all recursively referenced learning objects of the learning object version."

"3. A computer program product for use in an electronic learning system, wherein the computer program product manages versioned learning objects in a master repository (130) and in a local repository (132) according to the method of claim 1 or 2."

"4. An electronic learning system (59), comprising:
a master repository (130) which stores existing versions of learning objects being valid throughout the electronic learning system to be accessed by any user of the electronic learning system;
a local repository (132) which stores alternate versions of the learning objects stored in the master repository (130), wherein the local repository (132) allows creation and editing of new learning object versions which can be checked-in to the master repository (130) by storing; and
a processor that executes instructions to perform the steps of the method according to claims 1 or 2."

IV. The following documents are mentioned in this decision:

D1 = US 6 430 563 B

D2 = US 5 675 802 A

D5 = GB 2 373 625 A

V. The examining division essentially argued that:

- The claim had technical character in so far as the subject-matter of the claim was directed to a method for use in an electronic learning system. The claim, however, comprised the steps of an administrative method which was being implemented in the electronic learning system. The claim was thus prima facie made up of technical and non-technical features.
- The terms used in the method steps of claim 1, ie "learning object", "local repository", "master repository", "cascading conflict resolution", "propagating metadata" did not confer any technical meaning to these method steps. These method steps did not go beyond an abstract concept for managing objects and references between them. The method thus defined an administrative concept as such. Administrative methods as such fell under the list of exclusions from patentability under Article 52(2) and (3) EPC.
- The technical character of the claim resided in a general purpose networked computer system and a database used for data processing and data exchange.
- The non-technical features of the method did not interact with the technical features to alter the

technical character of the claimed subject-matter from that of a generic commonplace networked computer system and a database. Hence the non-technical features of the method could be used for defining the problem to be solved and presented a non-technical requirements specification which requested the skilled person to automate the method in some way. The objective technical problem to be solved could therefore be considered as the translation of the non-technical method into a general purpose networked computer system. Implementing non-technical functions through standard programming techniques into a general purpose networked computer system was a typical task carried out by the person skilled in the art of data processing which did not involve an inventive step.

VI. The appellant applicant essentially argued as follows:

- The technical problem to be solved by the present invention could be defined as to hold a learning system clear from any learning objects referring to different versions of learning objects in the master repository, ie to avoid that inconsistent data be supplied. To solve the above technical problem the invention provided a method for managing versioned learning objects in a master repository and in a local repository wherein upon detecting a version conflict, the version conflict was resolved by means of a cascading conflict resolution. The cascading conflict resolution was achieved by propagating metadata from a referencing learning object to the referenced learning object in the master repository wherein existing references in the referenced learning

object were overwritten by the metadata from the referencing learning object. This was carried out so that each learning object in the master repository did not reference directly or indirectly any other learning object in more than one version and each local repository did not contain any learning object in more than one version.

- Moreover, the specific way of resolving conflicts, namely by propagating metadata from a referencing learning object to the referenced learning object, could not be regarded as a simple administrative concept, but rather implied technical considerations on how the required versioning was achieved and implemented. Basically, "overwriting" in the meaning of "overriding" referred to a method step which could not be denied to be of technical character. Thus, technical considerations, at least the one leading to the use of the method step "overwriting", were made to achieve the subject-matter of the present invention so that the respective features had to be taken into account when assessing inventive step.

- In order to ensure that when editing a previously checked-in object version the local repository did not contain any learning object in more than one version, conflict resolution was performed according to the steps defined in the claim. This was an example of cascading conflict resolution, since metadata that defined references between objects was propagated to each object referring to the learning object with version information for the new version of the learning object. This

propagation started from a referencing object and continued (cascaded) through references to referenced objects. The version conflict was resolved in the sense that the checked-in version of the object was replaced with the new version, so that the checked-in version of the object was no longer present in the local repository.

- To ensure that the local repository did not contain any learning object in more than one version when reverting to an earlier version of a learning object, the most recent version of the learning object was deleted from the local repository. This was also an example of cascading conflict resolution, since metadata of all objects in the local repository referring to the most recent version of the learning object was revised, i.e. overridden. The propagation of metadata started from a referencing object and continued (cascaded) through references to referenced objects.

- Finally, the last step of claim 1 described how to ensure compliance (i.e. resolve conflicts) with the first conflict rule. Without this requirement, any conflict checks might be incomplete since locally referenced objects could still be changed after check-in.

- When applying the problem-and-solution approach, the skilled person could consider document D2 to be the closest prior art. However, D2 did not disclose the compliance with the two conflict rules defined in the claim. These rules assured that the consistency of the electronic learning system was maintained and that learning objects

could be easily and flexibly reused. It further allowed the separation of version conflict checking into two parts, making the version conflict detection step easy and efficient. Accordingly, the skilled person was confronted with the objective technical problem of how to maintain the consistency of an electronic learning system in an easy and efficient way. In order to solve this problem, the skilled person could start from D2. However, D2 disclosed that consistency could be maintained by ensuring that only a single user at a time modified a given branch, thereby avoiding version conflicts. In contrast to D2, claim 1 provided the skilled person with a different approach for solving the problem above. According to claim 1, version conflicts were detected by checking for compliance with two conflict rules and the conflict detection was performed at specific times, i.e. during creation and deletion of a version of a learning object. Any conflicts were thus resolved by ensuring compliance with the two conflict rules. Compliance with the second conflict rule was ensured at object check-in and when reverting to an earlier version of an object. Compliance with the first conflict rule was ensured by making sure that all referenced objects were checked-in with their referencing object. This ensured that all the dependencies of an object were present in the repository when the object itself was checked-in, so that the dependent objects were not changed after their referencing object had been checked-in. Such an approach was neither taught nor indicated by D2. In order to solve the objective technical problem the skilled person could also

consult D1. However, D1 did not provide any suggestions how to resolve version conflicts.

Reasons for the Decision

1. The appeal is admissible.
2. *Amendments*
 - 2.1 Independent method claim 1 was amended to include a definition of the master and the local repository. These definitions are based on page 10, lines 20 to 27 and page 11, line 32 to page 12, line 3, respectively. Claim 1 was further amended to specify the two conflict rules, the detection of a version conflict by checking compliance with these rules and the steps required for resolving a version conflict. These amendments are based on pages 14 to 16, "Version Conflicts" and "Check-In Process".
 - 2.2 It was clarified in claim 3 that it is the computer program product that manages the learning objects in accordance with the method of claim 1.
 - 2.3 Claim 4, directed to an electronic learning system, was amended to specify that its processor performs the method of claim 1.
 - 2.4 The "Background" of the invention on page 1 of the description now acknowledges the prior art documents D1, D2 and D5. The "Summary" of the invention on page 2 brings the description in concordance with the new claims.

2.5 Hence the board is satisfied that the requirements of Article 123(2) EPC are met.

3. *Inventive step (Article 56 EPC 1973).*

3.1 The only remaining issue in this appeal is that of inventive step.

3.2 The examining division considered that the method of claim 1 was composed of technical and non-technical features. In particular, it found that the method was an "administrative method", which did not go beyond an abstract concept of managing objects and references between them, that was implemented on a general purpose networked computer system and a database used for data processing and data exchange. As the technical and non-technical features did not interact with each other to alter the technical character of the claimed subject-matter, the "administrative method" could be used in the formulation of the problem to be solved as a requirements specification which required the skilled person to implement the "administrative method", ie to write the corresponding program code, for the general purpose networked computer system. This activity was considered not to involve an inventive step.

3.3 The board does not share the examining division's assessment, for the following reasons. The application addresses the problem of providing a versioned and referenced collection of digital objects (the "learning objects") that is free from version conflicts, ie free from inconsistencies. This is achieved by providing the referencing objects with metadata defining references to the referenced objects and using the metadata to solve versions conflicts by propagating them from the referencing to the referenced objects. The application

thus addresses a technical problem, the internal consistency of a data structure for the purposes of a specific application (in this case an electronic learning system), and solves it by technical means, ie by using metadata for detecting references to a learning object in more than one version. Metadata are "data over data". However, this does not deprive them from having technical character when used for purposes involving technical considerations. Thus the use of metadata for resolving version conflicts in a collection of digital objects involves technical considerations.

3.4 The board considers that document D5 represents the closest state of the art since it relates to an electronic learning system. Documents D1 and D2 are not related directly to electronic learning systems. Instead document D1 relates to a management and storage system of multiple versions and context variants of objects which are retrieved on the basis of a context resolution mechanism (D1, column 1, lines 14-20, column 2, lines 53-67), whereas document D2 relates to a geographically distributed software development system that allows parallel development of different branches of software at different sites (D2, column 3, lines 11-26).

3.5 Document D5 discloses an electronic learning system comprising a repository 600 for storing electronic learning objects together with metadata describing them at different levels and a method for allowing non-technical people to easily create, assemble and deploy digital learning material (page 1, line 19 to page 2, line 3; lines 22-26; Figure 2). A learning programme is constructed by the learning programme designer from the learning objects stored in the repository 600 making

use of the metadata describing the learning objects and the specification of the learning programme (page 6, lines 11-28).

3.6 The method of claim 1 differs from the method disclosed in document D5 in that versions of learning objects are managed in a master and in a local repository, the local repository being used for creating and editing new learning object versions which can be later checked-in to the master repository. The method defines two conflict rules. Version conflicts are detected by checking for compliance with these rules during each creation or deletion of a version of the learning objects. The detected version conflicts are resolved when editing a checked-in learning object in the local repository, when reverting to an earlier version of a learning object in the local repository and when checking-in edited versions of existing learning objects from the local repository to the master repository. In the learning system of document D5, on the contrary, it is not foreseen to have different versions of the same learning objects coexist in the repository.

3.7 Hence the objective technical problem can be seen in improving the flexibility for developing new learning courses by reusing existing learning objects while avoiding that version conflicts arise in the course of this development by referencing the same learning object in more than one version.

3.8 This is achieved by providing a local repository in which a course developer may use existing versions of learning objects and combine them at will. This improves the flexibility of course development, in particular, as several local repositories may be used

in parallel. Compliance with the two conflict rules when creating or deleting versions of existing learning objects keeps the local repository free from version conflicts. Checking-in each learning object with all recursively referenced learning objects assures that the learning objects in the master repository do not reference any other learning object in more than one version, keeping the master repository free from version conflicts.

In the learning system of document D5, each learning object is stored in a single version in the repository. Therefore the problem of version conflicts does not arise.

3.9 Documents D1 or D2 do not deal with version conflicts.

Document D1 allows for the simultaneous existence of different context specific variants of a digital object, eg different language versions of a manual, and retrieves the most appropriate object using a context resolution mechanism through correlation of attributes of the digital objects and attributes of a front-end client application (D1, Abstract).

Document D2 control files at a local development site within a geographically distributed multisite software development project by allowing a single user to modify locally a single branch at a time, creating thus different versions of a file. These versions however are a current and several historical versions of the file and do not conflict with each other. A master exchanger periodically updates the local replica by providing new versions of target file to geographically remote development sites and receiving any additional versions of the files or branches created at such

remote sites (D2, Abstract, column 3, line 11 to column 4, line 5).

Hence, the skilled person could not find in documents D1 or D2 any incentives to solve the objective technical problem addressed by the present application.

- 3.10 The board is satisfied for these reasons that the method of claim 1 involves an inventive step within the meaning of Article 56 EPC 1973. Since the computer program product of claim 3 and the electronic learning system of claim 4 implement the method of claim 1 they involve an inventive step for the same reasons.

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 with the following documents:

Description:

pages 1, 1a, 2, 2a, 22 as filed during the oral proceedings before the board,
pages 3-21 of the application as published;

Claims:

1-7 as filed during the oral proceedings before the board, titled final version;

Drawings:

sheets 1/15-15/15 as published.

The Registrar:

The Chairman:



S. Sánchez Chiquero

G. Eliasson

Decision electronically authenticated