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 15 November 2016

Case Number:	T 0661/11 - 3.5.07
Application Number:	04014741.5
Publication Number:	1460565
IPC:	G06F17/30
Language of the proceedings:	EN

Title of invention:
Method and system for uniformly accessing multiple directory services

Applicant:
Microsoft Technology Licensing, LLC

Headword:
Uniform directory-service access/MICROSOFT TECHNOLOGY LICENSING

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - main request (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 0661/11 - 3.5.07

DECISION
of Technical Board of Appeal 3.5.07
of 15 November 2016

Appellant:
(Applicant)

Microsoft Technology Licensing, LLC
One Microsoft Way
Redmond, WA 98052 (US)

Representative:

Grünecker Patent- und Rechtsanwälte
PartG mbB
Leopoldstraße 4
80802 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 10 November 2010 refusing European patent application No. 04014741.5 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman	R. Moufang
Members:	R. de Man
	M. Rognoni

Summary of Facts and Submissions

I. The applicant (appellant), which at the time was Microsoft Corporation, appealed against the decision of the Examining Division refusing European patent application No. 04014741.5. The application is a divisional application of European application No. 97110830.3 (the parent application).

II. With effect from 2 February 2015 the EPO registered a transfer of the application to Microsoft Technology Licensing, LLC, which thereby acquired the status of appellant.

III. The contested decision cites the following documents:

D1: Martin E.: "X/Open Federated Naming", Hewlett-Packard Journal, Vol. 46, No. 6, pp. 28-33, December 1995;

D2: Droms R.: "Access to Heterogeneous Directory Services", Proceedings IEEE INFOCOM '90, pp. 1054-1061, 1990;

D3: US 5377323, 27 December 1994;

D4: "Federated Naming: The XFN Specification", X/Open CAE Specification, ISBN 1-85912-052-0, July 1995; and

D5: Mansfield G. et al.: "Schema Publishing in X.500 Directory", ASID Working Group Internet Draft, March 1995.

The Examining Division decided that the subject-matter of claim 1 of a main request and of first and second auxiliary requests lacked inventive step in view of a combination of documents D1, D4 and D5.

IV. With the statement of grounds of appeal, the appellant resubmitted the main request and the first and second auxiliary requests considered by the Examining Division. It argued why the subject-matter of their independent claims involved an inventive step.

V. In a communication annexed to a summons to oral proceedings, the Board introduced the following document:

D6: Benford S.: "Dynamic Definition of Entries and Attributes in the Directory Service", Proceedings 8th International Conference on Distributed Computing Systems, pp. 563-568, June 1988.

The Board expressed the preliminary opinion that the subject-matter of claim 1 of the main request and of the first and second auxiliary requests lacked inventive step. It further raised objections under Articles 84 and 123(2) EPC with respect to claim 1 of the main request and under Article 76(1) EPC with respect to claim 7 of the main request.

VI. With a letter dated 12 August 2016, the appellant replaced its requests with an amended main request and amended first and second auxiliary requests. It explained why the amended requests complied with the requirements of the EPC.

VII. In the course of oral proceedings held on 15 November 2016, the appellant replaced its main request with a new main request. At the end of the oral proceedings, the chairman pronounced the Board's decision.

VIII. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of claims 1 to 9 of the new main request filed during the oral proceedings or, if that was not possible, on the basis of the first or second auxiliary request filed with letter dated 12 August 2016.

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

"A directory service system adapted to provide uniform access to a plurality of various different directory services (409, 412, 415), each directory service having information relating to the information for each object (410, 413, 416) being defined by an object class, each object class having properties defined by a schema object of said directory service system, a schema object for each object class being contained in a schema container object of said directory service system for each directory service, the schema container object being assigned a pre-defined name within the namespace of the respective directory service, the definition of the various object classes being determinable at run time by enumerating the schema objects contained in the schema container object, additional object classes being definable by adding schema objects to the schema container object, each property having a property name and property type, the information for each object being a property value for each property of its object class, comprising:

a schema browsing component adapted to retrieve the property name and property type of each property of each object class of each directory service, wherein the schema browsing component is adapted to be used by a client (406, 407) of the directory service system to retrieve property names and property types of the object classes;

a name resolving component adapted to receive a unique identifier of an object within a directory service and locate the object within the directory service;

a binding component adapted to bind to an in-memory object representing a located object within the identified directory service; and

an extending component adapted to define new object classes and new properties for each directory service, wherein the extending component is adapted to be used by a client of the directory service system to define new object classes and new properties."

Claims 2 to 6 are dependent on claim 1.

Claim 7 of the main request reads as follows:

"A directory service method to provide uniform access to a plurality of various different directory services, each directory service having information relating to the information for each object being defined by an object class, each object class having properties defined by a schema object of said directory service system, a schema object for each object class being contained in a schema container object of said directory service system for each directory service, the schema container object being assigned a pre-defined name within the namespace of the respective directory service, the definition of the various object classes being determinable at run time by enumerating the schema objects contained in the schema container object, additional object classes being definable by adding schema objects to the schema container object, each property having a property name and property type, the information for each object being a property value for each property of its object class, comprising:

operating a schema browsing component (601-604, 701, 801-812, 1001-1008) for retrieving the property name and property type of each property of each object class of each directory service by a client of the directory service system;

operating a name resolving component (501-504) for receiving a unique identifier of an object within a directory service and for locating the object within the directory service;

operating a binding component (505) for binding to an in-memory object representing a located object within the identified directory service; and

operating an extending component (901) for defining new object classes and new properties for each directory service by a client of the directory service system."

Claim 8 is dependent on claim 7.

Independent claim 9 is directed to "A computer-readable medium having instructions to cause a computer system to perform the method of claim 7 or 8".

X. The remaining application documents for the main request are as follows:

Description:

- pages 1 to 5, 7, 9 to 16, 18 to 20, 22 to 25 and 27 to 29 as originally filed;

- pages 6, 8, 8b, 21 and 26 as filed with letter of 20 October 2006; and
- pages 8a and 17 as filed with letter of 12 August 2016.

Drawings:
- sheets 1/11 to 11/11 as originally filed.

XI. The text of the claims of the first and second auxiliary requests is not relevant to this decision.

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 background section of the application explains that various vendors provide directory service systems. Directory services provide information repositories and are accessible by clients over a network. The information stored is generally arranged as a hierarchy of "objects", each object having a unique identifier within this hierarchy. If in this hierarchy a first object is the child of a second object, the second object is said to "contain" the first object.

Information is associated with an object in the form of a set of property values. The properties that a particular object has are determined by the "object class" to which the object belongs. In the example object hierarchy shown in Figure 1, object 101 belongs to the "Company" object class. Objects in this class have properties "Name" and "Address". The value of the property "Name" of object 101 is "MS". Object 102 belongs to the object class "Division". It has a property "Name" with value "System".

The unique identifier of an object typically corresponds to the path from the root object of the hierarchy to the object itself. The identifier of object 102, which is connected to the root object via object 101 (i.e. the root object contains object 101, and object 101 contains object 102), is "Company=MS \Division=System". The conceptual space of all such identifiers is referred to as the "namespace" of the directory service.

2.2 To allow clients (i.e. client applications) to access its directory service system, each vendor designs and implements its own application programming interface (API). Although API sets of different directory services generally provide similar functionality, they are still different. A client that needs to use different directory services is therefore required to implement code for each vendor-specific API set.

2.3 The invention addresses this problem by proposing a directory service system that provides a uniform interface for accessing a variety of directory services.

2.4 This uniform interface provides clients with name resolving and binding functionality for locating a directory object on the basis of its unique identifier and binding to an in-memory object representing the located object.

2.5 To allow clients to learn about the properties an object may have as defined by its object class, the uniform interface of the directory service system further provides access to "schema objects". Each schema object corresponds to an object class and defines the properties of the object class. The schema objects corresponding to the object classes of a particular directory service are all placed in a "schema container object" for the directory service. This schema container object is assigned a pre-defined name within the namespace of the directory service. Thus, clients are provided with a uniform way of inspecting the object classes of a particular directory service by locating and binding to the schema container object of the directory service and enumerating the (schema) objects contained therein.

2.6 The uniform interface also allows clients to define new object classes for a particular directory service by adding schema objects to the schema container for the directory service.

3. *Main request - clarity and added subject-matter*

3.1 Claim 1 corresponds essentially to originally filed claim 1 with amendments introducing the "schema container objects", "schema objects" and associated functionality discussed in points 2.5 and 2.6 above. These amendments are based on page 11, line 27, to page 12, line 12, of the description as filed.

Although this passage states that "Conceptually, each directory service is viewed as having a schema container object that contains a schema object for each object class", the skilled reader understands unambiguously from the passage as a whole that the schema container object for a directory service is presented to clients of the directory service system as "real" objects of the directory service system existing within the namespace of the directory service.

Thus, the subject-matter of independent claim 1 has a basis in the application as filed as required by Article 123(2) EPC. Since claim 1 as originally filed is identical to original claim 1 of the parent application and the above-cited passage is also contained in the parent application, the subject-matter of present claim 1 likewise has a basis in the parent application (Article 76(1) EPC).

3.2 Independent method claim 7 as amended essentially defines a method of operating the directory service system of claim 1 and is supported, for example, by the statement of the technical field of the invention ("a method and system for uniformly accessing the directory service") in both the parent application and the present application.

3.3 Dependent claims 2 to 6 and 8 and independent claim 9 are supported by the corresponding claims of both the parent application and the present application.

3.4 Thus, the main request complies with Articles 76(1) and 123(2) EPC.

3.5 Furthermore, the amendments to the claims have overcome the Board's outstanding clarity objections (Article 84 EPC).

4. *Main request - inventive step*

4.1 In its decision, the Examining Division considered document D1 in combination with document D4 to represent the closest prior art for the subject-matter of claim 1.

Document D1 is a paper presenting the "X/Open Federated Naming" (XFN) specification, which defines uniform naming interfaces for accessing a variety of naming systems (see abstract). The document explains on page 28, left-hand column, second to fourth paragraphs, that a variety of heterogeneous naming systems exist. Each of these naming systems has its own API, which forces application programmers to write custom software for each naming system that their applications use. To address this problem, the XFN specification was developed, which has become part of the "X/Open Common Application Environment (CAE)" (page 28, right-hand column, second full paragraph). The XFN API is layered over the APIs of specific naming services and thus hides the details of the underlying naming systems. This allows applications using the XFN API to access a variety of naming systems without modification (page 29, right-hand column, third paragraph).

Document D4 is an "X/Open CAE Specification" titled "Federated Naming: The XFN Specification".

4.2 The appellant contested that documents D1 and D4 could be regarded as a single disclosure.

The Board agrees that, although document D1 discusses the XFN specification, the details of that specification disclosed in document D4 do not for that reason form part of the disclosure of document D1. Document D1 in principle stands on its own; the skilled reader of document D1 will complement the document's disclosure with details from document D4 only in so far as he is taught to do so by document D1.

4.3 Nevertheless, document D1 discloses a system providing uniform access to a plurality of heterogeneous naming systems, among which are directory services such as the X.500 directory service and the DCE Cell Directory Service (page 28, left-hand column, second paragraph). By means of the "XFN base context interface", clients can *inter alia* look up a name to obtain a reference, bind to the reference and manipulate attributes of objects (page 29, right-hand column, fourth paragraph, to page 30, left-hand column, first full paragraph). The document therefore represents a suitable starting point for assessing inventive step.

As the Examining Division's analysis confirms, document D1 does not disclose - not even when read together with document D4 - the presence of a "container object" having a pre-assigned name within the namespace of each directory service, the container object containing a schema object for each object class of the directory service.

4.4 As explained in the paragraph bridging pages 11 and 12 of the description of the application, these distinguishing features allow a client to determine the definition of object classes at run time and to define additional object classes in a uniform manner for all directory services.

The objective technical problem solved by these features may thus be seen as that of allowing a client to discover and manipulate object classes of directory services at run time.

4.5 According to the contested decision, the distinguishing features are disclosed by document D5.

Document D5 relates to the X.500 directory service and proposes a solution to the "schema distribution problem" using the existing mechanisms of the directory, and it describes procedures "for fetching unknown schema from the directory at runtime" (see abstract). Although the document concentrates on schema access/retrieval, it mentions that since schema objects are defined and employed, "the modification, addition and deletion of schema objects can be carried out using existing directory mechanisms" (page 4, first paragraph). The skilled person in search of a solution to the above-mentioned problem would therefore consult document D5.

The solution presented in document D5 consists of a naming scheme for naming schema objects and a meta-schema for storing schema objects in the directory (see page 3, third and fourth paragraphs). It is described in section 3.1 in the context of a directory service comprising multiple distributed Directory Service Agents (DSAs). This section explains that schema information is stored in a distributed manner, each naming context storing the schema relevant to it (page 4, last paragraph). As stated on page 6, first paragraph, a "subschema" entry is stored below the root entry of each naming context. Schema information relevant to that naming context is stored below the subschema entry. This subschema entry hence plays a role similar to that of the schema container objects of the invention. But whereas the invention proposes a schema container object containing a subschema object for each object class of a directory service, document D5 discloses that schema objects are stored in subschema entries distributed throughout the directory tree.

This is confirmed by section 4 of document D5, which describes the retrieval of schema from the directory. As explained on page 7, first and second paragraphs, if an object of an unknown object class is encountered (with name "IPNI=spark, OU=Department of Computer Science, O=Indian Institute of Technology, Madras, C=IN"), then the schema information for its object class is sought first in a subschema entry located in the container object of the object ("oid=IPNI, CN=subschema, OU=Department of Computer Science, O=Indian Institute of Technology, Madras, C=IN"). If this fails, an attempt is made to find the schema information in a subschema entry located one level higher in the directory tree ("oid=IPNI, CN=subschema, O=Indian Institute of Technology, Madras, C=IN"), and so on. The various schema objects of the directory service are thus indeed distributed over multiple subschema entries.

Hence, the skilled person applying the teaching of document D5 to a system according to document D1 would not arrive at a schema container object that, as required by claim 1, contains a schema object for each object class of a directory service.

Thus, the subject-matter of claim 1 is not rendered obvious by a combination of documents D1 and D5 (nor by a combination of documents D1, D4 and D5).

4.6 Document D6 relates to directory services and describes "attribute definitions" and "entry definitions", which are intended to allow clients of a directory service ("Directory User Agents") to understand the structure of the directory information "at any time" (see abstract; page 563, right-hand column, third paragraph; section 1.2).

Entry definitions are generic descriptions of directory entries grouped by object class (see page 564, left-hand column, lines 37 to 39), i.e. they are object class definitions. Attribute definitions describe the structure of attributes within the directory information base (page 564, left-hand column, lines 48 to 51), for which the present application uses the term "properties" (see page 13, last paragraph, of the description of the application).

Sections 2.4 and 3.3 describe operations for adding, deleting and reading attribute and entry definitions.

4.7 According to page 564, left-hand column, lines 29 to 39, document D6 proposes extending the directory data model to include attribute definitions and entry definitions. Furthermore, these definitions "must be created dynamically via the Directory Access Protocol (DAP)".

This passage cannot, however, be understood as unambiguously disclosing that attribute and entry definitions are themselves stored as "entries" or "objects" in the directory information tree, which would allow them to be read and manipulated by means of existing operations of the directory access protocol. In fact, section 2.4 makes clear that the directory access protocol is *extended* to include operations for adding, deleting and reading attributes. The skilled reader of document D6 understands that the same holds true for the operations on entry definitions listed in section 3.3.

Hence, document D6 does not unambiguously disclose that object class definitions are to be stored as schema objects in the directory information tree. And in the Board's view, the skilled person trying to apply the teaching of document D6 to a system according to document D1 would not necessarily be led to store the "entry definitions" in the directory information tree, let alone in a pre-assigned container object per directory service; he could, instead, store these definitions in a repository separate from that tree.

Thus, the subject-matter of claim 1 is likewise not rendered obvious by a combination of documents D1 and D6.

4.8 In its communication, the Board developed an approach starting from the prior art acknowledged in the background section of the application. Since this approach leads to the same distinguishing features as found when starting from document D1 (or a combination of documents D1 and D4), it cannot lead to the conclusion that the subject-matter of claim 1 lacks inventive step. Similarly, documents D2 and D3 are not closer to the subject-matter of claim 1.

4.9 Hence, the subject-matter of claim 1 of the main request involves an inventive step within the meaning of Articles 52(1) and 56 EPC.

5. In view of the above, the claims of the main request satisfy the requirements of the EPC. However, the description and drawings may still require adaptation. In particular, the description on pages 26 to 29 contains embodiments which appear not to be covered by the independent claims.

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 9 of the main request filed during the oral proceedings and a description and drawings still 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 R. Moufang

Decision electronically authenticated