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

Datasheet for the decision of 2 December 2011

Case Number:	T 1043/08 - 3.5.06
Application Number:	04021618.6
Publication Number:	1526427
IPC:	G06F 1/00
Language of the proceedings:	EN

Title of invention:
Programming interface for licensing

Applicant:
MICROSOFT CORPORATION

Headword:
Programming interface for licensing/MICROSOFT

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

Keyword:
"Inventive step - all requests (no)"

Case Number: T 1043/08 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06
of 2 December 2011

Appellant:
(Applicant)

MICROSOFT CORPORATION
One Microsoft Way
Redmond
Washington 98052-6399 (US)

Representative:

Grünecker, Kinkeldey
Stockmair & Schwanhäusser
Leopoldstrasse 4
D-80802 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 11 December 2007 refusing European application No. 04021618.6 pursuant to Article 97(1) EPC 1973

Composition of the Board:

Chairman: D. H. Rees
Members: M. Müller
M.-B. Tardo-Dino

Summary of Facts and Submissions

I. The appeal lies against the decision of the examining division, posted on 11 December 2007, to refuse the European patent application 04021618.6.

II. An appeal was filed on 20 February 2008 and the appeal fee was paid on the same day. A statement of grounds of appeal was filed on 21 April 2008. It was requested that the decision be set aside and that a patent be granted based on the main or 1st auxiliary request subject to the decision or, as a 2nd auxiliary request, on the basis of a new set of claims filed with the grounds of appeal.

III. With summons to oral proceedings, the board referred to the following documents:

D1: US 5,204,897 A
D6: Oberg R. J. *et al.*, "Application Development using Visual Basic and .NET", Prentice Hall, 2002; Excerpt relating to Asynchronous Programming

and explained its preliminary opinion that the independent claims of all requests appeared to lack an inventive step over D1 in view of D6 and common knowledge in the art. In addition, objections under Articles 84 EPC 1973 and 123(2) EPC were raised.

IV. In response and with letter of 2 November 2011, the appellant filed two amended sets of claims to replace all claims on file so as to overcome the objections under Article 84 EPC 1973 and 123(2) EPC.

During the oral proceedings the appellant requested that a patent be granted based on the following application documents:

description, pages	
1, 1a-1c	received on 10 October 2006
28	received on 23 October 2007
2-27	as originally filed
claims, numbers	
1-12	according to a main or an auxiliary request, labelled "No. 1" and dated 2 November 2011
drawing, sheets	
1/5-5/5	as originally filed

V. Claim 1 according to the main request reads as follows:

"A system (110) for supporting the enforcement of a license for a computer program, the system comprising:

a licensing component (202) that maintains a license store (204) in which the license is stored, the license comprising a right in the software and a set of data associated with said right, the licensing component exposing a callable interface to the computer program (135), said callable interface comprising:

a right-consumption method which receives an identifier of said right from the computer program and determines whether the right can be exercised; and

an information-retrieval method which receives an identifier of said right from the computer program and

provides said set of data, or information based on said set of data, to the computer program,
wherein said licensing component (202) is usable by a plurality of computer programs, the computer program being included among said plurality of computer programs, wherein said callable interface further comprises:

a handle-opening method that provides a handle to the computer program;

wherein the rights-consumption method receives the handle from the computer program and uses the handle to identify the computer program from which a call to the rights-consumption method is received,

wherein the license is one of a plurality of licenses that are stored in said license store, and wherein the rights-consumption method causes the licensing component to select the license based on one or more factors comprising:

whether the license store is associated with the computer program; and

a conflict rule that determines which license to select from among a plurality of license that are associated with the computer program,

and wherein said callable interface further comprises:

an asynchronous-context-initiator method that establishes a context for asynchronous processing and provides an identifier of said context to the computer program;

wherein said rights-consumption method receives the identifier of said context from said computer program and processes a right-consumption request asynchronously in response to receipt of the identifier of said context."

Claim 1 according to the auxiliary request differs from that of the main request only in the paragraph defining the rights-consumption method, now reading as follows:

"a right-consumption method which receives an identifier of said right from the computer program and determines whether the right can be exercised by checking (a) a binding of the license to a product key of a computer program, (b) a binding of the license to the system (110) on which the computer program is running, and (c) a validity that the right has not expired, wherein the computer program is notified on the result of the determination of whether the right can be exercised;"

VI. At the end of the oral proceedings, the chairman announced the decision of the board.

Reasons for the Decision

Articles 84 EPC 1973 and 123 (2) EPC

1. The board is satisfied that the amended claims fulfill the requirements of Article 123(2) EPC and that the

amendments also overcome the objections under Article 84 EPC raised in the summons.

The Invention

2. The invention concerns a "callable" application programming interface (API) that enables computer programs to access software licensing services in a uniform manner. According to the invention, a central store holds licenses which express whether and under what conditions an individual computer program has the right to perform certain functions. Programs in need of a specific right can request it from the licensing service which will inform the caller whether a suitable license exists or not.

2.1 According to independent claim 1 of both requests, the interface comprises a right-consumption method, an information-retrieval method and a handle-opening method.

2.2 According to claim 1 of the main request, the right-consumption method "determines whether the right can be exercised". According to claim 1 of the auxiliary request this involves checking for a license which is correctly bound to a particular computer program and system and has not yet expired. The description explains the term "consume" to mean the "exercise of a specified right" (see par [0006]). It is a design decision that the licensing server in general and the right-consumption method in particular do not enforce the rights in question, but that the caller is relied upon to respect the license or its absence (see par. [0007]). As a consequence, "consuming" or "exercising" a right does not mean exercising the action to which the right refers: For example, exercising a print right does not subsume the printing.

2.3 The handle-opening method provides a "handle" to the calling computer program which is used subsequently to identify the computer program in calls to the rights-consumption method and, as the description explains, to other methods of the interface (cf. hSLC *e.g.* in pars. [0032]-[0034], [0066]-[0067], [0087]-[0088]).

2.4 The information-retrieval method returns information associated with the pertinent right. While the claims do not specify this information in more detail, the description explains that it may contain information about the permissibility of performing certain actions within the "consume[d] right" (par. [0008]).

Inventive Step

3. It is common ground that D1 is a suitable starting point for the assessment of inventive step.

3.1 D1 discloses a licensing infrastructure (see esp. fig. 7) with "licensing component" (no. 10) maintaining a "license store" (no. 23) which can be accessed by a plurality of computer programs (nos. 17a and 17b) to assess whether they can exercise certain required rights. Access to the licensing component is through a "callable interface" (col. 21, lines 20-24).

3.2 The interface provides a method lm_request_allocation which is called to determine whether the use of a desired product or product feature is (in principle) permitted (col. 21, lines 44-49). It is determined whether sufficiently many so-called "license units" are available to authorize the requested use (col. 14, line 67 - col. 15, line 3). If so, the appropriate number of license units is either "allocated" - in which case they are "unallocated" after use - or "consumed" - in which case they cannot be returned (cf. col. 15, line 17-63).

3.3 If a call to lm_request_allocation succeeds, a "grant handle" is returned that can be referred to later (col. 22, lines 2-8). For example, the grant handle can be passed to the method lm_query_allocation, *inter alia* to obtain further information about the grant (col. 22, lines 51-53). Such information may relate either to conditions under which the right can be exercised (cf. "product use authorization"; col. 22, lines 55-58) or whether transitive licensing is allowed (cf. "calling card request"; col. 22, lines 55-68, col. 25, line 62 - col. 26, line 30, and col. 8, lines 4-22).

3.4 While thus both methods of D1, lm_request_allocation and lm_query_allocation, determine whether a program may exercise a desired right, only lm_request_allocation actually selects a license from the license store (see *e.g.* col. 24, lines 27 ff.) while lm_query_allocation allows its further inspection. Accordingly, the board agrees with the appellant that the claimed right-consumption method corresponds to lm_request_allocation whereas the claimed information-retrieval method corresponds to lm_query_allocation (cf. grounds of appeal, *e.g.* p. 3, 4th par.).

3.5 D1 discloses that both methods must identify the relevant right: lm_request_allocation the required "license units" and lm_query_allocation the "subject argument" (col. 21, lines 44-46 and col. 22, lines 55-58). It is further disclosed that a call to lm_request_allocation must identify the calling program via a so-called "product identifier" (cf. col. 24, line 27-33). This product identifier corresponds in the present description to the "application GUID" which identifies the application to the handle opening method (cf. pars. 32-33).

3.6 The "grant handle" of D1 is generated by the license server and used for subsequent API calls (col. 22, lines 24-26 and 53-55), in particular those for accessing information about granted rights. In this regard hence the grant handle plays the same role as the handle according to the invention. However, the grant handle of D1 is returned only *after* a successful grant allocation request, whereas the handle according to claim 1 is produced by a "handle-opening method" *before* the right consumption method is called.

3.7 The license store may contain several licenses associated with the same computer program from which according to certain criteria a suitable one will be selected (cf. col. 24, lines 34 ff.). In the board's view, these criteria qualify as "conflict rules" as claimed.

4. The appellant argued in oral proceedings that for a program to exercise a right it would be obligatory to call both lm_request_allocation and lm_query_allocation. According to the appellant the call to lm_query_allocation was necessary so as to ensure that

license units which lm_request_allocation may have allocated at some point were still available for the calling program when needed later and had not meanwhile been "unallocated".

4.1 The board cannot follow this argument for two reasons. Firstly, D1 explicitly discloses that the call to lm_query_allocation is "optional and in most cases not needed" (col. 23, lines 18-21). Secondly, neither the claims nor, in fact, the description specify precisely what happens during the rights-consumption method - which could, hence, include the allocation and/or consumption of license units according to D1; And the claims leave open what the calling program is to do with a granted right (see also point 2.2 above) so that the claims do not exclude the possibility that a second method call might be obligatory to exercise a right.

4.2 Therefore, even if D1 were interpreted to disclose both method calls as obligatory, the board does not see that this would establish a difference over the independent claims of both requests.

Main Request

5. In the board's view, claim 1 according to the main request differs from D1 by the following features.

a) D1 does not disclose a handle-opening method and a handle as claimed.

b) The claimed interface comprises an "asynchronous-context-initiator method" establishing a "context for asynchronous processing".

c) The right-consumption method executes asynchronously if it receives such a context as an argument.

6. It is a common programming paradigm to organize access to data structures via handles and an associated open function. In particular this is the case for files - which the description appears to acknowledge when stating that the handle should operate "like a file handle" (see par. 41) - but also for sessions, threads, devices, objects or other data structures. It is also well-known in the art to have integrated methods for opening and directly accessing a data structure.

6.1 The claims specify an open method separate from a right consumption method whereas D1 can be viewed as integrating the open function into the request allocation method. The board considers these to be obvious alternatives which would be immediately available to the skilled person.

6.2 More specifically, if the skilled person were to extend the interface of D1 by additional methods, which is in the board's view *per se* an obvious desirable, for example by one which would allow a client to inspect the reason why a request could not be granted, the board judges that it would be obvious for the skilled person to consider rearranging the interface of D1 by providing a handle and a handle-opening method as claimed.

7. Regarding features b) and c) the appellant argues essentially as follows. The initial determination whether a caller has a license to exercise a desired

right is a costly procedure (grounds of appeal, p. 3, last par.). According to D1, the caller can have the initial determination via lm_request_allocation executed early and perform only the supposedly less costly rights inspection via lm_query_allocation during program run-time. The two methods of D1 thus provide some "kind of asynchronous processing" (grounds of appeal, p. 4, lines 14-16), and the claimed method via features b) and c) provides an alternative and non-obvious way to avoid blocking of the calling program.

7.1 The board does not follow this argument. The claimed invention also requires an "information-retrieval" method for the inspection of a granted right (cf. pars. 87-94: SLGetInformation) which can be executed separately and possibly long after the right-consumption method. These two methods provide the same "kind of asynchronous processing" which the appellant attributes to D1 and which thus cannot be identified with the claimed asynchronous processing of the right-consumption method itself. Also, D1 does not disclose when precisely the method lm_request_allocation is called, except "before use of a licensed product or product feature" (col. 21, lines 55-57) which could be at any time during run-time of the caller. If it is costly to check for a license then a synchronously executing lm_request_allocation in D1 would block the caller in much the same way as the right-consumption according to the invention.

7.2 The board agrees with the decision under appeal that features b) and c) address the problem that the caller of lm_request_allocation may want to avoid blocking.

7.3 It is commonly known in the art that asynchronous processing can solve this problem. Also D6 mentions this fact in general terms (p. 1, 1st par. in section "Asynchronous Programming").

7.4 In the board's view it is also obvious for the skilled person to observe that some callers may be able to tolerate blocking (or prefer blocking to the synchronisation effort involved in asynchronous processing) while others do not. The board thus sees it as an obvious measure to enable the caller to choose itself whether or not lm_request_allocation should execute asynchronously.

7.5 The board concedes that there are several ways of implementing this functionality, but considers that the claimed implementation follows in an obvious manner from certain design decisions required from the skilled person.

7.5.1 First, the program developer will have to decide whether asynchronous processing should be an option for all methods or only for certain ones. D6 discloses a scheme applicable to any given method, while the invention supports asynchronous processing only for the individual right-consumption method. The solution according to D6 is more flexible since the method definitions need not anticipate the possibility of asynchronous processing, while the invention provides a dedicated additional parameter to right-consumption just to support this possibility. This flexibility of D6 however comes at the price of requiring a complex dedicated library infrastructure. In the board's view, the choice between these alternatives according to circumstances is obvious for a person with the pertinent programming skill.

7.5.2 Second, if only individual functions should be executable asynchronously and this is to be supported by a dedicated parameter, there are further design decisions to make, such as what type such a parameter should have, where and when the asynchronous context (conventionally called a thread or a process) should be generated and whether that context should remain available for later reference, for instance in order to cancel it before termination (see *e.g.* D6, CancelXXX as mentioned in the middle of p. 1).

The invention claims a specific solution, according to which the context is created before the call to right-consumption and then passed to that function to request asynchronous execution (within that context). An alternative would be, for instance, to provide a Boolean argument to indicate that asynchronous processing is required but have the right-consumption method generate the context and return it to the caller for later reference. Other alternatives exist.

Each of these alternatives makes specific trade-offs between flexibility and simplicity of use for the caller and development effort on the side for the library developer. In the board's view, a person with the required programming skill would make such trade-offs as a matter of course.

7.6 The board therefore concludes that the independent claims of the main request would have been obvious over

D1 for the person skilled in the art and hence do not comply with Article 56 EPC 1973.

7.7 In view of this result it can be left open whether, as the appealed decision argues (point 10.5), the claimed implementation details of the asynchronous processing have a technical effect and could, therefore, in principle contribute to inventive step at all.

Auxiliary Request

8. Over claim 1 of the main request, claim 1 of the auxiliary request adds the features that the calling computer program is notified whether the right can be exercised or not, and makes explicit that the right-consumption method checks (a) a binding of the license to a product key of a computer program, (b) a binding of a license to the system on which the computer program is running, and (c) a validity that the right has not expired.

8.1 According to D1, the request for a license "returns a grant or denial status" (col. 21, lines 46-49). In the board's view this constitutes a notification as claimed.

8.2 D1 discloses that lm_request_allocation decides about the returned grant or denial status "based on the existence of an appropriate product use authorization" (col. 21, lines 49-50), which is defined (col. 7, lines 3-40) to relate *inter alia* to the used CPU (*i.e.* the system; col. 7, esp. lines 15-18; col. 16, lines 20-27; see also col. 1, lines 59-64), the product identifier (*i.e.* a "product key"; col. 24, see lines 28-32), and the validity of the requested right (see col. 12, lines 14-24).

8.3 Accordingly, the features added to the claim 1 of the auxiliary request are all known from D1 and thus insufficient to establish an inventive step over D1.

9. As there is no allowable request, the appeal must be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
B. Atienza Vivancos	D. H. Rees