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

Datasheet for the decision of 19 December 2012

Case Number:	T 1894/08 - 3.5.06
Application Number:	03793413.0
Publication Number:	1540472
IPC:	G06F 9/45, G06F 9/44, G06F 9/46
Language of the proceedings:	EN

Title of invention:
Wireless device operating system (OS) Application Programmer's Interface API

Applicant:
INTERDIGITAL TECHNOLOGY CORPORATION

Headword:
Operating system abstraction/INTERDIGITAL

Relevant legal provisions (1973):
EPC Art. 83, 84, 54(1), 54(2)

Keyword:
"Clarity - both requests (no)"
"Sufficiency of disclosure - both requests (no)"
"Novelty - main request (no)"

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

Case Number: T 1894/08 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06
of 19 December 2012

Appellant: (Applicant)

INTERDIGITAL TECHNOLOGY CORPORATION
Suite 527
300 Delaware Avenue
Wilmington, DE 19801 (US)

Representative:

Pierrou, Mattias
Awapatent AB
P.O. Box 45 086
SE-104 30 Stockholm (SE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 11 July 2008 refusing European patent application No. 03793413.0 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman: D. H. Rees
Members: M. Müller
C. Heath

Summary of Facts and Submissions

I. The appeal lies against the decision of the examining division, with written reasons dispatched on 11 July 2008, to refuse the European patent application no. 03793413.0 for lack of clarity, Article 84 EPC 1973, or an inventive step, Article 56 EPC 1973, over the document

D2: "Intelligent I/O (I_2O) Architecture Specification, Draft Revision 1.5", I_2O Special Interest Group, March 1997.

II. Notice of appeal along with a statement of the grounds of appeal was filed on 11 September 2008, the appeal fee being paid on the same day. The appellant requested that the decision under appeal be set aside and that a patent be granted based on claims 1-11 filed on 12 July 2006. On 30 September, the appellant filed a further set of claims 1-24 as an auxiliary request on the basis of which a patent should be granted in case the board considered the main request not to be allowable.

III. Independent claims 1 and 8 according to the main request read as follows:

"1. A method for targeting a software model to a plurality of different operating systems, the method comprising:

providing a software model;

providing an operating environment (112), the operating environment being common to said plurality of operating systems;

providing a porting layer (110), the porting layer porting the software model to said provided operating environment by converting software model elements into predefined operating environment constructs; and

providing a plurality of operating system abstraction layers (122), each abstraction layer being designed to interface the operating environment to at least one targeted operating system, wherein clients of the targeted operating system are given access to the operating environment and the ported software model.

8. A wireless communication device comprising:

at least one system processor (100) and at least one communication processor (102);

a communication module (104) to facilitate communication between each system and communication processor; and

a shared memory (106) associated with the communication module;

each system processor and communication processor having an associated operating system (124, 126), the operating system performing code generated from a software model, the device being arranged to port a software model to an operating environment (112) common to the operating systems by converting software model elements into predefined operating environment constructs, the device further being arranged with an operating system abstraction layer (114) for interfacing the operating environment to each associated operating system, wherein clients of the operating systems are given access to the operating environment and the ported software model"

IV. Claims 1 and 10 according to the auxiliary request read as follows:

"1. A method for exporting a specification and description language, SDL, software model to different operating systems, the method comprising:

providing an SDL software model;

providing an SDL porting layer, the SDL porting layer converting the SDL software model to an operating environment wherein the operating environment is common to all the different operating systems; and

providing a plurality of operating system abstraction layers, each abstraction layer designed to abstract the operating environment to at least one targeted operating system.

10. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system, the operating system performing code generated from an SDL software model, the SDL software model being ported to an operating environment wherein the operating environment is the result of an SDL porting layer converting an SDL software model to the operating environment, providing an operating environment, the operating environment common to all different operating system, an operating system abstraction layer abstracts the operating environment to each associated operating system."

V. In an annex to summons to oral proceedings the board expressed its preliminary opinion according to which the claims according to both requests lacked clarity, Article 84 EPC 1973. The board also raised objections under Article 123(2) EPC and under Articles 54(1,2) and 56 EPC 1973, in view of D2 but also, alternatively, in view of the prior art acknowledged in the application itself.

VI. The appellant did not respond to this communication. During a telephone conversation with the board's registrar, the representative expressed his intention not to attend the oral proceedings.

VII. Oral proceedings took place as scheduled and in the absence of the appellant. At the of the oral proceedings, the chairman announced the decision of the board.

Reasons for the Decision

The appellant's absence at the oral proceedings

1. The appellant was duly summoned, but did not attend the oral proceedings.

1.1 According to Article 15(3) RPBA, the board is not obliged to delay any step in the proceedings, including its decision, by reason only of the absence at the oral proceedings of any party duly summoned who may then be treated as relying only on its written case.

1.2 Since the appellant chose not to respond to the board's communication, the board has no reason to deviate from its preliminary opinion set out therein. Of the objections raised in the annex to oral proceedings, only those are reproduced herein which the board deems to be the most pertinent ones. The following reasons are substantially based on the board's preliminary opinion as set out in the annex to the summons to oral proceedings.

The invention

2. The application is concerned with the development of software for wireless devices in view of the variety of the operating systems (OS) they run and their diverse and possibly heterogeneous hardware structures (pars. 8-9). A central element of the proposed and claimed solution is an "operating environment common to all operating systems" (par. 11) which sits between the "software model" and the "clients" (figs. 1 and 11a) on the one hand and the operating systems on the other. The operating environment contains, in particular, an operating system API (fig. 1, nos. 112, 120) to which a given software model is "ported" (no. 110). Any concrete operating system in turn provides an abstraction layer (no. 122) which interfaces to the OS API.

Main Request

Articles 84 EPC 1973

3. The decision under appeal (point 3.1) considered the term "software model" to be unclear for not having a "generally accepted and well-defined meaning [in] software programming".

3.1 The appellant disagreed with this finding, arguing that the term "software model" is widely used in the context of the well-known software modelling languages UML and SysML and the related "modelling tool" Telelogic Tau.

3.2 The board first notes that, as already pointed out by the examining division, a term may well be widely used in the art - as the board has no doubt for the term "software model" - without having a well-defined *technical* meaning.

3.3 The board also notes that none of UML, SysML or Telelogic Tau is mentioned in any of the claims, so that an established meaning the term "software model" might have in the context of UML, SysML or Telelogic Tau is not implied by the claim language.

3.4 While modelling is indeed a central aspect of software development, the term "model" may, in the board's understanding, refer to a variety of things during the software development process, such as the modelling of the application domain, the choice of data structures - *i.e.* a data model - or the choice of a concrete programming language - defining, *inter alia*, a computational model. Moreover, the term "software model" appears to be ambiguous between a "model for software" or a "model expressed in software": UML for instance, is a graphical modelling language to enable and simplify communication between developers from requirements engineering to implementation: Although a UML model describes what is eventually to become a software product - and thus is a model *for* software - it is not, as a whole, meant or suitable for automatic execution, *i.e.* not a program and in this sense not "software" itself.

3.5 Beyond this ambiguity, the board notes that the claims do not specify explicitly any of the "software model elements" or any detail as to the object of the software model, *i.e.* neither what is modelled nor how.

3.6 The board therefore agrees with the decision under appeal that the term "software model" is unclear, Article 84 EPC 1973.

4. The claims according to the main request specify that the software model is "ported" to the "operating environment" by "converting" its "elements into ... operating environment constructs".

4.1 The board considers this feature to lack clarity for two reasons.

4.2 The conventional and established meaning of the term "porting" in the field of software engineering is adapting software for a different OS or hardware platform. In contrast, the "porting layer" according to the invention is meant to hide different platforms by way of abstraction, expressly so that software need *not* be adapted to individual OS or hardware structures (see description, par. 55). The board considers this conflict between the established and the intended meaning of "porting" to render the claims unclear, Article 84 EPC 1973.

4.3 The board also considers unclear the claimed phrase which explains the "porting layer" as "converting software model elements into ... operating environment constructs", because it leaves open the nature of the conversion, Article 84 EPC 1973.

5. Another central term of the claimed invention is the "operating environment". Also this term does not, in the board's judgment, have a well-defined meaning in the art. The operating environment is specified by the claims only insofar as it should have "predefined" but otherwise undefined "operating environment constructs" and is interface[d] ... to [a] targeted operating system" via what is called an "operating system abstraction layer". The board considers this definition to be insufficiently clear, Article 84 EPC 1973.

5.1 The description specifies that the operating environment comprises in particular an OS API (see *e.g.* pars. 43, 54, fig. 1) which defines the relevant operating system constructs into which the "porting layer" is to convert the software model elements. The board considers the OS API as an essential feature of the invention whose omission from the claims amounts to a deficiency under Article 84 EPC 1973.

5.2 The claims specify the "operating environment" as being "common to [a] plurality of operating systems" and interfacing to "operating abstraction layers". Thereby, the claims specify the operating environment as being independent of any particular operating system, *i.e.* "OS independent" (see par. 43). However, the claims neither specify the range of operating systems covered nor to what extent an "OS independence" is achieved. In general, an "OS abstraction layer" could be meant to hide superficial differences between two versions of the same operating system, larger differences between two members of the same family of operating systems such as BSD and Linux, or substantial differences between operating systems such as Windows and Unix. The board considers this vagueness as a lack of clarity, Article 84 EPC 1973.

5.3 Claim 8 of the main request is more concrete than claim 1 insofar as it specifies a concrete device with two processors which run two different operating systems both of which are interfaced with the same "operating environment". Claim 8 leaves open however, in what sense, if at all, and how the operating system achieves the express goal of the description (par. 9) to "hide this boundary" between processors and is thus also insufficient to establish a clear meaning of the term "operating environment".

Article 83 EPC 1973

6. As a consequence of the vague claim language, the board considers that the description does not disclose the concept of operating system abstraction in the claimed generality in a manner sufficiently clear and complete for the skilled person to carry it out, Article 83 EPC 1973: For example, the description does not disclose how operating environment, operating abstraction layers and porting layers had to be set up to cover, say, a major part of the operating systems so different as Windows and Unix.

Articles 54 (1,2) EPC 1973

7. The application (see esp. pars. 8 and 9) discloses that known wireless devices are equipped with several different operating systems and hardware structures and addresses the problem of simplifying the development of software for a range of such devices by hiding these

differences from the software developer through suitable software abstraction.

7.1 The board leaves open whether the simplification of software development by providing suitable operating system abstractions in general constitutes a technical effect which must be taken into account for assessing inventive step or is not, rather, a non-technical aspect of computer programming.

7.2 At any rate, the board considers it to be commonplace that software is developed for several different operating systems (say Windows, Linux and Mac OS; or for different Unix variants such as BSD or Solaris) and thus that it is a known desirable to make such cross-OS development as simple as possible.

7.3 The programming language Java is a prominent example for this fact: Java, released in 1996, was designed to simplify the development of software across platforms and operating systems ("write once, run everywhere"). Java is distributed with a set of standard class libraries which implement the Java application interface, *i.e.* an "operating environment" in the terms of the claims. The board therefore considers that, on a broad interpretation of the unclear terms discussed above (points 3-5), any mapping of a "software model" into Java qualifies as the claimed "porting" of the software model into "operating environment constructs". In order to execute Java programs, any computing platform must further provide a runtime environment which hides the underlying operating system and thus acts as an "operating system abstraction" in the terms of the claims.

7.4 Further well-known examples are toolkits for the development of uniform graphical user interfaces across platforms, such as Motif (for the Unix family), available since the 1980s, or Qt (available for Windows, Windows CE included, as well as for Linux and Mac OS), available since 1999. Such toolkits offer an API - *i.e.* an "operating environment" as claimed - which is common to different operating systems and which the client applications access for the definition of their GUIs. Thus any mapping of a "software model" to make use of a GUI toolkit reads on "converting software model elements into predefined operating environment constructs" and thus on a "porting layer". On an individual operating system, the installation of such a toolkit provides an "operating system abstraction layer" interfacing to the operating system itself.

7.5 The board is therefore unable to determine any clear difference between claim 1 of the main request and the conventional use of the Java API or a GUI toolkit in software development. The appellant, in not responding to the board's preliminary opinion, did not address this argument and, in particular, did not request that the board's understanding of the Java API and the GUI toolkits be further substantiated by written evidence.

7.6 The board concludes that claim 1 of the main request lacks novelty, Article 54(1,2) EPC 1973, over common knowledge alone.

Auxiliary Request

8. The claims of the auxiliary request are identical to the claims as originally filed except that the terms

"software model" and "porting layer" were limited to "SDL software model" and "SDL porting layer", respectively, SDL standing for "specification and description language".

8.1 The board leaves aside its doubts as to whether the terms "SDL software model" and "SDL porting layer" are, in themselves, clear terms and thus sufficient to overcome the clarity objection against the terms "software model" and "porting layer".

8.2 However, the board considers that the added reference to SDL is insufficient to overcome at least some of the objections raised against the main request. In particular, claim 1 of the auxiliary request also fails to specify the function of the SDL porting layer and the operating environment (see points 4.3 and 5 above) and is thus unclear as argued above, Article 84 EPC 1973.

8.3 Even though this objection was not specifically raised against the claims of the auxiliary request in the board's preliminary opinion, this does not violate the appellant's right to be heard, Article 113(1): The board considers that it was apparent for the appellant that the differences between the claims of the main and the auxiliary requests are insufficient to overcome these objections and therefore predictable that these would be raised against the auxiliary request, too.

Summary

9. There being no allowable request, the appeal has to be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

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