

**Internal distribution code:**

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

**Datasheet for the decision  
of 13 September 2011**

**Case Number:** T 1991/07 - 3.5.06

**Application Number:** 04250554.5

**Publication Number:** 1560117

**IPC:** G06F 9/54

**Language of the proceedings:** EN

**Title of invention:**

System and method for publishing and accessing application apis on a generic terminal

**Applicant:**

Research In Motion Limited

**Headword:**

Publishing and accessing application APIs/RIM

**Relevant legal provisions (EPC 1973):**

EPC Art. 54

**Keyword:**

"Novelty - no"



Case Number: T 1991/07 - 3.5.06

**D E C I S I O N**  
of the Technical Board of Appeal 3.5.06  
of 13 September 2011

**Appellant:**

RESEARCH IN MOTION LIMITED  
295 Phillip Street  
Waterloo  
Ontario N2L 3W8 (CA)

**Representative:**

Hibbert, Juliet Jane Grace  
Kilburn & Strode LLP  
20 Red Lion Street  
London WC1R 4PJ (GB)

**Decision under appeal:**

Decision of the Examining Division of the  
European Patent Office posted 11 July 2007  
refusing European patent application  
No. 04250554.5 pursuant to Article 97(1) EPC  
1973.

**Composition of the Board:**

**Chairman:** W. Sekretaruk  
**Members:** M. Müller  
A. Teale

## Summary of Facts and Submissions

- I. The appeal lies against the decision of the examining division, dispatched with letter dated 11 July 2007, to refuse the European patent application 04 250 554.5.
- II. A notice of appeal was filed on 19 September 2007, and the appeal fee was received on the same day. A statement of grounds of appeal was submitted on 21 November 2007.
- III. The appellant requests that the decision under appeal be set aside and that a patent be granted based on a set of amended claims 1-30 as filed with the statement of grounds of appeal.
- IV. Claim 1 reads as follows:

"A method for providing dynamic interaction between a pair of application programs via an interface module (312, 500) of a terminal, the pair of applications including a requestor application (400) desiring access to a target application (107), the method comprising the steps of:

registering access information of the target application (107) with the interface module (312, 500), the access information including: an access handler configured to transform data from a predefined structured language to an access format expected by the target application (107); and published access information to be made available in a data structure (210, 212) for retrieval by the interface module (312, 500);

receiving an access request at the interface module (312, 500) from the requestor application (400), the access request including request content corresponding to the published access information of the target application (107);

obtaining the access handler (122) by the interface module (312, 500) using the request content to search the data structure (210, 212), the access handler (122) configured to transform a data portion of the request content from a predefined structured language to an access format expected by the target application (107); and employing a predefined application program interface (API) (124) by the access handler (122) to access the target application (107) and satisfy the access request of the requestor application (400), the API 124 being stored at the interface module (312, 500)."

V. With a summons to oral proceedings the board expressed its preliminary opinion. The board *inter alia* made reference to the following documents

- D1: Nakada H. *et al.*, "GridRPC: A Remote Procedure Call API for Grid Computing", Grid Working Drafts Informational, Global Grid Forum, pages 1 to 10, July 2002
- D4: Nakada H. *et al.*, "Design and implementations of Ninf: towards a global computing infrastructure", Future Generation Computer Systems, Elsevier Science Publishers, vol. 15, no. 5-6, pages 649 to 658, October 1999

and argued that claim 1 appeared to lack novelty over D1.

VI. On 5 September 2011 the appellant informed the board of its intention not to attend the oral proceedings. Oral proceedings were held as scheduled on 13 September 2011 in the absence of the appellant. At the end of the oral proceedings, the chairman announced the decision of the board.

### **Reasons for the Decision**

1. The appeal is admissible because it complies with the EPC admissibility requirements (see points I and II).

#### *Appellant's absence at Oral Proceedings*

2. The duly summoned appellant did not attend the oral proceedings. In accordance with Article 15(3) RPBA the board relied for its decision only on the appellant's written submissions. The board was in a position to decide at the conclusion of the oral proceedings, since the case was ready for decision (Article 15(5,6) RPBA), and the voluntary absence of the appellant was not a reason for delaying the decision (Article 15(3) RPBA).

3. The following reasons are based on the board's preliminary opinion annexed to the summons to oral proceedings.

#### *The invention*

4. The application is concerned with the large number of mobile devices and services in use today and the difficulty of developing new applications across platforms. Moreover, the application considers undesirable that software applications are limited to built-in knowledge

about available services and cannot easily be upgraded or extended to use newly installed services (cf. pages 1-2 of the description).

- 4.1 As a solution to this problem, the invention proposes a directory service which allows a new application to register its application programming interface (API) and a "requestor application" to request and obtain access to some suitable such "target application".
  
- 4.2 More specifically, the invention according to claim 1 specifies an interface module which supports the interaction between a requestor application and a target application as follows: The target application must register with the interface module details about its API (as so-called "published access information") and an access handler which transforms data from a neutral format (in a "predefined structured language" such as XML, see original claim 10) into the native format of ("expected by") the target application. When the interface module receives a request by a requestor application specifying the desired service in terms of content corresponding to the published access information, it will search for a suitable target application and, if found, establish the interaction between the requestor and target applications via the registered access handler.

#### *The Prior Art*

5. Ninf is an infrastructure for enabling the dynamic interaction between different systems over a network. Ninf itself is disclosed in D4, whereas document D1

- discloses a "re-implementation" of Ninf called Ninf-G (see section 4.2.1, first sentence).
- 5.1 Originally, Ninf was devised as a computing platform for scientific computing (cf. D4, page 650, left, 2nd paragraph) and the Ninf communication overhead is judged acceptable especially for "typical scientific applications" which are "both compute and data intensive" (see D4, page 651, right column, last paragraph, lines 1 to 8).
  - 5.2 Throughout examination, D1 was used as the starting point to assess novelty and inventive step. The board agrees that this is a suitable choice.
  - 5.3 The appellant argues that the skilled man "would not look to a Ninf-based implementation for anything other than complex implementations or else the cost of overhead would be prohibitive" and that D1 should not be used as starting point for assessing patentability.
  - 5.4 However the present claims do not specify any detail about the kind or complexity of the interacting applications in question, be it in absolute terms or relative to the communication overhead. The board is also unable to see where the description discloses any such detail, and the appellant did not indicate any in response to the board's preliminary opinion either.
  - 5.5 The board therefore sees no compelling reason to dismiss D1 as a starting point for assessing patentability.

*Ninf, a network-wide computing infrastructure*

6. According to D1, for a host computer to make available its services to remote applications "on the grid", it must run a Ninf-G server process which handles all remote requests. In the wording of claim 1, the Ninf-G server acts as the claimed "interface module" of the host computer, the "terminal", to deal with a "requestor application desiring access to a target application".
  - 6.1 A library which is to be made available through Ninf-G must be "gridified" first (cf. D1, page 6, left column, "Server side IDL"). The library provider describes and publishes the library interface (*loc. cit.*), and uses the IDL compiler to produce "remote library executables" (cf. D1, figure 2, and section 4.2.2, no. 3) linked to the local library functions. Both the interface description and the remote executables are registered with the Ninf-G server (see D1, *loc. cit.*; page 6, left column, penultimate paragraph on MDS; and section 4.2.2, point 1).
  - 6.2 When requesting access to a library, clients will provide information (the "library signature as a key") corresponding to the published access information, which the MDS (the "Monitoring and Discovering Service") uses as a key to search the published interface information (D1, section 4.2.2, point 1) and to retrieve the remote executable. The remote executable then establishes the communication with the client on the one hand and the local library on the other and thus "satisf[ies] the access request of the requestor application" (cf. e.g. D1, section 4.2.2, point 4).



- 6.3 The Ninf-G interface at the client side will marshal the arguments to the desired function call (see D1, page 7, penultimate paragraph, lines 11 to 18). The corresponding unmarshalling operation must be done on the Ninf-G server side before the actual library call is executed. Unmarshalling is the transformation of a platform independent transmission format (defined in a "predefined structured language") into the native format expected by the target application. While D1 does not mention unmarshalling explicitly, the board observes that only the remote executable knows the expected target format and thus considers it to be implicit that the remote executable performs the unmarshalling or at least calls a suitable unmarshalling routine. Either way, the board considers that the remote executable according to D1 qualifies as the registered access handler as claimed.
- 6.4 D1 talks about a library being made available whereas claim 1 refers to the API of an application. The board considers that, say, a scientific computation package can well be dubbed an "application" and that the library functions defining how the package is to be used correspond to its application programming interface. In this sense, the board is of the opinion that the language of claim 1 does not establish a technical difference over Ninf-G as disclosed in D1.
- 6.5 For this reason, the board concludes that the subject matter of claim 1 lacks novelty over D1 alone, in violation of Article 52(1) EPC and Article 54(1,2) EPC 1973.

**Order**

**For these reasons it is decided that:**

The appeal is dismissed.

The Registrar:

The Chairman:

B. Atienza Vivancos

W. Sekretaruk