

Internal distribution code:

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

**Datasheet for the decision
of 31 July 2012**

Case Number: T 0417/09 - 3.5.06

Application Number: 05103037.7

Publication Number: 1712987

IPC: G06F 9/44

Language of the proceedings: EN

Title of invention:

System and method for unified visualization of two-tiered applications

Applicant:

Research In Motion Limited

Headword:

Mapping a two-tiered application/RIM

Relevant legal provisions (EPC 1973):

EPC Art. 84

Keyword:

"Clarity - no"



Case Number: T 0417/09 - 3.5.06

D E C I S I O N
of the Technical Board of Appeal 3.5.06
of 31 July 2012

Appellant: Research In Motion Limited
(Applicant) 295 Phillip Street
Waterloo, ON N2L 3W8 (CA)

Representative: Patel, Binesh
Barker Brettell LLP
100 Hagley Road
Edgbaston
Birmingham B16 8QQ (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 11 July 2008
refusing European patent application
No. 05103037.7 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 to refuse the European patent application no. 05103037.7 for lack of clarity, Article 84 EPC 1973.

II. The notice of appeal was filed on 8 September 2008, the appeal fee being paid on the same day. On 3 November 2008 the appellant filed a statement of grounds of appeal and requested that the decision be set aside and that a patent be granted based on the documents as subject to the decision under appeal, in particular claims 1-12 as filed on 5 June 2008.

III. Claim 1 according to this request reads as follows

"A method for programming a two-tiered application, comprising:

providing a unifying data structure (1204) defining a unifying data model for programming application components (400, 402, 404, 406) for each tier of the two-tiered application (105), the unifying data structure comprising mapping information for mapping respective application component logic (1206, 1208) between each tier of the two-tiered application;

providing an application development tool (106) having a graphical user interface 'GUI' (202, 1202) for providing a visualization of the two-tiered application as a single logical application in accordance with the unifying data model with the mapping information abstracted away;

receiving user input via the GUI to program the single logical application; and

transparently programming application components of the two-tiered application in accordance with the single logical application and the unifying data model by mapping the user input for programming the single logical application to program a respective tier of the two-tiered application (105) using the mapping information."

Claim 7 defines a system in terms of features which closely correspond to those of claim 1.

- IV. With a summons to oral proceedings, the board indicated its preliminary opinion that the decision under appeal would have to be confirmed for lack of clarity.
- V. In response, the appellant informed the board of its intention not to attend the scheduled oral proceedings. No arguments or amendments were filed.
- VI. The oral proceedings were held as scheduled in the absence of the appellant. At the end of the oral proceedings, the chairman announced the board's decision.

Reasons for the Decisions

Appellant's absence at oral proceedings

1. 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).

2. The reasons for this decision are based on the preliminary opinion communicated to the appellant with the summons to oral proceedings.

The Invention

3. The invention is concerned with method and corresponding system for programming a two-tiered - *i.e.* client-server - application in a way which resembles the development of a single-tier application. The claimed invention according to independent claims 1 and 7 comprises three major parts.
 - a) A "unifying data structure" comprising mapping information which associates each "application component logic" to its "tier".
 - b) A graphical user interface which visualizes the two-tiered application "as a single logical application" with the mapping information "abstracted away".
 - c) A development tool which enables "transparent programming application components" of the two-tiered application by mapping user input meant to "program the single logical application" to the respectively intended tier as indicated by the unified data structure.

4. The decision under appeal (point 3.1) finds the claimed invention to be unclear in particular because the last step of claim 1 (essentially part c) is "defined as a result to be achieved rather than in terms of technical features which achieve said effect". The appellant challenges this finding and alleges that the claimed invention is clearly distinguished over D1 and that the details missing from the claim language, in particular as regards implementing said last step, would be "immediately apparent to a person skilled in the art" (see grounds of appeal, p. 3, 3rd par.).

5. The board agrees with the decision under appeal that the subject matter of all claims is unclear, Article 84 EPC 1973. However, the board's concerns in this respect go beyond those of the examining division. Specifically, the board deems some of the central terminology of the claims to be unclear as is detailed below.

Origin of the mapping information

6. The unifying data structure is specified as containing "mapping information" for mapping each component to its respective tier. The claims are silent however as to how and when the mapping information is provided.
 - 6.1. The claims do not exclude the possibility that the mapping information is automatically derived from the application. In the board's view, the automated distribution of application components across tiers is a rather non-trivial task solutions to which do not, in general, belong to the common knowledge in the art. However, neither the claims nor the description specify how

such automation should be performed. The description is thus insufficient to support this interpretation.

6.2. Alternatively, mapping information may not be derived automatically but "provided up front" (cf. minutes of oral proceedings before the examining division, point 13), typically by the programmers themselves. For instance, programmers may be required to indicate the intended tier for each newly created component. Under these circumstances the board considers that the programmer will be and remain aware about the two tiers of the application under development, whether the mapping information is "visualized" or not, which puts into doubt whether and in what sense the programmer can be said to program the application "transparently" as a "single logical application".

6.3. In either case, the claims are deficient under Article 84 EPC 1973.

Single logical application

7. According to the claims, the application being developed is visualized as a "single logical application" with the "mapping information abstracted away".

7.1. The board considers the term "single logical application" to be unclear: Whether an application can "logically" be viewed as a "single" one is a matter of perspective and does not necessarily depend on whether or not it is distributed or how. For example, the entire world wide web or any web-based application comprising web servers and web clients (*i.e.* browsers) can justifiably be considered as "single logical applications".

- 7.2. It may be possible to interpret the term "single logical application" as an application *defined by* the fact that the "mapping information [is] abstracted away". However, the term "abstracted away" is unclear in itself and so is, as a consequence, the content of the claimed visualization.
- 7.3. In this context, it is noted that the board is not convinced that, as alleged by the appellant (cf. grounds of appeal, p. 3, 1st par.), figure 13a of the present application illustrates the "abstracting away of information" (cf. grounds of appeal, p. 3, 1st par.). While figure 13a itself indeed does not visualize "mapping information", figure 13a also does not depict the part of the GUI needed for "receiving user input ... to program the single logical application". Whether other parts of the GUI, say the contents of the "Components" subfolder or any actual editing tool, contain mapping information or not cannot be judged from that figure.
- 7.4. According to the claims, the user is to "program" the "single logical application [via the GUI]". In the context of the claims, the board takes this to mean that the programmer is to input program code using an interface which suppresses location information. This interpretation was indicated in the summons to oral proceedings and was not commented upon by the appellant. That location information is not displayed, however, does not exclude that the user is - at least at times - aware of the mapping information and takes it into account during programming.
- 7.5. In this context the board disagrees with the appellant's suggestion that the mapping information is "(from the

user's point of view) unnecessary" because "in the context of a multi-tier application the ultimate location of a module at run time is not an issue that is of concern to a developer" (cf. grounds of appeal, p. 2, last par.). While apparently not all decisions to be made during programming depend on the distributed nature of an application, some manifestly do: As soon as the implementation of a function involves several machines and thus requires communication between both, the programming details may well affect the overall performance of the application. In view of network latency the programmer may for instance want to minimize the amount of communication required so as to improve responsiveness of the application. In view of possible network failure, the programming may want to provide additional failure handling as to improve reliability.

- 7.6. According to the preferred interpretation of the claims (see point 6.2), the mapping information must be provided "up front" to the unifying data structure. Although mapping information and program code might, within the scope of the claims, be input separately and through different interfaces, it is unclear why the input of mapping information cannot, in a broad sense, be viewed as a part of the programming. This renders further unclear what the claims mean when referring to the "transparently programming ... [a] single logical application".

Unit of distribution and mapping information

8. The claims specify "mapping respective application component logic ... between each tier of the two-tiered application" but do not define the nature or size of the individual components to be mapped.

- 8.1. It is noted that not only pieces of program code can be "mapped" to run on different tiers, but also pieces of data structures can be "mapped" to be stored on different tiers. For example, the individual pages of a hypertext document may be hosted on different machines. But even if the term "application component *logic*" is interpreted to refer to pieces of program code, the units of distribution can be functions or procedures, larger ones such as objects, modules or entire libraries, or smaller ones such as individual program statements.
- 8.2. The board does not consider this ambiguity as a clarity problem in itself. However, the unit of distribution has an impact on the extent to which a programmer should or must be aware about the distinction between a single-tier or a two-tiered application and thus has a bearing on the question what "transparent programming" may mean and what it takes to enable it.
- 8.3. On the one hand, the distribution of small units may incur a larger communication overhead than the distribution of large units. On the other hand, larger components may comprise functionality to be executed on different machines: For example, a function developed as part of a "single logical application" may access a database at the server and display the retrieved results at the client. Evidently, such a function cannot be mapped as a whole to only one of the tiers; part of it must be mapped to the client and some to the server. It is unclear from the claims (and indeed the description) whether the "unifying data structure" is even able to express such "split" mapping and how, and how such mapping information would be used to "program a respective tier of the two-tiered application".

8.4. As a consequence, the board deems the nature of the mapping information in itself to be unclear, Article 84 EPC 1973.

Summary

9. The board concludes that the independent claims 1 and 7 - as well as all dependent claims - are unclear as regards their express goal ("transparent programming"), the means to achieve it ("mapping information", "abstracted away", "single logical application") and whether it is actually achieved, and are thus deficient under Article 84 EPC 1973.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar

The Chairman:

B. Atienza Vivancos

D. H. Rees