

**Internal distribution code:**

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

**Datasheet for the decision  
of 3 March 2016**

**Case Number:** T 2230/13 - 3.5.06

**Application Number:** 06779109.5

**Publication Number:** 1938183

**IPC:** G06F9/44, G06F9/445

**Language of the proceedings:** EN

**Title of invention:**

METHODS FOR USER INTERFACE GENERATION AND APPLICATION  
MODIFICATION

**Applicant:**

Corizon Limited

**Headword:**

Modifying user interface elements/CORIZON

**Relevant legal provisions:**

EPC 1973 Art. 84

**Keyword:**

Claims - clarity (no)

**Decisions cited:**

**Catchword:**



**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 2230/13 - 3.5.06

**D E C I S I O N**  
**of Technical Board of Appeal 3.5.06**  
**of 3 March 2016**

**Appellant:** Corizon Limited  
(Applicant) 119-121 Middlesex Street  
London  
E1 7JF (GB)

**Representative:** Kenrick, Mark Lloyd  
Marks & Clerk LLP  
1 New York Street  
Manchester, M1 4HD (GB)

**Decision under appeal:** **Decision of the Examining Division of the  
European Patent Office posted on 5 June 2013  
refusing European patent application No.  
06779109.5 pursuant to Article 97(2) EPC.**

**Composition of the Board:**

**Chairman** W. Sekretaruk  
**Members:** M. Müller  
S. Krischer

## **Summary of Facts and Submissions**

- I. The appeal lies against the decision of the examining division, with reasons dispatched on 5 June 2013, to refuse European patent application No. 06 779 109.5 because none of the pending requests conformed with Article 123(2) EPC. In a section entitled "Obiter dicta", the decision contains arguments as to why claim 1 of the then main request lacked clarity and, referring to objections raised in the summons to oral proceedings before the examining division, why claim 1 of the main request lacked an inventive step. The decision refers to several documents, but does not rely on any of them for its reasons.
- II. Notice of appeal was filed on 14 August 2013, the appeal fee being paid the following day. A statement of grounds of appeal was received on 14 October 2013, with which the appellant requested that the decision be set aside and that a patent be granted based on claims 1-9 according to a main or a first auxiliary request as filed with the statement of grounds of appeal, the main request corresponding to the main request as refused by the examining division.
- III. In an annex to the summons to oral proceedings, the board informed the appellant of its preliminary opinion, according to which the claims of both requests lacked clarity, Article 84 EPC 1973, and inventive step, Article 56 EPC 1973, because their subject-matter did not make a technical contribution to the art. An objection under Article 123(2) EPC was also raised.
- IV. In response to the summons, the appellant maintained its previous requests, but filed new claims 1-9 according to a further, second auxiliary request.

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

"A method of modifying at least one of a plurality of user interface elements of a user interface, the method comprising:

receiving at one of a plurality of communicating objects (106, 107, 108), from program code running to provide said user interface, a notification of a manipulation of the user interface, the notification being provided by a call by said program code to a predetermined function, wherein each of said communicating objects represents one of said plurality of user interface elements and wherein said predetermined function is called when a function associated with a predetermined user interface manipulation is called; and

processing said notification at said one of said objects (106, 107, 108), said processing causing modification of said at least one user interface element."

Claim 1 according to the first auxiliary request is identical to claim 1 of the main request, except for the preamble, which reads as follows:

"A method of modifying behaviour of a user interfaces by modifying at least one of a plurality of user interface elements of a user interface, the method comprising: ..."

Claim 1 according to the second auxiliary request is identical to claim 1 according the main request, except for the fact that the three occurrences of user interface elements are now specified to be "graphical" user interface elements.

All the requests contain an independent apparatus claim 9 corresponding closely to the respective method claim 1.

- VI. Oral proceedings were held as scheduled on 3 March 2016. No one was present for the appellant, although this had not been indicated beforehand to the board. At the end of the proceedings, the chairman announced the board's decision.

### **Reasons for the Decision**

#### *Decision in the appellant's absence*

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. Therefore, and further in accordance with Article 15(3) RPBA, the board treats the appellant as relying only on its written case.

#### *The invention*

2. The application is concerned with flexibility in generating and modifying user interfaces, for instance in order to provide one seamless interface for several applications (see e.g. page 1, paragraph 3 - page 2, last paragraph).
  - 2.1 As a solution, the application discloses a "method of modifying a source application" (i.e. program code) at run-time (page 3, paragraph 2; page 31, paragraph 2), which is "particularly suitable for modifying user interfaces" (paragraph 4). Furthermore, it discloses "a method of affecting the behaviour of a user interface

comprising a plurality of user interface elements"  
(page 5, paragraphs 2 and 4).

2.2 This effect is achieved by manipulating the object code of a given application (see figures 8-12 and pages 16 to page 18, paragraph 1; page 31, paragraph 2 - page 33, paragraph 3)), so that whenever the application requires the execution of a certain piece of handler code in response to a user interface event (see figure 8, no. 50: "GetMenu"), a piece of "hooking code" is also executed (figure 8, nos. 52, 53, 61, and 73). The hooking code in turn calls an "agent" (see page 13, middle paragraph; page 15, last paragraph; figure 8, rightmost box), which is mostly implemented in the interpreted language Python (see figure 8, no. 43, and figure 15), and which is "responsible for providing modifications for the source application" (page 13, paragraph 3).

2.3 It is also stated in the description - and in the claims - that the invention is based on "modelling" the user interface in terms of "a plurality of communicating objects" which are "preferably" defined using an object-oriented language (see page 5, paragraph 4).

*Article 84 EPC 1973*

3. The independent claims of the main request refer to "communicating objects",

- a) which "receiv[e] [...] from program code [...] a notification";
- b) each of which "represents one user interface element"; and
- c) the processing of which "caus[es] modification of user interface element[s]".

They further specify that

d) "program code running to provide said user interface" notifies the communicating objects.

3.1 *In the board's understanding*, a) implies that the "communicating objects" are themselves some form of program code, which, however, need not be written in an object-oriented programming language, as this choice of language is disclosed as merely preferable. In its letter of 3 February 2016 (page 2, penultimate paragraph), the appellant confirmed this interpretation. On that understanding, however, the board considers that neither the term "communicating objects" as a whole, nor its components "communicating" or "objects", are sufficient to imply any specific limitation on the program code referred to.

3.2 As regards b), the board agrees with the decision under appeal (see reasons 5.2) that this sentence is vague and ambiguous, Article 84 EPC 1973. On a broad reading, the skilled person might deduce that each communicating object "represents" one user interface element in that there is one such object per user interface element, noting however that this would have no further implication as to what the communicating objects are intended to "do" when executed. In the board's view, the claim wording also covers the interpretation that the communicating objects "implement" the user interface elements in terms of their appearance and behaviour. Alternatively, the communicating objects might be understood to implement only the behaviour of a user interface element, the appearance of which is defined elsewhere. Or, they might be understood to implement part but not all



of what constitutes a user interface element in terms of appearance and/or behaviour.

3.3 As regards c), it is noted that the claims do not define the term "modification of user interface element[s]" explicitly. Its meaning can also not be derived indirectly from the behaviour of the communicating objects because, as just argued, the claims also fail to define what execution of the "communicating object" is meant to achieve, other than by reference to said "modification". Therefore, it is unclear how the "processing of [a] notification at [the] objects" is meant to "cause" said modification, Article 84 EPC 1973.

3.4 Phrase d) implies that the "program code" is distinct from the "communicating objects", but it is left open in what way: Specifically, the claims leave open the exact difference between the program code "providing" the user interface and the communicating objects "representing" it.

4.

4.1 The appellant argues that "the term 'represents'" is "distinct from 'implements' and 'provides'", "takes its ordinary meaning and would be clear to the skilled person in the art, especially in light of the detailed description." The appellant also states that "the communicating objects neither implement appearance or behaviour, but merely represent, and cause modification of" user interface elements (see letter of 3 February 2016, page 2, paragraph 2). However, the appellant fails to define what it considers to be the "ordinary meaning" of the terms "represent" in the present context, except by illustrating it with the equally unclear term "model" (see the letter, page 1, last paragraph, to page 1, paragraph 1), to explain how specifically it considers this meaning to be different from that of the terms

"implement" and "provide" and why the skilled person would understand it not to cover the implementation of "appearance or behaviour". The appellant suggests that the "program code running to provide [the] user interface" referred to the user interface of an existing application which the communicating objects were meant to modify "at runtime" (the letter, loc. cit.) so that it would not have to be developed "from scratch" (page 2, paragraph 4). While it is accepted that method claim 1 refers to the behaviour of an "existing" user interface at runtime, the claim language implies that the "communicating objects" are "existing" and "running" in just the same sense. This explanation is thus insufficient to explain the relation and difference between the claimed program code and the communicating objects.

- 4.2 In summary, the board considers that the independent claims of the main request are unclear as regards the roles of the program code and the communicating objects in implementing the user interface (elements) and in bringing about any "modification" of them, Article 84 EPC 1973.
5. In the first auxiliary request, the independent requests are limited by the amended preamble to modifications of the "behaviour" of the user interface. The board takes this to exclude the interpretation discussed above with respect to a), i.e. that the communicating objects might affect the appearance of the user interface elements.
  - 5.1 The board observes that this may be regarded as being in conflict with the description, which focuses on the adaptation of the "look and feel", i.e. the appearance of user interfaces rather than their behaviour.

- 5.2 This observation aside, however, the board notes that the limitation to user interface behaviour is insufficient to explain the difference between the "program code running to provide [the] user interface" and the "communicating objects represent[ing] the [...] user interface elements", or how both piece of program code interact to "cause[e] modification of [the] user interface element[s]".
- 5.3 Therefore, the independent claims of the first auxiliary request also lack clarity, Article 84 EPC 1973.
6. In the second auxiliary request, the independent requests are limited to refer to graphical user interface elements.
- 6.1 The board considers that this limitation shifts the focus of the claims to the modification of the appearance of the user interface elements rather than their behaviour, in this respect being complementary to the claims of the first auxiliary request.
- 6.2 However, also this limitation to user interface appearance is insufficient to explain the difference between the "program code running to provide [the] user interface" and the "communicating objects represent[ing] the [...] user interface elements", or how both pieces of program code interact to "cause[e] modification of [the] user interface element[s]".
- 6.3 Therefore, the independent claims of the second auxiliary request also lack clarity, Article 84 EPC 1973.

**Order**

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

The appeal is dismissed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated