

Internal distribution code:

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

**Datasheet for the decision
of 12 November 2020**

Case Number: T 0436/15 - 3.5.06

Application Number: 05759370.9

Publication Number: 1902359

IPC: G06F9/44

Language of the proceedings: EN

Title of invention:

METHOD AND APPARATUS FOR USER INTERFACE MODIFICATION

Applicant:

Corizon Limited

Headword:

User Interface Modification/CORIZON

Relevant legal provisions:

EPC 1973 Art. 56

Keyword:

Inventive step (no) - all requests

Decisions cited:

Catchword:



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0436/15 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 12 November 2020

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

Representative: WP Thompson
138 Fetter Lane
London EC4A 1BT (GB)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 15 October 2014
refusing European patent application No.
05759370.9 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman M. Müller
Members: S. Krischer
B. Müller

Summary of Facts and Submissions

I. The appeal lies from the decision, dispatched with reasons on 15 October 2014, refusing European patent application No. 05 759 370.9 for lack of an inventive step over document

D1: WO 02/44977 A.

II. Notice of appeal was filed on 23 December 2014, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 25 February 2015. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the main request as considered by the examining division, filed on 25 February 2013 and re-filed with the grounds of appeal, in the alternative on the basis of claims according to auxiliary requests 1 and 2 as filed with the grounds of appeal.

III. In an annex to a summons to oral proceedings, the board introduced a further document from the International Search Report, namely

D5: US 2003/025732 A1,

and informed the appellant of its preliminary opinion that claim 1 according to all requests lacked an inventive step over D5 and, separately, over common knowledge in the art.

IV. In response to the summons, the appellant filed neither arguments nor amendments but indicated, with letter dated 5 October 2020, that neither the applicant/

appellant nor its representative would be attending the scheduled oral proceedings.

V. The oral proceedings were then cancelled.

VI. Claim 1 of the main request reads as follows:

"A method of modifying a user interface, the method comprising:

receiving user interface data (20, 21) comprising at least one function called to affect operation of said user interface, at a client (7, 9) from a first server (12, 13, 14);

receiving modification computer program code (22) at said client (7, 9); and

executing said modification computer program code (22) at said client (7, 9) to modify said user interface data (20, 21) to generate modified user interface data (20, 21), said modifying comprising modifying said at least one function."

Claim 1 of auxiliary request 1 differs from claim 1 of the main request in that the "executing" step is replaced by the following paragraph:

"... wherein said user interface computer program code comprises a call to said modification computer program code such that executing said user interface program code at said client (7, 9) causes execution of said modification computer program code (22) at said client (7, 9) to modify said user interface data (20, 21) to generate modified user interface data (20, 21), said modifying comprising modifying said at least one function before said function is called."

Claim 1 of auxiliary request 2 differs from claim 1 of auxiliary request 1 in that the "executing" step has been amended by addition of the underlined text, so that it reads as follows:

"... wherein said user interface computer program code comprises a call to said modification computer program code such that executing said user interface program code at said client (7, 9) causes execution of said modification computer program code (22) at said client (7, 9) to determine that said user interface computer program code comprises said at least one function and to modify said user interface data (20, 21) to generate modified user interface data (20, 21), said modifying comprising modifying said at least one function before said function is called."

Reasons for the Decision

The invention

1. The application relates, in terms of method and apparatus claims, to a way of enabling a client to dynamically modify a user interface provided to it by a server (see figure 2).
- 1.1 In a preferred embodiment, user interfaces are provided to the clients as HTML files and JavaScript code (see e.g. page 11, paragraph 4, page 12, paragraph 2, and figure 3).
- 1.2 The HTML code may, in particular, contain a call to a JavaScript method "modifyFunctions" which can modify functions (scripts) defined within the HTML file, for instance by producing a modified HTML file with a

modified function definition. Typically, this call is placed at the head of the HTML file so that any such modifications are made before the function is called (see page 19, paragraph 3, to page 21, paragraph 1; page 22, paragraph 3; page 26, last paragraph; page 27, paragraph 1; see also figures 7, 11 and 12).

- 1.3 According to the reference signs in claim 1 (of all requests), "user interface data" refers to the HTML and JavaScript code defining the user interface (see figure 3, items 20 and 21) and the "modification computer program code" to the "additional JavaScript" code (see figure 3, item 22, and page 13, paragraph 2). In the board's understanding, the "function called to affect operation of said user interface" refers to any function in the user interface data which has some bearing on how the user interface functions. Effectively, this seems to mean any function defined in the user interface data on the assumption that the latter only defines functions of relevance for the user interface.

Claim construction

2. Claim 1 (of all requests) contains separate steps of "receiving" the user interface data and the modification computer program code. The board notes that, according to the description, both may well be distributed (and thus "received") along with each other (see page 13, paragraph 2, first sentence).

The prior art

3. D1 discloses a JavaScript calendar application which runs in the web browser and generates HTML code on-the-fly (see page 3, lines 21-26). The necessary user

interface data, comprising HTML, JavaScript and "event data", is transmitted from or via a web server (see figure 1, item 104) to various web browsers on request (see page 4, lines 7-15 and 23-25; page 6, line 10 et seq.; and figure 1, items 106, 110, 112, 114). In response to events in use, the JavaScript code may produce new HTML code to be rendered (see page 5, lines 26-31). The HTML, JavaScript and event data may in part be provided by a separate, remote web server, e.g. for advertising purposes (page 4, lines 19-22). Clients may add their own event data and make it available to other clients via the web server (e.g. page 4, lines 25-27). Missing event data may also be requested by clients on need (page 5, line 16, and page 6, last paragraph et seq.). In response to such a request, a new event-interface description may be provided (page 7, lines 1-5). The latter will be processed by a JavaScript generator to build "corresponding JavaScript routines" (lines 5-7). Eventually, new HTML, JavaScript code (figure 1, item 220) and event data (item 214) is transmitted to and processed at the client (page 7, lines 7-8 et seq.).

4. D5 discloses method and system of providing customized graphical user interfaces (GUIs) on some device, preferably a scanner. The user interface may be defined in XML (see paragraph 16), which is provided by a server local to the device (see paragraph 17 and item 12 in figure 1) or a remote web server (paragraph 46). From this, a GUI is created at the device by constructing an XML DOM (paragraphs 35 and 36), "populating" it with application data and eventually creating HTML for display (paragraphs 17 and 38). The user interface can be changed dynamically by the user: in response to a user action, a script may

be invoked which recreates part of the user interface accordingly (paragraphs 18, 32, 33). More specifically, when the user edits a value in a user interface, a JavaScript function is invoked which selects and updates a node in the XML DOM (see paragraph 43 and 44). The updated data from the XML DOM may be saved by extracting the XML text from the local, updated XML DOM and sending it back to the remote web server (see paragraph 46).

The decision under appeal

5. The examining division started its inventive-step assessment from D1 and found the only difference to be that claim 1 required that the user interface function be modified by the receiving of "modification computer program code", whereas D1 disclosed generating and sending new interface code by the server based on queries received from the client (see the decision, point 16.1 of the reasons). This solved the problem of "how to modify the user interface disclosed in D1", to which "the obvious thing to do" would have been to send a "new version of the Javascript code". This corresponded to claim 1 of the main request; more specifically, the new Javascript code corresponded to "modification code modifying at least one function at the client when it is executed" (*loc. cit.*).

6. In a nutshell, the appellant argues that the skilled person wanting to modify the user interface of D1 would use the mechanism already disclosed in D1 (namely to have the locally available JavaScript code "emit" new HTML code) rather than come up with the invention as claimed, namely to have "modification computer program

code modif[y] a function of the user interface data"
(see the grounds of appeal, page 3, points 10 and 12).

The board's opinion

7. The board agrees with the appellant that D1 does not disclose that new JavaScript code transmitted to the client (see again D1, page 7, lines 1-14) will modify the "user interface data" also transmitted, but rather, presumably, that the latter will be replaced.
- 7.1 The board does not agree, that the skilled person would necessarily stick to an available solution but considers that he or she would, in the general desire for improvement, consider alternatives as well.
- 7.2 At the same time, D1 does not seem to disclose the need to modify any function comprised in the user interface data; at best, it would seem that D1 discloses the provision of additional user interface data to the client implementing additional functions.
8. For that reason, the board considers D5 to be a more suitable starting point for the assessment of inventive step.
- 8.1 D5 discloses a method of modifying a user interface comprising: receiving user interface data (in the form of XML, HTML, DHTML, including scripts, i.e. program code), evidently comprising functions called to affect operation of the user interface, at a client from a server (see, specifically, D5, paragraph 46; see also paragraph 19, sentence 6), and "modification computer program code" (the script referred to in paragraph 33, sentences 2 and 3), which executes at the client and

eventually modifies the user interface data, which defines "functions" of the user interface.

8.2 Arguably, D5 only discloses the change of "values" in a user interface, not functions. However, the board considers it to be obvious to also change, if need be, functions in just the same way. Notably, the mechanism of D5 may apply unchanged, provided the modified data happens to represent a function.

8.3 Accordingly, claim 1 of the main request lacks an inventive step over D5, Article 56 EPC 1973.

9. Claim of auxiliary request 1 differs from claim 1 of the main request in that the function of the user interface data is specified as being formulated in terms of "computer program code". The board considers it to be obvious to change any part of the user interface data including the scripts of D5, i.e. computer program code.

9.1 Moreover, claim 1 now specifies that the function is modified before it is called. In this regard, the board agrees with the examining division that the problem of modifying a particular function is not any more difficult at some points in time than at others (see the decision, point 17.1 of the reasons). Furthermore, the claims do not imply any technical reason why it would make a difference when exactly the function is changed.

9.2 Therefore, it must be assumed that the point in time at which the function is changed is given, i.e. part of the problem rather than the solution. As a consequence,

claim 1 of auxiliary request 1 also lacks an inventive step over D5, Article 56 EPC 1973.

10. Claim 1 of auxiliary request 2 differs from claim 1 of auxiliary request 1 in that the modification computer program code, when executed, determines that the user interface comprises "said" function.
- 10.1 If, as according to D5, some user interface functionality is changed in response to a user action, it is reasonable to assume that the corresponding function "code" is comprised in the user interface data (or code). A corresponding check may thus not be necessary. However, checking whether a function to be changed in some "user interface data" actually exists appears to be a common prudent programming style aimed at avoiding a run-time error if, for some unforeseen reason, a function to be changed happens not to exist.
- 10.2 As a consequence, claim 1 of auxiliary request 2 also lacks an inventive step over D5, Article 56 EPC 1973.
11. This conclusion is, in addition, also based on the following considerations.
- 11.1 Claim 1 of the main request specifies a way of having a client receive "user interface data" comprising a function and modification program code and carrying out the code so as to change the function. Preferably - and as claimed in claim 1 of the auxiliary request 1 - this takes place before the function is called. Claim 1 of auxiliary request 2 specifies the detail of the modification program code according to which it is tested whether the function to be modified exists in the user interface data to be modified.

- 11.2 In brief terms, thus, claims 1 of all requests specify a way in which the client is set up to provide a user interface in which a particular function is changed before called only once. The same effect would be achieved if the user interface data were directly provided to the client in changed form.
- 11.3 The question arises which technical effect is served by sending the client user interface data and code to change it rather than the modified user interface data directly.
- 11.4 Evidently, an effect could be that the client carried out a change which could just as well be carried out at the server. From that perspective, the invention could address the balancing of computational load between the server and the client.
- 11.5 The board takes the position that it would be obvious for the skilled person to carry out a necessary change at the client or at the server depending on load considerations within his or her common general knowledge. If it so happened that the change were to be better carried out at the client side, the corresponding code would have to be "received" at the client at some point.
- 11.6 Therefore, the board considers claim 1 of any of the requests to lack an inventive step also over any system in which a server provides "user interface data" to a client, in view of common general knowledge in the art, Article 56 EPC 1973.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



L. Stridde

M. Müller

Decision electronically authenticated