

Internal distribution code:

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

**Datasheet for the decision
of 10 October 2018**

Case Number: T 1305/17 - 3.5.06

Application Number: 10012512.9

Publication Number: 2320315

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:

User interface modification/CORIZON

Relevant legal provisions:

EPC Art. 56

Keyword:

Inventive step (no)

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 1305/17 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 10 October 2018

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

Representative: Williams, Michael David
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 2 January 2017
refusing European patent application No.
10012512.9 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 is against the decision of the examining division dated 2 January 2017 to refuse European patent application No. 10 012 512 for lack of inventive step over the document

D1: Hunt G et al., "Detours: Binary Interception of Win32 Functions", Microsoft Technical Report MSR-TR-98-33, February 1999.

Further documents were cited in the decision but not relied upon in its reasons. Of these, the board will refer to the following document, which is also mentioned in the application as filed (see page 41, paragraph 2):

D5: WO 2005/062175 A2

- II. Notice of appeal was filed on 10 March 2017, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 12 May 2017. The appellant requested that the decision be set aside and a patent be granted on the basis of claims 1-10 according to the main or auxiliary request as refused and as re-filed with the grounds of appeal, the other application documents on file consisting of the description page 3 of 11 November 2011 and pages 1, 2, 4-42 as originally filed, as well as drawing sheets 1-36 as filed with letter of 14 January 2011.
- III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed invention lacked inventive step over D5 and

D1, Article 56 EPC. Clarity objections were also raised, Article 84 EPC.

IV. In response to the summons, with letter dated 3 September 2018, the appellant filed new claims 1-10 according to a second auxiliary request. With letter dated 1 October 2018, it further indicated that it would not be attending the oral proceedings, which were then cancelled.

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

"A method of affecting operation of a user interface associated with a source computer program during execution of the source computer program on a server for providing a modified user interface for a target computer program executing on a remote client, the source computer program having an associated memory space of the server, the method comprising:

determining a memory location within said memory space at which predetermined computer program code is stored;

adding source agent computer program code to said memory space of said source computer program, said source agent computer program code including modification computer program code;

replacing at least part of said predetermined computer program code with further predetermined program code comprising an instruction directing control to said source agent to execute said modification computer program code, such that functionality and/or appearance of said user interface associated with said source computer program is modified to generate the modified user interface;

providing from the source agent to a target agent operating on the remote client the modified user interface for display at the client; and

transmitting events occurring within the source computer program executing on the server from the source agent to the target agent and receiving at the source agent from the target agent events occurring within the target computer program operating on the client."

Claim 1 of auxiliary request 1 differs from claim 1 of the main request in that the specification of the "providing" and "transmitting" steps are replaced by the following text:

"... providing the modified user interface to combining agent arranged to generate a combined user interface from the modified user interface and from a user interface associated with a second source computer program; and

providing the combined user interface to a client agent for display at the client."

Claim 1 of auxiliary request 2 reads as follows (with marked changes vis-à-vis claim 1 of the main request):

"A method of affecting operation of a user interface associated with a source computer program during execution of the first and second instances of the source computer program on a server for providing a first modified user interface for a first target computer program executing on a first remote client, and a second modified user interface for a second target computer program executing on a second remote client, the first and second instances of the source

computer program having ~~an~~ first and second associated memory spaces of the server, the method comprising:

determining a first and second memory location [sic] within said first and second memory spaces at which predetermined computer program code of the first instance is stored and predetermined computer program code of the second instance is stored respectively;

adding a first source agent computer program code to said memory space of said first instance of the source computer program, said first source agent computer program code including first modification computer program code;

adding a second source agent computer program code to said memory space of said second instance of the source computer program, said second source agent computer program code including second modification computer program code;

replacing at least part of said predetermined computer program code of the first instance and second instance with further predetermined program code comprising an instruction directing control to said first source agent and second source agent respectively, to execute said first and second modification computer program code, such that functionality and/or appearance of said first and second user interfaces associated with said first and second instances of the source computer program ~~is~~ are modified to generate the first and second modified user interfaces;

providing from the first and second source agents to ~~the~~ first and second target agents operating on ~~the~~ first and second remote clients respectively, the first and second modified user interfaces for display at the first and second clients; and

transmitting events occurring within the first instance of the source computer program executing on

the server from the first source agent to the first target agent and receiving at the first source agent from the first target agent events occurring within the first target computer program operating on the first client, and transmitting events occurring within the second instances of the source computer program executing on the server from the second source agent to the second target agent and receiving at the second source agent from the second target agent events occurring within the second target computer program operating on the second client."

Reasons for the Decision

The invention

1. The application is concerned with the flexibility of generating and modifying user interfaces, for instance in order to enable customers to interact with several inhomogeneous applications via a single (combined) graphical user interface with a uniform "look and feel" (see e.g. page 1, paragraph 3 - page 2, last paragraph).
- 1.1 As a solution, a "method of modifying a source application" is described "which allows desired modifications to be specified in a high level language" and at runtime (page 3, paragraphs 2 and 3) and which is "particularly suitable for modifying user interfaces" (paragraph 4; see also page 5, penultimate paragraph).

- 1.2 The modifiability is provided in terms of an "agent" (DLL), which comprises *inter alia* an interpreter for the object-oriented scripting language Python (see in particular figure 8, no. 43, and the corresponding disclosure on page 15, last paragraph, to page 16, paragraph 1; see also figure 15). Every modifiable function calls a Python function LogCall (see figure 8, nos. 52 and 70, and page 20, paragraph 3), which carries out the modified code (see page 19, paragraph 2) and eventually returns via a function LogReturn and a function called a "trampoline" (nos. 61 and 74). To render a function modifiable in this way, code overriding may be used (see page 31, paragraph 2 - page 33, paragraph 3, and figures 26A to 26C).
- 1.3 It is disclosed that the invention can be carried out on a stand-alone computer or in a client-server environment (see page 4, paragraph 1). The claims focus on the latter scenario. There, the server runs the mentioned source application and (source) agent which provides "user interface elements" to a client side interface (such as a web browser's) via a corresponding "target agent" (see figure 4 and page 10, last paragraph). The server may also run several instances of the same source application in communication with different clients, in which case each instance will have its own "source agent" (see figure 3 and the paragraph bridging pages 9 and 10).

The prior art

2. D1 presents the library Detours, which enables the run-time modification of system functionality by what is called "binary interception" (see e.g. abstract, and paragraph bridging pages 1 and 2). For a "target

function" to be modifiable, a "trampoline function" is provided (see figure 2, left hand side). For a modification, the "first few instructions of the target function" are replaced by a "detour function", whereas (some) instructions from the target function are moved to the trampoline function, which "returns" to the remainder of target function - or to just after the target function if there is no such remainder (see page 1, right column, paragraph 2; page 2, left column, last paragraph, to right column, paragraph 2; figure 2, right hand side). The detour function can perform any user supplied function and then either return directly to the source code, and thus bypass the original target function, or call the trampoline function to carry out the target function before returning (see page 2, right column, lines 9-17). Thereby, the target function can be replaced or extended.

3. D5 discloses several applications running on different servers (see figures 1 and 2, nos. 1-3 and 12) and which are accessed as "composite applications" at various client computers (see nos. 5 and 6). D5 is concerned with the generation of corresponding "composite user interfaces" with a uniform appearance (see page 1, paragraphs 1 and 2 and the paragraph bridging pages 1 and 2; page 4, paragraph 3; and page 11, paragraph 3). It is self-evident, but also disclosed, that this architecture requires the transmissions of events from the clients to the servers and *vice versa* (see e.g. page 11, paragraph 2). D5 also discloses that the interfaces may, at least in part, be defined in HTML (see figure 2, no. 12, and the corresponding description, in particular page 12, paragraph 2).

Claim construction

4. The appellant's case turns on the provision of "agents" - especially of a "source agent" - as claimed (see the grounds of appeal, points 17, 20 and 24, and the appellant's letter of 3 September 2018). *Inter alia*, the appellant challenged the examining division's finding that D1 disclosed a "source agent", stating that "the detour function of D1 is very clearly not a source agent as that term is used in claim 1 of the present application" (point 17). It is therefore relevant for the issue at hand how that term is used in claim 1.

5. The independent claims of all requests refer to "source agents" and "target agents". It is specified that the source agent has associated "source agent computer program code" including "modification computer program code", that the source agent "provid[es]" the "modified user interface" to the corresponding target agent, and that source and target agents transmit events to each other which occur, respectively, in the source computer program and a "target computer program". Beyond that, the meaning of the terms "source agent" and "target agent" is undefined.

- 5.1 In its letter of 3 September 2018 (page 2, paragraph 1), the appellant states that "a dedicated agent executing the modification code is useful as it can run as a separate thread to the source program". Whilst it is true that this option is disclosed in the application, it is not expressly specified in the independent claims. Nor does the term "agent" - alone or as opposed to "program" - imply a particular threading scheme.

5.2 The appellant further states (same letter, page 2, paragraph 4 from the bottom) that the target agents and the target computer program must be different "entities" because it is claimed that the latter "transmits events occurring within" the former. The board accepts that the skilled person would understand (source and target) "agents" and "programs" to be distinguishable - and, in that sense, different "entities" - but considers that two pieces of the same computer program can be construed as two different "agents" irrespective of whether they run in the same or in different threads. In the board's view, the independent claims do not exclude that the pieces of software referred to as source and target agents run in the same threads as the source and target computer programs, respectively.

5.3 The board therefore considers that the terms "source" and "target agents" must be construed as any suitable program code carrying out the claimed functions.

Inventive step, Article 56 EPC

6. In the decision under appeal, the examining division based its assessment of inventive step on D1, arguing that it would have been obvious for the skilled person to apply the general solution of D1 to the particular case of user interfaces and in the context of a commonly known client-server architecture (see in particular, page 4, last paragraph, and page 5, paragraphs 3 and 7).

7. The board prefers to start from a scenario in which the issue of providing user interfaces in a distributed architecture actually arises and to assess whether the

skilled person would apply the solution of D1 in that situation. For that reason, the board prefers to start the assessment of inventive step from D5.

Main request

8. As summarised above, D5 discloses the claimed situation in which a user interface is provided (and modified) during the execution of a "source computer program" on a server and "for a target computer computer program executing on a remote client". It is further disclosed that events occurring within the source computer program are transmitted to the target computer program and *vice versa*.

8.1 In view of this, claim 1 of the main request differs from D5 at best in the following features:

- (a) The in-place code modification based on the provision of a "source agent" - distinct from the source computer program - including and executing the modification code and the insertion of an "instruction directing control to [the] source agent",
- (b) the provision of a "target agent" distinct from the target computer program, and
- (c) the fact that events are transmitted between the source and target agent.

8.2 The board considers that feature (b) is common-place in the art and therefore obvious from D5 alone: The web browser disclosed in D5 qualifies as the claimed "target agent" and it is well-known in the art that a web browser may run application code ("apps") and

handle events generated by the associated user interfaces.

- 8.3 Feature (a) represents a different solution to the problem - addressed in D5 - to modify the proprietary user interfaces of different vendors so as to provide a composite user interface with a homogenous look-and-feel.
- 8.4 D1 discloses a solution for the easy modification of application "functionality" of applications if the customer has no access to all relevant source code (see the abstract and right column, paragraph 1). In the board's judgment, this fits precisely the situation in D5 (figures 1 and 2) where the combined applications are provided by "different vendors" (see page 1, paragraph 3).
- 8.5 The board thus considers that the skilled person, addressing the given problem, would consider the disclosure of D1 as a solution to the problem posed.
- 8.6 Adopting the solution of D1 to the situation in D5 would yield the insertion of modification code and an instruction directing control to the modified code into the appropriate memory location (see D1, figure 2, "jmp DetourFunction" and the DetourFunction itself, see e.g. figure 1).
- 8.7 The appellant takes the position that the combination of D1 and D5 does not imply the use of a "source agent" that would (include and) execute the modification code and transmit and receive events in communication with the target computer.

9. The board does not agree that the provision of the source "agent" establishes an inventive step over D5 and D1. In particular, the board disagrees with the appellant's position that the detour function code of D1 is "very clearly not" an agent as claimed. On the basis of the present claim language which, in particular, does not specify in what way (and why) the execution of the modification code by the "agent" is different from the execution of the other application code, the board finds this difference too vague to establish an inventive step (see also point 5 above).
- 9.1 In the annex to the summons to oral proceedings, the board has given a further reason for its opinion that the provision of a "source agent" separate from the source program being modified was obvious (see points 11.2-11.4 in the summons). Although the board has no occasion to deviate from that argument, this issue can be left open in view of the board's finding above.
- 9.2 The board also considers that feature (c) is obvious. That events must be exchanged between client and server is known from D5, but also a matter of necessity for user interfaces. That the modification code may define operations requiring such events appears to be an obvious consequence of the fact that it relates to a user interface. Therefore, if events occur in the running modification code - i.e. the agent - it will have to be transmitted to the suitable software on the client computer, an obvious option in this regard being the web browser.
- 9.3 Therefore, the board concludes that claim 1 of the main request lacks inventive step over D5 and D1, Article 56 EPC.

Auxiliary request 1

10. In view of the fact that D5 already discloses the "combined user interfaces", the board considers that the preceding assessment directly carries over to claim 1 of the auxiliary request, which, thus, lacks inventive step, too.

Auxiliary request 2

11. The independent claims of auxiliary request 2 have been amended to specify the "multi-user aspect" of the invention, in particular that there are "at least two (first and second) modified user interfaces for first and second clients, and that there are first and second instances of the source program and associated agents.
 - 11.1 The board considers it to be evident that different users of the system of D5 may have different preferences or requirements.
 - 11.2 Moreover, the solution of D1 is suitable to address this problem, too. D1 discloses expressly that "the procedures in a DLL can be detoured in one execution of an application, while the original procedures are not detoured in another execution running at the same time. In the board's view this at least suggests the further possibility to have two different "detours" for two "executions" running at the same time.
 - 11.3 Therefore, the board concludes that also claim 1 of auxiliary request 2 lacks inventive step over D5 and D1, Article 56 EPC.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated