

Internal distribution code:

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

**Datasheet for the decision
of 11 April 2024**

Case Number: T 1644/20 - 3.5.06

Application Number: 10712825.8

Publication Number: 2409222

IPC: G06F9/44, G06F3/048

Language of the proceedings: EN

Title of invention:
EVENT RECOGNITION

Applicant:
Apple Inc.

Headword:
Event recognition/APPLE

Relevant legal provisions:
EPC Art. 123(2), 56

Keyword:
Amendments - added subject-matter (no)
Inventive step - (yes)

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 1644/20 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 11 April 2024

Appellant: Apple Inc.
(Applicant) One Apple Park Way
Cupertino CA 95014 (US)

Representative: Barton, Russell Glen
Withers & Rogers LLP
2 London Bridge
London SE1 9RA (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted on 13 February
2020 refusing European patent application No.
10712825.8 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman M. Müller
Members: A. Teale
B. Müller

Summary of Facts and Submissions

I. This is an appeal against the decision, dispatched with reasons on 13 February 2020, to refuse European patent application No. 10 712 825.8 - published as WO 2010/107669 - on the basis that the application according to a main and four auxiliary requests did not satisfy Articles 123(2) (added subject-matter), 56 (inventive step in view of document D1) and 84 (clarity) EPC. The cited document is as follows:

D1: US 5 627 959 A.

A fifth auxiliary request was not admitted into the proceedings, Rule 137(3) EPC.

II. A notice of appeal and the appeal fee were received on 7 April 2020, the appellant requesting that the decision be set aside in its entirety.

III. With a statement of grounds of appeal, received on 16 June 2020, the appellant refiled the main request and filed claims according to new amended first to third auxiliary requests. The appellant requested that a patent be granted based on said main and first to third auxiliary requests and made an auxiliary request for oral proceedings.

IV. In a communication setting out its preliminary opinion, the board stated that the amendments to the claims of all requests seemed to have added subject-matter, Article 123(2) EPC. The claims, in particular claim 1, of all requests were unclear, Article 84 EPC. The subject-

matter of claim 1 of all requests seemed not to involve an inventive step in view of D1, Article 56 EPC.

V. With a response dated 29 March 2024 the appellant filed amended claims according to first to fourth auxiliary requests and two versions of the description adapted to the first and second auxiliary requests and the third and fourth auxiliary requests, respectively. The appellant also made the previous second auxiliary request its new fifth auxiliary request.

VI. In the oral proceedings, held on 11 April 2024, the appellant withdrew its main request and requested that the decision be set aside and the case remitted to the examining division with the order to grant a patent based on the claims of the first to fifth auxiliary requests. At the end of the oral proceedings the board announced its decision.

VII. The application is being considered in the following form:

Description:

First and second auxiliary requests: pages 1 to 35, received with the letter dated 29 March 2024.

Third and fourth auxiliary requests: pages 1 to 34, received with the letter dated 29 March 2024.

Fifth auxiliary request: pages 2 to 34, as published in WO 2010/107669 A2, pages 35 to 40, received on 5 October 2011, and pages 1 and 1a, received on 3 May 2012.

Claims (received with the letter dated 29 March 2024):

First auxiliary request: 1 to 8.

Second auxiliary request: 1 to 8.

Third auxiliary request: 1 to 8.

Fourth auxiliary request: 1 to 8.

Claims (received as second auxiliary request with the grounds of appeal)

Fifth auxiliary request: 1 to 10.

Drawings (all requests):

Pages 1/13 to 13/13, as published in WO 2010/107669 A2.

VIII. Claim 1 of the first auxiliary request reads as follows:

"A method for processing user inputs, comprising: at an electronic device configured to execute software that includes a view hierarchy with a plurality of views: displaying a plurality of views of the view hierarchy; executing a plurality of software elements, each software element being associated with a particular view, wherein each particular view includes one or more input event recognizers, each input event recognizer having: an input event definition based on one or more input sub-events; and an input event handler, wherein the input event handler: specifies an action for a target, wherein the target comprises an object or an application executed by the electronic device; and is configured to send the action to the target in response to the input event recognizer detecting an input event, corresponding to a user input, that corresponds to the input event definition; detecting a sequence of one or more sub-events of an input event corresponding to a user input; identifying one of the views of the view hierarchy, that is a lowest level view in the view hierarchy in which a first sub-event in the sequence of one or more sub-events of the input event occurs, as a hit view, wherein the hit view establishes multiple views in the view hierarchy as actively involved views

in which the first sub-event is detected, wherein the actively involved views comprise the hit view; delivering a respective sub-event of the input event to input event recognizers for each view of the multiple actively involved views within the view hierarchy; and at the input event recognizers for the actively involved views in the view hierarchy, concurrently processing the respective sub-event prior to concurrently processing a next sub-event in the sequence of sub-events of the input event at the input event recognizers for each actively involved view in the view hierarchy."

Independent claim 7, setting out a computer readable storage medium, and independent claim 8, setting out a computer system or device, both refer to the method of claim 1.

Reasons for the Decision

1. Admissibility of the appeal

In view of the facts set out at points I to III above, the appeal fulfills the admissibility requirements under the EPC and is consequently admissible.

2. Summary of the invention

2.1 In the following the paragraph numbers refer to the international publication identified above.

2.2 The application relates to recognising user interface "events" comprising one or more "sub-events" in an electronic device; see figures 1A and 1B and [25-32]. The user interface can comprise a display and/or input

devices such as a touch-sensitive surface. In the case of a touch-sensitive display (see figure 1B; 156), an event can be a touch-based gesture; see page 6, line 10. The device may recognise a set of touch-based gestures, such as a tap, double tap, swipe, pinch or "depinch"; see [2]. A gesture, being an event, comprises sub-events. For instance, the "tap" gesture starts with a "finger down" sub-event; see figure 4C; 465-1.

- 2.3 The invention concerns an application including a view hierarchy. The lowest (finest) view in the hierarchy is that in which a user input sub-event, such as a "finger down" sub-event (see above), is detected; see [24], lines 7 to 10. This "hit view" then determines which views are to be "actively involved" in recognising the sub-events. Each view is associated with software including one or more "event recognizers", each having an event definition based on one or more sub-events. When an event recogniser detects an event, an event handler specifies an action for a target and sends the action to the target; see [4].
- 2.4 Figure 3A illustrates a view hierarchy consisting of an outermost view (302) encompassing the entire user interface and including subordinate views (search results panel 304, search text field 306 and home row 310). Subordinate views, for instance the search results panel (304), may themselves contain subordinate views; see subordinate view 305 "Maps view" for each search result.
- 2.5 A touch sub-event 301-1 is processed by outermost view 302 and, depending on its location, the subordinate views that it lies in, such as search results panel 304 and maps view 305, shown in figure 3A as 301-2 and 301-3. Hence the "actively involved" views of the touch

sub-event (dotted circles 301-1, 301-2 and 301-3) shown in figure 3A include the views outermost view 302, search results panel 304 and maps view 305; see [40].

2.6 Figures 3B and 3C illustrate methods and structures related to event recognisers, the claims being directed to the case in figure 3B in which event handlers are associated with particular views within a hierarchy of views; see [43]. The hit view determination module (314) establishes whether a sub-event has taken place within one or more views (see [44]) and, if so, identifies the lowest view in the view hierarchy as the "hit-view". The "actively involved" views receive not only the first sub-event but also all following ones related to the same touch source, even if the gesture leaves the hit view; see [45, 49]. For each of the actively involved views, one or more gesture recognisers, as illustrated in figures 4A, 4B and 5A to 5C (see [84-105]), use a state machine to identify a predefined sequence of sub-events to recognise a gesture such as a "scrolling event"; see figure 5B; 582.

3. Added subject-matter, Article 123(2) EPC

3.1 In its provisional opinion (point 8.3), the board questioned whether claim 1 of the main and first auxiliary requests contained added subject-matter. According to paragraphs [47,49] as originally filed, the sub-event delivery module (318) (see figure 3B) delivered sub-events to event recognizers for all actively involved views and not, as set out in claim 1, (possibly) to only some of the views, possibly excluding the hit view. The description set out an example in which sub-events were delivered to the hit view, see [45].

3.2 Claim 1 of the present first auxiliary request overcomes this objection by stating that "the actively involved views comprise the hit view", thus setting out that sub-events are delivered to the hit view. Hence claim 1 of the first auxiliary request complies with Article 123(2) EPC.

4. The board's understanding of the invention

4.1 Claim 1 of the first auxiliary request sets out "processing user inputs" by displaying a hierarchy of views, each including one or more "input event recognizers", each having an "input event definition" based on one or more sub-events (for instance a "finger down" sub-event; see figure 4C; 465-1). As the user input sub-events are associated with views which are displayed by an electronic device (see figure 3A), the board understands the claimed input events to be user interface events.

4.2 The expression in claim 1 "hierarchy of views" requires interpretation. The board understands the "hierarchy" to relate to how the views are nested on the display of the electronic device; see paragraph [38] and figure 3A. The smallest view in which the initial sub-event is registered is, by definition, the hit view and those around it form the hierarchy. It is the hit view which determines the "actively involved" views which are then to receive, process and possibly recognise sub-events, including the initial one. Although the hit view is described as the lowest view in the hierarchy (see paragraphs [24, 38] and figure 3A; "maps view" 305), this does not mean that sub-events which occur in the hit view are only relevant to the hit view. On the contrary, as explained by the appellant in the oral proceedings, user input events occurring at the hit

view, for instance scrolling or a double tap (see figures 5B and 5C, respectively), may be relevant to, and recognised, by other views, for example those encompassing the hit view; see also paragraphs [47] and [124]. For instance, scrolling in the "maps view" (305) may be recognised by the "search results panel" (304) surrounding the maps view.

4.3 When an input event recogniser recognises an event, defined by its input event definition, then an input event handler specifies an action for a target, the target being an object or application executed by the electronic device. The board understands an "action for a target" in this context to be broad language referring to what happens in response to the specified event having been recognised.

4.4 Claim 1 sets out the input event recognisers of the actively involved views "concurrently processing the respective sub-event prior to concurrently processing a next sub-event". The board notes that, according to paragraphs [9,12] as originally filed, "In some embodiments, event recognizers for actively involved views in the view hierarchy may process the sequence of one or more sub-events concurrently 650; alternatively, event recognizers for actively involved views in the view hierarchy may process the sequence of one or more sub-events in parallel." In this context the board understands "parallel" processing to require separate hardware for the individual processes and "concurrent" processing to, possibly, allow the execution of the individual processes on shared hardware, for instance by time slicing. In a way, concurrency is the simulation of parallelism on not necessarily parallel hardware. The effect of the concurrent processing, set out in claim 1, is that no initial exclusive decision must be

made which event handler "handles" an event and that it is not necessary for one recogniser to finish its work (successfully or not) before another can start processing the same sub-event. This is what the appellant has broadly described as "processing individual sub-events in parallel whilst processing the sequence of sub-events in series".

5. Document D1 (US 5 627 959 A)

5.1 D1 relates to manipulating a database using graphic objects; see abstract and column 2, line 32, to column 4, line 19. This involves displaying graphic objects on a screen and using a mouse to control a pointer on the screen to manipulate the objects. Inputs from the mouse, such as "mouse button down" and "mouse button up" are received and can trigger a customized procedure, causing a graphic object, termed a "button" object, to be displayed. A tree-like organisation is used to group the graphic objects; see abstract and figure 4.

5.2 The "active layer", which is the only layer that the user interacts with, and other layers are displayed together, each layer containing a plurality of objects and grouped objects; see figure 2 and column 7, lines 48 to 59. As shown in figure 4, all the layers (412-418) and their groups of objects can be organised in a tree structure having a root (410). In the group tree the object on the right is always drawn last, i.e. it is "on top"; see column 8, lines 25 to 44.

5.3 Customised procedures are executed at "trigger" points when a mouse event, such as clicking, occurs at a "button" object; see column 8, lines 58 to 60. As shown in the example in figures 10 and 11, clicking (mouse

button down) on a "button" object (1030) for the city of San Francisco causes the procedure in table 4 to be executed, displaying a window containing a chart (1130) relating to the city; see column 13, line 21, to column 15, line 17. The window has a "close" button object (1140) which, when clicked (mouse button down), causes the procedure in table 5 to be executed, closing the window; see column 14, line 66, to column 15, line 17.

5.4 A mouse down (click) event determines which object having an associated procedure becomes "event active", this object receiving not only the "mouse down" event but also all subsequent events until the next "mouse down" event; see column 10, lines 21 to 31. When a "mouse click" event is detected, the first object having a procedure for handling a mouse event is located as the topmost object on the active layer pointed to by the mouse. If this object has a procedure for handling a mouse event, then it is designated "event active", otherwise known as an "active button object" (ABO), and put in the button execution history. This contains a record of executed button procedures; see figure 6; 622-4 and column 12, lines 13 to 52. If it has no such procedure, then a search is made up the object tree for the first object having such a procedure; see figures 2 and 4; see column 11, lines 9 to 30. If the ABO has a procedure for a "mouse down" event, then it is executed, the same being carried out for all subsequent events until a "mouse up" event is detected, causing the ABO to be deactivated; see column 11, lines column 11, lines 39 to 49.

5.5 As shown in figure 7, while the mouse button remains down (step 702), any mouse events are processed by the ABO. This only changes when the mouse button is released and a new layer object is identified (step

704). The board understands this to mean that all user interface events are processed by the ABO until the mouse button is released.

5.6 The board regards the objects in D1 as "views" in the terms of claim 1 of the first auxiliary request. The objects arranged in the tree structure of figure 4 form a hierarchy. The ABO in D1 has event recognisers, since it can recognise "mouse down", "mouse move" and "mouse up" sub-events; see figures 6, 7 and 8, respectively. A "button procedure" in D1 falls under an "event handler" in claim 1. The activation of the object layer "containing the SF chart" (see column 14, table 4, line 7) constitutes an action to be sent to a target.

5.7 Hence, in the terms of claim 1 of the first auxiliary request, D1 discloses a method for processing user inputs (from the mouse), comprising: at an electronic device configured to execute software that includes a view hierarchy with a plurality of views (see figures 2, 3A and 4): displaying a plurality of views of the view hierarchy; executing a plurality of software elements (button procedures; see column 10, lines 21 to 23), each software element being associated with a particular view (ABO), wherein each particular view includes one or more input event recognizers, each input event recognizer having: an input event definition based on one or more input sub-events; and an input event handler, wherein the input event handler: specifies an action for a target (see table 4), wherein the target comprises an object or an application executed by the electronic device (see tables 4 and 5, columns 13-15); and is configured to send the action to the target in response to the input event recognizer detecting an input event, corresponding to a user input, that corresponds to the

input event definition; detecting a sequence of one or more sub-events of an input event corresponding to a user input; identifying one of the views of the view hierarchy, that is a lowest level view in the view hierarchy in which a first sub-event (figure 7, mouse button down 702) in the sequence of one or more sub-events of the input event occurs (see column 11, lines 9 to 30), as a hit view (see "active button object (ABO)).

6. Inventive step, Article 56 EPC

6.1 The appealed decision discussed the inventive step of claim 1 of the then second auxiliary request starting from D1, although the decision does not state which features of claim 1 were known from D1. The board understands the decision to find that the subject-matter of claim 1 differed from the disclosure of D1 in that events were processed concurrently. According to point 7.1 of the decision, the skilled person would have added the difference feature to speed up event recognition. The skilled person could have solved this problem in two equally obvious ways: using multiple processors or concurrent processing, the second choice being an obvious one. The recognition problem in D1 was moreover naturally partitioned, figures 4 and 5 showing view layers in a tree-like structure. The processing of events in simple sequences in D1 did not teach away from concurrent processing of events. The skilled person would have been aware of registering event recognizers and conflicts. Concurrency did not make event processing more flexible.

6.2 The appellant has pointed out that D1 does not disclose touch input, relating instead to mouse input, and contains no hint at concurrently processing user input

events to recognise events. In D1 sub-events were only processing by one object at a time, and only for objects within an active layer, and conditional on rejection of the event by a preceding object in the object group. For subsequent sub-events D1 disclosed processing only by the active button object. It was not possible in the method of D1 to process a subsequent sub-event with multiple objects, in particularly concurrently, as claim 1 required. The appellant emphasised that, in contrast to the invention, D1 taught that subsequent sub-events were treated differently from the initial sub-event in that they were sent directly to the single active button object (ABO) determined from the initial sub-event. Compared to the one-by-one, mouse-based processing of D1, the invention carried out concurrent processing of a sub-event across multiple actively involved views. This speeded up gesture recognition whilst preventing event recognisers of the views from making event recognition decisions out of turn, for instance too early or too late. This decreased the chance of conflicts between views and so ensured that the sequence of sub-events was processed quickly and accurately. The method of claim 1 also increased the flexibility of processing of a sequence of sub-events, for instance if an actively involved view was initially interested in the sequence of sub-events based on the first sub-event, but subsequently lost interest in the sequence of sub-events. The technical problem to be solved was thus to increase the speed and flexibility of processing a sequence of sub-events input by a user on an electronic device. D1 did not permit concurrent processing by different nodes of an object group of subsequent sub-events, since all subsequent sub-events were sent to the same object.

6.3 The board finds that, in view of the above analysis, the subject-matter of claim 1 of the first auxiliary request differs from the disclosure of D1 in the following steps:

- a. the hit view establishes multiple views in the view hierarchy as actively involved views in which the first sub-event is detected, wherein the actively involved views comprise the hit view;
- b. delivering a respective sub-event of the input event to input event recognizers for each view of the multiple actively involved views within the view hierarchy and
- c. at the input event recognizers for the actively involved views in the view hierarchy, concurrently processing the respective sub-event prior to concurrently processing a next sub-event in the sequence of sub-events of the input event at the input event recognizers for each actively involved view in the view hierarchy.

6.4 Regarding the effect of these difference features, the board is not convinced that they accelerate event recognition in general. Concurrent processing, as opposed to parallel processing, cannot speed up the overall execution of all tasks. Notably if one uses non-parallel hardware, the overall execution time cannot improve if one carries out the individual tasks in an interleaved manner. One might achieve a speed increase in the situation in which the overall computation requires only some of the individual tasks to complete. For example, if, once the first (or the first few) event recogniser has recognised a series of

sub-events as an event, all other recognisers were terminated. The board also notes that the invention does not change the variety of sub-event sequences that can be recognised. The ABO in D1 can recognise sub-events, and, in the same way, each recogniser in claim 1 can recognise an event comprising a single sub-event.

6.5 The board finds however that the difference features do have a technical effect, namely to allow views to recognise user input events comprised by sub-events not in that view. For instance, a scrolling event (see figure 5A) in the maps view (305) in figure 3A can be recognised by a recogniser of the search results panel view (304). This adds a new functionality to the user interface, since the maps view cannot itself scroll, but the search results panel view (304), being a list, can.

6.6 In the system of D1, it may happen that two objects above each other in the hierarchy are both able to process a mouse event. If the user intends to interact with one of these objects via a sequence of mouse events, it may happen that another object covers it and receives the mouse events against the user's wishes. The board considers that this situation is undesirable and also that the user would recognise this situation as undesirable. Addressing this shortcoming of D1 would therefore have been obvious. The board considers that the skilled person would have addressed the problem of avoiding the problem that it is difficult, even impossible at times, for the user to interact with objects higher up in the hierarchy - or, equivalently, more in the background of the user interface.

6.7 The board sees several ways of approaching this problem. One is to instruct the user to always interact

with a visible part of the object of interest. Another would be to define the events in such a way that there is no ambiguity as to which event is addressed to which object. Both may not be possible for practical reasons. Yet another option seems to be to send a sequence of events to the event recognisers of several or all objects covering one another. This choice already differs substantially from the solution proposed in D1. And it still lacks the claimed concurrent execution of event handlers. This concurrency however provides a degree of flexibility in handling the mentioned conflicts between objects within a hierarchy, and allows for a speed up at least in certain situations. The board accepts this as a technical effect. Moreover, even though this speed up is not achieved for all possible scenarios in question - in particular event definitions and conflict resolution "policies" - the board considers that the skilled person would understand the options and limits of concurrency.

6.8 The board agrees with the applicant that D1 contains no hint of a plurality of objects processing and possibly recognising user input sub-events concurrently, as set out in difference features "a" to "c" above.

6.9 The board concludes that the subject-matter of claim 1 of the first auxiliary request involves an inventive step in view of the disclosure of D1.

7. Remittal, Article 111(1) EPC

Whilst the board finds that claim 1 of the first auxiliary request involves an inventive step in view of D1, the board finds it appropriate that the case be remitted to the examining division for further prosecution pursuant to Article 111(1), second sentence, EPC

since documents D2 to D4 were not considered in the context of inventive step in the appealed decision. This constitutes a special reason, Article 11 RPBA 2020, for remitting the case to the examining division, so that the relevance of documents D2 to D4 to inventive step may be considered in the light of the amended claims.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the examining division for further prosecution.

The Registrar:

The Chairman:



L. Stridde

M. Müller

Decision electronically authenticated