

Internal distribution code:

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

**Datasheet for the decision
of 2 March 2011**

Case Number: T 0004/08 - 3.5.01

Application Number: 02026855.3

Publication Number: 1406183

IPC: G06F 17/30

Language of the proceedings: EN

Title of invention:

Method and system for refreshing browser pages

Applicant:

SAP AG

Opponent:

-

Headword:

Refreshing browser pages/SAP

Relevant legal provisions:

EPC Art. 52(1)

Relevant legal provisions (EPC 1973):

EPC Art. 56

Keyword:

"Inventive step (no)"

Decisions cited:

-

Catchword:

-



Case Number: T 0004/08 - 3.5.01

D E C I S I O N
of the Technical Board of Appeal 3.5.01
of 2 March 2011

Appellant:

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf (DE)

Representative:

Rocke, Carsten
Müller-Boré & Partner
Grafinger Strasse 2
81671 München (DE)

Decision under appeal:

Decision of the Examining Division of the
European Patent Office posted 13 August 2007
refusing European patent application
No. 02026855.3 pursuant to Article 97(1) EPC
1973.

Composition of the Board:

Chairman: S. Wibergh
Members: P. Scriven
P. Schmitz

Summary of Facts and Submissions

- I. The appeal is against the Examining Division's decision to refuse European patent application 02026855.3.
- II. That decision was given in the form of references to three communications the Examining Division had previously sent, and was on the basis of a main and an auxiliary request. The main request was rejected due to a lack of inventive step, in consideration of the disclosure of document D1 (EP-A 1 221 661), as well as due to a lack of clarity. The auxiliary request was rejected on the ground of added subject matter.
- III. With the statement setting out the grounds of appeal, the appellant requested that the decision be set aside and that a patent be granted on the basis of a main request, or, alternatively, on the basis of first or second auxiliary requests. Oral proceedings were also requested, if the main request was not considered to be allowable.
- IV. The Board sent a summons to attend oral proceedings, to be held on 2 March 2011, and set out its provisional view that none of the requests was allowable.
- V. With the letter dated 19 January 2011, the appellant filed a new main request and a new auxiliary request, to replace the requests filed with the statement of grounds of appeal, and provided arguments in favour of disclosure, regarding the differences between the invention as claimed and the disclosure of D1, and regarding inventive step.

VI. Oral proceedings were held as scheduled. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the main request or the auxiliary request, both filed with the letter dated 19 January 2011.

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

A method (400) for modifying a document object model (DOM) hierarchy (800) in a browser at a client, wherein the client uses the browser and is provided with a framework that enables the client to render UI elements (701, 702, 703) included in a UI page, wherein the DOM hierarchy (800) represents UI elements (701, 702, 703) of the UI page in the browser, the browser presenting a screen presentation of the UI page that corresponds to the DOM hierarchy (800), and wherein the browser supports DHTML and can use the framework to manipulate the UI page without a need for server roundtrips, wherein the corresponding content that is needed for the manipulation is stored in a HTML-DOM cache at the client, comprising the steps:

identifying a change of a user interface (UI) element (701) that references (601) a node (801) of the DOM hierarchy (800);

determining (420) whether the change of the UI element (701) can be applied to the DOM hierarchy by using a delta renderer, wherein the delta renderer is a set of functions that can modify the DOM representation of a UI element directly by

using setter functions such as setValue, setMaxLength, setColor, wherein the delta renderer can be used if a function is included in the delta renderer that can be used to adjust the DOM hierarchy (800) according to the change;

if the change can be applied by using the delta renderer, finding (430) in the DOM hierarchy (800) the node (801) that is referenced by the UI element (701) and modifying (440) at least one attribute related to the node (801) according to the change by using the delta renderer, wherein each of the UI elements (701, 702, 703) of the UI page references a node (801, 802, 803) in the DOM hierarchy (800); and

else, setting (450) a dirty flag.

- (a) Claim 1 according to the auxiliary request adds, after *else, setting a dirty flag*, the following text:

wherein the dirty flag indicates invalidity of a cached DOM representation of any further UI element (702, 703) up to the root node of the UI tree of the UI page that is on the path of the UI element (701) in the UI tree (700) of the UI page.

VIII. The appellant's arguments can be summarized as follows.

During the written procedure, the appellant argued that of the three documents mentioned by the Examining Division, D1 was the most relevant, but that the Examining Division had failed to recognize substantial

differences between its disclosure and the subject matter claimed. In particular, while the invention was concerned with modifying attributes of nodes in a DOM hierarchy, D1 taught away from using a DOM hierarchy at all (D1, [0004]). Thus, the skilled person would not have found it obvious to modify the teaching of D1 in such a way as to arrive at the invention claimed.

During the oral proceedings, the appellant accepted that the first part of claim 1 according to both requests (*A method for modifying ... identifying a change of a user interface (UI) element that references a node of the DOM hierarchy*) was known from the prior art. The appellant also accepted that prior art DOM usage allowed automatic updating of the UI page. The assertion to the contrary, in [0004] of D1, was not accurate. The appellant argued that, from this starting point, the skilled person would not find it obvious to add the delta renderer and the dirty flag. The technical effect was to speed up the presentation of UI changes, because of the direct modification of DOM nodes, without any change in hierarchy. This meant that the modification could be carried out straight away, and that, at least sometimes, a wholesale re-rendering could be avoided.

The appellant has also argued that the delta renderer was sufficiently disclosed, in the sense of Article 83 EPC, because the skilled person could find at least one way of implementing it. Claim 1 according to both requests defined the delta renderer as consisting of setter functions which directly modify attributes of nodes without changing the structure of the DOM

hierarchy. Such setter functions were part of the skilled person's general knowledge.

Reasons for the Decision

1 The Board considers neither request to be allowable due to a lack of inventive step (Article 52(1) EPC and Article 56 EPC 1973). The reasons are as follows.

2 *Main request*

2.1. The appellant has argued that the approach taken in D1 was considerably different from that taken in the present invention. The Board agrees with this, to the extent that the status of the DOM in D1 is unclear. D1 identifies, at [0004], a number of problems with the *standard Document Object Model (DOM)* approach and proposes an alternative approach in which it is not quite apparent whether there is something which still corresponds to a DOM, or not.

2.2. The Board's view is that the first part of claim 1, which the appellant has acknowledged as prior art, represents a more reasonable starting point for the assessment of inventive step.

2.3. From this starting point, the invention amounts to this: when an event requires a change in the DOM, first check to see whether this can be effected by a delta renderer; if it can, then do it that way; if it can't, set a flag.

2.4. During the oral proceedings, it was established that the appellant understands a delta-renderer to be a

collection of setter functions which can modify attributes of DOM nodes directly. That is, the delta renderer is a subset of the set of those functions which can modify the DOM representation of a UI element directly. It is common ground that such setter functions were well known. The invention, in the view of the appellant, lies in the use of just these functions as a first step; if they are not sufficient to carry out all the changes needed, this is flagged, and a standard DOM modification procedure would take care of it in a later step.

2.5. According to the appellant, the technical effect of this is to speed up the presentation of UI changes. This is said to follow, because the setter functions modify the attributes of DOM nodes directly, but do not change the tree structure. If the changes can be made in this way, then it may be possible to avoid wholesale re-rendering of the UI page.

2.6. The Board notes, however, that claim 1 does not define any step involved in rendering. Rather, the method stops once the changes to the DOM have been made by the delta renderer, or else it has been flagged that they cannot be made. At best, then, the claimed invention achieves a speed up in modifying the DOM. Although this might contribute to a speed up in rendering as perceived by the user, that need not be the case. An event might well, for example, require a change in font size for the first line of a page. That is achievable by a setter function, and it is the only thing that need change in the DOM. However, the visual layout of the whole UI page is affected, and it has to be re-rendered, because subsequent lines must appear further

down in order to make room for the larger font size. Even the effect of speeding up the modification of the DOM is not certain, because there is some additional overhead involved in checking whether the delta renderer can be applied or not. The Board assumes, however, that the skilled person can specify a collection of functions whose benefits will, in general, outweigh the overhead. If it were not so, some essential feature would be missing, and, indeed, would not be disclosed.

2.7. The Board also notes that the delta renderer is defined, in claim 1, as consisting entirely of setter functions. According to the description, however, setter functions are simply an example of the sorts of function which can be included in the delta renderer (application as published, [0019]: *The delta renderer is a set of functions that can modify the DOM representation of a UI element directly by using, for example, setter functions ...*), and no particular advantage attaches to them, over and above what attaches to any function which *can modify the DOM representation of a UI element directly*. The restriction to setter functions, then, is an arbitrary selection.

2.8. The salient difference lies in the preliminary use of a set of functions which *can modify ... directly*. The term *directly* is not entirely clear, but the Board understands it to refer to functions which are, in some sense, relatively simple. When considering the whole set of functions available for modifying a DOM, some will certainly be computationally more straightforward and faster to apply, than others. The Board interprets the term *directly* as stipulating that the functions in

- the delta renderer are those which are relatively simple in this sense.
- 2.9. The question of inventive step comes down to this: would it have been obvious to the skilled person, concerned with the speed of updating to the DOM, first to try a reduced set of functions (the delta renderer), which are expected to be simple to apply and to indicate when this fails?
- 2.10. The Board's view is that this would have been obvious. A function's complexity is one of the things the skilled person, a computer programmer, considers daily. She is aware that some functions are simpler and faster to apply than others. She would expect that working entirely with such functions would generally be faster than working with more complicated functions. Thus she would formulate the idea of first trying some simple functions. It is inherent in the concept of trying, that the simple functions may be insufficient. A programmer knows quite well that the set of things that can be done with a small set of functions may be strictly smaller than the set of things which can be done with more functions. It would follow naturally, that the fact that the delta renderer is insufficient must be indicated. That is all that the flag does in the main request.
- 2.11. The subject matter of claim 1 according to the main request, therefore, does not involve an inventive step (Article 56 EPC 1973).

3 *Auxiliary request*

- 3.1. According to claim 1 of the auxiliary request, *the dirty flag indicates invalidity of a cached DOM representation of any further UI element (702, 703) up to the root node of the UI tree of the UI page that is on the path of the UI element (701) in the UI tree (700) of the UI page.*
- 3.2. The Board understands this to mean that those nodes in the DOM which are involved in changes the delta renderer cannot manage are flagged. The idea, then, is to flag where the failures occur. In the Board's view, once the skilled person has the idea of flagging failure, it would be obvious to go on to show where it occurs.
- 3.3. The subject matter of claim 1 according to the auxiliary request, therefore, lacks inventive step (Article 56 EPC 1973).

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:

T. Buschek

S. Wibergh