BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS

BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE

CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN DES BREVETS

Internal distribution code:

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

Datasheet for the decision of 13 February 2020

Case Number:	T 0452/14 - 3.5.01
Application Number:	06802783.8
Publication Number:	1952264
IPC:	G06Q10/00, G06F17/00, G06F17/21
Language of the proceedings:	EN

Title of invention:
INTEGRATING CHARTS IN DOCUMENTS

Applicant:
Microsoft Technology Licensing, LLC

Headword:
Integrating charts in documents / MICROSOFT

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - handling user action with non-visible instance of spreadsheet (no - obvious)

This datasheet is not part of the Decision.
It can be changed at any time and without notice

Europäisches Patentamt
European Patent Office
Office européen des brevets

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 0452/14 - 3.5.01

DECISION
of Technical Board of Appeal 3.5.01
of 13 February 2020

Appellant:
(Applicant)

Microsoft Technology Licensing, LLC
One Microsoft Way
Redmond, WA 98052 (US)

Representative:

Grünecker Patent- und Rechtsanwälte
PartG mbB
Leopoldstraße 4
80802 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 4 September 2013 refusing European patent application No. 06802783.8 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman	W. Chandler
Members:	W. Zubrzycki
	P. Schmitz

Summary of Facts and Submissions

I. This is an appeal against the decision of the examining division to refuse the European patent application No. 06802783.8 for lack of inventive step and added subject-matter (Articles 56 EPC and 123(2) EPC).

II. In the statement setting out the grounds of appeal, the appellant requested that the decision under appeal be set aside and a patent be granted on the basis of the main or first to third auxiliary requests filed therewith. Apart from minor amendments made to dependent claims, these requests corresponded to the refused requests. The appellant also requested oral proceedings.

III. The Board arranged for oral proceedings on 11 February 2020. In the communication accompanying the summons, the Board gave its preliminary view that claim 1 of the main and first to third auxiliary requests did not involve an inventive step (Article 56 EPC) and claim 1 of the second and third auxiliary requests contained added subject-matter (Article 123(2) EPC).

IV. In a reply dated 16 October 2019 the appellant requested a postponement of the oral proceedings. The Board acceded to the request for postponement and rescheduled the oral proceedings.

V. In a reply to the summons dated 20 December 2019, the appellant filed amended main and first to third auxiliary requests.

VI. Oral proceedings were held on 13 February 2020. At the oral proceeding the appellant confirmed the requests filed on 20 December 2019 and requested that the decision under appeal be set aside and a patent be granted on the basis of these requests.

VII. Claim 1 of the main request reads:

A computer-implemented method for managing and editing a chart (322) presented in a host application (300, 402, 404), the method being performed by processing computer-executable instructions stored on a computer-readable medium, comprising:
receiving (602) a user action associated with the chart;
determining (604) whether a spreadsheet (350, 410) is needed according to the user action, wherein a spreadsheet is determined to be needed when the user action affects chart data (122, 352-360, 412) provided by the spreadsheet;
determining (608) whether a visible instance of the spreadsheet is required to handle the user action, wherein a visible instance of the spreadsheet is determined to be required when handling the user action involves providing the user with the opportunity to change the chart data or allowing the user to interact with the chart data while in spreadsheet format;
generating (612, 614) a link between the host application and the spreadsheet;
sending (616) the user action to the spreadsheet;
determining (618) whether chart data (122, 352-360, 412) changes; and
determining (622) whether an update to the chart is required based on the changes.

VIII. Claim 1 of the first auxiliary request adds further features to claim 1 of the main request. It reads (added features underlined):

A computer-implemented method for managing and editing a chart (322) presented in a host application (300, 402,404) interacting with a spreadsheet application (350, 410), wherein a spreadsheet document is used as the source for chart data (122, 352-360, 412) for the chart, wherein the chart is edited according to the feature set provided by the spreadsheet application, the method being performed by processing computer-executable instructions stored on a computer-readable medium, comprising:
receiving (602) a user action associated with the chart;
determining (604) whether a spreadsheet is needed according to the user action, wherein a spreadsheet is determined to be needed when the user action affects chart data (122, 352-360, 412) provided by the spreadsheet;
if it is determined that a spreadsheet is not needed, handling (606) the user action without accessing the chart data provided by the spreadsheet document;
if it is determined that a spreadsheet is needed, determining (608) whether a visible instance of the spreadsheet application is required to handle the user action, wherein a visible instance of the spreadsheet is determined to be required when handling the user action involves providing the user with the opportunity to change the chart data or allowing the user to interact with the chart data while in spreadsheet format;
if it is determined that a visible instance of the spreadsheet application is required, initiating (610) a visible instance of the spreadsheet application, and

otherwise initiating a non-visible instance of the spreadsheet application;
generating (612, 614) a link between the host application and the spreadsheet application;
sending (616) the user action to the spreadsheet application;
determining (618) whether chart data changes; and
determining (622) whether an update to the chart is required based on the changes, and if so, requesting (624) an update of the chart data from the spreadsheet application.

IX. Claim 1 of the second auxiliary request differs from claim 1 of the first auxiliary request by:
- replacing the preamble with the following wording (differences struck through and underlined):
"A computer-implemented method for managing and editing a chart (322) presented in a host application (300, 402, 404) interacting with a spreadsheet application (350, 410), wherein a host application file (502) includes a chart file (532, 534) directing the host application to ~~spreadsheet document is used as the source for~~ a spreadsheet file (540, 550) that includes chart data (122, 352-360, 412) for the chart, the spreadsheet file being stored either as a spreadsheet file embedded within the host application file or as a spreadsheet file external to the host application file, wherein the chart is edited according to the feature set provided by the spreadsheet application, the method being performed by processing computer-executable instructions stored on a computer-readable medium, comprising"
- replacing in the third method step "spreadsheet *document*" by "*spreadsheet file*"

X. Claim 1 of the third auxiliary request differs from claim 1 of the second auxiliary request by:
- replacing all occurrences of "*host application*" by "*presentation application*" and replacing all occurrences of "*host application file*" by "*presentation file*"
- adding to the preamble the wordings *"wherein the spreadsheet application determines both the values and format to be used for a chart, and the presentation application displays the chart" and "the chart data including the values to be used for the chart"*

Reasons for the Decision

1. Background

The invention concerns a method for managing and editing a chart presented in a host application. Users may request updates to the displayed chart by means of user actions for example keystrokes or mouse clicks (originally filed application, page 10, lines 8 to 11). The update processing varies depending on the nature of the requested updates.

If the requested update merely aims at changing the chart position, the host application repositions the chart itself (page 10, lines 16 to 18). If, however, the update aims at modifying the values of the data in the chart, the spreadsheet application is launched and the update is performed using the spreadsheet application (page 10, lines 26 to 30). The spreadsheet application can be run as a "visible" or "non-visible instance". The visible instance is run, if it is established based on the user action that the user needs to interact with the spreadsheet data set; the non-visible one if the user has no such need (paragraph bridging pages 10 to 11).

The application does not give any examples of user actions that the non-visible spreadsheet instance should handle, or what processing it performs when it runs.

The user action notification is passed from the host application to the running spreadsheet application instance which can then modify the spreadsheet data set underlying the chart (page 11, last two paragraphs). If the change to the spreadsheet data set necessitates an update to the chart visualising it, the chart update is requested (page 12, paragraphs 1 to 4).

The invention is further concerned with a file structure enabling persistence of a host application document including an embedded chart and an associated spreadsheet data set. The host application document is stored as a file containing a pointer to the spreadsheet document containing the spreadsheet data set on which the chart is based. The spreadsheet document is either embedded in the host application file or stored in an external file (page 9, lines 20 to 29).

2. First auxiliary request, claim 1

The Board finds it convenient to discuss the more specific first auxiliary request before the main request.

2.1.1 Inventive step - Article 56 EPC

The closest prior art D5 discloses a system enabling a spreadsheet object to be incorporated into a word processing application document. The spreadsheet object comprises tabular data and a chart (column 8, lines 50 to 56; Figure 3). The user can select the spreadsheet object in the word processor window and edit the object directly in this window using the functionality of a spreadsheet application launched for this purpose by the word processor (column 8, lines 19 to 24 and lines 31 to 39). In addition to editing the spreadsheet object using the spreadsheet application, the user can cut or copy the embedded spreadsheet object (column 8, lines 29 to 40 and Figure 8). Activating menu items triggering the cut, copy and edit options corresponds to receiving a user action associated with the chart and determining whether a spreadsheet is needed to perform the action in the first two features of claim 1. Column 8, lines 22 to 28 discloses that if the edit action is selected, it is sent from the word processor to the launched spreadsheet application using a generated link. This corresponds to the generating and sending steps of the claim.

The appellant argued during the oral proceedings that D5 employed the object linking and embedding (OLE) API and had to be read in this context. More particularly, within the OLE-based framework, an object comprised only programming code and it was not correct to map the objects and spreadsheet instances of D5 to the claimed entities.

The Board does not find this argument persuasive. It is correct that D5 discloses that in the preferred embodiment the word processor and the spreadsheet application communicate using interface methods defined by the OLE API (column 8, line 50 to column 9, line 10 and column 9, lines 39 to 44). However, the passages of

D5 cited above are not part of this preferred embodiment. They rather provide general teaching on a functional level and are not limited to using the OLE API. Beyond that, the Board cannot see why using the OLE API for implementing communication between the host application (container application in terms of D5) and the spreadsheet application should invalidate the above straightforward reading of D5. In fact, the application also discloses at page 7, line 15 to page 9, lines 4 that the spreadsheet application and other software components communicate through a set of COM interfaces.

2.1.2 The main difference between the subject-matter of claim 1 and D5 is that:
A) *If it is determined that a spreadsheet is needed, it is determined, whether a visible instance of the spreadsheet application is required to handle the user action and a visible instance of the spreadsheet application is initiated if this determination is positive, otherwise a non-visible instance of the spreadsheet application is initiated.*

In addition to this main difference, the subject-matter of claim 1 further differs from D5 in that:
B) The chart is edited according to the feature set provided by the spreadsheet application.
C) *It is determined whether chart data changes and whether an update to the chart is required based on the changes, and if so, an update of the chart data is requested from the spreadsheet application.*

2.1.3 Concerning the main distinguishing feature A, the appellant agreed that the application did not explicitly discuss operations performed in the invisible mode. However, he argued that the skilled person would infer from description page 10, lines 28 to 31 that actions performed in the invisible mode were "uni-directional" actions which did not require the user to change the chart data or to interact therewith. It would be clear to the skilled person that straightforward examples of such actions were changing color themes of the chart or the chart's font size. The skilled person would also infer from the description that these uni-directional actions were triggered by predefined keyword shortcuts. Furthermore, performing the actions in the invisible mode contributed to saving computer and memory resources, as less information was displayed. Hence, in the appellant's opinion the objective technical problem was "how to save computer and memory resources"

The appellant argued further that recognising that certain actions could be performed in an invisible mode in order to save computer resources was not obvious over D5, as D5 did not provide any hint in this direction. In the absence of such a hint, the skilled person starting at D5 would perform all chart-related spreadsheet actions in the visible mode shown in Figure 4.

2.1.4 The Board considers that the problem of saving resources is in this case too general and speculative. The amount of consumed resources depends, in both visible and invisible mode, on the actions' nature and the manner in which they are processed rather than merely on the decision to perform actions visibly or invisibly for the user.

The fact that neither the claim nor the description discuss actions performed invisibly to the users in positive terms poses a difficulty in formulating the objective technical problem. As long as it is not clear what these actions are and what processing ensues, their effect cannot be determined. However, the identification of an effect produced by novel features is essential in the problem and solution approach, because such an effect provides basis for the formulation of the objective technical problem.

Furthermore, the problem cannot be based on the negatively formulated effect of enabling actions whose handling does not require the user to interact with the chart data. The reason is that such a problem comprises a pointer to the offered solution. And it is even less appropriate to include in the problem that certain spreadsheet actions should be performed invisibly for the user.

In the Board's view, one credible problem follows from the appellant's inferred feature that the actions that result in invisible processes must be triggered by predefined keyword shortcuts. Providing such shortcuts that then enable triggering spreadsheet actions increases user friendliness. Thus the Board bases the objective technical problem on this effect along the lines of "how to increase user friendliness of the system of D5". The skilled person confronted with this problem and referring to his common general knowledge would consider enhancing the host application by exposing required actions, also chart-related actions, by means of predefined shortcuts. The Board judges that using keyword shortcuts invoking applications in order to perform preprogrammed actions was commonplace in the field of human-computer interaction and formed part of the common general knowledge at the priority date. The shortcuts were employed to expedite operations by reducing input sequences to a keystroke; hence the term "shortcut".

The skilled person would immediately recognise that shortcuts would be most useful for actions that do not require any user input. For such actions, it is also not necessary to notify the user that the spreadsheet application instance is running, as the user is only interested in the result. Since user-friendliness implies avoiding presenting the user with unnecessary information, refraining from providing visual indication of the performed spreadsheet operations results naturally from the objective technical problem.

The appellant argued that the claimed solution went against the OLE framework of D5 and therefore would not have been considered by the skilled person. However, the Board cannot see why using the OLE facilities should hinder the skilled person from adapting the general teaching of D5 in the above manner (cf. remarks concerning the OLE in point 2.1.1 above)

The Board therefore judges that feature A is obvious.

2.1.5 Distinguishing features B and C do not add anything of inventive merit. They are motivated by the business requirement that the chart should visualise data displayed next to it and reflect changes to this data. Using the COMVIK approach, this business requirement is provided to the technically skilled person as part of the framework of the objective technical problem. The Board considers that it would be obvious for the skilled person, facing the problem of implementing this business requirement, to determine each time after the user edits the spreadsheet data set whether changes introduced by the user to the data set necessitate a chart update. If so, the chart would have to be updated accordingly. Hence, features B and C are obvious.

2.1.6 Accordingly, the subject-matter of claim 1 lacks inventive step.

2.2 Main request, claim 1

Since claim 1 of the first auxiliary request contains all features of claim 1 of the main request, the above objection under Article 56 EPC applies also to claim 1 of the main request.

Hence, claim 1 of the main request also lacks an inventive step

3. Third auxiliary request, claim 1.

The Board finds it convenient to discuss the more specific third auxiliary request before the second auxiliary request.

3.1 New features

Claim 1 of the third auxiliary requests adds to claim 1 of the first auxiliary request that:
D) *The spreadsheet application determines both the values and format to be used for the chart.*
E) A *presentation application (and not the host application) displays the chart.*
F) A *presentation file includes a chart file directing the presentation application to a spreadsheet file that includes chart data for the chart, the chart data including the values to be used for the chart.*
G) *The spreadsheet file is stored either as a spreadsheet file embedded within the presentation file or as a spreadsheet file external to the presentation*

file.

3.2 Inventive step - Article 56 EPC

3.2.1 D5 discloses implicitly that the spreadsheet application which created the displayed chart determines, upon the chart creation, the value and format for the chart. Thus, D5 discloses feature D.

Concerning feature E, the appellant argued during the oral proceedings that the description disclosed at page 5, lines 12 and 17 that the presentation application was not a word processor application but rather the PowerPoint application. Therefore, the claim should be read in the light of this disclosure. Thus, claim 1 differs from D5 by the provision of the "*presentation application*" instead of the word processor. However, the Board considers that the general expression "presentation application" encompasses any application presenting data to the user including a word processor application. Thus, feature E is disclosed in D5.

Hence, the subject-matter of claim 1 differs from D5 by features A-C, discussed above, and added features F-G.

3.2.2 Features A-C and F-G do not interact synergistically.

As set out above, features A-C are obvious.

Concerning features F-G, the Board judges that they are also obvious. D5 discloses that the embedded spreadsheet object of D5 is stored as an embedded or linked object (column 7, lines 65 to 67). Since, D5 does not provide any disclosure regarding used storage structures, the skilled person starting from D5 inevitably faces the objective technical problem of how to implement a storage structure enabling embedding a spreadsheet object within the word processor document or linking the object to the document. The Board judges that it would be obvious to the skilled person to store the word processor document as a file comprising a pointer to either an external spreadsheet file or to an embedded spreadsheet object. Such a pointer corresponds to the chart file interpreted in the light of the description at page 9, lines 20 to 21 saying that "*the chart file...is more similar to a pointer and directs the presentation application to another file...that includes the chart data*".

Furthermore, basing the chart on data included in the embedded spreadsheet object or external spreadsheet file, such that the embedded object or the file provides data for the chart, is an obvious implementation of the business requirement that the chart should visualise data displayed next to it. See comments concerning obviousness of features B and C in point 2.1.5 above.

The appellant argued that the claimed solution could not be implemented in the OLE-based framework of D5. OLE objects of D5 were something different than files and D5 taught away from the claimed solution. The Board does not find this argument persuasive for reasons similar to those given for the first auxiliary request in points 2.1.3 and 2.1.4 above. Here again, the Board cannot see why using the OLE in the preferred embodiment should hinder the skilled person starting from D5 from implementing the file structure discussed above.

Hence, the Board judges that features F and G are obvious and claim 1 of the third auxiliary request lacks inventive step.

4. Second auxiliary request, claim 1

Although the second auxiliary request relates to the host application instead of the presentation application, this does not add anything to the scope of the third auxiliary request, because the broad expression "host application" encompasses a presentation application. Apart from this difference, claim 1 of the third auxiliary request contains all features of claim 1 of the second auxiliary request.

Hence, the objection under Article 56 EPC raised for the third auxiliary request also applies to claim 1 of the second auxiliary request.

Hence, claim 1 of the second auxiliary request lacks inventive step.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts

Boards of Appeal of the European Patent Office

Chambres de recours de l'Office européen des brevets

T. Buschek W. Chandler

Decision electronically authenticated