BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS

BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE

CHAMBRES DE RECOURS DE L'OFFICE EUROPEEN 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 10 April 2013

Case Number:	T 2217/08 - 3.5.06
Application Number:	04028765.8
Publication Number:	1560112
IPC:	G06F 9/445, G06F 17/30, H04L 12/58, H04L 29/06
Language of the proceedings:	EN

Title of invention:
Detection of files that do not contain executable code

Applicant:
MICROSOFT CORPORATION

Headword:
Executable code/MICROSOFT

Relevant legal provisions:
-

Relevant legal provisions (EPC 1973):
EPC Art. 56

Keyword:
Inventive step - (yes) after amendment

Decisions cited:
T 0154/04

Catchword:
-

Europäisches Patentamt | European Patent Office | Office européen des brevets

Beschwerdekammern | Boards of Appeal | Chambres de recours

Case Number: T 2217/08 - 3.5.06

D E C I S I O N
of the Technical Board of Appeal 3.5.06
of 10 April 2013

Appellant: (Applicant)	MICROSOFT CORPORATION One Microsoft Way Redmond, WA 90852 (US)
Representative:	Grünecker, Kinkeldey Stockmair & Schwanhäusser Leopoldstrasse 4 D-80802 München (DE)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted 25 June 2008 refusing European patent application No. 04028765.8 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman: D. H. Rees
Members: A. Teale
C. Heath

Summary of Facts and Submissions

I. This is an appeal against the decision by the examining division, dispatched on 25 June 2008, to refuse European patent application No. 04 028 765.8 *inter alia* on the basis that the subject-matter of independent claims 1, 14 and 21 according to the then main request did not involve an inventive step, Article 56 EPC 1973, in view of document D1 combined with either common general knowledge or D5. In addition, the subject-matter of claims 1 and 14 lacked inventive step in view of D1 combined with D7 and that of claim 21 lacked inventive step in view of D1 combined with D7 and D8. These documents are as follows:

D1: WO 02/05072 A2

D5: Downs A.S., "HEADERDOC: Resend: Re: Bug 2479261 - should support ObjC tags", Apple Mailing Lists, Internet disclosure, [Online], 17 January 2001, XP002329241, retrieved from the Internet: URL: http://lists.apple.com/archives/headerdoc-development/2001/Jan/msg00024 html> [retrieved on 2005-05-24].

D7: US 2002/0078216 A1

D8: Perkins C.L. and Lemay L., "Teach Yourself Java in 21 Days, Professional Reference Edition", 1996, Sams.net Publishing, ISBN: 1-57521-183-1, Chapter "Day 19 - Streams and I/O", pages 469 to 496.

Amended claims according to two auxiliary requests submitted in the oral proceedings before the examining division were not admitted, as *prima facie* they also set out subject-matter not involving an inventive step.

II. In a notice of appeal, received on 5 September 2008, the appellant requested that the appealed decision be set aside and that a European patent be granted on the basis of the claims, description and drawings on file. The appellant also made an auxiliary request for oral proceedings. The appeal fee was paid on the same day.

III. With a statement of grounds of appeal, received on 5 November 2008, the appellant filed amended claims according to a main request and requested that the appealed decision be set aside and that a decision be taken on the allowability of the main request.

IV. In an annex to a summons to oral proceedings the board set out its preliminary opinion on the appeal. The board expressed doubts *inter alia* as to the technical character, Article 52 EPC, of the claimed subject-matter. The claimed subject-matter did not seem to have any "further technical effect" going beyond those resulting from the running of any computer program. The board consequently expressed doubts as to the inventive step of the subject-matter of claim 1, Article 56 EPC 1973, since the only feature which seemed to be non-obvious starting from D1 seemed to lack technical character. Although it was not yet claimed, the board pointed out that the application disclosed only storing files found to be free of executable code; see page 10, lines 20 to 22. The board tended to consider this a "further technical effect".

V. With a letter received on 8 March 2013 the appellant submitted amended claims according to a main and first and second auxiliary requests.

VI. At the oral proceedings on 10 April 2013 the appellant withdrew the main and first auxiliary requests, so that the second auxiliary request comprising three claims became the main request. The appellant's final request was that the decision under appeal be set aside and that the case be remitted to the first instance with the order to grant a patent based on the main request (previous second auxiliary request) and a description to be adapted thereto.

VII. Claim 1 of the main request (previous second auxiliary request) reads as follows:

"A computer-implemented method of detecting code-free files, comprising: parsing (702) an input file with a compound parser configured to include a plurality of component parsers wherein each component parser is configured to recognize a specific file format, wherein the compound parser is configured to allow extension by addition of a new component parser to the compound parser, wherein the new component parser recognizes a further file format and recognizes executable code within the further file format, wherein said parsing comprises the step of parsing said input file with all component parsers in order to check the input file for the presence of all specific file formats recognizable by said plurality of component parsers, and wherein said parsing continues (716) even in case that a particular component parser has already recognized said file format; determining (704) if the file format was recognized; analyzing (706) contents of the input file with each component parser (506) to detect executable code within the input file according to the recognized file format; sending (710) from the component parser to a controller (504) a file-has-no-code status when the component parser recognizes the file format and no executable code was found; sending (712) from the component parser to the controller (504) a file-has-code status when the component parser recognizes the file format and executable code was found; and storing the input file only if no executable code has been found in the input file."

The claims according to the main request also include an independent claim 3 setting out a processor readable medium comprising processor-executable instructions for performing a method according to claims 1 and 2.

VIII. The description and figures currently on file are as follows:

Description:
Pages 2 to 15 as originally filed.
Pages 1, 1a, 1b and 1c, received on 12 June 2006.

Figures:
1 to 6 as originally filed.

IX. At the end of the oral proceedings the board announced its decision.

Reasons for the decision

1. *The admissibility of the appeal*

In view of the facts set out at points I to III above, the appeal fulfils the admissibility criteria under the EPC and is therefore admissible.

2. *The amendments to the application*

2.1 Claim 1 is based on the combination of original claims 14, 15 and 29 and features taken from the original description and figures. In particular, the feature of continuing to parse the input file with all component parsers even if the file format has already been recognised by one of the component parsers is based on original figure 7, step 716, and page 10, lines 6 to 10. The feature of storing the input file only if no executable code has been found in it is based on page 10, lines 20 to 22, as originally filed.

2.2 Claims 2 and 3 are based on claims 16 and 1, respectively, as originally filed.

2.3 Editorial amendments aside, the description has been amended to acknowledge prior art in compliance with Rule 27(1)(b) EPC 1973. See however point 5 below regarding adaption of the description to the claims.

2.4 Consequently the board finds that the amendments to the application satisfy Article 123(2) EPC regarding added subject-matter.

3. *The prior art*

3.1 *Document D1*

3.1.1 D1 concerns the processing of e-mails to detect new virus outbreaks, a virus being defined as any software having undesired effects; page 1, lines 13 to 14. E-mails are decomposed and analyzed; see "decomposer/analyzer" 21 in figure 2 and page 5, lines 10 to 11. Page 9, lines 25 to 27, discloses decomposition into e-mail and mime headers, a message component and an attachment component. In the analysis step each component is subjected to several checks, including checking the message and attachment components for executable code; see page 9, line 30, to page 10, line 17. In particular, an attached Word document or ZIP file is checked for executable code; see page 10, lines 5 to 9. If executable code is found then the e-mail is logged in a database which is subsequently scanned for traffic patterns indicating a virus outbreak.

3.1.2 Hence the system known from D1 comprises a plurality of parsers for analyzing the various types of attachments and, contrary to the appellant's argument, if the format of the attachment is recognized, the attachment is searched for executable code. As the format of the attachment is not known in advance, it is implicit in D1 that several parsers must be tried to identify the attachment format, for instance Word or ZIP. The appellant has argued that the plurality of parsers known from D1 cannot be considered as the claimed "compound parser". The board disagrees; the individual parsers in D1 can be regarded as the claimed "component parsers", and the application does not provide any details, nor were any identified in the appealed decision, to support the argument that a plurality of parsers does not fall under the term "compound parser". It follows that the board also does not accept the appellant's argument that D1 does not disclose "parsing before analysis". It is implicit in D1 that the result of the search for executable code can be a "file-has-no-code" or a "file-has-code" status. The appellant has argued that the claimed "file-has-no-code" statement is a more definite statement than the one produced in D1, namely that it is *possible* that an e-mail contains a virus; see page 3, lines 1 to 3. The board however finds that there is no technical difference between the application and D1 in this regard. In both cases executable code is searched for, no distinction being made between benevolent executable code and malware. It is also implicit in D1 that the component parsers send their "file-has-no-code" and "file-has-code" status information to a "controller" of some sort, the appellant not having presented any arguments to the contrary.

3.1.3 In terms of claim 1, D1 discloses a computer-implemented method of detecting code-free files, comprising parsing an input file with a compound parser configured to include a plurality of component parsers wherein each component parser is configured to recognize a specific file format, determining if the file format was recognized and, if so, analyzing contents of the input file with each component parser to detect executable code within the input file according to the recognized file format, indicating a file-has-no-code status when the component parser recognizes the file format and no executable code was found and indicating a file-has-code status when the component parser recognizes the file format and executable code was found.

3.2 *Document D5*

D5 discusses using plug-in parser modules to determine the language (such as C++) of an input file; see page 1, lines 16 to 7 from the bottom. The addition of a new language is also mentioned; see page 1, lines 20 to 15 from bottom. There is however no mention of always using all parser modules. On the contrary, it is explicitly stated that parsing stops after the first success: "... we could call each parser for that language in succession <u>until one succeeds, then stop</u> processing that header" (emphasis by the board); see page 2, lines 12 to 14.

3.3 *Document D7*

D7 relates to the processing of data records having multiple formats, each format being parsed by a corresponding plug-in module; see paragraph [0005]. There is no mention of using the plug-in modules to recognize the format of data records. Instead the plug-in modules convert data records in the various different input formats to a common standard format; see paragraph [0005]. D7 mentions creating a new plug-in module to extend the system to a further input data format; see paragraph [0012], lines 7 to 11.

3.4 *Document D8*

D8 mentions using a sequence of parsers to identify the type of a Java input stream and states "let <u>each</u> parser run until it either throws an error or completes a successful parse. If an error is thrown, use reset() and try the next parser." (emphasis by the board); see page 474, lines 26 to 32. According to the appealed decision, D8 discloses using all parsers in all cases, continuing even if a parser has already recognized the stream type. The board takes a different view and interprets the cited passage as merely stating that the parsers are tried in sequence, the result (either error or success) of one parser being waited for before starting the next parser. It is not directly and unambiguously derivable from D8 that the sequence of parsers continues even if a one of them has already recognized the stream type.

4. *Inventive step, Article 56 EPC 1973*

4.1 The method according to claim 1 differs from the disclosure of D1 in the following features:

a. parsing comprises the step of parsing said input file with all the component parsers, said parsing continuing even if a component parser has already recognized said file format;

b. the compound parser is configured to allow extension by addition of a new component parser to the compound parser, the new component parser recognizing a further file format and recognizing executable code within the further file format and

c. the input file is stored only if no executable code has been found.

4.2 Difference feature "a" was set out in claim 1 of the auxiliary requests not admitted by the examining division into the proceedings. In the appealed decision the examining division nevertheless commented on the inventive step of the subject-matter set out in these claims, stating that, on the one hand, parsing the input file with all component parsers even when one component parser had already recognised the format of the input file and, on the other hand, stopping parsing as soon as one component parser recognised the format of the input file were two obvious alternatives. The former was more complete, as it would detect all file formats in an input file, thereby allowing an exhaustive analysis of the input file. The latter was more efficient, as it avoided additional parsing attempts after a file format had been recognized. In the context of anti-viral software, the skilled person would have chosen the former alternative, since it was the more complete and therefore more secure alternative.

4.3 The appellant has disputed this argument on the basis that it was not known at the priority date that an input file might comply with more than one of the formats recognized by the parsers. The board agrees with the appellant on this point. None of the prior art documents on file, in particular D5 and D8, which were relied upon by the examining division in the appealed decision, mentions or even hints at the possibility that an input file might comply with more than one of the formats recognized by the parsers, requiring that all parsers be tried in all cases. Not knowing that an input file might comply with more than one of the formats recognized by the parsers, the skilled person would have had no reason to add feature "a", since it would have been contrary to the usual principle of optimizing computing efficiency and there would have been no expectation of a benefit accruing from continuing parsing attempts once a format had been recognized in the input file by one of the component parsers. Hence difference feature "a" would not have been obvious to the skilled person starting from D1 and taking into account the other prior art documents on file.

4.4 An issue debated in these appeal proceedings has been whether difference feature "a" has technical character and can thus contribute to inventive step; see, for example, T 0154/04 ("DUNS", OJ EPO 2008, 046), reasons, point 5(F). In the annex to the summons to oral proceedings the board took the view that parsing *per se* did not necessarily rely on technical considerations or have technical effects and thus expressed doubts as to whether feature "a" had technical character. In response the appellant has now restricted claim 1 by adding the feature "c" that the input file is only stored if no executable code has been found in the input file. The storage of the input file is a technical step which has a further technical effect and thus lends technical character to the parsing steps leading to it. Hence feature "a" can contribute to inventive step.

4.5 Since feature "a" lends inventive step to the subject-matter of claim 1, there is no need, for the purposes of this decision, to go into the question of the obviousness of difference features "b" and "c".

4.6 The board finds that the subject-matter of claim 1 involves an inventive step, Article 56 EPC 1973.

4.7 It follows that the subject-matter of independent claim 3, which sets out a processor readable medium comprising processor-executable instructions for performing a method according to *inter alia* claim 1, likewise involves an inventive step, Article 56 EPC 1973.

5. *The description*

The board points out that the description, in particular paragraph [0030] on page 10, requires adaption to the claims in order to satisfy Rule 27(1)(c) EPC 1973, since it still states that difference feature "a" is only an optional feature of the invention.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the first instance with the order to grant a patent based on the main request (previous second auxiliary request) and a description to be adapted thereto.

The Registrar: The Chairman:

B. Atienza Vivancos D. H. Rees