**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) [ X ] To Chairmen
(D) [ - ] No distribution

## Datasheet for the decision of 4 October 2017

| | |
|---|---|
| **Case Number:** | T 2324/14 - 3.5.06 |
| **Application Number:** | 12007773.0 |
| **Publication Number:** | 2610745 |
| **IPC:** | G06F9/45, G06F9/44 |
| **Language of the proceedings:** | EN |

**Title of invention:**
Optimizing source code

**Applicant:**
SAP SE

**Headword:**
Optimizing source code/SAP

**Relevant legal provisions:**
EPC Art. 84, 114(2)
EPC R. 43(7), 116, 137(3)
RPBA Art. 12(1)(a), 12(4), 13(1), 13(3)

**Keyword:**
Non admission of auxiliary request by the examining division - full reasoning is incorrect prima facie assessment
Late filed auxiliary request - admitted and considered
Claims - clarity, all requests (no)

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

**Decisions cited:**
G 0007/93

**Catchword:**
See points 2-2.6

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

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

**Case Number: T 2324/14 - 3.5.06**

# DECISION
# of Technical Board of Appeal 3.5.06
# of 4 October 2017

**Appellant:** SAP SE
(Applicant)
Dietmar-Hopp-Allee 16
69190 Walldorf (DE)

**Representative:** Müller-Boré & Partner
Patentanwälte PartG mbB
Friedenheimer Brücke 21
80639 München (DE)

**Decision under appeal:** **Decision of the Examining Division of the European Patent Office posted on 17 July 2014 refusing European patent application No. 12007773.0 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 lies against the decision of the examining division, with reasons dated 17 July 2014, to refuse European patent application No. 12 007 773.0 for lack of clarity and lack of sufficient disclosure, Articles 83 and 84 EPC. An auxiliary request was not admitted under Rule 137(3) EPC because *prima facie* it did not overcome the deficiencies of the main request. No prior art is relied upon in the decision.

II. A notice of appeal was filed on 17 September 2014, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 27 November 2014. The appellant requested that the decision be set aside and that a patent be granted on the basis of claims 1-3 according to a main or an auxiliary request as subject to the decision and as re-filed with the grounds of appeal.

III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed invention lacked clarity, Article 84 EPC. Objections under Article 83 EPC and Article 56 EPC were also raised.

IV. In response to the summons, with a letter dated 31 August 2017, the appellant filed amended claims according to a new, second auxiliary request.

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

"A computer-implemented method for optimizing code, the method comprising:

identifying a decision table (120; 200; 500) comprising a number of input fields (205a-c) with values in column format and a result field (210), wherein the values for the input fields (205a-c) are stored in rows (215a-f), the values defining business rules;

evaluating the decision table (120; 200; 500) to generate one or more tables (125; 300; 325, 350) which are generated and stored temporarily comprising:

generating a first table (300) comprising a field ID column (305) generated by reading the values in the decision table (120; 200; 500) and by determining a maximum number of possible values for each of the input fields (205a-c) and the result field (210) in the decision table (120; 200; 500), wherein a row (320a-h) is generated for each of the possible values for each of the input fields (205a-c) and the result field (210), and a row 10 column (315) which identifies the particular rows (215a-f) in the decision table (120; 200; 500) in which the corresponding values of the input fields (205a-c) and the result field (201) can be found, wherein the possible values for each of the input fields (205a-c) and the result field (210)are stored in a value 10 column (310);

generating a second table (325) from the first table (300) and the decision table (120; 200; 500) comprising one or more unique ID values stored in a value ID column (340), each unique ID value associated in the second table (325) with a particular value defined in the particular row (215a-f) of the decision table (120; 200; 500); and

generating a third table (350) from the first table (300), the second table (325), and the decision table (120; 200; 500), wherein the third table (350) associates the unique ID values of the value ID column (340) of the second table (325) with a corresponding value of the possible values of the decision table (120; 200; 500) as stored in the value ID column (310) of the first table (300) that corresponds to the row/

column combination of the second table (325) that is unique and includes a cell buffer column (370), which indicates, for each of the values, whether a particular position in a string has a pointer already been set in the (x) position of the string to the value shown in the value ID column (310);

looping over the second and third tables (325, 350) to generate code (600) implementing the business rules stored in the decision table (120; 200; 500) by generating a string variable with positions assigned by the cell buffer column (370); and

deleting the tables (125; 300; 325, 350) once the code (600) is generated."

Claim 1 of the first auxiliary request differs from claim 1 of the main request in that the clause explaining the cell buffer column (370) now reads as follows:

"... which associates each of the unique ID values stored in the second temporary table with a particular position of the positions of the string variable ...".

Claim 1 of the second auxiliary request reads as follows (amendments to claim 1 of the first auxiliary request are marked):

"A computer-implemented method for ~~optimizing~~ generating optimized code, the method comprising:

identifying a decision table (120; 200; 500) comprising a number of input fields (205a-c) with values in column format and a result field (210), wherein the values for the input fields (205a-c) are stored in rows (215a-f) and~~,~~ the values defining ~~business~~ a set of rules, each row representing a rule implemented in an if/then conditional statement, wherein the input fields are compared against the values in the rows, the values being alphanumeric or Boolean values and including ranges;

evaluating the decision table (120; 200; 500) to generate one or more tables (125; 300; 325, 350) which are generated and stored temporarily comprising:

generating a first table (300) comprising a field ID column (305) generated by reading the values in the decision table (120; 200; 500) and by determining a maximum number of possible values for each of the input fields (205a-c) and the result field (210) in the decision table (120; 200; 500), wherein a row (320a-h) is generated for each of the possible values for each of the input fields (205a-c) and the result field (210), and a row 10 column (315) which identifies the particular rows (215a-f) in the decision table (120; 200; 500) in which the corresponding values of the input fields (205a-c) and the result field (201) can be found, wherein the possible values for each of the input fields (205a-c) and the result field (210)are stored in a value 10 column (310);

generating a second table (325) from the first table (300) and the decision table (120; 200; 500), wherein the second table (325) comprises~~ing one or more unique ID values stored in~~ a value ID column (340) with one or more unique value IDs stored therein, each unique ~~ID~~ value ID being a representative value for each particular cell of the decision table (120; 200; 500) not found previously in the decision table (120; 200; 500), and a column ID column (335) that associated to each unique value ID the field of the decision table (120; 200; 500) corresponding to the particular cell ~~associated in the second table (325) with a particular value defined in the particular row (215a-f) of the decision table (120; 200; 500)~~; and generating a third table (350) from the first table (300), the second table (325), and the decision table (120; 200; 500), wherein the third table (350) associates the unique ~~ID~~ value~~s~~ IDs of the value ID column (340) of the second table (325) with a corresponding value of the possible values of the decision table (120; 200; 500) as stored in the value ~~ID~~ column (310) of the first table (300) that corresponds to the ~~row/column combination of the second table (325) that is unique~~ particular cell of the decision table (120; 200; 500) and includes a cell buffer column (370), which associates each of the unique ~~ID~~ value~~s~~ IDs stored in the second temporary table (325) with a particular position of the positions of ~~the~~ a string variable;

looping over the second and third tables (325, 350) to generate code (600) implementing the ~~business~~ set of rules stored in the decision table (120; 200; 500) by generating ~~a~~ the string variable with positions assigned by the cell buffer column (370);

wherein looping comprises:

creating if statements for each value ID in the third table (350) corresponding to an input field, wherein the condition for the if statement is that the input field as identified in the column ID column (335) of the second table (325) is equal to the associated corresponding value of the possible values of the decision table (120; 200; 500) the the third table (350) and wherein the associated string variable position is set to true if the condition is true; and

creating if statements based on the cell buffer positions for setting the result field;

and wherein the code is optimized in that evaluations for setting the positions of the string variable need not be performed more than once; and deleting the tables (125; 300; 325, 350) once the code (600) is generated."

VI. Oral proceedings were held on 4 October 2017, at the end of which the chairman announced the board's decision.

## Reasons for the Decision

*The decision of the examining division not to admit the auxiliary request*

1. The decision under appeal objected that claim 1 of the main request lacked clarity and sufficient disclosure (see reasons 1-1.1.2 and 2-2.4.2). It then observed that claim 1 of the (first) auxiliary request still contained at least some of the expressions found to be unclear (reasons 3.1) and did not remedy "in any way" the lack of disclosure (reasons 3.2). In view of this, the examining division found that the auxiliary request "prima facie [... did] not overcome the objections under Article 84 EPC and Article 83 EPC", and thus it did not give its consent under Rule 137(3) EPC to the auxiliary request (see reasons 3).

2. Under Rule 137(3) EPC, "No further amendment may be made without the consent of the Examining Division". The examining division therefore had discretion to deny its consent to the auxiliary request.

2.1 Like any other decision open to appeal, the discretionary decision to deny consent to an amendment must be reasoned (Rule 111(2) EPC). In giving reasons for its decision, the examining division acted correctly in this respect, too.

2.2 The EPC does not define what it means for an examining division to give or deny its consent to an amendment under Rule 137(3) EPC.

2.2.1 Conventionally, giving "consent" to an amendment under this provision is equated to "admitting" it. However, the EPC also does not specify what it means for a submission to be "admitted". Admission is mentioned: in Rule 116 EPC, which provides that "New facts and evidence presented [late] need not be *considered*, unless *admitted* [...]"; in Article 12(4) RPBA, which states that "Without prejudice to the power of the Board to *hold inadmissible* facts, evidence or requests which [...] were *not admitted* in the first instance proceedings, everything presented by the parties under (1) shall be *taken into account* by the Board [...]"; in Article 13(1) RPBA, which states that "Any amendment to a party's case [...] may be *admitted and considered* at the Board's discretion"; and in Article 13(3) RPBA, which provides that "Amendments [...] shall not be *admitted* if they raise issues which the Board or the other party or parties cannot reasonably be expected to *deal with* [...]". Without express mention of admission, Article 114(2) states that "The European Patent Office may *disregard* facts or evidence [...]" (all emphasis added by the board).

2.2.2 Although these regulations use slightly different words, the board takes the view that they all refer to the same procedural step. A deciding body giving "consent" to a submission "admits" it or "holds it admissible" and cannot "disregard" it, but must, equivalently, "consider" it, "take it into account" and "deal with" it. It follows that a submission which is admitted must be considered fully, whereas a submission which is not admitted must be considered to the extent necessary to justify the non-admission decision, but in other respects its consideration may be limited.

2.3 The EPC also does not define how the examining division should exercise its discretion under Rule 137(3) EPC. The boards of appeal have, however, accepted that the examining division may base its decision to deny consent to an amendment on *prima facie* considerations and that it may deny its consent to an amendment with *prima facie* deficiencies.

2.4 A *prima facie* judgment is one which is made at first sight and is commonly understood to be one assumed to be correct until proven otherwise. Therefore, in making a *prima facie* judgment, the deciding body concedes that the judgment may not survive further scrutiny and may turn out to be incorrect on further investigation.

2.5 In the present case, the examining division stated that the auxiliary request "prima facie [... did] not overcome the objections" raised against the main request. However, it then indicated which specific objections were not remedied, and the decision leaves no doubt that the examining division found the auxiliary request to be deficient under Articles 83 and 84 EPC with the same degree of conviction as the main request.

2.6 Therefore, the board considers that, contrary to the wording of the decision, the examining division did not limit its examination of the auxiliary request to *prima facie* considerations. It concludes that the discretionary decision was incorrect even though it was based on right principles (see G 7/93, reasons 2.6). In fact, the examining division considered the auxiliary request fully, since it was able to give sufficient reasons for its conclusion that a patent could not be granted on the basis of the auxiliary request. The examining division thus having considered the auxiliary request fully, the board takes the view that there was no discretion left for the examining division "not to admit" it.

3. As a rule, "everything" presented by the appellant in its statement of grounds of appeal "shall be taken into account by the Board if and to the extent it relates to the case under appeal and meets the requirements in [paragraph] (2)" (see Article 12(4) and 12(1)(a) RPBA). As an exception, Article 12(4) RPBA refers to the board's discretion "to hold inadmissible facts, evidence or requests which [...] were not admitted in the first instance proceedings".

3.1 The board considers that its discretion in this regard requires the non-admission decision of the first instance to be correct.

3.2 This not being the case, the board considers that Article 12(4) RPBA obliges it to take into account the first auxiliary request.

*Admission of the second auxiliary request*

4. Under Article 13(1) RPBA the board has discretion to "admit[] and consider[]" the second auxiliary request, which it is to "exercise[] in view of inter alia the complexity of the new subject matter submitted" and "the need for procedural economy". Since the second

auxiliary request is a reasonable attempt to address the board's clarity objections and does not substantially complicate the case, the second auxiliary request will be considered in this decision.

*The invention*

5. The application relates to the generation of small and fast (source) code for so-called "decision tables". In comparison with "conventional code generation techniques", this is referred to as "optimizing source code" (see paragraphs 2, 3, 17 and 50).

5.1 Decision tables are not defined in the application but are illustrated by way of example (see figures 2 and 5, and paragraphs 46 *et seq.*). There is a column for each input (field) and for the result (field), and it is explained that each row of the decision table implements a "business rule" which defines the value of the result (field) on condition that the inputs (input fields) have the indicated value (see paragraph 49). It is stated that "values may be alphanumeric or Boolean values, including ranges or expressions, as well as" various "comparisons" or "determination[s]" (see paragraphs 47 and 50).

5.2 Code generation too is explained by way of example only. It is disclosed that the decision table of figure 2 is meant to be mapped to the code depicted in figure 4. It is stated and can be seen that the code avoids redundant evaluations of conditions (see paragraphs 50 and 61-64, especially the last sentence of paragraph 62).

5.3 The description explains that the code generation process is aided by three tables (figures 3A-3C) generated from a given decision table. In a nutshell, the first table (figure 3A) lists, for each combination of field and "value" entry in the decision table, the rows in which the decision table contains that combination. The second table (figure 3B) contains a row for each cell in the decision table, ordered left to right and top to bottom, where the "values" in the decision table are replaced by identifiers ("Value ID"); multiple occurrences of the same "value" for a given field in the decision table will receive the same identifier (e.g. the "1" in column 340 of rows 345a and 345e represents the "value" "=A" in the "field 1" column of the decision table). The third table (figure 3C) lists all possible "values" to be checked in the decision table (see column 360) and associates them with a unique position in a "cell buffer". Although only values are listed, it is understood that each cell buffer position is meant to represent an individual *condition* (such as field 1 =A) rather than just a *value* ("A").

5.4 In the generated code, such conditions will each be evaluated at most once, in the order in which they occur in the table of figure 2. If a condition holds, the corresponding cell buffer position will be set to "x". As soon as all conditions for a particular rule in the decision table are evaluated, it is tested whether the corresponding results in the cell buffer have all been set to "x", in which case the rule determines the result value.

*Main and first auxiliary request*

6. Contrary to what the claims specify, the invention is not about "optimizing code" - in the sense of transforming suboptimal code into optimal code - but about directly producing code which is meant to be "optimal".

7. More importantly, though, the term "optimal" on its own is unclear. *A priori*, code may be optimal in several ways, and different optimality criteria may contradict each other. Claim 1 of the main and the auxiliary requests does not define "optimality" explicitly (nor does the description), but it also does not define the code being generated in a way that allows the skilled person to deduce in which way it is meant to be "optimal", let alone establish that it actually is reliably and reproducibly "optimal" in this sense. In fact, claim 1 of these two requests does not specify the generated code at all, requiring only that it be generated by "looping over the second and third tables" (see lines 2-4 from the end of claim 1).

8. The claims also leave open how the claimed iteration ("looping over") is meant to use the second and third tables and the cell buffer to generate code.

9. The board thus concludes that claim 1 of these two requests is unclear, Article 84 EPC, because it fails to define the code to be generated, the central property of the code to be generated ("optimality") and the steps necessary to produce that code, even though the property is expressly claimed and the method is claimed as being for generating optimized code.

*Second auxiliary request*

10. As the preamble of claim 1 of the second auxiliary request refers to a "method for generating optimized code" rather than to "optimizing code" and now states in what way the generated code is meant to be "optimized", the objections raised under points 6 and 7 above do not apply to claim 1 of the second auxiliary request.

11. However, neither the claims nor the description define the decision tables being considered.

11.1 There are merely a few examples (see paragraph 50) defining how the values in the input fields determine the value of the result. It can be deduced from the description that every column in the decision tables of figures 2 and 5 corresponds to a "variable" and that every row in the decision tables defines the result as having a particular value if all variables have the values indicated in the corresponding fields (see e.g. page 13, last 4 lines). However, the general form of decision tables and which "sets of rules" they define are not expressly and exhaustively defined in the application.

11.2 The entries in the decision tables are referred to as "values" throughout the description, although it is disclosed that "values" may also be ranges (e.g. "1..100"), "comparisons" or "determination[s]". The board therefore considers that the skilled person would understand that the "values" in the decision tables represent "conditions" on the fields.

11.3 Nonetheless, the claim language is inconsistent in this regard, and thus unclear, as it specifies "values being alphanumeric or Boolean values and including ranges" but only one form of condition, namely whether an input field "is equal to the associated corresponding value" (see claim 1, page 2, line 17).

11.4 The appellant argued that the term decision table was common in the art and that, hence, their general form and what rules they represented were clear. This was clear, *inter alia*, from the prior-art documents cited in the search report. The appellant further argued that linguistic impreciseness and ambiguities in this respect were immaterial for the claims because the claimed method would work in the same manner regardless.

11.5 The board disagrees with the appellant at least on point 11.3 above. Otherwise, the board tentatively (and to the appellant's benefit) accepts the appellant's perspective.

12. As regards the first table, claim 1 refers to "a maximum number of possible values for each of the input fields" and states that a row "is generated for each of the possible values for each of the input fields".

12.1 In the board's view, the skilled person would understand the "possible values" of an input field to refer to all values an input field could possibly take. For instance, field 3 might represent a variable with possible integer values between 0 and 1000 (even though only ranges "1..100" and ">100" may be relevant for the "business decision" to be taken). If, in this situation, each value was "possible", the "maximum number of possible values" for field 3 would be 1001.

12.2 The board notes that this interpretation is in conflict with figure 3A and concludes that claim 1 is in conflict with the application as a whole and thus unclear.

12.3 The appellant argued (see letter of 31 August 2017, page 8, penultimate paragraph) that the "possible values" rather referred to the different values occurring in the decision table for a particular field (for instance, in table 2, there are two possible "values" for field 1: "=A" and "=B") and that the term "maximum number" was redundant ("pleonastic"). The appellant also argued that the skilled person would dismiss the interpretation proposed by the board as unreasonable: for instance, according to that interpretation, the first table would have to have an infinite number of rows for a real-valued field, which was impossible to implement.

12.4 The board agrees that the skilled person studying figure 3A would probably be able to deduce from the application as a whole that the first table was meant to contain what the appellant stated it did.

12.5 The board does not agree, however, that this makes the claim clear. It appreciates that claim 1 refers by way of reference numbers to the first table depicted in figure 3B. According to Rule 43(7) EPC, however, such reference numbers shall not be construed as limiting the claim. The board disagrees with the appellant that its interpretation proposed under point 12.1 was obviously incorrect. Firstly, the claims (or the description) do not specify real-valued fields, and secondly, computers cannot represent truly infinite ranges of numbers anyway, be it natural numbers, integers or reals. All such mathematically infinite ranges are approximated by finite ranges in the computer (e.g. finitely many floating point numbers are used instead of infinitely many reals). In summary, no reason is apparent to the board why the skilled person should have to interpret a claimed phrase contrary to its literal meaning.

13. Claim 1 specifies that the second table contains "one or more unique value IDs", "each" of them "being a representative value for each particular cell of the decision table not found previously in the decision table".

13.1 The board regards this sentence as being grammatically unclear. Moreover, the phrase refers to some temporal order ("found previously") which is not defined earlier in the claim. The description contains the same sentence (paragraph 67, page 17, lines 8-12) without any context that would make it clearer.

13.2 Claim 1 further specifies that the second table is generated not only from the decision table but also "from the first table". In what claim 1 then specifies about the second table, no reference is made to the first table at all. The board concludes that claim 1 leaves open how the second table is generated "from the first table".

13.3 The board considers that both deficiencies render claim 1 unclear.

14. The appellant argues that the skilled person would understand from the application as a whole what the second table contained and how it was computed.

14.1 The board agrees that the skilled person would probably be able to understand the structure of the table depicted in figure 3B - even though the application is very difficult to read and the skilled person has to invest substantially more effort than usual to come to that understanding. He would, for instance, notice that the indices in columns 330 and 335 correspond to a left-to-right, top-to-bottom traversal of the decision table, and he would probably also understand the manner in which the indices in column 340 mapped reoccurring "values" in the decision table to a single "value ID".

14.2 In the board's judgment, however, this is insufficient to make clear the definition of the second table in claim 1 (see also point 12.5).

15. Moreover, the board was unable to find (on its own or with the appellant's help) any disclosure in the application that would explain how the second table was meant to be generated from the first.

15.1 The appellant argued that this omission from the application was not detrimental to the clarity of claim 1, since the skilled person would notice that the first table contained relevant information and would find ways to access this information when generating the second table.

15.2 Even if true, however, this would be insufficient to overcome the clarity objection. The board takes the appellant's argument to be that claim 1, in leaving open how the second table was generated from the first one, is not unclear but merely broad. Noting however that the description does not describe in detail at least one way of carrying out this particular feature of the invention, the board considers that it is not, in its full breadth, supported by the application.

15.3 In summary, the board concludes that the step of generating the "second table [...] from the first table" in claim 1 does not comply with Article 84 EPC.

16. For the record only, the board notes that there seem to be more clarity problems, some of which were discussed during the oral proceedings. Already in view of the ones discussed above, however, the board concludes that claim 1 of the second auxiliary request is unclear, Article 84 EPC.

## 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 ✦

B. Atienza Vivancos W. Sekretaruk

Decision electronically authenticated