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

Datasheet for the decision of 24 November 2011

Case Number:	T 2048/07 - 3.5.06
Application Number:	02021572.9
Publication Number:	1403764
IPC:	G06F 9/45, G06F 9/44
Language of the proceedings:	EN

Title of invention:
Method and computer system for dynamic data type enrichment

Applicant:
SAP AG

Opponent:
-

Headword:
Dynamic data type enrichment/SAP

Relevant legal provisions:
EPC Art. 123(2)

Relevant legal provisions (EPC 1973):
EPC Art. 52, 56

Keyword:
"Further technical effect - no"
"Inventive step - no"

Decisions cited:
G 0003/08, T 1173/97

Catchword:
Lack of a further technical effect - see Reasons 7.

Case Number: T 2048/07 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06
of 24 November 2011

Appellant:
(Applicant)

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

Representative:

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

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 20 July 2007 refusing European patent application No. 02021572.9 pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: D. H. Rees
Members: G. Zucka
W. Sekretaruk

Summary of Facts and Submissions

I. The appeal is against the decision by the examining division dispatched on 20 July 2007 to refuse European patent application 02021572.9 on the basis that the subject-matter of claim 1 in the main and in the auxiliary request was not inventive, Article 56 EPC 1973. The following document was cited in the decision:

D1: J. Kung *et al.*, "A comparison of 4-GL development tools in the area of metadata and dynamic user interfaces", Computer Aided Systems Theory - Eurocast '97, pages 567-579, Springer-Verlag, 1997.

II. A notice of appeal was received on 20 September 2007, the appeal fee being paid on the same day. A statement of the grounds of the appeal was received on 20 November 2007.

III. The appellant's Main Request was that the board "[r]efer to the Examining Division with the provision to issue the Communication under Rule 51(4) EPC [1973]" on the basis of documents mentioned in the grounds of the appeal. A First and Second Auxiliary Request were identical to this Main Request, except for different claims, also filed with the grounds of the appeal.

IV. The board issued a summons to oral proceedings. In an annex to the summons, the board set out its preliminary opinion on the appeal, *viz.* that none of the requests satisfied the requirements of Article 84, 52(2) and 56 EPC.

V. During the oral proceedings, the appellant filed a new request, with amended claims, replacing all previously filed requests.

VI. The appellant requests that the decision under appeal be set aside and that a patent be granted on the basis of claims 1-11 filed during the oral proceedings. The further text on file is: description pages 1, 4-15 as originally filed and pages 2, 2a, 2b, 3 filed with the grounds of the appeal; drawing sheets 1/4-4/4 as originally filed.

Claim 1:

"A computer-implemented method (400) for dynamic data type enrichment comprising the steps:

accessing (415), through an application programming interface, API, (190) by an application program (210), metadata (150), wherein the application program (210) uses at least one variable (201) being an 'integer', 'character', or 'string' and wherein the application program (210) calls through the application programming interface(190) at least one metadata service (191, 192, 193) that relates to the corresponding variable (201); and

enriching the corresponding variable (201) with metadata (150) at runtime when the application program (210) is executed, wherein the metadata (150) is associated with a specific data type (120) defined in a metadata store (220) and the application program (210) provides a mapping (302) between the specific data type (120) and the corresponding variable (201), wherein to specify the specific data type (120), the corresponding metadata (150) includes data type details (151, 152, 153) including a text label, a range of allowed values, constraints or any other data that specifies the specific data type and wherein the data type details (151, 152, 153) are exposed to the application program (210) and its variable (201) through the API (190) in the form of the at least one metadata service (191, 192, 193)"

Claim 6:

"A computer program product comprising instructions that when loaded into a memory of a computer system (900) cause at least one processor of the computer system (900) to execute the steps of anyone of the claims 1 to 5"

Claim 7 is directed to a corresponding computer system.

Claim 11 is a further method claim directed to changing the metadata during runtime:

"A computer-implemented method for changing metadata (150) comprising the steps:

executing an application program (210) that uses: at least one variable (201) being an 'integer', 'character', or 'string';

at least one metadata service (191) to access the metadata (150) in a metadata store (220) for defining how the application program (210) can access metadata (150) at runtime;

a mapping (302) between the at least one variable and a specific data type (120) defined in a metadata store (210), wherein the metadata (150) is associated with the specific data type (120), wherein to specify the specific data type (120), the corresponding metadata (150) includes data type details (151, 152, 153) including a text label, a range of allowed values, constraints or any other data that specifies the specific data type and wherein the data type details (151, 152, 153) are exposed to the application program (210) and its corresponding variable (201) through the API (190) in the form of the at least one metadata service (191, 192, 193);

changing the metadata (150) in the metadata store (220) at runtime of the application program (210); and

using the at least one metadata service (191) in the application program (210) for enriching the the [*sic*] at least one corresponding variable (201) with the changed metadata (150) without restarting the application program (210)."

VII. At the end of the oral proceedings, the chairman announced the board's decision.

Reasons for the decision

1. Reference is made to the transitional provisions in Article 1 of the Decision of the Administrative Council of 28 June 2001 on the transitional provisions under Article 7 of the Act revising the European Patent Convention of 29 November 2000, for the amended and new provisions of the EPC, from which it may be derived which Articles of the EPC 1973 are still applicable to the present application and which Articles of the EPC 2000 shall apply.

2. *Admissibility of the appeal*

In view of the facts set out at points I and II above, the appeal is admissible, since it complies with the EPC formal admissibility requirements.

3. *The claimed invention*

Computer programs contain variables, such as "x", "y", "height" or "weight". In most programming languages, each variable is defined to have a certain "type", for example, "integer", "floating point", "character" or "boolean". Further, in some languages, variables may be defined with constraints specific to the program, or variables may be specified to be of a type which is in turn defined in the program to have specific constraints. Thus, to take a somewhat old-fashioned but clear example, in Pascal the programmer could write:

```
var x: 1..10;
```

to define a variable x of type integer constrained to belong to the "subrange" 1 to 10. Alternatively, one could write:

```
type capital = 'A'..'Z';
var letter: capital;
```

to define a variable "letter" of type character constrained to be an uppercase letter. The latter formulation might be used when several variables with the same constraints are to be defined. The constraints in the above examples are called "metadata", since they are data about the data which the variables "x" and "letter" can represent.

The application is concerned with such metadata, although it goes beyond constraints on variables to any data which may be associated with a class of variables within a program which may be defined as a "type" ("the corresponding metadata (150) includes data type details (151, 152, 153) including a text label, a range of allowed values, constraints or any other data that specifies the specific data type"). It advocates that, at least for some programs, when a type is defined in the program, such metadata should not be embedded as a constant in the program as in the examples above, but rather read into the program when it runs, as input. It further specifies that the mechanism for doing so be an "application programming interface" (API), *i.e.* a collection of standard "library" procedures.

This has some advantages: the metadata, *e.g.* range of permissible values, need not be decided when the body of the code is written; the same code may be used for situations where the metadata is different, but the functionality is the same, *i.e.* the code is more "reusable"; the metadata may even be changed while the program is running - *cf.* claim 11.

4. *Amendments to the appellant's case, Rule 13 RPBA*

The board agrees that the amended claims constitute a serious attempt to remedy the deficiencies that it had noted, without *prima facie* introducing new deficiencies or unjustifiably delaying the procedure. The request is, therefore, admitted into the appeal proceedings, Article 13(1) RPBA.

5. *Amendments, Article 123(2) EPC*

Claim 1

The present independent claim 1 is a combination of the original claims 1-3 and 5-7, with the following additional features, which are listed together with their basis in the original application documents:

- *The method is computer-implemented*

  This is disclosed on page 2, line 25 of the original description.

- *The at least one variable is an 'integer', 'character', or 'string'*

  This is disclosed on page 4, lines 13-14 of the original description.

- *To specify the specific data type, the corresponding metadata includes data type details including a text label, a range of allowed values, constraints or any other data that specifies the specific data type*

  This is disclosed on page 1, lines 12-16 of the original description.

- *The data type details are exposed to the application program and its variable through the API in the form of the at least one metadata service*

This is disclosed on page 7, lines 13-17 of the original description.

The board concludes that the present claim 1 contains no added subject-matter (Article 123(2) EPC).

Claims 6, 7, 11

The present independent claims 6, 7 and 11 contain no added subject matter, for the same reasons as given for claim 1 above.

6. *Exclusion from patentability, Article 52(2) and (3) EPC*

It is open to debate whether some of the claims, in particular claim 6, are excluded from patentability by Article 52(2) and (3) EPC. The question is, however, irrelevant in view of 7 below.

7. *Further technical effect*

7.1 On page 10 of the statement of grounds of appeal (lines 10 to 13), the appellant identifies the effects achieved by the claimed invention thus: "System failures due to an otherwise non-matching data type requirement in the underlying system [are] avoided and thereby robustness, reuse and consistency of (mobile) application programs is improved". However, the board is not convinced, at least insofar as this suggests that the claimed invention makes the computer system more reliable, in the sense of avoiding program or system crashes. A properly written program expecting a ten-digit customer number will not crash if presented with a twenty-digit one, it will simply reject the input. Indeed, if anything, the proposed invention introduces new dangers of system failure. The problems involved in at first accepting an assignment to a variable of a value consisting of twenty digits, and later changing the definition of that variable to specify that it can only have values of ten digits length, are evident.

The Board accepts that reading certain data usually embedded in program source code from a file would make at least that part of the source code more reusable and the program more flexible and portable (as it was formulated in the appellant's submission of 13 April 2007, page 5, lines 5 and 6). However, this is simply an advice on how to write a program, and belongs to the fundamentals of that art. To take an example, if one wanted to write a program to calculate the trajectory of a projectile, one option would be to embed the value of *g* (acceleration due to gravity, on the earth's surface approximately 9.8 ms^{-2}) in the source, either as a number or as a defined constant. The program would, however, not work for trajectories on the moon; the system would "fail" if one tried to use it for such a calculation. To make the program cover all cases, the value of *g* could be made an input variable. It would simply be a programming option that would directly follow from the specification drawn up by a systems analyst: if the specification states that some data may change, the programmer will make it a variable.

7.2 The program, therefore, contains only features that are necessary to ensure its proper functioning, independent of any technical context in which it should be used. As a consequence, the program lacks a "further technical effect", *i.e.* an effect that goes beyond the normal technical effects that would be caused by any program within the computer system on which it runs (T 1173/97, OJ 1999, 609, Reasons 6).

7.3 During the oral proceedings, the representative reformulated the "reliability" argument as an increased availability of the system on which the program is running, since a change of the metadata could be effected without stopping the program. It was argued that this was the required "further technical effect". However, the board considers that this again is simply a matter of how a program is specified. A correct program is precisely as flexible as it is designed to be. The way in which a formal specification is implemented in a computer program is part of the art of computer programming, which by virtue of Art. 52(2)(c) and (3) EPC cannot of itself be adduced as an indication of "technical" activity (see G 3/08, OJ EPO 2011, 10, Reasons 13.5).

7.4 In the course of discussion, the representative was asked if, taking the example trajectory program, he would consider there to be the same "further technical effect" in a program which read in the value of "g" rather than having it embedded in the source code. He said that there would indeed be a technical difference between the programs. The board agrees that if the "increased availability" argument were accepted, this would be the consequence. But the implication of this would be that any program which read in any data at all would show a "further technical effect". The board can not accept this: reading in data is part of the fundamental definition of computing - input, process, output. Thus the board considers that reading in data does not go beyond the normal technical effects that would be caused by any program within the computer system on which it runs, *i.e.* that reading data is not a further technical effect in itself. Thus the "increased availability" argument, at least in the context of this particular claimed invention, equally fails.

7.5 The board notes that the claims concern themselves with "metadata", *i.e.* a particular subset of data in general. However, the arguments given above are not affected by what the nature of the data is, nor does "metadata" have any inherently technical nature, as might be argued in the case of data relating to some technical process.

8. *Inventive step, Article 52(1) and 56 EPC 1973*

Since the program causes no further technical effect, it also solves no technical problem with respect to the prior art of computers in general. As a result, claims 1, 6, 7 and 11 lack an inventive step, according to the established case law of the Boards of Appeal (see e.g. G 3/08 at 10.13).

9. *Conclusion*

For the reasons given in 8 above, the appellant's request is not allowable.

Order

For this reason, it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

B. Atienza Vivancos D. H. Rees