

Internal distribution code:

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

**Datasheet for the decision
of 5 June 2018**

Case Number: T 0841/16 - 3.5.06

Application Number: 08732897.7

Publication Number: 2135160

IPC: G06F9/44, G06N5/04

Language of the proceedings: EN

Title of invention:
EDITING AND COMPILING BUSINESS RULES

Applicant:
Ab Initio Technology LLC

Headword:
Business rule interface/AB INITIO

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - all requests (no)

Decisions cited:

Catchword:



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 0841/16 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 5 June 2018

Appellant: Ab Initio Technology LLC
(Applicant) 201 Spring Street
Lexington, MA 02421 (US)

Representative: Lloyd, Patrick Alexander Desmond
Reddie & Grose LLP
The White Chapel Building
10 Whitechapel High Street
London E1 8QS (GB)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 5 November 2015
refusing European patent application No.
08732897.7 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 is against the decision of the examining division dated 5 November 2015 to refuse European patent application No. 08 732 897 for lack of inventive step over common general knowledge in the art.
- II. Notice of appeal was filed on 7 December 2015, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 15 March 2016. The appellant requested that the decision be set aside and that a patent be granted on the basis of claims 1-24 as filed with the grounds of appeal, in combination with the description and the drawings as originally filed.
- III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed invention lacked inventive step over common general knowledge alone.
- IV. In response to the summons, by letter dated 2 May 2018, the appellant filed claims 1-24 according to a first auxiliary request and claims 1-23 according to second and third auxiliary requests.
- V. Claim 1 of the main request reads as follows

"A method for specifying the behavior of a functional component in a graph-based computation system, including:

providing a user interface of a computer system for creating a table having at least one input column and at least one output column, wherein each input column is associated with an input variable and each output column is associated with an output variable in at least one row of the table,

receiving, over the user interface, one or more conditions on input values in respective input columns, the conditions in the at least one row identifying more than one set of potential values of the input variables, and

receiving, over the user interface, one or more output values in respective output columns,

thereby defining a rule case of a rule specification that applies to a given record to be processed by the graph-based computation system if the data values of that record, for each input column in which the rule case has conditions, meets the conditions, where an output value is generated for the given record based on the one or more output values for the rule case that applies to the given record;

storing a mapping table in the rule specification, the mapping table including a mapping of business names to dataset record names;

generating, using a generator executing on at least one processor of the computer system, a function for transforming data based on the rule specification including the mapping table, and

associating the function with the functional component in the graph-based computation system by storing a transform representing the function in a target location."

Claim 1 of auxiliary request 1 differs from claim 1 of the main request only in the last two steps, which read as follows (insertions underlined, deletions struck through):

"... generating, using a generator executing on at least one processor of the computer system, a transform~~function~~ for transforming data based on the rule specification including the mapping table, and

associating the transform~~function~~ with the functional component in the graph-based computation system by storing the transform ~~representing the function~~ in a target location."

Claim 1 of auxiliary request 2 differs from claim 1 of auxiliary request 1 by the following addition to the "generating" step:

"..., the generating including converting each of a plurality of rule cases in the rule specification to a logical expression to form a plurality of logical expressions, and compiling the plurality of logical expressions into computer-executable code; ..."

By the same addition, claim 1 of auxiliary request 3 differs from claim 1 of the main request.

All requests also contain corresponding independent computer system and computer program claims.

VI. Oral proceedings were held on 5 June 2018 as scheduled. At their end, the chairman announced the board's decision.

Reasons for the Decision

The invention

1. Generally, the application relates to "editing and compiling business rules" (see title) and their implementation and execution in "graph-based computation" (see the description, paragraph 1).

- 1.1 Business rules are a common format for expressing data conversion, data analysis ("making determinations about data") or the generation of "new data based on [...] input data" (see paragraph 27). These operations are also referred to as data "transforms" (see paragraphs 28, 29 and 56 and figure 1A). Typically, business rules are specified in the form of one or more tables (see e.g. figures 2A and 2B), where each row defines an individual rule or "rule case" (see paragraph 28). The columns define input and output variables.
- 1.2 A typical such table comprises a column for each input value and each output value and one row per rule "case". The entries in the input variable columns define conditions for the input variables. If all conditions of a rule case are satisfied, the rule case determines the values of the output variables from the corresponding output variable columns (see e.g. paragraphs 31-33). In a sense, therefore, a table defines a mathematical "function" from input to output variables.
- 1.3 Rules can be entered in one spreadsheet or distributed over several of them (see paragraphs 31 and 37, figures 2A and 2B).
- 1.4 In order to be automatically evaluated, business rules are compiled into a "function for transforming data" (see claim 1 as originally filed), also referred to as a "transform" (see, for instance, figure 1C, no. 156, and the corresponding description beginning with paragraph 29 of the description), using any suitable programming language (see paragraphs 34 and 137-140). The board takes it that the skilled person would understand this to mean that the tabular rule specification is translated ("compiled") into a computer

program (see e.g. paragraphs 137-140), which expresses in different terms the "data transformation" specified by the rules. The function is then "associated with" a "functional component", i.e. a vertex, of a computation graph (see claim 1 and paragraphs 2 and 3).

- 1.5 In the rules, the variables are referred to by user-readable "business names", but they correspond to "physical" or "technical names" in the datasets being processed (see paragraph 39). During compilation, the business names must be mapped to the corresponding technical names (see for instance paragraphs 59, 60, 70, 75, 88 *et seq.*).

A terminological issue

2. All claims refer to a graph-based computation system.
 - 2.1 The board agrees that such systems were known in the art, as stated in paragraph 2 of the description. More specifically, it was known in the art to describe computations in terms of graphs in which the vertices correspond to "components of the computation" - and thus to pieces of program code - and the links to data flows.
 - 2.2 Further limitations on the graph-based computation system are not implied by the claim language. In particular, the claimed subject-matter is not limited either by any feature of the exemplary graph-based computation system mentioned in paragraph 2 or by any feature of a specific graph-based computation system developed or marketed by the appellant.

- 2.3 The board considers that the term "graph-based computation system" is clear but would be construed by the skilled person as outlined under point 2.1 above.

Inventive step, Article 56 EPC

Main request

3. The examining division argued (see the decision, reasons 2.2-2.4) that the claimed invention solved the objective technical problem of adapting "a well-known use of a user interface for receiving data" to data of the claimed tabular form (feature 2) and processing this data by generating a "function" and associating it with a "functional component" in a computation graph (features 3 and 4). This formulation of the objective technical problem relied on the examining division's finding that the claimed "features [...] which describe the data and its processing are considered non-technical". The claimed solution was found to be an obvious instance of "programming by example".
- 3.1 The appellant challenged the decision by arguing that the cited well-known user interface was not "the most promising starting point" towards the invention (see the grounds of appeal, page 3, paragraph 3). It also stressed that the invention "provide[d] a generically applicable method of programming a computer system, the steps of which [were] neither dependent upon nor affected by the nature of the data used to define the transform to be created", which had "discernible technical advantages" and so the objective technical problem formulated by the examining division was incorrect (see page 3, paragraph 4, to page 4, paragraph 2). Finally it took the view that the adaptation of known spreadsheets via "programming by

example" would be neither trivial nor successful (page 4, paragraph 3).

- 3.2 As an alternative, the appellant proposed starting the inventive step assessment from the "status quo in the field of graph-based computation", the objective technical problem being considered as enabling users other than "highly skilled IT professionals" to develop programs for a graph-based computation system (page 4, paragraph 5) or as "simplify[ing] the creation of a transform for non-technical users" (see page 4, paragraph 2), i.e. the problem disclosed in the application itself (see paragraph 28).
- 3.3 The appellant also compares the technical advantages of the invention with those of a "'what-you-see-is-what-you-get' ('WYSIWYG') word processor" and other graphical user interfaces (page 5, paragraphs 4 and 5) and cites the judgment of the UK Court of Appeal in *HTC v Apple* (2013 EWCA Civ 451) in support of its view that these advantages are of a technical nature.
4. Like the appellant, the board finds it unrealistic to assume that a programmer (i.e. the skilled person) would start from a known user interface and search for new applications to be equipped with them. Conversely, programmers would normally look for a suitable user interface to enable users to get a given job done. The board thus also agrees that a well-known user interface is an inappropriate starting point for the assessment of inventive step in the present case.
5. Instead, the board follows the appellant's suggestion of starting from the "status quo in the field of graph-based computation", on the above proviso (see point 2),

however, that "graph-based computation" must be construed broadly.

- 5.1 More specifically, the board starts its considerations from a computation graph in which one vertex is responsible - and thus provides program code - for evaluating business rules.
- 5.2 It is natural that the business rules may have to be changed. It will require exclusively non-technical expertise, such as that of a business analyst, to detect such a need and to produce a new set of rules.
- 5.3 The board considers that it was well-known to render business rules in tabular form (see figures 2A and 2B). Moreover, just like the rules themselves, their tabular form *per se* is not a technical feature of the invention. The board had expressed this view in its summons, and the appellant did not challenge it.
- 5.4 Modifying the relevant graph component so as to implement the new set of rules may require programming skill and, thus, not be possible for business analysts. Instead, they may ask a programmer to translate a tabular specification of business rules into suitable program code.
- 5.5 Business analysts will perceive this as inconvenient and want the programming system to be changed such that they can enter new rules themselves.
- 5.6 The board considers that this is the objective problem addressed by the invention, and in this it agrees with the appellant (see its letter of 2 May 2016, point 16).

In view of the board's finding on inventive step, it may be left open whether this is a technical problem.

- 5.7 The skilled person setting out to solve that problem will find it obvious and straightforward to provide a tabular interface for a set of rules given in tabular form. For instance, as the description itself mentions (*loc. cit.*), commonly known spreadsheet programs provide suitable user interfaces.
- 5.8 The board takes the view that the further claimed step (or means) of "generating [...] a function for transforming data based on the rule specification" means no more than that the business rules are translated into program code while retaining their meaning. That the business rules *have to be* translated is part of the objective problem addressed, and that they *can be* is, in principle, obvious for the skilled person.
- 5.9 Executing business rules means applying them to actual values. It is obvious that, in practice, these may have to be obtained from a suitable database (e.g. of flight bookings and passenger data). Therefore, the compilation must map the variable names used in the business table to the names used in the databases. If these names happen to be identical, no explicit mapping step is required, nor would it be necessary to "store" one. However, the board considers that there are several obvious reasons why the variable names might be different. For instance, the database names might be too long or too short to be intelligible for a person or they might be in the wrong language (e.g. English for an application in German). It may also happen that the user interface software (e.g. the spreadsheet program) and the database management system follow

different naming conventions. The claimed "mapping" is an obvious way of handling such differences.

- 5.10 The appellant stressed that the invention "ring-fenced" the modifiable business-rule components and thus protected the remainder of the computation graph from inadvertent or unauthorised changes. While it is true that the claimed invention focuses on the modification of one computational component, it does not make any statements, positive or negative, on whether or how other components could be modified. The claims, therefore, do not imply any protective, "ring-fencing" measures. Moreover, the board considers that it is an immediate consequence of modular programming that individual components can be - and are meant to be - modified separately. Computation graphs as known in the art are modular at least in the sense that every two computation components are separate from each other.
- 5.11 The appellant also insisted that the claimed "simple and targeted (or "surgical")" modification of only a single component of a computation graph without affecting the rest was, in practice, less obvious than it might seem, due to implementation details such as how many files the business rules were stored in or how the interfaces between the computation graph components were specified. The board accepts that problems of this type may arise in practice and that solving them might involve significant work, but observes that such implementation details are not claimed and can, therefore, not be invoked in order to support an inventive step in the present case.
- 5.12 The board concludes that the claimed invention is an obvious solution to the problem of simplifying the input of business rules into a graph-based computation

system and therefore lacks inventive step in view of common general knowledge in the art, Article 56 EPC.

Auxiliary requests

6. The above analysis does not depend on whether the program code generated from the business rule is referred to as a "function" (see the main request and auxiliary request 3) or a "transform" (see auxiliary requests 1 and 2) or on whether this generation takes place in one step or in two steps via a "logical expression" (see auxiliary requests 2 and 3). The appellant confirmed this during oral proceedings and did not provide any argument why these differences could change the board's inventive step assessment.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated