

Internal distribution code:

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

**Datasheet for the decision
of 9 October 2020**

Case Number: T 2602/17 - 3.5.07

Application Number: 06785623.7

Publication Number: 1899873

IPC: G06F17/30, G06F9/44

Language of the proceedings: EN

Title of invention:

Managing metadata for graph-based computations

Applicant:

Ab Initio Technology LLC

Headword:

Metadata for graph-based computations/Ab Initio

Relevant legal provisions:

EPC Art. 56

Keyword:

Claims - clarity - main request (no)
Inventive step - all requests (no) - mixture of technical and
non-technical features

Decisions cited:

T 2055/08, T 1730/11



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 2602/17 - 3.5.07

D E C I S I O N
of Technical Board of Appeal 3.5.07
of 9 October 2020

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 28 June 2017
refusing European patent application
No. 06785623.7 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman M. Jaedicke
Members: P. San-Bento Furtado
C. Almberg

Summary of Facts and Submissions

- I. The appeal lies from the decision of the Examining Division to refuse European patent application No. 06785623.7, which was filed as international application PCT/US2006/024933 and published as WO 2007/002647, for lack of inventive step in the subject-matter of the independent claims of a main request and first and second auxiliary requests over prior art document
- D5: W. De Pauw et al., "Web Services Navigator: Visualizing the execution of Web services", IBM Systems Journal, vol. 44, no. 4, 2005.
- II. In the statement of grounds of appeal, the appellant submitted a main request and three auxiliary requests, where the main request and first and third auxiliary requests were based on the three requests considered in the appealed decision.
- III. In a communication accompanying a summons to oral proceedings, the Board introduced the following prior-art documents corresponding to patent applications by the appellant:
- D6: WO 02/11344 A2, published on 7 February 2002;
D7: WO 2005/001687 A2, published on 6 January 2005.

The Board expressed the preliminary opinion that claim 1 of all four requests did not satisfy the requirements of Articles 56, 84 and 123(2) EPC.

With regard to inventive step, the Board considered documents D6 and D7 to be better starting points than document D5. The subject-matter of claim 1 of all the requests did not seem to be inventive over the

disclosure of either D6 or D7. In its communication the Board expressed its opinion that the claimed subject-matter related at least in part to non-technical computer-program aspects, which as such were excluded from patentability.

- IV. With its letter of reply, the appellant filed a main request and four auxiliary requests. The appellant was conditionally prepared to withdraw its previous main and auxiliary requests filed with the statement of grounds of appeal. The appellant conditionally requested that the case be remitted so that its submissions in respect of the newly introduced documents D6 and D7 could be heard by the department of first instance.
- V. Oral proceedings were held by video conference as scheduled, during which the appellant withdrew its previous main and auxiliary requests and its request for remittal. At the end of the oral proceedings, the chairman pronounced the Board's decision.
- VI. The appellant's final requests were that the decision under appeal be set aside and that a patent be granted on the basis of the main request or one of the first to fourth auxiliary requests, all five new requests having been filed with the letter of reply dated 9 September 2020.
- VII. Claim 1 of the main request reads as follows:
"A method for determining metadata associated with graph elements of an executable graph-based computation including:
generating a worklist of graph elements for which metadata is to be determined with each graph element in an executable graph ordered in the worklist, the worklist comprising a partial ordering that is

determined at least in part by links interconnecting the graph elements, the links representing data flows of data elements; and

determining metadata for the graph elements of the executable graph according to the worklist, including propagating metadata internally within individual graph elements, and propagating metadata externally between different graph elements;

wherein propagating metadata internally includes propagating metadata within a graph element by functionally transforming metadata associated with a first portion (2310) of the graph element to generate transformed metadata associated with a second portion (2316) of the graph element that differs from the first portion; and

propagating metadata externally includes propagating metadata from the second portion of the graph element that has defined metadata to a further graph element of the graph that does not have defined metadata by:

determining a further element (2317) of the graph related to the second portion of the graph element by a link representing a data flow of data elements output from the second portion of the graph element and received at the further graph element; and

propagating the transformed metadata from the second portion of the graph element to the further graph element according to the link representing the data flow of data elements, and after propagating the transformed metadata moving the further graph element to the end of the worklist."

VIII. Claim 1 of the first auxiliary request reads as follows:

"A method for determining metadata associated with components of an executable graph-based computation including:

generating a worklist of components for which metadata is to be determined with each component in an executable graph ordered in the worklist, the worklist comprising a partial ordering that is determined at least in part by links interconnecting the components, the links representing data flows of data elements; and

starting at the beginning of the worklist, determining metadata for each of the components of the executable graph according to the worklist, including propagating metadata internally within individual components, and propagating metadata externally between different components;

wherein propagating metadata internally includes propagating metadata within a component by functionally transforming metadata associated with a first port (2310) of the component to generate transformed metadata associated with a second port (2316) of the component; and

propagating metadata externally includes propagating metadata from the second port of the component that has defined metadata to a further component of the graph that does not have defined metadata by:

determining a further component of the graph related to the second port of the component by a link representing a data flow of data elements output from the second port of the component and received at a port (2317) of the further component; and

propagating the transformed metadata from the second port of the component to the port of the further component according to the link representing the data flow of data elements, and after propagating the transformed metadata moving the further component to the end of the worklist."

- IX. Claim 1 of the second auxiliary request differs from that of the first auxiliary request in that "first port" and "second port" were replaced with "input port" and "output port", respectively. In addition, the text passages "from the second port of the component and received at a port (2317) of the further component" and "from the second port of the component to the port of the further component", were replaced with "from the output port of the component and received at an input port (2317) of the further component" and "from the output port of the component to the input port of the further component", respectively.
- X. Claim 1 of the third auxiliary request differs from claim 1 of the second auxiliary request in that after "with an output port (2316) of the component" the following text was added:
", wherein the metadata associated with the output port of the component is defined as a function of one or more parameters, at least one of the one or more parameters including metadata associated with the input port of the component".
- XI. Claim 1 of the fourth auxiliary request adds "computer-implemented" before "method" and the following text at the end of the claim:
"the method further comprising executing the graph-based computation, including:
 outputting data elements processed according to the transformed metadata from the output port of the component; and
 processing data elements received at the input port of the further component of the graph according to the transformed metadata."
- XII. The appellant's arguments, insofar as relevant to this decision, are addressed in detail below.

Reasons for the Decision

1. The appeal complies with the provisions referred to in Rule 101 EPC and is therefore admissible.

Application

2. The present application concerns determining metadata associated with components of an "executable graph", which is a directed data flow graph with vertices representing components, either data files or processes, and links indicating the flows of data between components (see page 1, line 4, to page 2, line 23, and page 8, lines 15 to 22, of the international publication).
 - 2.1 Such an executable graph can be created with the aid of a graphic development environment (GDE), which provides a user interface for creating executable graphs and defining parameters for the graph components. The GDE communicates with a repository and a parallel operating system (page 8, lines 15 to 22; Figure 1A). The repository stores all kinds of metadata, including (but not limited to) documentation, record formats, transform functions, graphs, jobs, and monitoring information (page 9, lines 1 to 14).
 - 2.2 Metadata describing the sequence of fields and data types corresponding to values in a record is called a "record format." A parameter including a record format for a port or other metadata associated with a component is bound to a value according to rules for parameter scoping. A parameter can be bound to a value at design time or at run-time (page 10, lines 3 to 23).

2.3 The value of metadata associated with a port, such as a record-format parameter, can be obtained by "propagation" (page 11, lines 6 and 7). Metadata that is propagated within a graph can include metadata that is functionally transformed, such as metadata that is defined as a function of other metadata (page 2, lines 17 to 23; page 5, lines 3 to 7). Metadata propagation can occur "externally" or "internally". In external metadata propagation, the value of a record-format parameter for a port of a first component is obtained by propagation from a port of a second component that is connected to the first component by a data flow. For internal metadata propagation, metadata defined for one port of a component propagates to another port of that component based on a sub-graph that implements the component (page 11, lines 6 to 23). According to page 34, lines 24 to 26, internal metadata propagation "includes deriving metadata for a port that has a metadata definition that refers to parameters of the graph and/or metadata for other port(s)".

2.4 An example of propagation of transformed metadata is given on pages 32 and 33 with reference to the smart-join component in Figures 23A and 23B. The smart-join component includes two input ports and one output port, each input port being connected to a sort component, and a join component which joins the two sets of ordered records from the sort components and outputs the result to the output port. Each of the sort components is to be used only if the respective input records are not yet sorted on a *key_field* parameter. The output port's metadata of the smart-join component is defined as a function of *key_field* and the input metadata at its two input ports. The function determines the output metadata by first determining whether the value of the *key_field* parameter is a

member of both sets of fields specified by the metadata of the input ports. If so (e.g. field sets *A,B* and *A,C*, with *key_field A*), the output metadata is the union of the two sets of fields (e.g. *A,B,C*). If not, the output metadata indicates an empty set of fields.

- 2.5 The claims are directed to the embodiment related to propagation of transformed metadata described on page 34, line 6, to page 35, line 14, with reference to Figure 24.

Admission of the requests into the proceedings

3. The main request and first to fourth auxiliary requests were filed in response to the Board's communication under Rule 100(2) EPC. That communication raised preliminary objections under Articles 84 and 123(2) EPC for the first time, and assessed inventive step taking into account newly introduced prior-art documents D6 and D7. The requests addressed those preliminary objections raised for the first time and the appellant justified its amendments. In exercising its discretion under Article 13(1) RPBA 2020, the Board found that those circumstances justified admitting the requests. Consequently, the Board admitted the requests into the appeal proceedings.

Main request - Article 84 EPC

4. *Clarity and support - claim 1*

- 4.1 The terms "first portion" and "second portion" are unclear and there is no support by the description for metadata propagation from a portion, or to a portion, of a graph element. According to the description, the metadata is propagated internally or externally from

one port to another port (page 34, line 18, to page 35, line 6).

4.2 In its letter and at the oral proceedings, the appellant argued that the terms were clear in the context of the application as a whole. If the skilled person was in any doubt as to the meaning of these terms, both the current claim 1 and claim 1 as originally filed provided reference numerals for these features. Referring to Figure 23A of the application, the first and second portions 2310 and 2316 in claim 1 were clearly indicating the input and output ports of the smart-join graph element 2306. Input and output ports in the appellant's view clearly represented portions of graph elements.

The Board does not find these arguments convincing. The original claims do not refer to a portion of a graph element. Instead, original claim 1 refers to a "portion of a graph" and original claim 6 specifies that a portion of a graph includes a port of a graph element. With regard to the argument that the ports in Figure 23A are portions, the Board notes that a port is a portion, but a portion is not necessarily a port. The claim covers propagation of metadata between any portions of graph elements, which is broader than justified by the description. Moreover, the meaning of claim features must be clear to the person skilled in the art from the wording of the claim alone which in this case they are not.

4.3 Therefore, claim 1 of the main request does not fulfil the requirements of Article 84 EPC.

*Main request, and first and second auxiliary requests -
Article 56 EPC*

5. In its letter and at the oral proceedings, the appellant agreed that claim 1 of the main request and the first and second auxiliary requests recited the same subject-matter using different terminology. Essentially, "graph elements" and "portions" in claim 1 of the main request correspond to "components" and "ports", respectively, in claim 1 of the first and second auxiliary requests. In the following assessment of inventive step, the Board uses the terminology of the second auxiliary request, the reasoning applying equally to the main request and the first and second auxiliary requests.

6. *Inventive step - claim 1*

6.1 The subject-matter of claim 1 of the second auxiliary request relates at least in part to a computer program, which as such is excluded from patentability under Article 52(2) and (3) EPC.

In its statement of grounds of appeal, the appellant referred to decision T 1730/11 of 17 November 2014, according to which the efficient execution of executable graphs provided a technical effect and the pre-computation of pools of processes dedicated to different types of processing and vertexes contributed to the efficient execution of executable graphs. The appellant argued that, by analogy, the present invention was concerned with propagation of metadata through data-flow graphs and also resulted in the efficient design, construction and execution of executable graphs.

The Board is not convinced that the metadata propagation in the present case is related to using shared distributed memory, processing multiple data streams concurrently or providing pools of processes, as was the case in decision T 1730/11 (points 4.2, 4.8 and 4.9 of the Reasons). The claim does not specify multiple data streams and does not specify any features related to such techniques. Therefore, those conclusions of that decision do not apply in the present case.

- 6.2 In the decision under appeal, the Examining Division considered that document D5 disclosed, with reference to the right-hand side of Figure 1, a method for determining metadata associated with graph elements of an executable graph-based computation. The appellant contested that document D5 was concerned with executable graphs and the problem of metadata propagation. The "Services Topology View" was used in document D5 merely to visualise the communications. Therefore, document D5 did not provide a good starting point.

In its communication, the Board introduced documents D6 and D7, either of which it considered a more suitable starting point to assess inventive step of the claimed subject-matter.

- 6.3 Document D6 discloses a method and system for creating and executing a graph having components with run-time parameters and for providing run-time graph parameters and conditional components for data flow graphs. With run-time parameters it is possible to defer the value of a parameter setting to run-time. The values of run-time parameters may be derived from a combination of other run-time parameters or objects stored in an object repository. A graph structure can be changed

based on parameter values and computed metadata (abstract; page 3, line 2, to page 4, line 22). Therefore, document D6 discloses a method for determining metadata associated with components of an executable graph-based computation.

In order to support conditional components and run-time parameters, the system of D6 computes metadata and propagates metadata between components (page 20, line 5, to page 25, line 27; claims 1, 2, 6, 7 and 9). The method of D6 thus includes a step of propagating metadata externally by determining a further element of the graph and propagating the transformed metadata.

In its letter and at the oral proceedings, the appellant argued that document D6 did not disclose propagating metadata internally from an input port to an output port of a component. In the Board's opinion, it is implicit from document D6 that in some components the metadata associated with an output port of a component is a function of the metadata associated with the input port. For example, in a component for selecting only the attributes "name" and "address" of the input records with format "name, address, date of birth", the metadata associated with the output port "name, address" is a function (a selection) of the metadata "name, address, date of birth" associated with the input port. However, the Board recognises that document D6 does not disclose internally propagating functionally transformed metadata within the context of the propagation method.

- 6.4 The method of claim 1 therefore differs from that of document D6 in that it includes the claim steps of
- generating an ordered worklist of components for which metadata is to be determined;

- determining metadata for the components of the executable graph according to the worklist, including propagating metadata internally within individual components by functionally transforming metadata associated with an input port of the component to generate transformed metadata associated with an output port of the component; and
- propagating the transformed metadata to a further component, and then moving the further component to the end of the worklist.

The distinguishing features can be divided into two main differences: adding internal propagation of metadata to the propagation method of D6 which already supported external propagation, and performing the propagation by means of a worklist. The first difference concerns how metadata used by related functions of a computer program evolves. The second difference is the result of the computer-programming task of implementing propagation. Therefore, both concern computer-programming aspects which as such are excluded from patentability under Article 52(2) and (3) EPC. They are to be taken into account for inventive step only if they contribute to a further technical effect or are based on further technical considerations.

6.5 In its letter in reply to the Board's communication, the appellant argued that the metadata definition specifying metadata as a function of other metadata allowed metadata to propagate even over transforming internal data paths. Furthermore, the metadata could propagate at edit time before the graph was run. The worklist was technical because it assisted in the efficient determination of metadata. The technical

effect of the distinguishing features was that the node at the top of the worklist would always be one which had not been processed. This feature made determining metadata more efficient.

According to the appellant, the objective technical problem could be formulated as how to provide a more efficient method for determining metadata associated with components of an executable graph-based computation in which the data that is to be processed by a component can change at run-time.

The appellant's arguments are not persuasive. As the Board explained at the oral proceedings, the claim defines the method in very abstract and broad terms. It cannot be recognised from the claim that the propagation occurs at edit time or is related to solving, over document D6, the problem of data or metadata being changed at run-time. The implementation of the propagation with a worklist is not based on any technical considerations regarding an efficient propagation of the metadata.

- 6.6 At the oral proceedings, the appellant further argued that the distinguishing features provided reliability and efficiency of execution of the graph. The propagation made the processing system run better and guaranteed that the graph was correct before execution. In accordance with decision T 2055/08 of 13 February 2014, the internal consistency of a data structure was technical.

The Board notes that the invention dealt with in decision T 2055/18 addresses the problem of providing a versioned and referenced collection of digital objects that is free from version conflicts, i.e. free from inconsistencies. This is achieved by providing the

referencing objects with metadata defining references to the referenced objects and using the metadata to solve version conflicts by propagating them from the referencing to the referenced objects. The board in that case found that the invention addressed a technical problem, i.e. the internal consistency of a data structure for the purposes of a specific application (an electronic learning system), and solved it by technical means, i.e. by using metadata for detecting references to a learning object in more than one version. In decision T 2055/18, the board considered that the use of metadata, understood as "data over data", for resolving version conflicts in a collection of digital objects involves technical considerations (point 3.3 of the Reasons).

The Board is, however, not convinced that the reasoning of T 2055/08 applies in the present case. In the claim dealt with in that decision, version conflicts were detected and resolved (Summary of Facts and Submissions, III). By contrast, the present claim 1 does not refer to conflicts or inconsistencies, and it does not specify that consistency is checked or what happens if the graph is found to be incorrect, for example, if the metadata is inconsistent in some way. The Board further notes that document D6 already provides for external propagation of metadata in the same context of executable graphs.

- 6.7 As the Board is not convinced that the distinguishing features contribute to a further technical effect or are based on further technical considerations, it concludes that the distinguishing features merely concern non-technical computer-programming aspects as such and do not have to be taken into account for inventive step. The appellant's argument that the

skilled person would not modify D6 to add internal data propagation because D6 was concerned only with propagating metadata when a conditional component was removed is not relevant because the distinguishing features are not technical. Besides, there is no technical reason why metadata propagation could not or would not be used for other purposes within the context of D6.

- 6.8 Therefore, the subject-matter of claim 1 of the main request and the first and second auxiliary requests is not inventive (Article 56 EPC).

Third and fourth auxiliary requests - Article 56 EPC

7. Claim 1 of the third auxiliary request differs from claim 1 of the second auxiliary request essentially in that it further specifies that "the metadata associated with the output port of the component is defined as a function of one or more parameters, at least one of the one or more parameters including metadata associated with the input port of the component".
8. Claim 1 of the fourth auxiliary request further specifies that the method is computer-implemented and comprises executing the graph-based computation, including
- outputting data elements processed according to the transformed metadata from the output port of the component; and
 - processing data elements received at the input port of the further component of the graph according to the transformed metadata.

9. *Inventive step - claim 1*

9.1 The appellant argued that the additional feature of the third auxiliary request further contributed to the effects of propagating metadata even over transforming internal paths and propagating at edit time. The technical problem solved was the same as that formulated for the higher-ranking requests.

However, specifying a function is, in the Board's view, a computer-programming feature. In the context of claim 1, the specification of the function defining the metadata at the output port in terms of the metadata at the input ports of a component does not contribute to a technical effect for essentially the same reasons as given for the higher-ranking requests.

9.2 In the inventive-step assessment of the higher-ranking requests the Board has already interpreted the claimed method as a computer-implemented method, and therefore inserting "computer-implemented" in claim 1 of the fourth auxiliary request has no effect on that assessment.

The method of document D6 includes steps of executing the graph-based computation including outputting data elements processed according to metadata from an output port of a component, and processing data elements received at the input port of a further component of the graph according to the metadata.

Therefore, document D6 discloses the additional features of the fourth auxiliary request, except that the metadata in the steps of outputting and processing is "transformed metadata".

The appellant argued that the metadata was used to control the execution of the graph to ensure that only necessary data was processed by a particular component and therefore contributed to more efficient execution of the graph-based computation.

The appellant's arguments are not persuasive. Even though the claim adds the execution of the graph, a technical effect over the method of D6 is still not recognisable. The prior-art method of D6 also includes the additional steps of executing the graph and its sub-steps and uses metadata to control execution of the graph. As explained for the higher-ranking requests, the Board is not persuaded that the support for "transformed metadata" contributes to a technical effect. It is not clear from the claim that the propagation occurs at the design phase. The data elements output by the component are not used to solve a technical problem over the method of document D6.

9.3 Therefore, claim 1 of the third and fourth auxiliary requests does not fulfil the requirements of Article 56 EPC.

Conclusion

10. Since none of the requests is allowable, the appeal is to be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



A.Voyé

M. Jaedicke

Decision electronically authenticated



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 2602/17 - 3.5.07

D E C I S I O N
of the Technical Board of Appeal 3.5.07
of 20 November 2020
correcting an error in the decision
of 9 October 2020

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 28 June 2017
refusing European patent application
No. 06785623.7 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chair: M. Jaedicke
Members: P. San-Bento Furtado
C. Almberg

Pursuant to Rule 140 EPC, the decision T 2602/17 of 9 October 2020 is hereby corrected as follows:

Replacement of "T 2055/18" with "T 2055/08" on page 14, last paragraph, and page 15, first paragraph.

The Registrar:

The Chair:



S. Lichtenwort

M. Jaedicke

Decision electronically authenticated