

Internal distribution code:

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

**Datasheet for the decision
of 11 September 2012**

Case Number: T 1122/08 - 3.5.06

Application Number: 99966713.2

Publication Number: 1149336

IPC: G06F 9/44, G06F 9/46

Language of the proceedings: EN

Title of invention:
Method and apparatus for a user extensible event structure

Applicant:
Computer Associates Think, Inc.

Headword:
Event structures/COMPUTER ASSOCIATES THINK

Relevant legal provisions (EPC 1973):
EPC Art. 56, 84

Keyword:
"Amended claims searched - no"
"Remittal for further prosecution - yes"



Case Number: T 1122/08 - 3.5.06

D E C I S I O N
of the Technical Board of Appeal 3.5.06
of 11 September 2012

Appellant: Computer Associates Think, Inc.
(Applicant) One Computer Associates Plaza
Islandia, New York 11749 (US)

Representative: Dunlop, Hugh Christopher
R.G.C. Jenkins & Co.
26 Caxton Street
London SW1H 0RJ (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 31 January 2008
refusing European patent application
No. 99966713.2 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman: D. H. Rees
Members: M. Müller
C. Heath

Summary of Facts and Submissions

- I. The appeal lies against the decision of the examining division, with written reasons dated 31 January 2008, to refuse the European patent application 99966713.2 for lack of an inventive step.
- II. An appeal was filed on 10 April 2008 and the appeal fee was paid on the same day. A statement of grounds of appeal was filed on 10 June 2008. It was requested that the decision be set aside and that a patent be granted based on one of five sets of claims filed with the statement of grounds of appeal, according to the main and first to fourth auxiliary requests, and, as the board understands the appellant's requests, in combination with the pending application documents
- | | | |
|--------------------|---------|-----------------------|
| description, pages | 1, 2 | dated 9 November 2005 |
| | 3-14 | as published |
| drawings, sheets | 1/5-5/5 | as published |
- III. With summons to oral proceedings, the board made reference to, *inter alia*, the following documents:
- D1: Abelson H *et al.*, "Structure and Interpretation of Computer Programs", excerpt from pp. 212-215, The MIT Press, 1993
- D3: Kent M, "A General-Arrays Implementation of Association Lists", ACM SIGAPL APL Quote Quad, vol. 23, no. 4, pp. 3-5, ACM Press, June 1993
- D5: Platinum Technology: "Application Note: ProVision Network Monitor Integration with ProVision - Version 4.2 and higher", 1 October 1998

The board raised a number of clarity objections and gave its preliminary opinion according to which the main request and auxiliary requests 1, 2 and 4 lacked an inventive step over D5 in view of common knowledge as evident from D1 or D3, Article 56 EPC 1973, whereas the third auxiliary request would have to be remitted to the department of first instance for further prosecution, including presumably an additional search.

- IV. In response to the summons, the appellant filed, for all five pending requests, amended sets of claims intended to overcome the clarity objections. The board informed the appellant that it was minded to remit the case for further prosecution on the basis of the third auxiliary request even though the clarity objections might not have all been overcome.
- V. In view of this, the appellant withdrew all requests except the third one, and further withdrew its request for oral proceedings conditional to the remittal to the first instance based on the sole remaining request labelled "third auxiliary request". The oral proceedings were thus cancelled.
- VI. Independent claims 1 and 11 of the remaining request read as follows:

"1. An event messaging method comprising:

extending an event structure representing an individual event, the event structure including a predefined field for storing information relating to an event, and further including a keys field array and a values field array, by:

submitting (510) a keyname and a corresponding value for the event structure;

determining (540) whether the keyname exists in the keys field array of the event structure;

if the keyname does not exist in the keys field array, incrementing (550) an index of the event structure, adding (560) the keyname to a position in the keys field array based on the index, and adding (570) the corresponding value to a position in the values field array based on the index; and if the keyname does exist in the keys field array, determining (580) the position of a previously stored value in the values field array associated with the keyname, and replacing (590) the previously stored value in the values field array with the corresponding value;

generating a message describing the event structure, wherein the message includes: descriptors of fields of the event structure; and the contents of the fields identified by the descriptors;

communicating the message from a first process to a second process; and

in the second process, populating a further event structure by:

searching the message for fields required by the further event structure; and

when the message does not comprise a field required by the further event structure, populating the

further event structure with a default value based upon fields in the message."

"11. An event messaging apparatus comprising first and second processes, wherein the first process comprises:

means for extending an event structure representing an individual event, the event structure including a predefined field for storing information relating to an event, and further including a keys field array and a values field array, the means for extending comprising:

means for submitting (510) a keyname and a corresponding value for the event structure;

means for determining (540) whether the keyname exists in the keys field array of the event structure;

means for, if the keyname does not exist in the keys field array, incrementing (550) an index of the event structure, adding (560) the keyname to a position in the keys field array based on the index, and adding (570) the corresponding value to a position in the values field array based on the index; and

means for, if the keyname does exist in the keys field array, determining (580) the position of a previously stored value in the values field array associated with the keyname, and replacing (590) the previously stored value in the values field array with the corresponding value;

means for generating a message describing the event structure, wherein the message includes: descriptors of

fields of the event structure; and the contents of the fields identified by the descriptors; and

means for communicating the message from the first to the second process,

and wherein the second process comprises means for populating a further event structure by searching the message for fields required by the further event structure, wherein the means for populating a further event structure is operable to populate the further event structure with a default value based upon fields in the message when the message does not comprise a field required by the further event structure."

Reasons for the Decision

The invention

1. Generally, the application relates to an event management system for a distributed computing environment (see original description, p. 1, lines 7-23 and p. 2, lines 18-30).
- 1.1. Individual events in this context are represented as instances of a predefined data type - referred to as "event structures" - with multiple fields (cf. p. 3, line 26 - p. 4, line 35). Some of these fields identify an event, some contain additional information about it (p. 3, lines 17-22). Amongst the identifying fields, there are two which jointly define a list of key-value pairs (p. 4, lines 15, 16 and 36-40); these are called, respectively, a "key fields array" and a "value array".

One aspect of the claimed invention relates to the extension of an event structure by setting a given key to be associated with a given value.

- 1.2. Another aspect of the invention relates to the communication of events between different "processes" of the distributed system which, as the application points out, may support event structures in different versions, *i.e.* with different fields (p. 13, lines 9-11 and p. 14, lines 3-6). Communication of an event structure from a first to a second process requires the first process to pack (or "marshal") it into a message structure and the second process to unpack (*i.e.* unmarshal) it and "populate" a new, local event structure (see *e.g.* p. 13, lines 11-15 and p. 13, lines 29 - p. 14, line 7). If the second process supports a newer version of event structures than the first process, with fields absent from the incoming message, a "compatibility mechanism" is invoked (p. 13, line 7 ff.) and these fields will receive a default value based upon other fields in the message (cf. p. 14, lines 6-7).

Article 123 (2) EPC

2. The board is satisfied that the invention according to independent claims 1 and 10 is disclosed in the parts of the original description just cited.

Clarity, Article 84 EPC 1973

3. The extension of an event structure by a new key-value pair according to claims 1 and 11 modifies the keys field array and the values field array but neither adds any new fields nor affects the value of the version number field (cf. p. 3, line 30). It follows that the

two aspects of the invention (cf. points 1.1 and 1.2 above), the "compatibility mechanism" for handling different versions of event structures and the operation of "extending" an event structure, are essentially unrelated to each other. That said, the board deems it to be unclear that claim 11 specifies the "means for extending an event structure" to comprise the "means for generating a message" and the "means for communicating the message" and that claim 1, analogously, specifies the step of "extending an event structure" to comprise the steps of "generating a message" and "communicating the message".

4. According to independent claims 1 and 11, when a given "keyname" does not exist in the event structure being extended, an index is incremented and the key name placed at the newly obtained position in the keys field array defined by the index. The skilled person would take this to mean that only allowable indexes are created, most probably only positive ones. This is in conflict with claims 6 and 16 according to which an error message is provided when "the index is less than one". Claims 6 and 16 are thus unclear.

4.1. In the board's understanding, claims 1 and 11 are based on the function PtEventSetKeyValuePair (p. 11, lines 10-18 and p. 12, lines 23-29), which takes a key name and a value and generates a suitable index, whereas claims 6 and 16 are based on PtSetKeyValueByIndex (p. 12, lines 15-22) which takes a key name, a value and an index and stores the key name and the value at the given index position in the arrays. The contradiction between the claims appears to be caused by the fact that the latter function needs an error message if

it is called with a negative value, while the former does not because it determines the pertinent index itself.

5. In view of its finding on inventive step (see below), the board deems it inappropriate to dismiss the appeal based on these clarity problem which can, as it appears, be easily resolved.

Closest Prior Art

6. The claims subject to the refusal were directed towards method and apparatus "for extending an event structure" and covered only the first aspect of the invention as now claimed (point 1.1. above). On the understanding that the invention claimed then "lies in the field of data structures", the examining division based their assessment of inventive step on D1 as the closest prior art (see reasons, points 13.1 and 15-15.4). In the grounds of appeal (p. 3, 1st par., and p. 4, 2nd par.) the appellant refuted D1 as an unsuitable starting point because, *inter alia*, D1 "is not concerned with event handling and event structures".

- 6.1. While the board is not convinced by this argument for the claims as refused, which relate to method and apparatus of "extending an event structure", it does have its merits for the present, amended independent claims which relate to an event messaging method and apparatus between processes possibly requiring event structures with different fields.

- 6.2. For this reason, the board considers that the most suitable starting point for the assessment of novelty and inventive step is D5 rather than D1.

Inventive Step, Main Request

7. D5 addresses event handling in a distributed environment as discussed in the application and, moreover, discusses an event management tool related to the one mentioned in the application (see p. 1, line 13).
- 7.1. D5 discloses which fields should be "sent as part of [a] published event" (p. 5, last two lines; p. 6). The skilled person would, in the board's judgment, take this to define an "event structure" with fields "relating to an event". D5 further defines that these field should comprise "Key Name/Value Pairs" (p. 6, last row in the table) and that, for different event types, different keys and a different number of keys may be defined.
- 7.2. D5 also discloses that events are communicated across the distributed environment, *i.e.* from a first to a second process as claimed (see *e.g.* p. 3 and p.4, last two pars.)
8. Independent claims 1 and 10 thus differ from D5 by the following features:
- i) the implementation of key/value pairs are in terms of two arrays, a "keys field array" and a "values field array";

ii) a manner of extending the event structure by a new key/value pair; and

iii) details about generating a message representing the event structure and "populating a further event structure" in the second process.

8.1. The feature under i) represents an implementation of the key/value list specified in D5. The features under ii) implement a way of setting and/or updating the key/value pairs in an event structure. In the board's view these two address the problem of implementing a data type "event structure" as specified in D5.

8.2. The features under iii) address the problem of communicating events between processes with possibly different definitions of event structures.

8.3. In the board's view, the problems solved by the features under i) and ii) as opposed to those under iii) are essentially unrelated so that their inventive merit may be assessed separately.

Implementing event structures

9. The board considers that the features under i) and ii) represent rather mundane and well-known programming patterns from which the skilled person would choose to implement the event structure according to D5.

9.1. In the board's view, the skilled person will be aware that the key/value pairs represent a "mapping" from keys to values and will consider known options for implementing such mappings. Doing this the skilled per-

- son will realise that the implementation of fundamental data types such as these belongs to the basic knowledge in the art of programming and to the syllabus of typical programming courses.
- 9.2. D1 is from a textbook for students of programming used in this context and discloses an implementation of a key/value "table" and an operation of "extending" this table with a new key/value pair according to an algorithmic specification as claimed (see p. 215, 1st par.).
- 9.3. That D1 happens to use the programming language Lisp (it actually uses a dialect of Lisp called Scheme) is irrelevant in this respect. While Lisp by its very nature as a "List Processing" language may favour list-based data structures, the teaching of D1 can easily be adapted to other programming languages, too. Indeed, it is considered a straightforward task for the skilled person to implement the "insert" specification according to D5 (*loc. cit.*, p. 215, 1st par.) in another programming language.
- 9.4. D1 happens to implement the key/value pairs as a list of pairs. While this is a preferred way of implementing mappings in Lisp, it is well-known that "association lists" can be equivalently represented as a pair of lists, too (cf. e.g. D3, p. 3, right col., lines 10-20). It is well-known that these representations, while functionally equivalent, may be interestingly different in supporting certain operations more directly (and thus more efficiently) than others. Thus, in a concrete case the skilled person might prefer one representation over the other depending on which operations on the data structure should be fast (for instance because they will be frequently used). If such considerations happen

to be irrelevant, the skilled person may also merely decide by convenience or personal preference. In the board's view it is a matter of routine for the skilled person to balance the requirements and make suitable decisions according to circumstances when designing the data types for an application of interest.

- 9.5. The board thus considers that the features under ii) represent a standard way of specifying the "extension" of a key/value list by a new key/value pair (see D1) and that the implementation of the key/value pairs in terms of a pair of "arrays", *i.e.* feature i), rather than an array of pairs is an obvious alternative for the skilled person (see D3).

Communication of event structures between processes

10. Regarding the features under iii), the board understands the decision under appeal as follows: It is common knowledge to use serialisation/marshalling to transmit data structures between computers, and it is implied by this operation that the sender generates a message describing a given data structure and the receiver "populates" - *i.e.* re-creates - a data structure according to this message (cf. reasons 16.1, items (a) and (b) and 16.3, first dash). And it is obvious to use a default value "in the given situation", especially considering that "no indication is given how to determine the default value" (reasons 16.1, item (c) and 16.3, second dash).
11. The board agrees that serialisation and marshalling are common techniques to communicate data structures between processes and considers that applying such known techniques to D5 would be obvious for the skilled person

given the indication that events need to be communicated between processes (see point 7.2 above).

- 11.1. Marshalling (and, likewise, serialisation) is commonly understood to denote the process of transforming a data object from a (high-level) data format in local memory into a (low-level) transportable format. The inverse process (unmarshalling) is used to produce a corresponding data object in local memory at the receiver side. The board agrees with the decision under appeal that this covers the claimed steps and means of generating a message, communicating it and "populating" a new data structure at the receiving end.
- 11.2. To the best of the board's knowledge, however, marshalling normally assumes that sender and receiver support the same data types and that, hence, communication between processes supporting incongruent data types, such as event structures in different versions, does not fall under the conventional meaning of serialisation or marshalling.
- 11.3. The board can only speculate whether in a distributed computing environment such as that of D5 the communication of data in the presence of incongruent data structures is a known issue, and whether it has been addressed in the prior art or how. Therefore, even the undisputed allegation from the decision under appeal that marshalling is well-known seems insufficient to show lack of an inventive step.
- 11.4. Hence, the board concludes that claims 1 and 10 show an inventive step over D1 in combination with any of D3, D5 or common knowledge in the art, Article 56 EPC 1973.

Scope of the search

12. According to the decision under appeal (point 18.1), the examining division had "good reason to believe" that the features under iii) were not searched.
- 12.1. The board finds this assumption plausible. The original application contained only a single claim, which corresponds to claim 1 of the main request, whereas all other claims are based on features taken from the description, which, moreover, relate to an issue unrelated to the one originally claimed (see also point 3 above).
- 12.2. The decision further explains (point 18.2) that the examining division did not consider it appropriate to carry out an additional search because the additional features were "sufficiently known from or obvious on the basis of common general knowledge".
- 12.3. Since the pertinent features appear not to have been searched the board has no basis to come to a final conclusion on the inventive merit of claims 1 and 10.
- 12.4. Therefore, the case must be remitted to the first instance for further prosecution, especially for an additional search to be carried out and to deal with the outstanding clarity objections (see points 3 and 4 above).

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the department of first instance for further prosecution.

The Registrar:

The Chairman:

B. Atienza Vivancos

D. H. Rees