

Internal distribution code:

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

**Datasheet for the decision
of 12 October 2011**

Case Number: T 1928/07 - 3.5.06

Application Number: 01130193.4

Publication Number: 1217518

IPC: G06F 9/46, G06F 9/54

Language of the proceedings: EN

Title of invention:

Processor unit for executing event process in real time in response to occurrence of event

Applicant:

DENSO CORPORATION

Opponent:

-

Headword:

Shared and dedicated tasks with event processes/DENSO

Relevant legal provisions:

-

Relevant legal provisions (EPC 1973):

EPC Art. 84

Keyword:

"Claim 1 of main and auxiliary requests 1-4: clarity (no)"
"Auxiliary request 5 (non-admissible)"

Decisions cited:

-

Catchword:

-



Case Number: T 1928/07 - 3.5.06

D E C I S I O N
of the Technical Board of Appeal 3.5.06
of 12 October 2011

Appellant: DENSO CORPORATION
(Applicant) 1-1, Showa-cho
Kariya-city
Aichi-pref. 448-8661 (JP)

Representative: TBK
Bavariaring 4-6
D-80336 München (DE)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 23 April 2007
refusing European patent application
No. 01130193.4 pursuant to Article 97(1)
EPC 1973.

Composition of the Board:

Chairman: M.-B. Tardo-Dino
Members: S. Krischer
G. Zucka

Summary of Facts and Submissions

I. The appeal is directed against the decision, posted on 23 April 2007, of the examining division, to refuse the application 01 130 193.4.

The reason for the refusal was lack of clarity, in violation of Article 84 EPC. As an obiter dictum, it was furthermore stated that the claims appeared to lack inventive step (Article 56 EPC).

II. A notice of appeal was received on 2 July 2007. The fee was paid on the same day. A statement setting out the grounds of the appeal was filed on 30 August 2007. Oral proceedings were conditionally requested. The following document was introduced with the grounds of the appeal:

D5 Microsoft Computer Dictionary, fifth edition, pages 512-423.

III. The board issued a summons to attend oral proceedings to be held on 12 October 2011. It raised objections relating to a lack of clarity (Article 84 EPC) and lack of inventive step (Article 56 EPC). The following document was cited:

D6 A. SILBERSCHATZ, P. GALVIN: "Operating System Concepts", ADDISON-WESLEY USA, fifth edition, 1998, ISBN 0-201-59113-8, pages 89-103.

IV. In a letter dated 5 September 2011, the appellant filed a fourth auxiliary request.

- V. During oral proceedings, held on 12 October 2011, a fifth auxiliary request was filed.
- VI. The appellant requested to set the decision aside and to grant a patent on the basis of the main request (claims 1-13) or of one of the five auxiliary requests (claims 1-13; claims 1-9; claims 1-12; claims 1-8; claims 1-6), filed with the grounds of the appeal (first to third auxiliary request) and on 5 September 2011 (fourth auxiliary request) and during oral proceedings (fifth auxiliary request), with the remaining documents as on file, i.e. description pages 1-3, 6-13, 13a as originally filed; pages 4, 4a, 4b, 5 as filed on 5 February 2007; drawing sheets 1-3 as originally filed.
- VII. Claim 1 of the *main request* reads as follows:
- "1. A processor unit (1) for executing an event process in response to occurrence of a plurality of predetermined events, comprising:
- a program for executing an event process in response to occurrence of said plurality of predetermined events,
 - wherein said program comprises:
 - a plurality of tasks (12c, 12d) comprising shared tasks (12c) each of which includes event processes and at least one dedicated task (12d) which includes one event process, wherein one of the event processes is executed in response to occurrence of one of said plurality of predetermined events;
 - an activation request program (12a) for requesting activation of a task of the plurality of tasks (12c, 12d) which includes an event process

corresponding to an event in response to occurrence of said event; and

a real time operating system (12b) for activating said task (12c, 12d) in response to the request for activation of said task (12c, 12d),

wherein said activation request program (12a) is adapted to store identification information on said event process corresponding to said event in a storage area in response to occurrence of said event, if said task is a shared task,

wherein said task (12c, 12d), which is activated by said real time operating system (12b), is adapted to obtain the identification information on said event process from said storage area if said task (12c, 12d) is a shared task, and is adapted to execute said event process corresponding to the obtained identification information,

wherein at least one specific event is selected from said plurality of predetermined events, and an event process corresponding to said specific event is included in the dedicated task (12d), so that said dedicated task (12d) identifies the event process corresponding to the specific event for execution without obtaining identification information from said storage area when said dedicated task (12d) is activated."

VIII. Independent record medium claim 11, system claim 12 and method claim 13 are almost identical to claim 1 apart from some adaptations to the claim category.

IX. Claim 1 of the *first auxiliary request* differs from claim 1 of the main request by adding "[a plurality of tasks (12c, 12d)], each of which causes the processing

unit to execute at least one associated event process" in line 17.

- X. Claim 1 of the *second auxiliary request* differs from claim 1 of the main request by adding "wherein said processor unit (1) is adapted to execute an event process in response to the occurrence of one of said plurality of predetermined events for controlling an engine, wherein a cyclic event, which occurs in synchronous with a run cycle of the engine or which occurs regularly at predetermined time intervals, is selected as said specific event." at the end of the claim.
- XI. Claim 1 of the *third auxiliary request* differs from claim 1 of the main request by adding "wherein one of a plurality of priority levels is assigned to each of said plurality of tasks (12c, 12d) and said event process included therein, and a same priority level is assigned to a task (12c, 12d) and an event process included therein;" at the end of the section starting with "a plurality of tasks", and by adding "if a priority level of said task (12c, 12d) is higher than a priority level of an active task," at the end of the section starting with "a real time operating system".
- XII. Claim 1 of the *fourth auxiliary request* reads as follows:

"1. A processor unit (1) for executing event processes in response to occurrences of a plurality of predetermined events for controlling an engine in a vehicle,

the processor unit (1) comprising:

a plurality of tasks (12c, 12d) that contains

(i) more than one shared task (12c), the shared task being for executing a shared-task event process, which is an event process of the shared task included in the plurality of event processes, and

(ii) at least one dedicated task (12d), the dedicated task being for executing only one dedicated-task event process, which is an event process of the dedicated task included in the plurality of event processes;

an activation request program (12a) for requesting an activation of the shared task (12c) or the dedicated task (12d) whichever is for executing an event process corresponding to one of the predetermined events that occurs; and

a real time operating system (12b) for activating the shared task (12c) or the dedicated task (12d) the activation of which is requested by the activation request program (12a), wherein:

(i) in cases that the activation request program (12a) requests an activation of a shared task (12c) for executing a shared-task event process corresponding to an event that occurs, the activation request program (12a) is adapted to store, in a storage area, identification information on said shared-task event process; and

(ii) when said shared task (12c) is activated by the real time operating system (12b), said shared task (12c) is adapted to obtain the identification information on said shared-task event process from the storage area to thereby execute said shared-task event process corresponding to the obtained identification information,

wherein in the processor unit (1):

(i) the dedicated-task event process is adapted to be executed by the dedicated task (12d) in response to an occurrence of a specific event included in the plurality of predetermined events, the specific event being a cyclic event, which occurs in synchronous with a run cycle of the engine or occurs regularly at predetermined time intervals; and

(ii) when the dedicated task (12d) is activated by the real time operating system (12b) based on the occurrence of the cyclic event, the dedicated task (12d) is adapted to execute the dedicated-task event process corresponding to the cyclic event without obtaining any identification information from the storage area."

XIII. Claim 1 of the *fifth auxiliary request* differs from claim 1 of the *fourth auxiliary request* by adding at the end of the claim: "wherein one of a plurality of priority levels is assigned to each of said plurality of tasks (12c, 12d) and said event process included therein, and a same priority level is assigned to a task (12c, 12d) and an event process included therein, and wherein said real time operating system is adapted to activate said task (12c, 12d) in response to the request for activation of said task if a priority level of said task (12c, 12d) is higher than a priority level of an active task, wherein the shared tasks (12c) are prepared for respective priority levels, and each of event processes corresponding to events other than the events selected as the specific processes is included in the shared task (12c) whose priority level is the same as that of the event process."

XIV. At the end of the oral proceedings, the chairwoman announced the board's decision.

Reasons for the Decision

1. *Admissibility of the requests*

1.1 As to the various amendments made in claim 1 of the *fourth* auxiliary request, the board finds that they satisfy the requirements of Article 123(2) EPC for the following reasons. Although the claim has been heavily reformulated in comparison to the main request, the content of the claim remains the same, the amendments being a serious attempt to clarify the subject-matter:

- The expression "a program for executing an event process ... wherein a program comprises [a plurality of tasks]" was deleted. This "comprised" relationship of a task in a program had been objected to in the summons, section 4.4, point a) which is now overcome by the deletion. There is no technical necessity of a "comprised" relationship for the remaining features.
- "More than one shared task" (line 8): see figure 2.
- The relationship that shared/dedicated tasks include event processes was deleted at several occurrences in the claim. This had been objected to in the summons, section 4.4, point b) which is also overcome now. There is no technical necessity for an inclusion relationship in the context of the remaining features.

Moreover, the fourth auxiliary request was filed more than one month before the oral proceedings. Thus, it is admitted to the procedure.

- 1.2 At the beginning of the oral proceedings, a *fifth* auxiliary request was filed. Claim 1 of this request is based on a combination of claims 1 and 2 of the fourth auxiliary request to which parts of the wording of original description page 9, lines 6-11 were added. Thus, the requirements of Article 123(2) EPC are fulfilled.

The appellant justified the late filing of this request by stating that the added feature of priority levels clarified the relation between tasks and event processes. The board did not see how the commonly known feature of priority levels could remedy the lack of clarity of the expression "event process" and of the "execution" of an event process by a task. Moreover, priority levels were already present in claim 1 of the third auxiliary request where they were found to introduce a new independent aspect of task scheduling, unable to clarify the above-mentioned expressions (see section 2.4 below).

Therefore, the fifth auxiliary request was not admitted to the proceedings.

2. *Clarity*

2.1 Main request

The following is unclear in claim 1:

- a) the expression "event process";
- b) how "tasks" are "comprised" in a "program" (lines 16 and 17);
- c) how "event processes" are "included" in a "task" (paragraph starting with "a plurality of tasks");
- d) how an "event process" is "executed" by a "task" (penultimate paragraph: "wherein a plurality of tasks, ... , is adapted to ..., and is adapted to execute said event process ...").

2.1.1 As to a), the board agrees with the examining division that the term "event process" is unclear. The board also endorses the decision of the examining division, section 2.1.3, paragraph 1, and considers that the standard meaning of the term "process" in computing is almost the same as that of the term "task", namely "a program in execution".

In order to show that this meaning of a "process" as "a program in execution" is well-known, the board cites the widely used university textbook D6, page 89, line 6:

"These needs resulted in the notion of a *process*, which is a program in execution."

Page 90, section 4.1.1 "The Process", line 1 says:

"Informally, a *process* is a program in execution."

Paragraph 2 gives a precise definition of a process:

"A process is more than the program code (sometimes known as the *text section*). It also

includes the current activity, as represented by the value of the *program counter* and the contents of the processor's registers. A process generally also includes the process *stack*, containing temporary data (such as subroutine parameters, return addresses, and temporary variables), and a *data section* containing global variables."

The aforementioned definition, given the type of book from which it is extracted, can be considered by the board to be general knowledge of the skilled person in the field of computing.

The use of "process" in the claim might have been influenced by the field of application stated in the original description, page 6, line 15, namely an *engine control unit (ECU) in a vehicle*. Also, the names of some example processes relate to that field, see page 2, line 15:

"The Tdc process is executed whenever the piston of a specific cylinder reaches the top dead center (TDC). The engine stall process is executed when the engine of the vehicle stalls. The fault detection process is executed for detecting a fault for self-diagnosis."

Nevertheless, claim 1 relates to "a processor unit for executing an event process" and thus clearly *belongs to the field of computing*. The potential field of its use contributes in no way to clarifying the claimed subject-matter.

2.1.2 As to b), the board agrees with the appellant's interpretation of the term "task". The passage in the grounds of appeal, page 2, paragraph 4, line 2 reads as follows:

' ... "task" is used for "run", i.e., "(actual) execution" related / as "execution unit", ...'

This corresponds to the definition of "process" in D6. However, such a "task" which is a "run", or a "program in execution" cannot be "comprised by a "program", since - as it is said in D6 and cited above in section 2.1.1 - a "task" (i.e. a "process" in D6) is much more than a "program", it comprises everything a "program" needs to run, i.e. a text section, a program counter, a stack, and a data section. A "task" comprises a "program" (in the text section), but a "program" cannot comprise "tasks".

2.1.3 As to c) and d), the skilled person might imagine two possible interpretations for the "inclusion" and the "execution":

(1) When taking the standard meaning of a "task" or "process" in computing, i.e. a "program in execution":

A "task" (i.e. a "process" in D6) creates a so-called *child process* (which would then correspond to an "event process" in the claim) and starts it asynchronously and concurrently in parallel, or it waits synchronously until the child process has terminated.

See D6, page 97, section 4.3.1, paragraph 1 for the creation of child processes; and page 98, the first two bullet points for the concurrent, i.e. asynchronous, execution of the parent process and the child processes, and for a synchronous waiting behaviour of the parent process.

- (2) When taking a more general meaning of a "process" as stated in D5, page 423, i.e. "a program or part of a program; a coherent sequence of steps undertaken by a program":

A program (which would then correspond to a "task" in the claim) jumps to a subprogram ("event process" in the claim). This is also termed a *subroutine call* and happens synchronously *in the same process*. No new process is created.

2.1.4 The appellant admitted during oral proceedings that the original description does not disclose more than the mere statement that "event processes" are included in and executed by a "task", without giving explanations about that (e.g. description page 9, paragraph 2, sentence 2 for the inclusion, and page 8, last paragraph, sentence 2 for the execution).

2.1.5 The appellant stated during oral proceedings that he interprets the inclusion as an *association* or an *assignment*, but he recognised that he could not find a support for this interpretation disclosed in the description. He continued that figure 2 showed the inclusion of e.g. an "ENGINE STALL PROCESS" and a "FAULT DETECTION PROCESS" in the "B SHARED TASK (MODERATE PRIORITY)".

Further on, there was a *hierarchy* of a "program" composed of "tasks" (original description page 1, paragraph 2, lines 4 and 5) which included "event processes" (page 2, lines 2 and 3).

However, the board finds that this does not clarify the situation. Also Figure 2 and the cited passages do not disclose more than the mere statement that there is an inclusion (or composition) relationship, without telling what this should technically mean.

- 2.1.6 The appellant cited a passage in D6 (page 89, section 4.1, line 2) where it is said that "a time-shared system has *user programs, or tasks*". He continued that this would mean that "program" and "task" were synonymous. Furthermore, this pointed to interpretation (2), i.e. a "task" being a program in the usual sense, and an "event process" being a sub-program, called by the "task"/program.

Questioned by the board whether there is a basis in the description for this interpretation, the appellant could not find any.

Moreover, an identification of a "task" with a program apparently contradicts the aforementioned hierarchy where the program is *composed* of tasks. Furthermore, document D6 says nothing about a "process" being a sub-program, but instead defines a process as a "program in execution" (page 90, section 4.1.1) and "a process is more than the program code" (same section, paragraph 2). This characterises a process as being on a *qualitatively* different level than a program, in contrast to the merely *quantitatively* different levels of program and sub-programs.

2.1.7 Thus, concerning b) no interpretation can be found for the "comprised" relationship. And concerning c) and d), there are at least two possible interpretations, both of them clearly not supported by the description.

2.1.8 The board concludes that claim 1 of the main request is unclear (Article 84 EPC).

2.2 First auxiliary request

The appellant argued that the added expression "[a plurality of tasks (12c, 12d)], each of which causes the processing unit to execute at least one associated event process" clarified that a "task" was a program that executes an "event process" as a sub-program (i.e. interpretation (2)).

However, the board is still of the opinion that an execution relationship between "tasks" and "event processes" is also fulfilled by a child process creation (i.e. interpretation (1)) and does not necessarily imply the program/sub-program scheme (interpretation (2)).

2.2.1 Therefore, claim 1 of the first auxiliary request is also unclear (Article 84 EPC).

2.3 Second auxiliary request

The appellant admitted during oral proceedings that the second auxiliary request did not further clarify the "task" "event process" relationship.

Therefore, claim 1 of the second auxiliary request is also unclear (Article 84 EPC).

2.4 Third auxiliary request

The appellant stated during oral proceedings that the addition of priority levels in the third auxiliary request pointed to interpretation (2).

The board disagrees with this interpretation, since priorities are usually assigned to processes (in the sense disclosed in D6) and not to programs, so that they would even point to interpretation (1).

Moreover, the addition of priority levels introduces a new independent aspect of task scheduling to the claim. The board cannot see how this new aspect could clarify the problems introduced by the already present features.

Therefore, claim 1 of the third auxiliary request is also unclear (Article 84 EPC).

2.5 Fourth auxiliary request

The board agrees that the fourth auxiliary request does not give rise anymore to clarity objections b) and c), but the unclear items a) and d) are still present. The appellant repeated that an "event process" had to be considered as a kind of sub-program executed by a task (interpretation (2)).

This was not convincing to the board since it still cannot see that the original description would disclose which interpretation is the right one.

Therefore, claim 1 of the fourth auxiliary request is also unclear (Article 84 EPC).

3. *Conclusion*

Since claim 1 of the main request and of auxiliary requests 1-4 lacks clarity and auxiliary request 5 was found inadmissible, the appeal is dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairwoman:

B. Atienza Vivancos

M.-B. Tardo-Dino