

Internal distribution code:

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

**Datasheet for the decision
of 22 November 2022**

Case Number: T 1427/17 - 3.5.06

Application Number: 07795841.1

Publication Number: 2030112

IPC: G06F9/46

Language of the proceedings: EN

Title of invention:
RUNTIME EXTENSION FRAMEWORK

Applicant:
Siemens Industry, Inc.

Headword:
Runtime extension framework/SIEMENS

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - (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 1427/17 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 22 November 2022

Appellant: Siemens Industry, Inc.
(Applicant) 100 Technology Drive
Alpharetta, GA 30005 (US)

Representative: Haseltine Lake Kempner LLP
One Portwall Square
Portwall Lane
Bristol BS1 6BH (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted on 10 February
2017 refusing European patent application No.
07795841.1 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman M. Müller
Members: S. Krischer
B. Müller

Summary of Facts and Submissions

- I. The appeal is directed against the decision of the examining division, dated 10 February 2017, to refuse application No. 07795841 for added subject-matter (Article 123(2) EPC), lack of support by the description and lack of clarity (Article 84 EPC). In a section entitled "Obiter dictum", the decision also contains objections concerning lack of inventive step over common general knowledge (Article 56 EPC).

Document D1 was referred to in communications of the examining division; D2 and D3 were cited in the search report:

D1 US 2004/034860 A1
D2 WO 00/70507
D3 US 2003/056024 A1

- II. A notice of appeal was received on 7 April 2017. The appeal fee was paid the same day. A statement of grounds of appeal was received on 8 June 2017. Sets of claims according to a main and a first auxiliary request were filed.
- III. In its preliminary opinion, the board noted that the objection under Article 123(2) EPC raised in the appealed decision was overcome by way of deleting the feature that was objected to, but raised two new objections concerning Article 123(2) EPC. The board left open whether the claims were clear and supported by the description (Article 84 EPC), but gave reasons

why the claims lacked an inventive step over D1 in combination with D2 or D3.

- IV. With a letter dated 10 November 2022, the appellant submitted further arguments and filed two new sets of claims to replace the pending ones. Its final requests were that the decision under appeal be set aside and that a patent be granted on the basis of the claims according to the main request or the auxiliary request as filed with its letter of 10 November 2022.
- V. Oral proceedings were held on 22 November 2022. At the end, the chairman announced the board's decision.
- VI. Claim 1 of the main request reads as follows:

"1. A runtime extension framework architecture (300) for a system, comprising:

an event manager (308) configured to coordinate between a plurality of event handlers and events received from a human machine interface (2A10, 304);

a plurality of external event handlers (306-306n) configured to receive and process events associated with implementing a protocol translation for at least one application program interface associated with a respective component (2A40, 2B30, 304n), the protocol translation supporting an extension of application behavior of the human machine interface (2A10, 304) so that it behaves as a protocol translation layer, wherein each of the plurality of external event handlers (306-306n) is subscribed to receive a respective event or category of event from the event manager (308), and each of the plurality of external event handlers (306-306n) is configured to perform a protocol translation associated with the respective event or category of event to which that external event

handler is subscribed by translating tag values of commands from a batch manager application of the system that have no inherent meaning to the respective component (2A40, 2B30, 304n) into tag values that the respective component (2A40, 2B30, 304n) can interpret; and

at least one internal event handler (310-31 On) configured to receive and process responses to events related to the extension of the human machine interface (2A10, 304)."

VII. Claim 1 of the auxiliary request reads as follows:

"1. A runtime extension framework architecture (300) for a distributed control system, DCS, comprising:

an event manager (308) configured to coordinate between a plurality of event handlers and events received from a human machine interface (2A10, 304);

a plurality of external event handlers (306-306n) configured to receive and process events associated with implementing a protocol translation between a batch manager application (2A00, 2B40, 304n) and DCS controllers (2A40, 2B30, 304n), the protocol translation supporting an extension of application behavior of the human machine interface so that it behaves as a protocol translation layer, wherein each of the plurality of external event handlers is subscribed to receive a respective event or category of event from the event manager (308), and each of the plurality of external event handlers (306-306n) is configured to perform a protocol translation associated with the respective event or category of event to which that external event handler is subscribed by:

observing a change written to a tag stored in a data manager (2A11, 2B11) of the human machine

interface (2A10, 304) by a batch command from the batch manager application (2A00, 2B40, 304n), wherein the batch command has no inherent meaning to the DCS controller for which the batch command is issued, and writing a corresponding change to a corresponding tag of the DCS controller that the DCS controller can interpret, and

observing a change originating in the tag of the DCS controller, and writing a corresponding change to the tag stored in the data manager (2A11, 2B11) that the batch manager application (2A00, 2B40, 304n) can interpret; and

at least one internal event handler (310-31 On) configured to receive and process responses to events related to the extension of the human machine interface (2A10, 304)."

Reasons for the Decision

1. *Summary of the invention*

- 1.1 The invention relates to a program for handling events from a human machine interface (HMI; see original description, [19]; i.e. events produced by a human operator via the HMI) or an application program interface (API, [19], i.e. events produced by other programs). For example, the program is suitable for use in the field of manufacturing systems, such as "Distributed Control Systems" (DCS, [2]). The event handling program contains an "event manager" (figure 1: 108) that coordinates plug-in event handlers (106, 110; see [21]) and forwards events inside the system ([21], second sentence; see also the horizontal arrows in figures 1 and 3-7). The invention distinguishes between internal

(110, 110n; [20]) and external (plug-in) event handlers (106, 106n; [19]).

- 1.2 An external event handler receives external events through an event manager host (see figure 1: vertical arrow between External Event 104 and External EventHandler1 106; [28], second and third sentences). It responds to events from the outside world ([21], fourth sentence), such as a DCS Tag Communication Layer 304 (see figure 3; [28], second sentence). However, "while there is no real difference between internal event handlers 110 and external event handlers 106, they are conceptually different in that they [i.e. the external event handlers] generally may include various use interfaces", such as the well-known "WinCC API" ([21], fifth and sixth sentences).
- 1.3 An external event handler also serves as so-called "protocol translation layer" to translate "commands" from a "batch manager" (not claimed) to legacy DCS controllers, or vice versa ([22], first sentence; [23], sentences 4-9). The protocol translation is triggered by an event caused by a "batch relevant" change of a tag that has been modified either by the batch manager or by a DCS controller (sentences 6-8). This protocol translation was not contained in the original claims and is described in detail only with respect to figures 2A and 2B (i.e. in [22]-[27]).
- 1.4 The internal event handlers play no role in this translation, but process events which are "responses to events related to the actual extension of the system" (see the first sentence of [20], [29], [31], [33], [35], [37] and [39]). Such a response event may be

produced by an operator via the HMI ([20], second sentence).

2. *Inventiveness*

2.1 In the following, only the auxiliary request is analysed, since the subject-matter of its claim 1 is narrower than that of claim 1 of the main request. The board's conclusion that the former lacks inventive step therefore also applies to the latter.

2.2 In its preliminary opinion, the board based its inventive-step objection on D1.

2.3 D1 discloses a plug-in framework for dynamic extensions to an application program (see [4]) including external event handlers ([39]: the so-called "protocol adapters" listening to specific events) and a centralised event manager ([39]: "event dispatcher").

2.4 D1 lacks the context of distributed control systems (DCS) and the communication between batch manager and DCS controllers over a "data manager" program (managing tags as shared memory, see 2A11 in figure 2A, 2B11 in figure 2B (both also in claim 1), and 304 in figure 3). D1 also does not disclose that the external event handlers translate values in tags of the data manager when an event signals that the content of the tag memory cell has been modified by either a batch manager or a DCS controller.

2.5 For this reason, the board considered inventive step starting from the prior art disclosed in the present patent application itself.

- 2.6 In its background section (see sections [2]-[4]), the application discloses Distributed Control Systems and its components, including DCS controllers, a Human/Machine Interface (HMI) and a batch manager. It also refers to "communication protocols" as the "particular language" these components "speak". In section [21], the description refers to a so-called "WinCC Data Manager" as a well-known component, and states that it "use[s] a WinCC API for accessing tag data from WinCC". With a view to claim 1 of auxiliary request 1, the board therefore takes the application to disclose as prior art a DCS system, including an HMI, in which communication between a batch manager and DCS controllers occurs by the modification of tags in a shared data manager, and events indicating the change of any of these tags. The board also interprets the rather broad term "protocol" in the claims to refer essentially to the choice of tags in the mentioned communication and the term "protocol translation" to the (back-and-forth) translation of these tags.
- 2.7 The appellant did not challenge the board's understanding that the application disclosed a - still rather generic - DCS system as just described.
- 2.8 The application does not disclose that the use of several event handlers, the distinction between external and internal event handlers, or the claimed "protocol translation" of tag values are known.
- 2.9 The board considers at least obvious in a prior-art system as defined the use of events to communicate the change of tags between the batch manager and the DCS controllers. Moreover, it considers to be common practice in programming to spread different functions over different sub-programs. Specifically, it considers

obvious to provide different event handlers for different events and to provide different types of event handlers for different types of events. As regards the internal event handlers, the board further notes that neither the claims nor the description exhibit a clear difference between the internal and the external event handlers. These features thus cannot establish an inventive step.

- 2.10 The problem to be solved by the tag (or "protocol") translation may be regarded as how to make interoperable two programs (a batch manager and a DCS controller) that communicate by events through a data manager, but use different "protocols" to encode their data. This also corresponds to the problem addressed in the application itself (see sections [4] and [5]).
- 2.11 The solution proposed cannot be considered as involving an inventive step for two reasons. Firstly, it is an obvious measure to use "translation" so that components "speaking" different "languages" can understand, and thus cooperate with, each other. And secondly, the translation of data during event or message handling is well-known, see for example D3, [179]-[181] (an event handler translating keyboard codes).
- 2.12 As to the argument in the grounds of appeal (24.) that the effect of the invention consists in extending a program (the HMI application) without the program having been designed for such an extension, the board notes that this effect is always present in programs communicating by events, since event handling decouples them in a transparent manner.

2.13 Therefore, the subject-matter of claim 1 of the auxiliary request is not inventive within the meaning of Article 56 EPC 1973.

2.14 It follows that the subject-matter of broader claim 1 of the main request is not inventive, either.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



L. Stridde

Martin Müller

Decision electronically authenticated