BESCHWERDEKAMMERN
DES EUROPÄISCHEN
PATENTAMTS

BOARDS OF APPEAL OF
THE EUROPEAN PATENT
OFFICE

CHAMBRES DE RECOURS
DE L'OFFICE EUROPÉEN
DES BREVETS

Internal distribution code:

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

Datasheet for the decision of 10 June 2021

Case Number:	T 0834/17 - 3.5.03
Application Number:	11153373.3
Publication Number:	2485457
IPC:	H04L29/08
Language of the proceedings:	EN

Title of invention:
A method for keeping a web session alive in a web application

Applicant:
Siemens Aktiengesellschaft

Headword:
Keeping web sessions alive/SIEMENS

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - main and auxiliary requests (no): obvious combination of known features

This datasheet is not part of the Decisior
It can be changed at any time and without notic

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 0834/17 - 3.5.03

DECISION
of Technical Board of Appeal 3.5.03
of 10 June 2021

Appellant:
(Applicant)

Siemens Aktiengesellschaft
Werner-von-Siemens-Straße 1
80333 München (DE)

Representative:

Siemens Patent Attorneys
Postfach 22 16 34
80506 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 14 December 2016 refusing European patent application No. 11153373.3 pursuant to Article 97(2) EPC.

Composition of the Board:

Chair	K. Bengi-Akyürek
Members:	J. Eraso Helguera
	N. Obrovski

Summary of Facts and Submissions

I. The appeal was lodged by the applicant against the decision of the examining division ("decision according to the state of the file") to refuse the present European patent application for lack of inventive step with respect to the claims of a main request.

II. During the examination proceedings, the examining division referred *inter alia* to the following prior-art documents:

D2: Sheridan, Malcom: "Keeping Your ASP.NET Session Alive Using jQuery", 25 January 2010, XP002648567, pp. 1-3;

D4: Siemens: "Configuration Instruction SIMATIC PCS 7 - SIMATIC IT - Integration SIMATIC Client Application Builder", 18 June 2008, XP055159100, pp. 1-81.

III. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the claims of either of a **main request** and an **auxiliary request**, both filed with their reply to a communication pursuant to Rule 100(2) EPC issued by the board.

IV. In a communication pursuant to Article 15(1) RPBA 2020, the board stated its preliminary opinion on both requests.

V. In their reply to the board's communication pursuant to Article 15(1) RPBA 2020, the appellant withdrew their request for oral proceedings and requested a "decision according to the state of the file".

VI. The board then cancelled the oral proceedings.

VII. Claim 1 of the **main request** reads as follows:

"A method for keeping a web session alive in a web application running in a manufacturing execution system MES, comprising the steps of:

a) providing a number of data processing units (4, 6, 8, 10) running the MES software for controlling and/or monitoring a production process operating a number of production components (12 to 24); wherein one of the data processing units being executed as master console (4);

b) providing in the master console (4) a client application builder (CAB) wherein the client application builder (CAB) provides an interface to create a web application and to customize an [*sic*] user program consistently implemented within the MES software or stemming from an external sources [*sic*];

c) enabling the client application builder (CAB) to offer at least one software-coded mechanism to keep the web session alive for a customizable period of time; said client application builder (CAB) comprising a library (L) that comprises a first part that manages a specific portion of code to be added to a web page of said web application and a second part that manages the responses to requests from a web server that is deriving from the portion of code added to the web page;

d) at engineering level during the development of a page of the web application, using the library (L) to assign the desired mechanism to the page of the web application, wherein said assignment adds the respective specific portion of code to the web page of said web application; and

e) at runtime of the web application, executing the specific portion of code added in order to answer the requests from a web server in the dependency of the desired mechanism."

Claim 1 of the **auxiliary request** reads as follows (board's highlighting indicating amendments vis-à-vis claim 1 of the main request):

"A method for keeping a web session alive in a web application running in a manufacturing execution system MES, comprising the steps of:

a) providing a number of data processing units (4, 6, 8, 10) running the MES software for controlling and/or monitoring a production process operating a number of production components (12 to 24); wherein one of the data processing units being executed as master console (4);

b) providing in the master console (4) a client application builder (CAB) wherein the client application builder (CAB) provides an interface to create a web application and to customize an [*sic*] user program consistently implemented within the MES software or stemming from an external sources [*sic*];

c) enabling the client application builder (CAB) to offer at least one software-coded mechanism to keep the web session alive for a customizable period of time; said client application builder (CAB) comprising a library (L) that comprises a first part that manages a specific portion of code to be added to a web page of said web application and a second part that manages the responses to requests from a web server that is deriving from the portion of code added to the web page;

d) at engineering level during the development of a page of the web application, using the library (L) to assign the desired mechanism to the page of the web application, wherein said assignment adds the respective specific portion of code to the web page of said web application, wherein said assignment adds the respective specific portion of code to the web page of said web application and the desired mechanism is selected from a mechanism group consisting of:
a) an always on mechanism,
b) an always off mechanism and
c) an activate/deactivate mechanism related to an event, i.e. a specific button click; and

e) at runtime of the web application, executing the specific portion of code added in order to answer the requests from a web server in the dependency of the desired mechanism."

Reasons for the Decision

1. MAIN REQUEST

Claim 1 of the **main request** comprises the following limiting features:

A method for keeping a web session alive in a web application running in a manufacturing execution system MES, comprising the steps of:

a) providing a number of data processing units running the MES software for controlling and/or monitoring a production process operating a number of production components; wherein one of the data processing units being executed as master console;

b) providing in the master console a client application builder wherein the client application builder provides an interface to create a web application and to customise a[n] user program consistently implemented within the MES software or stemming from an external source[s];

c) enabling the client application builder to offer at least one software-coded mechanism to keep the web session alive for [a] customisable period of time; said client application builder comprising a library that comprises a first part that manages a specific portion of code to be added to a web page of said web application and a second part that manages the responses to requests from a web server that is deriving from the portion of code added to the web page;

d) at engineering level during the development of a page of the web application, using the library to assign the desired mechanism to the page of the web application, wherein said assignment adds the respective specific portion of code to the web page of said web application;

e) at runtime of the web application, executing the specific portion of code added in order to answer the requests from a web server in the dependency of the desired mechanism.

1.1 Claim 1 - Inventive step (Articles 52(1) and 56 EPC)

1.1.1 Document **D4** was introduced by the examining division in the examination proceedings with a communication pursuant to Article 94(3) EPC dated 23 December 2014. D4 was also subsequently mentioned in the annex to the summons to oral proceedings issued by the examining division to which its "decision according to the state of the file" refers. This document relates to the specific architecture of the MES system and is to be considered as the closest prior art for the subject-matter of present claim 1.

Using the wording of claim 1, document **D4** discloses:

A method for keeping a web session alive in a web application running in a manufacturing execution system MES, comprising the steps of:

a) providing a number of data processing units (see p. 5, Fig. 1-1: "SIMATIC IT" components) running the MES software for controlling and/or monitoring a production process operating a number of production components (see p. 1: "Sensors, Actuators"); wherein one of the data processing units being executed as master console (see p. 5, Fig. 1-1: "Report Manager CAB engineering"; see p. 39, third paragraph: "... the CAB environment (CAB Server, CAB Webserver) is placed on a special CAB machine.");

b) providing in the master console a client application builder wherein the client application builder provides an interface to create a web application and to customise a user program consistently implemented within the MES software or stemming from an external source (see p. 39, second paragraph: "... CAB is composed of a set of modules, which allow the user to build GUIs (fully integrated with SIMATIC IT Production Suite) in a Web application and display the Web pages in a Web Browser. It collects data from heterogeneous sources, manipulates and aggregates these data before visualization ...");

c1) enabling the client application builder to offer at least one software-coded mechanism to keep the web session alive for a customisable period of time (see p. 75, Fig. 3-32: "Session timeout: 20 minutes"; see p. 75, first paragraph: " ... Additionally the default timeouts for the session and for scripts might be changed. The session timeout defines, after which time the connection to the web server closes after the last interaction ...").

1.1.2 The subject-matter of claim 1 thus differs from the disclosure of D4, besides **features d) and e)**, in (board's outline):

c2) said client application builder comprising a library that comprises a first part that manages a specific portion of code to be added to a web page of said web application and a second part that manages the responses to requests from a web server that is deriving from the portion of code added to the web page.

1.1.3 According to the present application as filed, the technical effect achieved by these distinguishing features is that the web session running in a web application can be kept alive for a predetermined amount of time *without* having to increase the session Timeout value in the web application's "ASP.NET's web.config file" or to ask the user to refresh their browser at regular intervals (see page 5, lines 25 to page 6, line 3 of the description as filed).

1.1.4 The associated objective technical problem can thus be framed as "how to keep a web session in a web application alive for a predetermined amount of time and manage individually the session timeout characteristics of the web session in the web application" (see page 6, lines 5-10 of the description as filed).

The solution proposed in claim 1, however, does not involve an inventive step (Articles 52(1) and 56 EPC) for the following reasons:

1.1.5 The person skilled in the field of communication networks, starting out from D4 and confronted with the above objective problem, would have consulted prior-art document **D2**, which aims at solving the objective problem in a generic ASP.NET web application (see page 1, first paragraph):

> "When you're working with the ASP.NET Session, it's important to remember that the session can timeout. The time before timing out is normally configured in the web.config file. Sometimes sessions time out at the most inconvenient time for your users ...".

D2 further discloses a library that comprises a first part (see page 3, lines 1-3 : "function KeepSessionAlive(){ ...") that manages a specific portion of code to be added to a web page of said web application and a second part (see page 1, third paragraph to page 2, line 4):

> "... I've added my handler and named it KeepSessionAlive.ashx. Here's the code for the handler: ..."

that manages the responses to requests from a web server that is deriving from the portion of code added to the web page.

D2 also discloses feature d) of claim 1 (see page 2, last paragraph to page 3, line 8:

> "... let's add some JavaScript to call this piece of code at set intervals. To do this I'm using the <u>setInterval</u> function. Here's the client side code: ...",

and feature e), see page 3, lines 9-15:

> "... When the page loads, I'm running setInterval to run my function every 10 seconds ... This means every 10 seconds the session will be updated behind the scenes. This will give the impression to the user that their session will not time out ...".

1.1.6 Hence, the skilled person not only could but also would readily envisage the introduction of the measures according to distinguishing features c2), d) and e) into the "client application builder" of the specific ASP.NET environment of **D4** without the involvement of any inventive skills.

1.1.7 In their reply to the board's communication under Rule 100(2) EPC, the appellant identified **D4** as EP 1 670 213 A1 ("Verifying and maintaining connection liveliness in a reliable messaging for web services environment"), and provided arguments for the presence of inventive step in view of this document and D2.

These arguments are moot, at least to the extent that <u>they are based on a piece of evidence different from the one actually used by the board and referred to as **D4**</u>. After having being notified of this circumstance in the board's communication pursuant to Article 15(1)

RPBA 2020, the appellant refrained from making further comments on the document identified by the board as D4.

1.1.8 For the sake of completeness, the board will address the following points raised by the appellant in section 2 of their reply:

> "*The appellant raises the following points:*
> *i) Features (=method steps) a), b) c) c2 d) and e) as depicted above are not contained in D4.*
> *ii) Features a) and b) are obviously not in D2.*
> *iii) The setInterval method according to D2 is static, since it runs only every 10 sec.*
> *iv) It has to be admitted, that the person skilled in the art would easily come to a dynamic setInterval method beyond a fixed value as given in D2. But D2 is silent how this is done in view of the features c) and c2).*"

As indicated above, point i) is moot because the appellant's arguments relate to a different document. The board agrees with point ii). However, as it is apparent from the preceding analysis, features a) and b) are already present in **D4**, which is used as starting point for the assessment of inventive step. This has not been challenged by the appellant.

With respect to points iii) and iv), **D2** discloses a software-coded mechanism to keep the web session alive for a <u>customisable</u>, i.e. predetermined (see point 1.1.4 above), period of time (see D2, page 3, lines 9-14, emphasis added: "... When the page loads, I'm running setInterval to run my function every 10 seconds: setInterval(KeepSessionAlive, <u>10000</u>) ... This means every 10 seconds the session will be updated behind the scenes ..."). The duration of the period of time is necessarily set individually, because the value ("10000" in this case) is explicitly included in the code added to the served page, and if need be, it may be changed from page to page. Hence, the alleged difference implied by the appellant ("dynamic" vs "fixed") cannot be acknowledged.

1.2 It follows that the main request is not allowable under Articles 52(1) and 56 EPC.

2. AUXILIARY REQUEST

Claim 1 of the **auxiliary request** comprises all the limiting features of claim 1 of the main request and adds the following additional features in feature d):

d2) wherein said assignment adds the respective specific portion of code to the web page of said web application and

d3) the desired mechanism is selected from a mechanism group consisting of:
a) an always on mechanism,
b) an always off mechanism and
c) an activate/deactivate mechanism related to an event.

2.1 Claim 1 - Inventive step (Articles 52(1) and 56 EPC)

2.1.1 The board holds that feature d2) is also disclosed in **D2**: the Javascript code containing the function setInterval is added to the loaded page (see page 2, last paragraph to page 3, line 8):

> "... let's add some JavaScript to call this piece of code at set intervals. To do this I'm using the setInterval function. Here's the client side

code: ...",

and page 3, line 9: "... When the page loads, I'm running setInterval to run my function ..."). D2 also discloses at least an "always-on mechanism" (see page 3, lines 14-15: "... This will give the impression to the user that their session will not time out ..."), and, consequently, the first of the three alternatives proposed in feature d3).

2.1.2 Since the additional features d2) and d3) are also disclosed by D2, the subject-matter of claim 1 of the auxiliary request does not involve an inventive step (Article 56 EPC) in view of the straightforward combination of **D4 and D2** for the same reasons as stated above for claim 1 of the main request.

2.1.3 The appellant's arguments with respect to the auxiliary request are also moot to the extent that they are based on a different piece of evidence.

2.1.4 In conclusion, the auxiliary request is not allowable under Articles 52(1) and 56 EPC either.

3. As there is no allowable claim request, it follows that the appeal must be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chair:

Beschwerdekammern des Europäischen Patentamts

Boards of Appeal of the European Patent Office

Chambres de recours de l'Office européen des brevets

B. Brückner K. Bengi-Akyürek

Decision electronically authenticated