(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 628 618 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
01.04.2020 Bulletin 2020/14

(51) Int Cl.:
B66B 1/34 *(2006.01)*

(21) Application number: **18196874.4**

(22) Date of filing: **26.09.2018**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(71) Applicant: **Inventio AG**
6052 Hergiswil (CH)

(72) Inventors:
- **SONNENMOSER, Astrid**
  6280 Hochdorf (CH)
- **HOSEMANN, Axel**
  5644 Auw (CH)
- **ZHU, Zack**
  6340 Baar (CH)
- **AKERET, Joel**
  8050 Zürich (CH)
- **TRAUBER-PICHLER, Marko**
  8152 Opfikon (CH)

(54) **ELEVATOR SYSTEM**

(57) The present invention relates to an elevator system (100), comprising a replaceable software module (101-1) for generating an elevator command (103) and putting the elevator command (103) in a command queue (105); and a basic software module (101-2) for fetching the elevator command (103) from the command queue (105) and controlling the elevator (107) on the basis of the elevator command (103).

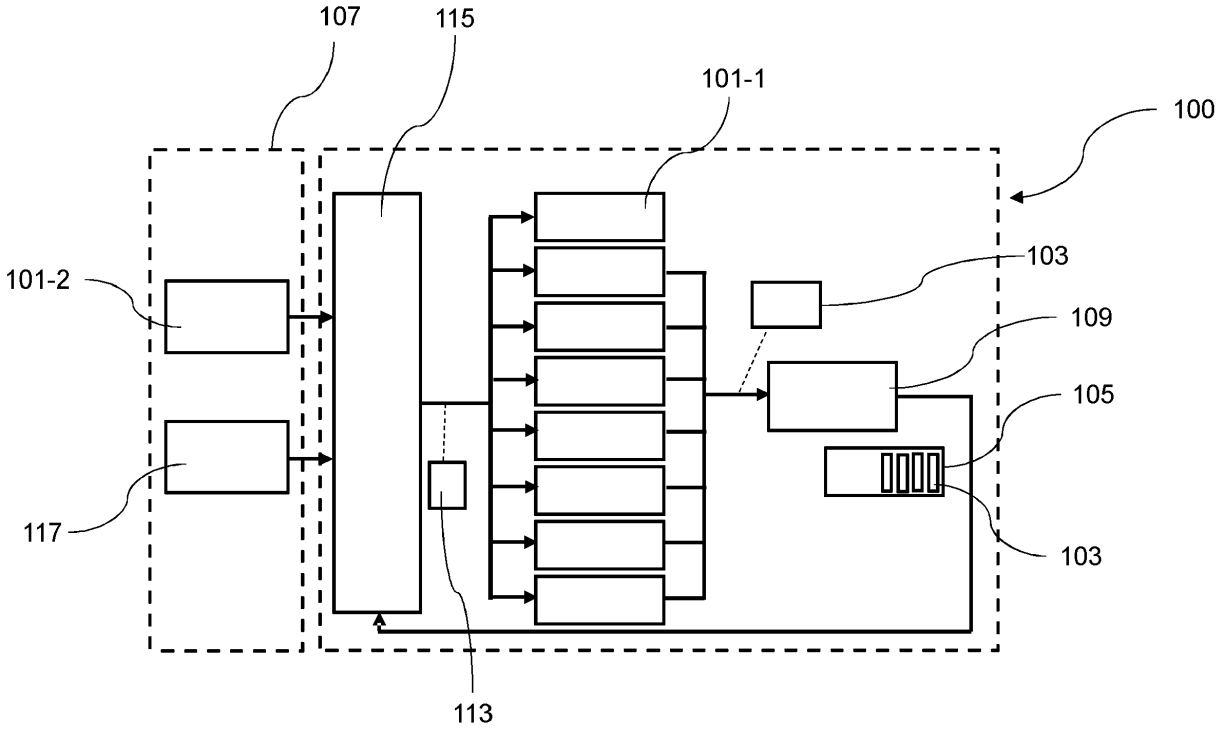

Fig. 1

EP 3 628 618 A1

Description

[0001] The present invention relates to an elevator system, a method for controlling an elevator system and a computer program.

[0002] Today's elevator systems comprise a monolithic software structure implemented by a single program for controlling the elevator. In these elevator systems, additional functions and specific elevator behavior can be integrated and tested only with high effort. Once having installed the original elevator system with the program, it is not easily possible to modify or add specific functions to the elevator system at a later point of time. Thus, a later technical modification of the elevator system is based on slow and complex configuration processes.

[0003] Thus, it is an object of the present invention to technically improve and to simplify a configuration of an elevator system.

[0004] This object is solved by subject-matter according to the independent claims. Technically beneficial embodiments are subject to the dependent claims, the description and the drawings.

[0005] According to a first aspect this object is solved by an elevator system, comprising a replaceable software module for generating an elevator command and putting the elevator command in a command queue; and a basic software module for fetching the elevator command from the command queue and controlling the elevator on the basis of the elevator command. This elevator system has the technical advantage that it can be modified and configured easily, since a monolithic software structure is avoided. Instead one or more replaceable software modules can be used flexibly within the elevator system. However, the basic software module executes the elevator commands and enables a safe operation of the elevator system. The replaceable software modules form the functional part, whereas the basic software module forms the safety part of the control software of the elevator system. The basic software module and the replaceable software modules are coupled via the command queue as a data interface.

[0006] In a technically advantageous embodiment of the elevator system, the basic software module is adapted to control an elevator drive, an elevator brake, a door drive and/or an elevator cabin light. This yields the technical advantage that specific elevator features can be controlled by the basic software module on demand of the elevator commands.

[0007] In a further technically advantageous embodiment of the elevator system, the basic software module is adapted to test the elevator command before executing the elevator command. This yields the technical advantage that security is increased, and malfunctions can be avoided.

[0008] In a further technically advantageous embodiment of the elevator system, the basic software module is adapted to test whether the elevator command is within predefined execution limits. This yields the technical advantage that testing and admitting of elevator commands can be performed with low effort.

[0009] In a further technically advantageous embodiment of the elevator system, the basic software module is adapted to output a feedback message if an elevator command is not executed. This yields the technical advantage that a user is informed about non-execution.

[0010] In a further technically advantageous embodiment of the elevator system, the replaceable software module is adapted to assign a priority level to the elevator command. This yields the technical advantage that elevator commands can be executed in a defined priority order.

[0011] In a further technically advantageous embodiment of the elevator system, the basic software module is adapted to execute the elevator command based on the priority level. This yields also the technical advantage that elevator commands can be executed in a defined priority order.

[0012] In a further technically advantageous embodiment of the elevator system, the elevator system has a network interface for downloading a software module to be replaced. This yields the technical advantage that software modules can be replaced or updated easily with low technical effort.

[0013] In a further technically advantageous embodiment of the elevator the elevator system is adapted to re-sequence existing elevator commands within the command queue in dependence on the priority level of a newly inserted elevator command. This yields the technical advantage that elevator commands that are newly put in the command queue can be executed fist.

[0014] According to a second aspect, this object is solved by a method for operating an elevator system, comprising the steps of generating an elevator command by a replaceable software module; putting the elevator command in a command queue; fetching the elevator command from the command queue by a basic software module; and controlling the elevator on the basis of the elevator command. The same technical advantages are achieved by this method as by the elevator system according to the first aspect.

[0015] In a technically advantageous embodiment of the method, the basic software module controls an elevator drive, an elevator brake, a door drive and/or an elevator cabin light. This yields also the technical advantage that specific elevator features can be controlled by the basic software module on demand of the elevator commands.

[0016] In a further technically advantageous embodiment of the method, the basic software module tests the elevator command before executing the elevator command. This yields also the technical advantage that security is increased, and malfunctions can be avoided.

[0017] In a further technically advantageous embodiment of the method, the basic software module tests whether the elevator command is within predefined execution limits. This yields also the technical advantage

that testing and admitting of elevator commands can be performed with low effort.

[0018] In a further technically advantageous embodiment of the method, the replaceable software module assigns a priority level to the elevator command. This yields also the technical advantage that elevator commands can be executed in a defined priority order.

[0019] In a further technically advantageous embodiment of the method, the basic software module executes the elevator command based on the priority level. This yields also the technical advantage that elevator commands can be executed in a defined priority order.

[0020] According to a third aspect, this object is solved by a computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method according to the second aspect. The same technical advantages are achieved by the computer program as by the method according to the second aspect.

[0021] Specific embodiments are shown in the figures and are explained in the following.

Fig. 1 shows a systematic view of an elevator system;

Fig. 2 shows steps when setting and fetching elevator commands in a command queue;

Fig. 3 shows a structure of the elevator system; and

Fig. 4 shows a block diagram of a method for controlling an elevator system.

[0022] Fig. 1 shows a systematic view of an elevator system 100. The elevator system 100 serves for controlling and operating an elevator 107 that vertically moves people or goods between floors, levels or decks. Floors to move the elevator 107 to can be manually selected by pressing a corresponding button of the elevator system 100. The elevator 107 comprises fixtures 117.

[0023] The elevator system 100 has in general numerous technical features that can be subject to a technical control, like an elevator drive, an elevator brake, a door drive and/or an elevator cabin light. In general, also other features of the elevator system can be controlled in a specific manner.

[0024] Direct controlling is performed by a basic software module 101-2. To this end elevator commands 103, i.e. control commands, are used that are transmitted to the basic software module 101-2. The basic software module 101-2 processes the elevator commands 103 and performs an elevator control as requested within the elevator commands 103.

[0025] The elevator commands 103 are digital datasets describing the task to be executed by the elevator system 100. The elevator commands 103 define the requested elevator behavior. Elevator commands 103 comprise for example:

- driving commands / go to floor x:
- disable incoming requests;
- delete all current requests;
- switch on/off cabin light; and/or
- open/close elevator doors.

[0026] The elevator commands 103 are generated by various replaceable software modules 101-1 (App - Applications) that are executed based on a common underlying operating system. The software modules 101-1 can be embedded in the underlying operating system. The software modules 101-1 can be replaced and adapted easily to address specific demands of the elevator system 100.

[0027] Several software modules 101-1 can be active simultaneously in a multitasking fashion. Nevertheless, parallel execution of specific software modules 101-1 can also be set as inadmissible. In addition, an activation of software modules can be changed during normal operation of the elevator system 100. The replaceable software modules 101-1 (App) are supplemental and can be replaced and installed as individually required within the elevator system 100.

[0028] For example, one of the software modules 101-1 serves for receiving user requests for moving the elevator 107 to a particular floor, while another software module 101-1 serves for adapting and controlling cabin light of the elevator system 100. Each of the software modules 101-1 is independent from the others and generates specific elevator commands 103 based on incoming event objects 113, like cabin calls, floor calls, emergency calls or VIP calls or cloud calls. In addition, the elevator status can be considered by the software modules 101-1 for generating elevator commands 103, like elevator 107 moves upward, elevator 107 stands on floor X, or elevator 107 not available.

[0029] A command scheduler 109 takes the generated elevator commands 103 and puts the elevator command 103 in a command queue 105. The command queue 105 is a particular kind of data collection in which the elevator commands 103 in the collection are kept in order and the operations on the collection are the addition of elevator commands 103 to a selected position, known as enqueue, or removal of elevator command 103 from a position, known as dequeue. The command queue 105 is the data interface between the replaceable software modules 101-1 and the basic software module 101-2. Elevator commands 103 are passed from the software modules 101-1 via the command queue 105 to the basic software module 101-2. The command scheduler 109 is in connection with a connector 115.

[0030] The basic software module 101-2 takes the elevator command 103 from the command queue 105 and performs a controlling of the elevator 107 on the basis of

the elevator command 103. The basic software module 101-2 determines whether the picked elevator command 103 can be executed or not, for example due to safety or operational reasons. If for example the doors of the elevator are open, a "goto" elevator command 103 is not executed. Therefore, all elevator command 103 are automatically examined within the basic software module 101-2 with respect to safety requirements.

[0031] If the elevator command 103 is not executed, the basic software module 101-2 generates and returns a corresponding feedback message (not acknowledge). This feedback message can be processed again by the corresponding software module 101-2 that generated the elevator command 103. In addition, a warning message can be output to a user.

[0032] Fig. 2 shows steps when setting in and fetching elevator commands 103 from the command queue 105. In response to a call event 111 the software modules 101-1 generates a "goto" elevator command 103.

[0033] The "goto" elevator command 103 is enqueued in the command queue 105 that already contains other elevator commands 103.

[0034] A priority level can be assigned to each elevator command 103. This priority level determines how and where the elevator command 103 is enqueued within the command queue 105 or the position of the elevator command 103 within the command queue 105. Existing elevator commands 103 within the command queue 105 can be re-sequenced in dependence on the priority level of a newly inserted elevator command 103. Re-sequencing can be performed in an increasing or decreasing order of the priority level within the command queue.

[0035] In this way, a priority and an order of the elevator command 103 within the command queue 105 can be established and maintained. If the elevator command 103, like the "goto" elevator command, is on the first position of the command queue 105 it is processed and executed by the basic software module 101-2. After being executed, the elevator command 103 is dequeued from the command queue 105. Then the next elevator command 103 is fetched from the command queue 105. In this way all elevator commands 103 are executed subsequently.

[0036] Fig. 3 shows a structure of the elevator system 100. Incoming events 113 given by signals trigger the software module 101-1 to generate the elevator command 103. The elevator command 103 is transferred to the command scheduler 109 that places the elevator command 103 in the requested order in the command queue 105 with the requested logic. From this command queue 105, the elevator commands 103 are transferred to the basic software module 101-2 that tests and executes the task as specified by the elevator command 103. After that, the basic software module 101-2 returns feedback and status information to the software module 101-1, like information on the successful execution.

[0037] Fig. 4 shows a block diagram of a method for controlling an elevator system 100. The method for operating an elevator system 100 comprises the step S101 of generating an elevator command 103 by a replaceable software module 101-1; the step S102 of putting the elevator command 103 in the command queue 105; the step S103 of fetching the elevator command 103 from the command queue 105 by a basic software module 101-2; and the step S104 of controlling the elevator 107 on the basis of the elevator command 103.

[0038] Generic functions, that are likewise requested by many software modules 101-1, like call acknowledgement, can again be implemented by special purpose software modules 101-1 also running on the underlying operating system and in communication with the command queue 105. Behavior of these special purpose software modules 101-1 can be controlled by means of transferring arguments or parameters. These special purpose software modules 101-1 can generate elevator commands 103 in dependence on the received arguments or parameters.

[0039] Replaceable software modules 101-1 can be downloaded via a network interface and can be installed within the elevator system in replacement or addition to existing software modules 101-1. The interface is for example an internet interface or a network interface for downloading software modules 101-1 from an external server.

[0040] Replaceable software modules 101-1 can be programmed and developed by independent suppliers. Local or individual adaptations to customers' needs can be carried out easily, while overall safety is maintained.

[0041] The software modules 101-1 and 101-2 can be executed as computer programs on a microcontroller of the elevator system 100 that has a processor for executing the instructions of the software modules 101-1 and 101-2 and a storage for storing the software modules 101-1 and 101-2 and the command queue 105.

[0042] All features discussed in the description or shown in the figures with respect to particular embodiments of the invention can be provided in various combinations in order to simultaneously realize the beneficial technical effects.

[0043] All method steps can be implemented by means that are adapted for carrying out the corresponding method step. All functions carried out by a specific structural feature can be a method step of a method.

[0044] The legal scope of protection of the present invention is defined by the claims and is not reduced by features discussed in the description or shown in the figures.

REFERENCE SIGNS

[0045]

100	elevator system
101-1	replaceable software module
101-2	basic software module
103	elevator command

105 command queue
107 elevator
109 command scheduler
111 call event
113 event object
115 connector
117 fixture

Claims

1. Elevator system (100), comprising:

   - a replaceable software module (101-1) for generating an elevator command (103) and putting the elevator command (103) in a command queue (105); and
   - a basic software module (101-2) for fetching the elevator command (103) from the command queue (105) and controlling the elevator (107) on the basis of the elevator command (103).

2. Elevator system (100) according to claim 1, wherein the basic software module (101-2) is adapted to control an elevator drive, an elevator brake, a door drive and/or an elevator cabin light.

3. Elevator system (100) according to one of the preceding claims, wherein the basic software module (101-2) is adapted to test the elevator command (103) before executing the elevator command (103).

4. Elevator system (100) according to claim 3, wherein the basic software module (101-2) is adapted to test whether the elevator command (103) is within predefined execution limits.

5. Elevator system (100) according to claim 3 or 4, wherein the basic software module (101-2) is adapted to output a feedback message if an elevator command (103) is not executed.

6. Elevator system (100) according to one of the preceding claims, wherein the replaceable software module (101-1) is adapted to assign a priority level to the elevator command (103).

7. Elevator system (100) according to claim 6, wherein the basic software module (101-2) is adapted to execute the elevator command (103) based on the priority level.

8. Elevator system (100) according to claim 7, wherein elevator system (100) is adapted to re-sequence existing elevator commands (103) within the command queue (105) in dependence on the priority level of a newly inserted elevator command (103).

9. Method for operating an elevator system (100), comprising the steps:

   - generating (S101) an elevator command (103) by a replaceable software module (101-1);
   - putting (S102) the elevator command (103) in a command queue (105);
   - fetching (S103) the elevator command (103) from the command queue (105) by a basic software module (101-2); and
   - controlling (S104) the elevator (107) on the basis of the elevator command (103).

10. Method according to claim 9, wherein the basic software module controls an elevator drive, an elevator brake, a door drive and/or an elevator cabin light.

11. Method according to claim 9 or 10, wherein the basic software module (101-2) tests the elevator command (103) before executing the elevator command (103).

12. Method according to claim 11, wherein the basic software module (101-2) tests whether the elevator command (103) is within predefined execution limits.

13. Method according to one of the claims 9 to 12, wherein the replaceable software module (101-1) assigns a priority level to the elevator command (103).

14. Method according to claim 13, wherein the basic software module (101-2) executes the elevator command (103) based on the priority level.

15. Computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method according to one of the claims 9 to 14.

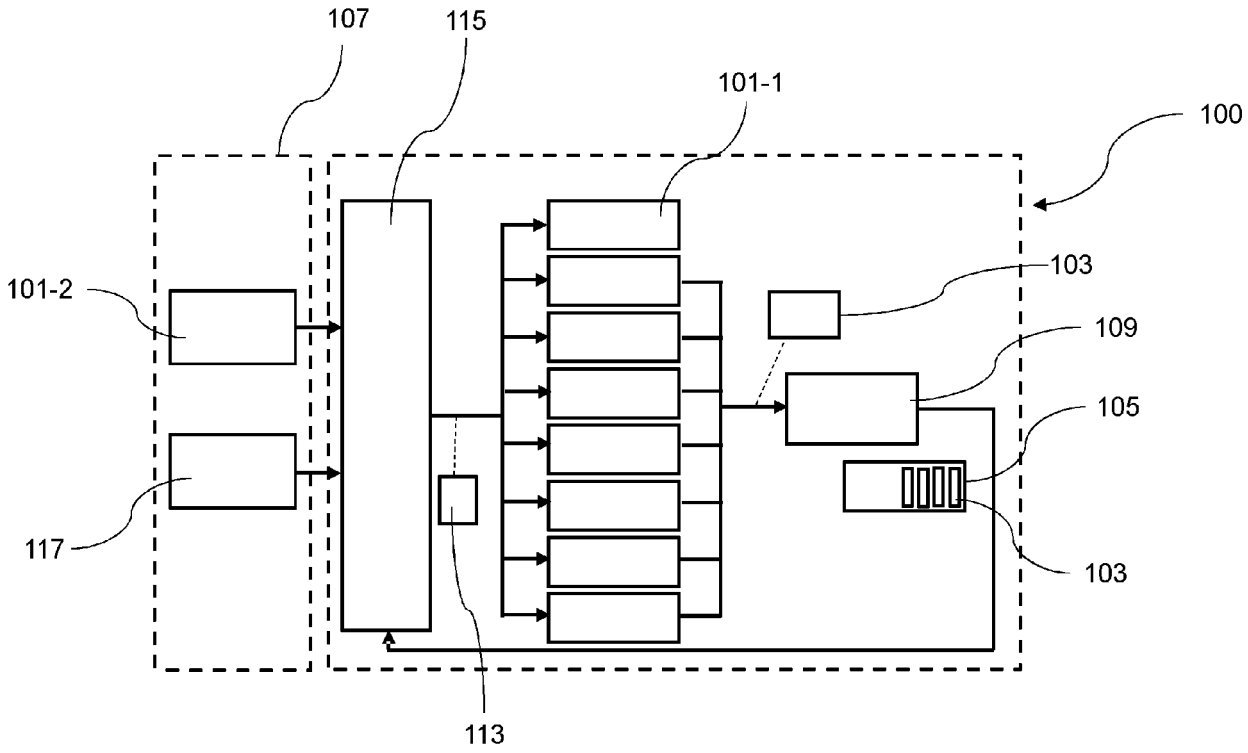

Fig. 1

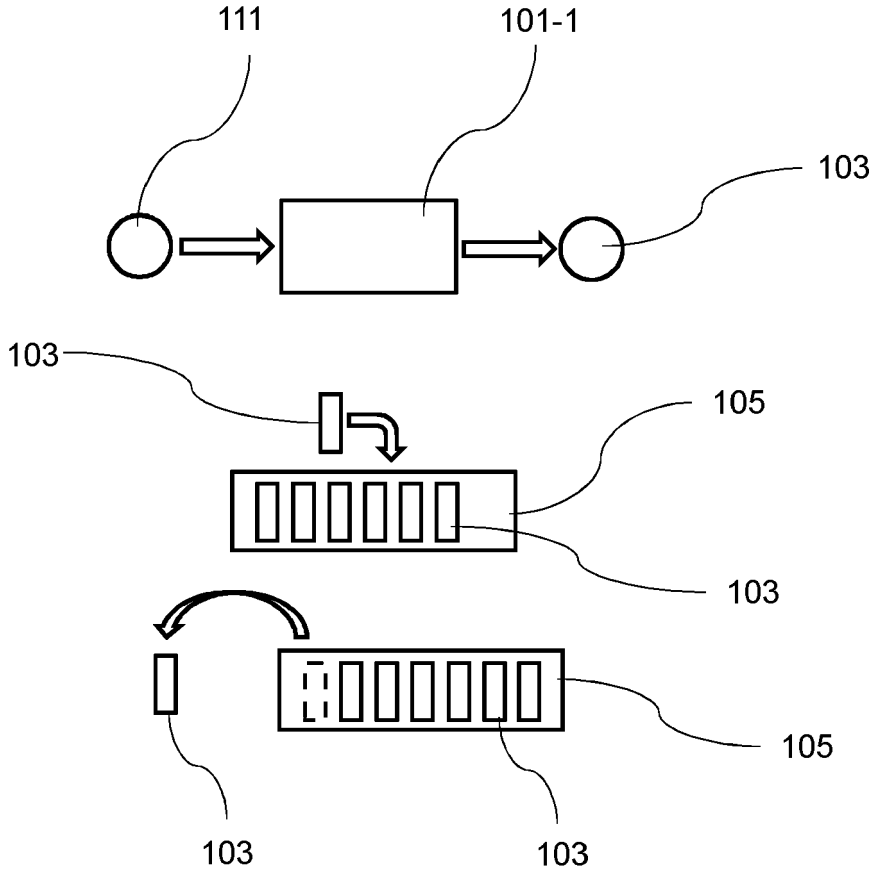

Fig. 2

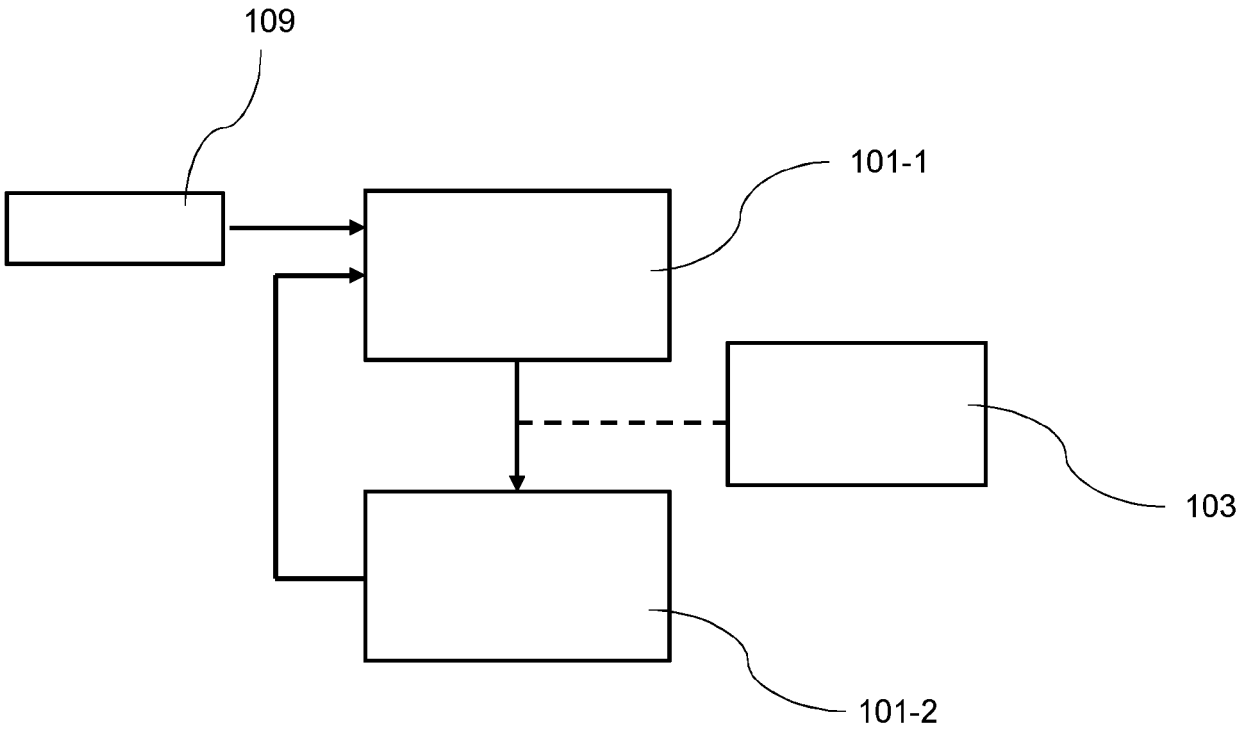

Fig. 3

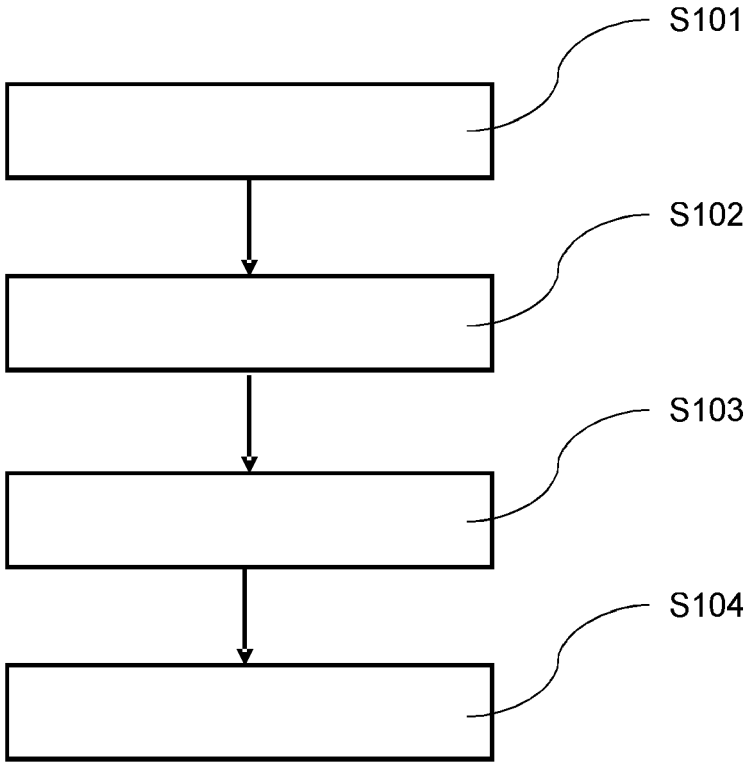

Fig. 4

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 18 19 6874

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 2001/022252 A1 (SCHUSTER KILIAN [CH]) 20 September 2001 (2001-09-20) * paragraphs [0009] - [0011], [0034], [0049]; figure 1 *	1-15	INV. B66B1/34
X	US 2003/034209 A1 (TANG PO-LOUNG [TW] ET AL) 20 February 2003 (2003-02-20) * abstract *	1-3,6-15	
X	CN 107 010 498 A (LI LIANGJIE) 4 August 2017 (2017-08-04) * abstract *	1-4,9-12	
X	JP H11 286379 A (MITSUBISHI ELECTRIC CORP) 19 October 1999 (1999-10-19) * abstract *	1,9	
A	JP 2011 131963 A (MITSUBISHI ELECTRIC CORP) 7 July 2011 (2011-07-07) * abstract *	1-15	
			TECHNICAL FIELDS SEARCHED (IPC)
			B66B

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	25 March 2019	Janssens, Gerd

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 18 19 6874

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

25-03-2019

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2001022252 A1	20-09-2001	AR 035175 A1	05-05-2004
		AT 417803 T	15-01-2009
		AU 778237 B2	25-11-2004
		BR 0101094 A	06-11-2001
		CA 2341145 A1	20-09-2001
		CN 1314301 A	26-09-2001
		EP 1136415 A1	26-09-2001
		ES 2319859 T3	14-05-2009
		HK 1041473 A1	31-07-2009
		JP 2001294371 A	23-10-2001
		NO 20011390 A	21-09-2001
		US 2001022252 A1	20-09-2001
		ZA 200101798 B	11-09-2001
US 2003034209 A1	20-02-2003	NONE	
CN 107010498 A	04-08-2017	NONE	
JP H11286379 A	19-10-1999	NONE	
JP 2011131963 A	07-07-2011	JP 5419675 B2	19-02-2014
		JP 2011131963 A	07-07-2011

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82