



- (51) International Patent Classification:  
G05B 19/042 (2006.01) G06F 8/41 (2018.01)  
G06F 8/76 (2018.01)
- (21) International Application Number:  
PCT/IB2017/057922
- (22) International Filing Date:  
14 December 2017 (14.12.2017)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
201641044555 28 December 2016 (28.12.2016) IN  
201741038622 31 October 2017 (31.10.2017) IN
- (71) Applicant: ABB SCHWEIZ AG [CH/CH]; Brown Boveri  
Strasse 6, 5400 Baden (CH).
- (72) Inventors: JETLEY, Raoul; Brigade Metropolis, Apt  
H-1706 Whitefield Main Road, Bangalore 560048 (IN).  
PHANI, Chitti; Apt. No 010, Krishnageet shelters, 16th  
E Cross, 1st main, Pai layout, Bangalore 560016 (IN).  
NAIR, Sreeja; C 603, SMR Vinay Galaxy, Hoodi, Banga-  
lore 560048 (IN). SUDARSAN, Sithu; Brigade Metropo-  
lis Apt J-0703, Whitefield Main Road, Bangalore 560048  
(IN). V, Govindaraj; B109, Cbr Aakruti, Thigalarapalya,

Mahadevapura Post, Bangalore 560048 (IN). MARALI, Mounesh; 391 10th Main 15th Cross, Narayan Nagar, 1st Block, Kanakapura Road, Bangalore, 560062 (NA).

- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: METHOD AND SYSTEM FOR MIGRATION OF CONTROL LOGIC IN A DISTRIBUTED CONTROL SYSTEM

(57) Abstract: Techniques for providing control blocks to distributed control systems (DCS) of industrial plants are described. The techniques can be used to provide a target control block (104) in a target DCS (102) corresponding to a source control block (106) in a source DCS (108) when DCS of an industrial plant (110) is migrated from the source DCS (108) to the target DCS (102). A source control block (106) is converted into an intermediate representation, which captures the function performed by the source control block (106) and input/output (I/O) parameters of the source control block (106). The intermediate representation of the source control block (106) is compared with an intermediate representation of each of a plurality of candidate control blocks (226, 228, 230, 232) predefined in the target DCS (102). A candidate control block is identified as the target control block (104) based on the comparison.

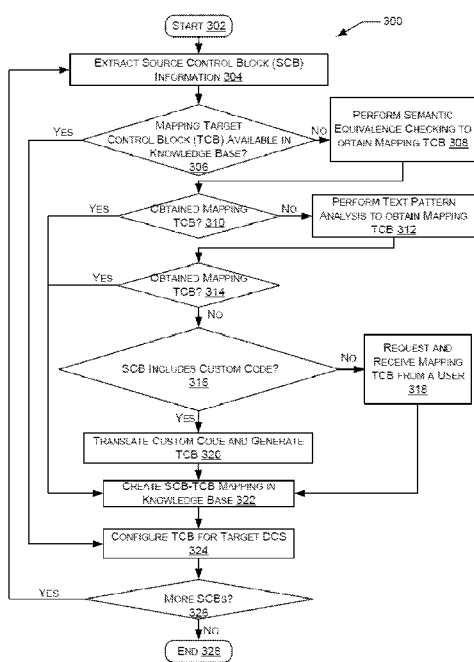


Fig. 3(a)



**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

**Published:**

- *with international search report (Art. 21(3))*

## METHOD AND SYSTEM FOR MIGRATION OF CONTROL LOGIC IN A DISTRIBUTED CONTROL SYSTEM

### TECHNICAL FIELD

[0001] The present subject matter relates, in general, to distributed  
5 control systems (DCS) and, in particular, to method for migrating control  
blocks to DCS.

### BACKGROUND

[0002] A distributed control system (DCS) is a control system for an  
10 industrial plant. The industrial plant may be of a different type of industry,  
such as chemical industry, petrochemical industry, power generation  
industry, and metallurgical industry. The DCS includes a plurality of control  
devices, which are distributed throughout the industrial plant for performing  
various control functions. The control devices include control blocks that  
15 enable the control devices to perform the control functions.

### BRIEF DESCRIPTION OF DRAWINGS

[0003] The features, aspects, and advantages of the present subject  
matter will be better understood with regard to the following description, and  
20 accompanying figures. The use of the same reference number in different  
figures indicates similar or identical features and components.

[0004] Fig. 1 illustrates a system for populating a target distributed  
control system (DCS) with target control blocks corresponding to source  
control blocks in a source DCS, in accordance with an implementation of  
25 the present subject matter.

[0005] Fig. 2 illustrates an environment including the system for  
populating the target DCS with target control blocks corresponding to  
source control blocks in the source DCS, in accordance with an  
implementation of the present subject matter.

[0006] Fig. 2(a) illustrates a control block along with the AST and the CFG associated with the control block in a first DCS, in accordance with an implementation of the present subject matter.

[0007] Fig. 2(b) illustrates a control block along with the AST and the  
5 CFG associated with the control block in a second DCS, in accordance with an implementation of the present subject matter.

[0008] Fig. 3(a) illustrates a flowchart of operations performed by the system for populating the target DCS with target control blocks, in accordance with an implementation of the present subject matter.

10 [0009] Fig. 3(b) illustrates semantic equivalence checking, in accordance with an implementation of the present subject matter.

#### DETAILED DESCRIPTION

[0010] The present subject matter relates to systems and methods for  
15 providing control blocks to distributed control systems (DCS). The systems and methods of the present subject matter can be used for automatically providing control blocks to a target DCS based on control blocks present in a source DCS.

[0011] DCS is a control system used for performing control functions  
20 in an industrial plant. The DCS includes control devices distributed across an industrial plant for performing the control functions. The control functions can include actuation of actuating components, such as a valve or a switch, in the industrial plant. The control devices can be, for example, AC 800M controllers.

25 [0012] A control device performing a control function can include control blocks that enable the control device to perform the control function or to implement the control logic in a control system. A control block is a modular block to be executed by a controller, comprising of one or more function blocks or instructions or logic statements for transforming one or  
30 more inputs into one or more desired outputs. The control blocks can

include, but are not restricted to, multiplexers, timers, counters, and proportional–integral–derivative (PID) loops. A control block may receive a physical input (i.e. I/O signals coming from the field devices such as sensors though I/O modules) or inputs from another control block in the same controller or another controller, and can transform the input into an output, which can be input to another control block (in the same or another controller via a control network) or field devices such as actuators for operating the process equipment via I/O modules.

[0013] Many times, in an industrial plant, the DCS controlling the industrial plant may have to be migrated. In other words, an existing DCS may have to be replaced with a target DCS. Such a replacement may be performed, for example, to upgrade the DCS in the industrial plant to a new version of the DCS. Another scenario when the replacement is performed is when a third party DCS controlling the industrial plant is to be replaced with a proprietary DCS.

[0014] When replacing the existing DCS with the target DCS, it is to be ensured that the control functions performed by the existing DCS are performed by the target DCS. Since the control functions are performed using the control blocks, the target DCS is to have control blocks that enable performing the control functions performed by the existing DCS. Since the existing DCS is different from the target DCS, it may not be possible to use the control blocks in the existing DCS directly in the target DCS. Therefore, the target DCS is to be populated with control blocks corresponding to the control blocks in the existing DCS.

[0015] Conventionally, populating the target DCS with control blocks involves understanding the functions performed by the control blocks in the existing DCS and identifying control blocks in the target DCS that can perform those functions (either independently or together with other control blocks). The understanding of the functions performed by the control blocks has to be performed by subject matter experts (SMEs), i.e., persons with industrial process and control expertise. Further, if a corresponding control

block cannot be identified in the target DCS, the SMEs have to manually generate the corresponding control blocks. Therefore, conventional process of populating the target DCS with control blocks takes a considerable amount of time. Further, the manual involvement makes the populating an error prone process.

[0016] The present subject matter relates to systems and methods for providing control blocks to a target DCS corresponding to control blocks in a source DCS. With the systems and methods of the present subject matter, fast and accurate population of target DCS with control blocks can be achieved.

[0017] In an implementation of the present subject matter, a source control block is converted to an intermediate representation that captures a function performed by the source control block and input/output (I/O) parameters of the source control block. The intermediate representation can be, for example, an abstract syntax tree (AST) or a control flow graph (CFG). The intermediate representation of the source control block is then compared with intermediate representations of a plurality of candidate control blocks. The candidate control blocks are predefined in the target DCS. Based on the comparison, a candidate control block is identified from amongst the plurality of candidate control blocks as the target control block corresponding to the source control block.

[0018] In an implementation of the present subject matter, the identification of the target control block can be performed by comparing source control block data with control block data of the plurality of candidate control blocks. The first source control block data can be extracted from a document related to the first source control block and the control block data of a candidate control block can be extracted from a document related to that candidate control block. Further, when the target control block is not identified for the source control block, the target control block is generated.

[0019] With the systems and methods of the present subject matter, the time consumed in providing control blocks for a target DCS is

considerably reduced. Further, the automatic provision of the target control blocks ensures an error-free process of target control blocks provision.

[0020] The above and other features, aspects, and advantages of the subject matter will be better explained with regard to the following description, appended claims, and accompanying figures.

[0021] Fig. 1 illustrates a system 100 for populating a target distributed control system (DCS) 102 with target control blocks 104 corresponding to source control blocks 106 in a source DCS 108, in accordance with an implementation of the present subject matter. The source DCS 108 may perform the control functions in an industrial plant 110. In some cases, the DCS performing the control functions in the industrial plant 110 may have to be migrated from the source DCS 108 to the target DCS 102. In other words, the existing DCS of the industrial plant 110 is the source DCS 108, and is to be replaced with the target DCS 102. In an example, the source DCS 108 is a legacy or heritage DCS, and the target DCS 102 is an upgraded version of the source DCS 108. The target DCS 102 may be unrelated to the source DCS 108, and may be, for example, proprietary DCS. The industrial plant 110 may belong to any industry sector, such as pharmaceutical, chemical, mineral, oil and gas, and power generation industry.

[0022] As mentioned earlier, the control functions in the industrial plant 110 are performed by the source DCS 108. To enable performance of the control functions, the source DCS 108 includes source control blocks 106. The source control blocks 106 can include, but are not restricted to, multiplexers, timers, counters, and proportional–integral–derivative (PID) loops. An example control function can be closing an inlet valve (not shown in Fig. 1) of an oil tank (not shown in Fig. 1) upon elapse of a predetermined time. For performing the control function, a timer control block can calculate the amount of time for which the inlet valve is kept open, and upon elapse of the predetermined time, can provide input to a second control block, which can then provide a command to close the inlet valve.

[0023] The source control blocks 106 can be distributed among different control devices (not shown in Fig. 1), which, in turn, are distributed throughout the industrial plant 110. Each control device includes a control logic having one or more control blocks. For example, referring back to the inlet valve closing example, the control logic includes the connection of the inlet valve to the timer, connection of the timer to the second control block, and connection of the second control block to the inlet valve. Such a control logic can be downloaded to a source control block which is disposed proximate to the inlet valve for controlling its operations. In another example, a control logic includes a single control block, which alone can enable performance of a control function. In general, a control logic can include one control block, or a plurality of control blocks connected to each other in a specific arrangement for performing one or more control functions.

[0024] As mentioned earlier, the source DCS 108 is to be replaced with the target DCS 102 in the industrial plant 110. Therefore, all the control functions performed by the source DCS 108 are to be performed by the target DCS 102. Since the source DCS 108 performs the control functions with the help of the source control blocks 106, the target DCS 102 is to have target control blocks, which, together, can perform all the control functions in the industrial plant. For this, the system 100 includes a control block mapper engine 120 and a code translation engine 122 that enable population of the target DCS 102 with the target control blocks 104 that correspond to the source control blocks 106. In an embodiment, the system 100 is a migration tool provided on a remote server for migration of the control blocks of the source DCS to the target DCS. In an embodiment, the migration tool is provided as a part of the target DCS for enabling ease of migration. The system 100 may be implemented as any computing system which may be, but is not restricted to, a server, a workstation, etc. The functioning of the system 100 will be explained in detail with reference to the subsequent figures.

[0025] Fig. 2 illustrates an environment 200 including the system 100 for populating the target DCS 102 with target control blocks corresponding to source control blocks in the source DCS 108, in accordance with an implementation of the present subject matter. The system 100 includes the control block mapper engine 120, the code translation engine 122, and processor(s) (not shown in Fig.). The control block mapper engine 120, in turn, includes a semantic equivalence checker engine 202 and a text pattern analyzer engine 204. The system 100 may also include interface(s), memory, other engines, and system data which are not shown in Fig. 2.

10 [0026] The environment 200 further includes an application extractor engine 206, a target code generator engine 208, a knowledge base 210, and a target library 212. In some example implementations, any, some, or all of the application extractor engine 206, the target code generator engine 208, the knowledge base 210, and the target library 212 may be part of the system 100.

[0027] Further, the control block mapper engine 120, the code translation engine 122, the application extractor engine 206, the target code generator engine 208, and the other engines may be implemented in hardware, instructions executed by a processing unit, or by a combination thereof.

[0028] The system data may serve as a repository for storing data that may be fetched, processed, received, or created by the control block mapper engine 120, the code translation engine 122, the application extractor engine 206, the target code generator engine 208, and the other engines or received from connected computing systems and storage devices.

[0029] The knowledge base 210 and the target library 212 may be any database system, database server, or part of a storage area network (SAN).

[0030] In an implementation, some or all of the control block mapper engine 120, the code translation engine 122, the application extractor engine 206, the target code generator engine 208, and the other engines

may communicate with each other through a communication network (not shown in Fig.). The communication network may be a wireless or a wired network, or a combination thereof.

[0031] In operation, the system 100 populates the target DCS 102 with the target control blocks 104 corresponding to the source control blocks 106. The target control blocks 104 includes a target control block 214, target control block 216, and target control block 218. The source control blocks 106 includes a source control block 220, source control block 222, and source control block 224. Although the source DCS 108 and the target DCS 102 are shown to include just three control blocks each, it will be understood that the source DCS 108 and the target DCS may include any number of control blocks depending on the number of control functions to be performed.

[0032] The source control blocks 220, 222, and 224 can be extracted from the source DCS 108 by the application extractor engine 206. In an implementation, the application extractor engine 206 can extract the source control blocks 220, 222, and 224 using one or more application programming interfaces (APIs) exposed by a source DCS platform. In another implementation, the application extractor engine 206 can be specifically tuned to extract the source control blocks 220, 222, and 224 from the source DCS 108.

[0033] In an implementation, the system 100 obtains the target control blocks 214, 216, and 218 from the target library 212. The target library 212 serves as a repository of all control blocks that can be used in the target DCS 102. The control blocks stored in the target library 212 may be referred to as candidate control blocks. For example, the target library 212 includes a plurality of candidate control blocks 226, 228, 230, and 232 that are predefined in the target DCS 102. Although the target library 212 is shown to include just four candidate control blocks, it will be understood that the target library 212 can include any number of candidate control blocks. The

system 100 can obtain a candidate control block from the target library 212 using one or more APIs.

[0034] The knowledge base 210 can be used to store a mapping of a source control block and a corresponding target control block. The knowledge base 210 is empty initially, as mappings of source control blocks and their corresponding target control blocks are not available. However, as the system 100 identifies or generates target control blocks corresponding to source control blocks, the mappings are created and stored in the knowledge base 210. Such a mapping can be utilized at a later stage for quickly identifying a target control block corresponding to a source control block. In order to identify and generate target control blocks corresponding to source control blocks, the system 100 utilizes the control block mapper engine 120 and the code translation engine 122, respectively.

[0035] The control block mapper engine 120 can identify an existing target control block that corresponds to a source control block, such as the source control block 220, source control block 222, or source control block 224. For this, the control block mapper engine 120 utilizes the target library 212, in which a plurality of candidate control blocks 226, 228, 230, and 232 is stored. In other words, the control block mapper engine 120 can determine whether any of the plurality of candidate control blocks 226, 228, 230, and 232 corresponds to the source control block. An example source control block for which the control block mapper engine 120 identifies a corresponding target control block is referred to as a first source control block. The target control block corresponding to the first source control block is referred to as a first target control block. As will be understood, the first target control block is one of the plurality of candidate control blocks 226, 228, 230, and 232. The control block mapper engine 120 can provide the first target control block for populating the target DCS 102.

[0036] If a corresponding target control block is identified for a source control block by the control block mapper engine 120, the code translation engine 122 generates a target control block corresponding to the source

control block. An example source control block for which a corresponding target control block cannot be identified is referred to as a second source control block. The target control block generated for the second source control block is referred to as the second target control block. This aspect is further explained in paragraph 57. The code translation engine 122 can provide the second target control block for populating the target DCS 102.

[0037] In order to identify the first target control block corresponding to the first source control block, the control block mapper engine 120 can compare an intermediate representation of the first source control block with intermediate representation of each of the plurality of candidate control blocks 226, 228, 230, and 232. The comparison of the intermediate representations may be performed by the semantic equivalence checker engine 202, and is explained in more detail with reference to Fig. 3(a).

[0038] In an implementation, to identify the first target control block, the control block mapper engine 122 can compare data related to the first source control block with data related to each of the plurality of control blocks 226, 228, 230, and 232. The data related to a control block is hereinafter referred as control block data. The first source control block data can be extracted from a document related to the first source control block. The document can be, for example, a technical manual of the first source control block. Similarly, candidate control block data can be extracted from a technical manual of the candidate control block. The comparison of the source control block data with the candidate control block data may be performed by the text pattern analyzer engine 204, and is explained in more detail with reference to Fig. 3(a).

[0039] Fig. 3(a) illustrates a flowchart 300 of operations performed by the system 100 for populating the target DCS 102 with target control blocks 104, in accordance with an implementation of the present subject matter.

[0040] The order in which the method 300 is described is not intended to be construed as a limitation, and any number of the described method blocks may be combined in any order to implement the method 300, or an

alternative method. Furthermore, the method 300 may be implemented by processor(s) or computing device(s) through any suitable hardware, non-transitory machine readable instructions, or a combination thereof.

[0041] It may be understood that steps of the method 300 may be performed by programmed computing devices and may be executed based on instructions stored in a non-transitory computer readable medium. Although the method 300 may be implemented in a variety of systems; the method 300 is described in relation to the aforementioned system 100, for ease of explanation.

10 [0042] The operations start at step 302, and at step 304, a source control block, such as the source control block 220, is extracted from the source DCS 108. As mentioned earlier, the extraction of the source control block can be performed by the application extractor engine 206. The extracted source control block is hereinafter referred to as a source control  
15 block.

[0043] At step 306, it is checked whether a target control block corresponding to the source control block is present in the knowledge base 210. The corresponding target control block may also be referred to as a mapping target control block. The knowledge base 210 may include a  
20 mapping target control block if a target control block corresponding to the source control block was stored in the knowledge base 210 by the control block mapper engine 120 or the code translation engine 122.

[0044] If a mapping target control block is not found in the knowledge base 210, at step 308, the control block mapper engine 120 compares  
25 intermediate representation of the extracted source control block with intermediate representations of each of the plurality of candidate control blocks 226, 228, 230, and 232. The comparison of the intermediate representations is also referred to as semantic equivalence checking. The semantic equivalence checking can be performed by the semantic  
30 equivalence checker engine 202.

[0045] In order to perform the semantic equivalence checking, first, the semantic equivalence checker engine 202 converts the source control block into an intermediate representation. The intermediate representation captures the function performed by the source control block and its input/output (I/O) parameters. The function performed by the extracted source control block and its I/O parameters are explained with reference to a control block for a PID loop.

[0046] The function performed by the control block for the PID loop can be represented as below:

$$10 \quad \text{EQUA (Proportional, Integral, Derivative, CV(s))} = \\ K * [(((1 + T_1 * s / T_1 * s) + T_2 * s) * (PVP(s) - SPP(s)))]$$

wherein,

CV refers to a controlled variable,

T<sub>1</sub> refers to Integral time constant in minutes,

15 T<sub>2</sub> refers to derivative time constant in minutes,

K refers to the proportional gain,

PVP refers to process variable in percentage, and

SPP refers to set point in percentage.

[0047] Further, if the control block for the PID loop receives a tag name, SP, PV, PV source, Mode (manual or automatic), Eq. type, gain, integral time, derivative time, and control action as inputs to produce its output, the I/O representation of the control block for the PID loop, capturing the I/O parameters, can be provided as below:

25 PID = f (Tag name, SP, PV, PV Source, Mode, Eq. Type, Gain, Integral time, Derivative time, Control Action)

[0048] Therefore, if the semantic equivalence checker engine 202 converts the control block for the PID loop into an intermediate representation, the intermediate representation captures the above mentioned function and I/O parameters of the control block. In an implementation, in addition to the inputs and outputs of a control block, the I/O parameters captured by the intermediate representation also includes numerical ranges of the I/O parameters. In addition to the function performed and the I/O parameters, the intermediate representation of the source control block preserves semantics and syntax of the source control block. The intermediate representation of the source control block can be, for example, an abstract syntax tree (AST) or a control flow graph (CFG). In addition to the source control block, the semantic equivalence checker engine 202 converts each of the plurality of candidate control blocks 226, 228, 230, and 232 into their respective intermediate representations.

[0049] Upon conversion of the extracted source control block and the plurality of candidate control blocks 226, 228, 230, and 232 into their representations, the semantic equivalence checker engine 202 can compare the intermediate representation of the source control block with the intermediate representation of each of the plurality of candidate control blocks 226, 228, 230, and 232 in order to identify target control block corresponding to the extracted source control block. The comparison of the intermediate representations can also be referred to as checking for a mapping target control block.

[0050] The rationale behind the comparison of intermediate representations is that if two control blocks perform the same function and have same inputs and outputs, the two control blocks can be considered to be semantically equivalent. Therefore, upon comparison, if the intermediate representation of a candidate control block is found to be matching with the intermediate representation of the extracted source control block, the candidate control block can be identified as the target control block corresponding to the extracted source control block.

[0051] In an example, the comparison involves comparison of the AST of the extracted source control block and AST of each of the plurality of candidate control blocks 226, 228, 230, and 232. In another example, the comparison involves comparison of the CFG of the extracted source control block and CFG of each of the plurality of candidate control blocks 226, 228, 230, and 232. In an implementation, in order to perform the comparison of the extracted source control block with a candidate control block, first, the ASTs of the extracted source control block and the candidate control block are compared. If the ASTs match, the candidate control block is identified as the target control block corresponding to the source control block. If the ASTs do not match, then, the CFGs of the extracted source control block and the candidate control block are compared. If the CFGs match, the candidate control block is identified as the target control block corresponding to the source control block. If the CFGs also do not match, the semantic equivalence checker engine 202 can perform a similar comparison for the next candidate control block in the plurality of candidate control blocks 226, 228, 230, and 232. Likewise, the comparison can be performed for each of the plurality of candidate control blocks 226, 228, 230, and 232. Although comparison of the intermediate representations has been explained with respect to both ASTs and CFGs, it will be understood that, in an implementation, any one comparison alone can be performed. Further, the intermediate representations are not restricted to AST and CFG alone, and any intermediate representation that captures functions and I/O parameters of a control block and preserves the semantics and syntax of the control block can be utilized. This is further explained using in the description of figures 2(a) and 2(b).

[0052] Figures 2(a) and 2(b) illustrate two control blocks namely control block 250 and control block 270. The two control blocks 250 and 270 defined in two different control systems using two different notations (control block 250 using functional block diagrams and control block 270 using textual programming. Both control blocks 250 and 270 implement the same

functionality i.e. of comparing the result of an AND operation against a baseline. If the computed value is greater than the baseline, the output is true, otherwise, the output is assigned the value false.

[0053] As explained previously, in order to determine if the control block 250 maps to control block 270, the knowledge base is checked for a direct mapping between the two blocks. In case there is no existing mapping for the source control block in the knowledge base, the system will try to establish a mapping using the intermediate representation (AST and CFG) of the control blocks.

10 [0054] The AST for the control block 250 is shown in figure 2(a) as AST 260 and the AST for the control block 270 is shown in figure 2(b) as AST 280. The ASTs 260 and 280 for the two control blocks 250 and 270 are compared to detect structural (or syntactic) similarity. The ASTs are generated by parsing the two control blocks (based on the semantics of the languages). In this case, there is no structural similarity between the two control blocks (at best, there is a partial similarity). Hence, no mapping can be established based on the ASTs. Therefore, a semantic equivalence check is needed, based on the CFGs generated for the two blocks.

[0055] The CFG for the control block 250 is shown in figure 2(a) as CFG 265 and the CFG for the control block 270 is shown in figure 2(b) as CFG 285. In this example, both CFGs reduce to the expressions

$$\begin{aligned} \text{out1} &= \text{in1 AND in2 AND in3} \\ \text{out2} &= (\text{bool\_to\_real}(\text{out1}) \geq \text{in4})? \text{true} : \text{false} \end{aligned}$$

25 Since the two CFGs are evaluated to be equivalent, a mapping between the two code blocks 260 and 280 is determined.

[0056] At step 310, it is determined if a matching target control block is obtained from the semantic equivalence checking performed at block 308. If intermediate representation of none of the plurality of candidate control blocks 226, 228, 230, and 232 matches with that of the extracted source control block, at step 312, the text pattern analyzer engine 204 performs

text pattern analysis to obtain a mapping target control block. In other words, the text pattern analysis is performed, at step 312, if it is determined at step 310 that a mapping target control block was not obtained at step 308.

[0057] For performing the text pattern analysis, the text pattern analyzer engine 204 extracts information related to the source control block from a document related to the extracted source control block. The information related to the extracted source control block may be referred to as source control block data. Further, the document related to the source control block can be, for example, technical manual having technical information about the source control block. The extraction of the source control block data from the document can be performed using text mining. The source control block data may include information related to inputs, outputs, and internal working of the source control block.

[0058] The text pattern analysis will be explained with the help of an example. Consider that the source control block is a control block titled "Analog Input", which converts an analog signal to a process variable, and takes tag name, low limit, and high limit as inputs. Consider also that the above information has been extracted from a technical manual having information about the "Analog Input" control block. Further, consider that there exists a candidate control block "AnalogConv", and text extracted from a technical manual having information about "AnalogConv" recites that "*The Control block AnalogConv converts inputs from analog devices into a process variable. The tag name, low limit, and high limit are used in this process.*". The text pattern analyzer engine 204, which compares the text extracted from the control block "Analog Input" with the text related to the control block "AnalogConv", can determine that the control block "AnalogConv" corresponds to the control block "Analog Input" in the source DCS 108.

[0059] Similar to extracting the source control block data, the text pattern analyzer engine 204 can extract control block data of each of the plurality of candidate control blocks 226, 228, 230, and 232. The extraction

of the control block data for the plurality of candidate control blocks 226, 228, 230, and 232 may be performed prior to extraction of source control block data. This enables reducing time involved in comparing the source control block data with the candidate control block data. Thereafter, the text pattern analyzer engine 204 can compare the source control block data with the control block data of each candidate control block using matching techniques, such as pattern matching and proximity analysis. Based on the comparison, if control block data of a candidate control block is found to be matching with the source control block data, the text pattern analyzer engine 204 can identify that candidate control block as the target control block.

[0060] In an implementation, based on the comparison of the source control block data with the candidate control block data, the text pattern analyzer engine 204 generates a relevance ranking for one or more candidate control blocks indicating the relevance of control block data of the one or more candidate control blocks with the source control block data. The relevance ranking may be provided to a user, such as a subject matter expert (SME), who, based on the relevance rankings, can determine whether a corresponding target control block has been found. Otherwise, or additionally, the text pattern analyzer engine 204 may, by itself, determine if a corresponding target control block is present among the plurality of candidate control blocks 226, 228, 230, and 232. Based on the determination, the text pattern analyzer engine 204 may provide a confidence value for each candidate control block determined to be a potential target control block, and may identify a candidate control block as the target control block if it has a confidence value greater than a threshold.

[0061] If, at step 314, it is determined that control block data of none of the plurality of candidate control blocks 226, 228, 230, and 232 matches with the source control block data, it may be determined that a target control block corresponding to a source control block does not exist in the target library 212. The non-existence of the corresponding target control block could be because of two reasons. One is that the source control block was

created using custom code, because of which a corresponding predefined candidate control block does not exist in the target library 212. The source control block may have been created using custom code if the source DCS 108 does not include a standard control block that performs an intended  
5 functionality. Another reason for non-existence of the corresponding target control block is that there may not be a precise one-to-one mapping between the source control block and a candidate control block. For example, the functionality of the source control block may be performed by a combination of two candidate control blocks in the target library 212.

10 [0062] Therefore, at step 316 it is determined whether the source control block includes custom code. If the source control block is determined to not include any custom code, as explained earlier, it may be because of the absence of a one-to-one mapping between the source control block and any of the plurality of candidate control blocks 226, 228, 230, and 232. In such  
15 a case, at step 318, a user, such as an SME, is requested to provide a corresponding target control block. The user can provide the corresponding target control block, for example, by understanding the function performed by the source control block, determining a combination of candidate control blocks that can perform the function, and provisioning the combination of  
20 candidate control blocks as corresponding to the single source control block. If there is no combination of candidate control blocks that can perform the function, the user can define a new control block altogether.

[0063] If, at step 316, it is determined that the source control block includes custom code, at step 320, the code translation engine 122 can  
25 generate a new target control block. For this, first, the code translation engine 122 parses the logic of the custom code and tokenizes it. Based on the tokenized data, the code translation engine 122 generates an AST for the custom code. The AST is then mapped on to a new code construct created using the notation system in the target DCS, which conforms to a  
30 standard, such as International Electrotechnical Commission (IEC) 61131 standard. Thereafter, a standard code is generated from the new code

construct. The intermediate code is then optimized to generate code for the target control block. The target control block is then generated using the generated code.

[0064] In an implementation, upon identification of a target control block corresponding to the source control block, a mapping of the source control block and the target control block is created in the knowledge base 210. For instance, at steps 310 and 314, if it is determined that the corresponding target control block is identified, at step 322, a mapping of the source control block and the target control block is created in the knowledge base 210. Similarly, if, at step 318, if the user provides the corresponding target control block, or, at step 320, if the code translation engine 122 generates the corresponding target control block, the mapping is created in the knowledge base 210. The mapping identified by the control block mapper engine 120 is created in the knowledge base 210 by the control block mapper engine 120 and the mapping generated by the code translation engine 122 can be created in the knowledge base 210 by the code translation engine 122.

[0065] In an implementation, before creating the mapping of the identified or generated target control block in the knowledge base 210, a feedback may be received from the SME regarding the correctness of the mapping. If the mapping is incorrect, the SME may provide the corresponding target control block.

[0066] Since the mapping is created in the knowledge base 210, if at a later stage, a target control block corresponding to the source control block is requested for, a determination can be made that the corresponding mapping exists in the knowledge base 210 at step 306. Therefore, the corresponding target control block can be provided readily from the knowledge base 210 instead of identifying or generating the corresponding target control block. The provision of the corresponding target control block from the knowledge base 210 can be performed by the control block mapper engine 120.

[0067] Since the knowledge base 210 is updated for each identified or generated mapping, over a period of time, the knowledge base 210 becomes updated with mapping control blocks for all source control blocks 220, 222, and 224. Therefore, it can be understood that the time taken to provide a corresponding target control block for a source control block considerably decreases in the long run.

[0068] Upon identification or generation of a target control block, the target control block is configured for use in the target DCS 102. The configuration of the target control block can be performed by the target code generator engine 208. The configuration of the target control block can include, but is not restricted to, arrangement of the target control block in a control logic. The configured target control block can be provided to the target DCS 102. In other words, the target code generator engine 208 populates the target DCS 102 with the configured target control block.

[0069] In addition to configuring the target control blocks, the target code generator engine 208 can also configure the control application used in the target DCS 102. The configuration of the control application includes, but is not restricted to, adding target control blocks, defining control modes of control blocks, and adding I/O modules. For configuring the control application, the target code generator engine 208 utilizes system configuration parameters from the source DCS 108 in order to automate and speed up the configuration process.

[0070] Upon configuration of the target control block, at step 326, it is checked whether the source DCS 108 includes any more source control blocks for which corresponding target control blocks are not found. If there are any such source control blocks, the process repeats. Otherwise, at step 328, the process ends.

[0071] Fig. 3(b) illustrates the semantic equivalence checking performed at step 308, in accordance with an implementation of the present subject matter.

[0072] The order in which the method 308 is described is not intended to be construed as a limitation, and any number of the described method steps may be combined in any order to implement the method 308, or an alternative method. Furthermore, the method 308 may be implemented by  
5 processor(s) or computing device(s) through any suitable hardware, non-transitory machine readable instructions, or a combination thereof.

[0073] At step 352, the source control block is converted into an intermediate representation, such as AST or CFG, which captures the function performed by the source control block and the I/O parameters of  
10 the source control block.

[0074] At step 354, the intermediate representation of the source control block is compared with the intermediate representation of each of a plurality of candidate control blocks. For example, the AST of the source control block may be compared with the AST of the candidate control blocks  
15 and the CFG of the source control block may be compared with the CFG of the candidate control blocks.

[0075] In an implementation, based on the comparison at step 354, a candidate control block is identified from amongst the plurality of candidate control blocks, as explained with reference to steps 310 and 312.

20 [0076] In an embodiment, the above mentioned method and system can also be used for migration of HMI from the source DCS to the target DCS. This is further explained below.

[0077] In the embodiment, the target library 212 includes candidate code blocks for HMI of the target DCS. Additionally, the knowledge base  
25 200 includes a set of production rules associated with one or more source languages in which the HMI of the source DCS is coded. The production rules govern the syntax of each statement that is written using the one or more source language. The production rules are defined in Backus-Naur form, and specify the grammar of the source language(s). Additionally, the  
30 knowledge base 200 contains a repository of symbols typically used in the source languages in which the HMI of the source DCS is coded. The

symbols include both terminals and non-terminals used in the language. For each symbol, the repository contains the type of the symbol, along with the associated fields and methods, if needed. The repository will be accessed in case the Type of a symbol is not known while tokenizing the input source.

5 [0078] For example by just parsing the discrete VB code snippet given below, the type of the symbol conndb is not known. But the associated methods ConnectionString and Open can help infer that the symbols is an object of the Type adodb.Connection.

```
with conndb
10     .ConnectionString = ""
     .CursorLocation = adUseClient
     .Open
end with
```

[0079] During migration, the source project and the code files associated with the HMI of the source DCS is provided to the system 100. Then, the control block mapper engine 120 takes the source project and code files associated with the HMI and identifies individual symbols based of the repository of the symbols and the set of production rules in the knowledge base 200, to aid in parsing. The control block mapper engine 20 120 also removes auto-generated code, comments and other coding constructs that do not need to be translated. The source project and code files are parsed based on the production and grammar rules. The suitable production rules are picked to parse the statements in the code. Then semantic analysis of the parsed source project of the HMI is performed. The 25 next step is to understand the semantic meaning of the symbols. This is required because a suitable target construct can only be identified based on the meaning of the source construct. There are several ways that the meaning of the symbol can be obtained: for literals and constants, the corresponding meaning can be obtained from the language definition itself, 30 for non-literals, the control block mapper engine 120 analyzer looks up the symbol in the Repository of symbols as described above. If a unique match

is found, the associated Type is assigned to the symbol. In case no unique match is found in the repository of symbols, control block mapper engine 120 finds the relevant type information for the symbol from online documentation, coding forums, programing blogs, etc., using textual analysis as mentioned above. Then the source project of the HMI is converted into an intermediate representation (for example abstract syntax tree).

[0080] Based on the AST of the source project of the HMI of the source DCS, the HMI of the target DCS is generated. The control block mapper engine 120 uses the AST of the source project to identify appropriate code blocks in the target library 212, as explained above for control blocks.

[0081] The methods and systems of the present subject matter automate the process of providing control blocks to a target DCS based on control blocks in a source DCS. This considerably decreases the amount of time taken in migrating from a source DCS to a target DCS. Further, the dependence on the SMEs for the migration is minimized. This also makes the migration an error-free process.

[0082] Although the present subject matter has been described with reference to specific embodiments, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as alternate embodiments of the subject matter, will become apparent to persons skilled in the art upon reference to the description of the subject matter.

I/We claim:

1. A method for providing a target control block (104) in a target distributed control system (DCS) (102) corresponding to a source control block (106) in a source DCS (108), wherein one or more control blocks of the DCS of an industrial plant (110) is to be migrated from the source DCS (108) to the target DCS (102), the method comprising:

5  
10  
15  
20  
25

converting the source control block (106) into an intermediate representation capturing a function performed by the source control block (106) and input/output (I/O) parameters of the source control block (106), wherein the intermediate representation comprises at least one of an abstract syntax tree (AST) and a control flow graph (CFG);

comparing the intermediate representation of the source control block (106) with an intermediate representation of each of a plurality of candidate control blocks (226, 228, 230, 232) predefined in the target DCS (102); and

identifying a candidate control block from amongst the plurality of candidate control blocks (226, 228, 230, 232) as the target control block (104) based on the comparison.

20

2. The method as claimed in claim 1, wherein identifying a candidate control block from amongst the plurality of candidate control blocks (226, 228, 230, 232) comprises:

25

determining whether an intermediate representation of any of the plurality of candidate control blocks (226, 228, 230, 232) matches with the intermediate representation of the source control block (106);

when the intermediate representation of a candidate control block from amongst the plurality of candidate control blocks (226,

228, 230, 232) matches with the intermediate representation of the source control block (106),

identifying the candidate control block as the target control block (104); and

5 when the intermediate representation of none of the plurality of candidate control blocks (226, 228, 230, 232) matches with the intermediate representation of the source control block (106),

extracting source control block (106) data from a document related to the source control block (106);

10 comparing the source control block (106) data with control block data of each of the plurality of candidate control blocks (226, 228, 230, 232); and

15 identifying a candidate control block from amongst the plurality of candidate control blocks (226, 228, 230, 232) when the source control block (106) data matches with control block data of the candidate control block.

3. The method as claimed in claim 2, wherein when the source control block (106) data matches with control block data of none of the plurality of candidate control blocks (226, 228, 230, 232), the method comprises:

20 translating code of the source control block (106) into an abstract syntax tree (AST);

converting the AST to a standard code; and

25 generating the target control block (104) from the standard code.

4. The method as claimed in any of one of claims 1-3, comprising:

creating a mapping of the source control block (106) and the target control block (104) in a knowledge base (210); and

30 providing the target control block (104) from the knowledge base (210) based on the mapping upon a subsequent request to

provide the target control block (104) corresponding to the source control block (106).

5. The method as claimed in claim 1, wherein comparing the intermediate representation of the source control block (106) with the intermediate representation of a candidate control block comprises:

comparing AST of the source control block (106) with AST of the candidate control block; and

10 comparing CFG of the source control block (106) with CFG of the candidate control block when the AST of the candidate control block does not match with the AST of the source control block (106).

6. A system (100) for populating a target distributed control system (DCS) with target control blocks (104) that correspond to source control blocks (106) in a source DCS (108), the source DCS (108) comprising a plurality of source control blocks (106), the system (100) comprising:

a control block mapper engine (120) to:

20 identify a first target control block corresponding to a first source control block of the plurality of source control blocks by at least one of:

25 comparing an intermediate representation of the first source control block with intermediate representation of each of a plurality of candidate control blocks (226, 228, 230, 232) that are predefined in the target DCS (102); and

30 comparing first source control block data with control block data of each of the plurality of candidate control blocks (226, 228, 230, 232), wherein the first source control block data is extractable from a document related to the first source control block and the control block data of a candidate control block of

the plurality of candidate control blocks (226, 228, 230, 232) is extractable from a document related to the candidate control block,

wherein the first target control block is one of the plurality of candidate control blocks (226, 228, 230, 232); and

provide the first target control block for populating the target DCS (102); and

a code translation engine (122) to:

generate a second target control block corresponding to a second source control block of the plurality of source control blocks (106), wherein a target control block corresponding to the second source control block is not identified by the control block mapper engine (120); and

provide the second target control block for populating the target DCS (102).

7. The system (100) as claimed in claim 6, comprising a knowledge base (210) to store mappings of source control blocks (106) and their corresponding target control blocks (104), wherein the control block mapper engine (120) is to:

create a mapping of the first source control block and the first target control block in the knowledge base (210),

wherein the code translation engine (122) is to:

create a mapping of the second source control block and the second target control block in the knowledge base (210), and

wherein the control block mapper engine (120) is to:

provide the first target control block for populating the target DCS (102) based on the mapping of the first source control block and the first target control block in the knowledge base (210) upon a subsequent request to populate the DCS with a target control block corresponding to the first source control block; and

5 provide the second target control block for populating the target DCS (102) based on the mapping of the second source control block and the second target control block in the knowledge base (210) upon a subsequent request to populate the DCS with a target control block corresponding to the second source control block.

8. The system (100) as claimed in claim 6, wherein the intermediate representation of the first source control block captures a function performed by the first source control block and input/output (I/O) parameters  
10 of the first source control block, and wherein the control block mapper engine (120) is to compare the first source control block data with the control block data of each of the plurality of candidate control blocks (226, 228, 230, 232) when the intermediate representation of none of the plurality of candidate control blocks (226, 228, 230, 232) matches with the intermediate  
15 representation of the first source control block.

9. The system (100) as claimed in claim 6, wherein, to generate the second target control block, the code translation engine is (122) to:  
20 translate code of the second source control block into an abstract syntax tree (AST);  
convert the AST to a standard code; and  
generate the second target control block from the standard code.

25 10. The system (100) as claimed in claim 6, comprising a target code generator (208) to:  
configure the first target control block and the second target control block; and  
30 populate the target DCS (102) with the first target control block and the second target control block.

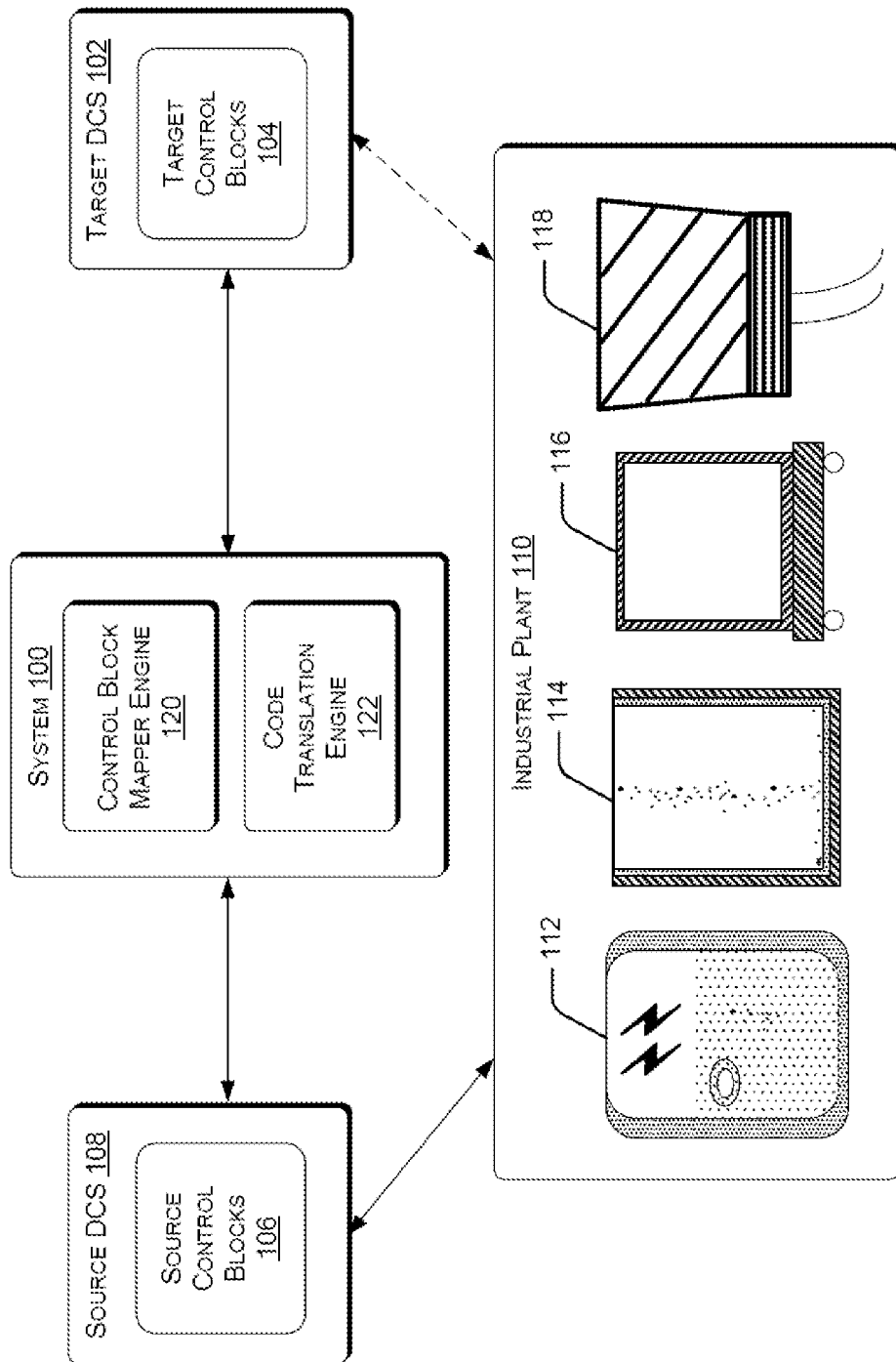


Fig. 1

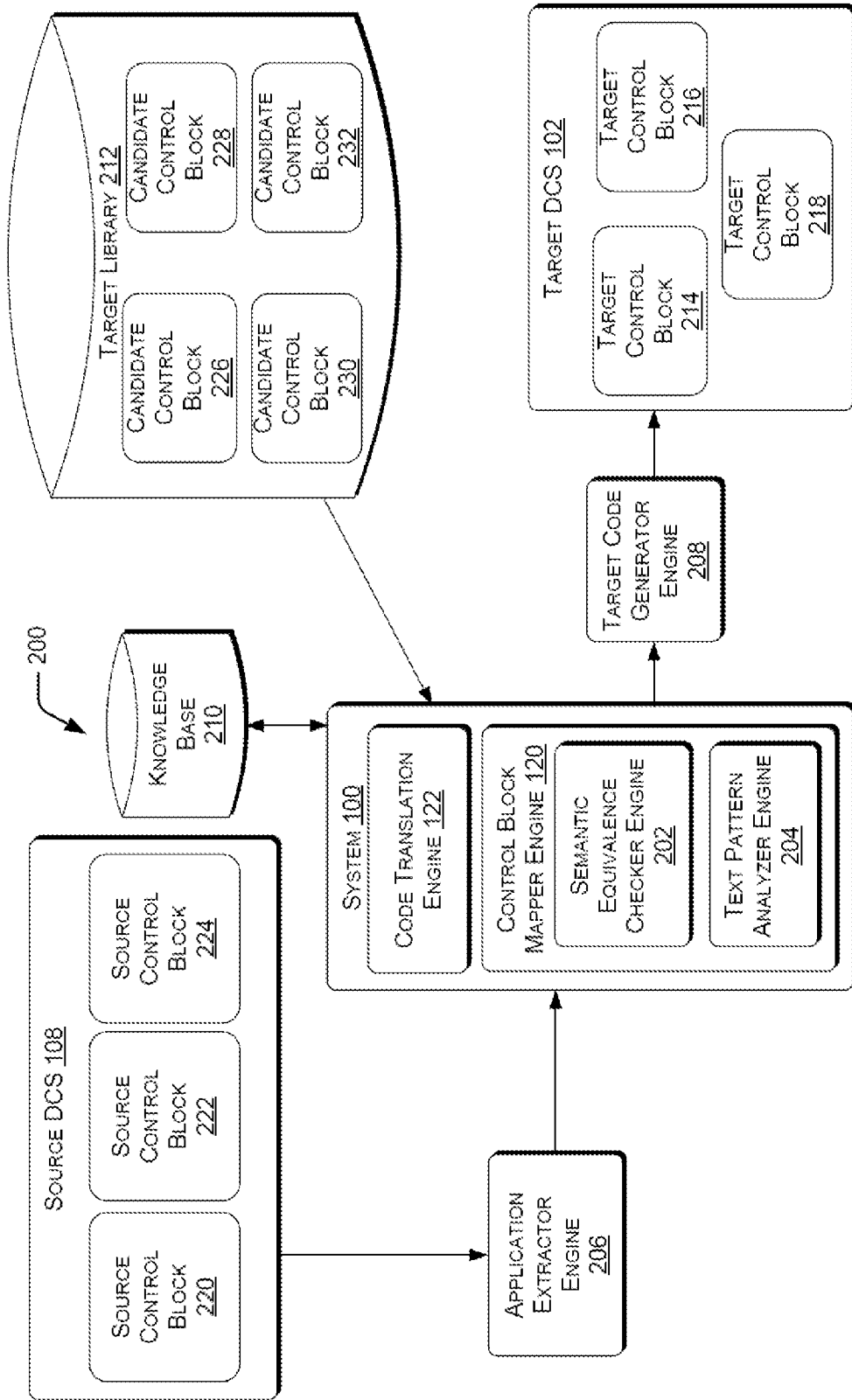
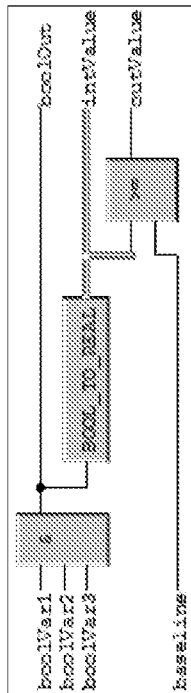
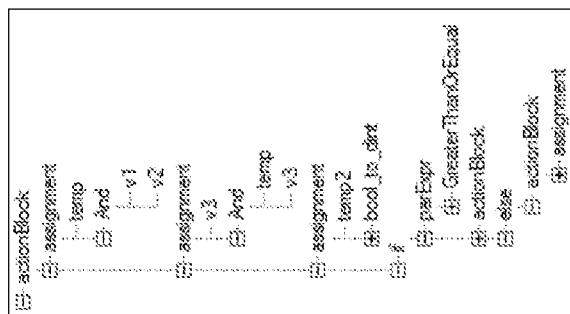


Fig. 2

250



260



265

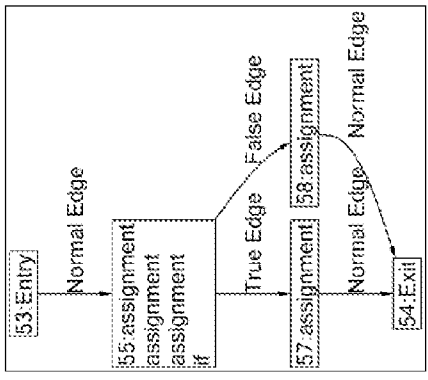


Fig. 2(b)

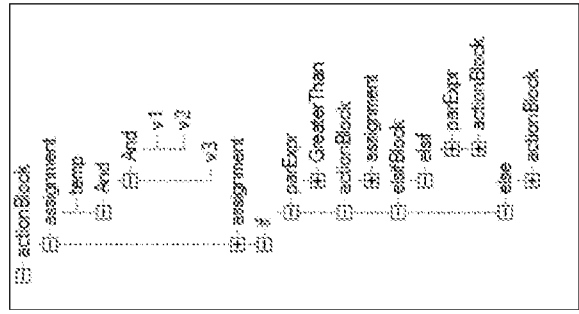
270

```

boolean v1:=false, v2:=false, v3:=false;
boolean temp;
real temp2, v4;

temp := v1 AND v2 AND v3;
temp2 := bool_to_real(v3);
if(temp2 > v4)
  vOut := true;
elseif (temp2 < v4)
  vOut := false;
else
  vOut := true;
end-if;
    
```

280



285

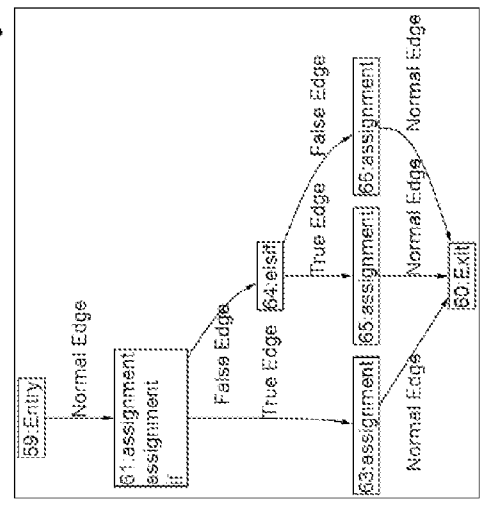


Fig. 2(c)

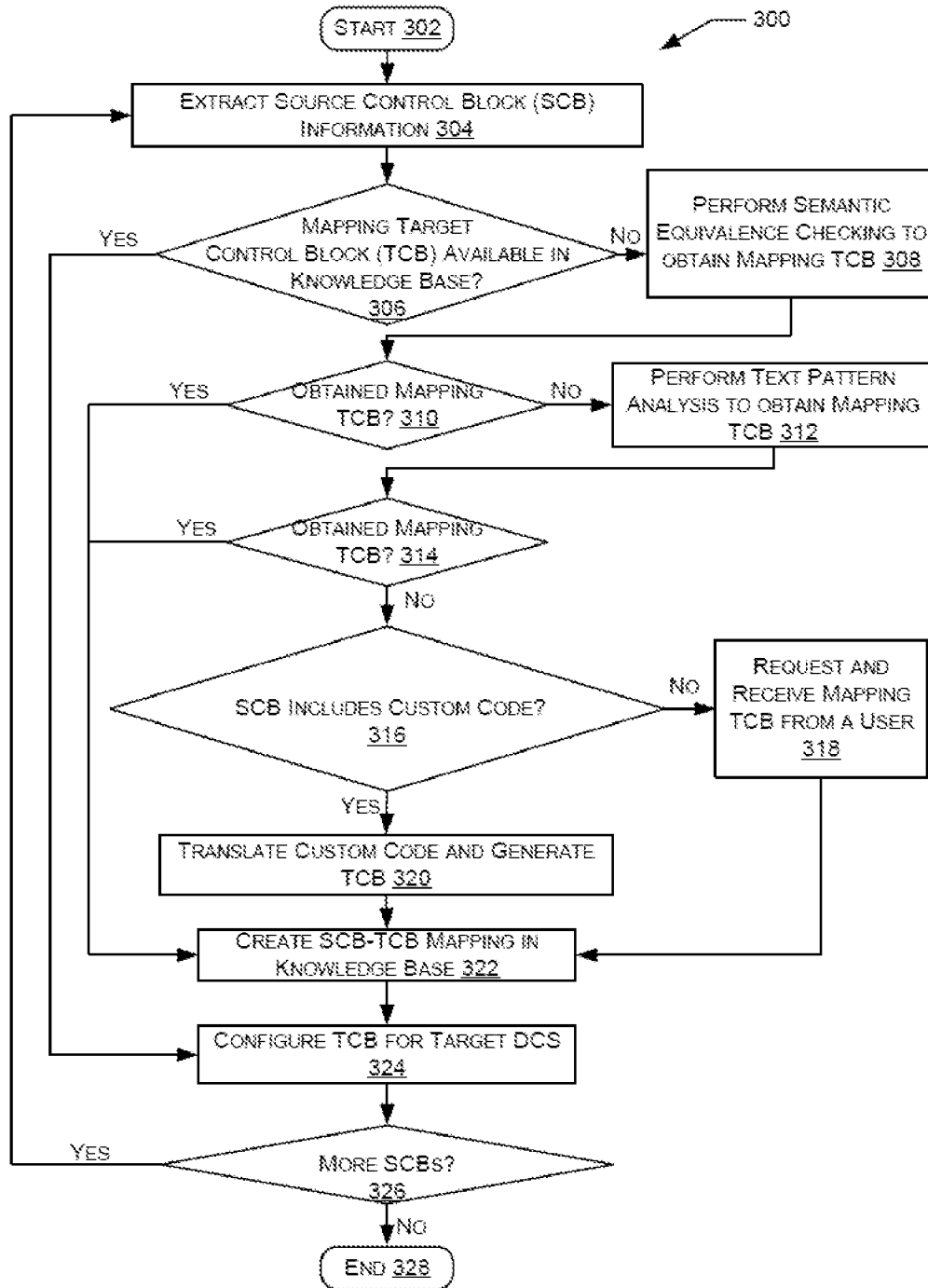


Fig. 3(a)

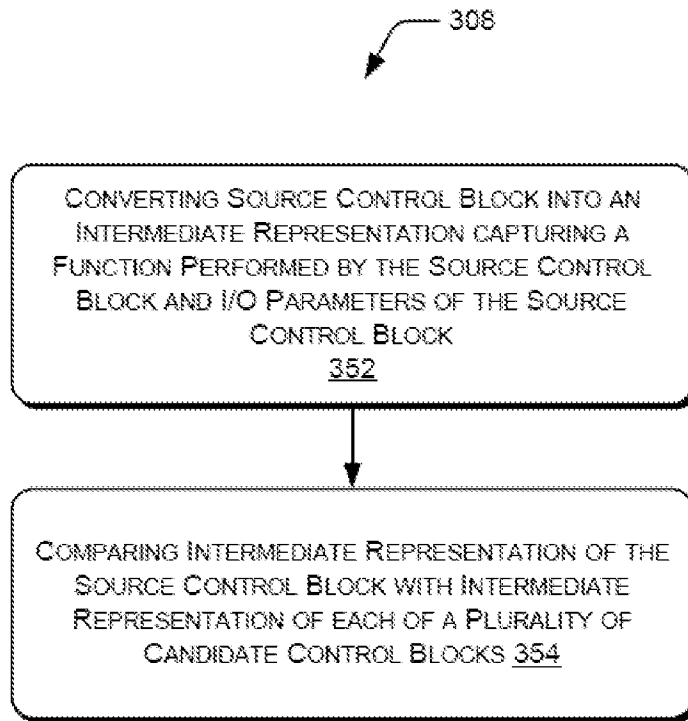


Fig. 3(b)

**INTERNATIONAL SEARCH REPORT**

International application No  
PCT/IB2017/057922

A. CLASSIFICATION OF SUBJECT MATTER  
INV. G05B19/042 G06F8/76 G06F8/41  
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
G05B G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages  | Relevant to claim No. |
|-----------|---|-----------------------|
| X         | US 9 411 581 B2 (GIZMOX TRANSPOSITION LTD [IL]) 9 August 2016 (2016-08-09)<br>the whole document  | 1-10                  |
| X         | WO 2008/013520 A1 (HONEYWELL INT INC [US]; KISHORE RAVINDRA [IN]; MAHESH JOSHI K [IN]) 31 January 2008 (2008-01-31)<br>paragraph [0013] - paragraph [0032];<br>claims 1,2; figures 1,2A | 1-10                  |
| X         | US 2015/205603 A1 (ARAYA CARLOS [CR] ET AL) 23 July 2015 (2015-07-23)<br>paragraph [0031] - paragraph [0035];<br>figures 1-3  | 1-10                  |
|           | -----<br>-/--   |                       |

Further documents are listed in the continuation of Box C.

See patent family annex.

\* Special categories of cited documents :

|   |  |
|---|--|
| "A" document defining the general state of the art which is not considered to be of particular relevance  | "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention  |
| "E" earlier application or patent but published on or after the international filing date   | "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone   |
| "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| "O" document referring to an oral disclosure, use, exhibition or other means  | "&" document member of the same patent family  |
| "P" document published prior to the international filing date but later than the priority date claimed  |  |

|  |  |
|--|--|
| Date of the actual completion of the international search<br><br>9 February 2018 | Date of mailing of the international search report<br><br>20/02/2018 |
|--|--|

|  |   |
|--|---|
| Name and mailing address of the ISA/<br>European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Authorized officer<br><br>Rakoczy, Tobias |
|--|---|

## INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2017/057922

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |   |                       |
|--|---|-----------------------|
| Category*  | Citation of document, with indication, where appropriate, of the relevant passages  | Relevant to claim No. |
| A  | <p>FAY A: "METHODEN ZUR UNTERSTUETZUNG DER MIGRATION VON PROZESSLEITSYSTEM-SOFTWARE//METHODS TO SUPPORT THE MIGRATION OF PROCESS CONTROL SYSTEM SOFTWARE",<br/>AUTOMATISIERUNGSTECHNISCHE PRAXIS - ATP, OLDENBOURG INDUSTRIEVERLAG, MUNCHEN, DE, vol. 44, no. 6,<br/>1 January 2002 (2002-01-01), pages 39-44, XP001536507,<br/>ISSN: 0178-2320<br/>the whole document</p> <p style="text-align: center;">-----</p> | 1-10                  |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2017/057922

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date            |
|--|------------------|-------------------------|-----------------------------|
| US 9411581                             | B2               | 09-08-2016              | EP 2839371 A2 25-02-2015    |
|  |                  |                         | US 2015082290 A1 19-03-2015 |
|  |                  |                         | WO 2013157008 A2 24-10-2013 |
| -----                                  |                  |                         |                             |
| WO 2008013520                          | A1               | 31-01-2008              | NONE                        |
| -----                                  |                  |                         |                             |
| US 2015205603                          | A1               | 23-07-2015              | NONE                        |
| -----                                  |                  |                         |                             |