(12) UK Patent Application (19) GB (11) 2502181 (13) A

(43) Date of A Publication 20.11.2013

(21) Application No: 1303691.8

(22) Date of Filing: 01.03.2013

(30) Priority Data:
(31) 2012050634 (32) 07.03.2012 (33) JP

(51) INT CL:
B61L 19/00 (2006.01) *B61L 27/00* (2006.01)

(56) Documents Cited:
WO 2013/014989 A1 **US 20100063657 A1**
US 20090014927 A1 **US 20080154452 A1**

(58) Field of Search:
Other: **Online: WPI, EPODOC, TXTE**

(71) Applicant(s):
Hitachi, Ltd.
(Incorporated in Japan)
6-6 Marunouchi 1-chome, Chiyoda-ku,
Tokyo 100-8280, Japan

(72) Inventor(s):
Tomoyuki Mochizuki
Akitoshi Shimura
Katsumi Kawai
Hisanori Teshima
Noboru Sato

(74) Agent and/or Address for Service:
Mewburn Ellis LLP
33 Gutter Lane, LONDON, EC2V 8AS, United Kingdom

(54) Title of the Invention: **Generation method for route control program and computer system**
Abstract Title: **Route control program with assignment of line information to developer on basis of skill level.**

(57) A method of generating a route control program amongst a plurality of developers utilizes a using a track database stored on a computer. The method divides the line information stored in the database and assigns it to developers according to a disclosure level and skill of the developer (117). In a second step control logic stored in the database corresponding to a block and line geometry for the divided line is assigned (119). In a third step, information for coupling the assigned control logics to each other. The method may calculate a complexity (115) of the divided line section on the basis of the block information and geometry of the line extracted from the database, the complexity being used to assign the line to a developer. A confidentiality priority (114) or man-hour reduction (113) policy may also be used to divide the line.

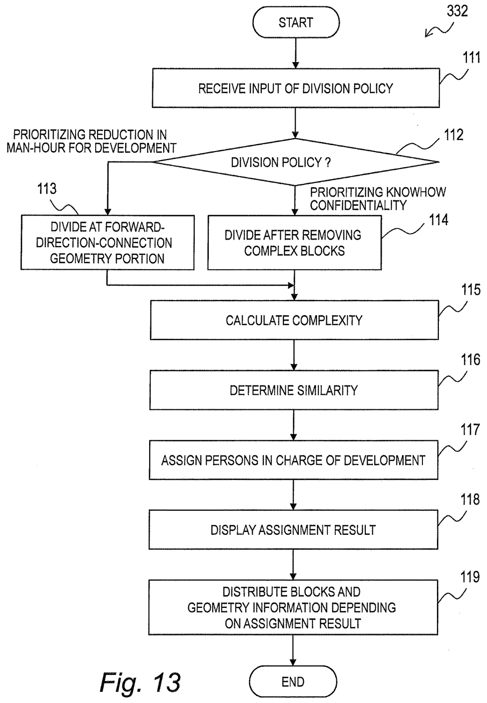

START
332
111
RECEIVE INPUT OF DIVISION POLICY
PRIORITIZING REDUCTION IN MAN-HOUR FOR DEVELOPMENT
112
DIVISION POLICY ?
113
PRIORITIZING KNOWHOW CONFIDENTIALITY
DIVIDE AT FORWARD-DIRECTION-CONNECTION GEOMETRY PORTION
DIVIDE AFTER REMOVING COMPLEX BLOCKS
114
115
CALCULATE COMPLEXITY
116
DETERMINE SIMILARITY
117
ASSIGN PERSONS IN CHARGE OF DEVELOPMENT
118
DISPLAY ASSIGNMENT RESULT
119
DISTRIBUTE BLOCKS AND GEOMETRY INFORMATION DEPENDING ON ASSIGNMENT RESULT
END

Fig. 13

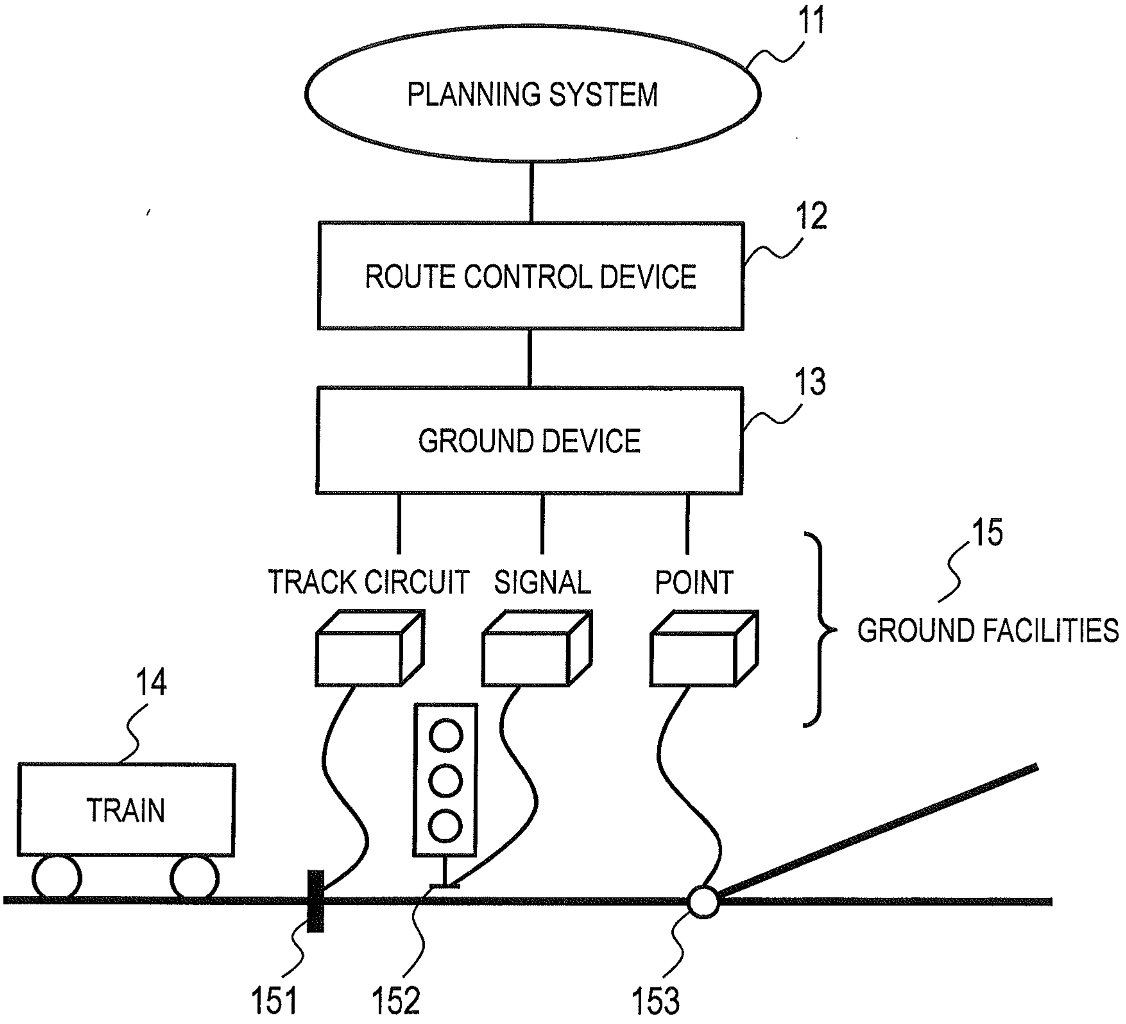

11
PLANNING SYSTEM
12
ROUTE CONTROL DEVICE
13
GROUND DEVICE
15
TRACK CIRCUIT
SIGNAL
POINT
GROUND FACILITIES
14
TRAIN
151
152
153

Fig. 1

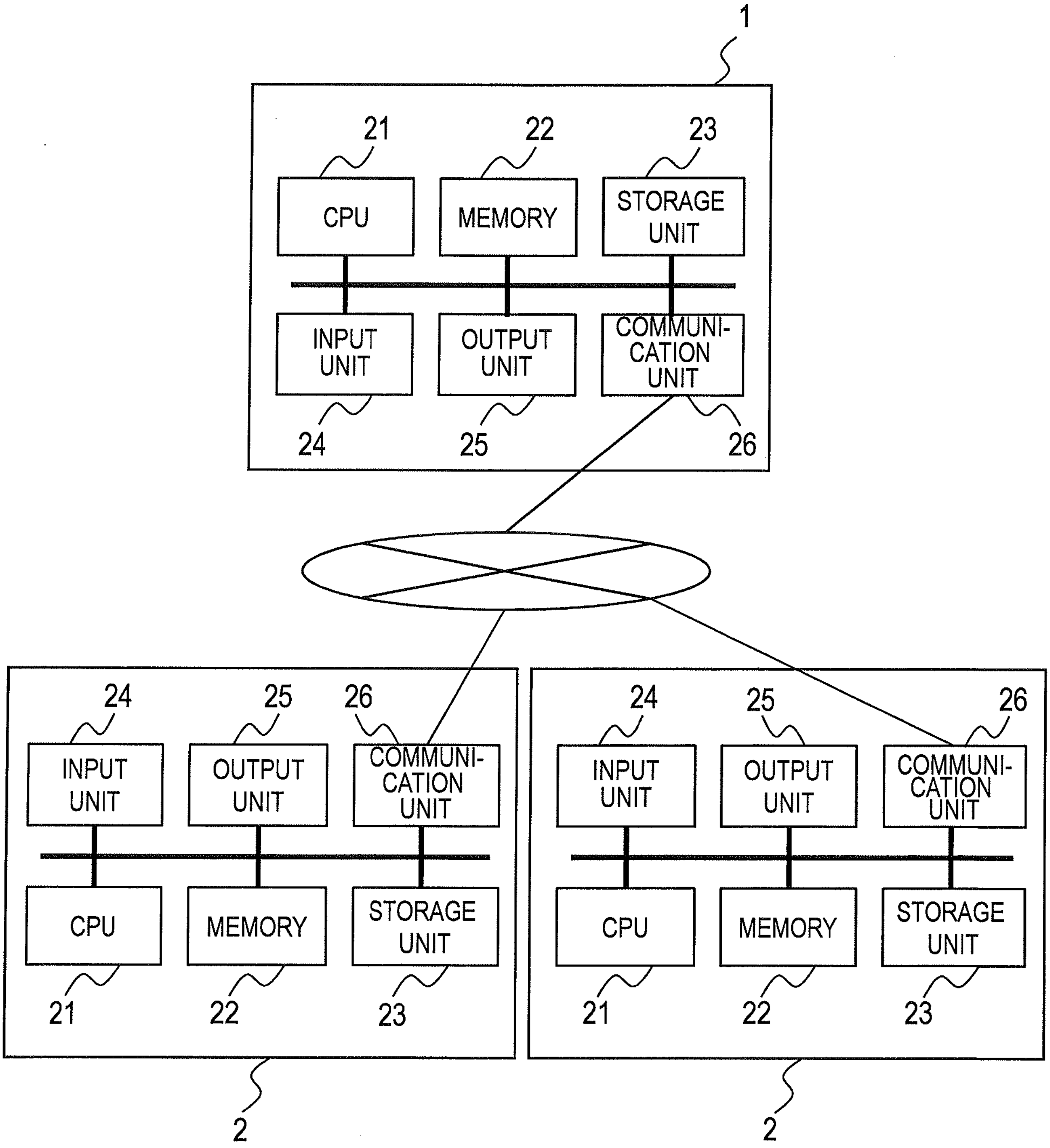

1
21
22
23
CPU
MEMORY
STORAGE UNIT
INPUT UNIT
OUTPUT UNIT
COMMUNI-CATION UNIT
24
25
26
24
25
26
INPUT UNIT
OUTPUT UNIT
COMMUNI-CATION UNIT
CPU
MEMORY
STORAGE UNIT
21
22
23
24
25
26
INPUT UNIT
OUTPUT UNIT
COMMUNI-CATION UNIT
CPU
MEMORY
STORAGE UNIT
21
22
23
2
2

Fig. 2

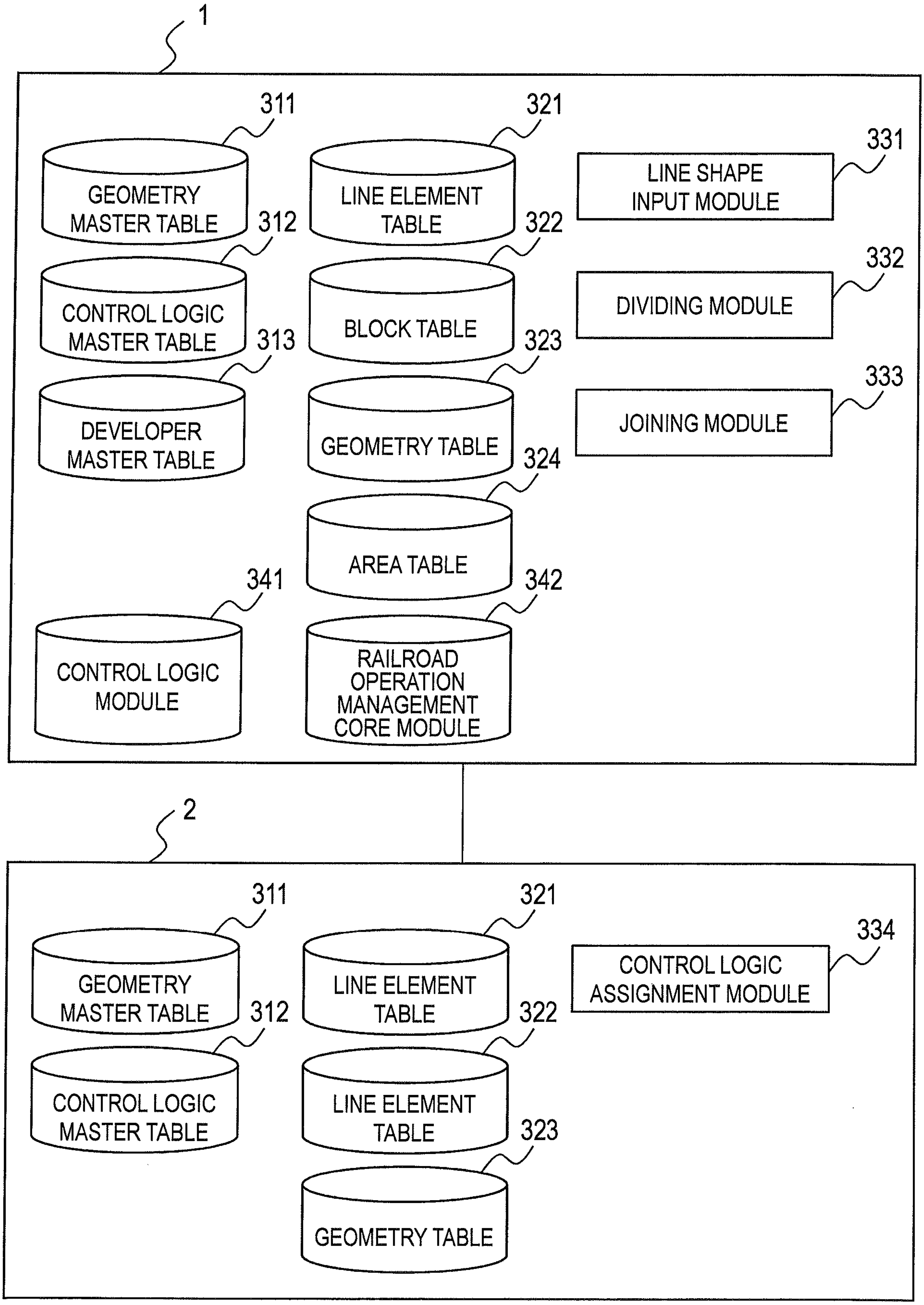

1
311
GEOMETRY MASTER TABLE
312
CONTROL LOGIC MASTER TABLE
313
DEVELOPER MASTER TABLE
321
LINE ELEMENT TABLE
322
BLOCK TABLE
323
GEOMETRY TABLE
324
AREA TABLE
331
LINE SHAPE INPUT MODULE
332
DIVIDING MODULE
333
JOINING MODULE
341
CONTROL LOGIC MODULE
342
RAILROAD OPERATION MANAGEMENT CORE MODULE
2
311
GEOMETRY MASTER TABLE
312
CONTROL LOGIC MASTER TABLE
321
LINE ELEMENT TABLE
322
LINE ELEMENT TABLE
323
GEOMETRY TABLE
334
CONTROL LOGIC ASSIGNMENT MODULE

Fig. 3

Fig. 4A

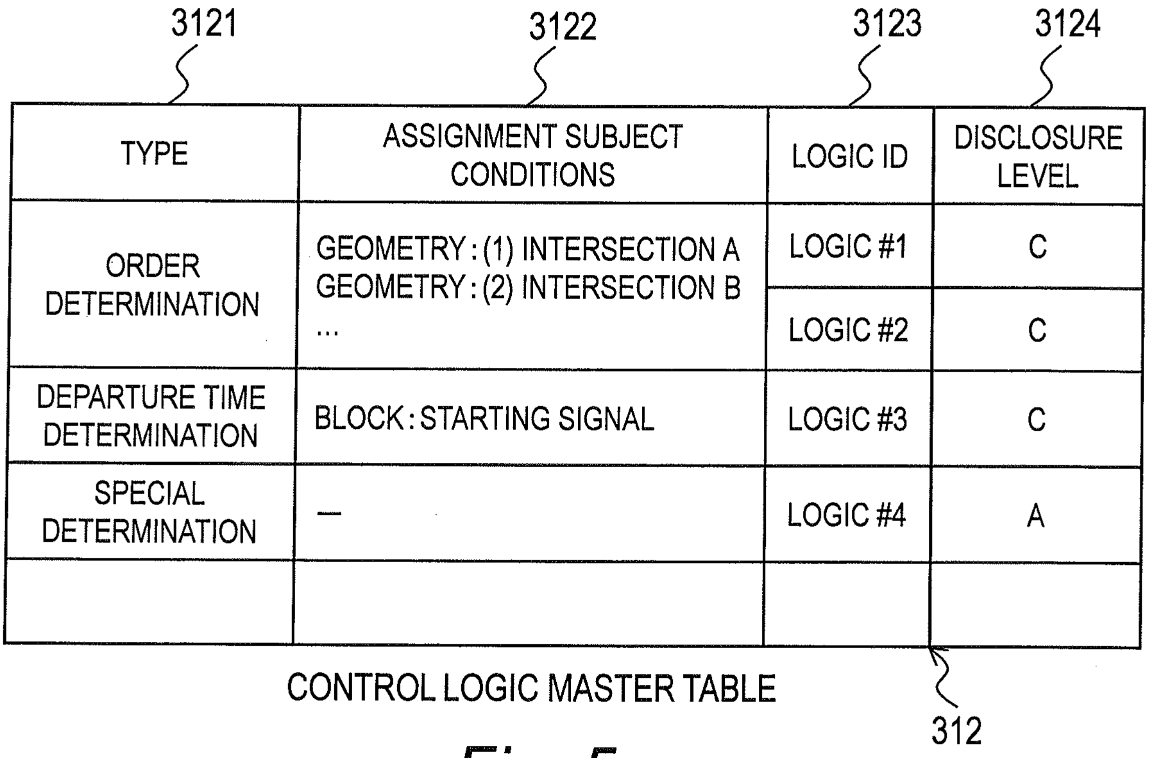
GEOMETRY MASTER TABLE

311

GEOMETRY PATTERN CLASSIFICATION CRITERIA							
NUMBER OF OVERLAPPING TERMINAL POINTS	TYPE OF OVERLAPPING TERMINAL POINTS	NUMBER OF INCLUDED TERMINAL POINTS 1 -> 2	TYPE OF INCLUDED TERMINAL POINTS	NUMBER OF INCLUDED TERMINAL POINTS 2 -> 1	TYPE OF INCLUDED TERMINAL POINTS	DIRECTIONAL RELATIONSHIP	GEOMETRY TYPE
0 OVER-LAPPING TERMINAL POINTS	NONE	0	NONE	0	NONE	FORWARD	(1) INTERSECTION A
						BACKWARD	(2) INTERSECTION B
		1	DEPARTURE POINT	0	NONE	FORWARD	(3) START POINT SINGLE CLOSURE A
						BACKWARD	(4) START POINT SINGLE CLOSURE B
			ARRIVAL POINT		NONE	FORWARD	(5) END POINT SINGLE CLOSURE A
						BACKWARD	(6) END POINT SINGLE CLOSURE B
		1	DEPARTURE POINT	1	ARRIVAL POINT	FORWARD	(7) START/END POINTS ALTERNATE SINGLE CLOSURE
						BACKWARD	NA
			DEPARTURE POINT		DEPARTURE POINT	FORWARD	NA
						BACKWARD	(8) DEADLOCK B
			ARRIVAL POINT		ARRIVAL POINT	FORWARD	NA
						BACKWARD	(9) END POINT BOTH CLOSURE C
		2	BOTH	0	NONE	FORWARD	(10) START/END POINTS SINGLE CLOSURE A
						BACKWARD	(11) START/END POINTS SINGLE CLOSURE B

Fig. 4B

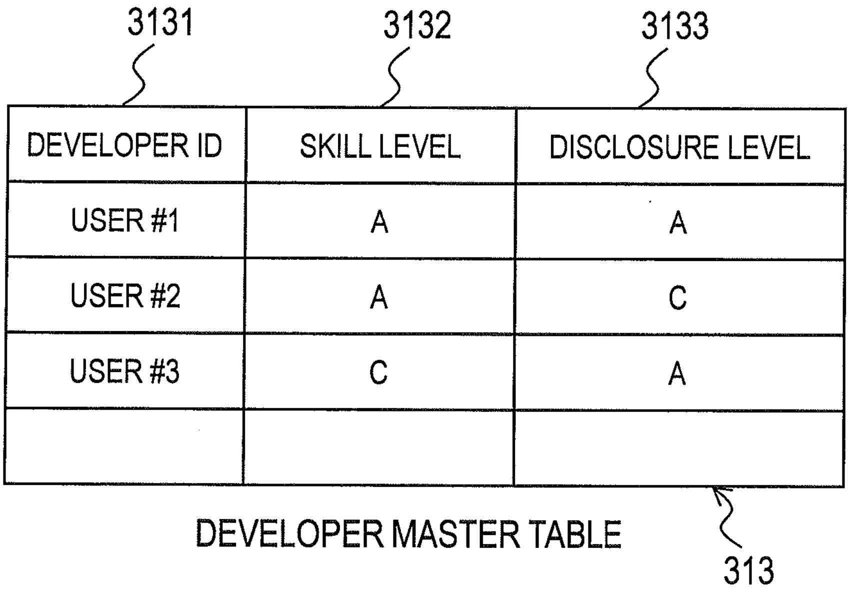
1 OVER-LAPPING TERMINAL POINT	BLOCK 1 ARRIVAL POINT AND BLOCK 2 DEPARTURE POINT	0	NONE	0	NONE	FORWARD	(12) FORWARD CONNECTION
						BACKWARD	(13) BACKWARD CONNECTION A
		1	DEPARTURE POINT	0	NONE	FORWARD	NA
						BACKWARD	(14) BACKWARD CONNECTION B
			ARRIVAL POINT		NONE	FORWARD	NA
						BACKWARD	(15) BACKWARD CONNECTION C
	BLOCK 1 DEPARTURE POINT AND BLOCK 2 DEPARTURE POINT	0	NONE	0	NONE	FORWARD	(16) ROUTE SELECTION A
						BACKWARD	(17) ROUTE SELECTION B
		1	ARRIVAL POINT	0	NONE	FORWARD	(18) ROUTE SELECTION C
						BACKWARD	NA
	BLOCK 1 ARRIVAL POINT AND BLOCK 2 ARRIVAL POINT	0	NONE	0	NONE	FORWARD	(19) END POINT BOTH CLOSURE A
						BACKWARD	(20) END POINT BOTH CLOSURE B
		1	DEPARTURE POINT	0	NONE	FORWARD	(21) START/END POINTS SINGLE CLOSURE C
						BACKWARD	NA
2 OVER-LAPPING TERMINAL POINTS	BLOCK 1 DEPARTURE POINT AND BLOCK 2 ARRIVAL POINT + BLOCK 2 DEPARTURE POINT AND BLOCK 1 ARRIVAL POINT	0	NONE	0	NONE	FORWARD	NA
						BACKWARD	(22) DEADLOCK A
	BLOCK 1 DEPARTURE POINT AND BLOCK 2 DEPARTURE POINT + BLOCK 2 ARRIVAL POINT AND BLOCK 1 DEPARTURE POINT	0	NONE	0	NONE	FORWARD	BLOCK 1 AND BLOCK 2 ARE SAME
						BACKWARD	NA


3121
3122
3123
3124
312

TYPE	ASSIGNMENT SUBJECT CONDITIONS	LOGIC ID	DISCLOSURE LEVEL
ORDER DETERMINATION	GEOMETRY : (1) INTERSECTION A GEOMETRY : (2) INTERSECTION B ...	LOGIC #1	C
		LOGIC #2	C
DEPARTURE TIME DETERMINATION	BLOCK : STARTING SIGNAL	LOGIC #3	C
SPECIAL DETERMINATION	—	LOGIC #4	A

CONTROL LOGIC MASTER TABLE

Fig. 5


3131
3132
3133
313

DEVELOPER ID	SKILL LEVEL	DISCLOSURE LEVEL
USER #1	A	A
USER #2	A	C
USER #3	C	A

DEVELOPER MASTER TABLE

Fig. 6

3211 3212

LINE ELEMENT ID	COORDINATES
Line #1	(0, 0) - (3, 0)
Line #2	(3, 0) - (5, 0)
Line #3	(3, 0) - (5, 5)

LINE ELEMENT TABLE

321

Fig. 7

3221 3222 3223 3224 3225 3226 3227

BLOCK ID	TYPE	DEPARTURE POINT	ARRIVAL POINT	DIRECTION	LINE ELEMENT IDs	LOGIC ID
BLOCK #1	HOME	(0, 0)	(5, 5)	UP	Line#1, Line#2, Line#3	LOGIC #11
BLOCK #2	HOME	(0, 0)	(5, 0)	UP	Line#1, Line#4	LOGIC #11
BLOCK #3	STARTING	(5, 0)	(10, 0)	UP	Line#5, Line#6	LOGIC #12

BLOCK TABLE

322

Fig. 8

3231	3232	3233	3234	3235
GEOMETRY ID	GEOMETRY TYPE	BLOCK 1 ID	BLOCK 2 ID	LOGIC ID
GEOMETRY #1	(16) ROUTE SELECTION A	BLOCK #1	BLOCK #2	LOGIC #21

GEOMETRY TABLE

323

Fig. 9

3241	3242	3243	3244
AREA ID	BLOCK ID	COMPLEXITY	DEVELOPER ID
AREA #1	BLOCK #1, BLOCK #2, BLOCK #3, BLOCK #4	C	DEVELOPER #1

AREA TABLE

324

Fig. 10

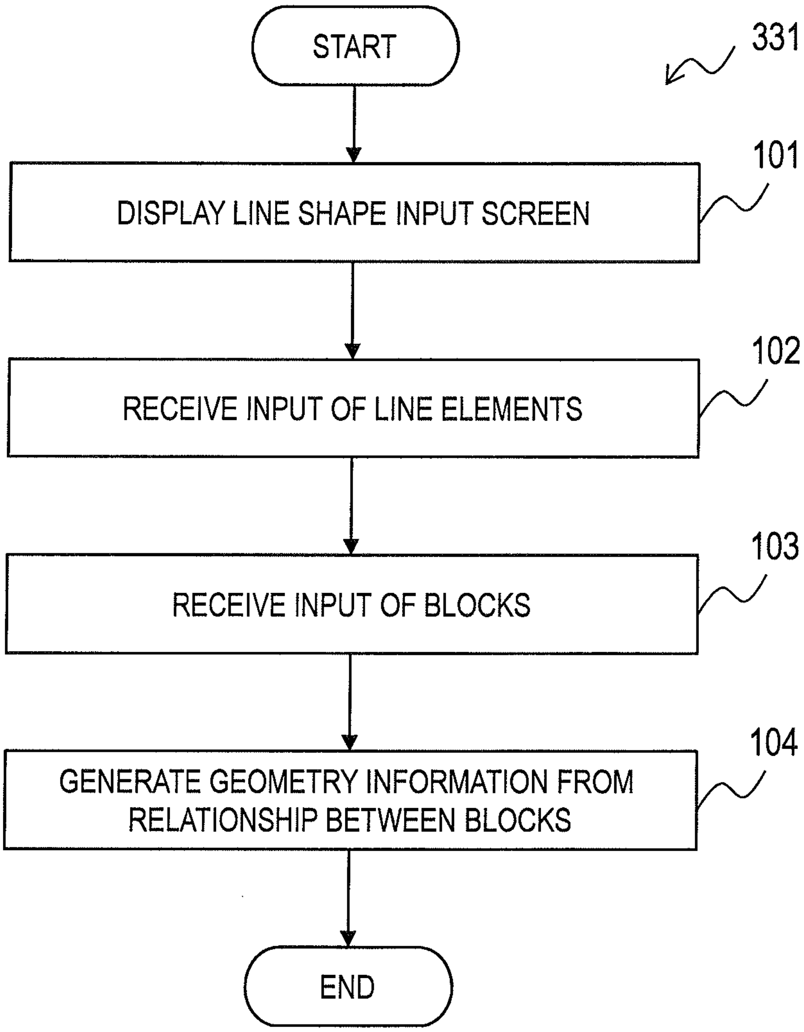
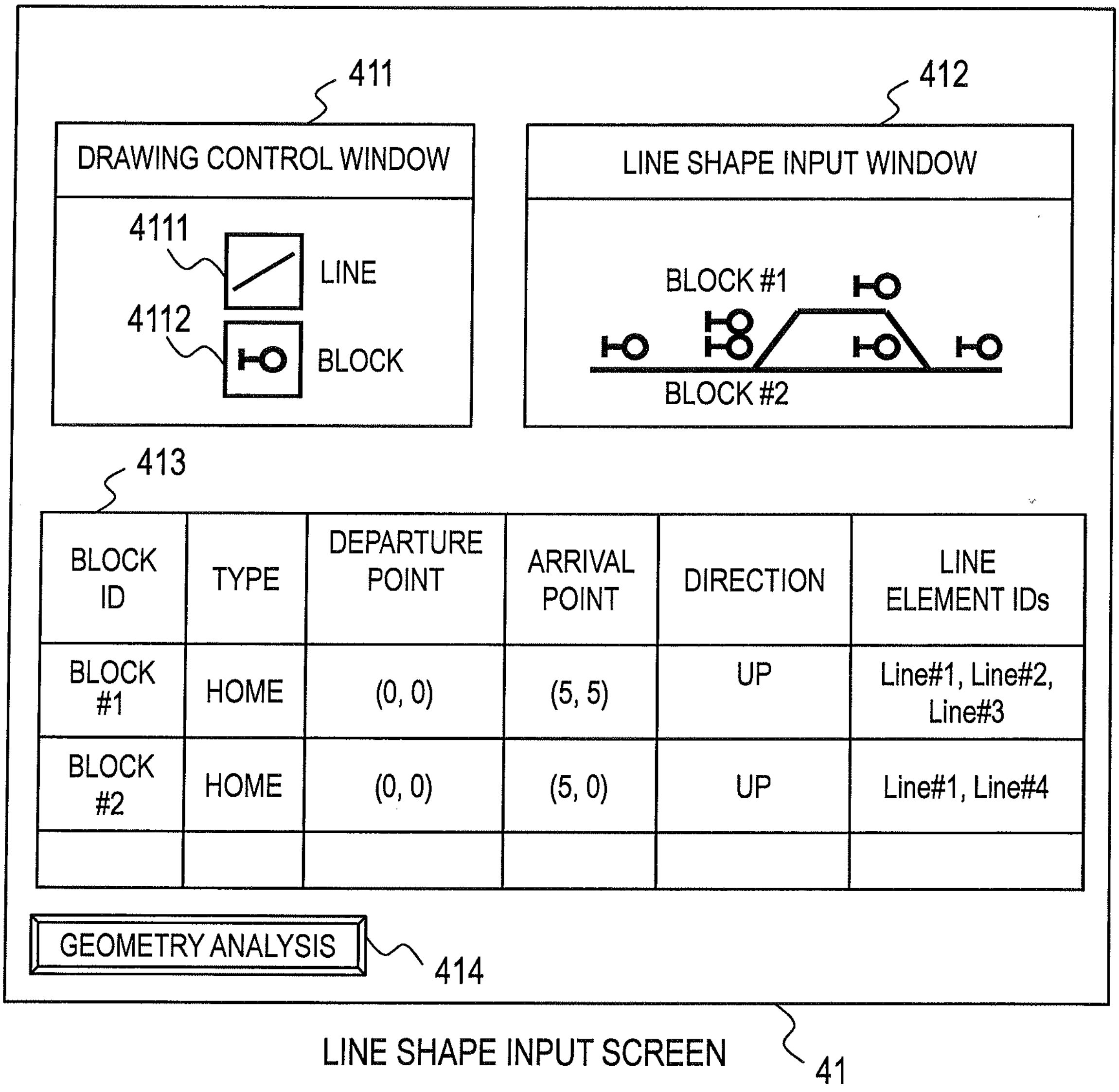

START
331
DISPLAY LINE SHAPE INPUT SCREEN
101
RECEIVE INPUT OF LINE ELEMENTS
102
RECEIVE INPUT OF BLOCKS
103
GENERATE GEOMETRY INFORMATION FROM RELATIONSHIP BETWEEN BLOCKS
104
END

Fig. 11



BLOCK ID	TYPE	DEPARTURE POINT	ARRIVAL POINT	DIRECTION	LINE ELEMENT IDs
BLOCK #1	HOME	(0, 0)	(5, 5)	UP	Line#1, Line#2, Line#3
BLOCK #2	HOME	(0, 0)	(5, 0)	UP	Line#1, Line#4

Fig. 12

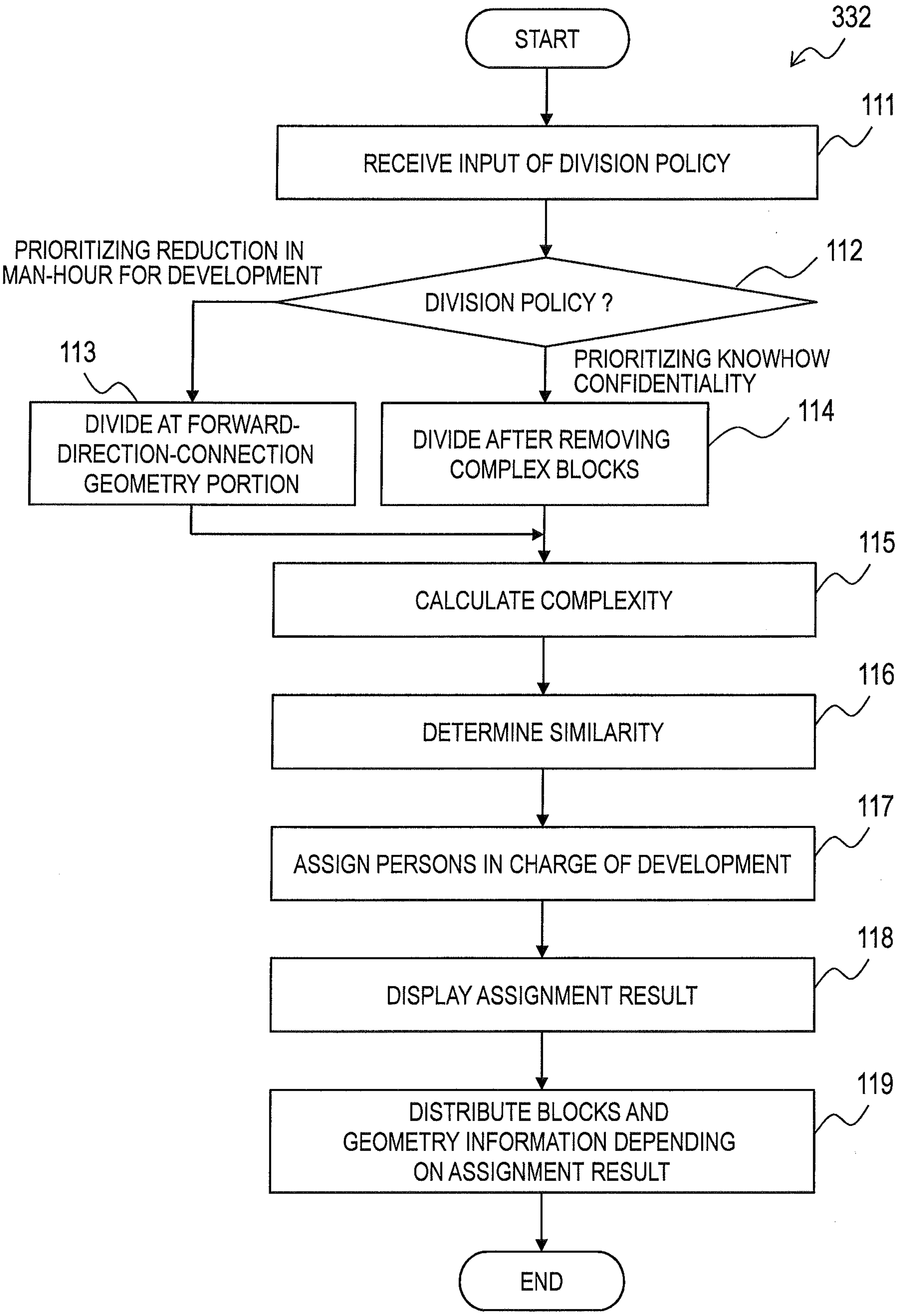
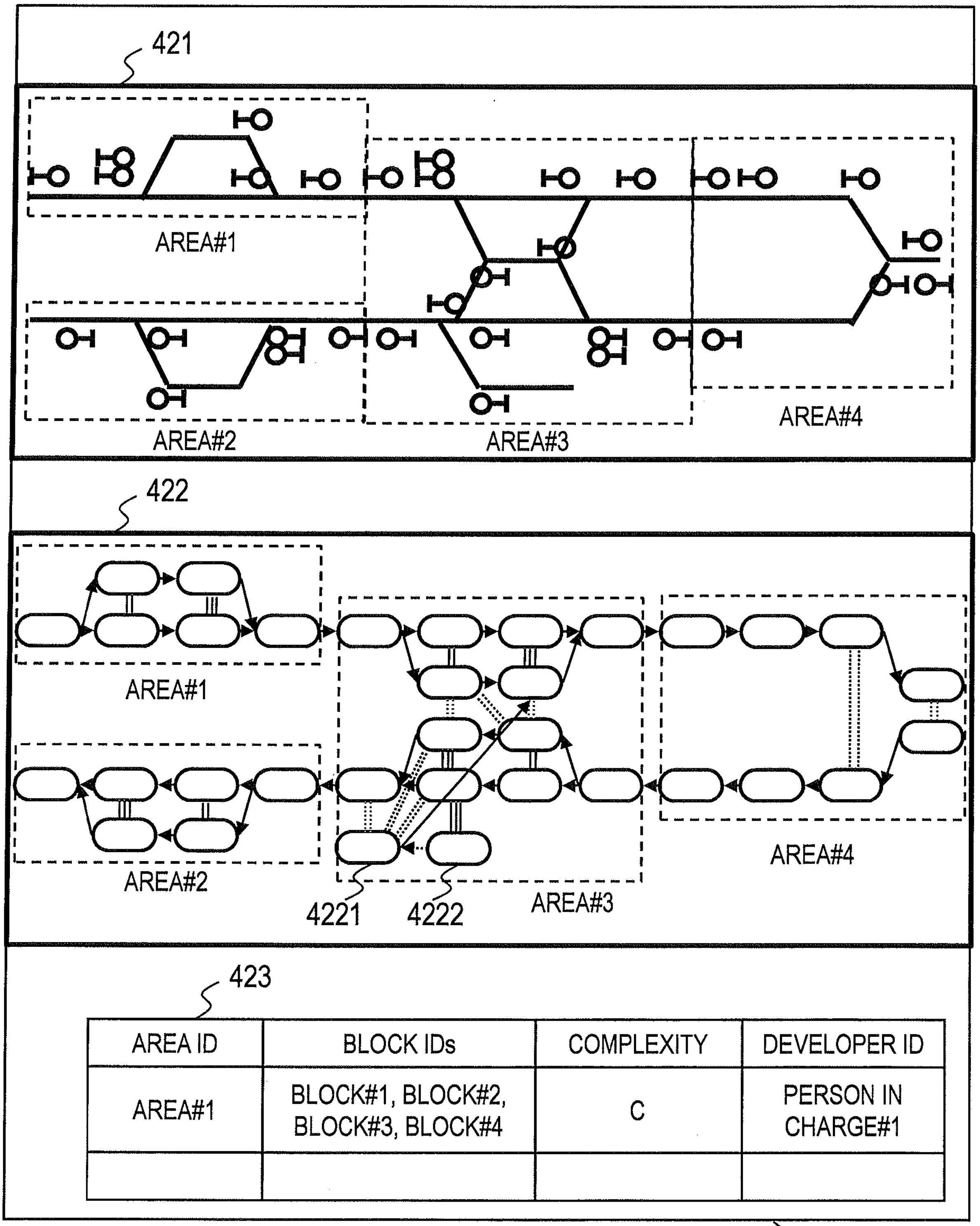

332
START
111
RECEIVE INPUT OF DIVISION POLICY
112
DIVISION POLICY ?
PRIORITIZING REDUCTION IN MAN-HOUR FOR DEVELOPMENT
PRIORITIZING KNOWHOW CONFIDENTIALITY
113
DIVIDE AT FORWARD-DIRECTION-CONNECTION GEOMETRY PORTION
114
DIVIDE AFTER REMOVING COMPLEX BLOCKS
115
CALCULATE COMPLEXITY
116
DETERMINE SIMILARITY
117
ASSIGN PERSONS IN CHARGE OF DEVELOPMENT
118
DISPLAY ASSIGNMENT RESULT
119
DISTRIBUTE BLOCKS AND GEOMETRY INFORMATION DEPENDING ON ASSIGNMENT RESULT
END

Fig. 13

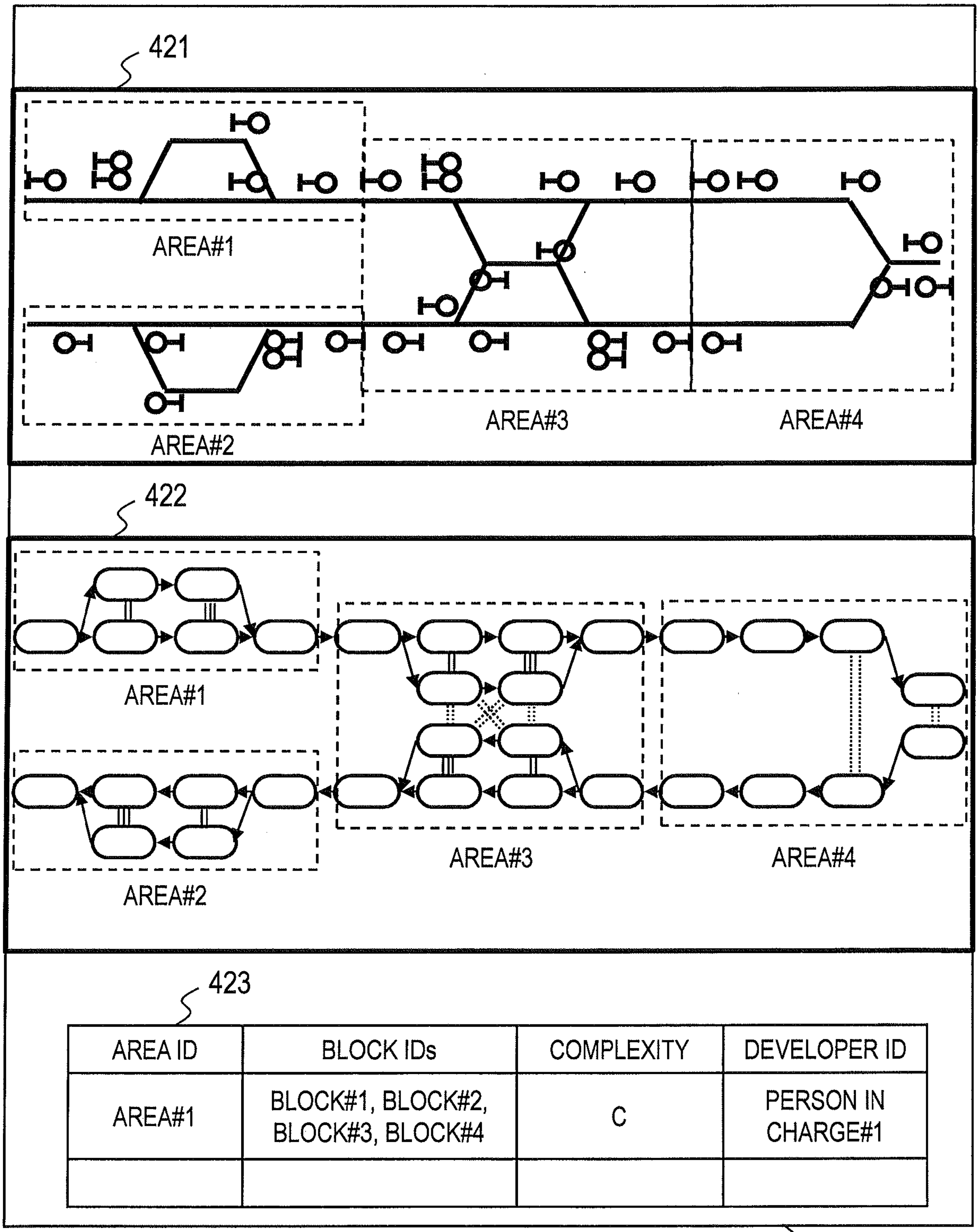


AREA ID	BLOCK IDs	COMPLEXITY	DEVELOPER ID
AREA#1	BLOCK#1, BLOCK#2, BLOCK#3, BLOCK#4	C	PERSON IN CHARGE#1

DIVISION RESULT DISPLAY SCREEN

42

Fig. 14



AREA ID	BLOCK IDs	COMPLEXITY	DEVELOPER ID
AREA#1	BLOCK#1, BLOCK#2, BLOCK#3, BLOCK#4	C	PERSON IN CHARGE#1

DIVISION RESULT DISPLAY SCREEN

42

Fig. 15

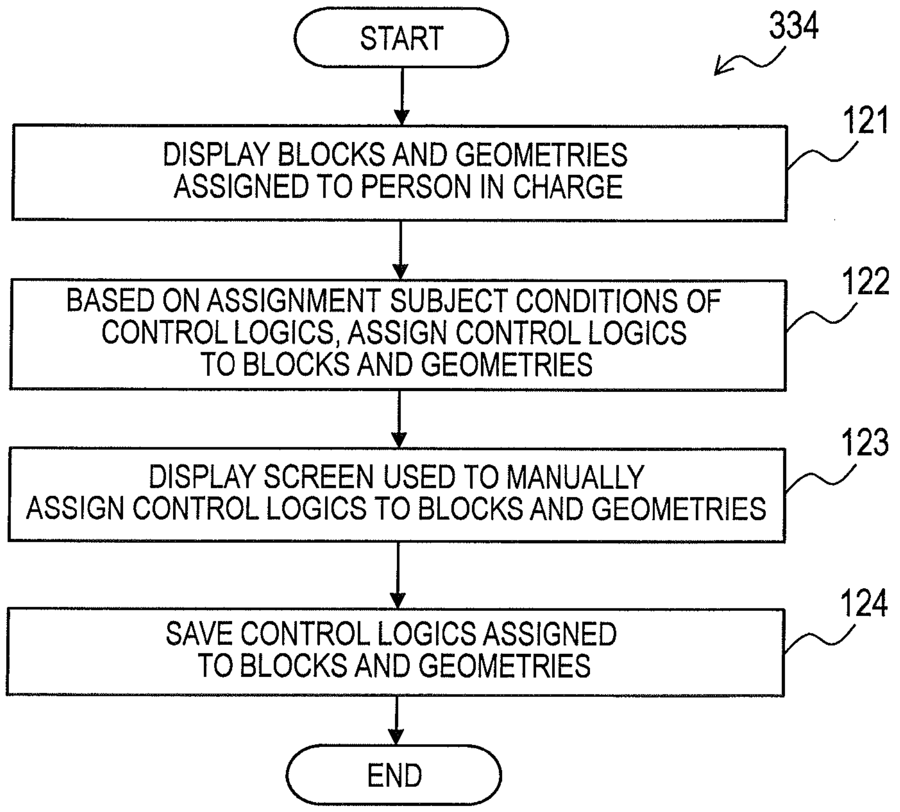
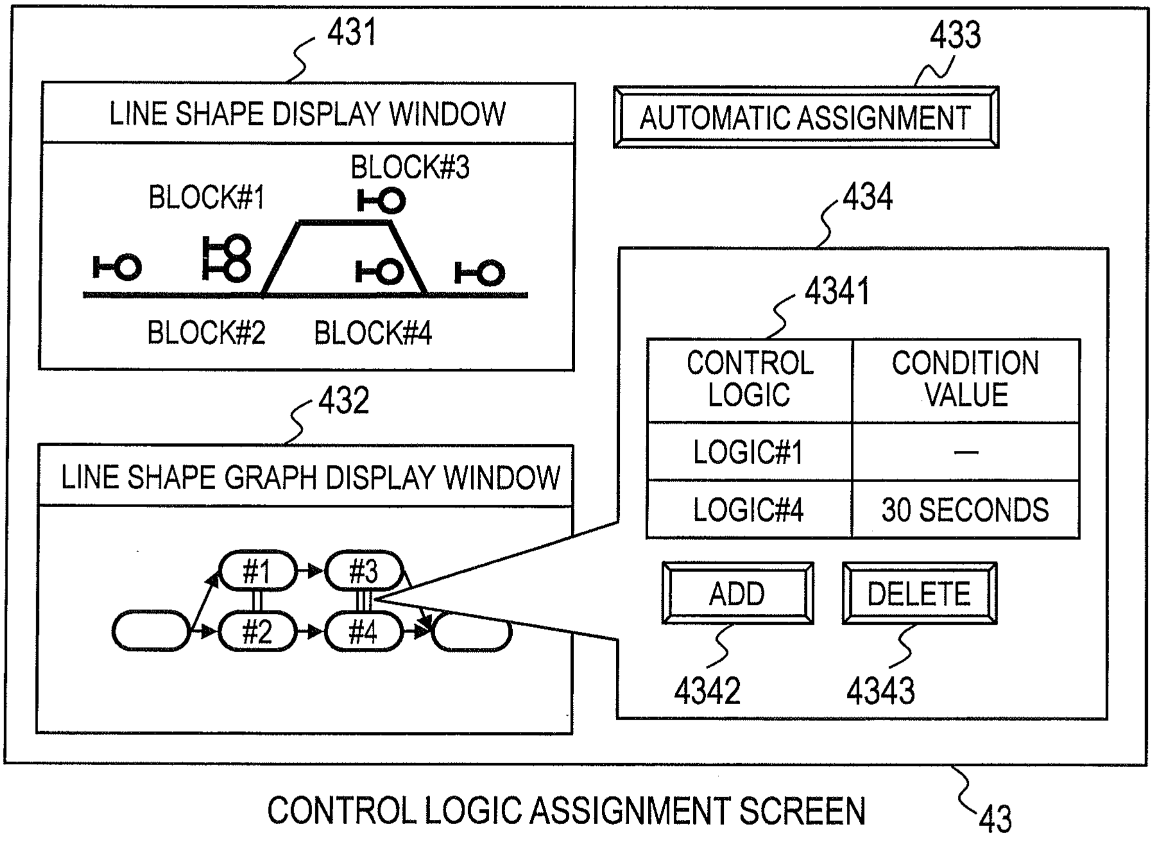

START
334
DISPLAY BLOCKS AND GEOMETRIES ASSIGNED TO PERSON IN CHARGE
121
BASED ON ASSIGNMENT SUBJECT CONDITIONS OF CONTROL LOGICS, ASSIGN CONTROL LOGICS TO BLOCKS AND GEOMETRIES
122
DISPLAY SCREEN USED TO MANUALLY ASSIGN CONTROL LOGICS TO BLOCKS AND GEOMETRIES
123
SAVE CONTROL LOGICS ASSIGNED TO BLOCKS AND GEOMETRIES
124
END

Fig. 16

CONTROL LOGIC



CONTROL LOGIC	CONDITION VALUE
LOGIC#1	—
LOGIC#4	30 SECONDS

Fig. 17

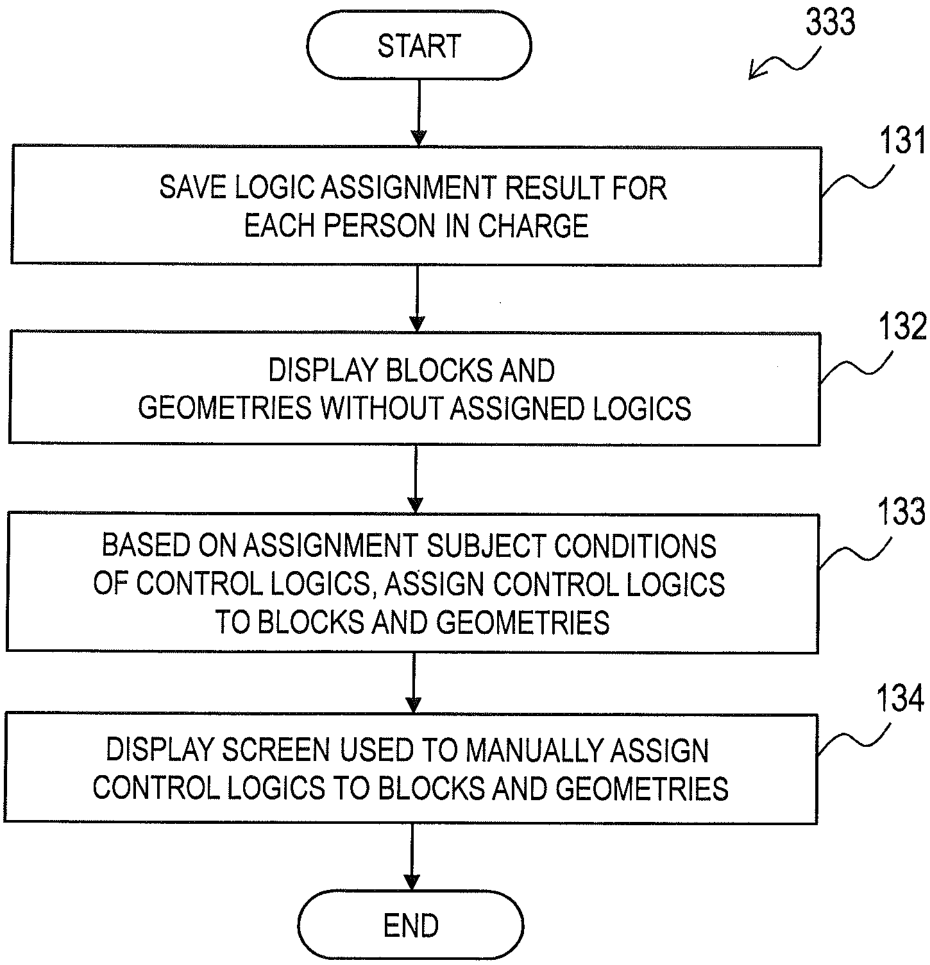
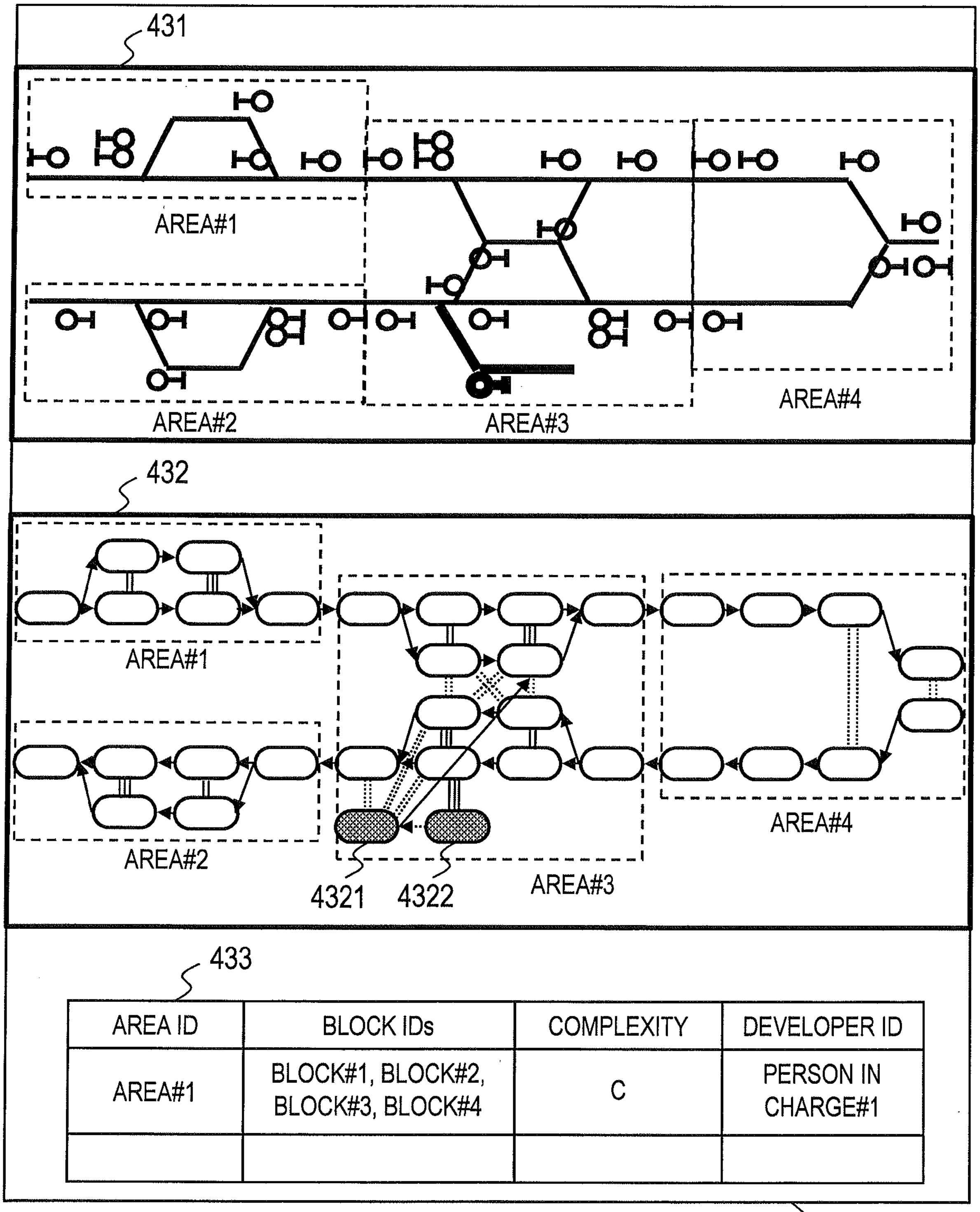

333
START
SAVE LOGIC ASSIGNMENT RESULT FOR EACH PERSON IN CHARGE
131
DISPLAY BLOCKS AND GEOMETRIES WITHOUT ASSIGNED LOGICS
132
BASED ON ASSIGNMENT SUBJECT CONDITIONS OF CONTROL LOGICS, ASSIGN CONTROL LOGICS TO BLOCKS AND GEOMETRIES
133
DISPLAY SCREEN USED TO MANUALLY ASSIGN CONTROL LOGICS TO BLOCKS AND GEOMETRIES
134
END

Fig. 18



AREA ID	BLOCK IDs	COMPLEXITY	DEVELOPER ID
AREA#1	BLOCK#1, BLOCK#2, BLOCK#3, BLOCK#4	C	PERSON IN CHARGE#1

DIVISION RESULT DISPLAY SCREEN

Fig. 19

GENERATION METHOD FOR ROUTE CONTROL PROGRAM AND COMPUTER SYSTEM

BACKGROUND OF THE INVENTION

This invention relates to a generation method for a program, and more particularly, to a method of generating a program for operating route control devices in a railroad operation management system.

In recent years, in response to problems such as the global warming, the railroad is attracting attention again as a mass transport traffic system with high energy efficiency. Then, in order to efficiently operate the railroad, an introduction of the railroad operation management system is being considered in many countries.

For the railroad operation management system, a program needs to be built depending on a line arrangement. In other words, a line arrangement varies depending on stations, and a unique program needs to be developed for each line section and station. Moreover, if the line arrangement is modified, the program needs to be corrected. Further, with the increased complexity of the line arrangement and the increased level of control in recent years, the program becomes larger in scale, and securing human resources for building a unique program for each line section and station becomes difficult. Therefore, such problems as a decrease in quality of the program, and increases in time and cost for the development are occurring.

In an electronic interlocking system, which is a subsystem of the railroad operation management system, JP 06-321107 discloses a technology for checking rationality for one route unit from a line arrangement and an interlocking table, thereby generating an interlocking database which the electronic interlocking system can use. The electronic interlocking system is a type of a railroad security device, and a system for controlling signals and points, which are in an interlocking relationship, based on an interlocking logic, is disclosed.

SUMMARY OF THE INVENTION

In such a railroad operation management system, a method which can develop a route control device common to line sections and stations different in line arrangement is required. In this case, it is generally assumed that entire design work from a line arrangement input to setting of control logics is carried out in an own company. Therefore, in order to deploy the railroad operation management system overseas, it is necessary to develop the railroad operation management system not only by the own company, but in cooperation with local vendors. In this case, it may be necessary to disclose design knowhow such as assortment of the control logics and setting of condition values outside the company. Moreover, if a distributed development is carried out, it is necessary to transmit/receive information on line shape and information on design between developing sites. These pieces of information are transmitted/received over networks, and even if the information is encrypted, information may leak due to communication interception and the like.

This invention has been made in view of above-mentioned problem, and therefore has an object to prevent a leak of knowhow outside a company in case where software is developed by dividing design work among distributed developing sites.

The representative one of inventions disclosed in this application is outlined as follows. There is provided a generation method for a route control program to be developed by a plurality of developers. The generation method is executed by a computer system including a computer including a processor for executing a program, and a memory for storing the program executed by the processor. The computer system includes a database for storing line information including a line shape and a block, and a control logic which is a component of the route control program. The generation method includes: a first step of dividing, by the processor, the line information stored in the database, and, assigning the divided line information to the plurality of developers based on a disclosure level and a skill level of each of the plurality of developers; a second step of assigning, by the processor, the control logic stored in the database to a block and a geometry included in the divided line information; and a third step of supplementing, by the processor, information for coupling the assigned control logics to each other.

According to this invention, when the railroad operation management software is developed in a distributed environment, knowhow is prevented from leaking outside a company.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention can be appreciated by the description which follows in conjunction with the following figures, wherein:

FIG. 1 is a block diagram illustrating an overall configuration of a railroad operation management system for managing an operation of trains;

FIG. 2 is a configuration diagram of hardware of a route control program generation master device and a route control program generation sub-devices according to an embodiment of this invention;.

FIG. 3 is a configuration diagram of software of a route control program generation master device and a route control program generation sub-device according to the embodiment of this invention;

FIGS. 4A and 4B are diagrams illustrating a configuration of a geometry master table according to the embodiment of this invention;

FIG. 5 is a diagram illustrating a configuration of a control logic master table according to the embodiment of this invention;

FIG. 6 is a diagram illustrating a configuration of a developer master table according to the embodiment of this invention;

FIG. 7 is a diagram illustrating a configuration of a line element table according to the embodiment of this invention;

FIG. 8 is a diagram illustrating a configuration of a block table according to the embodiment of this invention;

FIG. 9 is a diagram illustrating a configuration of a geometry table according to the embodiment of this invention;

FIG. 10 is a diagram illustrating a configuration of a area table according to the embodiment of this invention;

FIG. 11 is a flowchart of processing executed by a line shape input module according to the embodiment of this invention;

FIG. 12 is a diagram illustrating an example of a line shape input screen used by a line shape input module according to the embodiment of this invention;

FIG. 13 is a flowchart of processing executed by a dividing module according to the embodiment of this invention;

FIG. 14 is a diagram illustrating example of a division result display screen when prioritizing reduction in man-hour for development" is selected, according to the embodiment of this invention;

FIG. 15 is a diagram illustrating examples of a division result display screen when "prioritizing knowhow confidentiality" is selected, according to the embodiment of this invention;

FIG. 16 is a flowchart of processing executed by a control logic assignment module according to the embodiment of this invention;

FIG. 17 is a diagram illustrating an example of a control logic assignment screen according to the embodiment of this invention;

FIG. 18 is a flowchart of processing executed by a coupling module according to this embodiment of this invention; and

FIG. 19 is a diagram illustrating examples of a division result display screen according to the embodiment of this invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A description is now given of an embodiment of this invention referring to the drawings.

According to this embodiment, a description is given of a route control program generation system in a case where a program is developed by one management developer and a plurality of sub-developers. The management developer is a developer inside a company, and the sub-developers are developers inside or outside the company. The route control program generation system according to this embodiment divides line information input by the management developer into a plurality of areas, and assigns design work to the sub-developers. Moreover, the route control program generation system according to this embodiment couples design results by the sub-developers to one another, thereby generating a route control program.

It should be noted that, in this embodiment, "block" is a protected section by a signal, and is a line area provided in the protected section of the signal from a first protected track circuit to a last protected track circuit. A geometry is a relationship between two blocks.

FIG. 1 is a block diagram illustrating an overall configuration of the railroad operation management system for managing an operation of trains.

The railroad operation management system illustrated in FIG. 1 includes a planning system 11, a route control device 12, a ground device 13, and a ground facilities 15. The route control device 12 obtains train diagram information determined by the planning system 11 and state information on the ground facilities 15 collected by the ground device 13 via a network. Then, the route control device 12 refers to, by the processor executing a route control program, an online position of a train 14 obtained by a track circuit 151, and a departure time and a planned route contained in the train diagram information, thereby controlling signals 152 and points 153 provided on the route of the train 14 via the ground facilities 15.

FIG. 2 is a configuration diagram of hardware of a route control program generation master device 1 and a route control program generation sub-devices 2 according to the embodiment of this invention.

The route control program generation master device 1 generates a program for operating the route control device 12 by a method according to this invention.

The route control program generation master device 1 is a computer used by the management developer, and the route control program generation sub-device 2 is a computer used by the sub-developer. The route control program generation master device 1 includes a processor (CPU) 21 for executing a program, a memory 22 for storing the program and data required when the program is executed, a storage unit 23 for storing the data and the program, an input unit 24 such as a keyboard or a mouse, an output unit 25 such as a display device, and a communication unit 26 which is a network interface for communicating via a network with other devices.

The memory 22 is a high-speed and volatile storage device such as a dynamic random access memory (DRAM), and stores an operating system (OS) and application programs. By the processor 21 executing the operating system, basic functions of the route control program generation master device 1 are realized, and, by the processor 21 executing the application programs, functions provided by the route control program generation master device 1 are realized. It should be noted that the route control program generation master device 1 preferably includes an interface for reading data from storage media (such as a CD-ROM and a flash memory).

The storage unit 23 is a high-capacity and nonvolatile storage device such as a magnetic disk device and a flash memory, and stores programs executed by the processor 21 and data used when the programs are executed. In other words, the program executed by the processor 21 is read out from the storage device 23, is loaded on the memory, and is executed by the processor 21.

The route control program generation master device 1 and the route control program generation sub-devices 2 are coupled to each other via the network, and transmit/receive data via the communication units 26.

The configuration of the route control program generation master device 1 has been described, and the route control program generation sub-device 2 has the same configuration.

The route control program generation master device 1 and the route control program generation sub-devices 2 may be physically built on a single computer, or may be physically built in logical partitions constituted on one or more computers.

The route control program generation master device 1 is coupled to the route control device 12 via the network, and transfers the route control program generated by the route control program generation master device 1 to the route control device 12. The route control program generation master device 1 and the route control device 12 may not be coupled to each other, and the route control program generated by the route control program generation master device 1 may be stored in a storage medium, and may be provided via the storage medium for the route control device 12.

FIG. 3 is a configuration diagram of software of the route control program generation master device 1 and the route control program generation sub-device 2 according to the embodiment of this invention.

The route control program generation master device 1 includes, as databases, a geometry master table 311 (refer to FIG. 4), a control logic master table 312 (refer to FIG. 5), a developer master table 313 (refer to FIG. 6), a line element table 321 (refer to FIG. 7), a block table 322 (refer to FIG. 8), a geometry table 323 (refer to FIG. 9), and an area table 324 (refer to FIG. 10). These databases are stored in the storage unit 23.

Moreover, the route control program generation master device 1 includes a line shape input module 331, a dividing module 332, and a coupling module 333. Functions of these modules are realized by the processor 21 executing predetermined programs.

Further, the route control program generation master device 1 includes, as program modules for programs provided for the route control device 12, a control logic module 341 and a railroad operation management core module 342. These program modules are stored in the storage unit 23.

The route control program generation sub-device 2 includes, as databases, the geometry master table 311 (refer to FIG. 4), the control logic master table 312 (refer to FIG. 5), the line element table 321 (refer to FIG. 7), the block table 322 (refer to FIG. 8), and the geometry table 323 (refer to FIG. 9). These databases are stored in the storage unit 23.

Moreover, the route control program generation sub-device 2 includes a control logic assignment module 334. Functions of the control logic assignment module 334 are realized by the processor 21 executing a predetermined program.

Depending on a disclosure level to the sub-developer, which is described later, the control logic master table 312 of the route control program generation sub-device 2 stores control logics other than complex control logics relating to knowhow out of control logics stored in the control logic master table 312 of the route control program generation master device 1.

FIGS. 4A and 4B are is diagrams illustrating a configuration of the geometry master table 311 according to the embodiment of this invention.

The geometry master table 311 is a table for storing definition information on a geometry which is a relationship between two blocks, includes a number of overlapping terminal points between two blocks, types of overlapping terminal points, the number of terminal points in case where a block 1 includes a block 2, types of terminal points in case where the block 1 includes the block 2, the number of terminal points in case where the block 2 includes the block 1, types of terminal points in case where the block 2 includes the block 1, and information on a directional relationship between the block 1 and the block 2, and based on these pieces of information, defines types of geometry.

FIG. 5 is a diagram illustrating a configuration of the control logic master table 312 according to the embodiment of this invention.

The control logic master table 312 is a table for storing definition information on a control logic associated with a block or a geometry, and includes, as elements, determination types 3121, assignment subject conditions 3122, logic IDs 3123, and disclosure levels 3124.

The type of determination 3121 is a type of determination defined in the control logic master table 312.

The assignment subject conditions 3122 represent characteristics of a block and a geometry to which the determination is assigned. For example, the assignment subject conditions 3122 are represented by a type of geometry such as "intersection A, intersection B", and a type of block such as "starting signal". It should be noted that the assignment subject conditions 3122 may not be set.

The disclosure level 3124 is an extent to which a control logic is disclosed. The disclosure level according to the embodiment is represented by A, B, or C, and is defined so that A is the highest in security, and the security decreases in order of A, B, and C.

It should be noted that, to the control logic master table 312 of the route control program generation sub-device 2, only control logics in a range corresponding to a level which can be disclosed to the sub-developer who carries out development by using the route control program generation sub-device 2 are stored.

FIG. 6 is a diagram illustrating a configuration of the developer master table 313 according to the embodiment of this invention.

The developer master table 313 is a table for storing information on developers, and includes, as elements, developer IDs 3131 for identifying the developers, skill levels 3132 of the developers, and disclosure levels 3133 to the developers.

The developer ID 3131 is an identifier for uniquely identifying a developer. The developer ID 3131 may not be assigned to a developer, but to an organization to which the developer belongs to.

The skill level 3132 represents capability of each of the developers. Generally, the skill level of an expert developer who is engaged in the development for a long period of time is set to high, and the skill level of a novice developer who is short in period of engagement in the development is set to low. The skill level according to the embodiment is represented by A, B, or C, and is defined so that A is the highest in skill level, and the skill level decreases in order of A, B, and C.

The disclosure level 3133 is an extent of information which can be disclosed to the developer. Generally, the disclosure level of a person responsible for the development is set to highest, and the disclosure level of a developer from whom the leak of knowhow outside the company should be prevented is set to low. The disclosure level according to the embodiment is represented by A, B, or C, and is defined so that A is the highest in disclosure level, and the disclosure level decreases in order of A, B, and C. A developer can refer to control logics in the control logic master table 312 that have the disclosure level equal to or lower than the disclosure level in the developer master table 313. For example, a developer having a disclosure level "B" in the developer master table 313 can refer to control logics having disclosure levels B and C in the control logic master table 312.

FIG. 7 is a diagram illustrating a configuration of the line element table 321 according to the embodiment of this invention.

The line element table 321 holds information on line elements input to the line shape input module 331, and includes, as elements, line element IDs 3211 and coordinates 3212. The line element ID 3211 is an identifier for uniquely identifying a line element. The coordinate 3212 is information on coordinates at which the line element is arranged.

FIG. 8 is a diagram illustrating a configuration of the block table 322 according to the embodiment of this invention.

The block table 322 includes, as elements, block IDs 3221, block types 3222, departure points 3223, arrival points 3224, directions 3225, line element IDs 3226, and logic IDs 3227.

The block ID 3221 is an identifier for uniquely identifying a block. The block type 3222 represents a type of the block. For example, the block type 3222 includes a block of a home signal for entrance into a station, a block of a starting signal for departure from a station, and a block of a block signal between stations. The block type 3222 is used for the assignment subject condition 3122 in the control logic master table 312.

The departure point 3223 represents a location where a train entering into the block passes last outside the block. The arrival point 3224 represents a location where a train traveling from the block passes last inside the block. The direction 3225 represents a direction of the block. According to this embodiment, the direction 3225 is any one of up or down. The line element IDs 3226 are line element IDs of all line elements included in the block.

The logic ID 3227 is an identifier for a control logic for determining whether a train can enter the block when the route control program is executed. It should be noted that a plurality of control logics can be set to one block. The logic ID 3227 is set by the coupling module 333 and the control logic assignment module 334.

FIG. 9 is a diagram illustrating a configuration of the geometry table 323 according to the embodiment of this invention.

The geometry table 323 is a table for storing information on a geometry which is a relationship between two adjacent blocks, and includes, as elements, geometry IDs 3231, geometry types 3232, block 1 IDs 3233, block 2 IDs 3234, and logic IDs 3235.

The geometry ID 3231 is an identifier for uniquely identifying a geometry. The geometry type 3232 is type information obtained by classifying patterns of the relationships between two blocks based on the relationships defined in the geometry master table 311. The block 1 ID 3233 and the block 2 ID 3234 are block IDs of the two blocks constituting the geometry.

The logic ID 3235 is a logic ID of a control logic for determining whether a train can enter the block by referring to information on the geometry associated with the blocks when the route control program is operating. It should be noted that a plurality of control logics can be set to one geometry. The logic ID 3235 is set by the coupling module 333 and the control logic assignment module 334.

FIG. 10 is a diagram illustrating a configuration of the area table 324 according to the embodiment of this invention.

The area table 324 is a table for defining areas as units of development to be assigned to the developers, and includes, as elements, area IDs 3241, block IDs 3242, complexities 3243, and developer IDs 3244. The area is defined by a set of blocks. The information in the area table 324 is generated by the dividing module 332.

The area ID 3241 is an identifier for uniquely identifying an area. The block IDs 3242 are block IDs of blocks constituting the area. The complexity 3243 represents an indication of difficulty of development of the area. The complexity according to the embodiment is represented by A, B, or C, and is defined so that A is the highest in complexity, and the complexity decreases in order of A, B, and C. The developer ID 3244 is a developer ID of a developer who is in charge of the development of the area.

FIG. 11 is a flowchart of processing executed by the line shape input module 331 according to this embodiment of this invention, and FIG. 12 is a diagram illustrating an example of a line shape input screen 41 used by the line shape input module 331.

First, the line shape input module 331 displays the line shape input screen 41 (Step 101). As illustrated in FIG. 12, the line shape input screen 41 includes a drawing control window 411 on which information to be input is selected, a line shape input window 412 into which information on line elements, blocks, and the like is input, a block definition 413 into which definitions of the blocks are input, and a geometry analysis button 414.

Then, the line shape input module 331 prompts an input of a line shape on the line shape input screen 41 (Step 102). Specifically, a user selects a line 4111 in the drawing control window 411 on the line shape input screen 41, and inputs a line element in the line shape input window 412. Information on the input line element is saved in the line element table 321.

Moreover, the line shape input module 331 receives an input of a block on the line shape input screen 41 (Step 103). Specifically, a user selects a block 4112 in the drawing control window 411 on the line shape input screen 41, and inputs a block in the line shape input window 412. Moreover, the user inputs, for each of the input blocks, the type of the block, a departure point, an arrival point, a direction, and a line element in the block definition 413. The information input into the block definition 413 is saved in the block table 322.

Then, in case where the line shape input module 331 detects an operation on the geometry analysis button 414 by the user, the line shape input module 331 generates geometry information from a relationship between input blocks (Step 104). The line shape input module 331 refers to the geometry master table 311 based on combinations of blocks registered to the block table 332, and analyzes a geometry relationship between blocks, thereby determining the type of the geometry. The determined type of the geometry is saved in the geometry table 323.

FIG. 13 is a flowchart of processing executed by the dividing module 332 according to this embodiment of this invention, and FIGS. 14 and 15 are diagrams illustrating examples of a division result display screen 42 used by the dividing module 332.

First, the dividing module 332 receives an input of a division policy (Step 111). There are two types of division policies, specifically, "prioritizing reduction in man-hour for development" and "prioritizing knowhow confidentiality". The management developer can select a division policy out of the two policies.

Then, the dividing module 332 determines the input division policy, and branches the processing (Step 112).

In case where the input division policy is "prioritizing reduction in man-hour for development", the processing proceeds to Step 113, and the dividing module 332 divides the input route information input into the line shape input module 331 at forward-direction-coupling geometry portions (Step 113). For example, the division result display screen 42 includes, as illustrated in FIG. 14, a line shape display window 421 and a line shape graph display window 422. The line shape display window 421 displays the line information input into the line shape input module 331. The line shape graph display window 422 displays blocks input into the line shape input module 331 as rounded rectangles, and geometries as coupling lines. Out of the geometries, portions of geometry couplings in the forward direction coupling are represented by arrows. As illustrated in FIG. 14, in the case of "prioritizing reduction in man-hour for development", the line information is divided at the positions where couplings are made only by forward direction couplings, thereby generating areas. This is because, man-hours required for coupling areas can be reduced in case where a boundary of an area is constituted by forward direction coupling geometries.

On the other hand, in case where the input division policy is "prioritizing knowhow confidentiality", the processing proceeds to Step 114, and the dividing module 332 carries out division so as to remove complex blocks and geometries (Step 114). Determination of complexity can be carried out depending on whether the number of geometries relating to one block is equal to or more than a predetermined number, or whether a block or a geometry has certain geometry information. A block or a geometry may be determined to be complex in case where both of the two conditions are satisfied.

For example, on the division result display screen (FIG. 15) displayed in case of selecting "prioritizing knowhow confidentiality", compared with the display screen (FIG. 14) displayed in case of selecting "prioritizing reduction in man-hour for development", blocks 4221 and 4222 in a bottom portion of an area #3, and geometries relating thereto are removed. The block 4221 has the number of geometry relationships among blocks more than a predetermined number and the block 4222 is a block for entering into the block 4221, and the blocks 4221 and 4222 are thus complex blocks and are removed. As described above, by removing complex line shapes from a shape of a line, thereby obtaining a simple line shape, disclosure of knowhow such as information on the complex line and control logics set to the complex line shapes can be restricted.

Then, the dividing module 332 calculates complexities of areas (Step 115). The complexity is calculated based on the blocks and geometries constituting the area. For example, in the geometry master table 311, in case where a difficulty is set to a geometry pattern as a numerical value, a point of each of the areas is calculated by following "(number of blocks)×(constant value) + (sum of difficulties of geometries)", and the complexities (A to C) of the areas are assigned depending on the calculated points.

Then, the dividing module 332 determines similarities of the areas (Step 116). For example, the similarity can be determined depending on whether relationships among blocks and the geometries constituting an area are the same as relationships among blocks and geometries constituting another area. Specifically, the graph theory is applied considering the blocks and the geometries as nodes and edges, respectively. The edge includes type information represented by the geometry type 3232. On this occasion, it can be determined that the similarity exists in case where a graph constituting one area includes a graph constituting the other area.

Then, the dividing module 332 assigns developers in charge of development of the areas (Step 117). Specifically, the dividing module 332 refers to the skill level 3132 and the disclosure level 3133 in the developer master table 313 of a sub-developer, and the dividing module 332 assigns the area to the sub-developer in case where the complexity calculated in Step 115 is equal to or less than both the skill level 3132 and the disclosure level 3133. areas which are determined to have a similarity in Step 116 may be assigned to the same sub-developer in case where a plurality of areas are assigned to one sub-developer. The assignment result is registered to the developer ID 3244 of the area table 324.

According to this embodiment, the area is assigned depending on the result of comparison of the simple difficulty with the skill level and the disclosure level, but such a rule that the area is assigned to a developer having the disclosure level A may be applied in case where a specific geometry pattern, a specific combination pattern of blocks, or a specific combination pattern of geometries exists.

Then, the dividing module 332 generates information for displaying the assignment result (Step 118). As illustrated in FIGS. 14 and 15, an assignment result display screen 42 generated in Step 118 includes the line shape display window 421, the line shape graph display window 422, and an area definition window 423.

The line shape display window 421 and the line shape graph display window 422 display divided pieces of area information. The area definition window 423 displays information on the area IDs 3241, the block IDs 3242, the complexities 3243, and the developer IDs 3244 registered to the area table 324. The management developer checks the assignment result, and can change a developer of each of the areas as necessary by changing the information on the area definition window 423.

The dividing module 332 distributes, depending on the assignment result, the block and geometry information to each of the sub-developers (Step 119). Specifically, the dividing module 332 transmits, to the route control program generation sub-device 2 used by the sub-developer registered to the developer ID 3244 in the area table 324, information on the line element table 321, the block table 322, and the geometry table 323 relating to the block IDs 3242 contained in the area which the sub-developer is in charge of.

FIG. 16 is a flowchart of processing executed by the control logic assignment module 334 of the route control program generation sub-device 2 used by the sub-developer according to the embodiment of this invention, and FIG. 17 is a diagram illustrating an example of a control logic assignment screen 43 used by the control logic assignment module 334.

First, the control logic assignment module 334 displays blocks and geometries included in an area assigned to the sub-developer (Step 121). Specifically, the control logic assignment module 334 displays, on a line shape display window 431 and a line shape graph display window 432 of the control logic assignment screen 43, information on line shapes, blocks, and geometries of the area assigned to the sub-developer.

In case where the control logic assignment module 334 detects an operation by the sub-developer on an automatic assignment button 433 on the control logic assignment screen 43, the control logic assignment module 334 assigns control logics to the blocks and geometries based on assignment subject conditions of control logics (Step 122). Specifically, the control logic assignment module 334 refers to the control logic master table 312, and assigns logic IDs 3123 to blocks and geometries matching in the assignment subject conditions 3122. In case where a plurality of control logics are assigned to one type of control logic, the control logic assignment module 334 may not automatically assign a logic, or may assign a predetermined logic ID. For example, the predetermined logic can be defined in the control logic master table 312 as an attribute of the control logic.

Then, the control logic assignment module 334 displays a screen for manually assigning control logics to blocks and geometries (Step 123). the control logic assignment module 334 displays a control logic setting window 434 for setting a control logic to be assigned to the selected block or geometry after the sub-developer selects a block or a geometry in the line shape graph display window 432. The control logic setting window 434 displays a list 4341 of the control logics assigned to the block or the geometry in Step 122. Moreover, in the control logic setting window 434, condition values (such as a time for suspending determination) used in the control logics can be set. Further, by operating an add button 4342 or a delete button 4243 in the control logic setting window 434, the control logics recorded in the control logic master table 312 can be added or deleted.

Then, the control logic assignment module 334 saves the control logics assigned to the blocks and the geometries in Steps 122 and 123 (Step 124). Specifically, the control logics assigned to the blocks are saved in the logic IDs 3227 of the block table 322, and the control logics assigned to the geometries are saved in the logic IDs 3235 in the geometry table 323.

FIG. 18 is a flowchart of processing executed by the coupling module 333 according to this embodiment of this invention. As described above, the coupling module 333 generates the railroad operation management software by coupling control logic assignment results designed by the sub-developers.

First, the coupling module 333 saves the control logic assignment result by each of the developers (Step 131). Specifically, the coupling module 333 obtains information saved in the block table 322 and the geometry table 323 of the route control program generation sub-device 2, and saves the obtained logic information in the block table 322 and the geometry table 323 of the route control program generation master device 1.

Then, the coupling module 333 displays blocks and geometries to which control logics are not assigned and blocks and geometries to which control logics are assigned so that those sets of blocks and geometries can be distinguished from each other (Step 132).

Specifically, control logics are not assigned only to forward-direction-coupling geometry portions of boundaries of areas in case where it is determined that the division policy input in Step 112 of the processing carried out by the dividing module 332 is "prioritizing reduction in man-hour for development".

On the other hand, in case where it is determined that the input division policy is "prioritizing knowhow confidentiality", as in the bottom portion of the area #3 in the assignment result display screen 42 illustrated in FIG. 15, blocks and geometries are removed during the division, and hence control logics are not assigned to the removed blocks and geometries. Therefore, as illustrated in FIG. 19, blocks 4321 and 4322 which are not assigned to the sub-developers and geometries relating to the blocks 4321 and 4322 are highlighted.

Then, the coupling module 333 assigns, based on assignment subject conditions of control logics, control logics to the blocks and the geometries to which logics are not assigned (Step 133). This processing is the same as the processing in Step 122 executed by the control logic assignment module 334, and a detailed description thereof is therefore omitted.

Then, the coupling module 333 displays a screen for manually assigning control logics to blocks and geometries (Step 134). This processing is the same as the processing in Step 123 executed by the control logic assignment module 334, and a detailed description thereof is therefore omitted.

Then, the coupling module 333 refers to the block table 322, the geometry table 323, and the railroad operation management core module 342, and couples control logic modules 341 specified by the logic IDs 3227 in the block table 322 and the logic IDs 3235 in the geometry table 323, thereby generating the route control program.

As described above, according to this embodiment, the extent of design which the developer is in charge of is assigned depending on the disclosure level to the developer, and hence leak of knowhow can be prevented, and the risk in the security can thus be reduced. In other words, while the knowhow is prevented from leaking by setting control logics for complex line shapes for which the knowhow is accumulated in the company, and outsourcing setting of control logics to simple line shapes outside the company, the development work can be efficiently carried out by sharing the work.

Moreover, through the division of the line information at the forward-direction-coupling geometry portions so that areas are independent of one another to generate the areas, while knowhow is prevented from leaking, an amount of work required for the management developer to couple the design results can be reduced.

Further, the extent of design which the developer is in charge of can be assigned depending on the skill of the developer.

WHAT IS CLAIMED IS:

1. A generation method for a route control program to be developed by a plurality of developers,

the generation method being executed by a computer system including a computer including a processor for executing a program, and a memory for storing the program executed by the processor,

the computer system including a database for storing line information including a line shape and a block, and a control logic which is a component of the route control program,

the generation method including:

a first step of dividing, by the processor, the line information stored in the database, and, assigning the divided line information to the plurality of developers based on a disclosure level and a skill level of each of the plurality of developers;

a second step of assigning, by the processor, the control logic stored in the database to a block and a geometry included in the divided line information; and

a third step of supplementing, by the processor, information for coupling the assigned control logics to each other.

2. The generation method for a route control program according to claim 1, wherein:

the database stores information on the geometry corresponding to the line shape and information on a definition of the block included in the line information, and information on a disclosure level and a skill level of a person in charge of development; and

in the first step the processor

extracts the geometry based on the line shape and the block included in the line information stored in the database, calculates complexity of the line information from information on the block and the extracted geometry, and

assigns the divided line information to a developer having the disclosure level and the skill level satisfied by the calculated complexity.

3. The generation method for a route control program according to claim 1 or 2, including a step of receiving a policy for dividing the line information,

wherein in the first step the processor removes a geometry of high complexity and a block relating to the geometry, and then divides the line information, in case where the received policy prioritizes confidentiality of knowhow.

4. The generation method for a route control program according to claim 3, wherein in the third step the processor assigns a control logic corresponding to a divided portion of the line information, and assigning a control logic corresponding to the block and the geometry removed from the line information when the line information is divided, thereby supplementing information.

5. The generation method for a route control program according to claim 1 or 2, including a step of receiving a policy for dividing the line information,

wherein in the first step the processor divides the line information at a position including only a forward-direction-coupling geometry in case where the received policy prioritizes a reduction in man-hour for development.

6. A computer system for use by a plurality of developers in

developing a route control program, comprising a first computer and a second computer, wherein:

each of the first computer and the second computer includes a processor for executing a program and a memory for storing the program executed by the processor;

the first computer includes

a database for storing line information including a line shape and a block, and a control logic which is a component of the route control program, and

a dividing module for dividing the line information stored in the database, assigning the divided line information to the plurality of developers based on a disclosure level and a skill level of each of the plurality of developers, and transmitting the assigned line information and a necessary control logic to the second computer;

the second computer includes a control logic assignment module for assigning the control logic transmitted from the first computer to a block and a geometry included in the divided line information; and

the first computer includes a coupling module for supplementing information for coupling information on the assigned control logics to each other.

7. The computer system according to claim 6, wherein:

the database stores information on the geometry corresponding to the line shape and information on a definition of the block included in the line information, and information on a disclosure level and a skill level of a person in charge of development; and

the dividing module is configured to

extract the geometry based on the line shape and the block included in the line information stored in the database, calculate complexity of the line information from information on the block and the extracted geometry, and

assign the divided line information to a developer having the disclosure level and the skill level satisfied by the calculated complexity.

8. The computer system according to claim 6 or 7, wherein:

the first computer includes an input unit for receiving a policy for dividing the line information; and

the dividing module removes a geometry of high complexity and a block relating to the geometry, and then divides the line information, in case where the received policy prioritizes confidentiality of knowhow.

9. The computer system according to claim 8, wherein the coupling module assigns a control logic corresponding to a divided portion of the line information, and assigns a control logic corresponding to the block and the geometry removed from the line information when the line information is divided, thereby compensating information.

10. The computer system according to claim 6 or 7, wherein:

the first computer includes an input unit for receiving a policy for dividing the line information; and

the dividing module divides the line information at a position including only a forward-direction-coupling geometry in case where the received policy prioritizes a reduction in man-hour for development.

Application No:	GB1303691.8	**Examiner:**	Mr Nigel Hanley
Claims searched:	1-10	**Date of search:**	10 September 2013

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
Y	1-10	US 2008/0154452 A1 KAPP - See whole document especially paragraphs 0029-0037. Note the structure of a track database comprising full details of the track and wayside equipment and its role in generating a route.
Y	1-10	US 2010/0063657 A1 KUMAR - See whole document especially Fig 2 & 4 and paragraphs 18-22. Note the use of a track database in generating a route.
Y	1-10	US 2009/014927 A1 GEN ELECTRIC - See whole document especially Fig 1.Note the control of an area of a route under different dispatchers and their role in adjusting the route in the area under their control with respect to the neighbouring areas.
A,E	1-10	WO 2013/014989 A1 HITACHI - See whole document. Note use of a track database in a route control generation program.

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{X} :

Worldwide search of patent documents classified in the following areas of the IPC

The following online and other databases have been used in the preparation of this search report

Online: WPI, EPODOC, TXTE

International Classification:

Subclass	Subgroup	Valid From
B61L	0019/00	01/01/2006
B61L	0027/00	01/01/2006