



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
04.01.2012 Bulletin 2012/01

(51) Int Cl.:
B61L 27/00 (2006.01)

(21) Application number: **10006446.8**

(22) Date of filing: **22.06.2010**

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

(71) Applicant: **Universität Stuttgart**
70174 Stuttgart (DE)

(72) Inventors:
• **Martin, Ullrich, Prof. Dr.-Ing.**
70569 Stuttgart (DE)
• **Cui, Yong**
70569 Stuttgart (DE)

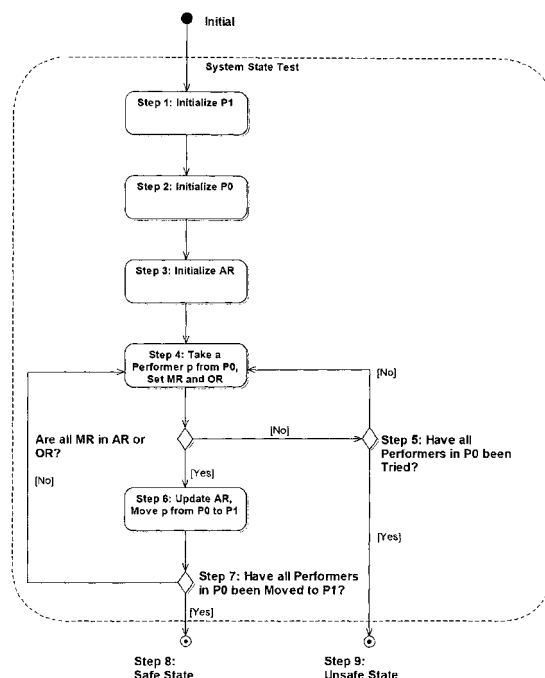
(74) Representative: **Mergel, Volker**
Blumbach - Zinngrebe
Patentanwälte
Alexandrastrasse 5
65187 Wiesbaden (DE)

(54) **Method and system for simulating, planning and/or controlling operating processes in a track guided transportation system**

(57) For deadlock avoidance in a track guided transportation system the invention proposes a method for simulating, planning and/or controlling operating processes of a track guided transportation system, wherein resources are defined which represent track segments of the transportation system, performers are defined which represent transportation vehicles moving into, within and/or out of the transportation system, wherein for each performer at least one associated route with a starting location and a destination location is defined, at least one request from a performer is provided requesting to occupy at least one resource, for each of said at least one request a corresponding system state is determined as safe or unsafe, and- for deadlock avoidance only requests are granted for which a safe state is determined.

The invention further proposes a system and a digital storage medium for performing the inventive method.

Fig. 1



DescriptionField of the invention

5 **[0001]** The invention relates to track guided transportation systems in general, and especially to a method and a system for simulating, planning and/or controlling operating processes in a track guided transportation system.

Background of the invention

10 **[0002]** In track guided transportation networks situations may occur, in which allowing a certain movement into a track segment makes it impossible to continue operation, because all further movements block each other. Such a situation is called deadlock. In the field of railway operation, a deadlock is accordingly defined as a situation in which a number of trains cannot continue their path at all, because every train is blocked by another one.

15 **[0003]** Known methods of deadlock avoidance comprise the movement consequence analysis (MCA) described in the dissertation by J. Pahl "Steuerlogik für Zuglenkanlagen zum Einsatz unter stochastischen Betriebsbedingungen", Technische Universität Braunschweig, 1993, the dynamic route reservation (DRR) described in "Avoiding Deadlocks in Synchronous Railway Simulations" by J. Pahl, Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis, 28 - 30 March, 2007, Hannover, Germany (<http://www.digibib.tu-bs.de/?docid=00020794>), the Petersen and Taylor algorithm described in "Line Block Prevention in Rail Line Dispatch" by E.R. Petersen and A.J. Tayler, INFOR Journal 21, 1983, No. 1, pages, 46-51, and the labeling algorithm described in "The effects of deadlock avoidance on rail network capacity and performance" by G. Mills and P. Pudney, Proceedings of the 2003 Mathematics-in-Industry Study Group, published 2008, pages 49-63.

20 **[0004]** The MCA and the Petersen and Taylor algorithm however have the drawback that many solutions of certain operating situations which are not a deadlock are however falsely indicated as deadlocks. These methods therefore are of very limited or no practical use, since too few valid solutions are detected. Furthermore, the DRR, the Petersen and Taylor algorithm and the labeling algorithm can only be employed in simple infrastructure networks and accordingly are of limited or no use for complex infrastructures or operating conditions.

25 **[0005]** From US 2006/0212188 A1 a method and apparatus for automatic selection of alternative routing through congested areas using congestion prediction metrics is known. Such a method might be useful for reducing the number of deadlocks, it is however not suitable for deadlock avoidance.

30 **[0006]** From US 7 428 452 B2 for instance a scheduling method and system for rail networks is known which uses the technique of problem space search, wherein a rich set of potential schedules is randomly searched discarding schedules which lead to deadlock. Due to the employed random search, however, this method requires a very high computational cost, thereby limiting its practical use.

35 **[0007]** It is therefore an object of the present invention to show a new and improved way for deadlock avoidance in a track guided transportation system, which in particular may be utilized for simulating, planning and/or controlling a track guided transportation system.

Summary of the Invention

40 **[0008]** The inventive solution of the object is achieved by each of the subject matter of the respective attached independent claims. Advantageous and/or preferred embodiments or refinements are the subject matter of the respective attached dependent claims.

45 **[0009]** Accordingly, an inventive method for simulating, planning and/or controlling operating processes of a track guided transportation system firstly comprises the step of defining resources which represent the transportation system, wherein in particular resources are defined for individual track segments. The track guided transportation system may exemplarily be a railway system. Resources may preferably defined for a complete track guided transportation network or only for parts thereof, for instance for a pre-determined area of a network to be observed. With advantage the defined resources are associated with respective parameters such as type of track segment, length of track segment and/or connections to other resources. Preferably the track guided transportation network under observation may be represented by a graph comprising the defined resources and their interconnections. The method further proposes to define performers which represent transportation vehicles moving into, within and/or out of the transportation system, wherein for each performer at least one associated route with a starting location and a destination location is defined. Preferably the routes associated with the respective performers are calculated by applying a route searching algorithm to the graph which represents the track guided transportation network under observation. Further, at least one request from a performer is provided, requesting to occupy at least one resource, and for each of said at least one request a corresponding system state is determined as safe or unsafe, wherein only requests are granted for which a safe state is determined, thereby ensuring deadlock avoidance. Any request for which a corresponding system state is determined as unsafe is denied.

However, not every request for which a corresponding safe system state is determined necessarily is granted, since additional boundary conditions might be provided, like for instance different priorities associated with different requesting performers.

[0010] Simulating, planning and/or controlling operating processes of a track guided transportation system preferably comprises the step of at least partially automatic dispatching, in particular dispatching in a railway operation control system.

[0011] A basic idea of the invention is the utilization of the per se known banker's algorithm in the context of simulating, planning and/or controlling a track guided transportation system. Hence, the system state preferably is determined utilizing the banker's algorithm. Most preferably the banker's algorithm is utilized in synchronous simulation of a track guided transportation system or network, wherein simulation results with special advantage are utilized for planning and/or controlling operating processes of the track guided transportation system, and in particular for dispatching purposes.

[0012] A general description of the banker's algorithm is given for instance in "The mathematics behind the banker's algorithm" by E.W. Dijkstra, New York, Springer, 1982, pages 308-312, the relevant disclosure of which is incorporated herein by reference.

[0013] In a preferred way of employing the banker's algorithm the step of determining the system state comprises the steps of

- a) determining a set of moving performers, comprising all performers which currently move along their associated routes,
- b) determining for each performer in the set of moving performers the respective occupied resources (OR), wherein said occupied resources comprise any resources requested to be occupied by the respective performer,
- c) determining as available resources all resources not occupied by a performer,
- d) initializing a set of feasible performers as an empty set,
- e) selecting a performer from the set of moving performers based on a pre-defined order,
- f) determining for the selected performer the resources required to perform a movement based on its associated route,
- g) checking for the selected performer whether the required resources are either available resources or resources presently occupied by the selected performer, wherein if the check is positive, the selected performer is moved from the set of moving performers to the set of feasible performers and the available resources are updated based on the premise that the selected performer performs the movement based on its associated route,
- h) repeating steps e) to g) until either all moving performers have been moved to the set of feasible performers or until no further moving performer can be moved to the set of feasible performers, and
- i) determining a safe system state if all moving performers have been moved to the set of feasible performers and otherwise determining an unsafe system state.

[0014] In order to adapt the banker's algorithm for simulating, planning and/or controlling operating processes of a track guided transportation system, preferably certain modifications are made to the known banker's algorithm.

[0015] The modifications in particular have the purpose of increasing the solution space for determining a safe system state and accordingly reducing the solution space for determining an unsafe system state. This is possible, since not all unsafe system states determined using the unmodified banker's algorithm lead to a deadlock and hence are so-called false positives.

[0016] According to a first preferred modification the request of a moving performer which has been moved to the set of feasible performers is granted, although an unsafe system state is determined, if no resource of the destination location of its associated route is requested by any moving performer which has not been moved to the set of feasible performers, or a resource of the destination location of its associated route is requested by a moving performer which has not been moved to the set of feasible performers, but it is occupied by another performer. Thereby requests of performers which are not the cause for the determined unsafe system state are granted, and thus the possibility of deadlocks due to an increasing number of such innocent performers being blocked is advantageously reduced.

[0017] The defined resources typically can be categorized into junction-type and non-junction type resources, wherein a non-junction-type resource in particular represents a track segment which is connectable to only two further track segments, typically one at either end, and wherein a junction-type resource represents a track segment which is connectable to more than two further track segments.

[0018] According to a second preferred modification of the banker's algorithm the step of determining a system state is repeated after determining an unsafe system state if a feasible track is identifiable for a requesting performer. A track is identified as a feasible track, if the track lies within the route associated with the requesting performer and all resources in the route between the currently occupied resources of the requesting performer and the track are available resources, i.e. are free to enter, the track comprises sufficient resources to hold the requesting performer, and - when the track is occupied by the requesting performer - at least one alternative route exists between the currently occupied resources

of the requesting performer and the next junction-type resource behind the track. Available resources again shall identify all resources not occupied by a performer.

[0019] In an advantageous embodiment of the method the feasible track and the requesting performer each are associated with a respective length, wherein the length associated with the feasible track is equal or larger than the length associated with the requesting performer. The feasible track may also comprise more than one track segment, i.e. more than one defined resource, wherein each defined resource preferably is associated with a length, so that the length of the feasible track is the sum of the lengths of the respective resources making up the feasible track.

[0020] The order in which the performers are selected may also have an influence on the determined system state and may result in a false positive. Therefore according to a third preferred modification of the banker's algorithm with advantage the order in which a performer is selected from the set of moving performers in step e), as described above, is defined such that first all performers are selected, for which the destination location of their respective associated route lies outside the defined resources, second all performers are selected, for which the destination location of their respective associated route comprises resources which represent a dead-end track segment, and third all remaining performers are selected.

[0021] In a track guided transportation network typically several alternative routes exist for connecting two given locations. A fourth preferred modification of the banker's algorithm is based on at least one performer being associated with at least two routes, wherein if an unsafe system state is determined, a moving performer which has not been moved to the set of feasible performers is associated with an alternative route, the system state is determined again and granting and/or denying requests is performed in response to the newly determined system state.

[0022] Preferably the defined resources and their interconnections are represented by a graph and based on that graph routes are determined by utilizing a pre-defined route searching algorithm. In the above described fourth modification the alternative route may then with special advantage be determined by applying said route searching algorithm to a modified graph from which at least one resource within the original route of the moving performer which is neither an available resource or a resource presently occupied by the moving performer is removed. Thereby an alternative route is forced to be calculated which does not comprise the resource which presumably is the cause of the determined unsafe system state.

[0023] A fifth preferred modification of the banker's algorithm relates to system performance in that the system state is not determined for certain requests. In particular a system state is determined only depending on requests requesting to occupy at least one resource which is not a non-junction type resource. Preferably a request requesting to occupy a non-junction type resource is simply granted if the respective resource is an available resource without determining a corresponding system state for such a request.

[0024] Since applying the above described modifications of the banker's algorithm typically leads to an increased processing and/or calculation cost, the modifications preferably are only applied when necessary. With advantage for instance a maximum waiting time may be defined and if a request has not been granted after said maximum waiting time, the steps for determining the system state are altered, i.e. one or more of the above described modifications of the banker's algorithm are employed. The maximum waiting time may be defined generally, individually for a request, individually for a requesting performer, and/or depending on further parameters such as for instance pre-defined priorities or other user-defined parameters.

[0025] That way processing-intensive modifications are applied only when necessary, since before the defined maximum waiting time is exceeded, it is still possible that an observed deadlock situation does not exist any longer due to a changed operation situation. Accordingly an increase in system performance can be achieved.

[0026] The described method most preferably allows for performers simultaneously occupying more than one resource and/or for performers dividing into at least two separate performers and/or at least two performers combining into one common performer during the simulation. That way the method may be utilized flexibly for even very complex track guided transportation networks.

[0027] An inventive system for simulating, planning and/or controlling operating processes of a track guided transportation system is adapted to perform a method as described above. Preferably the system is adapted for automated, semi-automated or computer aided steering of operating processes of the track guided transportation system. In particular the system may be provided as a dispatching system adapted for dispatching in an operation control system.

[0028] At least part of the functionality of the inventive system may preferably be provided by software components. Accordingly, also a digital storage medium lies within the scope of the invention, which comprises electronically readable control instructions adapted to perform, when executed in at least one computer, a method as described above.

Brief Description of the Figures

[0029] It is shown in

Fig. 1 schematically the steps of the banker's algorithm as utilized for preferred embodiment of the inventive method,

- Fig. 2 schematically a deadlock situation in a railway system,
- Fig. 3 schematically the abstraction of a macroscopic network,
- 5 Fig. 4 schematically a deadlock situation without considering potential state transitions,
- Fig. 5 schematically a situation without deadlocks after all the requests are approved for the situation shown in Fig. 4,
- Fig. 6 schematically an example of a performer with an internal destination,
- 10 Fig. 7 the situation as shown in Fig. 6 with train Z4 moved to a feasible track,
- Fig. 8 a possible movement arrangement after Z4 blocked G2 in the situation as shown in Fig. 7,
- 15 Fig. 9 schematically a situation in which fixed routes result in an unsafe system state, and
- Fig. 10 a schematic flowchart with steps for improvement of the banker's algorithm with alternative routes.

Detailed Description of the Invention

20 **[0030]** Subsequently, preferred but exemplar embodiments of the invention are described in more detail with regard to the figures.

[0031] In the following embodiments exemplary a synchronous simulation for simulating the operating process of a railway system is described. Such a synchronous simulation preferably forms the basis for automatic and/or semiauto-

25 matic planning, controlling and/or dispatching of the railway system.

[0032] In contrast to asynchronous simulation, deadlocks may be produced if there are track sections with bi-directional operations in synchronous simulation. In a synchronous simulation, to resolve deadlock problems is a necessary feature to prevent simulation failure. In the field of railway operation, a deadlock is defined as a situation in which a number of trains cannot continue their path at all, because every train is blocked by another one.

30 **[0033]** The deadlock problem for which the invention provides a solution however does not only occur in railway synchronous simulation, but also in numerous other fields such as for instance computer science or telecommunications.

[0034] In the following, trains are named with the prefix "Z". A simple example of a deadlock situation is shown in Fig. 1. As shown in the top part of Fig. 2, trains Z1, Z2 and Z3 are going to run along the routes R1, R2 and R3 respectively. The correct sequence of the train movements should be: Z1, Z3 and Z2. If they ran in another order, as shown in the

35 bottom part of Fig. 2, wherein train Z3 instead of train Z1 gets the chance to reach track G3, then train Z1 is waiting for train Z3 to leave track G3, whereas train Z3 is waiting for train Z1 to leave track G1. Hence train Z1 and train Z3 are blocked by each other and a deadlock situation occurs.

[0035] There are four necessary conditions, known as COFFMAN conditions, that lead to deadlocks. These are:

- 40 1) The "mutual exclusion" condition: Tasks claim to the requested resources exclusively. In a railway simulation, that means the infrastructure resources are not allowed to be entered when it is blocked by another simulation performer, except for shunting processes, e.g. when two shunting sets are supposed to be coupled, they are allowed to enter the same infrastructure resource.
- 45 2) The "wait for" condition: Tasks hold the blocked resources when waiting for the new required resources. In a railway simulation, a simulation performer may wait until the required resources are released, while the resources blocked by the performer are not allowed to be allocated to other trains.
- 3) The "no preemption" condition: A resource cannot be forcibly removed from the occupier until it is released by the task explicitly. This condition is also fit in railway simulation.
- 50 4) The "circular wait" condition: Two or more tasks form a circular chain, in which each task holds one or more resources requested by other tasks. In the example shown in Fig. 1, train Z1 and Z3 form such a circular chain, each train requests the resource blocked by the other one.

[0036] Deadlock avoidance is achieved by examining the system state dynamically before allocating a resource to a requester. A request is only granted if the system will still keep in a safe state. A safe state means the resources can

55 be allocated to each task in some order without leading to deadlock. In the following the test for system state is also referred to as deadlock-free test. To execute a deadlock-free test, the potential usage of resources typically is known in advance.

[0037] As long as a request does not lead the system to an unsafe state, the requested resource can be allocated to

the requester, even if all of the above mentioned four necessary conditions of deadlock are satisfied. Therefore, deadlock avoidance can deal with deadlock problems with high resource utilization.

[0038] A typical characteristic of deadlock avoidance is that a safe state can ensure the operation being out of deadlocks, however, an unsafe state does not always sufficiently lead to deadlocks. For deadlock-free tests, the situation called "false positive" may occur: a deadlock-free test result that is read as positive but actually is negative. A false positive test shows evidence of a deadlock when it not actually present.

[0039] The banker's algorithm is based on the approach of deadlock avoidance. The principle of deadlock avoidance is to grant a request only if that request will not lead the system to an unsafe state. The banker's algorithm is used to test the system state for deadlocks in order to determine whether an infrastructure resource is allowed to be allocated to a requester.

[0040] In railway synchronous simulation, the structure of the banker's algorithm fits the simulation model quite well. The money of a banking system, i.e. a resource, is mapped to the infrastructure of a railway network. A debtor is mapped to a simulation performer that requests infrastructure resources to the railway operating control system. A loan program, i.e. a process, is similar to a simulation movement task. In addition, the prediction of the maximal requested resources is easier to be handled in a railway simulation system than in a computer operating system. A movement task always specifies its target, from which the maximum requested infrastructure resources can be predicted via its route. Especially in a dispatching system, the default route and the request resources can be obtained based on the predefined train path in a timetable.

[0041] A request can be granted if the system will still be in a safe state in case the requested resources are allocated. For this reason, the banker's algorithm can be regarded as an algorithm to test a system state. Two sets of data structure need to be managed for the banker's algorithm: the processes and the resources. The purpose of the banker's algorithm is to try to allocate resources for each process in some order. If a process, i.e. a movement task, can get all the required resources, it can be regarded as a feasible task. A request can be approved if all the processes are proven as feasible tasks.

[0042] In a railway synchronous simulation, the processes are the movement tasks. Since a movement task is always related to one simulation performer and a simulation performer can only execute one movement task in a certain point of time, the performer of a movement task is chosen to identify a process. Two performer sets are used to represent two groups of processes, i.e. movement simulation tasks, in the banker's algorithm:

- P1: The performer with a feasible task will be included.
- P0: The performer, of which the simulation task has not yet been proven as a feasible task, will be included.

[0043] For a microscopic simulation, the resources in the banker's algorithm are the infrastructure resources of the track guided transportation network. Each resource is initialized with exact one instance. For a macroscopic simulation, a resource may be defined based on a large section, e.g. a station track section or a line section. A resource in a macroscopic infrastructure model can be initialized with many instances according to its capacity. Supporting a resource type with multiple instances is an important feature of the banker's algorithm.

[0044] In the banker's algorithm, three sets of resources are defined:

- Currently available resources (AR): The resources, those can be entered by a simulation performer.
- Currently occupied resources (OR): The resources, those are already blocked by the performer of the current concerned process or movement task.
- Maximal required resources (MR): The predicted maximal required resources for completing the currently concerned process or movement task.

[0045] The key to check the system state in the Banker's algorithm is to find a feasible task. If such a feasible task does not exist, the system state is unsafe. Otherwise, even in the case that only one feasible task is found, the feasible task is still able to be completed since it obtains all required resources. More important, the resources blocked by that feasible task will be returned to AR after the simulation task is completed. Therefore, the currently available resources are increased and the other tasks may get more chances to be completed. More feasible tasks are found, more resources are returned. A safe state will be concluded when all the simulation tasks are feasible tasks. The detailed steps of the banker's algorithm are shown in Fig. 1:

Step 1: Initialize an empty performer set P1.

Step 2: Initialize a performer set P0, in which the performer with a movement task is included.

Step 3: Initialize a set AR, in which all of the currently available resources are included.

Step 4: Take a performer p from P0, and initialize MR and OR for that performer.

Step 5: When there is a resource in MR that does not exist in AR or OR: go back to Step 4 if not all the performers

in P0 have been checked, otherwise go to step 9.

Step 6: When all resources in MR exist in AR or OR: all the resources of OR are released and returned to AR; if the destination of the movement task is still inside the observed network and the resource of the destination is in AR, the resource of the destination should be removed from AR; at last the current performer p is added to P1 and removed from P0.

Step 7: If not all the performers in P0 are moved to P1, go back to Step 4; otherwise go to Step 8.

Step 8: All the performers are able to complete their processes. That means that the approval of the request will lead to a safe state.

Step 9: There is a performer that is not able to get required resources to complete its process. That means that the approval of the request will lead to an unsafe state. The procedure will be ended at step 8 with the result of a safe state or at step 9 with the result of an unsafe state. According to the result of the system state, a request will be approved with the result of a safe state, since it is sufficient to avoid deadlocks. In case an unsafe state is derived, the situation is complicated. Although in some cases the request should be rejected, an unsafe state will not necessarily lead to deadlocks.

[0046] For a request that has not been proven to lead to a safe state, the requester itself might be irrelevant with the unsafe state. It means that the incompetence of accomplishing all the simulation tasks does not result from the tested performer and the request. In this case, rejection of the request will prevent the "innocent" performer from continuing its simulation task. The possibility of deadlocks will be increased as long as more and more such "innocent" performers are blocked. Therefore, it is necessary to identify the irrelevant requesters and approve the request of valid movements. After a deadlock with an unsafe result, a requester p can be sufficiently identified as an irrelevant requester if it satisfies all the following conditions:

- The requester p must be in the P1 list, in other words, all the required resources of p in MR are in AR or OR.
- The target resource of p is not requested by other performers that cannot accomplish their simulation tasks, i.e. the performers are not in P1, or the target resource of p is requested by a performer which is not in P1, but it is occupied by another performer. The latter situation indicates that the incompetence of allocate the target resource is not due to the tested request p, but the current occupier. Therefore, the requester p is still irrelevant to the result of unsafe state.

[0047] Identification of irrelevant requesters is a modification of the banker's algorithm providing an improvement.

[0048] In the following two examples are shown to demonstrate the procedure of the Banker's algorithm in different circumstances. The description of the detailed procedure is based on the flowchart shown in Fig. 1. Each step is also named with a step number as defined in Fig 1. There are several iterations from Step 4 to Step 7. As a convention, the number after a step number represents the iteration number.

Example 1: A Simple Network

[0049] The example 1 is based on the simple network shown in Fig. 2. The routes of the performers and the requested resources are listed in the following table:

Performer	Route	Requested Resource
Z1	G1, W1, G3, W2, G4, out	W1
Z2	G2, W1, G3, W2, G5, out	W1
Z3	G5, W2, G3, W1, G1, out	W2

[0050] Using the banker's algorithm, the system state test for the request of W1 from Z1 is shown in the following table:

Steps	Analysis for the Request of W1 from Z1
Step 1	P1 { }
Step 2	P0 {Z1, Z2, Z3}
Step 3	AR {G3, W2, G4}
Step 4-1	Take Z1 from P0 MR {G1, W1, G3, W2, G4}; OR {G1, W1}

EP 2 402 229 A1

(continued)

5

10

15

20

25

30

Steps	Analysis for the Request of W1 from Z1
	All resources in MR are in AR or OR, go to Step 6
Step 6-1	AR {W1, G3, W2, G4, G1} P1 {Z1}; P0 {Z2, Z3}
Step 7-1	P0 is not empty, go to Step 4.
Step 4-2	Take Z2 from P0 MR {G2, W1, G3, W2, G5} ; OR {G2} G5 is not in AR or OR, go to Step 5
Step 5-2	Z3 has not been tested, go back to Step 4
Step 4-3	Take Z3 from P0 MR {G5, W2, G3, W1, G1}; OR {G5} All resources in MR are in AR or OR, go to Step 6
Step 6-3	AR {W1, G3, W2, G4, G1, G5} P1 {Z1, Z3}; P0 {Z2}
Step 7-3	P0 is not empty, go to Step 4.
Step 4-4	Take Z2 from P0 MR {G2, W1, G3, W2, G5}; OR {G2} All resources in MR are in AR or OR, go to Step 6
Step 6-4	AR { W1, G3, W2, G4, G1, G5, G2} P1 {Z1, Z3, Z2}; P0 { }
Step 7-4	P0 is Empty, go to Step 8
Step 8	Safe State

[0051] The result shows that the request will lead a safe state.

[0052] The system state test for the request of W1 from Z2 is shown in the following table:

35

40

45

50

55

Steps	Analysis for the Request of W1 from Z2
Step 1	P1 { }
Step 2	P0 {Z1, Z2, Z3}
Step 3	AR {G3, W2, G4}
Step 4-1	Take Z1 from P0 MR {G1, W1, G3, W2, G4}; OR {G1} W1 is not in AR or OR, go to Step 5
Step 5-1	Z2 and Z3 have not been tested, go back to Step 4
Step 4-2	Take Z2 from P0 MR {G2, W1, G3, W2, G5}; OR {G2, W1} G5 is not in AR or OR, go to Step 5
Step 5-2	Z3 has not been tested, go back to Step 4
Step 4-3	Take Z3 from P0 MR {G5, W2, G3, W1, G1}; OR {G5} G1 is not in AR or OR, go to Step 5
Step 5-3	All performers are tested, go to Step 9
Step 9	Unsafe State

[0053] The system state test for the request of w2 from Z3 is shown in the following table:

Steps	Analysis for the Request of W2 from Z3
Step 1	P1 { }
Step 2	P0 {Z1, Z2, Z3}
Step 3	AR {G3, W1, G4}
Step 4-1	Take Z1 from P0 MR {G1, W1, G3, W2, G4}; OR {G1} W2 is not in AR or OR, go to Step 5
Step 5-1	Z2 and Z3 have not been tested, go back to Step 4
Step 4-2	Take Z2 from P0 MR {G2, W1, G3, W2, G5}; OR {G2} G5 and W2 are not in AR or OR, go to Step 5
Step 5-2	Z3 has not been tested, go back to Step 4
Step 4-3	Take Z3 from P0 MR {G5, W2, G3, W1, G1}; OR {G5, W2} G1 is not in AR or OR, go to Step 5
Step 5-3	All performers are tested, go to Step 9
Step 9	Unsafe State

[0054] The results show that these requests may lead to an unsafe state and cannot be approved.

[0055] It should be noted that AR and OR of each performer may vary in Step 3 for different request tests. This is due to the fact that a test is to check the system safe state for a request based on the assumption that the requested resource is allocated to the requester already. Although the requested resource has not been blocked yet, it will be put in OR of the requester instead of in AR.

Example 2: Using the banker's algorithm for macroscopic models

[0056] The banker's algorithm can also be used in a macroscopic model, in which a simulation is carried out without concerning the most detailed infrastructure information. A station section may be abstracted as a node in the network, and a track section links two nodes with each other. An example of such an abstraction is shown in Fig. 3.

[0057] All the operations in the network are bidirectional, with 3 abstracted nodes S1, S2 and S3 and 3 links L1, L2 and L3. The trains Z1 and Z3 are running from left to right and the trains Z2, Z4 and Z5 are running from right to left.

[0058] In a macroscopic simulation, a node or a link can be regarded as a resource. An important attribute of a node or link is the capacity. It defines the maximum number of performers that can be held in a resource simultaneously. For example, in the station S1, if there are two trains are allowed to stop or pass in the same time, the capacity of the station S1 is 2. The capacities of the resources are listed in the following table:

Resource	S1	S3	S3	L1	L2	L3
Capacity	2	2	2	1	1	1

[0059] The banker's algorithm supports a resource type with multiple instances. For executing an analysis of the system safe state for a request, the number of available instances is introduced as an attribute for AR. For instance, the available resource "S1(2)" means that the resource S1 is still available for two trains to stop or pass.

[0060] In the following table the procedure of state test for the request of L1 from train Z1 is demonstrated, with the result of an unsafe state:

Steps	Analysis for the Request of L1 from Z1
Step 1	P1 { }

EP 2 402 229 A1

(continued)

	Steps	Analysis for the Request of L1 from Z1
5	Step 2	P0 {Z1, Z2, Z3, Z4, Z5}
	Step 3	AR {S1(1), L2(1), S3(1)}
10	Step 4-1	Take Z1 from P0 MR {S1, L1, S2, L2, S3, L3}; OR {S1, L1} S2, L3 are not in AR or OR, go to Step 5
	Step 5-1	Z2, Z3, Z4 and Z5 have not been tested, go back to Step 4
15	Step 4-2	Take Z2 from P0 MR {S2, L1, S1}; OR {S2} L1 is not in AR or OR, go to Step 5
	Step 5-2	Z3, Z4 and Z5 have not been tested, go back to Step 4
20	Step 4-3	Take Z3 from P0 MR {S2, L2, S3, L3}; OR {S2} L3 are not in AR or OR, go to Step 5
	Step 5-3	Z4 and Z5 have not been tested, go back to Step 4
25	Step 4-4	Take Z4 from P0 MR {S3, L2, S2, L1, S1}; OR {S3} S2 and L1 are not in AR or OR, go to Step 5
	Step 5-4	Z5 has not been tested, go back to Step 4
30	Step 4-5	Take Z5 from P0 MR {L3, S3, L2, S2, L1, S1}; OR {L3} S2 and L1 are not in AR or OR, go to Step 5
	Step 5-5	All performers are tested, go to Step 9
	Step 9	Unsafe State

[0061] In the following table the procedure of state test for the request of L1 from train Z2 is demonstrated, with the result of a safe state:

	Steps	Analysis for the Request of L1 from Z2
40	Step 1	P1 { }
	Step 2	P0 {Z1, Z2, Z3, Z4, Z5}
	Step 3	AR {S1(1) , L2(1) , S3(1)}
45	Step 4-1	Take Z1 from P0
		MR {S1, L1, S2, L2, S3, L3}; OR {S1}
		L1, S2 and L3 are not in AR or OR, go to Step 5
	Step 5-1	Z2, Z3, Z4 and Z5 have not been tested, go back to Step 4
50	Step 4-2	Take Z2 from P0
		MR {S2, L1, S1}; OR {S2, L1}
		All resources in MR are in AR or OR, go to Step 6
55	Step 6-2	AR {S1(1), L2 (1) , S3 (1) , S2 (1) , L1 (1) }
		P1 {Z2}; P0 {Z3, Z4, Z5, Z1}
	Step 7-2	P0 is not empty, go to Step 4

EP 2 402 229 A1

(continued)

	Steps	Analysis for the Request of L1 from Z2
5	Step 4-3	Take Z3 from P0
		MR {S2, L2, S3, L3}; OR {S2}
		L3 not in AR or OR, go to Step 5
10	Step 5-3	Z4, Z5 and Z1 have not been tested, go back to Step 4
	Step 4-4	Take Z4 from P0
		MR {S3, L2, S2, L1, S1}; OR {S3}
		All resources in MR are in AR or OR, go to Step 6
15	Step 6-4	AR {S1(1), L2(1), S3(2), S2(1), L1(1)}
		P1 {Z2, Z4}; P0 {Z5, Z1, Z3}
	Step 7-4	P0 is not empty, go to Step 4
	Step 4-5	Take Z5 from P0
20		MR {L3, S3, L2, S2, L1, S1}; OR {L3}
		All resources in MR are in AR or OR, go to Step 6
	Step 6-5	AR {S1(1), L2 (1), S3 (2), S2 (1), L1(1), L3 (1)}
25		P1 {Z2, Z4, Z5}; P0 {Z1, Z3}
	Step 7-5	P0 is not empty, go to Step 4
	Step 4-6	Take Z1 from P0
		MR {S1, L1, S2, L2, S3, L3 }; OR {S1}
30		All resources in MR are in AR or OR, go to Step 6
	Step 6-6	AR {S1 (2), L2(1), S3(2), S2(1), L1(1), L3(1)}
		P1 {Z2, Z4, Z5, Z1}; P0 {Z3}
35	Step 7-6	P0 is not empty, go to Step 4
	Step 4-6	Take Z3 from P0
		MR {S2, L2, S3, L3 }; OR {S2}
		All resources in MR are in AR or OR, go to Step 6
40	Step 6-6	AR {S1(2), L2(1), S3(2), S2(2), L1(1), L3(1)}
		P1 {Z2, Z4, Z5, Z1, Z3}; P0 { }
	Step 7-6	P0 is empty, go to Step 8
45	Step 8	Safe State

[0062] Example 2 only demonstrates the basic principle for using the banker's algorithm for a macroscopic network. Since in most cases a microscopic simulation is more suitable than a macroscopic simulation for a dispatching system to operate single train movement, the following described embodiments are based on a microscopic simulation.

[0063] Based on the principle of deadlock avoidance, the banker's algorithm can guarantee a system against deadlocks when the system is in a safe state. A safe state will never lead to a deadlock situation, and a deadlock situation is always related to an unsafe state. However, not all unsafe states will lead to deadlocks. As long as an unsafe state is identified, the request of the resource will be rejected, and the performer should wait for next chances. If an unsafe state not sufficiently leads to deadlocks, such a stop will decrease the utilization level of infrastructure resources.

[0064] The main purpose of the inventive improvements of the banker's algorithm is to increase the space of safe states and reduce the possibility of false positives. Therefore, system efficiency and infrastructure utilization level will be improved with specific measures designed for railway operation.

[0065] Making a system state test to avoid deadlocks is a time-consuming work. Particularly for a dispatching system,

system performance is a decisive aspect for successfully putting the system into practice. Hence, according to the purposes, further suggested modifications can be classified into two categories: improvements of the banker's algorithm to increase the space of safe states and improvements of the system performance.

[0066] Measures to increase the safe state space also bring side effects to system performance, since additional processes, normally together with a new round of system state test, are introduced additionally. For these reasons, not all the suggested measures are required to be utilized in one implementation. An improvement will preferably be implemented only if the related problem of that improvement occurs frequently.

[0067] If an improvement of the banker's algorithm is introduced, the timing to apply the improvement preferably also is carefully designed. Although it is highly depending on the application context, the following principles are most commonly used:

- Longitudinal principle: for a certain request, if the waiting time due to deadlocks exceeds a pre-defined time threshold, an improvement of the banker's algorithm is applied.
- Transversal principle: in a single processing step, if all the deadlock-free tests have failed, at least one improvement of the banker's algorithm is applied.

[0068] Generally, the longitudinal principle is more practical than the transversal principle in a large network with not busy traffics, and the transversal principle is easier to be implemented than the longitudinal principle. It is also possible that both of them are integrated together. For example, if the transversal principle is applied in a single processing step, the performer with the highest waiting time due to deadlock will be given the highest priority to apply the improvements.

[0069] Unnecessary stops advantageously can be prevented when future potential state transitions are considered. A typical problem of the banker's algorithm is demonstrated in Fig. 4.

[0070] Train Z1 is associated with route R1 and train Z2 is associated with route R2. The routes and requested resources are given in the following table.

Performer	Route	Requested Resource
Z1	G1, W1, G2, W2, G4, out	W1
Z2	G4, W2, G3, W1, G1, out	W2

[0071] The procedure of the system state test for the request of W1 from Z1 is illustrated in the following table.

Steps	Analysis for the Request of W1 from Z1
Step 1	P1 { }
Step 2	P0 {Z1, Z2}
Step 3	AR {G2, G3, W2}
Step 4-1	Take Z1 from P0 MR {G1, W1, G2, W2, G4}; OR {G1, W1} G4 is not in AR or OR, go to Step 5
Step 5-1	Z2 has not been tested, go back to Step 4
Step 4-2	Take Z2 from P0 MR {G4, W2, G3, W1, G1}; OR {G4} W1 and G1 are not in AR or OR, go to Step 5
Step 5-2	All performers are tested, go to Step 9
Step 9	Unsafe State

[0072] The request will be rejected since it will lead to an unsafe state.

[0073] Similarly, the system state test for the request of W2 from Z2 will also result in an unsafe state. Therefore, both Z1 and Z2 are not allowed to continue their and the system is trapped in failure at this point. However, the system failed not necessarily into deadlocks after all the requests are approved as shown in Fig. 5. Both Z1 and Z2 can complete their movement tasks without any deadlock problems.

[0074] The system state test demonstrated in the table above concentrates only on the current situation, and future

potential state transitions - from an unsafe state to a safe state - are ignored. Unnecessary stops for "avoidable" deadlocks are introduced, and in some extreme cases as shown in Fig. 4, all performers are blocked with each other and the system is halted finally.

[0075] Hence, testing for potential state transitions will improve the performance of the banker's algorithm. A key point of the improvement is to identify a feasible track as the new location of the tested requester in future transited state. A feasible track for the tested requester is defined as:

- 1) The feasible track is in the route of the tested requester, and all the resources in the route from the current position of the tested requester to the feasible track are free to enter.
- 2) The feasible track is long enough to hold the tested requester.
- 3) Between the current position of the tested requester and the next junction type resource behind the feasible track, at least one alternative route exists when the feasible track is occupied by the tested requester, wherein junction type resource is searched in the route of the requester.

[0076] After moving from the current position to the feasible track, the tested requester will give a way for other performers to overtake or pass. The improvement with the consideration of future potential state transitions can be applied as follows:

[0077] When a request has been tested with an unsafe state, if a feasible track exists in MR, further state tests can be executed based on the assumption that the requester occupies the feasible track. The new round of tests are not only executed for the current deadlock test, but also for those tests failed in last round of tests, since the assumed state transition will create the chances for other performers to accomplish their trips. To examine if deadlocks are possibly to be avoided with potential state transitions, a new round of tests is started. The new tests are based on the assumption that the performer has moved to a new position, where the system state is transited. A feasible track that is long enough to hold the whole performer and give a way to other performers is a suitable new position, meanwhile all the resources in OR can be released to produce the possible state transition. An essential implication of the new round tests is to test the system safe state when the tested requester blocks the feasible track and releases all the resources from OR to the feasible track. In additional, it is necessary to ensure that the partial route from OR to the feasible track is able to enter (conflict-free).

[0078] If a test results in a safe state, the resources in the partial route preferably are reserved as a whole for the tested requester. Additional alternative route searching should be executed for the performers that are supposed to enter the feasible track. Since the feasible track will be blocked by the tested requester, other performers are not able to enter the feasible track. An alternative route that excludes the feasible track is required.

[0079] For the example in Fig. 4, after an unsafe state is concluded with the test for the request of W1 from Z1, the assumed new situation will be changed as given in the following table:

Performer	Route	Requested Resource
Z1	G2, W2, G4, out	W2
Z2	G4, W2, G3, W1, G1, out	W2

[0080] The requested resource for train Z1 has been changed to W2, based on the assumption that Z1 passes G1, W1 and occupies G2. The new test is to analyze the situation if G2 is blocked by Z1. In this situation, G1 and W1 will be regarded as available resource, and G2 will be put into the OR of Z1. The whole procedure is illustrated in the following table.

Steps	Analysis for the Request of W2 from Z2
Step 1	P1 { }
Step 2	P0 {Z1, Z2}
Step 3	AR {G1, W1, G3 }
Step 4-1	Take Z1 from P0 MR {G2, W2, G4}; OR {G2} W2 and G4 are not in AR or OR, go to Step 5
Step 5-1	Z2 has not been tested, go back to Step 4
Step 4-2	Take Z2 from P0

(continued)

Steps	Analysis for the Request of W2 from Z2
	MR {G4, W2, G3, W1, G1}; OR {G4, W2} All resources in MR are in AR or OR, go to Step 6
Step 6-2	AR { G1, W1, G3, W2, G4} P1 {Z2}; P0 {Z1}
Step 7-2	P0 is not empty, go to Step 4.
Step 4-3	Take Z1 from P0 MR {G2, W2, G4}; OR {G2} All resources in MR are in AR or OR, go to Step 6
Step 6-3	AR { G1, W1, G3, W2, G4, G2} P1 {Z2, Z1}; P0 { }
Step 7-3	P0 is Empty, go to Step 8
Step 8	Safe State

[0081] With the result of a safe state, the analysis shows that the further allocation of W2 to Z2 with the potential state transition is feasible. Although the test is only executed for the request of W2 by Z2, the whole route W1 and G2 preferably is reserved for Z1. An analysis can also be carried out for Z1 requesting for W2. The result will show that Z1 should not be allowed to enter W2, and Z1 should wait at G2 until Z2 passes. By testing with the potential state transition, the deadlock problem does not exist anymore as shown in Fig. 5.

[0082] However, it is not always recommended to extend new round tests too far, because otherwise too many resources might be involved too early to keep the route from the current position to the feasible track free to enter, which will influence the requests of the involved resources from other performers. It should be noted that the analysis with potential state transition requires additional processing time. In certain cases, it might be more efficient to give up the failed, current round of tests, and to examine potential state transitions for other performers.

[0083] A state test may unnecessarily lead to an unsafe state when an inappropriate order of processes is used in the test. An improvement is proposed, wherein the space of safe states is increased by executing the state test for processes in a right order.

[0084] In the examples described so far all the trains will leave the network finally. This is however not always the case. In some situations the destination of a movement task is still inside the observed infrastructure network, and such a destination is called an internal destination.

[0085] Possible examples are:

- A train is operated locally, and the whole train path is inside the observed network.
- A train comes from the outside of the network, and the destination of the train is a depot that is inside the observed network.
- A shunting set is operated in the observed shunting yard, and the shunting set is supposed to reside in the shunting yard until the whole simulation is finished.

[0086] Special attention should be given in system state tests when performers with an internal destination exist. The order of testing performers in P0 will influence the result of a state test. In the following the example shown in Fig. 6 is considered. Trains Z1, Z2, Z3 and Z4 are respectively associated with routes R1, R2, R3 and R4. The routes and requested resources are given in the following table.

Performer	Route	Requested Resource
Z1	G6, W3, G4, W2, G3	W3
Z2	G5, W3, G4, W2, G3, W1, G1, out	W3
Z3	G7, W4, G5, W3, G4, W2, G3, W1, G1, out	W4
Z4	G1, W1, G2, W2, G4, W3, G5, W4, G7, out	W1

[0087] For the request of W1 from Z4, it can be proven that it will lead to an unsafe state. When applying the improvement as described above referring to Figures 4 and 5, the request of G2 from Z4 will be considered as shown in Fig. 7 with the routes and requested resources as given in the following table.

Performer	Route	Requested Resource
Z1	G6, W3, G4, W2, G3	W3
Z2	G5, W3, G4, W2, G3, W1, G1, out	W3
Z3	G7, W4, G5, W3, G4, W2, G3, W1, G1, out	W4
Z4	G2, W2, G4, W3, G5, W4, G7, out	G2

[0088] In the new round of system state test for the request of G2 from Z4, the original P0 includes all the performers Z1, Z2, Z3 and Z4. A performer will be taken from P0 to test the possibility of resource allocation. Assume that Z1 is chosen as the first tested performer. All the required resources in MR ({G6, W3, G4, W2, G3}) are in AR ({W3, G4, W2, G3, W4}) or OR ({G6}). Z1 will be moved from P0 to P1 and its current blocked resources will be returned to AR.

[0089] Since the route of Z1 is supposed to be ended at an internal destination G3, the G3 will be taken out from AR. Continuing the deadlock-free test, all the performers in P0, i.e. Z2, Z3 and Z4, are not able to complete their simulation tasks. The state test leads to an unsafe state and the request of G2 from Z4 will be rejected.

[0090] However, if assumed that Z2 instead of Z1 is chosen as the first tested performer, a safe state will be concluded from the test. The sequences of the train movements can be arranged as Z2, Z3, Z4 and Z1 as shown in Fig. 8.

[0091] Starting to test a simulation task with an internal destination will decrease the possibility to find a right arrangement for all performers. If such a simulation task can be completed in the beginning, the destination resource will be removed from AR and reserved for the performer of that simulation task permanently. This reserved resource may also be required by other performers, and these performers will never be able to complete their simulation tasks eventually. Therefore, testing simulation tasks in an inappropriate order may result in an unsafe state unnecessarily and reduce the space of safe state. An improvement concerning on the order of testing performers is therefore preferably employed, wherein testing the performers in P0 for resource allocation is arranged in the following order:

- 1) At first all the performers supposed to leave the observed network should be tested until no such a performer can be moved from P0 to P1.
- 2) Then the performers with an internal destination lying at a dead-end track will be tested. If no such a performer can be moved from P0 to P1, go to 3); otherwise go back to 1).
- 3) The rest of the performers will be tested at last.

[0092] If a performer with an internal destination lying at a dead-end track, it will influence other performance very rare. For this reason, these performers are treated as a special group with a higher priority compared with other performers supposed to be ended in the observed network. This is particularly useful in some situations when several trains are operated inside the observed network.

[0093] Another improvement is proposed for the case that a state test fails with the original scheduled route, while a feasible solution exists when an alternative route is used. This improvement increases the space of safe state with alternative routes.

[0094] One of the most important preconditions of the banker's algorithm is to know the required resources of each process in advance. In railway synchronous simulation, it means the route information of each train typically is known before a system state test starts. Using fixed routes for all performers would be the easiest implementation. However, such an implementation is not flexible enough for optimal deadlock avoidance.

[0095] In Fig. 9 an example with predefined fixed routes is shown, wherein an unsafe state will be concluded in the system state test. Trains Z1, Z2, Z3 and Z4 are respectively associated with routes R1, R2, R3 and R4. The routes and requested resources are given in the following table.

Performer	Route	Requested Resource
Z1	G1, W1, G3, W2, W3, G4, W4, out	W1
Z2	G2, W1, G3, W2, G5, out	W1
Z3	G5, W2, G3, W1, G1, out	W2
Z4	G4, W3, W2, G3, W1, G2, out	W3

- [0096]** The steps for performing the system state test are similar to those described above with reference to Fig. 2.
- [0097]** Although all train movements are not allowed due to deadlocks, the deadlock situation can be avoided if Z1 changes its route to use G6 instead of G4 as shown in Fig. 9, wherein the alternative route is indicated by a dashed line.
- [0098]** The changed routes and requested resources are given in the following table.

Performer	Route	Requested Resource
Z1	G1, W1, G3, W2, W3, G6, W4, out	W1
Z2	G2, W1, G3, W2, G5, out	W1
Z3	G5, W2, G3, W1, G1, out	W2
Z4	G4, W3, W2, G3, W1, G2, out	W3

[0099] Therefore, a further important improvement proposed by the invention is to take alternative routes into consideration if a train with alternative routes is in a deadlock situation or causes an unsafe state to be determined.

[0100] Train routes preferably are determined by certain route searching algorithms. In a train schedule the station-relevant destinations are the scheduled destinations that should be respected in a fixed sequence during route searching processes. Any modification of scheduled destinations should be carefully evaluated during the dispatching and optimization process. The rest of the destinations in the original train route are the relative destinations. The original relative destinations may be excluded when an alternative route is utilized from a scheduled destination to the next scheduled destination. In the procedure of searching alternative routes, the determination of scheduled destinations starts at first, and alternative routes including a series of relative destinations between every two adjacent, scheduled destinations are searched afterwards.

[0101] A route searching procedure with the consideration of scheduled/relative destinations is for instance described in "Verfahren zur Bewertung von Zug- und Rangierfahrten bei der Disposition" by U. Martin, Technische Universität Braunschweig, Dissertation, 1995 or in "Beitrag zur Entwicklung eines Dispositionswerkzeugs zur Optimierung betriebsbedingter Wartezeiten im schienengebundenen Verkehr" by J. Schlaich, Universität Stuttgart, Studienarbeit, 2002.

[0102] If all the possible routes of a performer were available before a simulation task is in "running" state, it would be easy to improve the banker's algorithm with alternative routes. The route with the shortest running time can be selected as the alternative route. In many cases it is however not practical to get all the possible routes for all performers.

[0103] According to the predefined absolute destinations, a route with the shortest running time can be searched by some kind of route searching algorithm, for instance using Dijkstra's algorithm as described in „A Note on Two Problems in Connexion with Graphs" by E.W. Dijkstra, Numerische Mathematik 1, 1959, No. 1, pages 269-271.

[0104] When an alternative route is required, the route with the secondary shortest running time can also be searched with Dijkstra's algorithm in such a way: each time an edge in the shortest route is removed from the whole network, and the optimum solution to this network is searched. Repeating the operation for all the edges, a ranked list of less-than-optimal solutions is generated. The route with the secondary shortest running time can be obtained from the list. This approach requires many times of route searching, and possibly deadlocks still occur when applying the route with the second shortest running time.

[0105] A more practical solution can be utilized based on the principle of the banker's algorithm. The improvement with alternative routes is applied only if the request cannot pass the first round deadlock-free test with original route. Considering the steps of the banker's algorithm described above with reference to Fig. 1, the test must be ended at the Step 9 when all the performers in P0 are not able to get all required resources in MR. For these performers, there must be at least one resource that is in MR but not in AR or OR. This resource is removed from the graph, and the optimum solution is searched within the new collection of infrastructure resources. If such a solution exists, the route can be used as the alternative route. In this way, it is not necessary to execute route searching often, and the resource that lead to a problem in a previous test is excluded from the route. The process of the improvement with alternative routes is illustrated in Fig. 10.

[0106] A system state test for a request is unnecessarily executed when the result of test does not influence the system state. A further preferred improvement lies in skipping those unnecessary state tests, thereby improving the system performance. Deadlock-free tests are carried out when allocating new requested resources for performers. When the banker's algorithm is applied, the test will go through all the processes to determine the system state. In some cases such a time consuming job can be skipped to gain a higher system performance.

[0107] As discussed above, infrastructure resources can be defined based on block sections or infrastructure elements, and there are two types of infrastructure resources: junction type resources and non-junction type resources. A deadlock-free test, i.e. a system state test, preferably is not carried out for a request of entering a non-junction type resource.

[0108] In a deadlock-free test, the required resources for a requester are established based on its route. From the

point of view of the relation among routes, entering a non-junction type resource does not influence the resource allocation for other routes. The approval of the request of a non-junction type resource will not change system deadlock status.

[0109] By the inventive method simulating, planning and controlling operating processes and dispatching in a track guided transportation system is greatly improved, wherein unnecessary stops are significantly reduced and system performance is greatly improved also for complex operating conditions which are typically encountered in shunting operation or in a mixed operation of shunting and train runs. With special advantage the number of solutions which are actually valid, but which are nevertheless - falsely - classified as deadlocks is significantly reduced by the invention, so that the practical benefit is highly improved.

Claims

1. A method for simulating, planning and/or controlling operating processes of a track guided transportation system, wherein

- resources are defined which represent track segments of the transportation system,
- performers are defined which represent transportation vehicles moving into, within and/or out of the transportation system, wherein for each performer at least one associated route with a starting location and a destination location is defined,
- at least one request from a performer is provided requesting to occupy at least one resource,
- for each of said at least one request a corresponding system state is determined as safe or unsafe, and
- for deadlock avoidance only requests are granted for which a safe state is determined.

2. The method of claim 1, further comprising the step of at least partially automatic dispatching in a track guided transportation system.

3. The method of claim 1 or 2, wherein the system state is determined utilizing the banker's algorithm.

4. The method of any one of the preceding claims, wherein determining the system state comprises

- a) determining a set of moving performers (P_0), comprising all performers which currently move along their associated routes,
- b) determining for each performer in the set of moving performers the respective occupied resources (OR), wherein said occupied resources (OR) comprise any resources requested to be occupied by the respective performer,
- c) determining as available resources (AR) all resources not occupied by a performer,
- d) initializing set (P_1) of feasible performers as an empty set,
- e) selecting a performer (p) from the set of moving performers (P_0) based on a pre-defined order,
- f) determining for the selected performer (p) the resources (MR) required to perform a movement based on its associated route,
- g) checking for the selected performer (p) whether the required resources (MR) are either available resources (AR) or resources (OR) presently occupied by the selected performer (p), wherein if the check is positive, the selected performer (p) is moved from the set of moving performers (P_0) to the set of feasible performers (P_1) and the available resources (AR) are updated based on the premise that the selected performer (p) performs the movement based on its associated route,
- h) repeating steps e) to g) until either all moving performers (P_0) have been moved to the set of feasible performers (P_1) or until no further moving performer (P_0) can be moved to the set of feasible performers (P_1),
- i) if all moving performers (P_0) have been moved to the set of feasible performers (P_1) a safe system state is determined, otherwise an unsafe system state is determined.

5. The method of claim 4, wherein if an unsafe system state is determined, the request of a moving performer which has been moved to the set of feasible performers (P_1) is granted, if

- no resource of the destination location of its associated route is requested by any moving performer which has not been moved to the set of feasible performers (P_1), or
- a resource of the destination location of its associated route is requested by a moving performer which has not been moved to the set of feasible performers (P_1), but it is occupied by another performer.

6. The method of any one of the preceding claims, wherein the resources at least comprise junction-type and non-junction type resources, and wherein if an unsafe system state is determined and for a requesting performer a feasible track is identifiable, the system state is determined again and granting and/or denying requests is performed in response to the newly determined system state, wherein
 - the feasible track lies within the route associated with the requesting performer and all resources in the route between the currently occupied resources of the requesting performer and the feasible track are available resources (AR),
 - the feasible track comprises sufficient resources to hold the requesting performer, and
 - between the currently occupied resources of the requesting performer and the next junction-type resource behind the feasible track at least one alternative route exists, when the feasible track is occupied by the requesting performer.
7. The method of any one of claims 4 to 6, wherein the order in which a performer is selected from the set of moving performers (P_0) in step e) is defined such that
 - first all performers are selected, for which the destination location of their respective associated route lies outside the defined resources, and
 - second all performers are selected, for which the destination location of their respective associated route comprises resources which represent a dead-end track segment, and
 - third all remaining performers are selected.
8. The method of any one of claims 4 to 7, wherein at least one performer can be associated with at least two routes, and wherein if an unsafe system state is determined, a moving performer which has not been moved to the set of feasible performers (P_1) is associated with an alternative route and the system state is determined again and granting and/or denying requests is performed in response to the newly determined system state.
9. The method of claim 8, wherein routes are determined by a route searching algorithm applied to a graph which represents the defined resources and their interconnections, and wherein the alternative route is determined by applying said route searching algorithm to a modified graph from which at least one resource within the original route of the moving performer which is neither an available resource (AR) or a resource (OR) presently occupied by the moving performer is removed.
10. The method of any one of the preceding claims, wherein the resources at least comprise junction-type and non-junction type resources, and wherein a system state is determined only depending on requests requesting to occupy at least one resource which is not a non-junction type resource.
11. The method of any one of the preceding claims, wherein a maximum waiting time is defined and if a request has not been granted after said maximum waiting time, the steps for determining the system state are altered.
12. The method of any one of the preceding claims, wherein at least one performer simultaneously occupies more than one resource.
13. The method of any one of the preceding claims, wherein during the simulation a performer is divided into at least two separate performers and/or at least two performers are combined into one common performer.
14. A system for simulating, planning and/or controlling operating processes of a track guided transportation system adapted to perform a method according to any one of claims 1 to 13.
15. Digital storage medium, comprising electronically readable control instructions for simulating, planning and/or controlling operating processes of a track guided transportation system, adapted to perform, when inserted into a computer system, a method according any one of claims 1 to 13.

Fig. 1

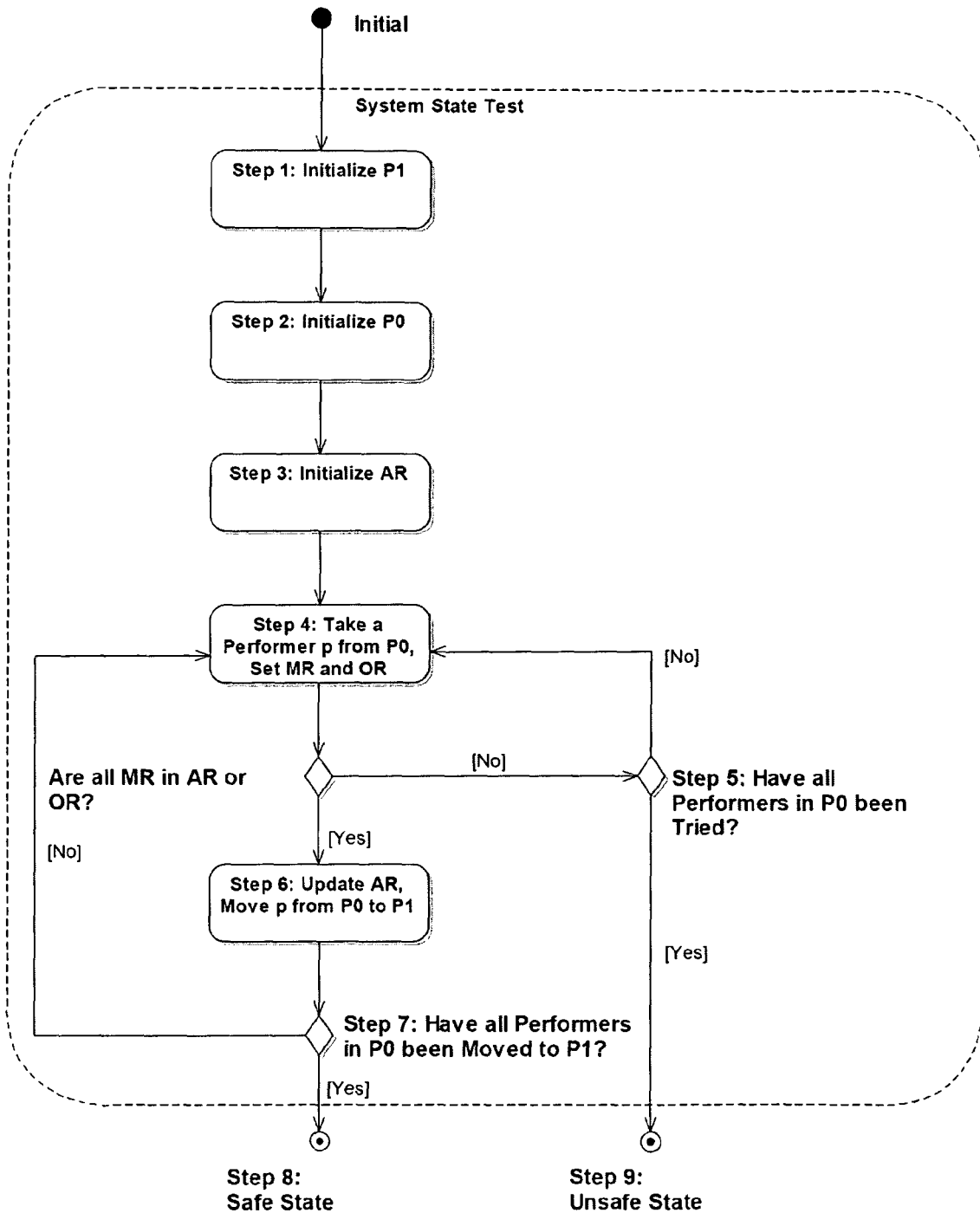


Fig. 2

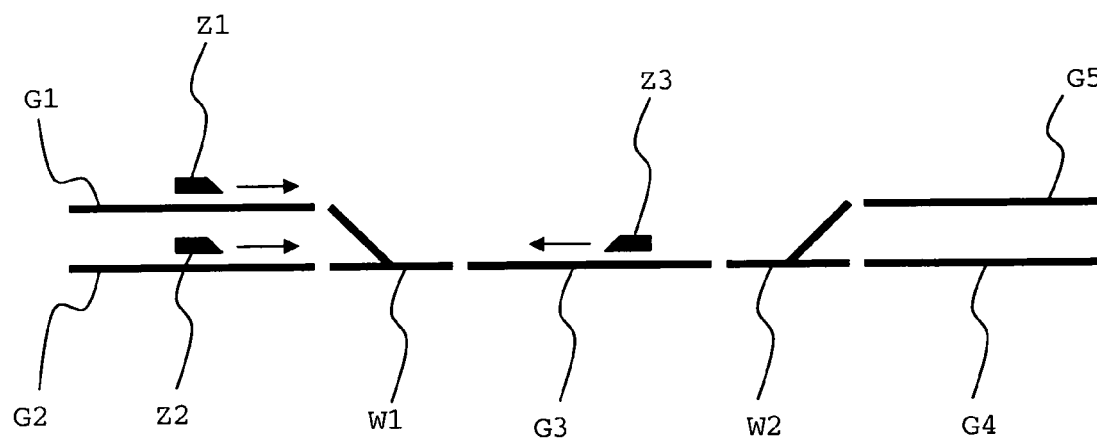
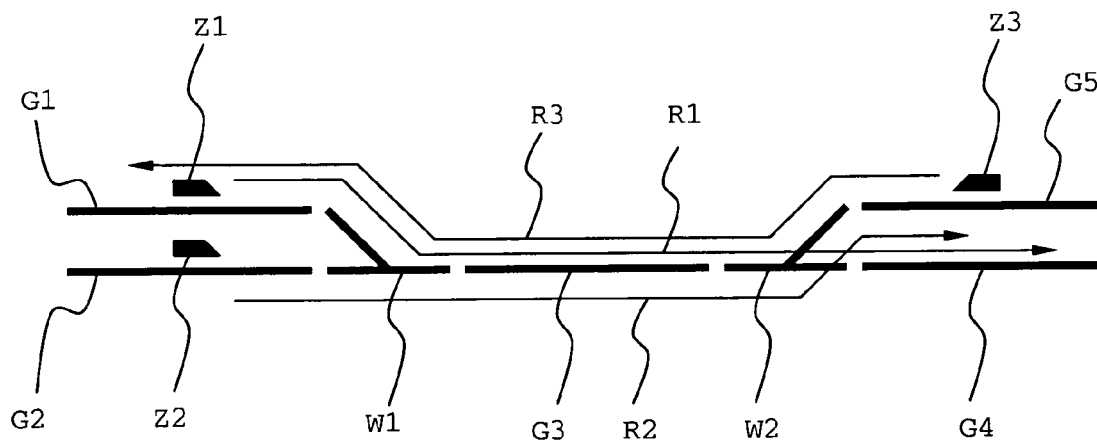


Fig. 3

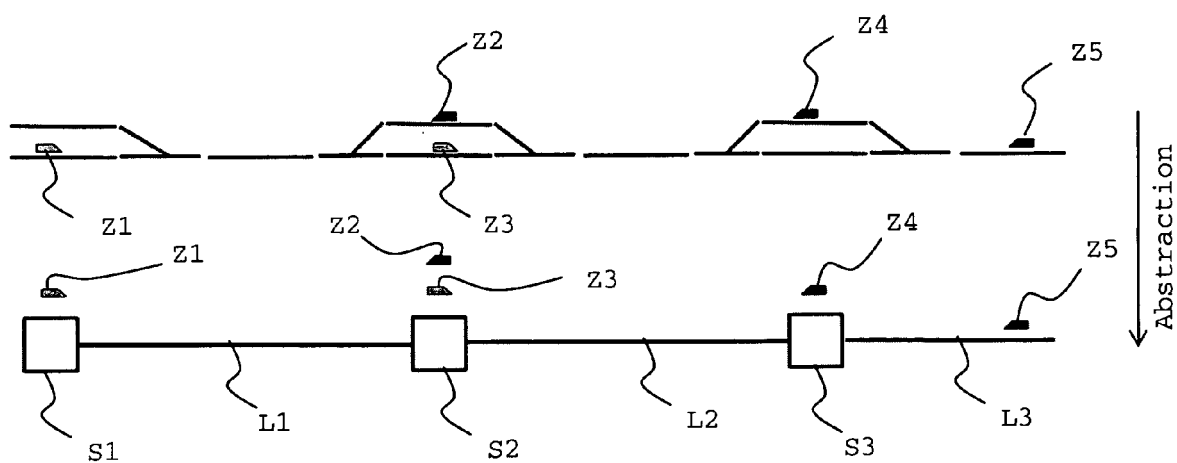


Fig. 4

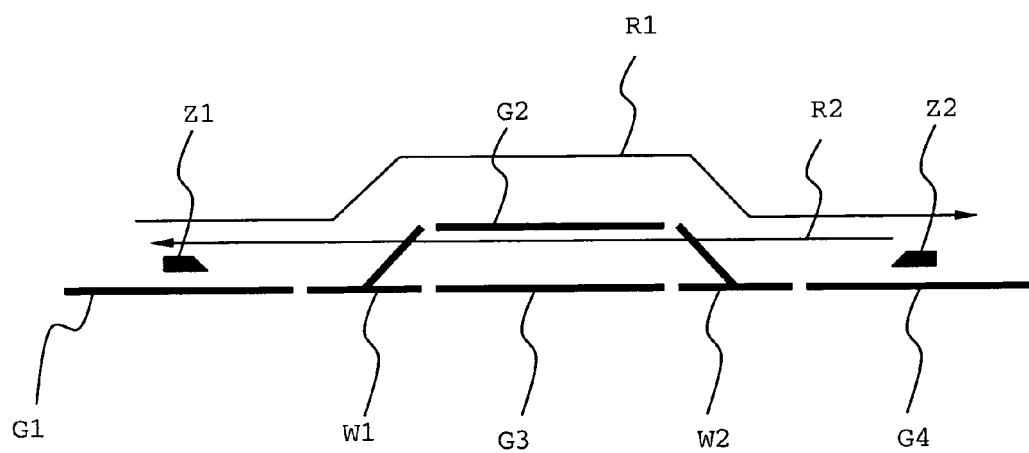


Fig. 5

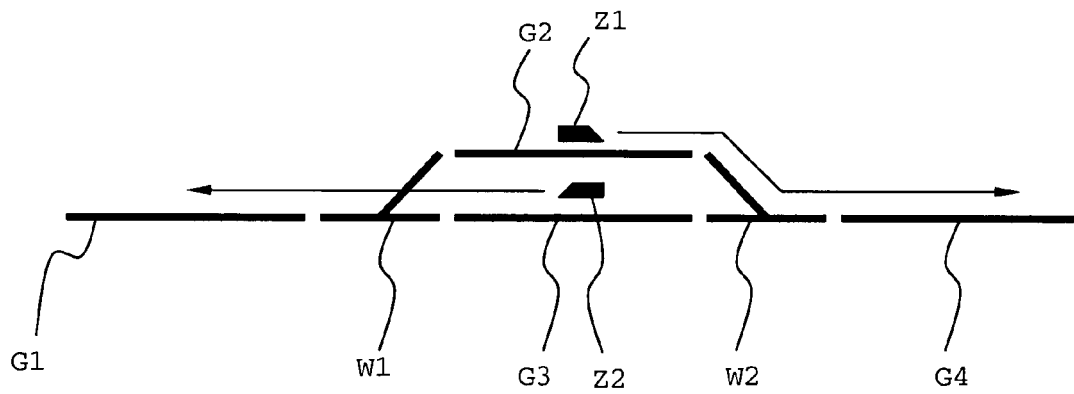


Fig. 6

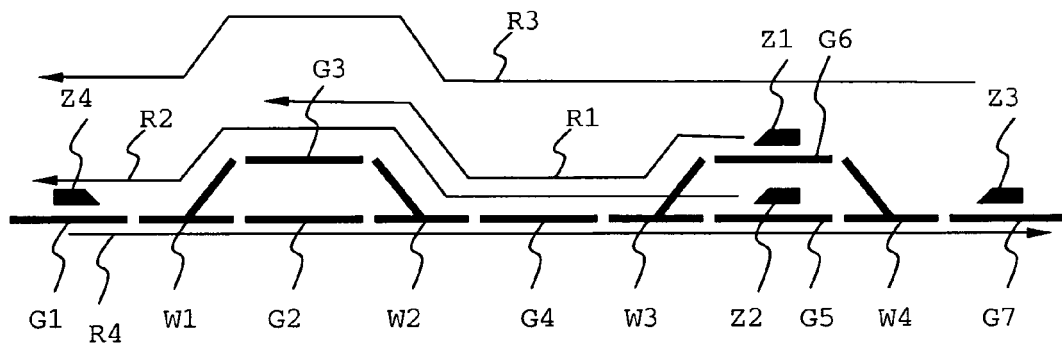


Fig. 7

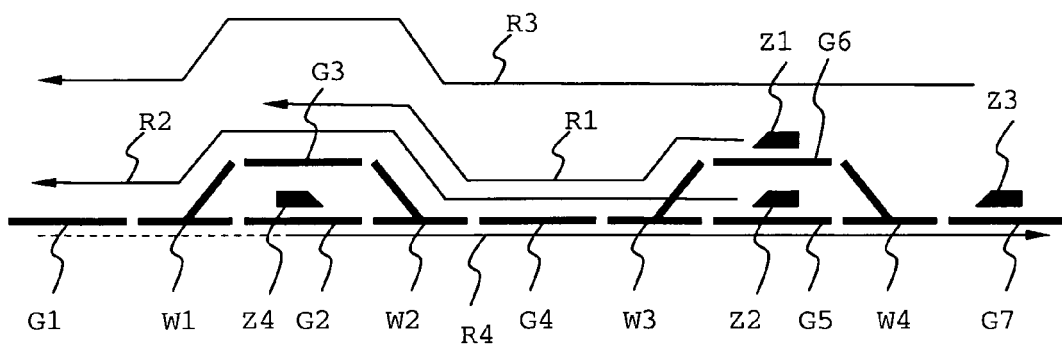


Fig. 8

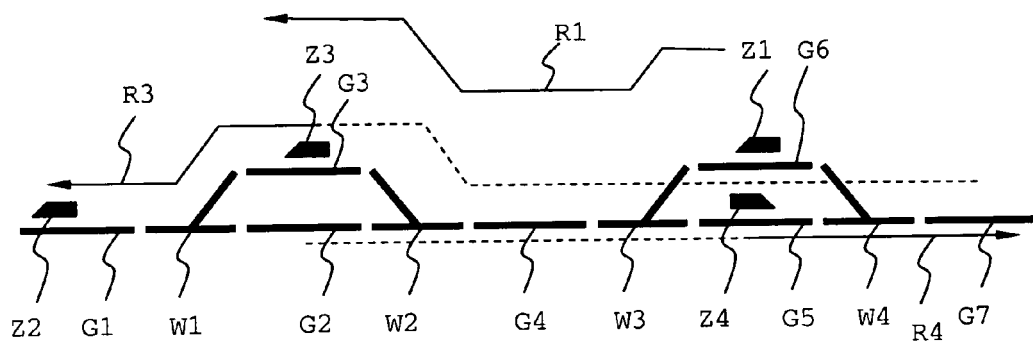


Fig. 9

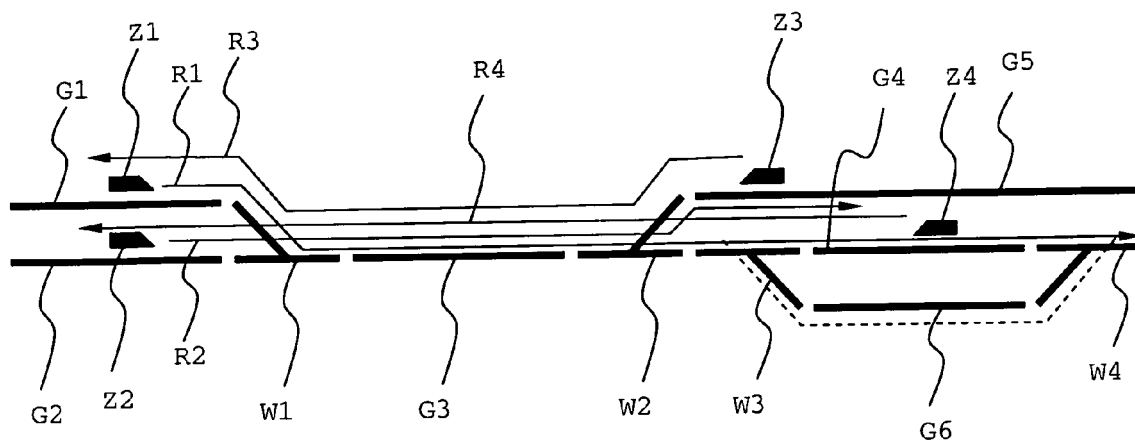
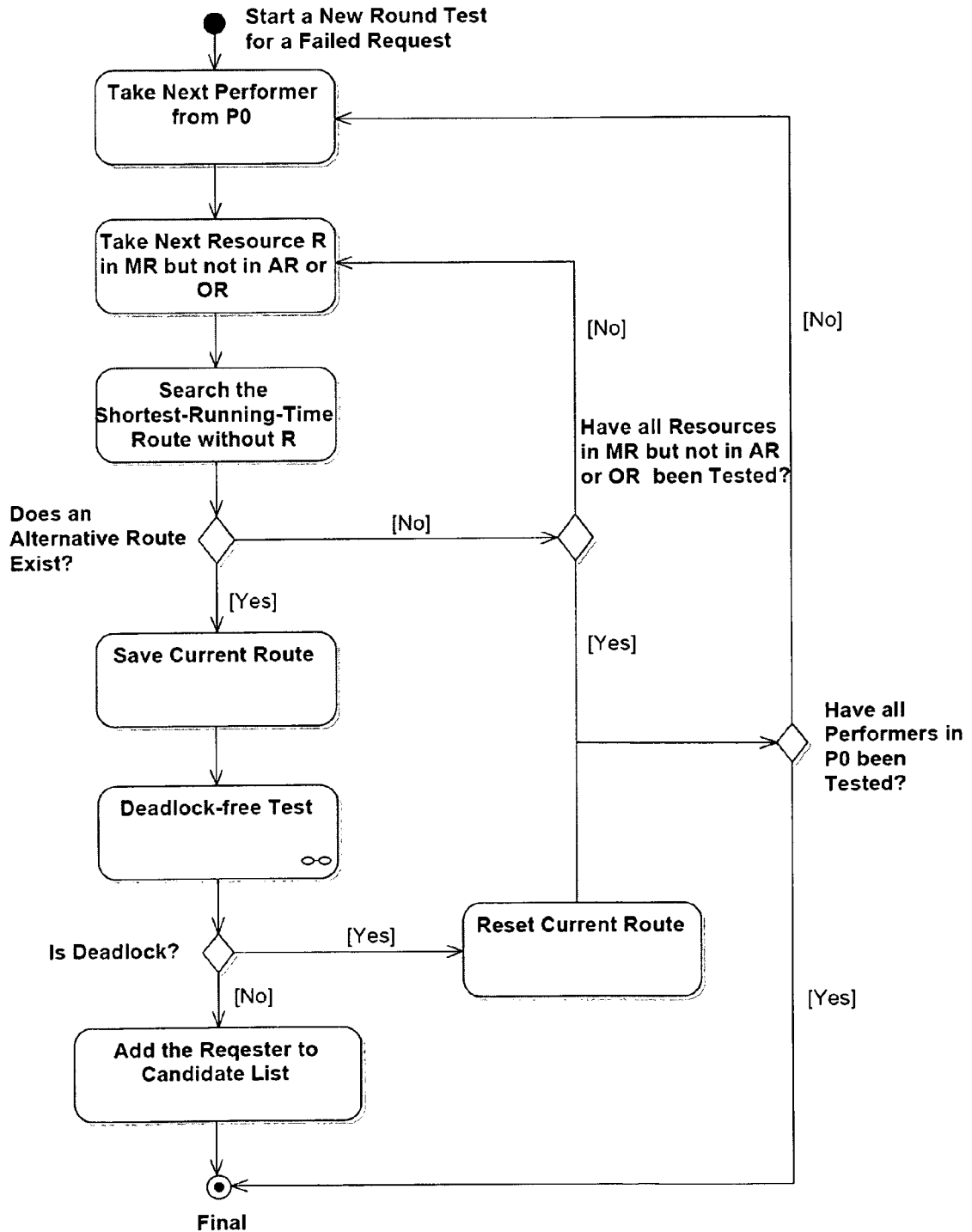


Fig. 10





EUROPEAN SEARCH REPORT

Application Number
EP 10 00 6446

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	BOBANAC V ET AL: "Routing and scheduling in Multi-AGV systems based on dynamic banker algorithm", CONTROL AND AUTOMATION, 2008 16TH MEDITERRANEAN CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 25 June 2008 (2008-06-25), pages 1168-1173, XP031308505, ISBN: 978-1-4244-2504-4 * the whole document *	1-3,14,15	INV. B61L27/00
A	WO 03/097425 A1 (TMG INTERNAT HOLDINGS PTY LTD [AU]; PUDNEY PETER JOHN [AU]; HOWLETT PH) 27 November 2003 (2003-11-27) * page 4, line 23 - page 5, line 11 * * page 7, line 24 - page 11, line 2 *	4,5,7-9	
A	US 5 828 979 A (POLIVKA ALAN L [US] ET AL) 27 October 1998 (1998-10-27) * column 3, line 24 - column 3, line 51 * * column 5, line 16 - column 6, line 35 *	4,5,7-9	
A	EP 0 911 778 A2 (TOYOTA MOTOR CO LTD [JP]) 28 April 1999 (1999-04-28) * paragraph [0037] - paragraph [0044] *	6,10	TECHNICAL FIELDS SEARCHED (IPC) B61L
A	US 4 883 245 A (ERICKSON JR THOMAS F [US]) 28 November 1989 (1989-11-28) * claim 1; figures 5,24 *	6,10	
3 ~~The present search report has been drawn up for all claims~~			
Place of search Munich		Date of completion of the search 8 February 2011	Examiner Janhsen, Axel
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 1503 03.82 (P04C01)



Application Number

EP 10 00 6446

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing claims for which payment was due.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for those claims for which no payment was due and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for those claims for which no payment was due.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☐ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

☒ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

1-10, 14, 15

☐ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

☐ The present supplementary European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims (Rule 164 (1) EPC).



**LACK OF UNITY OF INVENTION
SHEET B**

Application Number

EP 10 00 6446

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. claims: 1-3, 14, 15

automatic dispatching in a track guided transportation
system utilizing the banker's algorithm

2. claims: 1, 4, 5, 7-9

determination of a safe and unsafe system state using
feasible performers

3. claims: 1, 6, 10

determination of a safe and unsafe system state using
feasible track resources with junction-type resources

4. claims: 1, 11

consideration of waiting times

5. claims: 1, 12

consideration of double occupations

6. claims: 1, 13

splitting and joining of performers

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 10 00 6446

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

08-02-2011

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 03097425	A1	27-11-2003	CA 2526152 A1	27-11-2003
			GB 2405017 A	16-02-2005
			US 2005261946 A1	24-11-2005

US 5828979	A	27-10-1998	AT 343503 T	15-11-2006
			AU 712538 B2	11-11-1999
			AU 3374695 A	22-03-1996
			BR 9509462 A	21-10-1997
			CA 2198855 A1	07-03-1996
			CN 1162290 A	15-10-1997
			DE 69535279 T2	24-05-2007
			EP 0782521 A1	09-07-1997
			JP 10505036 T	19-05-1998
			PL 319030 A1	21-07-1997
			WO 9606766 A1	07-03-1996
			US 5623413 A	22-04-1997
			US 5794172 A	11-08-1998
			ZA 9507360 A	28-08-1996

EP 0911778	A2	28-04-1999	DE 69821437 D1	11-03-2004
			DE 69821437 T2	16-12-2004
			JP 3653954 B2	02-06-2005
			JP 11126294 A	11-05-1999
			US 6169495 B1	02-01-2001

US 4883245	A	28-11-1989	NONE	

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20060212188 A1 [0005]
- US 7428452 B2 [0006]

Non-patent literature cited in the description

- **J. PACHL.** *Steuerlogik für Zuglenkanlagen zum Einsatz unter stochastischen Betriebsbedingungen*, 1993 [0003]
- **J. PACHL.** Avoiding Deadlocks in Synchronous Railway Simulations. *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*, 28 March 2007, <http://www.digi-bib.tu-bs.de/?docid=00020794> [0003]
- **E.R. PETERSEN ; A.J. TAYLER.** Line Block Prevention in Rail Line Dispatch. *INFOR Journal*, 1983, vol. 21 (1), 46-51 [0003]
- **G. MILLS ; P. PUDNEY.** The effects of deadlock avoidance on rail network capacity and performance. *Proceedings of the 2003 Mathematics-in-Industry Study Group*, 2008, 49-63 [0003]
- **E.W. DIJKSTRA.** The mathematics behind the banker's algorithm. Springer, 1982, 308-312 [0012]
- **U. MARTIN.** Verfahren zur Bewertung von Zug- und Rangierfahrten bei der Disposition. *Dissertation*, 1995 [0101]
- **J. SCHLAICH.** Beitrag zur Entwicklung eines Dispositionswerkzeugs zur Optimierung betriebsbedingter Wartezeiten im schienengebundenen Verkehr. *Studienarbeit*, 2002 [0101]
- **E.W. DIJKSTRA.** A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1959, vol. 1 (1), 269-271 [0103]