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

(11) EP 3 395 643 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
31.10.2018 Bulletin 2018/44

(51) Int Cl.:
B61L 19/06 *(2006.01)* ***B61L 21/04*** *(2006.01)*
B61L 27/00 *(2006.01)* ***G06F 9/00*** *(2006.01)*

(21) Application number: **17305477.6**

(22) Date of filing: **28.04.2017**

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

(71) Applicant: **ALSTOM Transport Technologies**
93400 Saint-Ouen (FR)

(72) Inventor: **MINKOWITZ, Cydney**
40125 BOLOGNA (IT)

(74) Representative: **Lavoix**
2, place d'Estienne d'Orves
75441 Paris Cedex 09 (FR)

(54) METHOD FOR CHECKING SAFETY REQUIREMENTS OF SSI-BASED DATA USED IN AN INTERLOCKING CONTROL SYSTEM

(57) Method for checking safety requirements of SSI-based data used in an interlocking control system for controlling an interlocking equipment, the method comprising the steps of:
a) obtaining (2) application data representative of interlocking logic operations of the interlocking equipment;
b) preparing (4) a constraint violation file containing data representative of a plurality of constraint violation conditions, said data describing a plurality of unsafe scenarios of the interlocking equipment;
for each constraint violation condition of the plurality of constraint violation conditions:
c) selecting (6) data of the application data according to the constraint violation condition, said selected data corresponding to a predetermined unsafe scenario of the plurality of unsafe scenarios defined in the constraint violation file;
d) determining (7) at least one predetermined context associated to said unsafe scenario, said context comprising a plurality of paths through the application data;
e) initializing (8) variables that define all possible states of said scenario, thus obtaining a predetermined initial state, said variable being representative of the scenario from the point of view of settings of the interlocking equipment;
f) executing (10), starting from said initial state, all possible paths of the context in the application data, thus obtaining respective resulting states;
g) at an end of each path, determining (12) if an unsafe state has been reached by comparing a respective resulting state with the data of the constraint violation file;
h) if no unsafe state has been detected, determining (12) if the resulting state has not been reached;
-i) repeating steps f), g) and h), starting, for each path, from the respective resulting state, until unsafe states or no new states are reached.

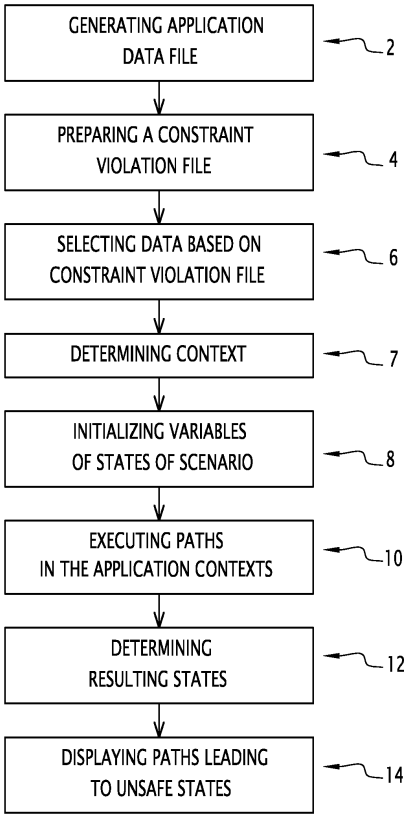

FIG.1

EP 3 395 643 A1

Description

[0001] The present invention concerns a method for checking safety requirements of SSI-based data (for Solid State Interlocking based date) used in an interlocking control system.

[0002] An interlocking control system is a computer based railroad signaling system whose purpose is to ensure the safe movements of railroad vehicles, in particular in order to prevent collisions and derailments.

[0003] The present invention relates therefore to the domain of circulation management of vehicles using a transportation network, such as a railroad transportation network, a subway network, a tramway network, or the like. In a manner known per se, the circulation management is achieved through interlocking equipment of an interlocking system, often called "yard equipment" and/or "wayside equipment" and/or "trackside equipment", which are designed to perform specific train circulation-related operations. This equipment may comprise signaling devices and/or railroad switches and/or track circuits, or the like.

[0004] This field equipment may be controlled remotely by operators through an interlocking control system, in a semi-automatic or automatic manner, based on a software architecture executed by computing means of the control system. The interlocking control system includes a communication network through which the system is interfaced with the field equipment.

[0005] An interlocking control system is usually designed through known methods which design both the hardware and the software components of the system, as well as the communication network used for interfacing the transportation network wayside equipment with the developed hardware.

[0006] The interlocking control system comprises hardware, including one or more control units, for controlling the interlocking equipment, each control unit comprising computing means intended for controlling the interlocking equipment remotely through the communication network.

[0007] Practically, the control units may be station-based, the communication network being configured to link the units to the interlocking equipment arranged trackside. For this purpose, the communication network usually comprises a network of cables connected to each element of the interlocking equipment and to one or several of the control units.

[0008] For controlling the interlocking equipment, the computing means of the control units execute software, in other words a system of one or more computer programs.

[0009] In particular, the interlocking control system comprises a plurality of interlocking applications, and the functioning of each interlocking application, also called the logic of each application, is configured using software tools.

[0010] A particular product is nowadays used, the so-called SML400GP SSI-based product, which is part of a family of products known as Smartlock 400, which was designed as a successor to previous Solid State Interlocking (SSI) products.

[0011] In particular, the logic for a specific SML400GP SSI-based interlocking application is expressed using an extension of the SSI language, known per se as the SML400 language. The SML400GP SSI-based product consists of interlocking hardware, software, communications technology, as well as tools for the preparation and verification of data.

[0012] The core of the product is the well-known Central Interlocking (CIXL) module, and it interprets, in a manner known per se, SSI data, which in the following of the description will be referred as application data. The application data are prepared off-line in a manner known per se, preferably in an office environment using standard computers, and they are subsequently embedded in the CIXL.

[0013] The application data is the data specific to the signaling area that the interlocking control system controls; the CIXL contains fixed software used to interpret these specific application data.

[0014] As above disclosed, the operation of the interlocking control system is based, in a manner known per se, on exchange of data, in the form of signals, between the interlocking control system and the interlocking equipment.

[0015] The application data define the interlocking equipment logic for a predetermined signaling application and are usually verified and validated before being supplied to the interlocking control system. This verification includes, in a manner known per se, a series of manual and automatic procedures aiming at checking that the data are suitable for being installed in the interlocking control system from a formal point of view, i.e. that they fulfill formal constraints.

[0016] In particular, the application data must satisfy the processing requirements of the CIXL in order to be sufficiently self-consistent and complete so as not to cause the CIXL interpreter software to malfunction. Tools exist to automatically verify the application data, but they only perform checks to ensure that the application data meets run-time constraints of the CIXL. They do not check if certain logical conditions, in particular safety conditions, are satisfied, i.e. conditions that could lead to incidents that may cause harm to trains and their passengers.

[0017] Other tools are used to execute the application data in a laboratory test environment to provide confidence that the data are compliant to the signaling principles and the interlocking logistical requirements, however, the setting-up and running of the tests is a time consuming process, and it is almost impossible to create test scenarios that would take into account every possible circumstance that could lead to a violation of a safety condition.

[0018] In order to solve the problem of safety valida-

tion, some methods for checking safety requirements of application data used in an SSI interlocking control system have been proposed.

[0019] Document "Towards an Integrated Model Checker for Railway Signalling Data", Michael Huber and Steve King, Springer-Verlag Berlin Heidelberg 2002, p. 20, discloses the use of model checking techniques, known per se, for automatically checking safety conditions of application data used in an SSI interlocking control system. The technique disclosed involves the automatic translation of a subset of the whole application data into a predetermined model expressed using a formalism different from the one with which the application data have been created, so that they can be checked by using a pre-existing general purpose model checking software. The behavior of the predetermined model is a close approximation of the behavior of an interlocking control system employing the application data, but the use of the general purpose model checking software is inefficient and requires expert skills. The document concludes that a purpose-build SSI data checker would be more optimal and usable.

[0020] Document "Verification of railway interlocking systems", Simon Busard, Quentin Cappart, Christophe Limbrèe, Charles Pecheur, Pierre Schaus, in Proceedings ESSS 2015 discloses a more recent experiment of using a general purpose model checker software, known per se, on application data of SSI interlocking systems, which shows better performance results. Like the previous solution, this solution involves the automatic translation of the application data into the formalism required by the model checker software, and the behavior of the model is an approximation of the behavior of the interlocking control system employing the application data. However, only certain types of data are included in the model, based on the assumption that only those types of data are safety related, ignoring the possibility that the misuse of other types of data could lead to an unsafe state. The technique used involves checking the model against test scenarios that manifest different train movements that may be made in the interlocking application, so it is more a test via simulation approach than a proof approach. The technique is also currently limited to checks on a single interlocking application, i.e. it has not been used to verify application data of communicating interlockings.

[0021] In view of the above, there is therefore the need to provide a method for checking safety requirements of SSI-based data used in an interlocking control system which checks automatically all possible scenarios, which does not employ a general purpose model checking software and/or which does not requires expert skills, thus overcoming the limitations of the prior art solutions.

[0022] This and other objects are fully achieved by virtue of a method for checking safety requirements of SSI-based data used in an interlocking control system having the characteristics defined in independent claim 1.

[0023] Preferred embodiments of the invention are specified in the dependent claims, whose subject-matter is to be understood as forming an integral part of the present description.

[0024] Further characteristics and advantages of the present invention will become apparent from the following description, provided merely by way of a non-limiting example, with reference to the enclosed drawings, in which:

- Figure 1 shows a block diagram of the steps of a method for checking safety requirements of SSI-based data used in an interlocking control system according to the present invention.

[0025] The method according to the present invention is arranged to validate the application data that the CIXL subsystem interprets against safety requirements.

[0026] The method for checking safety requirements of SSI-based data used in an interlocking control system according to the present invention uses the same data model in the same environment in which the existing tools of the SML400GP SSI-based toolset run.

[0027] The method for checking safety requirements of SSI-based data used in an interlocking control system according to the present invention is preferably executed by a computer-implemented software module, and it comprises the steps and sub-steps defined herein below and schematically illustrated in the block diagram of figure 1.

[0028] In a first step 2, an application data file is generated in a manner known per se. The application data file comprises SML400 application data of the type which is usually installed on a CIXL. SML400 application data is a code that represents interlocking logic, i.e. an implementation of the signaling requirements that a CIXL must satisfy.

[0029] The interlocking logic is applied to the interlocking equipment in zones corresponding to signaling areas of the transportation network that the CIXL controls. In other words, the application data are created from code related to one or more virtual interlockings, referred to as VIXLs.

[0030] The logic for each VIXL of a CIXL is defined using a procedural, object-centered language, which is designed to be backwards compatible with the language used for configuring SSI-based interlocking control systems.

[0031] The application data are interpreted on contents of memories representing states of signaling functions (e.g. signals, switches, track sections and routes) that the interlocking control system controls. Iterating in cycles, the CIXL receives indications of the current states of the signaling functions, updates the memories accordingly as the logic demands, sends commands to control the signaling functions to their new states, and processes requests from a signaler (or requests made within the application data).

[0032] In a subsequent step 4, a constraint violation

file is prepared, this file being preferably written in a manner known per se in SSI-like notation. The constraint violation file contains data representative of constraint violations expressed in SML400-like syntax. The constraint violation file comprises therefore a plurality of conditions that describe a plurality of unsafe scenarios of the interlocking equipment.

[0033] For example, a constraint violation may be used to express conditions when it would be unsafe to move a particular switch, such as the following scenario. It is unsafe whenever the switch is in one position and at least one of the track sections on which it lies becomes occupied, in the case when the switch is moved to the other position and at least one of the track sections on which it lies is already occupied.

[0034] In a subsequent step 6 data of the application data file are selected according to a first constraint violation condition of the plurality of violation constraint conditions of the constraint violation file. In particular, only the data which correspond to an unsafe scenario defined in the first constraint violation condition are selected from the whole data of the application data file. For example, with reference to the above-disclosed example related to the movement of a switch, all the data which refer to the unsafe movement of the switch are selected.

[0035] At this point, in a subsequent step 7, at least one application data context is determined, for the unsafe scenario of the first constraint violation condition, in a manner per se known. A context comprises in a manner per se known a plurality of paths through the application data that process a specific signaling request. Taking the above example of the switch movement, one context may be for a request to move the switch, and another context may be for a request to set a route which in turn relies on the switch being moved. The paths in a context typically contain data that test and/or update relevant parts of the signaling function memories so that, when executed, enables the particular request to be fulfilled. As a consequence, to each unsafe scenario of the plurality of unsafe scenarios contained in the constraint violation file corresponds at least one context, preferably a plurality of contexts, each context comprising a plurality of paths in the application data. If a context comprises paths that could lead to an unsafe state, the method forces them to be executed.

[0036] Returning now to figure 1, at step 8, for the unsafe scenario identified at step 6, all the variables that define all possible states of the scenario are initialized, thus setting an initial state for said variables. In particular, it is known that a scenario corresponds to a set of variables disclosing the scenario from the point of view of the settings of the interlocking equipment, i.e. each variable discloses a predetermined setting of one of the interlocking equipment. At step 8, predetermined initialized values are therefore assigned to said variables, thus obtaining an initial state for the unsafe scenario.

[0037] Subsequently, at step 10, the paths belonging to the at least one context identified at step 7 are executed, in the application data context itself. In particular, for the unsafe scenario identified at step 6, starting from the initial state, all possible paths of the at least one context are followed. Therefore, for the given unsafe scenario, starting from a predetermined initial state, each path is executed in the application data and respective resulting states are obtained.

[0038] A this point, at step 12, at the end of each path, a check is done to determine if an unsafe state has been reached, by comparing the respective resulting state with the data of the scenario (i.e. with the data of the constraint violation file).

[0039] If no unsafe state has been detected, it is further determined if the resulting state (which should therefore be a safe state) has not been reached.

[0040] Steps 10 and 12 are repeated, starting each time, for each path, with the corresponding resulting state reached at the previous step, until unsafe states or no new safe states are reached.

[0041] At the end of the loop, at step 14 each path that leads to an unsafe state is displayed to a user in a manner per se known.

[0042] The above disclosed steps 6 to 12 are repeated for any constraint violation condition of the plurality of violation constraint conditions of the constraint violation file.

[0043] The method for checking safety requirements of SSI-based data above disclosed operates on the same data model on which the other SML400 application data tools operate. The data model is accessed and interpreted in a way that is equivalent to the way the CIXL interprets the application data. This means that the method according to the present invention is run on data that are semantically correct and that manifest similar behavior as the target data.

[0044] As clearly detailed in the above, the method according to the present invention allows to validate CIXL application data against specified safety constraints, or rather to prove that there is no execution path in the application data that would result in a state that violates the constraints.

[0045] To this end, starting from an initial state, the method performs an exhaustive search on parts of the application data that could lead to an unsafe state, by executing those parts on that state and all new safe states that result from the execution, until an unsafe state is reached or all states have been searched.

[0046] The main advantages of the method according to the present invention are the followings:

- limited informatics skills are required to perform the method;
- the validation method is performed by a user-friendly software system that provides support for the interlocking configuration process: it is enough to create a text file with the constraint violations expressed in a few lines using SSI-like notation and to select a menu command of the interlocking configuration

system;

- the method is applied to application data of single VIXLs or on data that relate to multiple VIXLs of the same interlocking or different interlockings, thus enabling safety requirements of data related to communicating interlockings to be checked.
- the method can be performed in phases, by running separate proofs to validate safety requirements related to different signaling functions of different VIXLs, to enable the validation activity to be performed in parallel;
- the method splits the application data into manageable partitions and it uses bespoke algorithms to search them efficiently;
- the method applies domain-specific heuristics (knowledge about signaling applications in general and SSI-based applications in particular) that are derivable automatically from the types of constraint violations. These techniques serve to limit the search space for each proof, thus making the solution feasible for complex data.

[0047] In experiments done, the method has found known data errors, related to a certain type of safety condition, in reasonable time.

[0048] Although the SML400GP SSI-based application data tools operate on a model of the data rather than on the target data, unlike other solutions, all of the target data are represented in the model. In particular, according to the method of the present invention, all data are considered in the proofs, excluding parts of the data that are evaluated to be irrelevant to the safety condition currently being proved, rather than ignoring certain parts of the data a priori.

[0049] Advantageously the invention concerns also a computer program product comprising programming code instructions adapted for running a method as described here-above when executing by a calculation unit, such as a processor.

[0050] Clearly, the principle of the invention remaining the same, the embodiments and the details of production can be varied considerably from what has been described and illustrated purely by way of non-limiting example, without departing from the scope of protection of the present invention as defined by the attached claims.

Claims

1. Method for checking safety requirements of SSI-based data used in an interlocking control system for controlling an interlocking equipment, the method comprising the steps of:

    a) obtaining (2) application data representative of interlocking logic operations of the interlocking equipment;
    b) preparing (4) a constraint violation file containing data representative of a plurality of constraint violation conditions, said data describing a plurality of unsafe scenarios of the interlocking equipment;

    for each constraint violation condition of the plurality of constraint violation conditions:

    c) selecting (6) data of the application data according to the constraint violation condition, said selected data corresponding to a predetermined unsafe scenario of the plurality of unsafe scenarios defined in the constraint violation file;
    d) determining (7) at least one predetermined context associated to said unsafe scenario, said context comprising a plurality of paths through the application data;
    e) initializing (8) variables that define all possible states of said unsafe scenario, thus obtaining a predetermined initial state, said variables being representative of the scenario from the point of view of settings of the interlocking equipment;
    f) executing (10), starting from said initial state, all possible paths of the context in the application data, thus obtaining respective resulting states;
    g) at an end of each path, determining (12) if an unsafe state has been reached by comparing a respective resulting state with the data of the constraint violation file;
    h) if no unsafe state has been detected, determining (12) if the resulting state has not been reached;
    -i) repeating steps f), g) and h), starting, for each path, from the respective resulting state, until unsafe states or no new states are reached.

2. Method according to claim 1, further comprising, after step i), the step of displaying (14) each path that leads to an unsafe state.

3. Method according to claim 1 or 2, wherein the application data comprise SML400 application data of the type installed on a central interlocking.

4. Method according to any of the preceding claims, wherein the constraint violation file contains data representative of constraint violations expressed in SML400-like syntax.

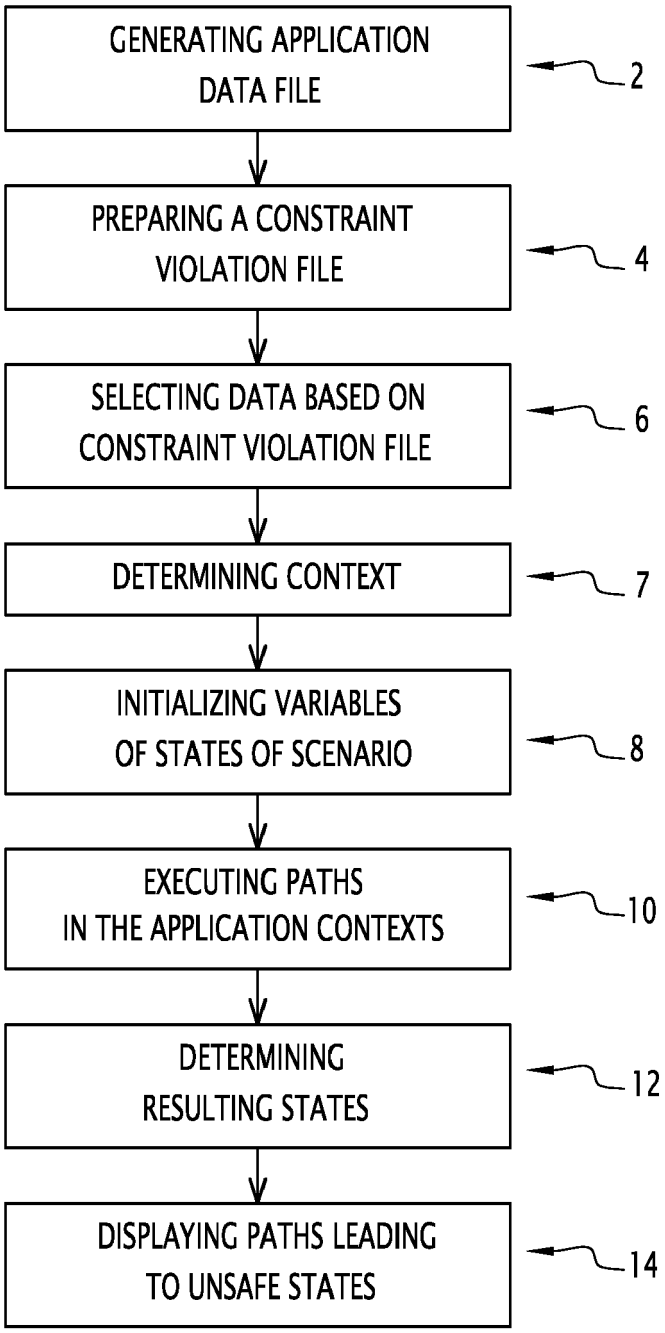
GENERATING APPLICATION DATA FILE
2
PREPARING A CONSTRAINT VIOLATION FILE
4
SELECTING DATA BASED ON CONSTRAINT VIOLATION FILE
6
DETERMINING CONTEXT
7
INITIALIZING VARIABLES OF STATES OF SCENARIO
8
EXECUTING PATHS IN THE APPLICATION CONTEXTS
10
DETERMINING RESULTING STATES
12
DISPLAYING PATHS LEADING TO UNSAFE STATES
14

FIG.1

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 17 30 5477

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A,D	SIMON BUSARD ET AL: "Verification of railway interlocking systems", ELECTRONIC PROCEEDINGS IN THEORETICAL COMPUTER SCIENCE, vol. 184, 1 January 2015 (2015-01-01), pages 19-31, XP055419704, DOI: 10.4204/EPTCS.184.2 * chapter 1, first and third paragraph; chapter 3 *	1-4	INV. B61L19/06 B61L21/04 B61L27/00 G06F9/00
A	WO 2006/111469 A2 (ALSTOM FERROVIARIA SPA [IT]; TRAMONTANA FRANCESCO [IT]; MINKOWITZ CYDN) 26 October 2006 (2006-10-26) * claims 1 and 2; paragraph bridging pages 8 and 9 *	1-4	
A	N N: "signalling solutions -Smartlock 400", Signalling Solutions Ltd., 31 August 2010 (2010-08-31), pages 1-16, XP055419677, Hertfordshire Retrieved from the Internet: URL:https://signallingsolutions.com/wp-content/uploads/SSL-A4-SL400-Bro.pdf [retrieved on 2017-10-27] * the whole document *	1-4	

TECHNICAL FIELDS SEARCHED (IPC)
B61L
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	2 November 2017	Plützer, Stefan

CATEGORY OF CITED DOCUMENTS

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

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

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 17 30 5477

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.

02-11-2017

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2006111469 A2	26-10-2006	AU 2006237274 A1	26-10-2006
		CN 101167050 A	23-04-2008
		EP 1875342 A2	09-01-2008
		US 2008189689 A1	07-08-2008
		WO 2006111469 A2	26-10-2006

EPO FORM P0459

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

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.

Non-patent literature cited in the description

- **MICHAEL HUBER ; STEVE KING.** Towards an Integrated Model Checker for Railway Signalling Data. Springer-Verlag, 2002, 20 **[0019]**
- **SIMON BUSARD ; QUENTIN CAPPART ; CHRISTOPHE LIMBRÈE ; CHARLES PECHEUR ; PIERRE SCHAUS.** Verification of railway interlocking systems. *Proceedings ESSS,* 2015 **[0020]**