Standardisation Considerations for Autonomous Train Control Technical Report

Anne E. Haxthausen[1][0000−0001−7349−8872], Thierry Lecomte[2][0000−0001−8977−4827], and Jan Peleska[3][0000−0003−3667−9775]

[1] DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark
aeha@dtu.dk
[2] CLEARSY, Aix en Provence, France
thierry.lecomte@clearsy.com
[3] University of Bremen, Department of Mathematics and Computer Science, Germany
peleska@uni-bremen.de

Abstract. This technical report is an extended version of a paper submitted to Safecomp 2022. We review software-based technologies already known to be, or expected to become essential for autonomous train control systems. It is discussed which types of technology can be developed and certified already today on the basis of existing railway standards. Other essential technologies, however, require modifications or extensions of existing standards, in order to provide a certification basis for introducing these technologies into non-experimental "real-world" rail operation. Regarding these, we check the novel pre-standard ANSI/UL 4600 with respect to suitability as a certification basis for safety-critical autonomous train control functions based on methods from artificial intelligence. As a thought experiment, we propose a novel autonomous train controller design and perform an evaluation according to ANSI/UL 4600. This results in the insight that autonomous freight trains and metro trains using this design could be evaluated and certified on the basis of ANSI/UL 4600.

Keywords: Autonomous train control, Standards, Certification, Verification, Validation

1 Introduction

This technical report is an extended version of a paper submitted to Safecomp 2022[4].

Motivation Recently, the investigation of autonomous trains has received increasing attention, following the achievements of research and development for

[4] https://safecomp22.iks.fraunhofer.de

autonomous vehicles in the automotive domain. The business cases for autonomous train control are very attractive, in particular for autonomous rolling stock and metro trains [21].

It is well known that current safety-related standards for train control systems are not prepared for verification, validation (V&V), and certification of autonomous systems. Erskine et al. [8] point out that the current standards even expressly forbid AI-based software components to be used as soon as the required safety integrity level is above SIL-1 (ATP and interlocking systems are SIL-4).

However, if these explicit exclusions of AI-based functions were removed from today's standards, they would still be insufficient for autonomous products, because several essential characteristics of autonomous transportation systems are not addressed. (1) For modules using machine learning, the *safety of the intended functionality* no longer just depends on correctness of a specification and its software implementation, but also on the completeness and unbiasedness of the training data used [12]. (2) Agent behaviour based on belief databases and plans cannot be fully specified at type certification time, since the behaviour can change in a significant way later on, due to machine learning effects, updates of the belief database, and changes of plans during runtime [1]. (3) Laws, rules applying to the transportation domain, as well as ethical rules, that were delegated to the responsible humans (e.g. train engine drivers) in conventional transportation system, are now under the responsibility of the autonomous system controllers. Therefore, the correct implementation of the applicable rule bases, as well as the override conditions for certain rules (e.g. *"A red traffic light may be disobeyed if this helps to avoid an accident"*) needs to be validated [9].

In this light, we analyse the pre-standard ANSI/UL 4600 [22] that addresses the safety assurance of autonomous systems on system level. Together with several sub-ordinate layers of complementary standards, it has been approved by the US-American Department of Transportation for application to autonomous road vehicles.[5] While examples and checklists contained in this document focus on the automotive domain, the authors claim that the standard should be applicable to *any* autonomous system, potentially with a preceding system-specific revision of the checklists therein. To the best of our knowledge, the ANSI/UL 4600 pre-standard is the first "fairly complete" document addressing system-level safety of autonomous vehicles, and its applicability to the railway domain has not yet been investigated.

Main Contributions We propose a novel design for an autonomous train control architecture that should serve as the setting for a thought experiment analysing whether such a system could (and should) be certified on the basis of conformance to the pre-standard ANSI/UL 4600 [22]. As a design rule, we advocate the strict separation between conventional control sub-systems that

[5] see `https://www.youtube.com/watch?app=desktop&v=xCIjxiVO48Q&feature=youtu.be`

can be certified on the basis of existing standards, and novel, AI-based subsystems that are needed to enable autonomy.

We assume a heterogeneous operational environment with diverse track-side equipment, as can be expected in Europe today. Furthermore, we assume the availability of controlled allocation and assignment of movement authorities, as is performed by today's interlocking systems (IXL, potentially supported by radio block centres (RBC)). Apart from the communication between train and RBC/IXL, no further "vehicle-to-X" communication channels are assumed. Summarising, we analyse scenarios, how autonomous trains could travel through today's existing railway networks in a way that a convincing safety case conforming to ANSI/UL 4600 can be elaborated.

We demonstrate that even this fairly moderate scenario of autonomous train control will only be certifiable for freight trains, metro trains, and trams. In contrast to this, we deem the trustworthy safety assurance of high-speed passenger trains to be infeasible today. This assessment is justified by the fact that obstacle detection function can only be executed to operate with sufficient reliability for trains with speed up to 120 km/h.

Distinction from Related Work It is important to point out that visions of autonomous train control far beyond the "fairly moderate" concepts considered in this technical report exist. Trentesaux et al. [21] point out the attractiveness of business cases based on trains autonomously negotiating their way across a railway network in an open, uncontrolled (i.e. not fully secured) environment. To this end, they suggest a train control architecture whose behaviour is based on plans that are continuously adapted to increase safety and efficiency. A typical software implementation paradigm for this type of behaviour would be *belief-desire-intention (BDI) agents* [1]. Unsurprisingly, the authors come to the conclusion that the safety assurance and certification of such systems will be quite difficult. Indeed, we will point out below that exactly this type of train control is the one with the least prospects of becoming certifiable in the future.

The technical report presented here is inspired by the work of Koopman et al. discussing certification issues of road vehicles [15,14,13]. It will become clear in the remainder of this technical report, however, that their results cannot be "translated in one-to-one fashion" for the railway domain.

Overview In Section 2, the standards and pre-standards of interest in the context of this technical report are briefly reviewed. In Section 3, we describe existing technology that is needed to realise autonomous train control systems. Up to now, most of these technologies have been used in proof-of-concept projects, so that conformance to standards and certification was not yet an issue. In Section 4, we present a new reference architecture for autonomous train control systems that we advocate, due to having fair chances of becoming certifiable in the near future. In Section 5, we perform an evaluation of certifiability according to ANSI/UL 4600 for the reference architecture introduced before. Section 6 contains a conclusion.

In Appendix A, interfaces and behaviour of essential train control functions (the so-called *kernel*) are formally modelled. This model serves as a "proof of concept" to demonstrate that these functions can still be realised *without* AI-based methods and therefore be evaluated and certified according to existing standards in the railway domain.

Throughout the text, we refer to related work where appropriate.

2 Standardisation and Certification

In the railway domain, safety-critical track-side and on-board systems in Europe must be designed, verified and validated according to the CENELEC standards EN50126, EN50128, and EN50129 [4,3,5], in order to pass type certification. None of these documents provides guidance for V&V of AI-based sub-functions involving machine learning, classification techniques, or agent-based autonomous planning and plan execution. Since, as outlined in Section 3, autonomous train control depends on such AI-based techniques, this automatically prevents the certification of autonomous train control systems on the basis of these standards.

To the best of our knowledge, the ANSI/UL 4600 safety pre-standard for the evaluation of autonomous products [22] is the first document that is sufficiently comprehensive to serve (in modified and extended form) as a certification basis for system-level safety aspects of autonomous products in the automotive, railway, and aviation domains. The standard is structured into 17 sections and 4 annexes. Section 5 addresses the elaboration of safety cases and supporting arguments in general, and Section 6 covers general risk assessment. For the context of the technical report presented here, Section 7 and Section 8 are the most relevant parts.

The focus of Section 7 is on interaction between humans, animals and other systems and the autonomous system under evaluation (denoted as the *item* in the standard). While this section needs extensive cover for autonomous road vehicles in urban environments, its application is more restricted for the railway domain: here, the pre-planned interaction between humans and autonomous trains takes place in train stations on platforms, during boarding and deboarding. The safety of these situations is handled by the passenger transfer supervision sub-system introduced in Section 4. On the track, humans are expected on railway construction sites and level crossings, otherwise their occurrence is illegal. For both legal and illegal occurrences, the on-track interaction between humans and the train is handled by the obstacle detection sub-system described in Section 4.

Section 8 of the standard explicitly addresses the autonomy functions of a system, as well as auxiliary functions supporting autonomy. It explains how the impact of autonomy-related system functions on safety should be addressed by means of hazard analyses. For the non-negligible risks induced by these functions, it has to be explained how mitigating functions have been incorporated into the system design. The operational design domain and its sub-domains for each operational mode (e.g. degraded functionality in exceptional situations) have to be specified. To present hazards caused by autonomy functions, associated design

decisions and mitigations in a well-structured manner, the section is structured according to the *autonomy pipeline*

> *sensing → perception → evaluation (possibly based on machine learning) → planning → prediction → control by actuation.*

The other sections of ANSI/UL 4600 cover the underlying software and systems engineering process and life cycle aspects, dependability, data, networking, V&V, testing, tool qualification, safety performance indicators, and assessment of conformance to the standard. These aspects are beyond the scope of this technical report.

3 Technology

A number of technologies are required to implement autonomous train control on existing railway networks. The non-modification of existing infrastructure, in particular track-side signalling equipment, is sought in order to facilitate their deployment at lower cost.

We agree with the recommendations of the Federal Railroad Administration of the U.S. Department of Transportation [26] who envision a *sensor platform* combining several different technologies to identify *objects of interest (OOI)* (obstacles, landmarks enabling the improvement of position calculation, train stations, ...) and *conditions of interest (COI)* (*"track is free of obstacles up to location ..."*, *"the train location has distance n meters to its end of movement authority"*, ...). The perception of the immediate train environment is mandatory to ensure a correct navigation regarding signalling equipment but also to avoid catastrophic collisions with obstacles (trains, objects, animals) by perceiving the scene up to its braking distance. The use of different types of sensing techniques and technologies (radar, laser, LiDAR, camera time-of-flight, camera IR) is necessary to obtain a functional capacity for a wide variety of environmental situations. By using different wavelengths or physical principles (or combination of), it is possible to avoid receiving incorrect information (from radar secondary lobe) or becoming completely blind under certain situations. Indeed, weather conditions (precipitation, snow, humidity, high light levels, mist, dust, ...) have a direct impact on the quality and accuracy of the perceived information, which can strongly alter the representation of the observed scene. For example, an occlusion (spot on an optic) could hide an obstacle; a low sun on the horizon in the axis of the rails could prevent the detection of a light due to sensor saturation.

Similar to autonomous cars, which have been the subject of numerous high-profile attacks in recent years (e.g. the addition of tape to speed limit signs, the transmission of a false GPS signal), the possibility of malicious attacks, that might have been detected by a human operator, must also be taken into account when processing perceived information. One can imagine the illumination of a switched-off signal by an external light source. Deep learning techniques are likely to be susceptible to these attacks, which will not have been part of the training

data and will require significant deployment time (building up training and test vectors, certification, software updates) compared to a vigilance instruction given to drivers and applied quickly.

These different sensors each provide specific information that will have to be merged with each other and with the prior knowledge of the environment. The perception of the environment has to be linked with the known, slowly evolving topology and train position to minimise false alarm and faulty signalling element recognition. Localisation is currently based on a number of technologies such as odometer, beacon (metro, main lines), forthcoming GPS positioning (ERTMS/ETCS level 3), and associated algorithms / techniques (which differ from one train manufacturer to another) in order to obtain a target precision.

Moreover localisation could benefit from image analysis from the environment to detect and identify a number of landmarks: the direction of travel combined with the azimuth of one or several detected landmarks could be used to precise the train position with triangulation. However, the geographic position of the track is currently not always well known (this knowledge was not required before), at least not with a precision sufficient to ensure a correct positioning. A number of on-going projects are collecting data to build accurate 3D models including precise track positioning. These 3D models represent a picture at a given time. As such, associated algorithms have to ensure continuous recognition even if the scene is evolving due to expanding vegetation, new infrastructures near the tracks, or landform modified by climatic conditions. Regular 3D models and landmarks would have to be released regularly (with a period to be defined in accordance with the degree of evolutivity of the environment). Marginally evolving regulations could also have an negative impact on the recognition process like the ban on the use of effective herbicides or the installation of anti-noise walls in dense urbanised areas, both modifying the aspect or signature of the zones close to the tracks.

Replacing the driver with a sensor platform has side effects on train handling and the health of the railway system. First because of the wide variety of open world situations, it is difficult to anticipate problems with the perception of the environment (and the difficulty to come to a sound decision), especially when viewed from a perception platform for which there is no/little feedback. The installation of a remote control link appears necessary to allow a human being to regain control if necessary. This of course assumes that the autonomous train has the self-assessment capability to determine if it is in trouble and needs external help. Second because the driver could perceive subtle variations in operating conditions such as the sound of a rail breaking or the excessive amplitude of the damper stroke during overspeed (switch passed at 170km/h instead of 100km/h for a TGV [6]). Specific technical means must then be used to maintain knowledge of the system status, such as maintenance trains able of analysing the rail structure with ultrasound or Bayesian networks to optimise maintenance strate-

[6] http://www.bea-tt.developpement-durable.gouv.fr/IMG/pdf/rapport_beatt_2020_01.pdf

gies [2]. Rail wear can also be addressed with the design of lightweight trains, suitable for regional lines with low traffic.

A number of concurrent projects are underway to demonstrate the feasibility of operating autonomous trains on standard tracks. The Rio Tinto unmanned trains operated since 2019 over 1500km network paved the way of automated heavy-haul freight railway after 10 years of effort. The challenge is now to reproduce this performance with saturated train lines and many possibilities for interaction with the intertwined natural and urban environments. SNCF (French railways) has initiated projects aimed at freight and passenger transport. Experiments were carried out to test both the perception and recognition systems for signals located along the track, and the geolocation system. The focus is on the development of the semi-autonomous driving system which includes automatic acceleration and braking of the train. As for autonomous cars, the experiments are carried out under the supervision of a human driver and with a limited speed (up to 25km/h). Other experiments like EcoTrain [7] or FerroCampus [8] are dedicated to low emission trains for low traffic lines. Finally Supraways [9] or UrbanLoop [10] explore new mobilities, suspended or through connected loops, and as such and are likely to use significantly different capture technologies.

4 A Reference Architecture for Autonomous Train Controllers

Operational Design Domain The *operational design domain (ODD)* is defined in ANSI/UL 4600 as *"The set of environments and situations the item is to operate within."* [22, 4.2.30]. Safety cases conforming to this standard need to refer to the applicable ODD sub-domains, when presenting safety arguments for autonomous system functions.

For the autonomous trains architecture advocated in this technical report, we structure the ODD into four sub-domains, as shown in Fig. 1. The autonomous trains admissible in this ODD are restricted to the classes freight trains, metro trains, and trams. This decision will be justified below when analysing the perception functions involved: according to the state of the art, obstacle detection can only be expected to operate reliably up to a train speed of 120 km/h [18]. We are not aware of any available technology providing reliable obstacle detection functionality for high-speed trains.

In the *autonomous normal operation (ANO)* sub-domain, the train is fully functional and controlled with full autonomy within the range of its current position and the end of movement authority (MA) obtained from the interlocking system (IXL) via radio block controller (RBC). The only special environment conditions required is that an operative IXL is able to communicate with the

[7] https://www.dailymotion.com/video/x7ujwl0
[8] https://www.ferrocampus.fr/ferrocampus/defis-et-road-map/
[9] http://www.supraways.com/
[10] https://urbanloop.univ-lorraine.fr/

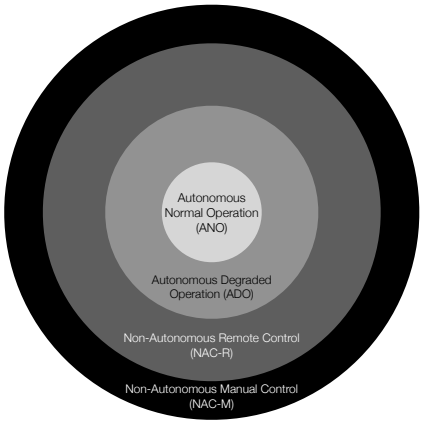

Fig. 1. Subdomains of the ODD for autonomous trains.

train. The system is supposed to be operative under arbitrary weather conditions, on single track or multi-track rail networks, day and night, in broad daylight, in tunnels, and in train stations. However, the ANO sub-domain requires that certain sub-systems of the overall train control system are operative.

In *autonomous degraded operation (ADO)*, the train is operated still autonomously, but with degraded performance (e.g., with lower speed). The ADO sub-domain is entered from ANO, for example, if the available position information is not sufficiently precise, so that the train needs to be slowed down until trustworthy position information is available again (e.g. because the train passed a balise with precise location data). Another example for a transition from ANO to ADO is the situation where a train trip has been caused due to a violation of an end-of-MA, so that the train had to be stopped by emergency brakes and resides outside its MA. After having come to a standstill, the train may still be controlled autonomously, but under command of an RBC telling the train to reverse to the most recent end-of-MA, or to proceed to the next safe location.

In case of a loss of vital autonomous sub-functions (see description of these functions below), the train enters one of the *non-autonomous control (NAC)* domains. In NAC-R, the train can still be remotely controlled by a human from some centralised facility. The remote control technology exists for decades already, and it can be verified, validated , and certified by conventional means. If no remote control facility is available, the train has to be manually controlled by a train engine driver boarding the train or by another manually controlled locomotive that can be used to tow the train to the next station (or any other suitable maintenance point). The transitions between the four ODD sub-domains can be formally modelled and hard-coded at type certification time. They need not rely on AI-based methods (see Appendix A).

In the subsequent paragraphs, we will investigate an autonomous on-board train controller, whose functional top-down decomposition is shown in Fig. 2.

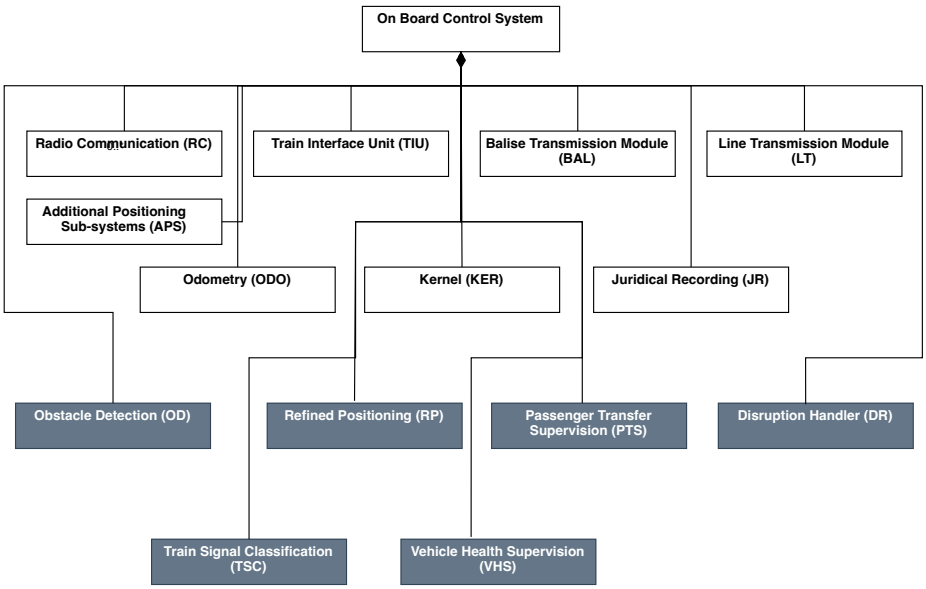

Fig. 2. Reference architecture of autonomous train to be considered for certification.

The grey boxes are functions required for autonomous trains only, the white boxes are typical components of modern conventional on-board units supporting automated train protection (ATP) [23]. These modules can be re-used (in case of the kernel in extended form) for the autonomous case, as explained below. We advocate that autonomous train architectures should carefully separate conventional ATP modules from (potentially AI-based) modules supporting autonomous automated train operation (ATO). Such a careful segregation in the on-board controller design allows for conventional certification of the conventional modules, so that only the modules implementing the grey boxes shown in Fig. 2 need to be certified according to novel standards.

Conventional On-Board Modules The central module of the conventional ATP functionality is the *kernel* which executes the essential ATP operations like speed monitoring, update and observation of movement authorities, and brake control. All decisions about interventions of the normal train operation are taken in the kernel. Based on the status information provided by the other sub-systems, the kernel controls the transitions between ODD sub-domains. This control function can be specified and implemented in the conventional way, since a comprehensive behavioural model can be provided. Therefore, the kernel can still be certified by conventional means, though it has to be extended by new functions induced by autonomous operation.

Interventions are executed by the kernel through access to the *train interface unit*, for activating or releasing the service brakes or emergency brakes. The decisions about interventions are taken by the kernel based on the information provided by peripheral modules: (1) The *odometry module* and *balise transmission module* provide information for extracting trustworthy values for the actual train positions. In modern high-speed trains, *additional positioning sub-systems* provide satellite positioning information in combination with radar sensor information to improve the precision and the reliability of the estimated train location. (2) The *radio communication module* provides information about movement authority and admissible speed profiles, as sent to the train from interlocking systems via radio block centres. In the train-to-trackside transmission direction, the train communicates its actual position to radio block centre/interlocking system. (3) The *line transmission module* provides signal status information provided by trackside equipment for the train. (4) The *juridical recording module* stores safety-relevant kernel decisions and associated data.

Note that, depending on the technical stage of construction of the track-side equipment, not all the data providers listed above in (1) to (3) will be available. In the non-autonomous case, the missing information is compensated by the train engine driver who, for example, visually interprets signals if trackside line transmission equipment is unavailable.

Note that the kernel is extended by additional sub-modules that become necessary in the autonomous case. This is specified in more detail below. These additional sub-modules, however, do not required AI-based technology, and their behaviour can be fully specified before type certification time. Consequently, this extended kernel can still be evaluated and certified according to the CENELEC standards.

Modules Supporting Autonomous Trains Operation The *obstacle detection module (OD)* uses a variety of sensors (cameras, LiDAR, radar, infrared, ...) [26] to determine whether obstacles are on the track ahead. In case of an obstacle detection, an estimate for the distance from train to obstacle is needed in order to decide (in the kernel) whether an activation of emergency brakes is required or if the service brakes suffice. A further essential functional feature is the distinction between obstacles on the train's track and obstacles or approaching trains on neighbouring tracks, where no braking intervention is necessary. Camera-based obstacle detection can be performed by conventional computer vision algorithms or by means of image classification techniques based on neural networks and machine learning [18,27]. None of the available technologies are sufficiently precise and reliable to be used alone for obstacle detection [26]. Instead, a sensor fusion based on several technologies is required. In any case, experimental evidence is only available for train speeds up to 120 km/h [18], this induces our restriction to autonomous freight trains, metro trains, and trams. From the perspective of the autonomy pipeline described in Section 2, the obstacle detection module performs sensing and perception. It provides the *"obstacle*

present in distance d" information to the kernel which operates on a state space aggregating all situational awareness data.

The *refined positioning module (RP)* provides additional train location information, with the objective to compensate for the train engine driver's awareness of the current location that is no longer available in the autonomous case. A typical use case for refined positioning information is the train's entry into a station, where it has to stop exactly at a halt sign. To achieve the positioning precision required for such situations, signposts and other landmarks with known map positions have to be evaluated. This requires image classification, typically based on trained neural networks [20]. Again, conventional image recognition based on templates for signs and landmarks to expect can be used [16] to allow for fusion of conventional and AI-based sub-sensors.

The *train signal classification module (TSC)* is needed on tracks without line transmission facilities. Signals and other signs need to be recognised and classified. This task is very similar to that of identifying traffic signs in autonomous cars. Again, implementations are based on trained neural networks or on conventional technology [19,16], enabling mixed conventional and AI-based sensor fusion.

Summarising, the OD, RP, and TSC modules represent perception functions helping the kernel to update its situational awareness status. All three modules can be realised by means of sensor fusion techniques involving both conventional image recognition methods and trained neural networks. These observations become important in the sample evaluation performed in Section 5.

The *passenger transfer supervision module (PTS)* is needed to ensure safe boarding and deboarding of passengers. It applies to the fully autonomous case of passenger trains being operated without any personnel. This module requires sophisticated image classification techniques, for example, to distinguish between moving adults, children, and other moving objects (e.g. baggage carts on the platform). Again, PTS is a perception function providing the kernel with the *"passengers still boarding/deboarding at door ..."* and *"passengers or animals dangerously close to train"* information which shall prevent the train from starting to move and leave the station. Sensor fusion with conventional technology could be provided by various sorts of light-sensors, in particular, safety light curtains[11].

The *vehicle health supervision module (VHS)* is needed to replace the train engine drivers' and the on-board personnel's awareness of changes in the vehicle health status. Indications for such a change can be detected by observing acoustic, electrical, and temperature values. The conclusion about the actual health status, however, strongly relies on the experience of the personnel involved. This knowledge needs to be transferred to the health supervision in the autonomous case [21]. Since the effect of human experience on the train's safety is very hard to assess, it is quite unclear how "sufficient performance" of module VHS should be specified, and how it should be evaluated. Therefore, we do not consider this component anymore in the sequel.

[11] `https://en.wikipedia.org/wiki/Light_curtain`

The *disruption handler (DH)* is responsible for re-starting and controlling the train after a *trip* situation leading to an emergency stop [25]. In conventional trains, this situation is handled under the responsibility of the train engine driver who manually steers the train with low speed to the next safe location from where normal operation can be resumed.

We suggest that only very limited capabilities should be implemented in the disruption handler, so that these can be triggered by well-defined conditions and modelled by a library of fully specified state machines (e.g. *"After a train trip due to violation of end-of-MA, communicate with RBC to obtain a safe target destination from where a restart in ANO is possible"*). As a consequence, the disruption handler can be integrated into the kernel and verified, validated, and certified by conventional means.

The obvious disadvantage of these limited disruption handler capabilities is that human interaction due to transitions to sub-domains NAC-R or NAC-M is more frequently required. A more powerful disruption handler, however, would have to be capable of fully rule-based behaviour according to very general principles and priorities, because this is the only available basis of action for coping with *unforeseen events* [9]. Unfortunately, the situation-dependent rules for train control are extremely extensive, and they vary between European countries in a considerable way [21]. Therefore, the verification and validation of rule base adequacy would require a tremendous effort. In the light that also the proof of timing accuracy of rule-based decision making is extremely hard to verify, we conclude that such an addition of capabilities is not desirable. It is much more advisable to strengthen the capabilities of the disruption handler by adding scenarios that can be specified and implemented in the conventional way, to cover more situations.

Dual Channel Plus Voting Design Pattern As a further design decision, we introduce a two-channel design pattern for the modules OD, TSC, RP, and PTS, as shown in Fig. 3.

Each channel has a sensor front end (camera, radar, LiDAR, light sensor, ..., as described in Section 3) for receiving environment information. The sensor front ends use redundant hardware (HW sensors, wiring, power suplies, ...) so that they can be assumed to be stochastically independent with respect to hardware faults. The remaining common cause faults for the sensors (like sand storms blinding all camera lenses) can be detected with high probability, because both sensor data degrade nearly simultaneously.

The sensor frontends pass their raw data to the perception sub-modules, where one should be based on conventional evaluation technology and the other on AI-based methods. Both perception sub-modules pass their result data and possibly failure information from the sensor front ends to a joint voting function that compares the results of both channels and relays the voting result or a failure flag to the kernel. The calculation of the voting result depends on the perception function, as described next.

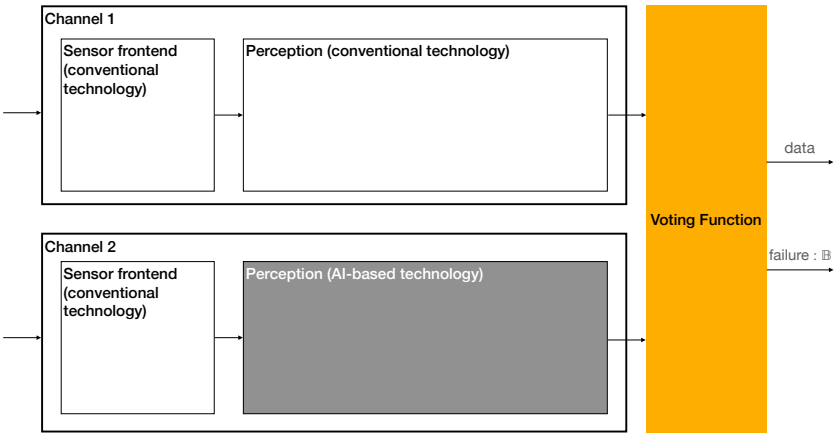

Fig. 3. Two-channel design pattern used for modules OD, TSC, RP, and PTS.

Design of Voting Functions For the obstacle detection (OD), the voting function raises the failure flag if both channels provided contradictory *"no obstacle/obstacle present"* information over a longer time period. For unanimous *"obstacle present"* information with differing distance estimates, the function "falls to the safe side" and relays the shorter distance to the kernel.

For the refined positioning module (RP), the voting function raises a failure if the channels detect different landmarks over a longer period, or if the distance and angle information provided by both channels for the same landmark differs too much. Otherwise, the landmark classifications with mean values calculated from distances and angles are passed on to the kernel.

For the train signal classification module (TSC), the failure flag is raised if different signals are detected by the two channels or if they indicate contradictory signal aspects, or if the train-to-signal distance information differs too much. Otherwise, the signal aspect, together with the shorter distance value is passed on to the kernel.

For passenger transfer supervision (PTS), a failure is raised if one channel continuously shows absence of passengers in the supervised locations, while the other channel indicates the presence of passengers. Otherwise, the *"all clear"* information is passed to the kernel, as soon as both channels indicate that no passengers are present in any of the supervised locations. Conversely, the *"passengers present"* flag is set, as soon as at least one channel indicates the presence of passengers.

Note that it may be advisable to use either more than one two-channel sensors for a given task or one n-channel sensor with $n > 2$. For obstacle detection, for example, channels with time-of-flight camera, infrared camera, radar, and

LiDAR may become necessary to provide sufficient robustness for day/night changes, long/short distances, and various weather conditions. The 2-channels-with-voter principle described here can obviously be extended to an n-channel design.

Extended Kernel – Detailed Design and Behaviour The structure of the operational design domain and the modules supporting autonomy induce kernel extensions. Their design and behaviour is described in Appendix A.

5 A Sample Evaluation according to ANSI/UL 4600

Evaluation Procedure In this section, Section 8 (*Autonomy Functions and Support*) of ANSI/UL 4600 is applied to analyse whether a safety case for the autonomous train control architecture described in Section 4 conforming to this standard could be constructed. The procedure required is as follows [22, 8.1]. (1) Identify all hazards related to autonomy and specify suitable mitigations. (2) Specify the autonomy-related implications on the ODD. (3) Specify how each part of the autonomy pipeline contributes to the identified hazards and specify the mitigations designed to reduce the risks involved to an acceptable level.

Effect of Separation Between Conventional and AI-based Modules on Evaluation As described in Section 4, the central control functions deployed on the kernel module of the autonomous train design advocated here can be evaluated and certified by conventional means, on the basis of the CENELEC standards listed in Section 2. In particular, all planning, prediction, and actuation activities performed by the kernel and its train interface unit are fully specified at type certification time. The control behaviour can be completely modelled by means of state machines with formal semantics [7]. This includes all discrete control aspects (e.g. *"While in ODD sub-domain ANO, never start moving without movement authority"*, or *"trigger emergency brake if close obstacle is detected"*), as well as dynamic control functions (e.g. *"start braking to target according to current speed and position"*, or *"slow down train if confidence in actual position is too low"*). Moreover, the conditions for transiting between the ODD sub-domains introduced in Fig. 1 (e.g. *"Transit to NAC-R in case of obstacle detection module failure, if train can be remotely controlled; otherwise transit to NAC-M"*) can be modelled by a state machine whose behaviour is fully specified at type certification time.

As described in Section 4, sensing and perception for each of the modules OD, RP, TSC, and PTS (see Fig. 2) are based on a two-channel design, each channel containing a sensor and a perceptor sub-component (see Fig. 3). The sensing parts are based on conventional technology, and common cause failures in the redundant sensors can be detected with high probability. We conclude that the sensing parts can be evaluated according to the existing CENELEC standards.

Indeed, according to our analysis, the requirements of ANSI/UL 4600 regarding sensor evaluation are equivalent to those of the CENELEC standards, as long as no novel AI-related technology is already applied in the sensing part [22, 8.3].

As a result of this discussion, we have reduced the evaluation obligations according to ANSI/UL 4600, Section 8, Step (3) of the evaluation procedure to the demonstration that the perceptor components used in the two channels of modules OD, RP, TSC, and PTS conform to the requirements of the ANSI/UL 4600 pre-standard [22, 8.4 and 8.5]. If the perceptor is designed and implemented with conventional technology, only Section 8.4 of ANSI/UL 4600 needs to be applied; we denote this as evaluation step (3a). If, however, the perceptor uses AI-based technology, Section 8.5 (*Machine learning and "AI"-techniques*) of the standard has to be applied as well; we denote this as evaluation step (3b).

Step 1. Hazard Identification of Autonomy Functions – Risk Mitigation It is obvious that failures in any of the four autonomy functions OD, RP, TSC, and PTS shown in Fig. 2 can cause collisions or derailing, leading to severe injuries for some or very many passengers, as presented in Table 1. Therefore, it is obvious that none of these autonomy functions can conform to the standard without providing risk mitigations.

Table 1. Hazards associated with autonomy functions shown in Fig. 2.

Failure in Function ...	**... induces hazards**
Obstacle detection (OD)	Collisions between train and humans, animals, cars and other obstacles; derailing as consequence of collision
Train signal classification (TSC)	Only in absence of line transmission: collisions between train and other trains, humans, animals, cars and other obstacles; derailing due to unrecognised speed restriction
Refined positioning (RP)	Collision with buffer stop in train station, halt in train station with passenger cars outside platform
Passenger transfer supervision (PTS)	Injury of boarding/deboarding passengers

As risk mitigation strategy, the two-channel design with voting described in Section 4 for each of the modules OD, TSC, RP, and PTS is suitable to reduce the risks involved to an acceptable value: due to the methodological diversity of the algorithms involved, the conventional perceptor in Channel 1 and the AI-based perceptor in Channel 2 may be regarded as stochastically independent, provided that both are free of systematic errors. This independence-assumption is justified, because the conventional perceptor uses hand-crafted algorithms to extract features (a pair of rails, an obstacle) from image frames, whereas the AI-based perceptor uses neural networks whose weights have been calibrated during

training phases and possibly further AI-based algorithms for object separation and classification in images. Therefore, the failure probability of the perceptor pair is the product of the individual failure probabilities. The voters attached to the two channels will produce results "on the safe side" as explained in Section 4. A safety-critical failure of the voter (e.g. indicate *"no obstacle present"* though there is an obstacle) can only occur if the two perceptors of the channel not only fail simultaneously but also *fail with the same erroneous result*. Due to the stochastic independence between the two perceptors, this probability is significantly lower than that of a simultaneous failure.

The voters that are part of the two-channel design either deliver results "to the safe side" or indicate a sensor/perceptor failure. In the unsafe failure case, the mitigation actions specified in Table 2 are adequate to avoid hazardous situations (a more general theory of risk mitigation by means of synthesised safety supervisors has been presented by Gleirscher et al. [10]). Both voters and mitigation actions can be specified and implemented by conventional means and evaluated on the basis of the CENELEC standards.

These considerations serve to justify the adequateness of the mitigation strategy for the risks involved in failures of modules OD, RP, TSC, and PT: the two-channel design pattern leaves us with an acceptable risk, if the sensor and perceptor components are free of *systematic* errors.[12]

Table 2. Mitigation actions associated with autonomy functions from Table 1.

Failure in Function ...	**... leads to mitigation action**
Obstacle detection (OD)	Emergency stop and transition to ODD sub-domain ADO.
Train signal classification (TSC)	Only in absence of line transmission: emergency stop and transition to ODD sub-domain ADO.
Refined positioning (RP)	Speed reduction
Passenger transfer supervision (PTS)	Halt train in station until boarding/deboarding with manual assistance has been completed.

Step 2. Autonomy-related Implications on the ODD The operational design domain has already been described in Section 4. As explained there, the transition between ODD sub-domains can be completely and deterministically specified by means of state machines. These state machines will also trigger the mitigation actions listed in Table 2 when required.

[12] To prove full conformance to ANSI/UL 4600, it is also necessary to perform a quantitative calculation of the residual risks. This, however, is beyond the scope of this technical report.

Step 3a. Perceptor Evaluation Both conventionally designed and AI-based perceptors need to be evaluated according to Section 8.4 of ANSI/UL 4600. For the AI-based perceptors, an additional evaluation step is required [22, 8.5.] which is discussed below in Step 3b. In a comprehensive evaluation, this step would be performed of each of the perceptor types needed for the modules OD, TSC, RP, and PTS. Here, we just describe the crucial evaluation steps and provide examples from the perception sub-components of these modules.

All perceptors (conventional or AI-based) but one use image classification techniques: the conventional perceptor for passenger transfer supervision is based on light curtain sensor technique, so it just reports *"object present in door area/object in danger zone close to the train/sensor failure"* without further classification. Therefore, this constitutes a trivial perceptor which just relays light sensor information to the voter. This can be verified and validated by conventional means and will not be considered further here.

All perceptors using image classification techniques need a well-documented test data set which is under configuration control. It has to be argued that the test data set is sufficiently large to cover the ODD. For the passenger transfer supervision (PTS) module, for example, this means that the test data set needs to cover all objects occurring on train platforms: adults, children, disabled persons, animals, baggage trolleys etc. Furthermore, the test data set needs to be unbiased, in the sense that it tests *all* objects occurring in the ODD, with a uniform distribution of test cases.

Next, an ontology needs to be specified, and it has to be checked during testing, that the perception results map correctly to the ontology. For the PTS module, for example, a test should fail if the AI-based perceptor produces result *"adult in door area"* for a child in the door area. For the OD module, the ontology would be simpler, since no detailed classification of obstacles would be needed. However, a suitable ontology would need to introduce at least concepts like *"my track"* and *"other tracks"* and *"obstacle [far from/close to/on] own track"*. A test should fail, for example, if no obstacle is present, but the OD classifies the situation as *"some object on other track"* which would still lead in the correct *"no obstacle close to/on own track"* information, but with an erroneous evaluation for the other tracks.

The data set has to provide sufficiently many images to cover all ontologies used *internally* by the perception component: a conventional image processing algorithm for obstacle detection, for example, might first reduce the image to a "relevant" sub-image corresponding to the area that is close enough to the train's track to be of interest. Then the data set needs to contain image material covering boundary cases like *"obstacle on boundary between relevant and irrelevant image areas"*.

It has to be tested that the perceptor works acceptably robust with objects and events outside the ontology. For example, the PTS classifier should not malfunction if unknown objects (e.g. pieces of furniture not captured in the ontology) are present on the platform.

The sufficiency of the test evaluation criteria has to be justified. For obstacle detection, for example, the false positive and the false negative rates of the *"obstacle present"* flag are insufficient: additionally, the precision of the obstacle distance estimates needs to be taken into account.

We do not discuss the standard evaluation requirements that are specified in ANSI/UL 4600, like verification of software by means of analyses, inspections, and tests that do not differ from the existing CENELEC standards.

Step 3b. Evaluation of AI-based Perceptors All AI-based perceptors in our design apply image classification techniques. To this end, neural networks are trained with deep learning strategies [18]. Therefore, Section 8.5 (*"Machine learning and AI-techniques"*) needs to be applied for further evaluation.

In contrast to conventional image processing methods, a *training data set* is used to calibrate the weights of the underlying neural network. Of course, the training data must be disjoint from the test data set. Similar to the test data set, the training data requires *qualification* [22, 8.5.3.2]. To this end, it has to be justified why the training data set covers the ontology[13] in an appropriate way and why it is unbiased. It has to be ensured that the probability for erroneous classifications is distributed uniformly over the elements of the ontology: it would be unacceptable, for example, if the overall correctness rate for classifications in the PTS system would be very good, but the system would fail systematically for an ontology subset like *"persons in wheel chairs"*. Referring to ODD and ontologies, it has to be demonstrated that *"events with low probability but high severity of failure"* have been taken into account in the training data.

The paradigm of *explainable AI* needs to be applied. During testing, a correct classification result (e.g. *"child in door region"*) must only pass the test if the classification was obtained *for the correct reasons* (e.g. the object has been classified as a child, because of its size and its face, and not just because it carries a doll). The technology for performing such checks is already available [20]. Moreover, it has to be demonstrated that the training data set covers the edge cases of the machine learning algorithm.

Finally, it has to be demonstrated that the perceptor is sufficiently robust, so that it not only works well with the training and the test database.

If it were planned that the AI-based perceptors should continue to learn during system operation, then it has to be demonstrated that this can never affect the system safety. We consider this to be very difficult. Therefore, we advocate to refrain from learning in operation. Instead, neural networks should be re-calibrated with additional data as a maintenance activity, and only become operative again after re-evaluation.

We consider all evaluation Steps (1), (2), (3a,b) to be feasible for the specified ODD. The redundant pairing of conventional and AI-based channels for sensors and perceptors reduces the evaluation task for AI-based perceptors to demonstrating that they are free of *systematic* errors. This can be performed

[13] which in turn needs to cover the ODD

with smaller test and training data sets than would be needed if perceptors based on machine learning were the only ones used.

6 Conclusion

We have presented an architecture for autonomous train controllers and demonstrated how this could be evaluated and certified on the basis of the novel ANSI/UL 4600 pre-standard dedicated to the assurance of autonomous transportation systems. As a main result, it has been shown that such an evaluation is feasible, and, consequently, such systems are certifiable for freight trains, metro trains, and trams. This restriction is necessary because no reliable solutions for obstacle detection in high speed trains seem to be available today.

The architecture presented here has the advantage that most of the system (sub-)components can be verified, validated and certified based on the CENELEC standards applicable today in the railway domain. Only the analysis of hazards induced by autonomy-related functions, their mitigations, and the evaluation of AI-based perceptor sub-functions need to be evaluated according to the ANSI/UL 4600, since these aspects are not addressed in the CENELEC standards.

We consider the 2-channel concept suggested for sensing and perception, where conventional technologies are paired with AI-based ones to be essential for the certifiability of the train control architecture advocated here. This pairing ensures that functionality based on machine learning and neural networks can never represent a single point of failure for any of the modules supporting autonomy. We are aware that the redundant channel design combining conventional and AI-based techology is costly. However, the image collections available for training data in the railway domain are significantly smaller than those available in the automotive domain. As a consequence, the estimation of classification failure rates and the demonstration of unbias is far less trustworthy than in the case of autonomous road vehicles. Therefore, from our perspective, an autonomous train control system exclusively using trained AI-based perceptors is currently not certifiable, and also not advisable from the safety perspective.

References

1. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason. John Wiley&Sons Ltd, West Sussex, England (2007)
2. Bouillaut, L., François, O., Putallaz, Y., Granier, C., Cieux, C.: A hybrid approach for the evaluation of rail monitoring and maintenance strategies for the Grand Paris Express new metro. International Journal of Performability Engineering 16, 1685–1697 (12 2020)
3. CENELEC: EN 50128:2011 Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems (2011)
4. CENELEC: EN 50126 Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Generic RAMS Process (2017)

5. CENELEC: Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling (2018)
6. Eder, K., Huang, W., Peleska, J.: Complete Agent-driven Model-based System Testing for Autonomous Systems – Technical Report (Aug 2021), https://doi.org/10.5281/zenodo.5203111
7. Eder, K.I., Huang, W., Peleska, J.: Complete agent-driven model-based system testing for autonomous systems. In: Farrell, M., Luckcuck, M. (eds.) Proceedings Third Workshop on Formal Methods for Autonomous Systems, FMAS 2021, Virtual, 21st-22nd of October 2021. EPTCS, vol. 348, pp. 54–72 (2021), https://doi.org/10.4204/EPTCS.348.4
8. Erskine, M., Milburn, D.: Digital Train Control Functional Safety for AI Based Systems. In: Proceedings of the International Railway Safety Council Conference, 13-18 October 2019, Perth, Australia. pp. 1–24 (2019), https://international-railway-safety-council.com/digital-train-control-functional-safety-for-ai-based-systems-david-milburn-senior-technical-d
9. Fisher, M., Mascardi, V., Rozier, K.Y., Schlingloff, B.H., Winikoff, M., Yorke-Smith, N.: Towards a framework for certification of reliable autonomous systems. Autonomous Agents and Multi-Agent Systems 35(1), 8 (Dec 2020), https://doi.org/10.1007/s10458-020-09487-2
10. Gleirscher, M., Calinescu, R., Woodcock, J.: Riskstructures: A design algebra for risk-aware machines. Formal Aspects Comput. 33(4-5), 763–802 (2021), https://doi.org/10.1007/s00165-021-00545-4
11. Huang, W., Peleska, J.: Complete model-based equivalence class testing. Software Tools for Technology Transfer 18(3), 265–283 (2016), http://dx.doi.org/10.1007/s10009-014-0356-8
12. ISO: ISO/DIS 21448: Road vehicles — Safety of the intended functionality. European Committee for Electronic Standardization (2021), iCS: 43.040.10, Draft International Standard
13. Koopman, P., Kane, A., Black, J.: Credible autonomy safety argumentation. In: Proceedings of the 27th Safety-Critical Systems Symposium, Feb. 2019. (2019), https://users.ece.cmu.edu/~koopman/pubs/Koopman19_SSS_CredibleSafetyArgumentation.pdf
14. Koopman, P., Wagner, M.: Toward a Framework for Highly Automated Vehicle Safety Validation. In: Proceedings of the 2018 SAE World Congress / SAE 2018-01-1071 (2018), https://users.ece.cmu.edu/~koopman/pubs/koopman18_av_safety_validation.pdf
15. Koopman, P., Wagner, M.D.: Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intell. Transp. Syst. Mag. 9(1), 90–96 (2017), https://doi.org/10.1109/MITS.2016.2583491
16. Marmo, R., Lombardi, L., Gagliardi, N.: Railway sign detection and classification. In: 2006 IEEE Intelligent Transportation Systems Conference. pp. 1358–1363 (2006)
17. Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. Tech. rep., OMG (2017)
18. Ristić-Durrant, D., Franke, M., Michels, K.: A Review of Vision-Based On-Board Obstacle Detection and Distance Estimation in Railways. Sensors (Basel, Switzerland) 21(10), 3452 (May 2021), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8156009/
19. Ritika, S., Mittal, S., Rao, D.: Railway track specific traffic signal selection using deep learning. CoRR abs/1712.06107 (2017), http://arxiv.org/abs/1712.06107

20. Sun, Y., Chockler, H., Huang, X., Kroening, D.: Explaining image classifiers using statistical fault localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII. Lecture Notes in Computer Science, vol. 12373, pp. 391–406. Springer (2020), https://doi.org/10.1007/978-3-030-58604-1_24
21. Trentesaux, D., Dahyot, R., Ouedraogo, A., Arenas, D., Lefebvre, S., Schön, W., Lussier, B., Chéritel, H.: The Autonomous Train. In: 2018 13th Annual Conference on System of Systems Engineering (SoSE). pp. 514–520 (Jun 2018)
22. Underwriters Laboratories Inc.: ANSI/UL 4600-2020 Standard for Evaluation of Autonomous Products – First Edition. Underwriters Laboratories Inc., 333 Pfingsten Road, Northbrook, Illinois 60062-2096, 847.272.8800 (April 2020)
23. UNISIG: Basic System Description, chap. 2. Vol. Subset-026-2 of [24] (February 2006), Issue 2.3.0
24. UNISIG (ed.): ERTMS/ETCS – Class 1 System Requirements Specification, vol. Subset-026 (February 2006), Issue 2.3.0
25. UNISIG: Modes and Transitions, chap. 4. Vol. Subset-026-4 of [24] (February 2006), Issue 2.3.0
26. Withers, J., Stoehr, N.: Automated Train Operations (ATO) Safety and Sensor Development. Technical Report RR 20-21, U.S. Department of Transportation – Federal Railroad Administration (Nov 2020), https://railroads.dot.gov/elibrary/automated-train-operations-ato-safety-and-sensor-development
27. Zhang, Z., Wang, Y., Brand, J., Dahnoun, N.: Real-time obstacle detection based on stereo vision for automotive applications. In: 2012 5th European DSP Education and Research Conference (EDERC). pp. 281–285 (2012)

A Extended Kernel Design

The extended kernel design presented here is a revised and extended version of an initial and more limited design proposed by Eder et al. [6]. The crucial extensions presented here concern

- the controlled change between ODD sub-domains,
- the distinction between service brake and emergency brake,
- introduction of door control,
- introduction of sub-controllers for non-autonomous operation, and
- re-distribution of behaviour across ODD sub-domain controllers.

The informal system description given above is now modelled using UML state machines [17]. The formal model semantics can be specified, for example, by associating a variant of Kripke structures with state machines, as described in [11]. Alternatively, the model can be flattened and transformed into a symbolic finite state machine, whose interpretation is slightly simpler, because SFSMs can be regarded as a simpler sub-class of these Kripke structures. In the following, we explain the behaviour formalised with these machines in an intuitive way.

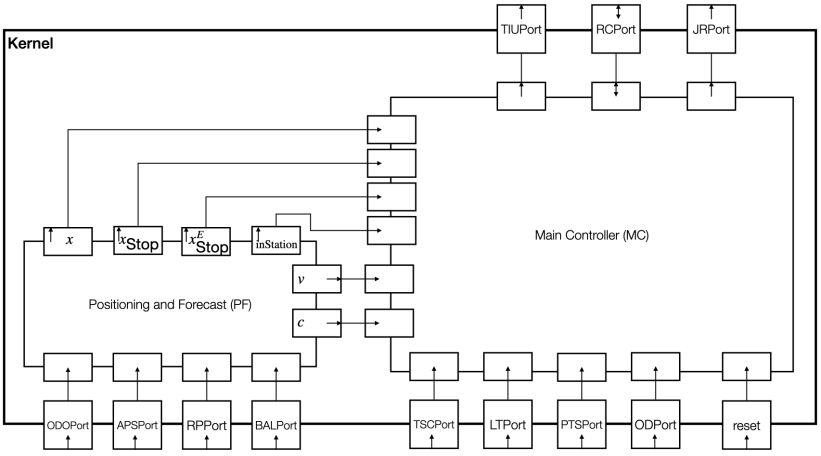

Fig. 4. Kernel software decomposition.

Kernel structure and interfaces The kernel software is structured as shown in Fig. 4. The *Positioning and Forecast (PF)* task aggregates the actual train position from values provided by odometry (ODO), additional positioning subsystems (APS), refined balise (BAL), and positioning (RP). Each port interface p is structured as a pair (x_p, ζ_p), where x_p is the position estimate achieved by

the positioning source (ODO, APS, BAL, or RP), and $\zeta_p \in [0,1]$ is a confidence value, where value zero stands for *"no estimate available"*.

From these inputs, the positioning and forecast task determines

- a position estimate $x \in \mathbb{R}_{\geq 0}$,
- a confidence value $c \in [0,1]$ for x,
- a speed estimate $v \in [-v_{\mathrm{Max}}, v_{\mathrm{Max}}]$,
- estimates $x_{\mathrm{Stop}} \in \mathbb{R}_{\geq 0}$ and $x^E_{\mathrm{Stop}} \in \mathbb{R}_{\geq 0}$ for the position where the train would come to a standstill if the service brakes and the emergency brakes would be applied in the next processing cycle[14], respectively, and
- a flag inStation $\in \mathbb{B}$ indicating whether the train in its current position is inside a station.[15]

For interface failure indication on the ports described below, we define enumeration type

$$\texttt{Faults} = \{\texttt{OK}, \texttt{transientFault}, \texttt{permanentFault}\}$$

The *main controller (MC)* task receives inputs on the following interfaces.

- Reset information from the reset port $\mathsf{reset} : \mathbb{B}$.
- Obstacle detection information from ODPort, structured as tuples

$$(\texttt{obs}, x_{\texttt{obs}}, \texttt{fail}) \in \mathbb{B} \times \mathbb{R}_{\geq 0} \times \texttt{Faults}.$$

  If $\texttt{fail} = \texttt{OK}$, obstacle detection is indicated by $\texttt{obs} = \texttt{true}$, and $x_{\texttt{obs}}$ is the estimate for the distance from train to obstacle.
- Passenger transfer supervision information on port PTSPort, structured as tuple lists

$$(d_1, p_1, \texttt{fail}_1) \ldots (d_k, p_k, \texttt{fail}_k) \in (\texttt{Location} \times \mathbb{B} \times \texttt{Faults})^*,$$

  where d_i denotes locations where passengers need to be detected (doors and other positions dangerously close to the train), Boolean p_i indicates whether passengers have been detected at the specified location, and $\texttt{fail}_i$ indicates the sensor/perceptor status at the specified location.
  For freight trains, the PTS module and the associated port are removed from the model described here.
- Line transmission information on port LTPort, structured as tuples

$$(x_{\texttt{sig}}, \texttt{aspect}, \texttt{fail}) \in \mathbb{R}_{\geq 0} \times \texttt{Aspects} \times \texttt{Faults},$$

  where Aspects is an abstraction of the possible signal aspects. Since there are very many signal types and associated aspects used in Europe, we abstract

[14] We assume a cyclic execution of the PF and MC tasks on a single-core system.

[15] This information is extracted from configuration data (typically sent by the RBC). We omit the information about whether the platform is on the left-hand side or right-hand side and let the train controller simply release doors for opening and close doors without side distinctions.

here from the possible values in this type. Instead, we assume the existence of functions

$$\texttt{aspectToMaxSpeed} : \texttt{Aspects} \longrightarrow [0, v_{\text{Max}}] \cup \{\texttt{noEffect}\}$$
$$\texttt{aspectToEoA} : \texttt{Aspects} \longrightarrow \mathbb{R}_{\geq 0} \cup \{\texttt{noEffect}\}$$

mapping aspect values to their effect (if any) on speed and end of movement authority.

- Train signal classification information on port TSCPort, typed just as LTPort.
- Position and forecast information

$$(x, x_{\text{Stop}}, x^E_{\text{Stop}}, v, c) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times [-v_{\text{Max}}, v_{\text{Max}}] \times [0, 1]$$

from task PF.

- From the radio communication module, the main controller receives inputs on port TCPort typed as tuples

$$(\texttt{EoA}, v_{\text{Ceil}}, \texttt{rc}, \texttt{fail}) \in \mathbb{R}_{\geq 0} \times [0, v_{\text{Max}}] \times \mathbb{B} \times \texttt{Faults},$$

where EoA denotes the current end of movement authority, and v_{Ceil} denotes the admissible ceiling speed in the current track section.[16] Boolean flag rc indicates whether remote control is available. As usual, fail indicates the status of the radio communication module.

The main controller task writes to the following output ports.

- Train interface data on port TIUPort, structured by pairs

$$(a, d) \in \{a^E_-, a_-, a_0, a_+\} \times \{\texttt{release}, \texttt{close}\}$$

The first component a abstracts the interaction with service brake, emergency brake, and engine by means of acceleration values: $a = a^E_-$ is the negative acceleration of the train to be achieved when activating the emergency brakes. $a = a_-$ is the negative acceleration to be achieved when activating the service brakes. $a = a_0$ (no acceleration) is achieved by by releasing all brakes and keeping the speed constant. $a = a_+$ is achieved by releasing all brakes and accelerating the train (we abstract from different positive acceleration values).
The second component controls the doors: $d =$ release releases the door locks, so that passengers may open the doors. $d =$ close closes the doors and locks them (typically by keeping the air pressure high).

- In the output direction of the radio communication port RCPort, the train transmits its current position estimate $x \in \mathbb{R}_{\geq 0}$ to RBC/IXL.
- On the interface to the juridical recording system (port JRPort), the main controller transmits all safety-relevant decisions (start, brake, stop the train), together with the causes leading to these decisions. The associated behaviour is outside the scope of this technical report.

[16] We abstract here from more realistic implementations, where the RBC transmits lists of (location, ceiling speed) pairs, so that the train controller can extract the applicable ceiling speed from this list, using its actual position value.

Main Controller The control of transitions between ODD sub-domains is modelled by the state machine shown in Fig. 5.

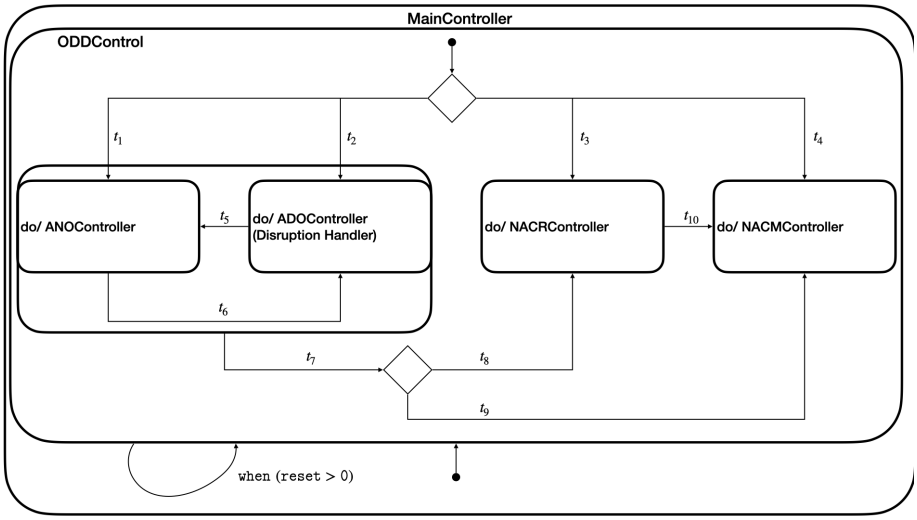

Fig. 5. Transitions between ODD sub-domains.

The autonomous operational modes (see also Fig. 1) are controlled by sub-machines ANOController and ADOController. The latter implements the disruption handler introduced in Section 4. When falling back to non-autonomous operation, the behaviour is controlled by sub-machines NACRController (if remote control is available) and NACMController (if manual control by a train engine driver is required).

The NACRController processes the remote control commands sent from the control centre to the train via radio communication. Moreover, it activates the video camera streaming from the train to the control centre that is required for remote supervision of the track. The NACMController enables manual train control by a train engine driver through a conventional dashboard and provides the display information at the man-machine interface. These sub-machines for manual, non-autonomous operation are outside the scope of this report and will not be discussed further.

When having entered one of the two non-autonomous ODD sub-domains, there are no direct transitions leading back to autonomous operation: the autonomous mode needs to be reinstated manually by resetting the kernel controller, as indicated by the reset-selfloop of state ODDControl.

After each reset, the main controller enters the appropriate operational mode as specified by the guard conditions in transitions t_1, t_2, t_3, t_4.

$$\begin{aligned} t_1 \equiv [&c > c_{\text{Min}} \wedge \mathsf{RCPort.fail} = \mathtt{OK} \wedge (\mathsf{ODPort.fail} = \mathtt{OK} \wedge \neg \mathsf{ODPort.obs}) \wedge \\ &(\neg \text{inStation} \vee \mathsf{PTSPort.fail} = \mathtt{OK}) \wedge \\ &\mathtt{OK} \in \{\mathsf{LTPort.fail}, \mathsf{TSCPort.fail}\}] \end{aligned}$$

The guard of t_1 contains all conditions to enter full autonomous mode; these conditions are

- sufficient confidence in the actual train location,
- fully functional radio communication and obstacle detection and no obstacle present,
- fully functional passenger transfer supervision, if the train is in a station, and
- fully functional line transmission (LT) or train signal classification (TSC).

Transition t_2 enters the degraded autonomous mode; its guard is specified as follows.

$$\begin{aligned} t_2 \equiv [&\mathtt{permanentFault} \notin \{\mathsf{RCPort.fail}, \mathsf{ODPort.fail}\} \wedge \\ &\big(0 < c \leq c_{\text{Min}} \vee \\ &(\mathsf{RCPort.fail} = \mathtt{transientFault} \vee \mathsf{RCPort.obs}) \vee \\ &\mathsf{ODPort.fail} = \mathtt{transientFault} \vee \\ &(\text{inStation} \wedge \mathsf{PTSPort.fail} = \mathtt{transientFault}) \vee \\ &(\mathtt{OK} \notin \{\mathsf{LTPort.fail}, \mathsf{TSCPort.fail}\} \wedge \mathtt{transientFault} \in \{\mathsf{LTPort.fail}, \mathsf{TSCPort.fail}\})\big)] \end{aligned}$$

As specified in this guard condition, the degraded mode will be entered if there is insufficient confidence in the current position, or if any of the essential subsystems have a non-permanent fault, from which the associated sensor/perceptor sub-system can recover.

Transition t_3 applies if neither the guard of t_1, nor that of t_2 is fulfilled, but remote control is available (flag $\mathsf{RCPort.rc} = \mathtt{true}$). Otherwise transition t_4 applies. When already in one of the autonomous operational modes, subsystem failures captured by the guard of t_3 lead to a change event triggering compound transition $t_7; t_8$. Failures captured by the guard of t_4 lead now to change events triggering compound transition $t_7; t_9$. When the train is in non-autonomous remote control mode, failure of the communication line or of the video streaming connection to the remote control centre trigger transition t_{10}, so that the non-autonomous manual mode is entered.

When entering the non-autonomous modes, the train is always brought to a stop, before non-autonomous train movement can commence. This also applies when the remotely controlled mode is left, and the manual mode is entered.

When in fully functional autonomous mode controlled by sub-machine ANOController, non-permanent failures or insufficient position confidence trigger a change event with the same condition as the guard of t_2. This triggers transition t_6 and

leads to degraded autonomous operation controlled by the disruption handler. From there, full autonomous mode can be reached again via transition t_5, if the respective sub-systems have recovered from transient faults and sufficient position confidence is available. This transition is triggered by a change event with the same condition as the guard of t_1.

ANO Controller. The behaviour of the control sub-task for autonomous normal operation is structured by a hierarchic state machine whose top-level layer is shown in Fig. 6. Initially, the maschine branches according to the conditions "*movement authority available* $[\texttt{EoA} - \texttt{x} > \alpha]$*/not available* $[\texttt{EoA} - \texttt{x} \leq \alpha]$". The guard conditions evaluate the current value `EoA` of the end-of-MA received from the interlocking on port RCPort, the current location estimate x received from the positioning and forecast task, and a small constant value $\alpha > 0$: condition $\texttt{EoA} - \texttt{x} \leq \alpha$ evaluates to `true` if the train is so close to end-of-MA (or even has overrun the end-of-MA) that it should stop or remain stopped, until a new MA is provided. Conversely, $\texttt{EoA} - \texttt{x} > \alpha$ means that the train still has to move to reach the `EoA` location.

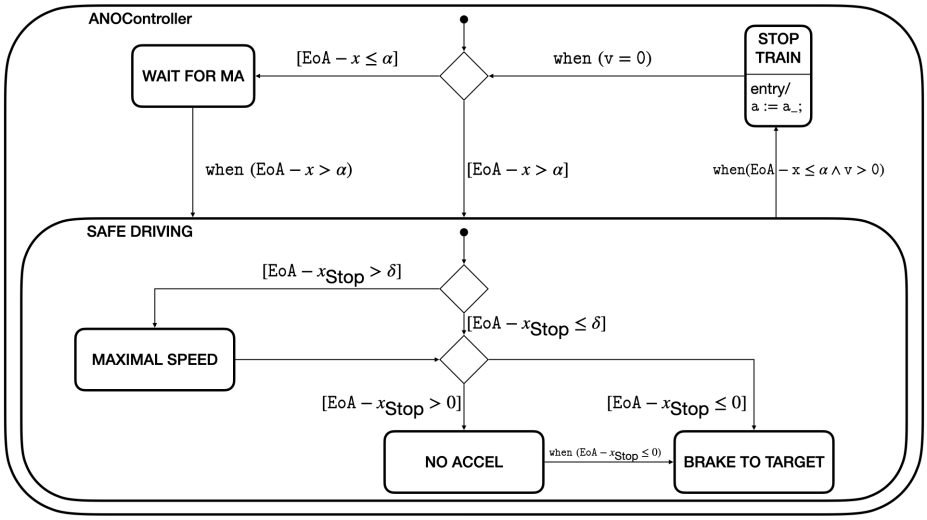

Fig. 6. ANOController behaviour – top-level state machine.

In state WAIT FOR MA, the train is braked to a stop (if not already halted), as specified in sub-machine in Fig. 7.

If a movement authority is available and the train is still far enough from its destination ($\texttt{EoA} - x_{\text{Stop}} > \delta$), the controller branches into submachine SAFE DRIVING.MAXIMAL SPEED. There, the train will be accelerated to its maximal

speed v_{Max}, as shown in state machine Fig 8. The train will be slowed down if it is too fast and accelerated if it is slower than the maximal speed allowed.

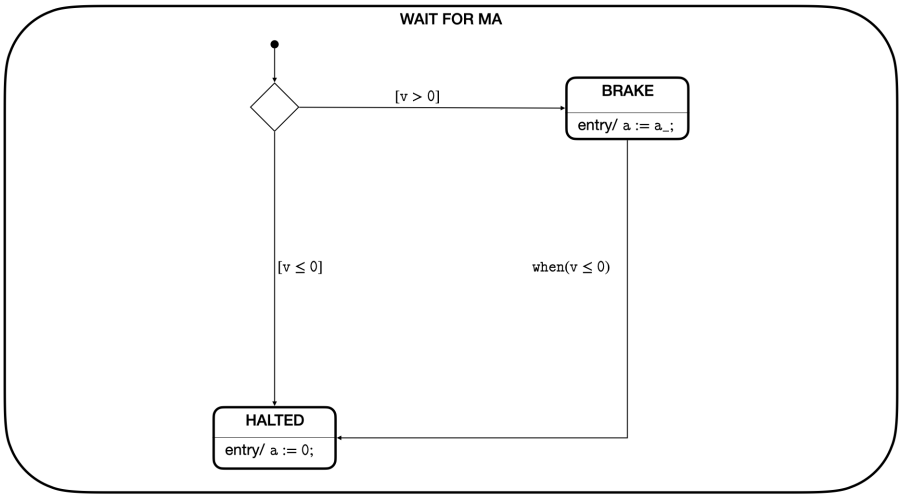

Fig. 7. Train controller behaviour – sub-machine of state WAIT FOR MA.

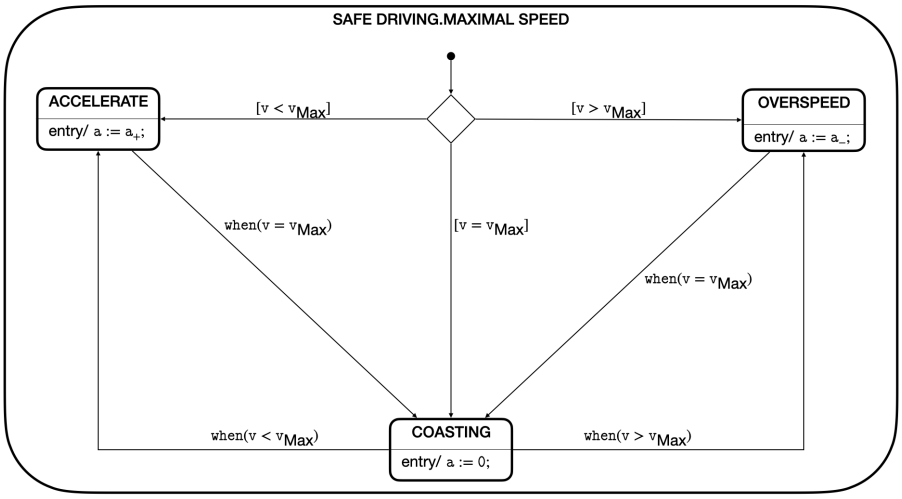

Fig. 8. Train controller behaviour – safe driving with maximal speed.

If the predicted stopping location comes as close as δ to `EoA` (change condition $\texttt{EoA} - x_{\text{Stop}} \in (0, \delta]$), the train is not allowed to accelerate further. It will be kept at a constant velocity $v_{\text{const}} \leq v_{\text{Max}}$ in state NO ACCEL (see sub-machine in Fig. 9).

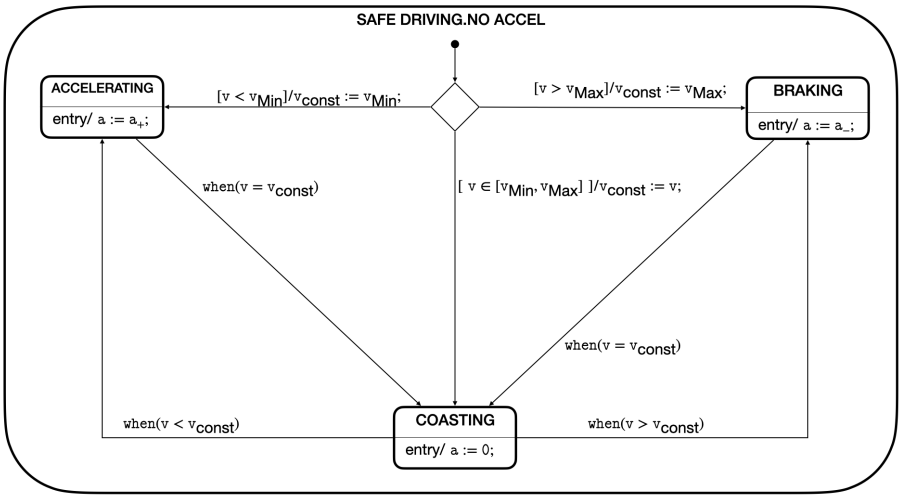

Fig. 9. Train controller behaviour – sub-machine of state NO ACCEL.

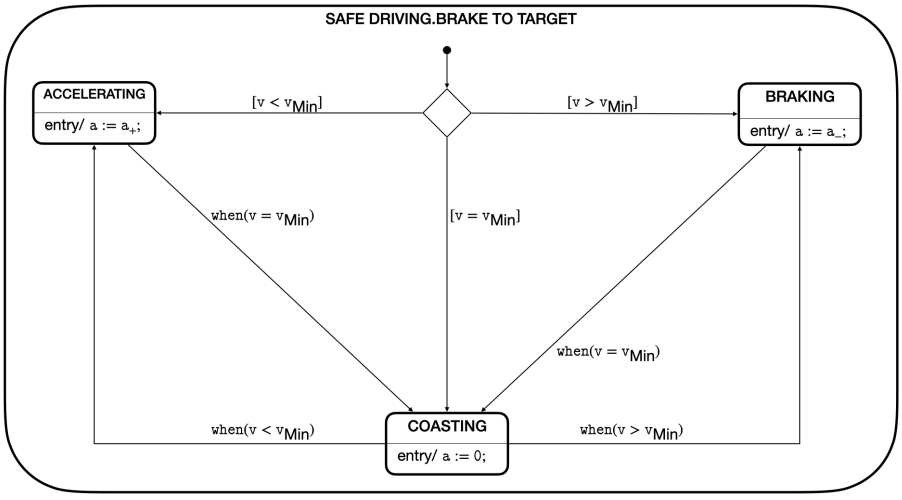

Fig. 10. Train controller behaviour – braking to target destination.

When it is time to brake ($\texttt{EoA} - x_{\text{Stop}} \leq 0$), state BRAKE TO TARGET is entered (Fig. 10), where the train is slowed down to a positive speed value v_{Min}.

This positive speed value is maintained until the train is very close to its destination ($\texttt{EoA} - x \leq \alpha$). Then state STOP TRAIN is entered, where the train is slowed down to a halt.

The train automatically loses its movement authority when coming close to its destination ($\texttt{EoA} - x \leq \alpha$). Therefore, after having come to a standstill, the control will transit from STOP TRAIN to WAIT FOR MA where it stays until a new movement authority arrives.

The detection of signals via LT or TSC is always transformed into a change of the current v_{Ceil} value and the EoA value, respectively. This is performed by a sub-task of the main controller which is not detailed here.

ADO Controller. The disruption handler ADOController is activated when the controller starts and the guard condition of transition t_2 applies. Otherwise, it is triggered by transition t_6 in Fig. 5 for various reasons. Depending on these reasons, the disruption handler branches into a library of sub-machines.

For example, on detection of an obstacle, the train is stopped using the service brakes if the obstacle is sufficiently far away ($x_{\texttt{obs}} > x_{\text{Stop}}$); otherwise, the emergency brakes are used. If the obstacle is removed while still braking, the controller can return into sub-domain ANO which ends the braking procedure, if the train still has movement authority and has not yet reached its EoA.

As a second example, the disruption handler is activated when the position confidence is low. In this case, the train is slowed down to a safe velocity, where the train can easily brake for obstacles occurring "unexpectedly", since the train's position is not quite clear. A return to the ANOController is performed as soon as the position confidence values are sufficiently high again.