Tagungsband

Scientific Railway Signalling Symposium 2019

Mehr Verkehr auf die Schiene durch Digitalisierung?! – Was kann die Leit- und Sicherungstechnik dazu beitragen?

Herausgeber
Prof. Dr.-Ing. Andreas Oetting

26.06.2019

TECHNISCHE UNIVERSITÄT DARMSTADT

Institut für Bahnsysteme und Bahntechnik

Vorwort Tagungsband SRSS 2019

Prof. Dr.-Ing. Andreas Oetting,

Leiter des Instituts für Bahnsysteme und Bahntechnik an der Technischen Universität Darmstadt

Liebe Mitglieder der Fachcommunity und Bahn-Interessierte,

zum dritten Mal durften wir am 26. Juni 2019 ca. 110 Fachexpertinnen und Fachexperten aus dem Bereich der Leit- und Sicherungstechnik im Gästehaus der TU Darmstadt begrüßen, um über die Zukunft der Sicherungstechnik unter den Vorzeichen von Digitalisierung und Industrie 4.0 zu sprechen und den Horizont mit spannenden Einblicken aus Forschung und Praxis zu bereichern.

Das Motto des diesjährigen Scientific Railway Signalling Symposiums lautete „Mehr Verkehr auf die Schiene durch Digitalisierung?! – Was kann die LST dazu beitragen?“. In diesem Sinne diskutierten die Teilnehmerinnen und Teilnehmer in sieben Arbeitsgruppen, welchen Beitrag aktuelle und zukünftige Ansätze der digitalen Bahn zur Kapazitätsgewinnung bringen können und welche Knackpunkte es auf dem Weg dorthin noch zu lösen gilt. Daran anschließend brachte Herr Messerli von der SBB die Anwesenden auf den neusten Stand der Entwicklungen in der Schweiz. Dort geht das Branchenprogramm „smartRail 4.0“ nun in den Endspurt der Konzeptphase. Ein bereits konkret untersuchtes Anwendungsszenario für Kapazitätssteigerungen durch Maßnahmen im Bereich der LST bietet die S-Bahn Stuttgart. Dort wurden verschiedene zukünftige LST-Ausrüstungsszenarien in Hinblick auf ihr Potential zur Verringerung von Verspätungsminuten untersucht und die Ergebnisse auf dem SRSS vorgestellt. Die Untersuchung ergab signifikante Unterschiede zwischen den untersuchten Varianten. Die bevorzugte Variante „ETCSe mit ATO-Light“ unterstreicht zudem, dass der größte Nutzen nur durch die Kombination mehrerer technischer Innovationen umgesetzt werden kann. Weitere thematische Inputs lieferten Einblicke in die Themen IT-Security und „ATO GoA2 over ETCS-L2-FS“.

Zusätzlich zu diesen Beiträgen aus der Industrie versteht sich das SRSS auch als Plattform für wissenschaftliche Vorträge, die einen Einblick in die gegenwärtige Forschung im Bereich der LST bieten. So bereicherten drei wissenschaftliche Beiträge zu den Themen „Ortung“, „automatisierte Planung von ETCS-Infrastruktur“ sowie „Sicherungslogik im Stellwerkskern“ das SRSS. Diese Beiträge bestehen neben dem Vortrag aus einem Aufsatz, der zuvor einen wissenschaftlichen Review-Prozess durchlief. Mit diesem Tagungsband möchten wir Ihnen diese interessanten Aufsätze zur Verfügung stellen und hoffen, dass sie hilfreiche Impulse für Ihre weitere Arbeit bieten.

Eine anregende Lektüre der vorgenannten Arbeiten wünscht Ihnen

Prof. Dr. Andreas Oetting

Modellierung generischer sicherungstechnischer Prüfprozesse unter Ausnutzung aktueller Informationen zum Betriebsgeschehen

Frederik Düpmeier[1]

[1] Institut für Bahnsysteme und Bahntechnik, Technische Universität Darmstadt

1 Einleitung

Verspätungen prägen derzeit die öffentliche Wahrnehmung des Verkehrsträgers Schiene und werden breit diskutiert. Ursache sind unter anderem Kapazitätsengpässe, insbesondere in den hochbelasteten zentralen Knoten und auf verkehrsreichen Korridoren. Neben punktuellen Ausbaumaßnahmen werden Maßnahmen zur „Digitalisierung“ der Eisenbahn (vgl. z. B. Programm „Digitale Schiene“ [1]) als Abhilfe diskutiert. Dabei liegt der Fokus insbesondere auf dem europäischen Zugsicherungssystem ETCS, dem automatisierten Fahren (ATO) und standardisierten Schnittstellen zwischen verschiedenen Komponenten der Sicherungstechnik. Letzteres wird derzeit im Rahmen der europäischen Initiative EULYNX und der sogenannten „digitalen Stellwerke“ (DSTW) angegangen [ebd.]. Der klassische „Stellwerkskern“, also die Sicherungslogik im zentralen System der infrastrukturseitigen Eisenbahnsicherungstechnik, in diesem Kontext häufig auch als Stellwerkslogik bezeichnet, bleibt davon allerdings unberührt. Über die Jahrzehnte haben sich zwar die Umsetzungsformen dieser Sicherungslogik weg von mechanischen Abhängigkeiten im Verschlussregister über Relais hin zu programmierten Verschlusstabellen oder spurplanbasierten Logiken in modernen elektronischen Stellwerken (ESTW) weiterentwickelt, die zugrundeliegenden sicherungstechnischen Prinzipien blieben im Kern aber unverändert. Die Kapazitätspotentiale von ETCS und ATO können allerdings nur vollständig genutzt werden, wenn auch die Sicherungslogik als zentrale Komponente der Eisenbahnsicherungstechnik den neuen technischen Möglichkeiten angepasst wird (vgl. [3]). Denn häufig kann die Infrastruktur nicht optimal ausgelastet werden, weil gerade die Sicherungslogik dies nicht zulässt. Hintergrund sind zum Beispiel fehlende (gesicherte) Informationen, z.B. über die Position des Zuges oder dessen Eigenschaften bzw. über die tatsächliche Betriebslage. In solchen Fällen geht das heutige Stellwerk im Sinne größtmöglicher Sicherheit in der Regel vom schlechtesten Fall (z. B. schlechtes Bremsvermögen eines schweren Güterzuges – auch wenn es sich tatsächlich um eine schnellbremsende S-Bahn handelt) aus. Mittlerweile sind jedoch in vielen Fällen deutlich präzisere Informationen vorhanden, die – solange sie vorhanden und valide sind – auch genutzt werden können und sollten.

Im Projekt smartLogic innerhalb der Innovationsallianz von Deutscher Bahn AG und TU Darmstadt wird daher eine innovative Sicherungslogik erarbeitet, die, abhängig von der Verfügbarkeit sicherungstechnisch verwertbarer Informationen zum aktuellen Betriebsgeschehen und dem Grad der Verlässlichkeit dieser Informationen, eine größtmögliche Flexibilität bei der Ausgestaltung von Fahrerlaubnissen (MAs) und Stellbefehlen ermöglicht. Dabei stehen die Ziele einer optimaleren Ausnutzung der Infrastruktur und eines robusteren Bahnbetriebs, z. B. bei Verspätungen, im Vordergrund.

[1] Korrespondierender Autor: Frederik Düpmeier, duepmeier@verkehr.tu-darmstadt.de

In diesem Paper wird vorgestellt, wie die einzelnen Prüfprozesse der smartLogic aufgebaut sind und wie sie modelliert werden.

Zuvor erläutert Kapitel 2 die Hintergründe zur smartLogic. Dabei wird insbesondere auf die Aufgabe der Sicherungslogik in der Eisenbahnsicherungstechnik und die Besonderheiten der smarten Sicherungslogik eingegangen. Kapitel 3 beschreibt die Methodik bei der Modellierung der Sicherungslogik. Diese wird in Kapitel 4 anhand der Beispiel-Prozessfunktion „Kürzen einer Fahrerlaubnis“ veranschaulicht. Kapitel 6 schließt mit Fazit und Ausblick.

2 Die Aufgabe der Sicherungslogik und die Besonderheiten der smarten Sicherungslogik

In diesem Kapitel wird die Aufgabe der Sicherungslogik im Gesamtkontext der zukünftigen eisenbahnsicherungstechnischen Komponenten erläutert. Des Weiteren wird der Begriff der „smarten“ Sicherungslogik von anderen Sicherungslogik-Konzepten abgegrenzt. Da in diesem Paper der Fokus auf den Aspekt der Modellierung der Prüfprozesse und das Anwendungsbeispiel zum Flankenschutz gelegt wird, sind die Erläuterungen in diesem Kapitel kurz gehalten. Ausführlichere Informationen finden sich in [2] und [3].

Abb. 1 zeigt die (smarte) Sicherungslogik in ihrer Systemumgebung (vgl. dazu die Erläuterungen in [2]). Ein intelligentes Traffic Management System (TMS) (dunkelroter Bereich)[2] errechnet[3] Trassenslots für Eisenbahnfahrten[4]. Aus diesen resultieren spezifische Stellanforderungen an die Infrastrukturelemente und Fahrerlaubnis-Anfragen (MA-Requests) für die Triebfahrzeuge. Aufgabe der Sicherungslogik ist es, diese Anforderungen des TMS, welches sich komplett im nicht sicherheitskritischen Bereich (SIL-0-Bereich) befindet, daraufhin zu überprüfen, ob sie zu einem unsicheren Zustand[5] führen. Dies beurteilt die smarte Sicherungslogik anhand der aktuellen Betriebslage und mit allen mit hinreichender Sicherheit[6] zur Verfügung stehenden Informationen. Die Informationen stammen dabei aus sicheren Datenquellen (grauer Bereich)[7], die nicht Teil der Sicherungslogik sind, aber dennoch zum sicherheitskritischen Bereich gehören. Die genaue Ausgestaltung der Datenquellen ist nicht Teil des Projekts smartLogic, ihre Existenz wird aber vorausgesetzt. Führt die Anfrage des TMS nicht zu einem unsicheren Zustand, wird sie über standardisierte Schnittstellen (Fahrerlaubnisse (engl. Movement Authority (MA)) über ETCS zum Fahrzeug und Stellbefehle über EULYNX-Schnittstellen zur Infrastruktur)[8] weitergereicht. Die Sicherungslogik erfüllt demnach eine Art Wächterfunktion zwischen dem TMS und den Feldelementen.

[2] welches ungleich dem Bedienplatzsystem ist (gelber Bereich)

[3] in verschiedener Granularität und mehrfacher Verfeinerung durch verschiedene Subsysteme

[4] Dies können theoretisch sowohl Zug- als auch Rangierfahrten sein.

[5] Der Zustand ist dann unsicher, wenn in ihm gemäß Gefährdungsbeurteilung die zulässige Grenzwahrscheinlichkeit für das Eintreten eines gefährlichen Ereignisses, die im für den Eisenbahnbereich vorgeschriebenen SIL-4-Bereich definiert ist, überschritten wird.

[6] Vgl. vorherige Fußnote

[7] Von der Existenz solcher sicheren Datenquellen wird im Projekt grundsätzlich ausgegangen. Es ist nicht Teil des Projekts diese näher zu beschreiben. Die smartLogic ist immer auch für den Fall gerüstet, dass Daten nicht zur Verfügung stehen. Dies wird ggf. bei der Berechnung der Schutzrate (siehe unten) entsprechend berücksichtigt.

[8] Zur Vereinfachung wird hier nur von Fahrerlaubnissen und Stellbefehlen gesprochen. Es gibt aber noch weitere Kommunikation, die über die Sicherungslogik läuft.

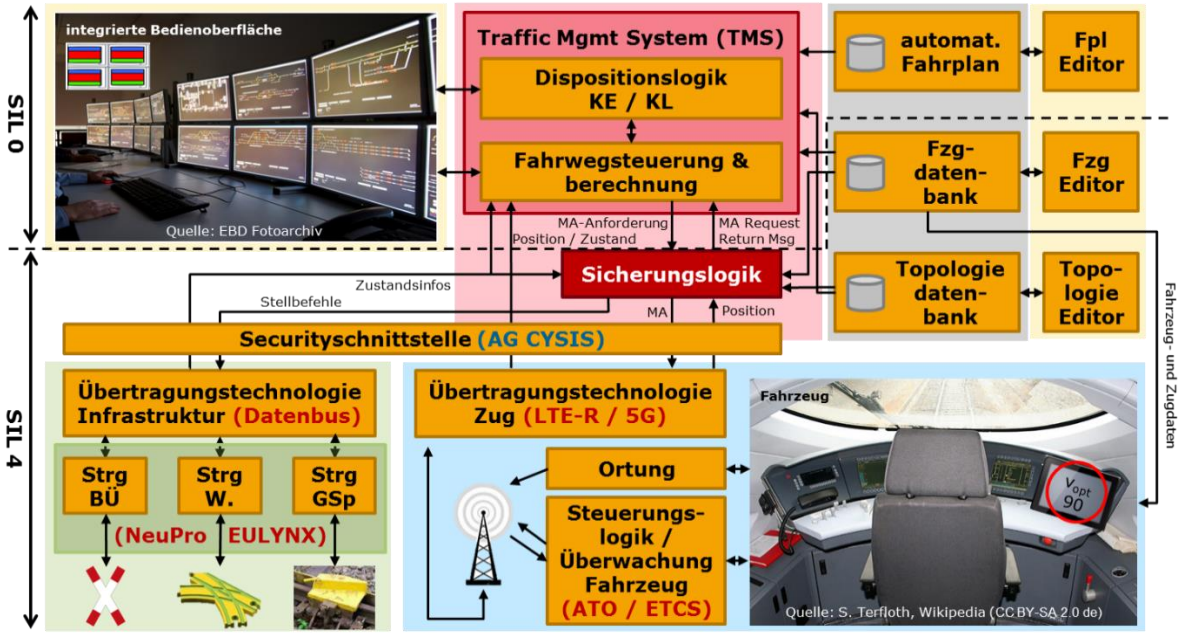

Abb. 1 Systemumgebung der Sicherungslogik
Quelle: [2]

Würden die Fahrerlaubnisse und Stellbefehle direkt vom TMS ausgehen, müsste für das TMS ebenfalls die Sicherheit nachgewiesen werden. Das TMS wäre dann im SIL-4-Bereich anzusiedeln. In diesem Fall wäre eine Neu-Zulassung bei jeder Änderung des TMS notwendig. Um die Zulassungsverfahren zu vereinfachen und Änderungen am TMS schneller umsetzen zu können, ist eine strikte Trennung in SIL-0-Komponente TMS und SIL-4-Komponente Sicherungslogik sinnvoll. Dies entspricht auch der Forderung von EN 50128 nach einer möglichst geringen Komplexität der sicherheitsrelevanten Software [5].

Abgrenzung der „smarten“ Sicherungslogik von anderen Sicherungslogik-Konzepten

Die Sicherungslogik ist – auch wenn das Wort bisher nicht sehr gebräuchlich ist – keine neue Erfindung, sondern bereits in mechanischen Stellwerken – dort häufig als Stellwerkslogik[9] bezeichnet – in Form der Verschlusslogik und von Blockabhängigkeiten zu finden. Bei mechanischem Stellwerken, wie auch in elektromechanischen Stellwerken und in Relaisstellwerken älterer oder bewusst einfach-gehaltener Bauform, herrscht das Prinzip der Fahrstraßenlogik vor[10]. Das bedeutet, die sicherungstechnischen Abhängigkeiten sind in Fahrstraßentabellen und Verschlussplänen in der Form vorgegeben, dass für jede einstellbare Fahrstraße angegeben ist, in welcher Lage sich die an der Fahrstraße beteiligten Elemente befinden müssen und welche Fahrstraßen sich gegenseitig ausschließen. Beim Spurplanprinzip neuerer Relaisstellwerke und einiger ESTW sucht sich das Stellwerk die Fahrstraße mittels einer Suche (beim Relaisstellwerk durch Such- und Echoströme) über die standardisierten Fahrstraßenelemente selbst zusammen [4]. Allerdings sind auch bei diesem Verfahren die Freiheitsgrade begrenzt. Fahrstraßen bilden sich vom vorgegebenen festen Start- zum Zielsignal immer auf die gleiche Weise. Nur bestimmte, vorprojektierte Abweichungen existieren, wie verkürzte

[9] Der Begriff „Stellwerkslogik“ wird hier nicht verwendet, da der Begriff „Stellwerk“ in seiner klassischen Bedeutung nicht mehr klar zu den Komponenten im in Abb. 1 dargestellte Ökosystem zugeordnet werden kann.

[10] Diese Stellwerke werden daher in Abgrenzung zum „Spurplanstellwerk“ häufig auch als „Fahrstraßenstellwerk“ bezeichnet.

Durchrutschwege für bestimmte Einfahrgeschwindigkeiten und das aktuelle Betriebsgeschehen wird nur vereinzelt, z. B. bei Zwieschutzweichen berücksichtigt. Ist eine Fahrstraße festgelegt, lässt sie sich ohne Hilfshandlung nicht mehr verändern.

Zahlreiche Funktionen von ETCS können weder mit Fahrstraßenstellwerken noch mit bestehenden Spurplanstellwerken (in Relais- oder ESTW-Bauart) nicht genutzt werden, beispielsweise

- das Nutzen von vollüberwachten Fahrzeugen zur Gewährung des Flankenschutz[11],
- das Übermitteln von Fahrprofilen von beliebigen zu beliebigen Punkten im Schienennetz mit gefahrpunktorientierten Geschwindigkeitsprofilen[12],
- das dynamische Anpassen des Durchrutschweges je nach Restgeschwindigkeit des Fahrzeugs und
- das dynamische Umbauen der Fahrstraße in Rücksprache mit dem Fahrzeug bei geänderter Betriebssituation[13].

Im Projekt smartLogic wird daher eine neue Sicherungslogik entwickelt, die von der tatsächlich in der aktuellen Betriebssituation vorhandenen Gefährdungslage ausgeht. Es werden nicht mehr feste Voraussetzungen für die Zugfahrt geprüft, sondern ob eine Zustandsänderung durch einen Stellbefehl oder eine Fahrerlaubnis mit hinreichend großer Wahrscheinlichkeit zu keiner Gefährdung führt. Dabei werden alle verfügbaren Informationen, wie aktuelle Fahrzeugpositionen und -geschwindigkeiten, miteinbezogen.

Abb. 2 beschreibt die Entwicklungsschritte der smartLogic. Da die neue Sicherungslogik bewusst zunächst unabhängig von der bestehenden Sicherungstechnik entwickelt werden sollte, wurde zunächst eine ausführliche Gefährdungsanalyse durchgeführt [2]. Daraus wurden in einer Funktionsanalyse Prozessfunktionen, Unterfunktionen und Prüfbedingungen abgeleitet (vgl. [3]). Diese werden formal modelliert und bilden die Basis-Logik der smartLogic. Anschließend folgt eine Testphase mit Hilfe eines Prototyps im Eisenbahnbetriebsfeld Darmstadt. Das nachfolgende Kapitel beschreibt die Basis-Logik-Entwicklung in Form der Modellierung der in der Funktionsanalyse identifizierten Funktionen.

[11] ETCS kann ein Fahrzeug nach Anforderung auf den Stillstand hin sicher überwachen.

[12] Heute gelten Geschwindigkeitseinschränkungen in der Regel ab dem deckenden Signal. Mit ETCS kann dagegen der genaue Gefahrpunkt (z. B. eine langsam zu befahrende Weiche) übermittelt werden. Das Fahrzeug bestimmt dann selber seinen sicheren Bremsweg auf diesen Gefahrpunkt und die Zielgeschwindigkeit dort.

[13] Mittels ETCS kann eine bereits erteilte Fahrerlaubnis vom Zug nach Anforderung durch die Infrastrukturseite wieder zurückgegeben oder gekürzt werden, wenn sich das Fahrzeug sicher ist, dass es vor dem neuen Gefahrpunkt sicher zum Stillstand kommen bzw. die Zielgeschwindigkeit erreichen kann.

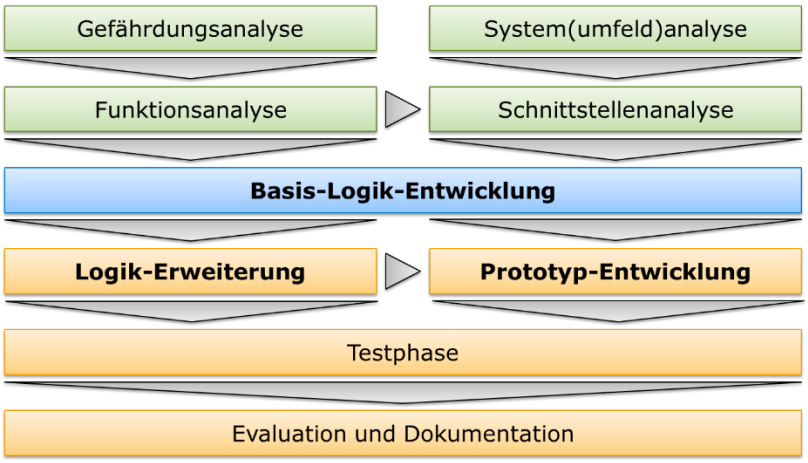

Abb. 2 Entwicklungsschritte der smartLogic
[Eigene Darstellung]

3 Methode bei der Modellierung der Basis-Logik

Die Basis-Logik der smartLogic umfasst die grundsätzlichen Prüfprozesse der Sicherungslogik, um zu beurteilen, ob eine Fahrerlaubnis-Anfrage oder ein Stellbefehl zu einem unsicheren Zustand führt, sowie die Prozesse zur Reaktion auf eine Benutzer-Eingabe oder ein anderes unerwartetes Ereignis, welches über eine der bereitgehaltenen Schnittstellen an die Sicherungslogik gemeldet wird. In diesem Kapitel wird die Methode, mit der die Basis-Logik im Projekt smartLogic modelliert wird, vorgestellt.

3.1 Prozessfunktionen, Unterfunktionen und Prüfbedingungen

smartLogic realisiert die erforderlichen Prüfprozesse mit einem Set von Prozessfunktionen und Unterfunktionen, die in einer Funktionsanalyse bestimmt wurden (vgl. [3]). Abb. 3 verdeutlicht die nachfolgend vorgestellten Begriffe.

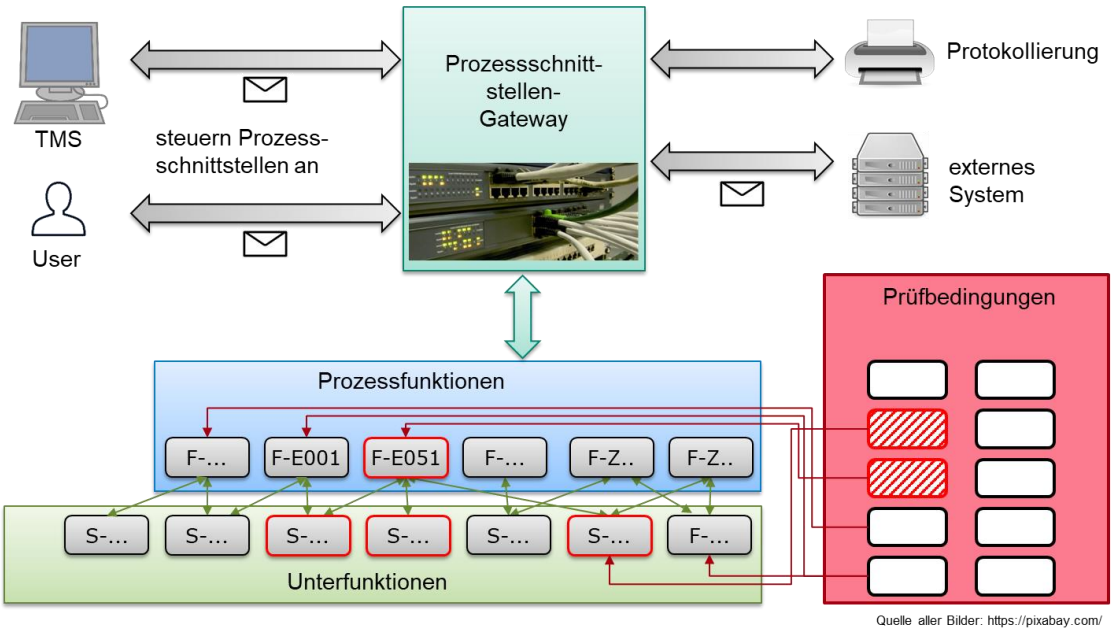

Abb. 3 Übersicht Prozessfunktionen, Unterfunktionen und Prüfbedingungen
[Eigene Darstellung]

Einer Prozessfunktion liegt jeweils ein von einem externen System kommender Input zugrunde, der über das Prozessschnittstellen-Gateway eingeht. In der Regel sind dies Prüf-Anfragen, z. B. zur Prüfung einer Fahrerlaubnis oder eines Stellbefehls, die vom TMS ausgehen. Fällt die Prüfung positiv aus, wird die zugrundeliegende Fahrerlaubnis oder der Stellbefehl an den Zug bzw. das Infrastrukturelement weitergegeben. Andernfalls erfolgt eine begründete Zurückweisung an das TMS. Prozessfunktionen können auch vom Bediener oder einem externen System über eine bereitgestellte Schnittstelle ausgelöst werden, beispielsweise ein Nothaltauftrag. Unterfunktionen erfüllen spezielle Teilaufgaben, welche von mehreren Prozessfunktionen genutzt werden können.

Prüfbedingungen sind aus den Gefährdungen abgeleitete Schutzanforderungen, gegen welche die Prozess- und Unterfunktionen die Zulässigkeit der jeweils zugrundeliegenden Anfrage an das TMS prüfen und bewerten. Eine Prüfbedingung kann aus mehreren Einzelprüfungen bestehen. Das Ergebnis der Prüfung einer Prüfbedingung durch eine Prozess- oder Unterfunktion führt ggf. zu einer Verringerung der Schutzrate. Am Ende der Prozessfunktion wird eine Gesamtbewertung der Schutzrate durchgeführt, auf deren Basis eine Anfrage an die smartLogic entweder zugelassen oder mit Begründung abgelehnt wird. Alle Anfragen und Antwortnachrichten an die smartLogic werden für spätere Auswertungen mitprotokolliert.

Die in Abb. 3 rot umrandeten Kästen verdeutlichen beispielhaft einen Prüfprozess mit Prozessfunktion F-E051 und zum Prüfprozess zugehörigen Unterfunktionen und Prüfbedingungen.

3.2 Auswahl einer geeigneten Modellierungssprache

Da die smartLogic ein sicherheitskritisches Software-System[14] ist, müssen bei der Entwicklung die Voraussetzungen der EN 50126 und EN 50128 berücksichtigt werden. In den Normen wird zwischen Software-Entwurf und Implementierung unterschieden. Bestandteil des Forschungsprojekts smartLogic ist dabei nur die Entwurfsphase und nicht die Implementierung. Letztere wird in einem Prototyp nur exemplarisch umgesetzt. EN 50128 schreibt unter anderem vor, dass die wesentlichen Algorithmen und Abläufe genau dargestellt werden müssen. Hierzu sind gemäß der Norm auch Diagramme zu verwenden [5]. Die Darstellung muss eindeutig sein und daher einer genau definierten Semantik und Syntax folgen. „Das gewählte Entwurfsverfahren muss Merkmale haben, die […] Folgendes unterstützen:

i) Abstraktion, Modularität und andere Eigenschaften, die die Komplexität kontrollieren;

ii) die klare und genaue Darstellung von

- Funktionalität;
- Informationsfluss zwischen den Komponenten;
- Reihenfolge und zeitbezogene Informationen;
- Parallelverarbeitung;
- Datenstrukturen und -eigenschaften;

iii) Verständlichkeit für den Menschen;

iv) Verifikation und Validierung.“ ([5], S. 22f)

Die Ziele der formalen Entwurfs-Modellierung sind also vor allem das Herstellen von Eindeutigkeit der verwendeten Begriffe, Nachrichten und Abhängigkeiten und die Veranschaulichung und unmissverständliche Darstellung der Prozesse. Zudem sollen verschiedene Umsetzungsvarianten vor

[14] Es wird davon ausgegangen, dass in Zukunft eine sichere generische Hardware Plattform existiert, auf der die smartLogic ausgeführt werden kann. Die Betrachtung der Hardware ist nicht Teil des Projekts smartLogic.

einer vollständigen Implementierung verglichen und diskutiert werden können. Vor allem für das letztgenannte Ziel ist eine modulare Darstellungsweise hilfreich, die es ermöglicht, verschiedene Teilaspekte ein- oder auszublenden bzw. in feinerer oder gröberer Darstellung anzuzeigen.

Eine solche Entwurfsmodellierung kann mit formalen oder semiformalen Notationssprachen erfolgen oder mittels grafischer Notation. Da smartLogic ein Forschungsprojekt ist, hat das verständliche Aufzeigen neuer Denkansätze einen hohen Stellenwert. Zudem ist es wichtig, dass neue Erkenntnisse auch nachträglich noch einfach in das Modell eingefügt werden können und Veränderungen an der Struktur unproblematisch möglich sind. Diese Anforderungen erfüllt die grafische Modellierung besser als formale und semiformale Notationssprachen. Formale Notationssprachen haben dagegen den Vorteil, dass sie bei korrekter Anwendung noch präziser sind. Dieser Nachteil der grafischen Notation kann im Forschungsprojekt akzeptiert werden, da das Ergebnis des Forschungsprojektes kein marktreifes Produkt sein soll. Aus diesen Gründen wird in der vorliegenden Arbeit die grafische Modellierung zur Beschreibung der Basis-Logik genutzt. Die Unified Modelling Language (UML) stellt dabei den Standard dar. Sie ermöglicht mit einer Reihe verschiedener Diagrammtypen, die aus demselben Datenmodell heraus erzeugt werden, die oben geforderten Informationen abzubilden. Weiterhin ist die UML weltweit gebräuchlich, leicht verständlich und aus Teilen der Diagramme ist für die Prototypentwicklung direkt Java-Code ableitbar. Zudem existiert eine Reihe mächtiger Werkzeuge für die UML-Modellierung, mit denen große Modelle übersichtlich und konsistent erstellt werden können. Deshalb wird die UML als Modellierungssprache in smartLogic verwendet (vgl. zur UML z. B. [6]).[15] UML ist von SysML (Systems Modeling Language) abzugrenzen, welche häufig bei der Modellierung sicherheitskritischer Systeme zur Anwendung kommt und eine Abwandlung von UML darstellt. SysML eignet sich besonders, wenn ein komplexes System aus verschiedenen Software- und physischen Komponenten und deren Zusammenwirken beschrieben werden soll. (Für weitere Informationen zur SysML vgl. [7].) Die Modellierung der smartLogic fokussiert sich auf das Verhalten der Sicherungslogik, während mit den benachbarten Komponenten über bereits klar definierte Schnittstellen kommuniziert wird. Aus diesem Grund bietet eine Verwendung der SysML im Projekt smartLogic keine Vorteile im Vergleich zum weiter verbreiteten Standard UML.

Die unterschiedlichen Diagrammarten der UML sind verschiedene Darstellungsformen des gleichen zugrundeliegenden Modells. Es wird unterschieden in Strukturdiagramme und Verhaltensdiagramme. Von den Strukturdiagrammen ist für die Modellierung im Projekt smartLogic vor allem das Klassendiagramm von Bedeutung, da mit seiner Hilfe das Vokabular (genauer gesagt die Substantive) für die Modellierung der Abläufe definiert wird. Jeder Begriff ist in Form einer Klasse mit seinen Attributen und Funktionen sowie den Abhängigkeiten zu anderen Begriffen modelliert. Auf eine genauere Beschreibung der smartLogic-Klassendiagramme wird an dieser Stelle verzichtet, da sich das vorliegende Paper auf die Modellierung der Prozesse konzentriert. Die Klassendiagramme werden parallel zur Modellierung der Prozessfunktionen mitgepflegt. D.h. ein Begriff wird im Klassendiagramm ergänzt, wenn bei der Beschreibung des Prozesses auffällt, dass dieser Begriff zur Beschreibung des Ablaufs benötigt wird, aber noch nicht enthalten ist.

Für die Beschreibung der Prozessfunktionen eignet sich zunächst auf der groben Abstraktionsebene das Aktivitätsdiagramm, welches im Folgenden eingesetzt wird. Je nach Bedarf können weitere Verhaltensdiagramme wie das Sequenzdiagramm für sequentielle Abläufe mehrerer beteiligter Systeme

[15] Es gibt natürlich weitere grafische Modellierungssprachen, die zur Beschreibung eingesetzt werden könnten, z. B. Petri-Netze, EPKs, etc. Aufgrund der geringen Relevanz einer ausführlichen Diskussion der Vor- und Nachteile dieser Sprachen für das eigentliche Forschungsprojekt wird an dieser Stelle darauf verzichtet und auf einschlägige Literatur zu der Thematik verwiesen.

oder das Zustandsdiagramm für die Modellierung des Verhaltens einer einzelnen Komponente zusätzlich zum besseren Verständnis des gewünschten Systemverhaltens beitragen.

3.3 Das UML-Aktivitätsdiagramm

Die aus der Funktionsanalyse abgeleiteten Prozessfunktionen werden also zunächst in einem UML-Aktivitätsdiagramm beschrieben. Das Aktivitätsdiagramm enthält die einzelnen Aktionen des Prozesses. Die Kontrollflüsse verbinden die Aktionen, so dass die Reihenfolge, in denen diese ausgeführt werden, ersichtlich wird. Der Kontrollfluss kann sich beliebig verzweigen und wieder vereinigen, so dass sequentielle und parallele Ausführungen von Aktionen möglich sind. Werden im Rahmen des modellierten Prozesses Nachrichten an benachbarte Systeme geschickt, werden diese ebenfalls im Aktivitätsdiagramm in einem je System abgegrenzten Bereich (als Schwimmbahnen bezeichnet) dargestellt. Zusätzlich wird mit sogenannten Objektflüssen die Weitergabe von Informationen modelliert. Einzelne Aktionen aus einem übergeordneten Aktivitätsdiagramm können in weiteren Aktivitätsdiagrammen verfeinert werden. Dies ermöglicht die Darstellung des Gesamtprozesses in verschiedenen Aggregationsstufen. Im Projekt smartLogic werden Aktionen der Prozessfunktionen häufig als Unterfunktionen verfeinert, die dann auch in anderen Prozessfunktionen nutzbar sind. (Vgl. [6])

3.4 Ablauf der Modellierung der Prozess- und Unterfunktionen

Ziel der Prozessfunktion ist i.d.R. das Bestimmen der Schutzrate, auf deren Basis die Entscheidung über die Zulässigkeit der zugrundeliegenden Anfrage getroffen wird. Hierfür ist Klarheit erforderlich, welche Prüfbedingungen für die Prozessfunktion relevant sind. Deshalb werden zunächst anhand der Liste der Prüfbedingungen, die ebenfalls in der Funktionsanalyse entstanden ist, diejenigen bestimmt, für die eine mögliche Verletzung durch eine fehlerhafte Ausführung des betrachteten Prozesses nicht ausgeschlossen erscheint. Dies ist zum Beispiel der Fall, wenn die Prüfbedingung ein Element enthält, dessen Zuweisung sich durch die Umsetzung der Prozessfunktion ändern könnte oder dessen Eigenschaften in einer anderen Form tangiert werden. Weiterhin, wenn sich die Prüfbedingung auf einen Akteur innerhalb der Prozessfunktion bezieht (z.B. das Fahrzeug oder das Infrastrukturelement, dessen Status durch den Prozess verändert werden soll).

Die Kriterien zur Bewertung, ob eine Prüfbedingung für die Prozessfunktion oder eine der Unterfunktionen von Relevanz ist, sind sehr vielfältig. Eine vollständige Auflistung ist daher nicht möglich und die bestehende Auflistung ist nur als Grundlage für die Bewertung zu verstehen. Diese muss letztlich qualitativ erfolgen. Im Rahmen dieser Arbeit erfolgt die Bewertung durch den Autor. Sie kann bzw. sollte bei Entwicklung eines Produktivsystems später von Fachexperten überprüft werden.

Damit die Schwimmbahnen der beteiligten externen Systeme gezeichnet werden können, werden zur Erstellung des Aktivitätsdiagramms im zweiten Schritt diejenigen externen Systeme identifiziert, mit denen im entsprechenden Prozess kommuniziert wird.

Darauf aufbauend wird das Aktivitätsdiagramm erstellt. Die Prüfbedingungen werden dabei i.d.R. über Aktionen abgeprüft. Bei sehr einfach zu prüfenden Bedingungen, die auch in der späteren Implementierung mit einer einfachen if-Abfrage umgesetzt werden können, wird zur Vereinfachung der Darstellung auf das Einzeichnen einer Aktion verzichtet und die Bedingung direkt an der Kontrollfluss-Verzweigung notiert. Komplexe Prüfbedingungen, wie z. B. ob eine Gefährdung durch Flankenfahrt ausgeschlossen oder das beantragte Geschwindigkeitsprofil zulässig ist, werden in eigenen Unterfunktionen modelliert, die im Aktivitätsdiagramm der Prozessfunktion als sogenannte „Call

Behavior Actions“ eingebunden sind. Unterfunktionen durchlaufen ebenfalls den gesamten hier geschilderten Modellierungsprozess. Die Anordnung der Aktionen erfolgt gemäß den Design-Prinzipien, die in den nachfolgenden Unterkapiteln beschrieben werden. Werden Informationen von anderen Systemen benötigt (z. B. vom Zug, von den Infrastrukturelementen oder über die Topologie), wird die entsprechende Nachricht sowie die Antwort modelliert. Dabei wird die Nachricht im Klassendiagramm ergänzt, sofern sie dort noch nicht vorhanden ist. Am Ende jeder Prozessfunktion folgt die Aktion der Berechnung der Schutzrate (vgl. Kapitel 3.5). Deren konkrete Berechnung ist jedoch nicht Teil des Projekts smartLogic. Wie in Kapitel 3.2 beschrieben, können ggf. mit weiteren UML-Verhaltensdiagrammen Teilaspekte der Prozesse zusätzlich verdeutlicht werden.

Anschließend an die Erstellung des Aktivitätsdiagrammes wird der modellierte Prozess nochmal gegen alle Prüfbedingungen – nicht nur die im Schritt 1 ausgewählten – geprüft und ggf. ergänzt bzw. modifiziert.[16]

Bei der Modellierung stellte sich heraus, dass die grafisch zu modellierenden Prozesse schnell umfangreich werden. Deswegen tritt vor die grafische Modellierung als Schritt 2 noch eine einfache textuelle Formulierung des grundsätzlichen Prozesses in natürlicher Sprache, so dass auch die Identifizierung der beteiligten externen Systeme einfacher wird. Der gesamte Ablauf der Modellierung für jede Prozess- und Unterfunktion besteht daher aus den in Abb. 4 dargestellten fünf Schritten:

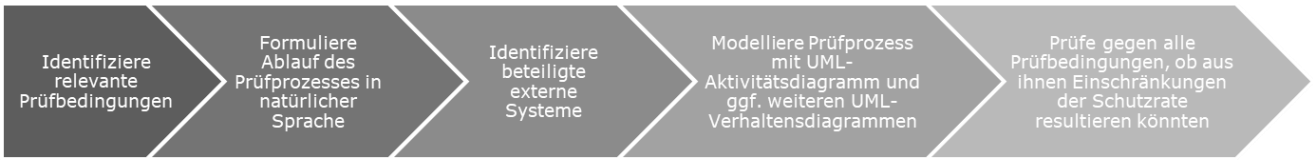

Abb. 4 Ablauf der Modellierung
[Eigene Darstellung]

3.5 Berechnung der Schutzrate

Die smartLogic entscheidet, anders als bisher übliche Sicherungslogiken, nicht auf Basis fest definierter Voraussetzungen über die Zulässigkeit einer Anfrage, sondern auf Basis der errechneten Schutzrate. Dabei ist die Bedingung, dass eine Gefährdung der zu schützenden Zugfahrt hinreichend unwahrscheinlich ist[17]. Dies unterstützt die Anforderung, dass eine Anfrage nur abgelehnt werden soll, wenn die Grundprämisse (siehe Kapitel 3.6) nicht erfüllt ist. Jede Prüfbedingung, die nicht vollständig erfüllt wird, führt zu einer Verringerung der Schutzrate. Dies kann bei schwerwiegenden Verstößen direkt zur Ablehnung der Anfrage führen. Unter bestimmten Voraussetzungen kann eine Verringerung der Schutzrate durch Verletzung einer bestimmten Prüfbedingung aber auch durch alternative Maßnahmen ausgeglichen werden, z. B. durch eine niedrigere Geschwindigkeit.

Die Schutzrate kann z. B. mittels Ereignisbäumen ermittelt werden, wie sie häufig in Risikoanalysen verwendet werden (vgl. [8]). Diesen liegt die Überlegung zugrunde, dass von jedem Ausgangsereignis ein Pfad über verschiedene weitere Ereignisse oder begünstigende Umstände bis zu einer Gefährdung existiert. Jedem dieser Ereignisse oder Umstände kann eine Wahrscheinlichkeit zugeordnet werden, mit der das Ereignis eintritt bzw. der Umstand vorliegt. Die Wahrscheinlichkeit für den Eintritt der

[16] Damit ist die Verifizierung nicht abgeschlossen. Wie bereits erwähnt, ist eine detailliertere Testphase auf Basis einer beispielhaften Umsetzung in einem Prototyp ebenfalls Teil des Projekts

[17] siehe Fußnote 5

Gefährdung auf Basis des betrachteten Ausgangsereignisses ist dann das Produkt der Einzel-Wahrscheinlichkeiten auf dem Pfad. Die Bestimmung der genauen Wahrscheinlichkeit zur Errechnung der Schutzrate ist jedoch, wie oben bereits erwähnt, nicht Teil des Projekts smartLogic.

Abb. 5 zeigt ein Beispiel eines solchen Ereignisbaums. Zugrunde liegt das Ausgangsereignis, wonach sich Wagen eines sich von der zu schützenden Zugfahrt wegbewegenden Personenzuges lösen. Zu sehen sind mehrere Ereignispfade, von denen drei im Ergebnis zu einer Flankenfahrt mit der zu schützenden Zugfahrt führen. Die eingetragenen Wahrscheinlichkeiten dienen nur der Veranschaulichung und haben keinen realen Hintergrund.

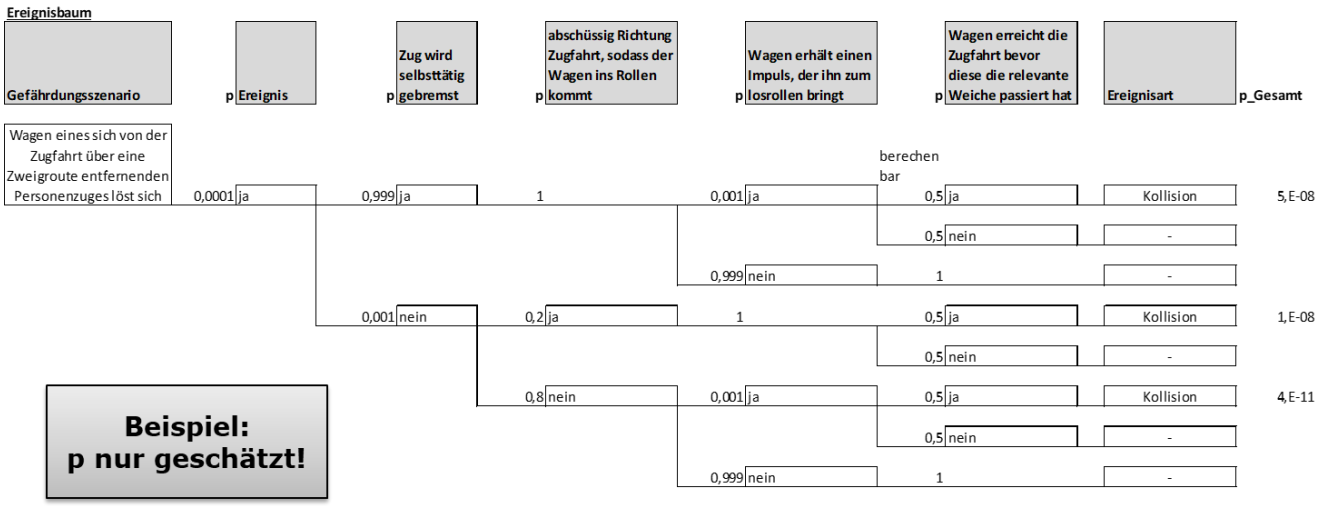

Abb. 5 Beispiel eines Ereignisbaums
[Eigene Grafik, angelehnt an die Darstellung in der ETCS-Risikoanalyse der DB Netz AG]

3.6 Designprinzipien der Prozessmodellierung

Die Prozessfunktionen und die darin aufgerufenen Unterfunktionen bestehen häufig aus einer ganzen Reihe von Prüfbedingungen, die entweder positiv ausfallen oder zur Generierung eines Fehlercodes und ggf. einer Einschränkung der Schutzrate führen. Auf den ersten Blick erscheint dies, wie ein sequentieller Ablauf, der wenig Freiraum für Design-Entscheidungen lässt. Es gibt aber doch einen gewissen Lösungsraum, innerhalb dessen bestimmte Entscheidungen zur Ausgestaltung der Modellierung regelmäßig getroffen werden müssen. Basis für diese Entscheidungen bilden dabei die aus den Zieldimensionen des Projekts abgeleiteten Anforderungen an die neue Sicherungslogik smartLogic (vgl. [2]). Aus diesen Anforderungen an die gesamte Sicherungslogik können wiederum Anforderungen an die Modellierung der smartLogic abgeleitet werden. Diese sind in Tab. 1 aufgelistet.

Bei den ersten beiden Punkten handelt es sich um die bereits erwähnten Sicherheitsanforderungen, die aus der Grundprämisse der smartLogic abgeleitet werden, wonach die smartLogic unsichere Zustände verhindern muss. Diese Anforderungen stehen natürlich an erster Stelle und sind nicht verhandelbar. Die anderen Anforderungen sind zunächst gleichberechtigt. Eine Abwägung erfolgt daher jeweils im Einzelfall.

Tab. 1 Übersicht der Anforderungen an die Modellierung

	Zieldimension	Anforderung smartLogic	Modellierungs-Anforderung
1	*Grundprämisse*	*smartLogic ist sicher*	*Übergang in unsicheren Zustand verhindern*
2	*Grundprämisse*	*smartLogic ist sicher*	*alle relevanten Prüfbedingungen erfüllt*
3	hohe Kapazität	Keine vermeidbaren Fahrtausschlüsse	keine Elemente blockieren, die nicht unmittelbar benötigt werden
4	hohe Kapazität	schnelle Zugfolge	Zuweisungszeiten[18] möglichst kurz halten
5	hohe Kapazität	den Fahrzeugen möglichst viel Freiraum zum Ausfahren ihrer Fahrerlaubnis lassen	Vorgaben so wenig restriktiv wie möglich
6	hohe Kapazität	schnelle Verarbeitung von Anfragen	möglichst wenig einzelne Nachrichten zwischen externen Systemen
7	hohe Kapazität	schnelle Verarbeitung von Anfragen, keine unnötigen Wartezeiten	möglichst parallele (asynchrone) Bearbeitung der Arbeitsschritte
8	Verkürzung der Zulassungsprozesse	schlanke Sicherungslogik	nur notwendige Arbeitsschritte enthalten
9	Beschleunigung der Planungsprozesse / Flexible Einsetzbarkeit / Geringer Projektierungsaufwand	Flexibilität für Erweiterungen / Einsetzbarkeit auf verschiedenen Infrastrukturen	Arbeitsschritte möglichst generisch
10	Robustheit	Rückfallebenen möglichst in Logik integrieren	Abbruch des Prozesses nur, wenn Grundprämisse nicht erfüllt wird
11	Robustheit	Änderung bereits erteilter MA ermöglichen	Rücknahme von Verschlüssen und Zuweisungen[19] vorsehen
12	Robustheit	schnelles Finden von Alternativen durch das TMS	Möglichst präzise Fehlercodes

[20]

[18] In diesem Paper wird der Begriff „Zuweisung“ (engl. assignment) statt „Belegungszeit“ verwendet, da unter letztgenanntem Begriff häufig nur die Zeit verstanden wird, in der ein Fahrzeug ein Infrastrukturelement tatsächlich besetzt. Unter Zuweisungszeit ist hier und im Folgenden aber die gesamte Zeit gemeint, in der ein Element einer Fahrt zugewiesen ist. Eine Zuweisung muss nicht ausschließlich sein. Das Element kann gleichzeitig auch noch einer oder mehrerer anderen Fahrt(en) zugewiesen sein.

[19] Siehe Fußnote 18.

[20] Aus den Zieldimensionen „Flexible und ergonomische Bedienung“ und „Senkung der Kosten“ leiten sich keine Modellierungsanforderungen ab, da die Bedienung für die innere Funktionsweise der smartLogic keine Rolle spielt (Das System ist als automatisiertes System ausgelegt) und die Kostensenkung über Fortschritte bei den anderen Zieldimensionen erreicht wird.

3.7 Beispiel für Design-Entscheidungen: Interne oder externe Fehlerbehandlung

Eine grundsätzliche Design-Entscheidung ist, inwieweit die smartLogic Anfragen, die aus Sicherheitsgründen nicht wie beantragt genehmigt werden können, selbstständig in einer modifizierten (reduzierten) Form genehmigt.

Denkbar ist zum Beispiel eine Fahrerlaubnis-Anfrage, die 60 km/h für die Fahrt über eine Weiche vorsieht, welche aufgrund eines Fehlers beim Verschluss allerdings momentan nur mit 5 km/h befahren werden darf. Eine Möglichkeit wäre, die Sicherungslogik so anzulegen, dass sie die Fahrerlaubnis-Anfrage mit einem entsprechenden Hinweis als unzulässig zurückweist. In diesem Fall müsste das TMS eine erneute modifizierte Anfrage stellen.

Sie könnte aber auch die Fahrerlaubnis – abweichend zur Anfrage – mit 5 km/h an der entsprechenden Weiche genehmigen.

Eine dritte Möglichkeit wäre automatisch einen alternativen Fahrweg zu kommandieren. Um diese Entscheidung zu unterstützen, wurde verallgemeinert jede Einzelprüfung (als Teil einer Prüfbedingung) im Diagramm farblich in eine Gruppe eingeteilt. Die Farbe beschreibt die Schwere der Abweichung im Vergleich zur ursprünglichen Anfrage, die nötig wäre, um eine aus der Einzelprüfung resultierende Verringerung der durch Änderung der zugrundeliegenden Anfrage zu kompensieren. (Dies ist von der Kompensation einer Verringerung der Schutzrate zu unterscheiden, die bereits in der Anfrage berücksichtigt ist, z. B., wenn die 5 km/h bereits im beantragten Geschwindigkeitsprofil vorgesehen wären.). Die Bedeutung der Farben ist in Tab. 2 beschrieben.

Tab. 2 Farbliche Zuordnung der Folgen einer fehlgeschlagenen Prüfbedingung

grün	Die Verletzung der Einzelprüfung hat keine Auswirkung auf die Genehmigung der Anfrage, sondern kann in jedem Fall intern kompensiert werden.
gelb	Die Verletzung der Einzelprüfung kann durch eine Änderung im Geschwindigkeitsprofil oder einen Wechsel in einen anderen überwachten ETCS-Modus kompensiert werden.
orange	Die Verletzung der Einzelprüfung kann durch eine größere Änderung der Anfrage, z. B. durch eine andere Route oder den Wechsel in den Modus SR kompensiert werden.
rot	Die Anfrage wird in jedem Fall zurückgewiesen.

Die Unterscheidung zwischen gelb und orange erfolgt, weil gelb letztlich nur eine Verzögerung der Fahrt zur Folge hat, während orange eine deutlich größere Auswirkung auf das Betriebsgeschehen insgesamt hat.

Bei der Beurteilung, welche dieser Kompensationen von der smartLogic intern vorgenommen werden sollen, handelt es sich um eine Abwägung verschiedener Zieldimensionen. Ein selbstständiges Agieren der smartLogic könnte die Robustheit erhöhen und ggf. durch eine schnellere Bearbeitung der Fahrerlaubnis-Anfragen eine höhere Kapazität ermöglichen. Auf der anderen Seite würde eine solche interne Fehlerbehebungslogik der Anforderung der schlanken Sicherungslogik entgegenstehen. Möglicherweise würde ein selbstständiges Agieren auch zu einer Kapazitätseinschränkung führen, da das TMS den besseren Gesamtüberblick hat und so beispielsweise zwischen den Varianten mit reduzierter Geschwindigkeit und alternativen Fahrweg eher die bessere Variante auswählen kann als die Sicherungslogik. Bei der smartLogic wurde diese Design-Fragestellung zugunsten der Anforderung der möglichst schlanken Sicherungslogik entschieden. Dies bedeutet, dass eine Anfrage bei einer

nicht ausreichenden Schutzrate abgewiesen wird und das TMS dann eine neue Anfrage stellen muss, welche auf Basis des Fehlercodes das entsprechende Problem berücksichtigt. In der Regel führen also sowohl gelb, als auch orange und rot zu einer Ablehnung der Anfrage, es sei denn bei gelb ist die entsprechende Einschränkung bereits durch die Fahrererlaubnis-Anfrage kompensiert. Im obigen Beispiel wäre das der Fall, wenn bereits 5 km/h an der einschränkenden Weiche in der Fahrerlaubnis-Anfrage vorgesehen sind.

4 Beispiel Kürzen einer Fahrerlaubnis (aus dispositiven Gründen)

In diesem Kapitel wird die Durchführung der Modellierung generischer sicherungstechnischer Prüfprozesse mit der in Kapitel 3 beschriebenen Methode anhand des Beispielprozesses „Kürzen einer Fahrerlaubnis“ veranschaulicht.

Zu unterscheiden ist bei der Kürzung einer Fahrerlaubnis, aus welchem Grund dies geschieht. Falls der Hintergrund die Abwehr einer unmittelbar drohenden Gefahr ist, wird die Rücknahme i.d.R. mit einem Nothaltauftrag verbunden (ETCS-Message 15 oder 16) [9] und hat Priorität. In diesem Beispiel geht es um eine vom TMS beantragte Kürzung aus dispositiven Gründen, aufgrund einer kurzfristig geänderten Betriebslage.

In den folgenden Unterkapiteln werden die einzelnen Schritte der Modellierung des Beispielprozesses gemäß dem Ablauf aus Abb. 4 beschrieben. Dem abschließenden Schritt 5 (erneutes Prüfen aller Prüfbedingungen) ist kein eigenes Unterkapitel gewidmet, da Ergebnisse dieser Prüfung direkt iterativ in den Modellierungsprozess eingeflossen sind und dort berücksichtigt wurden.

4.1 Wirkprinzip der ETCS-Funktion „Kürzen einer Fahrerlaubnis“

ETCS hält für das nachträgliche Kürzen einer Fahrerlaubnis die Messages 9, 137 und 138 bereit. Das Prinzip dahinter sieht vor, dass eine bereits erteilte Fahrerlaubnis uneingeschränkt gültig ist und davon ausgegangen wird, dass sie das Fahrzeug auch ausnutzt. Allerdings kann das Fahrzeug auf die Nutzung verzichten, wenn es mit hinreichender Sicherheit davon ausgeht, dass es bis zu einem neu übermittelten Gefahrpunkt zum Halten kommen kann. Mit Message 9 (Request to shorten MA) beantragt die Infrastrukturseite (heute das RBC, hier die smartLogic) beim Fahrzeug die Kürzung der Fahrerlaubnis, indem es eine neue Fahrerlaubnis mit einem neuen Gefahrpunkt übermittelt. In diesem Fall antwortet das Fahrzeug mit Message 137 (Request to shorten MA granted), andernfalls mit Message 138 (Request to shorten MA rejected). Damit die Funktion sinnvoll, automatisiert genutzt werden kann, muss durch die Sicherungstechnik die zugehörige Fahrstraße teilaufgelöst werden, so dass die freiwerdenden Elemente für andere Fahrzeugbewegungen neu zugewiesen werden können. Dies ist bei klassischen Stellwerken nur mit Hilfshandlung möglich, da die Fahrstraße für das Ausstellen der ursprünglichen, ungekürzten Fahrerlaubnis bereits festgelegt gewesen sein muss.

4.2 Identifizieren der für den Prüfprozess relevanten Prüfbedingungen

Zunächst erscheint der Prozess „Kürzen einer Fahrerlaubnis“ für die Infrastrukturseite sicherungstechnisch unproblematisch zu sein, da das Fahrzeug für sich selbst prüft, ob es einen bestimmten Fahrweg noch benötigt oder nicht und mit der Freigabe auch garantiert, dass es den freigegebenen Fahrweg nicht mehr verwenden wird, sofern es keine neue Fahrerlaubnis dafür bekommt. Eine detailliertere Prüfung der Prüfbedingungen ergibt aber dennoch vielfältiges Potential zur Verringerungen der Schutzrate, so dass die betroffenen Prüfbedingungen näher betrachtet werden

sollten. Ob eine Verletzung der einzelnen Schutzbedingungen auch wirklich eine signifikante Einschränkung der Schutzrate zur Folge hat, wird dabei an dieser Stelle nicht betrachtet. Im Folgenden sind die identifizierten Prüfbedingungen mit Begründung aufgeführt:

1. smartLogic muss korrekt arbeiten. (Dies ist vor jedem Prüfprozess sicherzustellen.)
2. Verhindern der Ausgabe einer Fahrerlaubnis an ein nicht klar definiertes Fahrzeug.
   Das Fahrzeug, für welches die Kürzung beantragt wurde, muss von der smartLogic eindeutig identifiziert werden.
3. Eine Nachricht muss an das richtige Fahrzeug übermittelt werden.
   Es muss also sichergestellt werden, dass die Freigabe wirklich von dem Fahrzeug erfolgt, für welches die Elemente, die freigegeben werden sollen, auch verschlossen waren.
4. Sichern der beweglichen Fahrwegelement gegen Umstellen unter dem Zug.
   Daraus folgt: Elemente dürfen erst freigegeben werden, wenn sie durch keine Zuweisung mehr beansprucht werden. Es muss verhindert werden, dass das Kürzen einer Fahrerlaubnis dazu führt, dass ein Element freigegeben wird, welches noch von anderen Fahrzeugen (z.B. überlappte Fahrerlaubnis im Bereich des Durchrutschwegs, Beanspruchung als Flankenschutzelement, Festlegung in der aktuellen Lage aufgrund eines Defekts…) in der aktuellen Lage benötigt wird.
5. *Die neue Fahrerlaubnis muss zulässig sein.*
   Es wird mit Message 9 eine komplett neue Fahrerlaubnis übermittelt, welche die vorgesehene Kürzung enthält und im Falle der Annahme durch das Fahrzeug die bestehende Fahrerlaubnis ersetzt. Diese könnte auch noch an anderen Stellen als dem neuen Zielpunkt und Gefahrpunkt von der ursprünglichen Nachricht abweichen. Eine ganze Reihe von Prüfbedingungen ist für die Ausstellung neuer Fahrererlaubnisse von Relevanz. Daher muss entweder festgestellt werden, dass die neue Fahrerlaubnis nicht weniger restriktiv ist als die alte oder die neue Fahrerlaubnis muss wie eine komplett neue Fahrerlaubnis den Prozess „Ausstellen einer Fahrerlaubnis“ durchlaufen. Da es sich nicht um eine einzelne Prüfbedingung, sondern um ein ganzes Set an Prüfbedingungen handelt, ist dieser Punkt kursiv dargestellt.
6. Stopps in “Non stopping areas” (NAS) vermeiden.
   Bestimmte Zugtypen sollen in bestimmten Bereichen nicht zum Halten kommen, wenn dies vermeidbar ist. Ein vorzeitiger Stopp aus dispositiven Gründen darf daher nicht in einer NSA liegen.
7. Sicherstellen, dass ein Personenzug im Bahnhofsbereich an einem Bahnsteig hält.
   Dies klingt erstmal nicht wie eine Aufgabe der Sicherungslogik. Eine Verletzung dieser Bedingung kann aber durchaus eine Verringerung der Schutzrate zur Folge haben, da Fahrgäste durch einen vorzeitigen Halt im Bahnhofsbereich dazu ermuntert werden können, die Türen unzeitig zu öffnen. Dies kann bei gleichzeitig vorzeitig freigegebenen Türen zu einer erhöhten Wahrscheinlichkeit eines Unfalls führen.
8. Vermeiden, dass Fahrzeuge mit zu geringem Zugkraftüberschuss beim Anfahren in steil geneigten Rampen zum Stehen kommen.
   Für das Anfahren in Rampen ist mehr Energie nötig, als für das Durchrollen. Deshalb kann ein Stopp eines Zuges mit geringem Zugkraftüberschuss beim Anfahren in steil geneigten Rampen zum Problem werden. Zu prüfen ist an dieser Stelle, ob dies nur ein betriebshemmendes oder auch ein sicherheitsrelevantes Problem ist. Eine solche Bewertung wird hier nicht abschließend vorgenommen. Die Umsetzung einer solchen Bedingung ist aber über die Definition einer „Non stopping area“ möglich, von der anhand bekannter Zugdaten solche Fahrzeuge befreit werden, die noch zu ermittelnde Grenzwerte erfüllen.
9. Verhindern, dass ein Flankenschutz-gebendes Element vorzeitig entsperrt wird.

Bei vollüberwachten Fahrzeugen stellt ETCS sicher, dass diese definitiv bis zum Stillstand kommen und überwacht diesen Stillstand auch. Daher ist es unter bestimmten Voraussetzungen möglich, dass vollüberwachte ETCS-Fahrzeuge auch Flankenschutz bieten können.[21] Beim Kürzen einer Fahrerlaubnis ist daher darauf zu achten, ob dies Auswirkungen auf Schutzraten anderer Züge durch veränderte Flankenschutzbedingungen haben und falls ja, ob diese kompensiert werden können.

10. Sicherstellen der maximalen Schließzeit eines Bahnübergangs (wo möglich).
    Durch das Kürzen der Fahrerlaubnis und vorzeitige Bremsungen könnten Bahnübergänge später befahren werden, als ursprünglich vorgesehen. Dies könnte zur Überschreitung der maximal vorgesehenen Schließzeiten der BÜ führen, was Auswirkungen auf die Schutzrate hat.
11. Verhindern der gleichzeitigen Nutzung von Tunneln durch Reise- und Güterzüge ab einer Grenz-Relativgeschwindigkeit der beteiligten Züge.
    Das Tunnelbegegnungsverbot hat erst jüngst Einzug in die Eisenbahnsicherungstechnik gehalten. Derzeit ist noch nicht genau abzusehen, wie weitreichend das Tunnelbegegnungsverbot in Zukunft ausgelegt sein wird. Dennoch wird an dieser Stelle diese Prüfbedingung der Vollständigkeit halber mit aufgeführt, da eine vorzeitige Verlangsamung des Zuges auch zu geänderten Begegnungskonstellationen im Tunnel führen könnte.[22]

Wie bei der fünften identifizierten Prüfbedingung beschrieben, wird mit Message 9 eine neue Fahrerlaubnis übertragen. Für den Fall, dass diese den normalen Prüfprozesse für das „Ausstellen einer Fahrerlaubnis“ durchläuft, brauchen einige der anderen Prüfbedingungen nicht mehr gesondert betrachtet werden, da deren Betrachtung bereits durch den Prüfprozess der neuen Fahrerlaubnis abgedeckt ist.[23] Es verblieben dann neben 5 die Prüfbedingungen 3, 4 und 9.

4.3 Formuliere Ablauf des Prüfprozesses in natürlicher Sprache

Dieser Arbeitsschritt dient auch dazu, grundsätzliche Design-Entscheidungen vorzudenken. Im vorliegenden Beispiel stellt sich die Frage, ob:

- die neue Fahrerlaubnis als verkürzte Variante der alten Fahrerlaubnis nur daraufhin überprüft wird, dass sie keine weniger restriktive Vorgabe enthält als die ursprüngliche Fahrerlaubnis (zum Beispiel eine höhere erlaubte Geschwindigkeit innerhalb des neuen Geschwindigkeitsprofils als an derselben Stelle im alten)

oder ob

- sie in jedem Fall wie eine ganz neu beantragte Fahrerlaubnis den kompletten Prüfprozess für das „Ausstellen einer (neuen) Fahrerlaubnis“ durchläuft.

Letztgenannter Prüfprozess ist sehr umfangreich und es besteht daher die Gefahr, dass viele eigentlich nicht benötigte Abfragen durchgeführt werden (vgl. Tab. 1: Anforderung der schnellen Verarbeitung

[21] Das Flankenschutzkonzept zur smartLogic ist noch nicht veröffentlicht. Der geschilderte Umstand ist hier dennoch der Vollständigkeit halber mitaufgeführt.

[22] Dies wird hier betrachtet, auch, wenn die Wahrscheinlichkeit für einen solchen Fall – ohne detaillierte Berechnung – gering erscheint, da nur hohe Geschwindigkeiten der beteiligten Züge für das Tunnelbegegnungsverbot eine Rolle spielen, diese aber durch den eingeleiteten Bremsvorgang des betrachteten Zuges bereits abgenommen haben muss.

[23] An dieser Stelle wird auf den vollständigen Prozess zur Prüfung der Fahrerlaubnis verwiesen. Dieser Prozess ist zum Zeitpunkt der Veröffentlichung dieses Beitrags noch nicht veröffentlicht. Das primäre Ziel des vorgestellten Prozesses „Kürzen einer Fahrerlaubnis“ als Beispiel zur Verdeutlichung der in Kapitel 3 beschriebenen Methode wird dadurch jedoch nicht beeinträchtigt.

von Anfragen > möglichst wenig Nachrichten zwischen externen Systemen). Andererseits bietet dieses Verfahren die größtmögliche Sicherheit, da alle Implikationen einer neuen Fahrerlaubnis auch neu bewertet werden und es vereinfacht die Modellierung der Prozessfunktion „Kürzen einer Fahrerlaubnis“, da ein größerer Teil der in Kapitel 4.2 identifizierten Prüfbedingungen im Prüfprozess „Ausstellen einer Fahrerlaubnis“ bereits abgeprüft werden. Dadurch wird der vorliegende Prozess übersichtlicher (vgl. Tab. 1: schlanke Sicherungslogik > nur notwendige Arbeitsschritte enthalten). Zusätzlich bietet es die größtmögliche Flexibilität, da prinzipiell auch Abweichungen möglich werden, die im Vergleich zur ursprünglichen Fahrerlaubnis weniger restriktiv sind (vgl. Tab. 1: Vorgaben so wenig restriktiv wie möglich). Insgesamt kann davon ausgegangen werden, dass das Kürzen einer Fahrerlaubnis im Vergleich zum normalen Ausstellen von Fahrerlaubnissen eher selten stattfindet. Deshalb erscheint in diesem Fall die Lösung mit vollständiger Prüfung der neuen Fahrerlaubnis sinnvoller zu sein und wird im Folgenden weiterverfolgt.

Der grobe Ablauf des Prüfprozesses „Kürzen einer Fahrerlaubnis“ stellt sich demnach wie folgt dar:

1. Prüfe Funktionsfähigkeit der smartLogic
2. Prüfe Anfrage auf syntaktische Korrektheit
3. Prüfe anhand der Elementzuweisungen, ob das richtige Fahrzeug adressiert wurde
4. Prüfe die neu beantragte Fahrerlaubnis auf Zulässigkeit
5. Prüfe, ob die Elementfreigabe Auswirkungen auf den Flankenschutz für andere Züge hat
6. Kalkuliere die Schutzrate
7. Falls hinreichender Schutz besteht, um die Anfrage auf Kürzung zu genehmigen, sende Message 9
8. Prüfe Antwort auf Korrektheit
9. Falls Antwort des Zuges positiv (Message 137), lösche die Zuweisungen für die nicht mehr benötigten Elemente
10. Lösche ggf. den Verschluss von Elementen am Element selbst, wenn dieses nicht mehr benötigt wird und keine weiteren Zuweisungen mehr hat.
11. Sende eine Rückmeldung über das Ergebnis des Prozesses an das TMS

4.4 Beteiligte externe Systeme

Neben dem Traffic Management System, von welchem der Prozess angestoßen wird, ist natürlich das Fahrzeug, dessen Fahrerlaubnis gekürzt werden soll, ein beteiligtes externes System.

Über den in den Ablauf integrierten Prozess der Prüfung „Ausstellen einer Fahrerlaubnis“ sind jedoch auch alle bei diesem Prozess beteiligten externen Systeme im Betrachtungsraum wie Infrastrukturelemente und z. B. Bahnübergänge von Interesse. Auch andere Züge spielen dafür zum Beispiel bei den Themen Flankenschutz und Tunnelbegegnungsverbot eine Rolle. Im UML-Aktivitätsdiagramm des Prozesses „Kürzen einer Fahrerlaubnis“ wird der Prüfprozess „Ausstellen einer Fahrerlaubnis“ jedoch als einfache Aktion beschrieben, welche auf die Modellierung des letztgenannten Prozesses verweist. Deshalb brauchen externe Systeme, die nur im Rahmen des letztgenannten Prozesses beschrieben werden, im Diagramm für den Prozess „Kürzen einer Fahrerlaubnis“ nicht aufgeführt werden.

Am Ende des Prozesses werden ggf. Verschlüsse am Element selbst aufgeboben. Hierfür ist eine Kommunikation mit diesen Elementen erforderlich. Auch dieser Prozess ist in einer eigenen Unterfunktion modelliert, da er von verschiedenen Prozessfunktionen angestoßen werden kann. Deshalb ist auch diese Kommunikation zur Vereinfachung der Darstellung im Aktivitätsdiagramm nicht dargestellt.

Demnach wird im Aktivitätsdiagramm nur der beteiligte Zug als externes System berücksichtigt.

4.5 Aktivitätsdiagramm

Zur Beschreibung des Prozesses „Kürzen einer Fahrerlaubnis“ wurde das in Abb. 6 dargestellte Aktivitätsdiagramm erstellt.

An den Verzweigungsknoten und den Ausgabe-Pins der Aktionen, die für Unterfunktionen stehen, ist dargestellt, welche Auswirkungen die Nicht-Erfüllung der jeweiligen Prüfbedingung hat. Die Einteilung erfolgt gemäß Tab. 2. Im vorliegenden Beispiel sind die meisten Verzweigungsknoten rot eingefärbt. Dies bedeutet, dass die Nicht-Erfüllung direkt zur Zurückweisung der Anfrage führt. Dies hat den Grund, dass zu Beginn des Prüfprozesses die grundsätzliche Validität der Anfrage geprüft werden muss. Ist diese nicht gegeben, muss das TMS eine erneute Anfrage stellen. Bei sofortiger Zurückweisung wird ein Fehlercode generiert. Dieser wird am Ende des Prozesses mit der Antwortnachricht an das TMS übermittelt, so dass dieses den Fehler analysieren und eine erneute korrigierte Antwort stellen kann.

Übt das betroffene Fahrzeug, dessen Fahrerlaubnis gekürzt werden soll, eine Flankenschutzfunktion aus, wird versucht diesen Flankenschutz über einen alternativen Weg herzustellen. Diese Unterfunktion generiert nicht direkt eine Zurückweisung der Anfrage, sondern ggf. eine Einschränkung der Schutzrate, sofern der Flankenschutz nicht gleichwertig anderweitig hergestellt werden kann. Dies hat den Hintergrund, dass das durch den reduzierten Flankenschutz entstehende Risiko, ggf. über andere Faktoren, wie niedrige erlaubte Geschwindigkeiten soweit ausgeglichen werden kann, dass die Anfrage dennoch genehmigt werden kann. Dies ist durch den gelben Ausgabe-Pin an der entsprechenden Aktion dargestellt. Über diesen fließt die Einschränkung der Schutzrate in die Gesamtbewertung der Schutzrate für den Prüfprozess ein.

Ist die Gesamtrate zu niedrig, bricht der Prozess ebenfalls ab und es wird ein weiterer Fehlercode erzeugt. Nach Tab. 2 ist diese Verzweigung jedoch orange zu markieren, da theoretisch durch eine veränderte Anfrage, die Schutzrate auch durch smartLogic-interne Maßnahmen erhöht werden könnte. Wie in Kapitel 3.7 beschrieben, wurde jedoch die Design-Entscheidung getroffen, keine interne Veränderung der Anfrage des TMS vorzunehmen. Deshalb wird die entsprechende Einschränkung mit einer Ablehnung der Anfrage an das TMS über die Antwortnachricht zurückgegeben.

Ist die Gesamtrate ausreichend, war die Anfrage des TMS, die Fahrerlaubnis zu kürzen, erfolgreich. In diesem Fall werden die Zuweisungen für die nicht mehr für die neue Fahrerlaubnis benötigten Elemente gelöscht. Anschließend wird bei beweglichen Fahrwegelementen überprüft, ob der Schutz gegen Umstellen am Element selbst noch benötigt wird. Dies hängt davon ab, ob noch weitere Zuweisungen vorhanden sind.

Abschließend wird die Antwortnachricht an das TMS generiert und an dieses versendet.

Die meisten Aktionen sind hier sogenannte „Call Behavior Actions“, die auf Unterfunktionen verweisen, in denen ihr Verhalten weiter beschrieben ist. Dies wird durch das kleine Symbol rechts unten an den Aktionen angezeigt.

Abschließend wurde erneut die Liste der Prüfbedingungen dahingehend analysiert, ob sich aus dem nun modellierten Prozess weitere Einschränkungen der Schutzraten einzelner Prüfbedingungen ergeben.

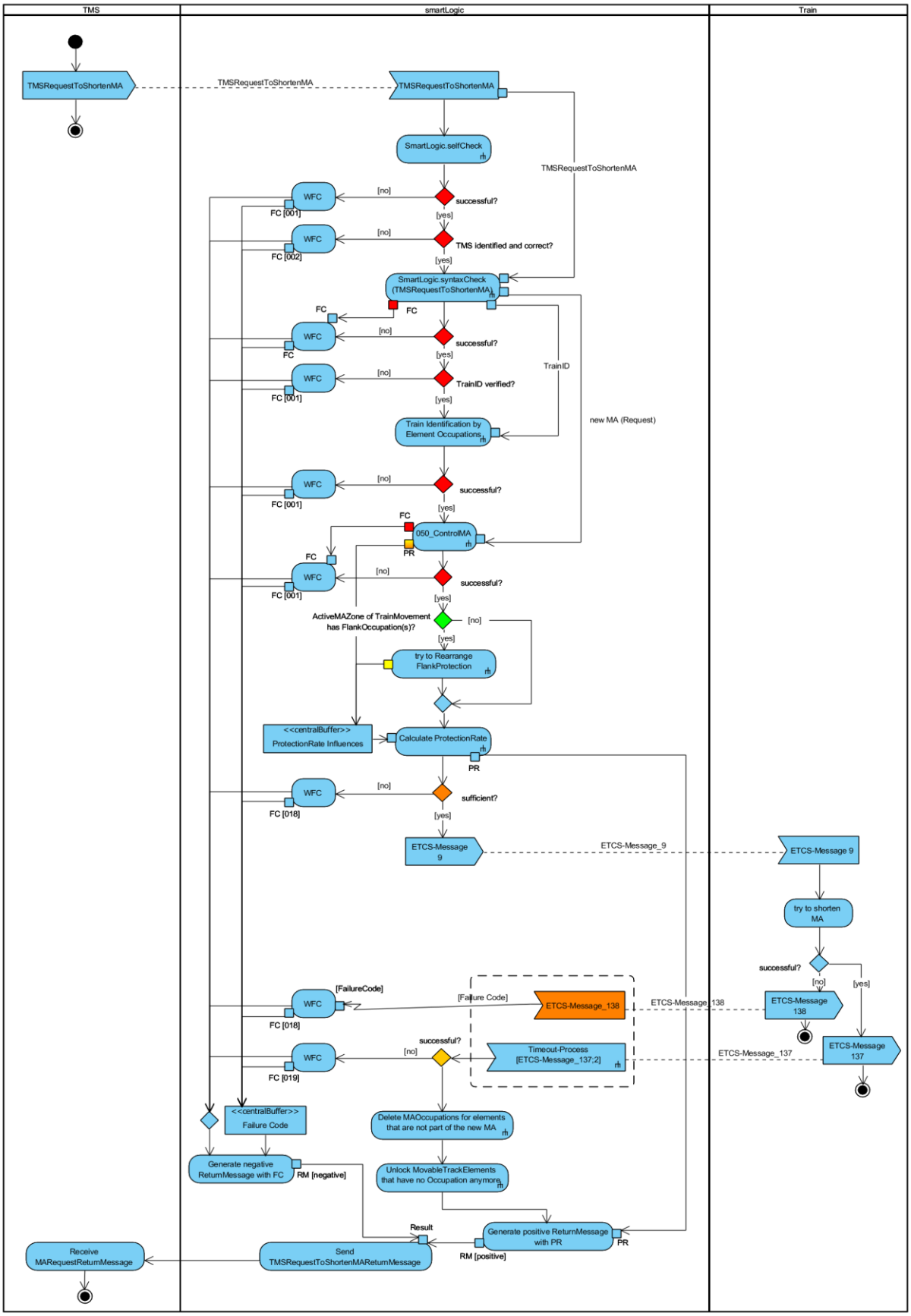

Abb. 6 UML-Aktivitätsdiagramm für Prozess Kürzen einer Fahrerlaubnis
[Quelle: Eigene Darstellung, erstellt mit Visual Paradigm]

5 Diskussion

Das gewählte Beispiel verdeutlicht am relativ einfachen Prozess „Kürzen einer Fahrerlaubnis“, wie Prüfprozesse für die smartLogic modelliert werden. Aufgrund der vielschichtigen sicherungstechnischen Zusammenhänge bei der Eisenbahn kommen auch bei einfachen Prozessen schnell zahlreiche Prüfbedingungen zusammen. Die strukturierte Identifizierung der Prüfbedingungen in [3] und die Identifizierung der relevanten Prüfbedingungen für den zu modellierenden Prozess liefern die notwendige Voraussetzung für die Modellierung. Allerdings kann durch die begrenzten Ressourcen des smartLogic-Projektteams trotz des strukturierten Vorgehens deren Vollständigkeit nur bedingt garantiert werden. Eine weitere Schleife mit Fachexperten wäre in diesem Schritt bei der Entwicklung eines Produktivsystems ratsam. Der Hauptfokus des Projekts smartLogic liegt jedoch auch auf der Identifizierung von Potentialen für eine effizientere Nutzung der Infrastruktur durch eine Modifizierung der Sicherungslogik. Solche Potentiale können durch das gewählte Verfahren aufgezeigt und zur weiteren, professionellen Untersuchung vorgeschlagen werden.

Das zugrundeliegende Datenmodell in Form der Klassenmodellierung stellt sicher, dass Begriffe eindeutig definiert sind. Durch eine modulare Gestaltung der Prozesse mit Hilfe von Unterfunktionen und der Einbindung bereits modellierter Prozessfunktionen wird im Sinne einer schlanken Sicherungslogik redundante Beschreibung verhindert, so dass die für den Beispielprozess getrennt zu modellierenden Prüfbedingungen signifikant reduziert werden können. Das Aktivitätsdiagramm erlaubt die kompakte, aber dennoch vollständige Darstellung des Prozesses. Allerdings ist die Darstellung nicht ganz so eindeutig, wie eine formale Notation. Für den beschriebenen, wissenschaftlichen Zweck der Arbeit, ist die grafische Modellierung aus Sicht des Autors aber hinreichend präzise.

6 Fazit und Ausblick

In diesem Paper wurde die im Projekt smartLogic verwendete Methode für die Modellierung sicherungstechnischer Prüfprozesse bei der Schaffung einer neuen infrastrukturseitigen Sicherungslogik erläutert und anhand des Beispiels „Kürzen einer Fahrerlaubnis“ veranschaulicht. Die entwickelte Methode ermöglicht die übersichtliche Darstellung des Ablaufs der Prozessfunktionen und kann somit als Input für eine weitergehende Diskussion und spätere professionelle Implementierung dienen. Kapazitätspotentiale durch den Einsatz einer smarten Sicherungslogik können mit ihren umfangreichen Implikationen veranschaulicht werden. So dient die smartLogic-Prozessfunktion „Kürzen einer Fahrerlaubnis“ dazu, eine Funktionalität, die über ETCS bereits bereitgestellt wird, auch von der infrastrukturseitigen Sicherheitstechnik aus verwendbar zu machen und somit eine situationsgerechte Reaktion durch das Traffic Management System auf sich kurzfristig ändernde Betriebslagen zu ermöglichen. Dies hat insbesondere bei Abweichungen vom Regelbetrieb Vorteile, so dass geringere Folgeverspätungen zu erwarten sind. Die mit der Funktion „Kürzen einer Fahrerlaubnis“ verbundene, sichere automatische Wiederfreigabe von bereits einer Zugfahrt zugewiesenen Infrastrukturelementen (heute Fahrstraßenhilfsauflösung) vermeidet auch sicherheitskritische Hilfshandlungen durch Bediener*innen.

Das Projekt smartLogic umfasst den gesamten Prozess der Zulässigkeitsprüfung und Überwachung von Stellanforderungen und Fahrerlaubnis-Anfragen. Weitere Prozess- und Unterfunktionen werden nach und nach modelliert und dabei mögliche Kapazitätssteigerungspotentiale untersucht. Ein ausführlicher Test-Prozess mit Hilfe eines Prototypens im Eisenbahnbetriebsfeld Darmstadt sowie mit formalen Prüfmethoden ist in einer weiteren Projektphase vorgesehen. Damit sollen die gewonnenen

Erkenntnisse einem Praxistest unterzogen und die identifizierten Kapazitätspotentiale näher untersucht werden.

Im Projekt smartLogic wurden bereits mehrere Potentiale für Kapazitätssteigerungen und Steigerungen der Betriebsqualität durch eine smarte Sicherungslogik identifiziert. Dies zeigt, dass eine Umsetzung zusammen mit weiteren Komponenten der aktuellen Digitalisierungsstrategie für den Bahnverkehr, wie ETCS und DSTW sinnvoll ist.

7 Literaturverzeichnis

[1] Deutsche Bahn AG (2018): *Digitale Schiene Deutschland: Revolution für den Bahnbetrieb*, Themendienst, Berlin 2018, verfügbar auf https://www.deutschebahn.com/de/presse/suche_Medienpakete/medienpaket_digitale_schiene_deutschland-1177310, abgerufen am 26.03.2019.

[2] Düpmeier, Frederik (2017): *Entwurf einer neuen, regelbasierten Sicherungslogik unter Annahme der vollständigen Ortung aller Schienenfahrzeuge*, Beitrag zum 1. Scientific Railway Signalling Symposium, Darmstadt April 2017.

[3] Düpmeier, Frederik (2018): *Funktionsumfang einer Sicherungslogik zur effizienten Ausnutzung der Möglichkeiten von ETCS“*, Beitrag zu den 26. Verkehrswissenschaftlichen Tagen, Dresden März 2018.

[4] Maschek, Ulrich (2012): *Sicherung des Schienenverkehrs – Grundlagen und Planung der Leit- und Sicherungstechnik.* 2., überarbeitete und erweiterte Auflage, erschienen bei Springer Vieweg, Wiesbaden.

[5] DIN EN 50128: *Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Software für Eisenbahnsteuerungs- und Überwachungssysteme*; Deutsche Fassung

[6] Kleuker, Stephan (2018): *Grundkurs Software-Engineering mit UML – Der pragmatische Weg zu erfolgreichen Softwareprojekten*, 4. Auflage, erschienen bei Springer Vieweg, Wiesbaden.

[7] The Object Management Group (2019): *What is SysML?*, abgerufen unter http://www.omgsysml.org/what-is-sysml.htm am 04.04.2019

[8] Wang, John X.; Roush, Marvin L. (2000): *What every engineer should know about risk engineering and management,* erschienen bei Marcel Dekker, Inc., New York, S. 69ff.

[9] ERA; UNISIG; EEIG ERTMS USERS GROUP (2016): ERTMS/ETCS Subset-026 System Requiremen Specification, Chapter 8 Messages, version 3.6.0, 2016-05-13.

Train-borne Localization Exploiting Track-Geometry Constraints – A Practical Evaluation

Hanno Winter[1], **Volker Willert**[1], **and Jürgen Adamy**[1]

All authors: Control Methods and Robotics Laboratory, TU Darmstadt, Germany

1 Introduction

To manage the constantly increasing traffic volume in the railway system it is necessary to utilize the existing infrastructure much more efficiently. Train-borne localization can be a key to achieve this. It allows for an optimal utilization of the existing infrastructure and at the same time many track-side elements, e.g. track-side signals, could be omitted. By that train-borne localization systems help to make the whole railway system not only more efficient in an operational but also in an economical sense.

However, there is no train-borne localization system available on the market yet [3]. This is due to the demanding safety requirements such a system has to fulfill according to EN 50126 [4]. The main challenge is to ensure a reliable and always available track-selective localization result, i.e. ± 1.5m in cross-track direction [5]. Up to now, none of the investigated sensor set-ups could achieve the postulated safety requirements at such a positioning accuracy level. This is why train-borne localization has gained interest in research and development recently [3,6].

To help overcome the issues related to train-borne localization, on the one hand, we focus on developing new methods which help to improve the availability and accuracy of train-borne localization systems. On the other hand, we generate digital track-maps specifically tailored for train-borne localization. This is due to the fact that maps pose a single point-of-failure in the overall localization process. Therefore, we deem it is helpful to integrate a mapping method directly into the localization process which makes it possible to continuously provide accurate maps and to detect possible mapping errors.

Consequently, we presented a new localization approach in [1]. It is characterized by an increased positioning accuracy especially in bad GNSS situations and in cross-track direction. Additionally, we appended this approach by a mapping functionality which creates compact geometric track-maps being advantageous for train-borne localization applications [2].

Until now both approaches have only been evaluated with the help of simulations. In contrast to that, here we want to present an evaluation with the help of real measurement data.

2 Basics

The work presented in the remainder of this paper is based on two previous publications of ours. In [1] we presented a new train-borne localization filter and in [2] we extended it with a mapping function. In both, the working principal is demonstrated in simulations. Here we want to focus on the evaluation of the approaches with real measurement data. Prior to that, the most important ideas of the localization and mapping filter as well as the utilized sensor set-up will be presented briefly.

[1] {hanno.winter,vwillert,adamy}@rmr.tu-darmstadt.de

2.1 Localization and Mapping Filter

Our localization and mapping filter is characterized by an improved positioning accuracy, especially in bad GNSS situations and in cross-track direction. This is achieved by directly incorporating track-geometry constraints in the sensor-fusion process of the filter. The unique feature is that the track-geometry constraints are identified in real time [1]. Thus, no initial digital track-map is needed. Furthermore, the identified track-geometries are used to generate digital track-maps. Compared to the mostly used track maps, our maps are very compact, even though they contain additional track-geometry information [2]. The function principal of the filter is illustrated in Figure 1.

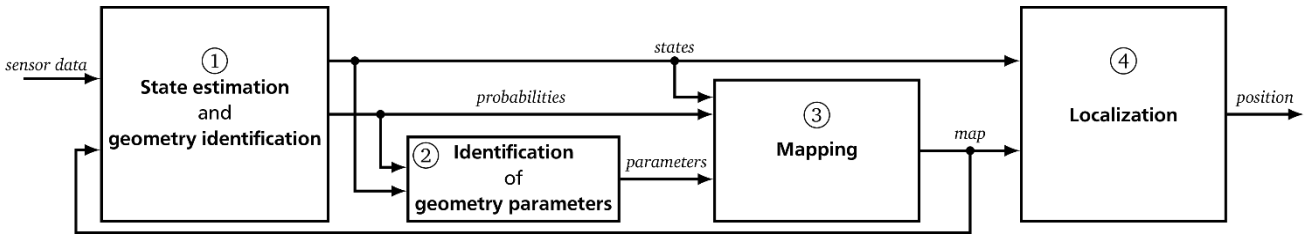

Figure 1: Working principal of the localization and mapping filter

The first step (1) is to identify the track geometry the train is currently moving on, e.g. a straight or a circular arc track-segment. At the same time the kinematic states of the train, e.g. its current speed or heading angle are estimated. Therefore, an interacting multiple model (IMM) filter with three models is used. Each model is adapted to the train's motion on a particular track-geometry, i.e. motion on a straight line, motion on a circular arc or neither of them. In our simulations we assumed GNSS and IMU data to be available as input data, since at least these sensors can be considered to be available in any kind of train-borne localization system of the future [3,6]. This is also the kind of data which is available for the evaluation in the remainder of this paper. However, if available, it would make sense to integrate more sensor data in the filter, e.g. additional odometer or a speed data. In the next step (2) the state estimates and the geometry information are used to calculate the track's current geometry parameters, e.g. the starting point and elevation of a straight line. Then (3) the current parameters are combined with all previously identified track-geometries to generate a compact geometric track-map. Last (4), the map and the state estimates are combined to a position estimate. Once a track-map has been generated it can serve as additional input for future journeys on this track. By that the generated map and the achievable positioning accuracy will improve over time.

2.2 Measurement Set-Up

The underlying data for this paper originate from a test drive between Annaberg-Buchholz and Schwarzenberg in the Erzgebirge in Germany. They were recorded at the 24-th of October in 2018 under cloudy weather conditions. The track is visualized in **Fehler! Verweisquelle konnte nicht gefunden werden.**. It is a non-electrified secondary line in a harsh railway environment, i. e. tight curves, steep slopes, forested embankment and strongly changing weather conditions. The track is not used regularly but still fully maintained. As a result, this track is often used for testing new technologies. The following evaluations were carried out on the section red marked in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

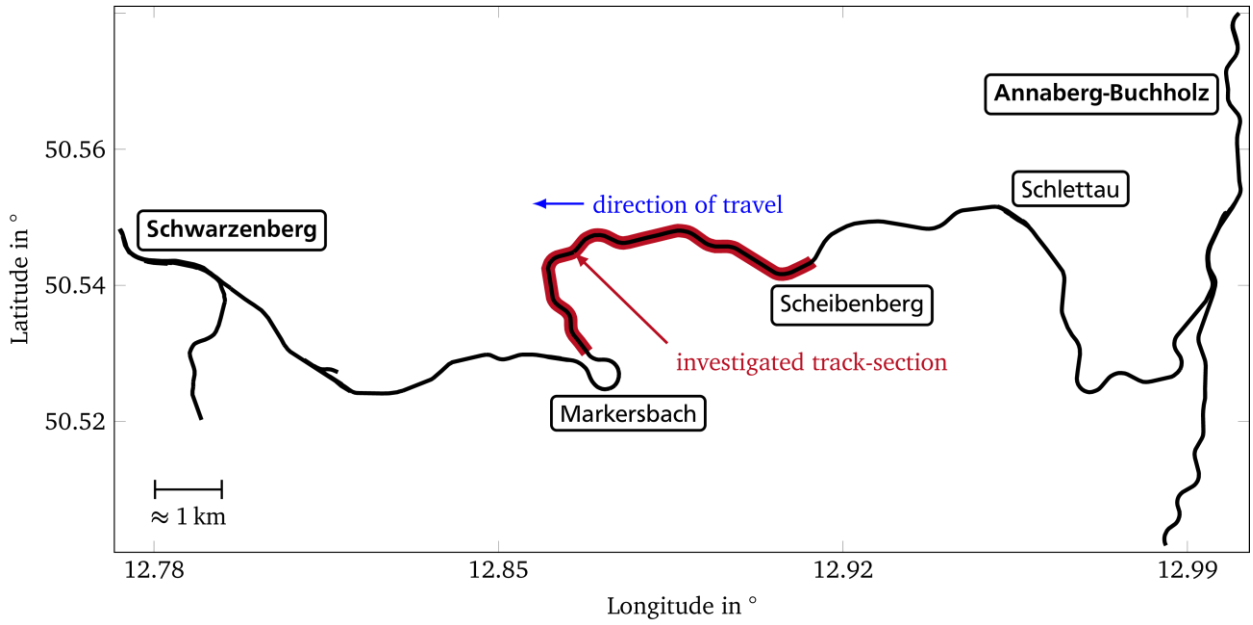

Figure 2 Overview of the track between Annaberg-Buchholz and Schwarzenberg

It is the 5.7 km long section from the railroad crossing in Scheibenberg to the end of the Markersbacher viaduct. All data for this paper was recorded with an iMAR iNAT-M200/STN sensor. It is a powerful MEMS based INS/GNSS navigation system. Here only the raw GNSS- and IMU-data[2] provided at a rate of 1Hz and 500Hz respectively were used to calculate a separate navigation solution with our algorithm. The GNSS antenna was installed at a roof platform at the front of the train as shown in Figure . The IMU was installed in the passenger cabin near the front doors, a bit in front of the bogie and slightly shifted to the right in the direction of travel (c.f. Figure).

The combination of GNSS and IMU is rather common in navigation applications as they complement each other very well. Most of the time the GNSS provides an absolute positioning solution which can be continued during GNSS outages with the help of the IMU data. Since the IMU data are drift afflicted the accuracy of the positioning solution decreases with time. Therefore, only GNSS outages of a few seconds (depending on the quality of the utilized IMU) can be tolerated in a train-borne localization system. Despite that, the combination of GNSS and IMU can be said to be one certain component of train-borne localization systems of the future which has to be supplemented by additional sensors [3,6]. Although reasonable, no additional sensors are considered since there were none available. However, the following evaluation will demonstrate how our algorithm helps to increase the positioning accuracy of a solely IMU/GNSS based localization system especially in bad GNSS situations and in cross-track direction.

[2] More precisely, the position, velocity and time (PVT) solution of the GNSS receiver, as well as the 6-DOF IMU (accelerations and turn rates) measurements, are used.

Figure 3 Front view of the test vehicle. The GNSS-Antenna was mounted on the marked roof platform

Figure 4 Measurement platform in the passenger cabin. The IMU/GNSS unit can be seen in the marked region

3 Analysis

In this section we evaluate our localization and mapping algorithm [1,2]. The evaluation is separated in three steps: First, some raw data are presented to get a feeling of the situation at hand and what kind of quality can be expected of the localization and mapping filter. Second, the localization results are evaluated and third, the mapping results are evaluated.

3.1 Sensor Data Preview

To get a feeling of the situation at hand some measurements which are thought to have the biggest influence on the localization and mapping results are shown in Figure . The presented measurements are the GNSS positioning error in terms of the standard deviation σ, the GNSS speed and the yaw rate of the vehicle. The first two measurements are provided by the GNSS receiver whereas the yaw rate is provided by the IMU.

In the top diagram of Figure the maximum 3σ GNSS positioning error related to a confidence level of 99.73% is presented. This error corresponds to a confidence level of 99.73% (assuming a normally distributed error) for the real position to be within the interval of ±3σ around the position estimate. The minimum 3σ error is 3.4m and the mean 3σ error is 25.3m. A valid GNSS solution was available around 95% of the time. The longest GNSS outage of 13s occurred at 268 – 281s. The degradation and losses of the GNSS signal mostly occur in places with dense forest around the track. Although this performance may seem poor compared to the absolute requirements in the sense of accuracy and availability for train-borne localization systems [7], it gives a good impression of the performance that can be expected of GNSS without additional correction data. It is worth to notice that the following evaluation of the localization performance is limited to the quality of these GNSS measurements as there are no better reference measurements available for this test drive. This is a general problem when evaluating train-borne localization systems. For this reason, we and others recommend to carry out more test drives and, by that, help to create extensive reference datasets for the evaluation of train-borne localization systems [6,8].

The diagram in the middle of Figure shows the speed measured by the GNSS receiver. It varies between 22km/h and 56km/h (neglecting the outliers around the main GNSS outage at 268s). This speed profile is typical, as it ranges from medium speeds to the maximum speed allowed on this track.

The bottom diagram of Figure shows the yaw rate of the vehicle. The data is available at all time since it is provided by the IMU. On straight track-elements the yaw rate varies around zero whereas it is non-zero on curved track-elements. This behavior can be observed very well in the course of the yaw rate as there are clear variations from zero. This can make it plausible how the investigated filter can identify the current track-geometries online, although the yaw rate is not the only feature utilized for this purpose in the filter.

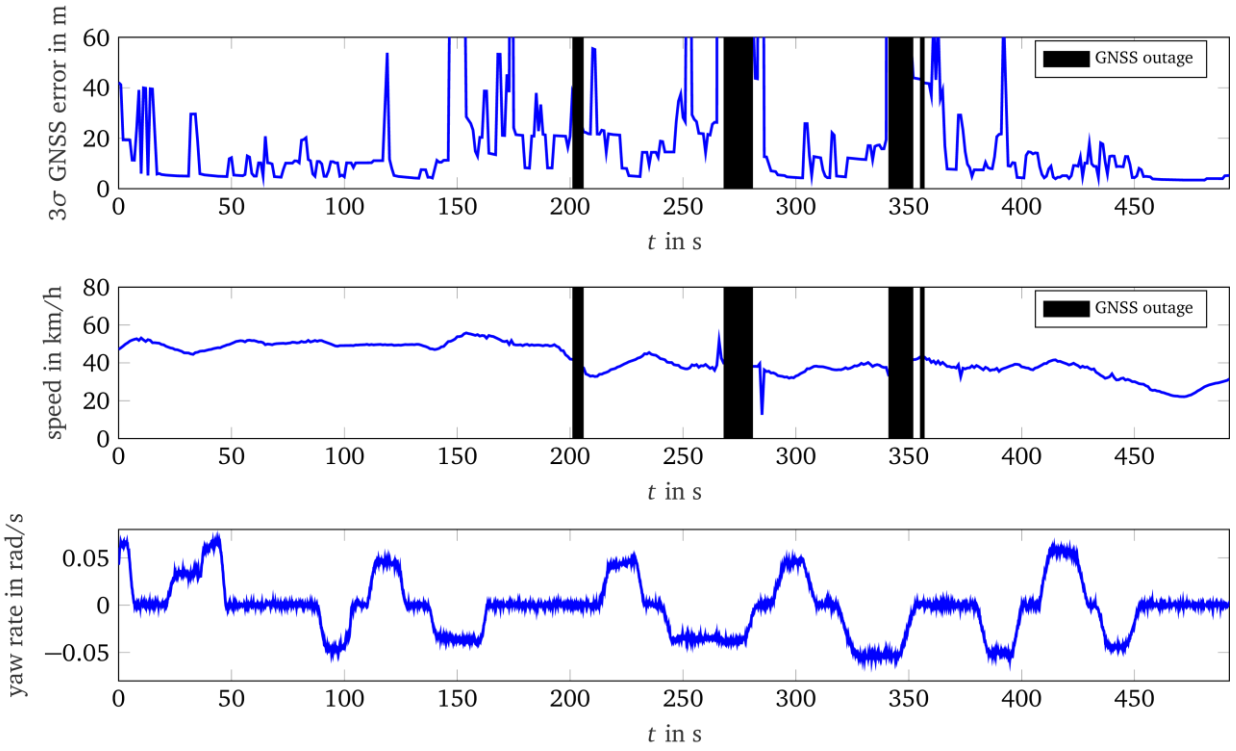

Figure 5 Raw sensor data of the GNSS receiver (top and center plot) and the IMU (bottom plot). Top: 3σ GNSS positioning error (corresponding to a confidence level of 99.73%) provided by the sensor itself. Center: Speed measurement provided by the GNSS receiver. Bottom: Yaw rate provided by the IMU

3.2 Localization

At first the localization accuracy is evaluated in the sense of the 3σ positioning error. In total three different positioning approaches are compared. These are the GNSS positioning solution, the sensor fusion result obtained with a standard Kalman filter (KF) approach and the sensor fusion result of our algorithm presented in [1]. The 3σ positioning error is calculated based on the covariance matrices provided by either the GNSS receiver or the sensor fusion filters.

In Figure the cumulative distribution function (CDF) of the 3σ positioning error is shown. It shows the availability of the positioning result over its accuracy. This makes it possible to easily compare the performance of the different localization approaches in the sense of accuracy and availability. Three different CDF plots are shown. From the left to the right these are the maximum 3σ positioning error

(left), the error in along-track direction (middle) and the error in cross-track direction (right). Some interesting data points of these diagrams are summarized in Table 1 and Table 2.

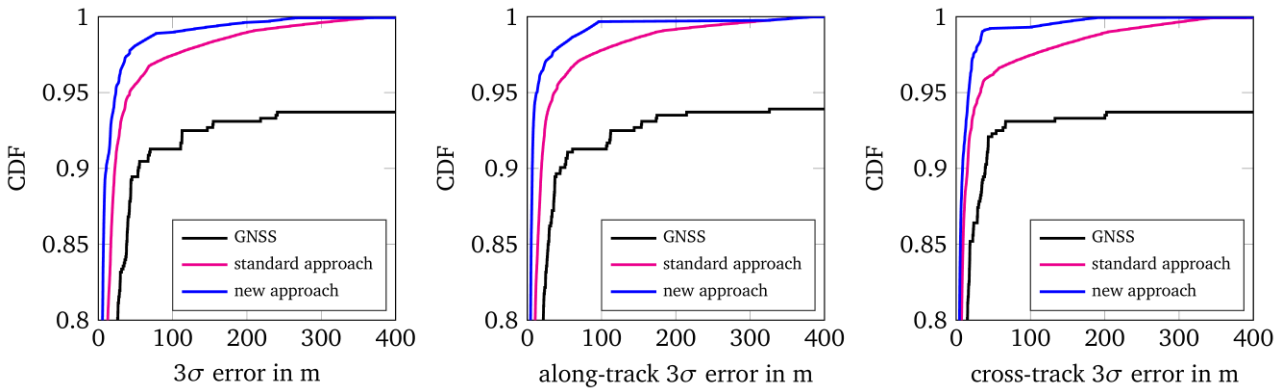

Figure 6 Cumulative distribution function (CDF) of the 3σ positioning error (corresponding to a confidence level of 99.73%). The error is calculated from the covariance matrices provided by either the GNSS receiver or the sensor fusion algorithms

By the plots the preliminary simulation results can be confirmed also for real measurement data. The new algorithm clearly increases the positioning accuracy compared to the standard KF sensor fusion approach. This is especially true for the error in cross-track direction (c.f. Figure and Table 1). A comparison of, e.g. the 0.99 CDF values (c.f. Table 1) shows that the new approach improves the positioning accuracy in along-track direction by approximately 55% and in cross-track direction by approximately 80% compared to the standard KF approach. A solely GNSS based positioning solution is not available for a CDF value of 0.99 due to the GNSS outages. In absolute numbers a GNSS positioning solution is only available for approximately 95% of the time, but this also includes very inaccurate positioning solutions with a 3σ error of more than 400 meters (c.f. Figure). In contrast to that the sensor fusion approaches are available all the time, which can be seen in the CDF plots due to the fact that both curves reach a CDF value of 1. Admittedly, this happens only for poor 3σ positioning errors (c.f. Table 1).

An exemplary comparison of the positioning solutions at fixed 3σ errors of respectively ±5m in along-track and ±1.5m in cross-track direction is shown Table 2. These values are either motivated by the allowed error for the localization with balises in ETCS, ±5m, or the theoretical value of ±1.5m to enable a track-selective localization. Again, the new algorithm shows an increased performance for both error values. The new algorithm at least allows a track-selective localization in 25% of the time (at a confidence level of 99.73%) whereas this is impossible with the other two approaches.

A qualitative evaluation of the localization results is shown in Figure . It shows the localization results of the standard KF approach and the new approach together with their corresponding 3σ error-ellipses in a phase with reduced GNSS performance. Additionally, the GNSS positioning results are shown but without their error-ellipses for the sake of clarity. All results are mapped on a satellite image.

The ellipses of the new approach are significantly smaller than the ellipses of the standard approach. This is a direct result of the track-geometry information which is incorporated in the new approach. It leads to a more constrained positioning uncertainty in cross-track direction. As this information is not available in the standard approach the GNSS uncertainty effects the localization result much more. As a reference, the track data from OpenStreetMap (OSM) [9] is plotted in Figure additionally. From that it can be seen that neither the GNSS measurements nor the localization results of both sensor fusion

approaches lie near the OSM track. Besides that, the OSM track does not even match with the track visualized on the satellite image. Therefore, it can be suspected that the OSM track is not very accurate in the viewed section. This may be important to have in mind throughout the following evaluation of the mapping results.

Table 1 3σ positioning error (corresponding to a confidence level of 99.73%) at specific values of the cumulative distribution function (CDF)

	CDF	3σ error in m		
		GNSS	standard	new
AT[3]	0.90	44.6	19.2	6.9
CT[4]	0.90	39.1	15.2	8.7
AT	0.99	inf	182.9	76.5
CT	0.99	inf	204.5	35.3
AT	1.00	inf	382.9	377.7
CT	1.00	inf	340.2	188.3

Table 2 Cumulative distribution function (CDF) at specific values of the 3σ positioning error (corresponding to a confidence level of 99.73%)

	3σ error in m	CDF		
		GNSS	standard	new
AT	±5m	0.314	0.523	0.833
CT	±1.5m	0.000	0.058	0.259

[3] AT: along-track

[4] CT: cross-track

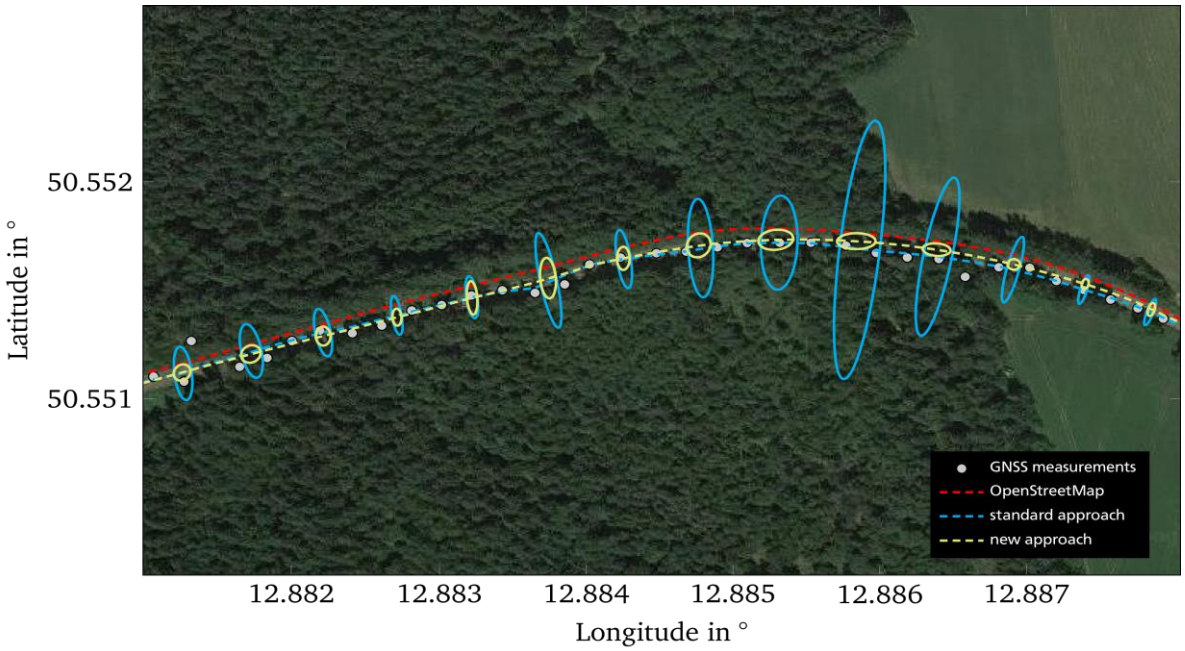

Figure 7 Visualization of the different positioning accuracies on a satellite image (Image © 2019 Google, Maps © 2019 GeoBasis-DE/BKG (© 2009), Google) during a phase of reduced GNSS performance. By comparison of the sizes of the 3σ error ellipses (corresponding to a confidence level of 99.73%) the increased accuracy in cross-track direction of the new approach can be seen clearly

3.3 Mapping

Next, the quality of the map generated with the new algorithm (c.f. Section 2.1and [2]) will be examined. Therefore, the results of the track-geometry identification process and the finally generated map are visualized on a satellite image in Figure .

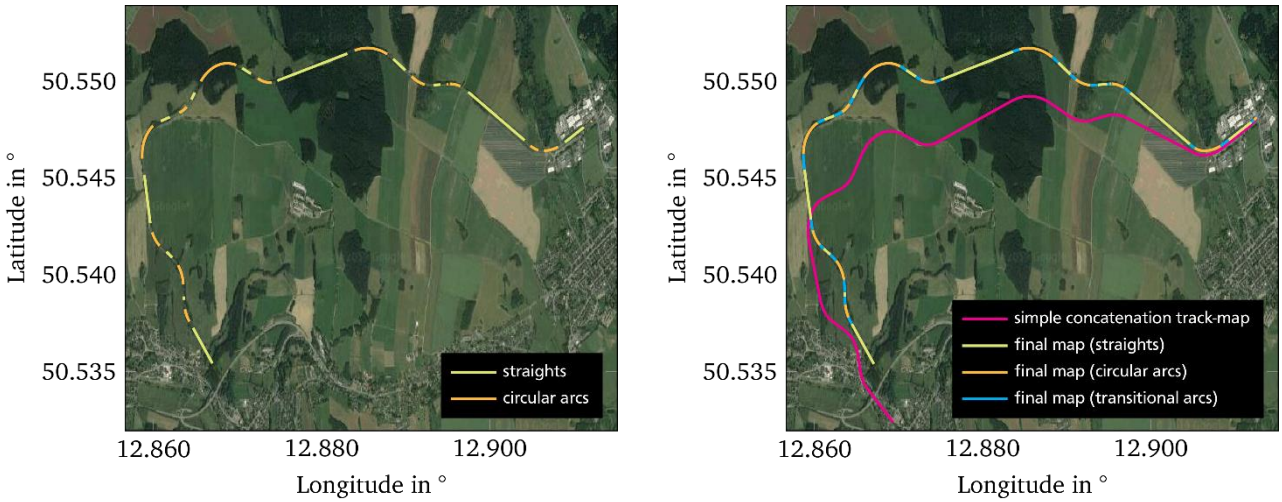

Figure 8: Visualization of the track-geometry identification process (left) and the finally generated map (right) on a satellite image (Image © 2019 Google, Maps © 2019 GeoBasis-DE/BKG (© 2009), Google).

Left: Visualization of the results of the second step in the localization and mapping filter (geometry-parameter identification). At this step only the parameters of straights and circular arcs are fully identified. Finding suitable parameters for the connecting transitional arcs is the task of the next step.

Right: Visualization of the mapping result. The final map consists of a continuous sequence of geometric track-elements, describing the real course of the track very well. The poor initial map results from the simple concatenation of the identified track-geometries which have been identified by the localization filter

The online identified track-geometry parameters are those of straights and circular-arcs. The resulting track elements are shown in Figure (left). It can be seen that all main straights and circular-arcs were identified quite accurate. This confirms the simulation results once again, i.e. the track-geometry parameter identification also works on real measurement data. In most cases the sequence "straight – gap – circular-arc – gap – straight" is present. Two times a sequence of "circular-arc – gap – circular-arc" occurs. This is best visible at the beginning of the track (take the direction of travel from east to west into account) in the first right turn. This is not a misdetection. Taking a look at the yaw-rate diagram in Figure between $20 - 50$s makes it plausible that at this point indeed two circular-arcs with different radii are following after each other. This is plausible due to the yaw-rate being at one non-zero level between $20 - 35$s and then changing to a higher level between $35 - 50$s without first returning to zero. The second time a "circular-arc – gap – circular-arc" sequence occurs at $100 - 105$s. Here the yaw-rate curve is not that clear. Only a small level-changing behavior can be seen. Thus, without construction plans it is difficult to tell if a misdetection has occurred or not. But for the following processing of the identified track-elements it is better to have several small track elements representing the true course of the track very well than having few track-elements with a rather poor fitting quality. In this sense it can be noted once again that the track-geometry parameter identification works well.

In the next step (c.f. Section 2.1) the identified track-elements are combined to a continuous total track. Therefore, the gaps between the identified track-elements are assumed to be transitional-arcs realized as clothoids. This can be assumed because of the construction principals of railway tracks [10]. Theoretically all parameters determining the connecting clothoids are already known from the localization and mapping filter. These parameters are: Starting point, starting direction, starting respectively ending radius of the clothoid and its length. However, the simple concatenation of all track-

elements including the clothoids gives a rather poor mapping result which can be seen in Figure (right). The problem is, that even small deviations in one of the first track-element's parameters have a huge impact on the position of all following track-elements due to the continuous concatenation of all tracks. Therefore, an optimization strategy is applied to find a good fit of all track-element parameters into the known positioning results [2]. The result is also shown in Figure (right). It can be seen that the track matches with the course on the satellite image in the background.

The final map parameters representing the whole track can be stored in a very compact form as exemplary shown in Table 3. This compact representation is worth the initial effort as it has some major advantages compared to the often used data-point based map representations. One advantage is its compactness itself, resulting in huge memory savings which in turn have many benefits, e. g. in wireless applications or situations where the map has to be searched through. Moreover, all geometric parameters of the track are directly accessible and can be used easily for further calculations, e. g. the calculation of the perpendicular distance of a point to the track.

Table 3 Excerpt of the identified compact geometric track-map

Track element number	1	2	3	4	5	6	7	8	9	10	11	...	53
Shape[5]	st	ta	ca	ta	st	ta	ca	ta	ca	ta	st	...	st
Length L in m	0	11	27	25	218	76	136	37	87	48	581	...	296
Radius r in m	∞	213	213	213	∞	376	376	197	197	197	∞	...	∞
Starting point	$p_0 = [50.548°\ 12.913°]^T$							$\psi_0 = 231.2°$					

From the previous qualitative evaluation, it can be stated that our mapping algorithm also works in practice as it did in the simulations. Despite that, next a quantitative evaluation of the mapping result should be carried out with the help of a reference track extracted from OSM data. In Figure the CDF of the absolute mapping error $|\varepsilon|$ in perpendicular direction from the calculated map to the OSM map is shown. Additionally, some significant values of $|\varepsilon|$ are listed in Table 4.

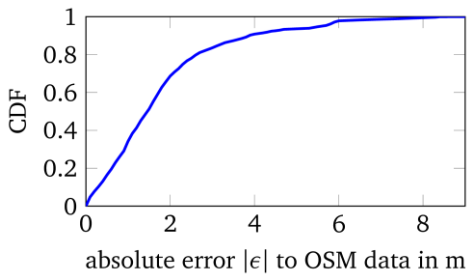

Figure 9 Cumulative distribution function (CDF) of the absolute error $|\varepsilon|$ in meters between the calculated map and the map from OpenStreetMap (OSM)

Table 4 Absolute error $|\varepsilon|$ in meters between the calculated map and the map from OpenStreetMap (OSM)

	mean	CDF		
		0.95	0.99	1.00
$\|\varepsilon\|$ in m	1.80	5.7	7.5	8.7

[5] st: straight, ta: transitional arc, ca: circular arc

The mean error of the calculated map to the OSM map is 1.8m and the maximum error is 8.7 m (c.f. Figure and Table 4). These values confirm that the calculated map is quite accurate. However, as mentioned above, also the OSM map can contain inaccuracies which make it difficult to generalize from the presented errors.

3.4 Concluding Remarks for Train-Borne Localization Systems

Based on the above analyses we want to make some remarks concerning further steps when developing train-borne localization systems of the future.

We examined the localization quality in the sense of the 3σ positioning error. This corresponds to a confidence level of 99.73% for the real position to be within a range of ±3σ. However, this confidence level is by far not acceptable for safety critical railway applications in the common sense of EN 50126 [4]. Therefore, presumably the 6σ positioning error would have had to be considered. If this confidence level had been considered here, none of the here presented localization techniques would have provided a track-selective localization result ever. That raises the question if it is even possible to achieve a track-selective localization result with any combination of the available sensors for train-borne localization systems under the current circumstances of EN 50126 [4].

We think that it is very likely impossible to achieve a valid safety prove by mainly focusing on the sensor configuration for train-borne localization systems. Instead also new methods to assess the safety of train-borne localization systems and the dynamic overall systems associated with them should be investigated further. This becomes clearer with the following explanations: In the above statements it was implicitly assumed that a track-selective localization has to be possible based on a single localization result. This must not necessarily be the case as the determination of the correct track can also be treated as a dynamic process. Additionally, as the uncertainty of the positioning solution could be estimated at any time, it would make sense to dynamically account for it in the current movement authorities of all trains. This principal would enable a safe operation independent of the absolute size of the current uncertainties. By that it becomes plausible that it makes sense to develop customized methods and measures for a safety prove of train-borne localization systems.

Additionally, in order to develop suitable safety methods and safety measures we think it is essential to carry out much more real tests like the one presented here. Only by that all the short and long-term effects influencing the performance of train-borne localization systems can be identified and treated accordingly. Therefore, we and others already recommended to create extensive reference datasets for the evaluation of train-borne localization systems [6, 8].

4 Summary

In this paper we investigated the performance of a fully train-borne localization and mapping filter we presented earlier. Until now the performance has only been investigated using simulations. In contrast to that, in this work the performance has been evaluated with the help of real GNSS and IMU measurement data collected on a test drive. By that, our preliminary simulation results could be confirmed. The new approach is capable of increasing the localization accuracy which is especially true for the accuracy in cross-track direction and during bad GNSS situations. Moreover, an accurate compact geometric track-map could be generated confirming the preliminary simulation results once again. Beside the evaluation of our localization and mapping filter we gave some concluding remarks

concerning the further development of train-borne localization systems of the future. According to that we recognized a great need for developing new safety methods specifically designed for train-borne localization systems. Additionally, we suggest to carry out much more real tests and, by that, to build extensive reference datasets for the evaluation of train-borne localization systems.

5 Acknowledgements

We kindly thank DB Netz AG for supporting this research project. Furthermore, we like to thank Thales affording us to collect the raw data with their test vehicle LUCY, and the group of Geodetic Measurement Systems and Sensors at TU Darmstadt for providing the IMU/GNSS sensor platform.

6 References

[1] H. Winter, V. Willert, and J. Adamy, “Increasing accuracy in train localization exploiting track geometry constraints,” in Proc. IEEE ITSC, 2018, pp. 1572–1579.

[2] H. Winter, S. Luthardt, V. Willert, and J. Adamy, “Generating compact geometric track-maps for train positioning applications,” in Proc. IV, 2019, accepted.

[3] J. Otegui, A. Bahillo, I. Lopetegi, and L. E. Díez, “A survey of train positioning solutions,” IEEE Sensors Journal, vol. 17, no. 20, pp. 6788–6797, 2017.

[4] EN50126 - Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS), Eur. Committee for Electrotechnical Standardization

[5] K. Gerlach and M. M. zu Hörste, “A precise digital map for galileo-based train positioning systems,” in Proc. IEEE ITST, 2009, pp. 343–347.

[6] “smartrail 4.0: Zwischenbericht Technologie PoC Lokalisierung,” Jan. 2019. [Online]. Available: https://smartrail40.ch/index.asp?inc=downloads.asp&typ=Nav2&cat=5310

[7] M. Grimm, K. Hartwig, and M. M. zu Hörste, “Anforderungen an eine sicherheitsrelevante Ortung im Schienenverkehr,” in 20. Verkehrswissenschaftliche Tage, TU Dresden, 2005. [Online]. Available: https://elib.dlr.de/20857/

[8] H. Winter, V. Willert, J. Adamy, M. Leining, M. Spindler, M. Lauer, D. Stein, O. Heirich, J. Groos, A. Geffert, U. Becker, and M. Breuer, “Localization Reference Train – Sichere Ortung für den Schienenverkehr,” in Scientific Railway Signalling Symposium, 2017, pp. 17–26. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/106619/

[9] OpenStreetMap contributors, “Extract retrieved from https://overpass-api.de,” https://www.openstreetmap.org, 2019.

[10] J. Haldor and F. Lademann, „Planung von Bahnanlagen: Grundlagen-Planung-Berechnung,” 2nd ed. Carl Hanser Verlag GmbH & Co. KG, 2017.

Zeit- und Kostenersparnis bei der ETCS L2 Planung durch Digitalisierung[1]

Stefan Dillmann[2], Miroslav Pejic[3], Andreas Oetting[4] und Reiner Hähnle[5]

TU Darmstadt

1 Einleitung

Die Einführung des europäischen Zugsicherungssystems ETCS (European Train Control System) in Deutschland stellt eine gewaltige Herausforderung für die DB Netz AG als verantwortlichen Infrastrukturbetreiber dar. Zum einen führen bedeutende transeuropäische Korridore durch Deutschland (insbesondere Korridor A: Rotterdam-Genua) und erfordern eine Hochrüstung auf ETCS, um Interoperabilität zu gewährleisten. Zum anderen wurde das vor allem auf Hochgeschwindigkeitsstrecken vorhandene nationale Zugsicherungssystem LZB (Linienförmige Zugbeeinflussung) vom Hersteller abgekündigt, wodurch auf diesen Teilstrecken die Migration auf ETCS in der nahen Zukunft erforderlich wird. Neubauprojekte müssen gemäß EU-Richtlinien ohnehin mit ETCS ausgerüstet werden. Insgesamt muss daher in den kommenden Jahren eine große Anzahl von ETCS-Strecken geplant werden (BMVI).

Gegenwärtig wird der ETCS-Planungsvorgang größtenteils von Hand durchgeführt und es existiert nur wenig Werkzeugunterstützung. Eine wesentliche Unterstützung gibt es durch die Verwendung von CAD-Programmen und Symbolbibliotheken im Zeichenvorgang (untere Zeile in Tab. 1). Die Anordnung der zu planenden Elemente und Berechnung der korrekten Position erfolgt individuell und manuell. Darüber hinaus können die Zeichenprogramme nicht überprüfen, ob der erstellte Plan konform zu den ETCS-Planungsrichtlinien (Deutsche Bahn AG) ist. Dies muss in einem aufwändigen Planprüfungsprozess bewerkstelligt werden. Daher ist im Moment die Planung der ETCS-Ausrüstung von Strecken ein zeitaufwändiger, kostspieliger und fehleranfälliger Prozess.

In den letzten Jahren hat die DB Netz AG das XML-basierte Format PlanPro (Maschek, Klaus, Gerke, Uminski, & Girke, 2012) als künftigen Standard für die digitale Planung von LST-Anlagen entwickelt. Zurzeit beginnt die Einführung von PlanPro in kommerziellen Planungswerkzeugen wie ProSig[6]. Im Gegensatz zu den reinen Zeichnungsinformationen der CAD-Dateien enthält PlanPro zusätzlich *semantische* Informationen (mittlere Zeile in Tab. 1), d.h. Signale werden auch als Signale mit Signaleigenschaften modelliert und nicht als Komposition von Kreisen und Linien. Dies ermöglicht die Modellierung von Beziehungen der Elemente untereinander, sowie einfache Plausibilitätsprüfungen. PlanPro ermöglicht darüber hinaus den Datenaustausch zwischen verschiedenen Anwendungen und damit die *Interoperabilität* auf semantischer Ebene, beispielsweise können die erforderlichen

[1] Darstellung basiert in Teilen auf S. Dillmann & R. Hähnle, Automated Planning of ETCS Tracks, RSSRail 2019, Lille (Dillmann & Hähnle, 2019)

[2] dillmann@cs.tu-darmstadt.de

[3] pejic@verkehr.tu-darmstadt.de

[4] oetting@verkehr.tu-darmstadt.de

[5] haehnle@cs.tu-darmstadt.de

[6] http://www.ivv-gmbh/de/prosigr.html

Datentabellen für den Planprüfungsprozess mit dem Programm „PlanPro-Werkzeugkoffer“ automatisch generiert werden.

Tab. 1 Informationsebenen der Planungsdomäne

Ebene	Informationstyp	Funktionen	Art des Tools	Beispiel
Pragmatik	Algorithmen	Automatisierung	Planer, Validierer	FormETCS-Tools
Semantik	Funktion, Zweck	Interoperabilität	OODB, XML	ProSig
Fakten	Geometrie, ID	Zeichnen, Visualisieren	CAD	AutoCAD

Als nächster Schritt wurde von der DB Netz AG das Projekt „FormETCS“ ins Leben gerufen, in dem die Möglichkeiten untersucht werden, einen Großteil der Planungsprozesse zu automatisieren. Dazu ist der Automatisierungsprozess in drei Phasen unterteilt: (1) Datenaufnahme und Datenprüfung, (2) automatisierte Planung und (3) automatisierte Planprüfung. Als Datengrundlage soll ein digitales Gleislayout im PlanPro-Format verwendet werden. Für die automatisierte Planung und automatisierte Planprüfung sollen passende Algorithmen entwickelt und jeweils in einem Demonstrator implementiert werden (obere Zeile in Tab. 1). Das Planungstool soll eine PlanPro-Datei, die lediglich Gleislayout und einige grundlegende LST-Elemente enthält, als Eingabe verwenden und darauf verschiedene Algorithmen anwenden, die das vorhandene Modell um zusätzliche ETCS-Elemente erweitern. Anschließend kann mit dem Prüfungstool das modifizierte Modell mit Hilfe unabhängiger Algorithmen auf Konformität zu den ETCS-Planungsrichtlinien überprüft werden. Da sowohl Eingabe- als auch Ausgabedatei im PlanPro-XML-Format vorliegen, können diese auch mit anderen Werkzeugen weiterverarbeitet werden. Diese Interoperabilität hat entscheidende Vorteile:

- Da das Planungstool ein bereits vorhandenes Gleislayout mit optionaler konventioneller Signalisierung als Eingabe benötigt, ist es ideal dazu geeignet, um bestehende Strecken auf ETCS hochzurüsten, das dann parallel zum vorhandenen Sicherungssystem betrieben werden kann.
- Die Ausgabedatei kann durch das offene Format weiterverarbeitet werden, etwa um manuelle Anpassungen vorzunehmen.
- Das Planprüfungstool kann auch manuell überarbeitete Pläne überprüfen, auch eine Überprüfung komplett händisch erstellter Pläne ist möglich.
- Existierende PlanPro-kompatible Werkzeuge wie ProSig und der PlanPro-Werkzeugkoffer können zur Erstellung der Eingabedatei und zur Weiterverarbeitung der Ausgabedatei verwendet werden, etwa um Datentabellen für den Planprüfungsprozess zu generieren.
- Da PlanPro als Standardformat für die digitale LST-Planung vorgesehen ist, können die Tools in den dafür angedachten Workflow integriert werden.

In dieser Arbeit werden die Konzepte zur automatisierten Planung und Planprüfung beschrieben, außerdem einige Strategien zu Test und Validierung. In Kapitel 2 wird ein kurzer Überblick über die zugrundeliegenden ETCS-Planungsrichtlinien gegeben, in Kapitel 3 das PlanPro-Datenmodell beschrieben, in Kapitel 4 Konzept und Implementierung des Planungstools und in Kapitel 5 das Konzept der Planprüfung. In Kapitel 6 werden einige Strategien zur Validierung diskutiert, in Kapitel 7 verwandte Arbeiten vorgestellt und ein Ausblick auf künftige Entwicklungen gegeben.

2 ETCS-Planungsrichtlinien

Die ETCS-Planungsrichtlinien sind Teil der Richtlinie (Ril) 819: LST-Anlagen planen (Deutsche Bahn AG). Sie gliedern sich in sieben Module:

- 819.1340 – Grundsätze für die Ausrüstung von Strecken mit ETCS;
- 819.1341 und 819.1342 – Entwurfs- und Ausführungsplanung Planteil 1 (PT1) für Level 1 Vollüberwacht;
- 819.1343 – Vorgaben für die qualifizierte Ausführung der Entwurfsplanung zur Ausrüstung von Strecken mit ETCS Level 2;
- 819.1344 – Ausführungsplanung PT1 für Strecken mit ETCS Level 2 und Level 2 ohne Signalisierung;
- 819.1345 und 819.1346 – Entwurfs- und Ausführungsplanung PT1 für Strecken mit ETCS Level 3;
- 819.1347 – zusätzliche Hinweise bei der Doppelausrüstung von Strecken z.B. mit LZB und ETCS Level 2;
- 819.1348 – Vorgaben zur Erstellung der Ausführungsplanung für Strecken mit ETCS Level 1 Signalgeführt.

Alle genannten Richtlinien erfordern eine bereits vorhandene Stellwerksplanung und bauen auf dieser auf. Insbesondere müssen Weichen, Gleisfreimelder, Blockabschnitte und Fahrstraßen vorhanden sein. Bei Planung einer ETCS-Strecke mit zusätzlicher konventioneller Signalisierung müssen auch die Signale bereits vorhanden sein. Diese Objekte dienen als Referenzpunkte für die Verortung von ETCS-Datenpunkten.

Der Fokus dieser Arbeit liegt auf dem Inhalt des Moduls 819.1344. In diesem Modul sind unter anderem Planunterlagen, Vorgaben zu Datenpunktorten, GSM-R, Level 2 ohne konventionelle Signalisierung, Wechsel zwischen Level 2 und Level 1, ETCS-Adressen und Auswirkungen auf Nachbargewerke beschrieben.

Das Modul definiert insbesondere verschiedene Typen von Datenpunkten. Dabei handelt es sich um Balisengruppen, die eine bestimmte Funktion erfüllen und einen gemeinsamen Satz an ETCS-Datenpaketen enthalten. Die Verortung von Datenpunkten eines Typs erfolgt stets nach den gleichen Regeln und wird in den meisten Fällen als Abstandswert zu einem Referenzobjekt (z.B. dem zugehörigen Signal) definiert[7].

Die Regeln im Modul 819.1344, wie auch in den anderen Modulen, sind in natürlicher Sprache formuliert. Die textuelle Darstellung ist durch Tabellen und Formeln ergänzt. Die Abstände zwischen zwei bestimmten Feldelementen können folgendermaßen definiert sein:

- Als ein fester Wert (evtl. mit Einbautoleranzen).
- Als eine Formel, wenn weitere Einflussgrößen betrachtet werden müssen.
- Als Tabellenwert, wenn die Einflussgrößen und deren Intervalle parallel betrachten werden müssen (Abb. 1).

[7] Die Kilometrierungsangabe der Datenpunkte ist nur informativ und dient zur besseren Orientierung.

D_{End} [m]	900 m ≥ d_{LRBG_ESig} ≥ 740 m	740 m > d_{LRBG_ESig} ≥ 600 m	600 m > d_{LRBG_ESig} ≥ 400 m	400 m > d_{LRBG_ESig} ≥ 200 m
$D_{End} \leq 55$	486 / 450	555 / 476	749 / 670	949 / 870
$55 < D_{End} \leq 100$	535 / 457	559 / 481	749 / 670	949 / 870
$100 < D_{End} \leq 200$	646 / 567	630 / 551	755 / 676	949 / 870
$200 < D_{End} \leq 300$	756 / 677	740 / 661	765 / 687	949 / 870

Abb. 1 Mindestabstand dESig_Folgesig_2 und Stellwerk meldet irregulären Haltfall
Quelle: (Deutsche Bahn AG)

Alle diese Regeln sind deklarativ beschrieben und bilden eine Reihe von Bedingungen, die jeder endgültige Plan ganzheitlich erfüllen muss. Im Modul sind dagegen keine Algorithmen zu finden, wie ein Planer von Grund auf schrittweise vorgehen muss, bis er einen gültigen Plan erhält. Die Regelwerke sind somit eher aus der Sicht eines Planprüfers als eines Planers beschrieben. Ein Planprüfer kann daher direkt mit dem Regelwerk arbeiten, indem er für jeden vorhandenen Datenpunkt überprüft, ob alle für diesen Typ spezifizierten Bedingungen eingehalten sind und ob alle erforderlichen Datenpunkte vorhanden sind. Bei Planern funktioniert dieses Vorgehen nicht, da mit jedem neu hinzugefügten Datenpunkt die Bedingungen für bereits vorhandene Datenpunkte verletzt werden können (z.B. durch zu geringe Abstände untereinander). Erfahrene Planer haben daher Strategien erarbeitet, wie die Datenpunkte platziert werden können, damit das Konfliktpotential minimiert wird. Sie erfüllen ihre Aufgaben mittels eines iterativen Prozesses: Datenpunkte werden solange auf dem Plan platziert, bis kein weiterer Datenpunkt möglich ist. Dann muss man einen früheren Planungsschritt verändern und neu versuchen, bis der Plan vollständig und gültig ist. Planer und Planprüfer haben somit unterschiedliche Vorgehensweisen und dadurch auch unterschiedliche Sichten auf ein und dasselbe Regelwerk. Daraus resultieren auch die unterschiedlichen Ansätze bei der Konzeption von Algorithmen für den Planungsprozess (Kap. 4) und den Planprüfungsprozess (Kap. 5).

Es ist wichtig zu wissen, dass die Regeln in (Deutsche Bahn AG) reine Richtlinien sind und keinen Gesetzescharakter haben. Dies bedeutet, dass es möglich ist, von den Regeln abzuweichen, wenn sie bei einer bestimmten Infrastruktur nicht erfüllt werden können.

3 PlanPro-Datenmodell

Eine digitale LST-Planung benötigt ein Datenmodell, das semantische Informationen über die Funktion der Feldelemente und ihre Beziehungen untereinander enthält. Eine reine CAD-Datei ist dazu nicht geeignet, da sie nur Zeichnungsinformationen wie Linien, Kreise und einfache geometrische Formen enthält, diese aber ohne semantische Informationen keine übergeordnete Bedeutung (etwa Weichen oder Signale) haben.

Als Datengrundlage für FormETCS wird das PlanPro-Objektmodell (Maschek, Klaus, Gerke, Uminski, & Girke, 2012) verwendet, das von der DB Netz AG für die digitale Planung von LST-Anlagen entwickelt wurde. PlanPro ist ein objektorientiertes Modell und definiert Klassen für alle Elemente, die im LST-Planungsprozess vorkommen (DB Netz AG). Dazu gehören die Gleisgeometrie (modelliert als Graph mit Knoten und Kanten in mehreren Abstraktionsebenen), Weichen (einschließlich der mechanischen und elektrischen Komponenten), Signale, Blöcke, Fahrstraßen, Gleisfreimelder und Zugsicherungssysteme

(PZB-Gleismagnete und ETCS Balisen). Jede Klasse definiert verschiedene Attribute, welche die Eigenschaften der daraus erzeugten Objekte beschreiben. Beispielsweise haben Signale einen Typ (z.B. Hauptsignal, Sperrsignal), eine Funktion (z.B. Einfahrsignal, Ausfahrsignal) und ein System (z.B. H/V, Ks). Darüber hinaus hat jedes PlanPro-Objekt eine eindeutige ID, mit der Querverweise zwischen den Objekten realisiert werden. Die Klassen in PlanPro bilden eine Hierarchie, und Unterklassen erben die Eigenschaften der übergeordneten Klassen. Beispielsweise erben alle Klassen in PlanPro, die eine Positionsinformation benötigen (z.B. Signale), von der gemeinsamen Oberklasse Punkt_Objekt, in der die zur Positionierung notwendigen Attribute definiert sind.

PlanPro Objekte werden als unsortierte lineare Sequenz verwaltet. Beziehungen unter den Objekten werden durch Verweise auf die zugehörige Objekt-ID realisiert (bei der es sich um eine eindeutige 128 Bit UUID gemäß RFC 4122 handelt). Die Sequenz kann als XML-Datei abgespeichert werden und somit zum Datenaustausch benutzt werden.

PlanPro wurde ursprünglich als reines Datenaustauschformat zwischen den unterschiedlichen Schritten im Planungsprozess entwickelt (z.B. zwischen Planern und Planprüfern oder Planern und der Signalbauindustrie). Die Hauptanwendung ist die Erstellung von Übersichtsplänen und Datentabellen für den Planprüfungsprozess und die Vermeidung mehrfacher (redundanter) Datenhaltung durch unterschiedliche proprietäre Datenformate in jedem Planungsabschnitt. Eine Automatisierung des Planungsprozesses stand bei der Entwicklung des Formats nicht im Fokus, daher war auch eines der Ziele dieser Arbeit zu evaluieren, ob das PlanPro Modell genügend Informationen hierfür besitzt und auch von der Struktur her für die Automatisierung geeignet ist.

4 Automatisierte Plangenerierung

4.1 Entwicklung des Algorithmus

Voraussetzung sowohl für die automatisierte Planung als auch Planprüfung ist die Verfügbarkeit eines formalen und digitalen Modells sowohl des Gleislayouts (das mit dem PlanPro-Format bereits vorhanden ist) als auch der anzuwendenden Regeln (die noch informell in Papierform vorliegen). Ein möglicher Lösungsansatz wäre, die Planungsregeln deklarativ zu formalisieren, etwa als logische Bedingungen, und dann mit Hilfe einer logischen Programmiersprache eine systematische, vollständige Suche durchzuführen, die den Streckenplan so lange modifiziert, bis alle Bedingungen erfüllt sind und die gefundene Lösung somit gültig ist. Solch ein ungesteuertes Vorgehen kann besonders bei der Streckenplanung sehr ineffizient sein, da es bei der Lösungssuche zu einem hohen Verzweigungsfaktor kommt. Darüber hinaus ist ein auf diese Weise generierter Plan, der formal „gültig“ ist, nicht zwangsläufig auch als „gut“ oder intuitiv für einen Menschen zu betrachten. Das kann besonders bei einer noch von Menschen durchzuführenden finalen Planprüfung von Nachteil sein, wenn man einen „unnatürlich“ aussehenden Plan erst einmal zeitintensiv nachvollziehen muss. Außerdem ist die Integration von Optimierungen (z.B. Minimierung der Anzahl benötigter Balisen) in einen solchen Ansatz komplizierter.

Es wurde daher der Entschluss gefasst, anstatt einer ungesteuerten Suche einen heuristischen Konstruktionsalgorithmus zu verwenden, der auf der Erfahrung und dem Wissen menschlicher Planer basiert. Anders ausgedrückt werden nicht die Planungsregeln, sondern die heuristische Vorgehensweise der menschlichen Planer formalisiert. Die Planungsregeln selbst sind zwar noch vorhanden, aber eingebettet in den Planungsalgorithmus und nicht als solche erkennbar. Ebenfalls kommt von der Vielzahl an Regeln der Ril 819.1344 [2] nur ein kleiner Teil bei einer konkreten Planung tatsächlich zur

Anwendung, dann aber mehrfach innerhalb ein und desselben Plans, während andere Regeln für Spezialfälle vorgesehen sind, die man normalerweise zu vermeiden versucht. Zudem beschreibt ein Großteil der Richtlinien formale Anforderungen an den Plan als Dokument, was im Rahmen dieses Projektes komplett ignoriert werden kann[8].

Eine heuristische Vorgehensweise funktioniert nur in einer sehr spezifischen Domäne, in der das Wissen und die Vorgehensweise der menschlichen Experten in einem ausreichenden Maß eindeutig sind. Am Beginn der Arbeit war nicht ersichtlich, ob dies der Fall ist. Daher wurde eine Reihe von Interviews mit erfahrenen ETCS-Planern von DB Engineering & Consulting durchgeführt. Das Ziel war es herauszufinden, wie ein Planer einen Plan von Null beginnend konstruiert, und welche Planungsschritte in welcher Reihenfolge durchgeführt werden. Es musste sichergestellt werden, dass die Vorgehensweise der Planer ausreichend strukturiert und eindeutig ist, um algorithmisch abgebildet werden zu können. Ebenfalls musste in Erfahrung gebracht werden, welche Planungsschritte zu automatisieren sind und welche zunächst weiterhin von Hand erfolgen sollen.

Die wesentliche Erkenntnis hierbei war, dass die Planer in einer sehr strukturierten Weise arbeiten, in der der Planungsprozess in klar trennbare sequenzielle Phasen unterteilt wird, zwischen denen kein Backtracking auftritt. Daraus ergibt sich eine implizite Hierarchie der zu platzierenden ETCS Elemente, die so nicht aus dem Regelwerk ersichtlich ist.

Für eine Planung mit ETCS Level 2 kann daraus die folgende Vorgehensweise formuliert werden:

1. Platzierung der Datenpunkte, die direkt an einem Hauptsignal[9] positioniert sind. Dazu gehören die „Datenpunkte an Signalen“ (Typ 20) und „Datenpunkte an Ausfahrsignalen“ (Typ 21), die beide eine Vorbeifahrt im Mode „Staff Responsible“ (SR) verhindern, und bei Typ 21 zusätzlich die GSM-Verbindung zur ETCS-Zentrale aufbauen sowie überprüfen, ob der auf dem Fahrzeug aktive Level auch auf der Strecke vorhanden ist.
2. Platzierung des „Ersten Ortungs-Datenpunktes vor Signalen“ (Typ 23), der 300 m vor einem Signal zu platzieren ist und den Ortungsfehler des Zuges bei der Anfahrt auf das Signal (bei dem die Fahrerlaubnis enden kann) zurücksetzt.
3. Platzierung der „Datenpunkte für Start-of-Mission (SoM)“ (Typ 28) zur Absicherung von startenden Zugfahrten ohne gültige Positionsinformation. Ein SoM-Bereich wird vor Ausfahrsignalen eingerichtet und enthält bis zu 8 separate Datenpunkte (mit bis zu 14 einzelnen Balisen), deren Position abhängig von der Entfernung des Ausfahrsignal zur darauffolgenden Weiche sowie der maximal erlaubten Geschwindigkeit im Gleis ist. Die Regeln enthalten viele Fallunterscheidungen und Sonderregeln, wodurch eine manuelle Verortung aufwändig und fehleranfällig ist. Ebenfalls beschreiben die Regeln Kombinationsmöglichkeiten mit dem „Zweiten Ortungs-Datenpunkt vor Signalen“ (Typ 24, siehe nächster Punkt), der in diesen Fällen ebenfalls bereits in diesem Schritt platziert wird.
4. Platzierung der Datenpunkte vom Typ 24 als Einzel-Datenpunkt, wenn keine Kombinationsmöglichkeiten mit Typ 28 gegeben sind. Diese sind dann 50 m vor dem Signal zu platzieren und setzen erneut den Ortungsfehler des Zuges zurück.

[8] Eine vollständig digitale LST-Planung benötigt keine Papierpläne und demzufolge auch keine formalen Anforderungen an solche. Es ist lediglich die korrekte Befüllung des PlanPro-Datenmodells zu gewährleisten. Daraus kann der „PlanPro-Werkzeugkoffer“ Papierpläne erzeugen, so dass nur dieser die dazu notwendigen formalen Vorschriften zu implementieren hat.

[9] Bei einer Planung für ETCS Level 2 ohne konventionelle Signalisierung (L2oS) werden stattdessen ETCS-Halt-Tafeln (Ne 14) sowie Blockkennzeichen als Referenzpunkte herangezogen (Deutsche Bahn AG).

5. Platzierung der „Temporary Speed Restriction (TSR) Datenpunkte“ (Typ 26), die die Geschwindigkeit bei Anfahrt auf ein Signal im Mode SR begrenzen. Dies ist erforderlich für Signale, bei denen der Gefahrpunktabstand kleiner als 325 m ist. Die Position hängt von der Streckenneigung ab und der Datenpunkt muss wiederholt werden, wenn andere Signale oder spitz befahrene Weichen zwischen Datenpunkt und Bezugssignal liegen.
6. Platzierung der „Datenpunkte zur TSR Rücknahme“ (Typ 37) auf allen Verzweigungen ausgehend von Datenpunkt 26, die nicht zum geschützten Signal führen.
7. Platzierung der „Allgemeinen Ortungs-Datenpunkte“ (Typ 25) zur Erkennung des Fahrwegs bei gestörten Weichen. Im Fall einer gestörten Weiche, bei der die ETCS-Zentrale die tatsächliche Weichenlage nicht feststellen kann, muss die restriktivste Movement Authority (MA) ausgestellt werden, was dazu führen kann, das die Fahrerlaubnis auf dem längeren Weg vorzeitig endet. Aus diesem Grund können zusätzliche Datenpunkte erforderlich sein, um den tatsächlich gefahrenen Weg frühzeitig zu erkennen und die MA rechtzeitig verlängern zu können. Die zusätzlichen Datenpunkte von Typ 25 werden nur dann verwendet, wenn eine rechtzeitige Bestimmung des gefahrenen Weges mit bereits vorhandenen Datenpunkten nicht möglich ist. Die manuelle Berechnung dazu ist komplex und fehleranfällig.
8. Platzierung von Ortungs-Datenpunkten von Typ 25 zwischen zwei aufeinanderfolgenden bereits verorteten Datenpunkten, wenn diese einen Abstand von mehr als 1800 m haben. Dies ist notwendig zur Verringerung der Unsicherheit bei Positionsmeldungen der Fahrzeuge.

In all diesen Schritten wird die Liste der PlanPro-Objekte durchlaufen, bis ein potenzieller Referenzpunkt für einen Datenpunkt gefunden wird. Anschließend werden die Positionen berechnet, an denen Datenpunkte platziert werden müssen (wobei mögliche Verzweigungen in der Gleisgeometrie berücksichtigt werden), die Datenpunktobjekte erzeugt und der Liste hinzugefügt. Zum Beispiel wird in Punkt 8 die Liste nach bereits vorhandenen Datenpunkten durchsucht, deren Nachbarn bestimmt und die Abstände berechnet. Wenn ein Abstand größer als 1800 m ist, wird der Pfad wiederholt in kleinere Untersegmente gleicher Größe unterteilt und neue Datenpunkte von Typ 25 dazwischen platziert, bis keine Lücken größer als 1800 m mehr vorhanden sind.

Es wurde entschieden, zunächst keine weiteren Datenpunktypen zu betrachten: Diese können entweder einfach manuell verortet werden, sind nur in Spezialfällen erforderlich oder erfordern zusätzliche manuelle Angaben. Beispiele sind die Datenpunkte für Ein- und Ausstieg, die eine Festlegung des Ortes durch den Auftraggeber der jeweiligen Planung erfordern und daher nicht vollautomatisch bestimmt werden können.

Bei der automatisierten Planung kann die Situation auftreten, dass kein gültiger Plan generiert werden kann, beispielsweise wenn die vorhandene Infrastruktur auch unter Ausnutzung aller Planungsspielräume keine konfliktfreie Platzierung der Datenpunkte ermöglicht[10]. In diesem Fall darf der Planer vom Regelwerk abweichen und muss seine Entscheidungen begründen. Durch die Verwendung des PlanPro-Formates kann der generierte Plan zu diesem Zweck einfach in ein kompatibles CAD-Programm importiert und dort manuell überarbeitet werden.

4.2 Implementierung

Das Planungstool ist als Konsolenanwendung namens „EPlan“ realisiert, geschrieben in der Programmiersprache Java. Alle notwendigen Planungsparameter wie ETCS-Level, Eingabe- und

[10] Dieser Fall kann auch bei einer rein manuellen Planung auftreten.

Ausgabedatei usw. werden als Kommandozeilenargumente übergeben. Wie bereits erwähnt, erfordern die ETCS Planungsrichtlinien ein bereits geplantes Stellwerk, und die Informationen darüber müssen in der PlanPro-Eingabedatei vorhanden sein. Hauptsignale oder konventionelle Zugsicherungssysteme sind dagegen nicht erforderlich. Die Ausgabedatei enthält zusätzliche ETCS-Komponenten wie Datenpunkte. Alle anderen Informationen aus der Eingabedatei bleiben unverändert. Der Kontrollfluss sieht damit wie folgt aus:

1. Die Kommandozeilenargumente werden ausgewertet.
2. Die PlanPro XML Datei wird geladen, der Inhalt geparst und als ein hierarchischer DOM-Baum gespeichert (W3C, 2015).
3. Für jeden der bereits beschriebenen Planungsschritte wird ein Konstruktionsalgorithmus zur Platzierung der ETCS-Feldelemente implementiert. Jeder Algorithmus hat Zugriff auf den gesamten DOM-Baum, um die Platzierungsparameter zu berechnen (z.B. den Abstand zu bestimmten Objekten). Die neuen Datenpunkte werden als DOM-Nodes erstellt und in den DOM-Baum eingefügt.
4. Der resultierende DOM-Baum wird als PlanPro XML Datei gespeichert.

Der Platzierungsalgorithmus hat Zugriff auf verschiedene Hilfsfunktionen. Diese operieren auf dem DOM-Baum und stellen Funktionen zur Verfügung wie Graph Traversierung, Abstandberechnung zwischen zwei Objekten oder Berechnung von Platzierungspositionen mit bestimmter Distanz zu einem Referenzobjekt unter Berücksichtigung aller möglichen Verzweigungen. Diese Funktionen werden innerhalb des Platzierungsalgorithmus aufgerufen, um Bedingungen auszuwerten und die Position der Datenpunkte zu berechnen.

4.3 Performance

EPlan besteht aus über 2000 Zeilen Java-Code. Die zur Evaluation verwendeten Eingabedateien haben eine Größe von über 25 MB, enthalten bis zu 3900 Objekte mit mehr als 35000 Attributen, darunter Base64-codierte Binärdaten. Dies entspricht einem Bahnhof mittlerer Größe mit 8 Gleisen, 138 Signalen, 32 Weichen und 75 Geleisfreimeldeeinrichtungen. Die Erstellung eines Plans dauert in diesem Fall ca. 5 Sekunden. Da alle bekannten realen ETCS-Planungen eine vergleichbare Größe haben, sind keine Performanceprobleme bei einem produktiven Einsatz unter realen Bedingungen zu erwarten.

5 Prüfalgorithmus

5.1 Grundlage zur Erstellung des Prüfalgorithmus

Die ETCS Regelwerke sollen einem Planprüfer die systematische und strukturierte Planprüfung ermöglichen. Dies ist durch die Funktion der Regelwerke:

- Randbedingungen für die Planung;
- Anordnungsbeziehungen zwischen ETCS – Elementen und
- Einbautoleranzen für die ETCS – Elemente gewährleistet.

Der Prüfalgorithmus basiert auf den Inhalten der Module 819.1343 und 819.1344 (Deutsche Bahn AG). Die Logik, die im Prüfalgorithmus etabliert ist, ist auch für die anderen ETCS Module verwendbar. Zusätzliche Arbeit bei der Formalisierung der anderen ETCS Module wird dann durch die im Prüfalgorithmus geänderten Randbedingungen, Anordnungsbeziehungen oder Einbautoleranzen entstehen. In Abbildung 2 ist der Aufbau eines ETCS – Regelwerkes dargestellt.

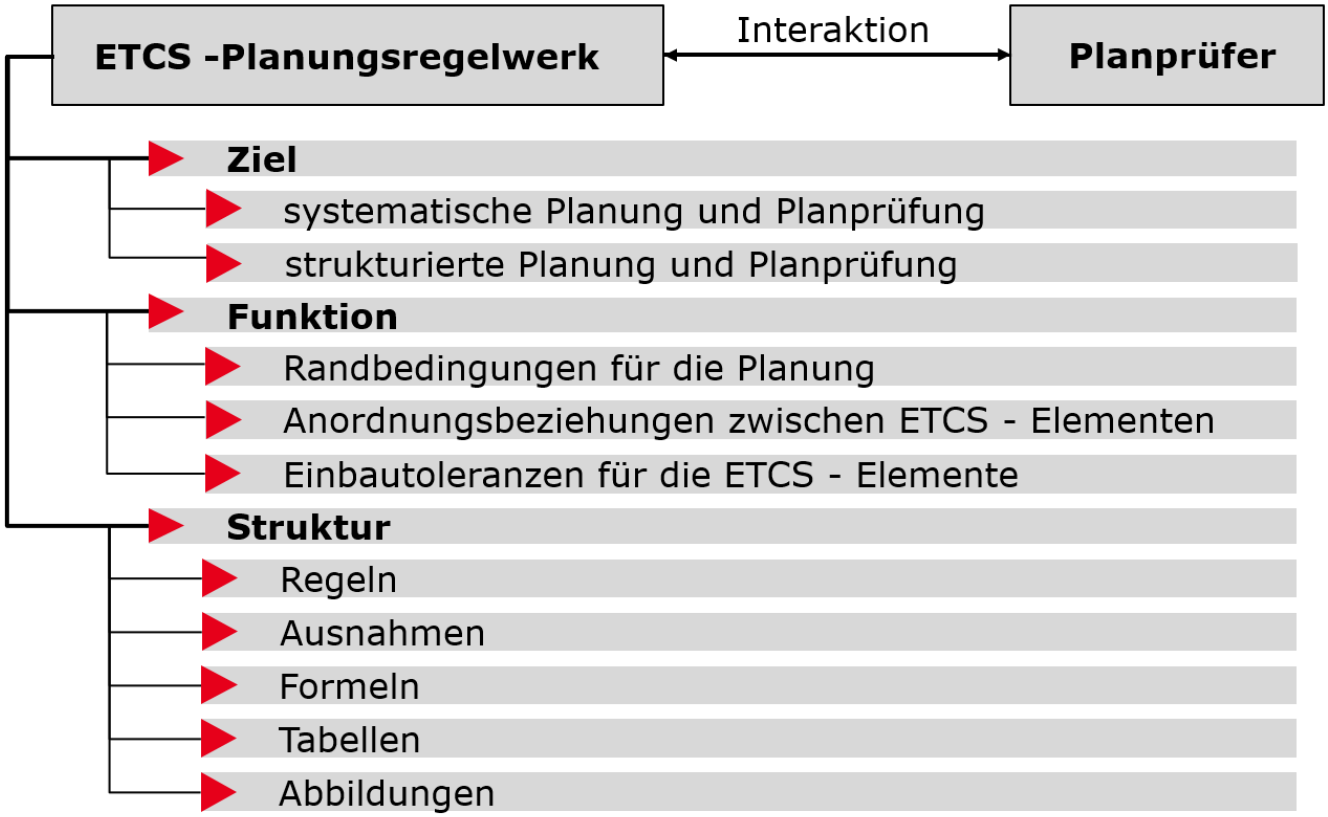

Abb. 2 Aufbau eines ETCS-Planungsregelwerkes

Die Modulinhalte bestehen aus Texten, Tabellen und Abbildungen. Die Regelwerktexte enthalten die Regeln, die Bedingungen, Spielräume für die Planer und die Formeln. Diese werden im ersten Schritt „Kategorisierung“ identifiziert und markiert, dann im zweiten Schritt „halbformale Darstellung“ im Pseudocode dargestellt und anschließend im dritten Schritt „formale Darstellung“ durch Ablaufdiagramme vervollständigt.

Die genannten Schritte sind an einem Beispiel für DP vom Typ 22 gezeigt. In diesem Beispiel ist nur ein Teil des Textes aus dem Modul 819.1344 [2] betrachtet. Im natürlichsprachlich geschriebenen Text sind eine Regel, ein Spielraum für den Planer und eine Bedingung markiert. Der nicht markierte Satz stellt eine Definition dar, die für den Prüfalgorithmus keine Relevanz hat. (Abb. 3) Die auf diese Weise markierten Regeln, Bedingungen, Spielräume für die Planer werden im zweiten Schritt „halbformale Darstellung“ (Abb. 4) durch Wenn - Dann Beziehungen im Pseudocode dargestellt. Die halbformale Darstellung ist ein Schritt-für-Schritt Entwurf der markierten Textabsätze. Im dritten Schritt „formale Darstellung“ (Abb. 5) werden die Ablaufdiagramme erstellt, die eine Abfolge von Entscheidungen, Anweisungen und Ausgaben bei der Planprüfung innerhalb eines ETCS – Elementes vorschreiben.

DP an Blockkennzeichen, Typ 22

(27) Der Datenpunkt vom Typ 22 dient neben Ortungszwecken vor allem zur Haltfallbewertung an Blockkennzeichen. Dieser ist für alle Blockkennzeichen innerhalb des L2-Bereiches am Ort des Blockkennzeichens vorzusehen. Eine Verlegung bis 6 m davor ist möglich.

In begründeten Ausnahmefällen, wenn eine Verlegung vor dem Blockkennzeichen nicht möglich ist, darf der Datenpunkt bis 6 m hinter dem Blockkennzeichen positioniert werden.

Regeln Bedingungen Spielraum für den Planer

Abb. 3 Schritt 1 - Kategorisierung

```
Regel: "Der Datenpunkt vom Typ 22 ist für alle Blockkennzeichen innerhalb des L2-Bereiches am Ort des Blockkennzeichens
vorzusehen"
falls (L2Bereich == true && blockkennzeichen == true):
dann (DpTyp22): IST_ERFORDERLICH

        Spielraum für den Planer: "Eine Verlegung bis 6 m davor ist möglich."
        falls (Abstand (blockkennzeichen, kmDpTyp22) >= 0 && Abstand (blockkennzeichen, kmDpTyp22) <= 6):
        dann (DpTyp22): IST_ERLAUBT

                Ausnahme: "In begründeten Ausnahmefällen, wenn eine Verlegung vor dem Blockkennzeichen nicht möglich ist,
                darf der Datenpunkt bis 6 m hinter dem Blockkennzeichen positioniert werden."
                falls (Abstand (blockkennzeichen, kmDpTyp22) < 0 && Abstand (blockkennzeichen, kmDpTyp22) >= -6):
                dann (DpTyp22): BEDINGT_ERLAUBT
```

Abb. 4 Schritt 2 - halbformale Darstellung

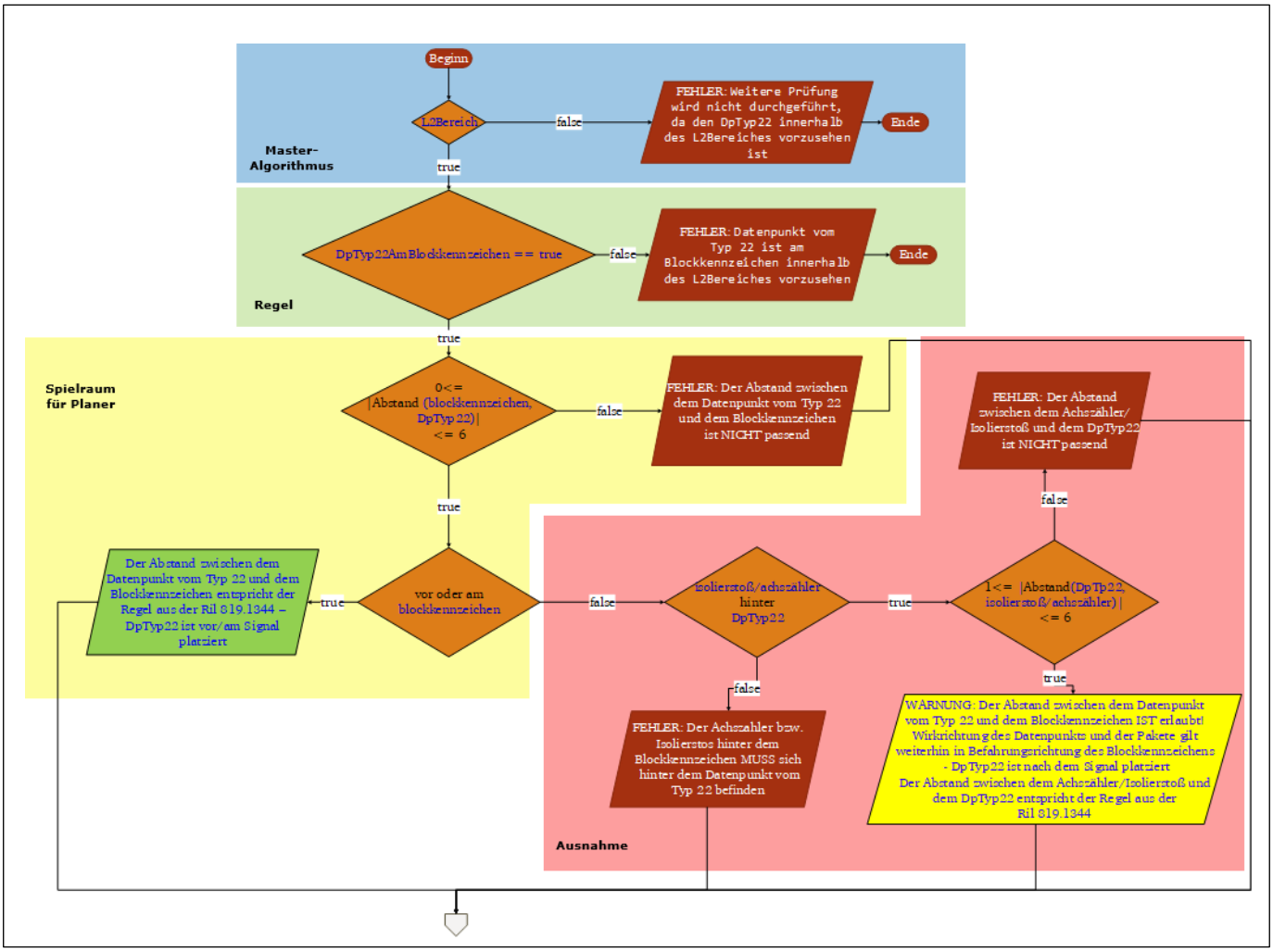

Abb. 5 Schritt 3 - formale Darstellung

5.2 Ablaufdiagramm am Beispiel des Datenpunktes vom Typ 22

Am Anfang werden die zwei Variablen (*DpTyp22* und *L2Bereich*) eingelesen, die zur Prüfung der Grundvoraussetzungen notwendig sind. Falls das zu prüfende Element oder der zu prüfende Bereich nicht passend ist, muss die weitere Prüfung abgebrochen werden. In diesem Fall wird ein „Fehler“ aus dem roten Parallelogramm gemeldet (Abb. 5). Anderenfalls müssen die neuen Variablen (*inKilometrierungRichtung, kmBlockkennzeichen, kmDpTyp22, kmAchszähler* und *kmIsolierstoß*), die für die weitere Prüfung erforderlich sind, eingelesen werden.

Bei der weiteren Prüfung sind acht Szenarien zu unterscheiden. Hier ist zu erwähnen, dass in Abbildung 5 nur die ersten zwei Szenarien zu sehen sind. Die ersten vier Szenarien beziehen sich auf die Prüfung der vor dem Blockkennzeichen platzierten Balisen und die restlichen Szenarien sind für die Fälle, bei denen die Balisen hinter dem Blockkennzeichen platziert sind, reserviert.

Szenario 1: Betrachtet in Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 entweder am Standort des Blockkennzeichens oder in einem Abstand von bis zu 6 Metern davor. Wenn alle Verzweigungen innerhalb des ersten Szenarios „*true*“ sind, dann erfolgt die grüne Ausgabe. Szenario zwei wird aufgerufen, wenn einer der Verzweigungen innerhalb des ersten Szenarios „false“ ist.

Szenario 2: Betrachtet in Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 in einem Abstand von mehr als 6 Metern vor dem Blockkennzeichen. Wenn alle Verzweigungen innerhalb des zweiten Szenarios „*true*“ sind, dann erfolgt die rote Ausgabe und wird ein „Fehler“ gemeldet. Szenario drei wird aufgerufen, wenn einer der Verzweigungen innerhalb des zweiten Szenarios „false“ ist.

Szenario 3: Betrachtet gegen die Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 entweder am Standort des Blockkennzeichens oder in einem Abstand von bis zu 6 Metern davor.

Szenario 4: Betrachtet gegen die Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 in einem Abstand von mehr als 6 Metern vor dem Blockkennzeichen.

Szenario 5: Betrachtet in Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 hinter dem Blockkennzeichen in einem Abstand von bis zu 6 Metern und der Achszähler oder der Isolierstoß muss sich 1-6 Meter hinter dem Datenpunkt vom Typ 22 befinden.

Szenario 6: Betrachtet in Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 hinter dem Blockkennzeichen in einem Abstand von mehr als 6 Metern.

Szenario 7: Betrachtet gegen die Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 hinter dem Blockkennzeichen in einem Abstand von bis zu 6 Metern und der Achszähler oder der Isolierstoß muss sich 1-6 Meter hinter dem Datenpunkt vom Typ 22 befinden.

Szenario 8: Betrachtet gegen die Kilometrierungsrichtung befindet sich der Datenpunkt vom Typ 22 hinter dem Blockkennzeichen in einem Abstand von mehr als 6 Metern.

Wenn eines der acht Szenarien eintritt, werden die anderen sieben Szenarien nicht in Erwägung gezogen.

Im Anschluss an die abgeschlossenen Prüfungen werden die neuen Variablen (*balisenanzahl* und *ungesteuerteBalise*) eingelesen. Ein Datenpunkt vom Typ 22 muss aus einer ungesteuerten Balise bestehen. Diesbezüglich wird bei dieser Iteration die Balisenanzahl geprüft. Ob es sich um eine gesteuerte oder ungesteuerte Balise handelt, wird am Ende des Ablaufdiagrammes geprüft.

5.3 Systematische Planprüfung

Die systematische Planprüfung ist in zwei Stufen aufgebaut. In der ersten Stufe wird ein sogenannter Master-Algorithmus ausgeführt. Die zweite Stufe enthält die sogenannten Slave-Algorithmen. Der Master-Algorithmus entscheidet, welche nachfolgenden Slave-Algorithmen aufgerufen werden.

Jeder der in Abb. 6 dargestellten untergeordneten weißen Kästen stellt ein eigenes Slave-Algorithmus dar. Der Master-Algorithmus muss beim ersten Start automatisiert entscheiden, wo die Planprüfung beginnt.

Der Master-Algorithmus der allgemeingültigen Regeln prüft:

- Platzierung der Balisen im Weichenbereich;
- die Abstände zwischen Balisen innerhalb eines Datenpunktes;
- Streckenhöchstgeschwindigkeit;
- minimaler Abstand zwischen der letzten Balise eines Datenpunktes und der ersten des nachfolgenden Datenpunkts (außer bei den Datenpunkten mit einer Balise);
- minimaler Abstand zwischen Balisen und Balisenantennen;
- minimale Abstände im Bereich von Bahnübergängen;
- minimale Abstände zu Metallobjekten im Gleis;

- Bogenradius bei der Balisenmontage auf Stahlschwellen;
- minimaler Bogen- und Ausrundungsradius;
- die erlaubten Planungstoleranzen.

Die Abstände zwischen Balisen innerhalb eines Datenpunktes und der minimale Abstand zwischen der letzten Balise eines Datenpunktes und der ersten des nachfolgenden Datenpunkts (außer bei den Datenpunkten mit einer Balise) sind zwei Slave-Algorithmen, die meistens zur Anwendung kommen.

Diese sind immer aufzurufen, wenn ein Datenpunkt aus zwei Balisen zusammengesetzt ist. Die restlichen Slave-Algorithmen der allgemeingültigen Regeln werden nach Bedarf (Weichenbereich, Bahnübergang, Bogen, usw.) aufgerufen.

Nach der Prüfung der allgemeingültigen Regeln ist die Prüfung der ein- und ausstiegsrelevanten Datenpunkte durchzuführen. Danach wird die Platzierung der Datenpunkte innerhalb des L2-Bereiches, an Bahnübergängen und an RBC-Wechsel geprüft.

Die im zweiten Schritt halbformal dargestellten Anordnungsbeziehungen müssen in diesem Schritt als Ablaufdiagramme dargestellt werden. Jeder der in Abbildung 6 dargestellten untergeordneten weißen Kästen stellt ein eigenes Ablaufdiagramm dar. Der Prüfalgorithmus muss beim ersten Start automatisiert entscheiden, wo die Planprüfung beginnt. Für die automatisierte Entscheidung ist ein übergeordnetes Ablaufdiagramm notwendig (rote Kästen).

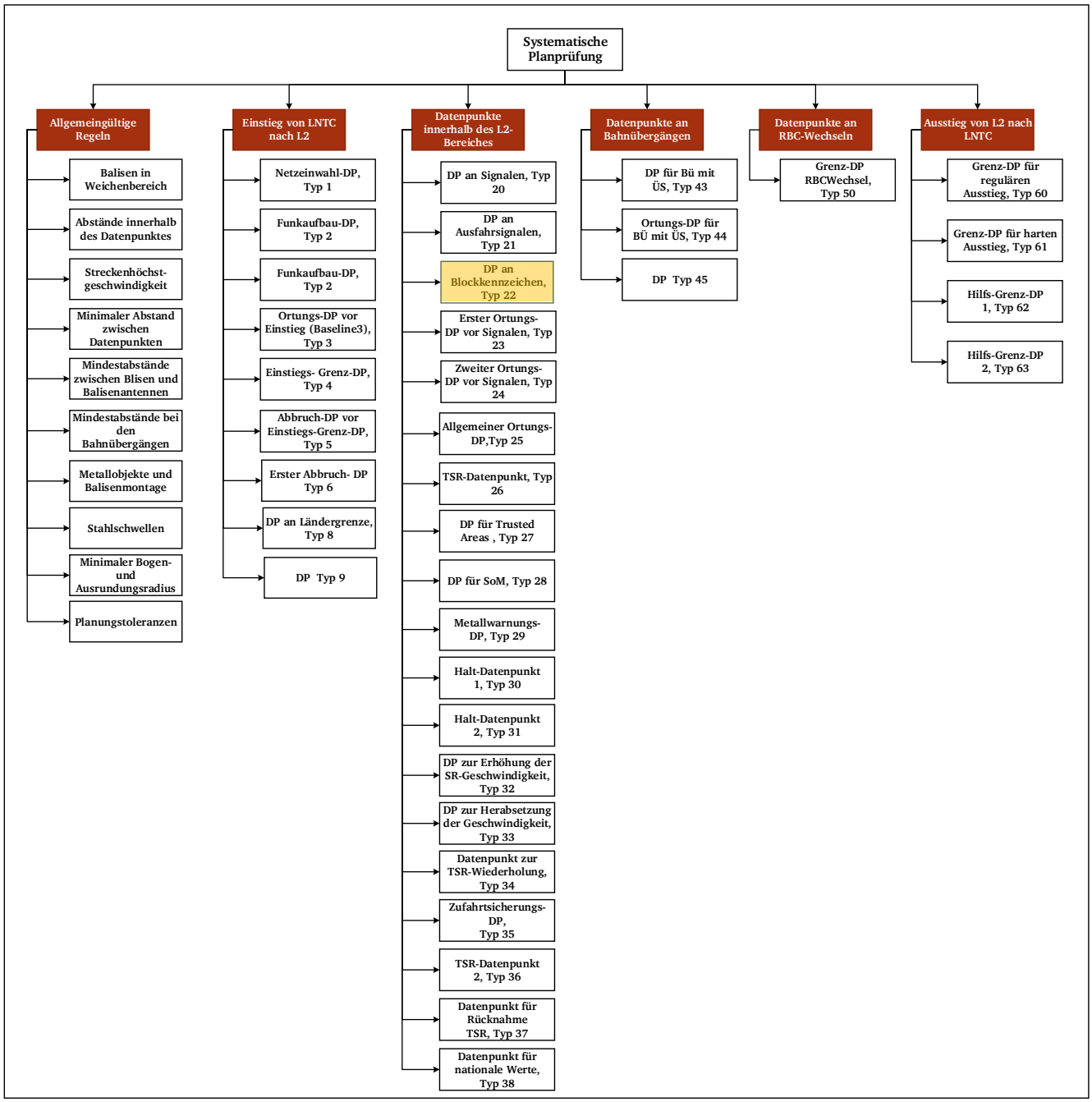

Abb. 6 Schritt 4 - systematische Planprüfung

6 Verifizierung und Validierung

6.1 Allgemeine Überlegungen

Eine perfekte Teststrategie erfordert eine *Validierung*, dass alle Planungsregeln (Deutsche Bahn AG) korrekt interpretiert und formalisiert worden sind, sowie eine formale *Verifizierung*, dass die Regeln korrekt implementiert worden sind und das Planungstool nur ETCS-konforme Pläne produziert bzw. das Planprüfungstool nur solche Pläne als korrekt bewertet. Da davon auszugehen ist, dass auch in absehbarer Zukunft alle Pläne vor der Baufreigabe eine finale, manuelle Planprüfung absolvieren müssen und es zudem unklar ist, ob die Zulassung vollständig formal verifizierter Tools (bei denen auf eine manuelle Prüfung verzichtet werden kann) möglich ist oder von den Zulassungsbehörden überhaupt gewollt ist, erscheinen solche Ziele vor allem in einem universitären, nicht-produktiven Umfeld ambitioniert.

Daher wurde die Priorität dahin gehend gesetzt, die entwickelten Demonstratoren so schnell wie möglich in den produktiven Einsatz zu bringen, um das Konzept in einer realen Umgebung zu evaluieren. Es können in diesem Fall schwächere Korrektheitsanforderungen hingenommen werden, weil die existierenden manuellen Planungsprozesse hier nur begleitend unterstützt werden und Fehler in den automatisch erstellten und automatisch geprüften Plänen im abschließenden manuellen Planprüfungsprozess entdeckt werden können – genau wie bisher bei ausschließlich manuell erstellten Plänen – und durch das offene Datenformat jederzeit korrigierbar sind. Um einen akzeptablen Vertrauensgrad in die entwickelten Tools zu erreichen, muss die Notwendigkeit manueller Korrekturen auf ein Minimum reduziert werden. Hier ist es wichtig sich bewusst zu machen, dass bislang nur die Planprüfer das Wissen und die Autorität haben zur Entscheidung, ob ein Plan korrekt ist oder nicht. Allerdings können sie nur *fertige* Planergebnisse bewerten, nicht jedoch die Einzelschritte eines Algorithmus. Es ist daher an dieser Stelle nicht sinnvoll, separate Korrektheitseigenschaften für die verschiedenen Schritte der Planungs- und Prüfungsalgorithmen zu spezifizieren. Stattdessen wird einen zweigleisigen Ansatz mit Fokus auf fertigen Plänen verfolgt: Um die Prüfer bei der *Validierung* der Pläne zu unterstützen, wurde ein zusätzliches Visualisierungstool entwickelt (siehe Kap. 6.2). Um Planungs- und Planprüfungstool zu *verifizieren*, wurde ein *domänenspezifisches Testabdeckungskriterium* definiert: Eine Zusammenstellung von Testplänen, die alle möglichen Fallunterscheidungen des Regelwerks abdecken, wird zur Planprüfung eingereicht. Dies erlaubt es, aus der Korrektheit der überprüften Pläne in Bezug auf die Testabdeckung die Korrektheit der Algorithmen zu schließen (Kap. 6.3).

6.2 Visualisierungs- und Analysetool

XML-Rohdaten sind wenig intuitiv und müssen in einem für Menschen verständlichen Format dargestellt werden, um für eine Begutachtung zugänglich zu sein. Leider sind die aktuell verfügbaren Tools, die PlanPro-Dateien verarbeiten können, für die Planung konventioneller Signalsysteme ausgelegt und unterstützen keine PlanPro-Daten mit ETCS Level 2 Objekten. Aus diesem Grund wurde ein eigenes PlanPro-Visualisierungstool namens „PlanPro Viewer“ entwickelt. Es kann einen Gleisplan aus einer PlanPro XML-Datei rendern. Das Gleislayout enthält alle Objekte, die eine geografische Positionsangabe besitzen (wie Signale, Datenpunkte, Gleisfreimelder etc.). Es ähnelt den für die Planprüfung benötigten Übersichtsplänen und stellt genügend Informationen zur Verfügung, einschließlich der relativen Positionen der Elemente zueinander. Die untenstehenden Abbildungen zeigen zwei Screenshots aus PlanPro Viewer und demonstrieren die Änderungen, die das Planungstool an einem gegebenen Gleislayout durchführt. Abb. 7 ist der Ausgangsplan mit konventioneller Signalisierung (grüne Punkte), der als Input für das Planungstool dient. Abb. 8 zeigt den gleichen Plan

nach der Ausführung des Planungsalgorithmus, bei dem die neu hinzugefügten ETCS-Datenpunkte als gelbe Kreise dargestellt sind.

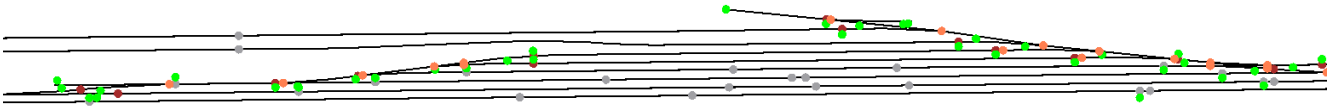

Abb. 7 Eingabegleisplan

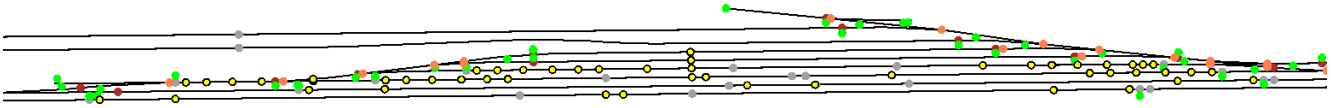

Abb. 8 Gleisplan nach Platzierung von ETCS-Elementen

Alle im Plan vorhandenen Objekte, einschließlich logischer Objekte ohne Positionsinformation (Telegramme, Signalbegriffe etc.), können separat aufgelistet und abgefragt werden. PlanPro Viewer ermöglicht eine interaktive Abfrage von Objekteigenschaften wie auch eine Objektsuche auf Basis der Objekt-ID. Ein zentrales Feature ist die Distanzberechnung zwischen zwei beliebigen positionierbaren Objekten. Diese Abstandsmessung ist eine häufig benutzte und mühsame Aufgabe in einem Prüfungsverfahren. Korrekt berechnete Abstände sind sicherheitsrelevant, daher ist die Implementierung dieser Funktion einer der primären Kandidaten für eine formale Verifizierung (siehe Kap. 7.2).

6.3 Regelüberdeckung

Die ETCS-Planungsrichtlinien (Deutsche Bahn AG) sind nach Platzierungsregeln für jeden Datenpunkttyp geordnet. Jede Regel enthält verschiedene Bedingungen, die Verzweigungen und Unterregeln innerhalb der Hauptregel bilden. Allerdings müssen nicht alle Regeln betrachtet werden, um einen Datenpunkt zu platzieren oder die korrekte Position zu überprüfen. Die verwendeten Regeln hängen vielmehr von der konkreten Planungssituation ab (siehe Kap. 5).

Generell sind generische, codebasierte Testabdeckungskriterien (Ammann & Offutt, 2008) ungeeignet für Programme, die mit komplexen DOM-Objekten arbeiten. Wie bei Compilern und anderen Codetransformatoren ist es adäquater, ein *domänenspezifisches* Testkriterium zu definieren und eine entsprechende Testsuite aufzubauen.

Um das Vertrauen in die Korrektheit des automatischen Planungsprozesses zu erhöhen, müssen alle möglichen Unterregeln mindestens einmal in einem Szenario der Testsuite ausgewählt werden. Dies wird hier *Regelüberdeckung* genannt. Für jedes Testszenario wird ein *erwarteter* Plan konstruiert, der von einem Planprüfer abgenommen werden muss. Daraus kann ein *Testorakel* konstruiert werden, indem der erwartete Plan für jedes Szenario mit dem tatsächlich generierten Plan verglichen wird. Dies kann beispielsweise als Ausgangspunkt für einen automatisierten Regressionstest dienen.

Diese Teststrategie kann alternativ als eine Instanz von modellbasiertem Testen (Utting & Legeard, 2007) verstanden werden, wobei die formalisierten Regeln (Deutsche Bahn AG) die Rolle des Modells einnehmen.

7 Verwandte und zukünftige Arbeiten

7.1 Verwandte Arbeiten

Es gibt eine große Anzahl an Arbeiten zu formaler Modellierung und Verifizierung von Stellwerkstechniken, wobei hier der Schwerpunkt auf den *dynamischen* Aspekten des Bahnbetriebs liegt. Das ist nicht der Fokus unseres Projektes, das sich mit der automatisierten Planung *statischer* Infrastrukturen beschäftigt.

Es sind nicht viele Arbeiten zum Thema automatisierte ETCS Planung bekannt. Der vorliegenden Arbeit am nächsten kommt das norwegische NFR Projekt *RailCons*[11], das von der Universität von Oslo und dem Unternehmen RailCOMPLETE AS verfolgt wird. Letzteres vertreibt ein CAD-basiertes Planungstool ähnlich ProSig, welches AutoCAD mit bahnspezifischen semantischen Daten erweitert. In dem Projekt wurde das Programm mit zusätzlichen semantischen Informationen erweitert, die auf der standardisierten Railway Markup Language railML[12] basieren und direkt in den AutoCAD-Dateien abgespeichert werden. Dieses railML-Modell kann aus den CAD-Dateien extrahiert werden und gegen formalisierte Planungsregeln verifiziert werden (Luteberget & Johansen, 2018). Im Gegensatz zu dem hier beschriebenen Ansatz werden diese Regeln in einer logischen Programmiersprache codiert. Die Autoren haben das Konzept mit Streckendaten aus einem realen Bauvorhaben evaluiert und können aufzeigen, dass nicht-triviale Eigenschaften innerhalb von Sekunden verifizierbar sind. Das RailCons-Projekt behandelt nur die regelbasierte *Prüfung* manuell erstellte Pläne. In FormETCS dagegen wird zusätzlich, zumindest teilweise, auch die *Erstellung* der Pläne automatisiert. Das RailCons-Modell scheint etwas abstrakter als das hier verwendete PlanPro-Modell und die Regeln für die Verifikation scheinen nicht vollständig zu sein. Insbesondere werden ETCS-Regularien in (Luteberget & Johansen, 2018) unter zukünftige Arbeiten erwähnt.

7.2 Zukünftige Arbeiten

Als nächster Schritt soll eine real existierende ETCS-Strecke mit dem Planungswerkzeug nachgeplant werden und die Ergebnisse von einem DB Expertenteam verglichen werden. Dies ist ein Schritt in Richtung Verifikation der Planungsalgorithmen. Es soll außerdem zeigen, dass das Tool für ein reales Betriebsszenario benutzbar ist. Für einen produktiven Einsatz ist ebenfalls eine weitergehende Unterstützung von ETCS Level 1 erforderlich.

Ein verwandtes Projekt zu FormETCS ist FormbaR[13], in dem Regelwerke für den *Betrieb* (konkret die Ril 408: Fahrdienstvorschrift) in einem ausführbaren Modell formalisiert wurden (Kamburjan, Hähnle, & Schön, 2018), das sowohl Simulation als auch statische Kostenabschätzung erlaubt. Es erscheint erstrebenswert, beide Projekte zu kombinieren: Zuerst kann mit den FormETCS-Tools eine Streckenplanung in verschiedenen Varianten (z.B. Level 1 und 2) erstellt und geprüft werden. Diese können dann mit dem FormbaR-Tool auf ihre relative Leistungsfähigkeit unter verschiedenen Lastsituationen evaluiert werden. Dies würde es erlauben, Planungsentscheidungen hinsichtlich der Kapazität auf Grundlage von feingranularen, realistischen, dynamischen Modellen zu treffen.

Während eine formale Verifizierung oder Spezifizierung der FormETCS-Tools als Ganzes, wie bereits erwähnt, im aktuellen Status eher wenig Sinn ergibt, ist dagegen die Verifikation der wichtigsten

[11] https://www.mn.uio.no/ifi/english/research/projects/railcons

[12] https://www.railml.org

[13] https//formbar.raillab.de/de/about

sicherheitsrelevanten und fehlerträchtigen Teile des Codes (etwa Distanzberechnungen) wünschenswert. Eine formale Verifikation solcher einzelner Java-Codebestandteile ist ohne größere Schwierigkeiten durchführbar (Ahrendt, et al., 2016).

8 Fazit

In diesem Artikel ist ein Konzept für eine automatisierte ETCS-Planung mit dem Schwerpunkt auf ETCS Level 2 beschrieben, unterteilt in die Teildisziplinen Planung und Planprüfung. Für beide Disziplinen wurden Algorithmen entwickelt, die in bestehende Workflows und Tools integriert werden können.

Der Planungsalgorithmus basiert auf der Vorgehensweise, der Erfahrung und dem Wissen eines Planers und nicht auf dem Regelwerk direkt. Die für die ETCS–Planung notwendigen Planungsregeln sind nur implizit im Planungsalgorithmus vorhanden. Die Algorithmen sind in einem Demonstrator implementiert, der als Eingabe ein digitales Gleislayout im PlanPro-Format verwendet und dieses um ETCS-Elemente anreichert.

Im Prüfalgorithmus sind die einzelnen Regeln aus der Richtlinien 819.1344 strukturiert dargestellt. Der Regelwerkstext wird zunächst kategorisiert, halbformal dargestellt und dann in ein Ablaufdiagramm überführt. Alle diese Schritte bilden die Voraussetzungen für die Realisierung der systematischen Planprüfung.

Abschließend ist eine Validierungs- und Verifikationsstrategie beschrieben, die einerseits aus der Visualisierung der generierten Planungsdaten mit einem eigens entwickelten Anzeigeprogramm besteht und andererseits ein domänenspezifisches Testkriterium zur Verifikation der Algorithmen definiert. Eine formale Validierung und Verifikation sind als zukünftige Arbeiten denkbar.

9 Literaturverzeichnis

[1] BMVI, „Nationaler Umsetzungsplan ETCS,“ 2017.

[2] Deutsche Bahn AG, „Richtlinie 819: LST-Anlagen planen,“ Frankfurt.

[3] U. Maschek, C. Klaus, C. Gerke, V. Uminski und K.-J. Girke, „PlanPro: Durchgängige elektronische Datenhaltung im ESTW-Planungsprozess,“ Signal+Draht, Bd. 104, Nr. 9, pp. 22-26, 2012.

[4] DB Netz AG, „PlanPro-Glossar,“ [Online]. Available: http://confluence.plan-pro.org/display/G180/Index.

[5] W3C, „Document Object Model DOM 4,“ 2015.

[6] P. Ammann und J. Offutt, Introduction to Software Testing, Cambridge University Press, 2008.

[7] M. Utting und B. Legeard, Practical Model-Based Testing - A Tools Approach, Morgan Kaufmann, 2007.

[8] B. Luteberget und C. Johansen, „Efficient verification of railway infrastructure designs against standard regulations,“ Formal Methods in System Design, Bd. 52, Nr. 1, pp. 1-32, 2018.

[9] E. Kamburjan, R. Hähnle und S. Schön, „Formal modelling and analysis of railway operations with Active Objects,“ Science of Computer Programming, pp. 167-193, Nov 2018.

[10] W. Ahrendt, B. Beckert, R. Bubel, R. Hähnle, P. Schmitt und M. Ulbrich, Deductive Software Verification-The KeY Book: From Theory to Practice, Springer, 2016.

[11] Deutsche Bahn AG, „Richtlinie 301: Signalbuch," Frankfurt.

[12] S. Dillmann und R. Hähnle, „Automated Planning of ETCS Tracks," in RSSRail, Lille, 2019.