Bachelorarbeit

Fahrwegsuche nach dem Spurplanprinzip - modellbasierte Anforderungsspezifikation und Validierung

eingereicht am Institut für Informatik der Freien Universität Berlin
Arbeitsgruppe Software Engineering
Fachgruppe modellbasierte Entwicklung und
Qualitätssicherung software-basierter Systeme

in Zusammenarbeit mit der
Berner & Mattner Systemtechnik GmbH
Geschäftsbereich Transportation - Abteilung Signalling

von

Dennis Kunz
geb. am 09.02.1988 in Berlin.
Matrikelnummer: 4356356
dpkunz@zedat.fu-berlin.de

Erstgutachter: Prof. Dr.-Ing. Ina Schieferdecker
Zweigutachter: Prof. Dr. Elfriede Fehr

Betreuer Freie Universität Berlin: Prof. Dr.-Ing. Ina Schieferdecker
Betreuer Berner & Matter Systemtechnik GmbH: Dr. Oliver Lemke

Berlin, 24. Oktober 2014

Zusammenfassung

Modellbasierte Anforderungsspezifikationen stellen neben sprachlich formulierten Anforderungen eine Möglichkeit dar, zu beschreiben, was ein zu entwerfendes System leisten soll. In dieser Arbeit wird eine Anforderungsspezifikation mit Hilfe eines SysML-Modells erstellt, das in der Domäne der Leit- und Sicherungstechnik der Eisenbahn angesiedelt ist. Es wird das technische Verfahren der Fahrwegsuche in Relaisstellwerken nach dem Spurplanprinzip allgemein und unabhängig von einer Implementierung spezifiziert. Dieses ermittelt dynamisch den Fahrweg für eine Zug- oder Rangierfahrt von einem definierten Start- zu einem Zielpunkt über verschiedene Gleis- und Weichenanordnungen. Dabei wird das Modell aus gegebenen Anforderungen in Textform entwickelt, wobei der Fokus auf der Struktur- und Verhaltenssicht liegt. Das Verhalten wird dabei durch Zustandsautomaten beschrieben. Eine dazu erstellte Modellsimulation ermöglicht das Testen der modellbasierten Anforderungsspezifikation der Fahrwegsuche mit einer Benutzeroberfläche. Zusätzlich werden aus den sprachlichen Anforderungen Testfälle abgeleitet, wodurch das entstandene Modell gegen die ursprünglichen Anforderungen validiert wird. Mit den erfolgreichen Tests jedes Modellelements liegt nun eine allgemeine und modellbasierte Spezifikation der Fahrwegsuche in SysML vor.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, 24. Oktober 2014 _____

Inhalt des beiliegenden Datenträgers

Dieser schriftlichen Ausarbeitung liegt ein Datenträger (Compact Disc) bei, auf dem sich die in dieser Arbeit erstellen Modellsimulationen befinden. Sie bestehen jeweils aus dem Visual Basic-Quellcode mit den zugehörigen Projektdateien, sowie dem kompilierten Code in Form einer ausführbaren Anwendung. Die einzelnen Simulatoren sind nachfolgend aufgelistet.

- Simulator Fahrstraßenelement (Verzeichnis „TestumgebungFE“)
- Simulator Weichenelement (Verzeichnis „TestumgebungWE“)
- Simulator Umfahrstraßenelement (Verzeichnis „TestumgebungUE“)
- Simulator Testbahnhof2 (Elementverbindungsplan) (Verzeichnis „Testbahnhof2“)

Das ausführbare Programm jedes Simulators ist dabei im Unterverzeichnis „\bin\release“ abgelegt.

Alle Quellcodes sind für die Entwicklungsumgebung *Microsoft Visual Basic 2008 Express* generiert, so dass empfohlen wird, die Projektdateien mit dieser zu öffnen.

Die Diagramme dieser Arbeit sind ebenfalls in digitaler Form im Verzeichnis „SysML-Diagramme“ hinterlegt.

Vorwort

Diese Arbeit ist während meiner Tätigkeit als Werkstudent bei der Berner & Mattner Systemtechnik GmbH in der Niederlassung Berlin im Bereich Transportation entstanden. Mein Projektleiter Oliver Lemke schlug mir im Herbst 2013 vor, meine Bachelorarbeit über eine SysML-Modellierung im Bereich der Eisenbahnsicherungstechnik zu schreiben. Zu meiner Freude konnte ich somit mein Studium der Informatik und das private Interesse an Stellwerkstechnik der Eisenbahn miteinander verbinden. Das Thema grenzte sich auf die Relaisstellwerke in Spurplantechnik ein, bei dem das Verfahren der *Fahrwegsuche* mit Hilfe der SysML zu spezifizieren war. Schließlich setzte ich dies in die Tat um und das Ergebnis ist die vorliegende Arbeit.

Deshalb möchte ich mich an dieser Stelle bei Oliver Lemke für den Themenvorschlag und die Betreuung insbesondere im eisenbahn-fachlichen Teil bedanken. Außerdem gilt mein Dank meinem Teamleiter Cengiz Genc für die Ermöglichung, die Arbeit bei Berner & Mattner schreiben zu dürfen. Für technische und methodische Fragen standen mir Maria Bertram und Johann-Christoph Jacob zur Seite, die mir u.a. bei der Bedienung der Modellierungssoftware Atego Artisan Studio mit Tipps und Tricks behilflich waren.

Bedanken möchte ich mich auch bei Prof. Dr.-Ing. Ina Schieferdecker für die universitäre Betreuung und die Beantwortung offener Fragen zur Gestaltung der Testfälle, Anforderungen und methodischen Aspekten.

Zu guter Letzt danke ich meinen Eltern Detlef und Susanne Kunz, die mich durch Höhen und Tiefen im Studium begleitet, mir immer Mut zugesprochen und diese Arbeit zur Korrektur gelesen haben.

Inhaltsverzeichnis

1 Einleitung **3**
1.1 Motivation 4
1.2 Aufbau der Arbeit 4

2 Stand der Wissenschaft und Technik **5**
2.1 Bisherige Arbeiten über modellbasierte Anforderungsspezifikationen im Bahnbereich 5
2.2 Praktischer Einsatz modellbasierter Anforderungsspezifikationen in der Leit- und Sicherungstechnik 6

3 Domänenspezifische Grundlagen der Eisenbahn **8**
3.1 Begriffe des Bahnbetriebs 8
3.2 Fahrwege, Fahrstraßen und Stellwerke 13
3.3 Relaisstellwerke nach dem Spurplanprinzip 15
3.3.1 Gruppenverbindungsplan 18
3.3.2 Objektorientierung 18
3.3.3 Fahrwegsuche nach dem Such-Echostrom-Prinzip 19

4 Modellierung der Fahrwegsuche **23**
4.1 Einordnung in das V-Modell nach EN 50126 23
4.2 Rahmenbedingungen 23
4.2.1 Funktionale Anforderungen in sprachlicher Form 23
4.2.2 Themenabgrenzung der Fahrwegsuche 26
4.2.3 SysML als Beschreibungsmittel der Anforderungsspezifikation . . . 26
4.3 SysML-Struktursicht 26
4.3.1 Modell- und Benennungskonventionen des Modellierungsstandards 28
4.3.2 Block Fahrstraßenelement 28
4.3.3 Block Weichenelement 30
4.3.4 Optionaler Block Umfahrstraßenelement 30
4.4 SysML-Verhaltenssicht 33
4.4.1 Zustandsautomat Fahrstraßenelement 35
4.4.2 Zustandsautomat Weichenelement 37
4.4.3 Optionaler Zustandsautomat Umfahrstraßenelement 41
4.5 Konstruktion eines Elementverbindungplans 42
4.5.1 Ablauf einer Fahrwegsuche in SysML 43

5 Test des Modells durch Validierung **46**
5.1 Ableitung von Testfällen aus den sprachlichen Anforderungen 46
5.2 Erstellung von ausführbaren Modellen mit Simulationsszenarien 48
5.3 Validierung Fahrstraßenelement 51
5.4 Validierung Weichenelement 52
5.5 Validierung Umfahrstraßenelement 54
5.6 Auswertung der Einzelvaliderung 56
5.7 Test der Fahrwegsuche an einem exemplarischen Elementverbindungsplan 56
5.8 Zusammenfassung der Validierung 59

6 Zusammenfassung und Fazit **60**

7 Anhang **64**
7.1 Auflistung der Anforderungen 64
7.1.1 Allgemeine funktionale Anforderungen 64
7.1.2 Funktionale Anforderungen Fahrstraßenelement 64
7.1.3 Funktionale Anforderungen Weichenelement 65
7.1.4 Zusätzliche funktionale Anforderungen 66
7.2 Auflistung der optionalen Anforderungen 66
7.2.1 Funktionale Anforderungen Umfahrstraßenelement 66
7.3 Auflistung der Testfälle 67
7.3.1 Testfälle für das Fahrstraßenelement 67
7.3.2 Testfälle für das Weichenelement 67
7.3.3 Testfälle für das Umfahrstraßenelement 68
7.3.4 Testfälle Beispiel-Lageplan 68
7.4 Ausgabeprotokolle von Artisan Studio der ausgelösten Transitionen in den Tests 69
7.4.1 Ausgabeprotokolle bei Simulationsstart 72
7.5 Abbildung des internen Blockdiagramms des Simulationszenarios Testbahnhof2 72

1 Einleitung

Im Bereich der Leit- und Sicherungstechnik der Eisenbahn wurden für die Spezifikation von Anforderungen an ein zu entwickelndes System bisher natürlich-sprachliche Formulierungen genutzt. Obwohl mittlerweile formale oder grafische Beschreibungsmittel als mögliche Alternative zur Verfügung stehen, wurden diese bis vor einiger Zeit in diesem Gebiet nicht verwendet.

Für die Entwicklung von Software der Eisenbahnsicherungstechnik ist der Einsatz formaler Methoden und der Unified Modeling Language (UML) bereits durch die CENELEC-Norm EN 50128 empfohlen [6]. Zu diesem Zeitpunkt des Entwicklungsprozesses eines Systems handelt es sich aber um jene Schritte, die nach der Anforderungsspezifikation des Gesamtsystems stattfinden. Mögliche Fehler oder falsche Interpretationen, die durch nicht präzise genug formulierte, sprachliche Anforderungen entstehen können, übertragen sich im schlechtesten Fall auf die Phase der Implementierung. Deshalb liegt es nahe, auch für Systemanforderungen Beschreibungsmittel wie die UML zu verwenden. Zu entwerfende Systeme bzw. Teilsysteme müssen jedoch nicht zwangsläufig vollständig softwarebasiert realisiert werden, wodurch die UML nicht umfassend geeignet ist. Formale Methoden basierend auf mathematischer Logik sind zwar präzise, benötigen jedoch entsprechendes Verständnis. Aus ingenieurmäßigem Vorgehen heraus bietet sich als geeignetes Beschreibungsmittel für den Zweck modellbasierter Anforderungsspezifikationen die Systems Modeling Language (SysML) an.

Die Überführung sprachlich formulierter Anforderungen in eine modellbasierte Anforderungsspezifikation mit der SysML ist das Ziel dieser Arbeit. An einem bekannten Verfahren aus der Eisenbahnsicherungstechnik wird dies exemplarisch durchgeführt. Dieses Verfahren, die Fahrwegsuche, ist ein technischer Vorgang in einem Relaisstellwerk nach dem Spurplanprinzip, bei dem von einem Start- zu einem Zielpunkt über verschiedene Gleis- und Weichenanordnungen ein Fahrweg ermittelt wird. Aus der dazu gegebenen Anforderungsmenge in Textform wird mittels geeigneter SysML-Sprachelemente und Systemsichten eine modellbasierte Anforderungsspezifikation entwickelt, die die Fahrwegsuche beschreibt.

Der Fokus liegt dabei insbesondere auf einer ausführbaren Verhaltensmodellierung, die durch entsprechende Software-Tools in Quelltext übersetzt und ausgeführt werden kann. Als entscheidender Vorteil ergibt sich außerdem die Möglichkeit, aus den sprachlich formulierten Anforderungen Testfälle abzuleiten und folglich das SysML-Modell gegen die ursprünglichen Anforderungen zu validieren.

1.1 Motivation

Die Motivation zu dieser Arbeit hat zwei wesentliche Aspekte. Zum einen geht es darum, an einer konkreten Anwendung die Möglichkeiten modellbasierter Anforderungsspezifikationen zu zeigen. Dabei ist eine vergleichende Gegenüberstellung der aufgeführten sprachlichen Anforderungen und der modellierten Anforderungen in SysML möglich. Ein weiterer Aspekt betrifft den fachlichen Nutzen für die Leit- und Sicherungstechnik.

Denn obwohl Relaisstellwerke bei Eisenbahnen nicht mehr Stand der Technik sind und zu den Bestandsanlagen zählen, haben die dort elektrotechnisch umgesetzten Aspekte der Eisenbahn-Sicherungstechnik weiterhin eine fachliche Wichtigkeit. Es geht darum, das Kernprinzip der Fahrwegsuche von der damaligen, elektrotechnischen Umsetzung zu lösen und allgemein darzustellen. Folglich ergibt sich eine sinnvolle Abstraktion von der Implementierung, aber auch eine Aktualisierung und Aufbereitung von bereits gewonnenen Wissen. Als Vorteil ergibt sich, dass die Fahrwegsuche als SysML-Modell sowohl akademisch, als auch praktisch allgemein nutzbar gemacht wird.

1.2 Aufbau der Arbeit

Beginnend mit Kapitel 2 folgt zunächst eine Zusammenfassung bisheriger wissenschaftlicher Arbeiten, die das Thema modellbasierter Anforderungsspezifikation im Bahnbereich behandelt haben. In Ergänzung dazu werden konkrete Einsätze in der Praxis erläutert. Kapitel 3 befasst sich mit der Einführung von benötigten Fachbegriffen aus dem Bahnbetrieb und der Leit- und Sicherungstechnik. Letztere sind zum Verständnis der Fahrwegsuche unabdingbar. Anschließend enthält Kapitel 4 die eigentliche Modellierung der Fahrwegsuche und die Überführung der Anforderungen in das SysML-Modell. Kapitel 5 dient der Beschreibung der Tests der erstellten Anforderungsspezifikation durch Validierung. Es wird dabei auf die Erstellung von Simulationsszenarien für das erstellte SysML-Modell eingegangen. In Kapitel 6 folgt zum Schluss die Zusammenfassung und ein Ausblick. Im Anhang sind u.a. die sprachlichen Anforderungen und Testfälle enthalten.

2 Stand der Wissenschaft und Technik

In diesem Kapitel wird eine Übersicht über die bisherige Verwendung modellbasierter Anforderungsspezifikationen in der Wissenschaft und Technik gegeben. Der Themenbereich der Eisenbahnleit- und Sicherungstechnik liegt dabei im Fokus.

2.1 Bisherige Arbeiten über modellbasierte Anforderungsspezifikationen im Bahnbereich

Wissenschaftlich wurde die Verwendung modellbasierter Spezifikationen von Anforderungen bereits in nachfolgenden Arbeiten behandelt.

Saeid Arabestani beschreibt in [1] eine ausführliche Methodik zur Entwicklung einer Systemanforderungsspezifikation auf Basis der Objektorientierung. Die Darstellung dieser Spezifikation erfolgt mittels der Unified Modeling Language (UML). Anhand eines Systems aus der Eisenbahnsicherungstechnik wird gezeigt, wie ausgehend von den funktionalen Anforderungen die Komponenten des Systems als Objekte identifiziert und mit Hilfe passender UML-Sprachelemente modelliert werden können. Dabei werden auf die Systemstruktur, das Systemverhalten und auf die Kommunikation der Systemkomponenten untereinander eingegangen. Die Systemstruktur wird durch Klassendiagramme, das Systemverhalten durch Zustandsautomaten und die Kommunikation mittels Sequenzdiagrammen modelliert. Eine formale Verifikation der erstellten Systemanforderungsspezifikation wurde mit Model-Checking durchgeführt.

Oliver Lemke stellt in [14] ein Konzept vor, bei dem eine passende Struktur für modellbasierte Anforderungsspezifikationen von sicherheitskritischen Bahnsystemen entworfen wird. Diese Struktur wird als eine Teilmenge der Systems Modeling Language (SysML) herausgearbeitet, welche diejenigen SysML-Sprachelemente enthält, die im Bahnbereich besonders geeignet sind. Im Mittelpunkt steht dabei das funktionale Modell, welches verschiedene auf Sichten auf das System bereitstellt.
Dabei sind u.a. eine Systemabgrenzungs-, Systemfunktions- und Systemverhaltenssicht beschrieben, welche durch Blockdefinitions-, Anwendungsfall- und Aktivitätsdiagramme der SysML modelliert werden. Als Ziel des funktionalen Modells wird zu dem eine frühe Ausführbarkeit hervorgehoben, um das Modell testen zu können. Ergänzt wird dies durch ein iterativ-inkrementelles Prozessmodell, das speziell auf die Erstellung modellbasierter Anforderungsspezifikationen zugeschnitten ist. Es kann mit einer geringen Menge an Anforderungen und Informationen begonnen und die Spezifikation flexibel modifiziert werden. In definierten Aufgaben werden schließlich nach und nach die einzelnen Sichten modelliert. Das Gesamtkonzept aus gewählten SysML-Sprachelementen und dem Prozessmodell wird schließlich an einer fiktiven Bahnübergangssteuerung demonstriert.

In [13] stellt Yuen Man Hon im Vergleich zu Arabestani und Lemke eine rein formale Methode zur Anforderungsspezifikation von sicherheitskritischen Bahnsystemen wie Stellwerken vor, die ohne grafische Beschreibungsmittel wie der UML oder SysML auskommt. Dabei können natürlich-sprachliche Formulierungen von Anforderungen als Formeln der Aussagenlogik und Entscheidungstabellen modelliert werden. Bei dieser Arbeit wurde ein sogenanntes „Objektbasiertes Lastenheft“ mit passenden Softwaretool entwickelt, welches speziell Eisenbahningenieuren den Zugang zu formalen bzw. mathematisch beschriebenen Aussagen erleichtert. Mit Hilfe dessen können Anforderungen analysiert, verifiziert und auf Korrektheit überprüft werden.

2.2 Praktischer Einsatz modellbasierter Anforderungsspezifikationen in der Leit- und Sicherungstechnik

Wie bereits eingangs erwähnt, wurden bei der Entwicklung von Systemen der Leit- und Sicherungstechnik der Eisenbahn bisher Anforderungen in sprachlicher Form verwendet. Seit ein paar Jahren ist ein Übergang zu modellbasierten Spezifikationen von Anforderungen zu erkennen.

Im Jahr 2007 wurde bei der Deutschen Bahn AG im Rahmen einer Projektstudie ein Verbesserungspotenial im Bereich des Requirements Engineering festgestellt. Darüber wird in [3] ausführlich berichtet. Aus dieser Studie heraus startete das Projekt mit dem Namen „Datenbank für Anforderungen“ (DAFA), um konkrete Verbesserungen in der Handhabung der betreiberseitigen Anforderungen an sicherungstechnische Systeme vorzunehmen. Dabei soll insbesondere die Konformität zu den CENELEC-Normen EN 50126, 50128 und 50129 sichergestellt werden.

Das konkrete Vorgehen zur Verbesserung des Requirements Engineering bei DAFA wird in [3] darstellt. Dieses enthält und beschreibt u.a. die Möglichkeit der modellgetriebenen Spezifikation von Anforderungen. Herausgestellt werden hier ebenfalls die Fehlerquellen und Probleme, die sich aus Anforderungen in natürlicher Sprache ergeben können. Dies spielt bei Eisenbahnsicherungssystemen eine besondere Rolle, da dessen Sicherheitsrelevanz von zentraler Bedeutung ist. Anforderungen, die sicherheitskritische Bereiche im System betreffen, dürfen weder fehlerbehaftet, noch missverständlich formuliert sein. Vorgeschlagen wird eine Vorgehensweise mit Hilfe eines grafischen Beschreibungsmittels in der Phase der Anforderungsanalyse. Dabei soll eine ausführbare Anforderungsspezifikation entstehen, an der Fehler frühzeitig durch Modellsimulation und Tests erkannt werden können. Als Beschreibungsmittel wird dabei die SysML empfohlen. Das Modell soll dabei aus der Beschreibung der Struktur und des Verhaltens des Systeme bestehen. SysML-Blockdefinitionsdiagramme und interne Blockdiagramme dienen dabei für die Darstellung der Struktur und SysML-Zustandsautomaten zur Darstellung des Verhaltens. Unstimmigkeiten, Missverständnisse oder Fehler in den Anforderungen können dadurch noch vor der Phase des Systemdesigns und der Implementierung bereinigt werden.

Signifikant für die dargestellte Vorgehensweise ist das umfangreiche Projekt „NeuPro“

(Neuausrichtung Produktionssteuerung) der DB Netz AG. Dieses wurde im Jahr 2008 mit dem Ziel gestartet, die bei der Deutschen Bahn eingesetzte elektronische Stellwerkstechnik zu standardisieren. Dazu werden modulare Stellwerksteilsysteme und zugleich einheitliche Schnittstellen zwischen diesen Teilsystemen spezifiziert. Die dabei zu erstellenden modellbasierten Spezifikationen werden aus sprachlichen Anforderungen heraus entwickelt. Als Beschreibungsmittel kommt die SysML zum Einsatz.

Als eines der Ergebnisse von NeuPro ergibt sich beispielsweise, dass einzelne Komponenten eines Stellwerks von unterschiedlichen Herstellern geliefert werden können. Mit den standardisierten Schnittstellen der Komponenten bzw. Teilsystemen ist dann eine Verbindung dieser untereinander möglich. Der Aufbau des Stellwerk-Gesamtsystems obliegt dann dem Systembesteller bzw. Betreiber. Im Gegensatz dazu mussten Stellwerke bisher immer als Gesamtsystem von einem einzigen Hersteller geliefert werden, welches alle benötigten Komponenten enthält.

Während der bisherigen Arbeiten an den Schnittstellenspezifikationen und Teilsystemen entstand dafür der sogenannte Modellierungsstandard von NeuPro [9]. Dies ist ein Regelwerk, welches allgemeine Vorgaben über die einzusetzenden SysML-Sprachelemente macht und bei der Modellierung neuer Schnittstellen und Teilsysteme anzuwenden ist. Zudem sind einheitliche Benennungskonventionen für die SysML-Sprachelemente definiert. Ziel ist es ebenfalls, eine ausführbare Spezifikation zu erstellen.

3 Domänenspezifische Grundlagen der Eisenbahn

Für die nachfolgenden Kapitel wird ein Verständnis von fachlichen Aspekten der Eisenbahn benötigt. Dazu folgen nun Begriffsbestimmungen zum Bahnbetrieb und insbesondere zur Leit- und Sicherungstechnik. Aufgrund der Komplexität des Eisenbahnwesens konzentrieren sich die Ausführungen darauf, was für die Beschreibung und Modellierung der Fahrwegsuche wichtig ist.

3.1 Begriffe des Bahnbetriebs

Bevor das Augenmerk hauptsächlich auf dem Bereich der Leit- und Sicherungstechnik liegt, werden zunächst grundlegende Begriffe des Bahnbetriebs eingeführt. Zu diesem Zweck sind nachfolgende Definitionen aufgelistet, die sowohl aus den betrieblichen Regelwerken der DB Netz AG, als auch der Eisenbahnfachliteratur entnommen und sinngemäß wiedergegeben sind. Zusätzlich sind zum besseren Verständnis Ergänzungen anhand von Abbildungen enthalten.

Bahnhof Ein Bahnhof ist eine Eisenbahnanlage mit mindestens einer Weiche, in dem Züge u.a. beginnen, halten und enden [8].

Freie Strecke Die freie Strecke ist eine Bahnanlage außerhalb von Bahnhöfen. Sie besteht aus Gleisen, die von Zügen nach einem Fahrplan befahren werden [18].

Zugfahrt Eine Zugfahrt ist eine Fahrt von einem einzelnen Eisenbahnfahrzeug oder einer aus mehreren Fahrzeugen zusammengesetzten Einheit, welche von einem Bahnhof auf die freie Strecke übergeht oder innerhalb eines Bahnhofs nach einem konkreten Fahrplan verkehrt [8].

Rangierfahrt Rangierfahrten sind Fahrten von Fahrzeugen innerhalb von Bahnhöfen, die zum Umsetzen von Fahrzeugen oder Bilden von Zügen dienen [18].

Signal Der Begriff Signal hat zwei unterschiedliche, aber themenverwandte Bedeutungen. Ein Signal aus sicherungstechnischer Sicht ist eine ortsfeste, technische Einrichtung am Gleis. An diesem werden sichtbare Zeichen angezeigt. Diese Zeichen, welche Informationen im betrieblichen Sinne den an den Triebfahrzeugführer übermitteln, werden jedoch ebenfalls als Signal bezeichnet. [7] Welche Bedeutung der Begriff Signal in einer Verwendung konkret hat, ist aus dem gegebenen Kontext zu entnehmen.

Signalbild Das Signalbild ist das Aussehen eines sichtbaren Zeichens, das an einem Signal als ortsfeste Einrichtung gezeigt wird. Es wird anhand von Formen, Farben oder Symbolen beschrieben [7].

Hauptsignal Hauptsignale sind Signale, die anzeigen, ob der dahinter liegende Gleisabschnitt befahren werden darf. Zugfahrten werden nur an Hauptsignalen zugelassen [18]. Die Abbildung 3.1 zeigt ein Hauptsignal in der konstruktiven Ausführung als Lichtsignal.

Abbildung 3.1: Ein Hauptsignal, welches mit einem roten Lichtpunkt „Halt“ anzeigt.

Sperrsignal Sperrsignale signalisieren, ob in den dahinter liegenden Gleisabschnitt eingefahren und rangiert werden darf [8]. Sie gelten für Zug- und Rangierfahrten, können aber keine Zugfahrten zulassen [18]. Ein Sperrsignal als Lichtsignal ist in Abbildung 3.2 dargestellt.

Vorsignal Ein Vorsignal (Abbildung 3.3) ist ein Signal, welches das nachfolgende Hauptsignal ankündigt [18].

Einfahrsignal Einfahrsignale sind Hauptsignale, die die Grenze zwischen Bahnhof und freier Strecke darstellen [8].

Ausfahrsignal Ausfahrsignale sind Hauptsignale, die Ausfahrten aus einem Bahnhof auf die freie Strecke sichern [18].

Blocksignal Blocksignale sind Hauptsignale, die Einfahrten in Abschnitte der freien Strecke (Blockabschnitte) sichern [18].

Abbildung 3.2: Ein Sperrsignal in niedriger Bauform.

Abbildung 3.3: Ein Vorsignal, welches „Langsamfahrt erwarten“ anzeigt.

Lichtsignal Haupt-, Sperr- und Vorsignale, die Signalbilder mittels farbiger, leuchtender Lichtpunkte darstellen, heißen Lichtsignale [12].

Signalsystem Signalbilder, die an Haupt- und Vorsignalen gezeigt werden, lassen sich zu einzelnen Systemen zusammenfassen. Dabei sind diese anhand der gegebenen betrieblichen Information der Signalbilder aufeinander abgestimmt.

H/V-Signalsystem Das **H**aupt-/**V**orsignalsystem ist ein Signalsystem, welches aus getrennten Haupt- und Vorsignalen besteht.

Abbildung 3.4: Ein Hauptsignal des H/V-Signalsystems als Lichtsignal. Am oberen Signalschirm wird durch ein grünes und gelbes Licht der Auftrag „Langsamfahrt“ signalisiert. Darunter ist ein Vorsignal angebracht, das „Fahrt erwarten“ mittels zwei grüner Lichter anzeigt.

Weiche Aufgrund der konstruktiven Merkmale von Weichen gibt es spezielle Bezeichnungen für dessen einzelne Komponenten [15]. Die für diese Arbeit wichtigen Bezeichnungen sind in Abbildung 3.5 dargestellt. Weil es unterschiedliche Weichenbauarten gibt und dabei nicht immer ein gerader Strang vorhanden sein muss (z.B. eine Weiche liegt in einer Kurve), werden nachfolgend die allgemeineren Bezeichnungen *linker und rechter Strang* verwendet. Die Betrachtung erfolgt dabei von der Weichenspitze aus. Befährt ein

Fahrzeug den linken Strang einer Weiche, so befindet sich diese in Linkslage. Analog dazu ist beim Befahren des rechten Strangs die Weiche in Rechtslage eingestellt. Heutzutage wird dies mit „L“ für Linkslage und „R“ für Rechtslage abgekürzt [15]. Ebenfalls üblich ist auch die Bezeichnung der Weichenstränge mit „Plusstrang“ und „Minusstrang“, bei der eine Grundstellung einer Weiche mit dem Plus-Zeichen „+“ definiert ist. Der andere ist dann mit dem Minus-Zeichen „-“ ausgewiesen [15, 22].

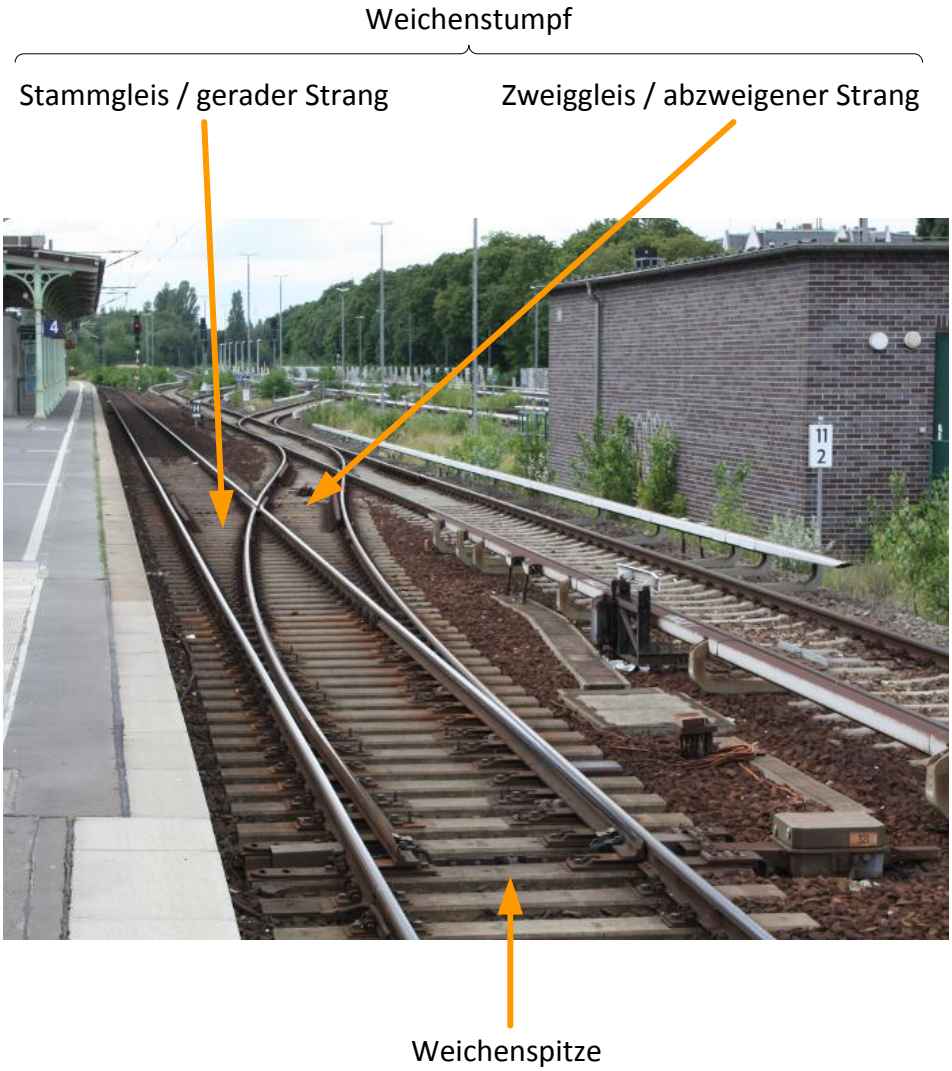

Abbildung 3.5: Eine Weiche in Linkslage.

Sicherungstechnischer Übersichtsplan Zur grafischen Verdeutlichung der eben beschriebenen Begriffe und Zusammenhänge des Bahnbetriebs ist in Bild 3.6 ein sogenannter sicherungstechnischer Übersichtsplan eines beispielhaften Bahnhofs dargestellt. Dieser wird aus sicherungstechnischen Lageplänen abgeleitet, welche die wichtigsten Pläne in der Leit- und Sicherungstechnik sind [15]. Die Symbolik und Beschriftungen der Signale, Weichen usw. entsprechen den üblichen Bezeichnungen [siehe 15, 18]. In den folgenden Abschnitten und Kapiteln werden bestimmte Sachverhalte anhand von sicherungstechnischen Übersichtsplänen erklärt.

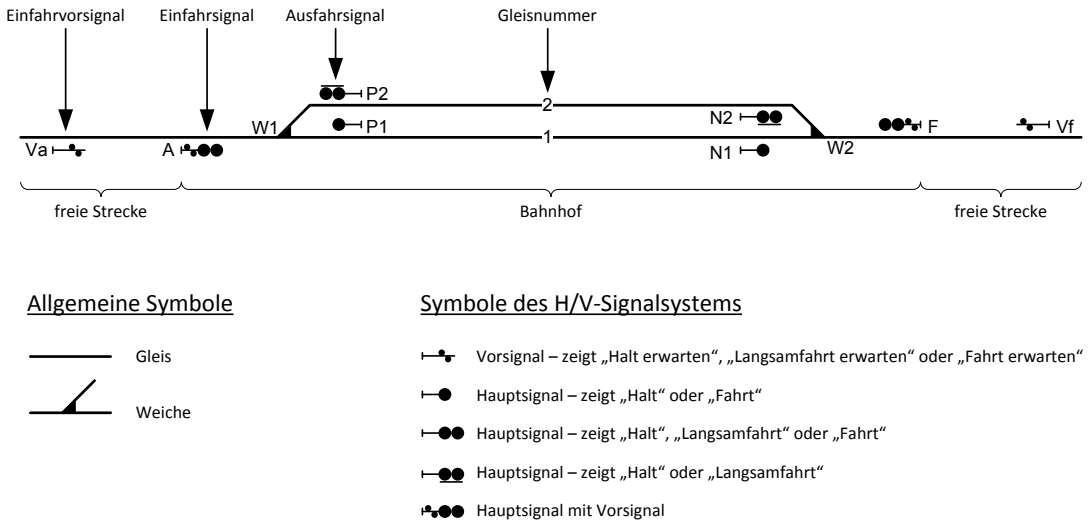

Abbildung 3.6: Lageplan eines Beispiel-Bahnhofs.

3.2 Fahrwege, Fahrstraßen und Stellwerke

Fahrweg Ein Fahrweg ist der durch die Spurführung der Eisenbahn vorgegebene Weg, auf dem sich ein Schienenfahrzeug bewegen kann. Fahrwege bestehen aus Gleisen und die jeweils in einer Stellung festlegten Weichen, wobei Gleise und Weichen miteinander verbunden sind [15].

Fahrstraße Die Fahrstraße ist ein technisch gesicherter Fahrweg [18]. Sie hat einen definierten Start- und Zielpunkt [15] und enthält alle Gleise und Weichen, die zwischen Start- und Zielpunkt liegen. Obwohl es auch Fahrstraßen ohne Zielpunkt gibt [18], liegt in dieser Arbeit das Augenmerk auf dem Fall, bei dem ein Zielpunkt vorhanden ist.

Grundsätzlich werden bei Fahrstraßen für Zug- und Rangierfahrten zwischen *Zugstraßen* und *Rangierstraßen* unterschieden. Der wesentliche Unterschied ist, dass bei Rangierstraßen der Fahrweg zum Bilden von Zügen oder Umsetzen von Wagen nicht frei von anderen Fahrzeugen sein muss [18]. Der Startpunkt einer Zugstraße kann ein Hauptsignal

sein oder ein Sperrsignal, wenn die Zugstraße hinter dem Sperrsignal an einem Hauptsignal vorbeiführt [10, 18]. Zielpunkte von Zugstraßen können Haupt- oder Sperrsignale sein. Bei Rangierstraßen sind als Start- und Zielpunkte Haupt- oder Sperrsignale möglich [10].

Fahrstraßen spielen eine zentrale Rolle in der Sicherungstechnik und müssen daher für die Zulassung und Sicherung einer Fahrt besondere Sicherheitsanforderungen erfüllen. Diese Anforderungen werden durch entsprechende technische Maßnahmen und Einrichtungen realisiert (Stellwerk). Weil dies jedoch über das Thema dieser Arbeit hinausgeht, sei für ein ausführliches und weitergehendes Wissen diesbezüglich an dieser Stelle auf [18, 15] verwiesen.

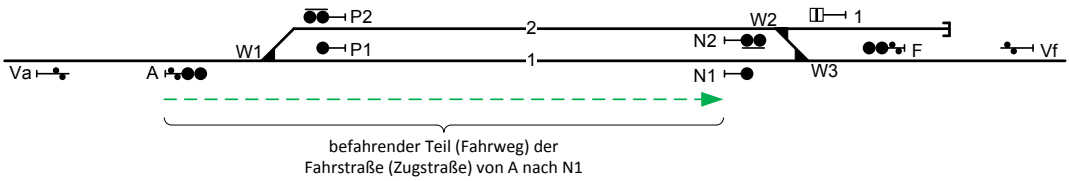

Abbildung 3.7: Zugstraße von Signal A nach Signal N1

Abbildung 3.7 zeigt anhand eines Übersichtplans eine Zugstraße vom Einfahrsignal A über die Weiche W1 in Rechtslage zum Ausfahrsignal N1. Eine Rangierstraße (Abbildung 3.8) ist vom Sperrsignal 1 zum Signal P1 möglich.

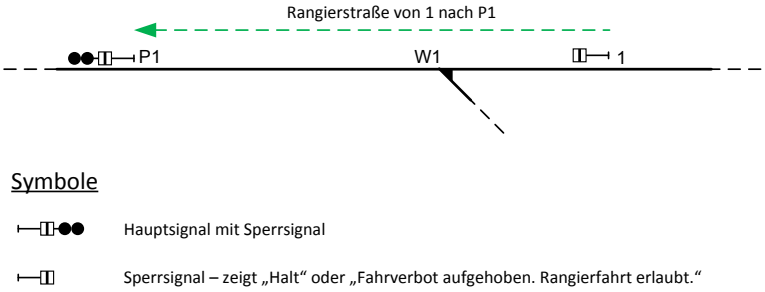

Abbildung 3.8: Rangierstraße von einem Sperr- zu einem kombinierten Haupt- und Sperrsignal

Umfahrstraße Existieren von einem Start- zu einem Zielpunkt einer Fahrstraße mehrere mögliche Fahrwege, so gibt es keine eindeutige Fahrstraße zwischen diesem Start- und Zielpunkt. Deshalb unterscheidet man in diesen Fällen zwischen definierter Regelfahrstraße und sogenannter Umfahrstraße [12, 10]. Ein mögliche Situation illustriert Bild 3.9. Die Regelfahrstraße von Signal A zu Signal N1 führt hier über die Weichen W1 und W4. Die Umfahrstraße nimmt den Weg über W1, W2, W3 und W4 und „umfährt“ den direkten Weg zwischen W1 und W4.

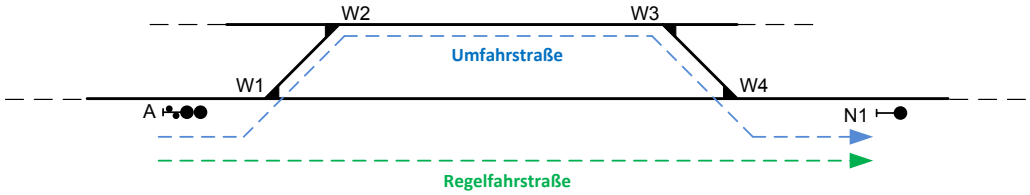

Abbildung 3.9: Regelfahrstraße (grün) und Umfahrstraße (blau) von Einfahrsignal A zu Ausfahrsignal N1

Stellwerk Das Prinzip der Fahrstraße wird durch Stellwerke realisiert, welche eine der wichtigsten Einrichtungen in der Eisenbahninfrastruktur und für den Bahnbetrieb sind. In [18] ist dafür folgende Definition gegeben:

> „Stellwerke sind Anlagen zum zentralisierten Einstellen und Sichern der Fahrwege für Zug- und Rangierfahrten. Die Stellwerksanlage enthält eine Sicherungslogik zur Herstellung von Abhängigkeiten zwischen den zu steuernden Fahrwegelementen zum Ausschluss von Gefährdungen.“

Im Laufe der Zeit haben sich unterschiedliche Bauformen von Stellwerken entwickelt, die zum Zeitpunkt ihrer Entwicklung dem jeweiligen Stand der Technik entsprachen. Im Folgenden liegt der Fokus auf den Relaisstellwerken.

3.3 Relaisstellwerke nach dem Spurplanprinzip

Stellwerke, die die Sicherungslogik durch rein elektrische Schaltungen mittels speziell dafür entwickelter Relais realisieren, werden als Relaisstellwerke oder auch elektrische Stellwerke bezeichnet [10, 18, 15]. Eine Bauform von Relaisstellwerken stellen die Spurplanstellwerke nach dem sogenannten *Spurplanprinzip* dar, welches auch geografisches oder topologisches Prinzip genannt wird [12, 18, 15]. Spurplanstellwerke bestehen aus einer Außen- und Innenanlage, die zusammen eine technische Einheit sind [10].

Die technische Funktionalität ist dabei in sogenannten *Relaisgruppen* organisiert, welche genormte, vorgefertigte und universell einsetzbare Einzelbauteile darstellen und jeweils eine Gruppierung von mehreren Relais enthalten [10, 19]. In Abbildung 3.11 ist eine solche dargestellt. Im Stellwerksgebäude (Innenanlage) sind die Relaisgruppen zentral im sogenannten Relaisraum untergebracht und bilden das Kernstück des Spurplanstellwerks [10].

Es gibt unterschiedliche Typen von Relaisgruppen, die für jeweils einen vorbestimmten Einsatzzweck konzipiert sind [19]. Jedem Element des Gleisfeldes bzw. Gleisplans wie Weichen und Signalen wird im Spurplanstellwerk eine konkrete Relaisgruppe passenden Typs zugeordnet [10, 22]. Wichtige Relaisgruppen-Typen sind dabei die *Zugstraßengruppe* für Zugstraßen und die *Rangierstraßengruppe* für Rangierstraßen, welche die Signale als Start- und Zielpunkte von Fahrstraßen repräsentieren. Relevant ist auch die *Weichengruppe*, welche als Repräsentation einer Weiche dient. Obwohl es noch weitere Typen gibt

Abbildung 3.10: SpDrS60-Stellwerk Hamburg Hbf [20]. Dieses Stellwerksgebäude entspricht der typischen Bauweise von Relaisstellwerken, bei der im obersten Stockwerk der Bedienraum angeordnet ist [10]. Darunter befindet sich u.a. der Relaisraum.

Abbildung 3.11: Eine Relaisgruppe der Firma Siemens. Im Inneren des Gehäuses sind die einzelnen Relais für Schaltaufgaben untergebracht.

[siehe 22, 19], liegt in dieser Arbeit der Fokus auf den Relaisgruppen für Fahrstraßen und Weichen. Diese sind für die Betrachtung der Fahrwegsuche von Bedeutung.
Mittels sogenannter *Spurkabel*, an deren Enden spezielle *Spurkabelstecker* montiert sind [11], werden alle Relaisgruppen genau so miteinander verbunden und steckbar zusammengeschaltet, wie es der tatsächlichen Anordnung der Weichen und Signale im Gleisfeld entspricht [10, 22]. In der Stellwerksinnenanlage ist also ein technisches Abbild der Außenanlage realisiert, das als *Gruppenverbindungsplan* bzw. *Spurplan* [10, 22, 19, 11] oder *Elementverbindungsplan* [15] bezeichnet wird.
Bekannte Vertreter von Spurplanstellwerken sind beispielsweise die Bauformen „SpDrS60“ [19, 11] und dessen Weiterentwicklung „SpDrS600“ [23] der Firma Siemens. Die Abkürzung steht für **Sp**urplan-**Dr**ucktasten-Stellwerk **S**iemens Entwicklungsjahr 19**60** [10] bzw. 600. Der Begriff Drucktaste deutet auf die Bedienung des Stellwerks durch drückbare Tasten auf der Bedieneinrichtung hin (z.B. zur Start-Zielbedienung einer Fahrstraße). Beide Stellwerkstypen arbeiten mit dem H/V-Signalsystem. Das SpDrS60-Stellwerk wurde ab 1963 eingesetzt [2]. Zwei Bahnhöfe, die mit dieser Stellwerksbauform ausgestattet wurden, sind beispielsweise Hamburg Hbf [20] (siehe auch Bild 3.10) und München Hbf [24]. Gegen Ende 1970er Jahre wurde schließlich das SpDrS600-Stellwerk entworfen, welches ab dem Jahr 1979 [4] verwendet wurde. Ein Beispiel für die SpDrS600-Bauform ist das Zentralstellwerk Nürnberg Hbf [21].

In beiden Stellwerksbauformen werden die Relaisgruppen, die für Fahrstraßenstart- und Zielpunkte zuständig sind, als *Zugstraßengruppen* und *Rangierstraßengruppen* entsprechend für Zug- und Rangierstraßen bezeichnet [22, 19]. Weichen werden durch *Wei-*

chengruppen repräsentiert.

3.3.1 Gruppenverbindungsplan

Ein Beispiel für einen Gruppenverbindungsplan ist in Abbildung 3.12 illustriert. Die Symbolik ist dabei angelehnt an [15]. Dieser Gruppenverbindungsplan repräsentiert den Gleis- bzw. Lageplan des Beispiel-Bahnhofs in Bild 3.6. Vergleichend wird der Zusammenhang zwischen Lageplan und Gruppenverbindungsplan deutlich.

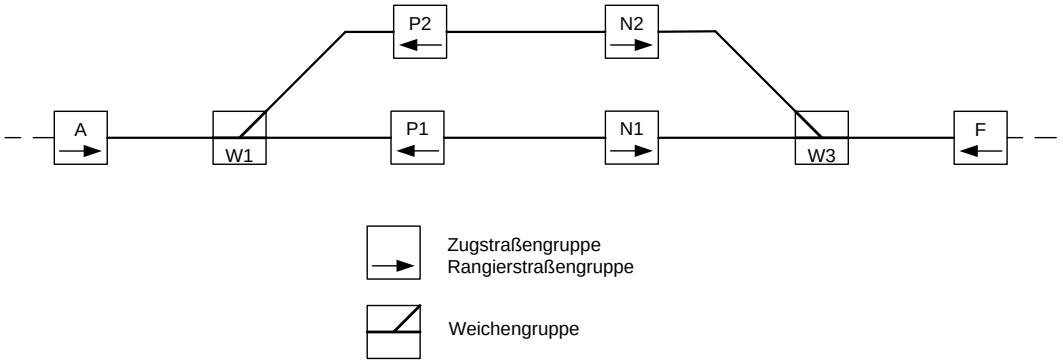

Abbildung 3.12: Gruppenverbindungsplan des Beispiel-Bahnhofs

Die Pfeile an den Zugstraßen- und Rangierstraßengruppen geben an, in welche Fahrtrichtung das Hauptsignal gilt, bzw. zeigen die Richtung an, in der sich der dahinter liegende Gleisabschnitt befindet, den das Hauptsignal sichert. Diese Richtung wird nachfolgend, insbesondere in Kapitel 4, als *Geltungsrichtung* bezeichnet.

3.3.2 Objektorientierung

Ein auffallender Aspekt ergibt sich bei der genaueren Betrachtung des Konzepts der verschiedenen Relaisgruppen-Typen und ihre Verbindung untereinander mit dem Gruppenverbindungsplan. Wie bereits beschrieben, existieren verschiedene Relaisgruppen-Typen für das entsprechende, zu repräsentierende Element der Fahrstraße. Dabei ist das Verhalten für jede Relaisgruppe desselben Typs gleich. Beispielsweise arbeiten alle Weichengruppen identisch, obwohl jede Weichengruppe einer anderen Weiche im Gleisfeld zugeordnet ist. Im Gruppenverbindungsplan werden also Relaisgruppen jedes Typs jeweils einem passenden Element im Gleisplan zugewiesen, quasi „instanziert“ bzw. einem konkreten Objekt zugeordnet. Jede Relaisgruppe stellt dann ein eigenständiges Objekt mit vorgegebener Funktionalität dar.

Diese Aspekte sind übertragbar auf die Entwurfsprinzipien der objektorientierten Programmierung. Dabei können die Relaisgruppen-Typen als Klassen betrachtet werden. Die Spurkabelstecker an den Relaisgruppen sind die Schnittstelle nach außen, über die die

Relaisgruppen Informationen auf die Spurkabel in Form von Stromflüssen legen. Die interne Funktionalität und der Zustand der enthaltenden Relais einer Relaisgruppe ist nach außen nicht sichtbar, was dem Geheimnisprinzip bzw. der Modularisierung entspricht.

3.3.3 Fahrwegsuche nach dem Such-Echostrom-Prinzip

Die *Fahrwegsuche* ist ein Verfahren zum dynamischen Ermitteln des Fahrwegs zwischen dem Start- und Zielpunkt einer Fahrstraße mit Hilfe des Gruppenverbindungsplans. Es ist elektrotechnisch implementiert und arbeitet mit elektrischen Strömen. Im SpDrS600-Stellwerk wird es auch als *Fahrstraßenanwahl* bezeichnet [22] und besteht aus sogenannten Such- und Echoströmen. [10, 22] Ein Suchstrom wird von einer Zug- oder Rangierstraßengruppe (Startpunkt) ausgelöst und läuft zu allen möglichen Gruppen, die ein Zielpunkt der einzustellenden Fahrstraße sein können. An Weichengruppen verzweigt sich der Suchstrom. Der Echostrom wird von der Zielgruppe ausgelöst und läuft zurück zur Startgruppe. Dadurch wird der Fahrweg zwischen Start- und Zielgruppe eindeutig festgelegt. Zusammenfassend wird diese Vorgehensweise als *Such-Echostrom-Prinzip* bezeichnet [22].

Das Verfahren ist der erste von mehreren Schritten der Fahrstraßenbildung, der nach der Festlegung des Fahrstraßenstart- und Zielpunktes durch die Bedieneinrichtung des Stellwerks stattfindet [22]. Die weiteren Schritte bleiben in der Arbeit außer Betracht, da diese über das Thema der Fahrwegsuche hinausgehen und daher nicht relevant sind. Anhand des nachfolgenden Beispiel-Gruppenverbindungsplans in Abbildung 3.13 wird die Fahrwegsuche erläutert, bei dem die Ausführungen an eine Schaltungsbeschreibung des SpDrS600-Stellwerks [22] angelehnt und durch Erläuterungen aus [12] ergänzt sind.

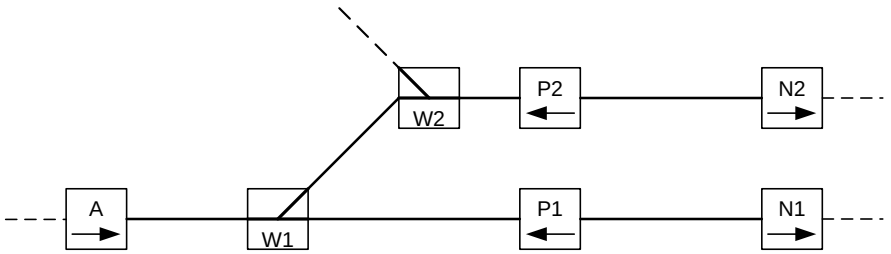

Abbildung 3.13: Gruppenverbindungsplan für die Erläuterung der Fahrwegsuche

Zur Erklärung der Fahrwegsuche soll der Fahrweg der Fahrstraße von Signal A zu Signal N2 ermittelt werden. Dazu ist die Zugstraßengruppe vom Signal A als Start und diejenige vom Signal N2 als Ziel markiert worden. Die Fahrwegsuche lässt sich in zwei Phasen aufteilen, bei denen der Verlauf des Such- und Echostrom jeweils einzelnen betrachtet wird.

Der erste Schritt in der Suchstrom-Phase (Abbildung 3.14) besteht darin, dass durch die Zugstraßengruppe des Signals A ein Suchstrom (rote, gestrichelte Linie) in die Richtung

ausgelöst wird, in der Signal A Startpunkt einer Fahrstraße sein kann (Geltungsrichtung). Der Suchstrom trifft dann auf die Weichengruppe W1 an der Weichenspitze. Die Weichengruppe W1 verzweigt den Suchstrom und leitet ihn an den linken und rechten Strang weiter. Der Suchstrom erreicht dann die Weichengruppe W2 im linken Strang und trifft andererseits auf die Zugstraßengruppe P1. Letztere ist nicht aktiv an der Fahrwegsuche beteiligt und leitet deshalb den Suchstrom Richtung Signal N1 weiter. An der Weichengruppe W2 wird der Strang, an dem der Suchstrom eingetroffen ist (hier der linke Strang) eingespeichert. W2 leitet den Suchstrom weiter an die Zugstraßengruppe des Signals P2, welche auch nicht aktiv an der Fahrwegsuche beteiligt ist. Schließlich erreicht der Suchstrom auch die Zugstraßengruppe N2.

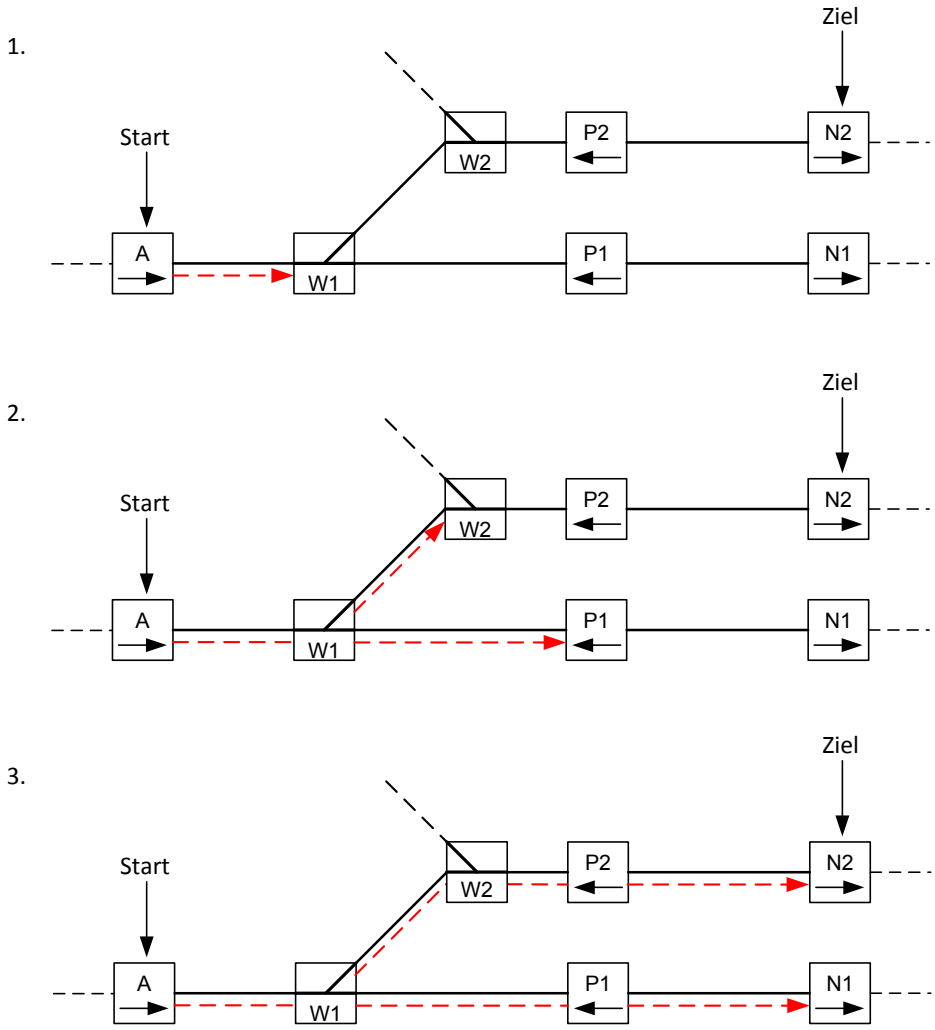

Abbildung 3.14: Suchstrom-Phase der Fahrwegsuche

In der Echostrom-Phase (Abbildung 3.15) wird der Echostrom ausgelöst. Da als Ziel

das Signal N2 gewählt wurde, löst die Zugstraßengruppe N2 nach dem Eintreffen des Suchstroms einen Echostrom aus (grüne, gestrichelte Linie). Dieser läuft über die Zugstraßengruppe P2 zur Weichengruppe W2 zurück. Da an W2 bereits gespeichert wurde, dass der Suchstrom vom linken Strang kam, nimmt der Echostrom nur diesen Weg zurück. An der Weiche W1 nimmt der Echostrom den Weg zur Weichenspitze. Der Suchstrom in Richtung der Zugstraßengruppe N1 wird abgeschaltet, da er nicht mehr benötigt wird. Schließlich erreicht der Echostrom die Zugstraßengruppe des Startsignals A. Damit ist der Fahrweg von A nach N2 eindeutig festlegt.

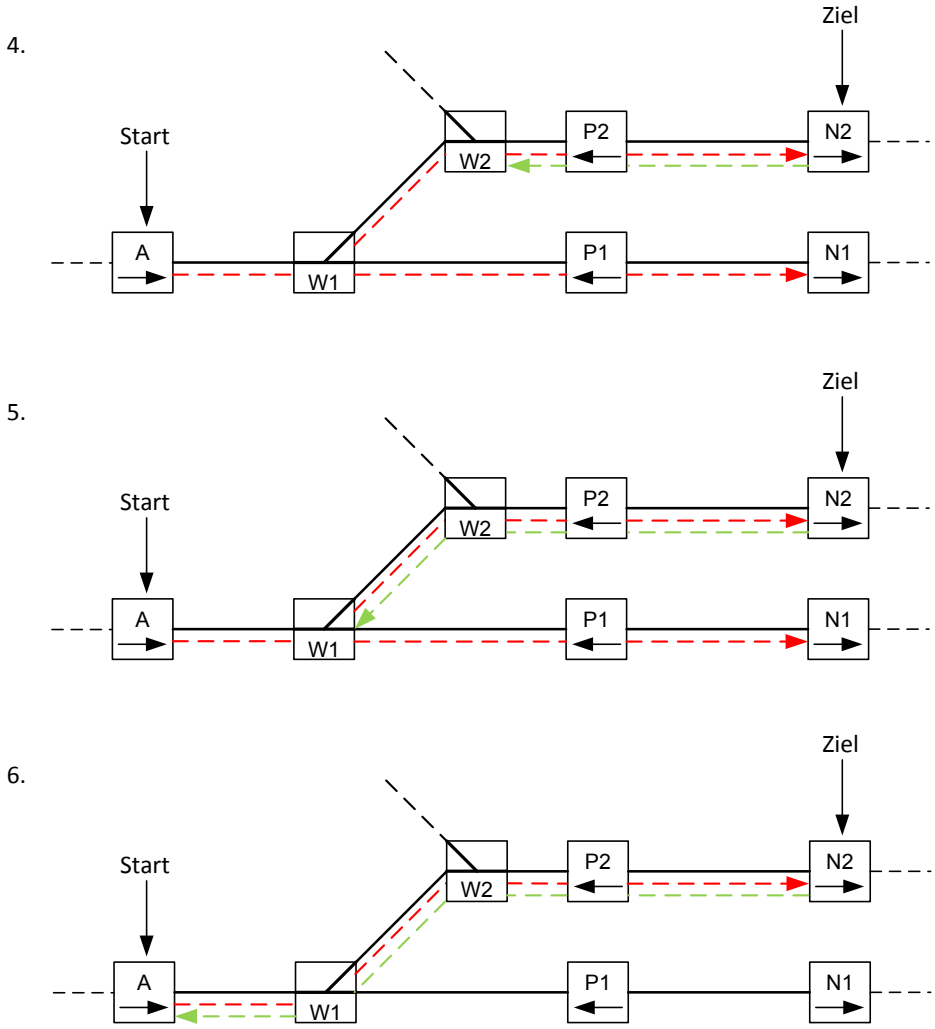

Abbildung 3.15: Echostrom-Phase der Fahrwegsuche

Mit der Beschreibung der Fahrwegsuche sind nun alle wichtigen Inhalte für die zu erstellende, modellbasierte Anforderungsspezifikation erläutert worden. Im nächsten Kapitel erfolgt dann die ausführliche Modellierung der Fahrwegsuche auf Grundlage der in

diesem Kapitel eingeführten Hintergründe der Eisenbahn.

4 Modellierung der Fahrwegsuche

Dieses Kapitel enthält mit der Modellierung der Fahrwegsuche den Hauptbestandteil dieser Arbeit. In den nachfolgenden Schritten wird aus gegebenen, sprachlich formulierten Anforderungen der Fahrwegsuche eine modellbasierte und ausführbare Anforderungsspezifikation entwickelt. Ziel ist es, eine Modellierung zu entwerfen, mit der ein Gruppenverbindungsplan konstruiert bzw. nachgebildet werden kann. Zuvor werden benötigte Rahmenbedingungen festgelegt und die Modellierung in das V-Modell der EN 50126 eingeordnet. Anschließend erfolgt die Modellierung der Struktur und des Verhaltens.

4.1 Einordnung in das V-Modell nach EN 50126

Bei der Entwicklung sicherheitskritischer Systeme im Bahnbereich nehmen die CENELEC-Normen EN 50126, 50128 und 50129 eine zentrale Rolle ein. Die Norm EN 50126 [5] legt dabei einen Schwerpunkt auf den sogenannten Systemlebenszyklus, der den kompletten Weg eines Systems von dem Entwurf und der Konzeption über die Entwicklung, Fertigung und Inbetriebnahme bis zur Stilllegung beschreibt. Abbildung 4.1 zeigt das V-Modell mit den einzelnen Stufen des Systemlebenszyklus.

Die Ermittlung der Anforderungen in sprachlicher Form als auch die Erstellung einer modellbasierten Anforderungsspezifikation sind in die Phase 4 des V-Modells (*Anforderungen an das System*) einzuordnen.

4.2 Rahmenbedingungen

Für die Modellierung werden nachfolgend Rahmenbedingungen domänenspezifischer Art und für das Beschreibungsmittel des zu erstellenden Modells festgelegt.

4.2.1 Funktionale Anforderungen in sprachlicher Form

Die modellbasierte Anforderungsspezifikation der Fahrwegsuche wird aus Anforderungen in Textform entwickelt, welche nachfolgend hergeleitet werden. Wie im vorherigen Kapitel beschrieben, wurde die Fahrwegsuche in Relaisstellwerken elektrotechnisch implementiert. Als fachlicher Ausgangspunkt der sprachlich formulierten Anforderungen wird dazu eine Schaltungsbeschreibung des SpDrS600-Stellwerks [22] verwendet, bei der das Such-Echostrom-Prinzip an einem beispielhaften Gruppenverbindungsplan erläutert wird. Weitere Ergänzungen dazu sind [12] entnommen.

Für eine allgemeine und von der Implementierung unabhängige Spezifikation sind das Prinzip und die Eigenschaften der Fahrwegsuche zu abstrahieren. Die Verallgemeinerung

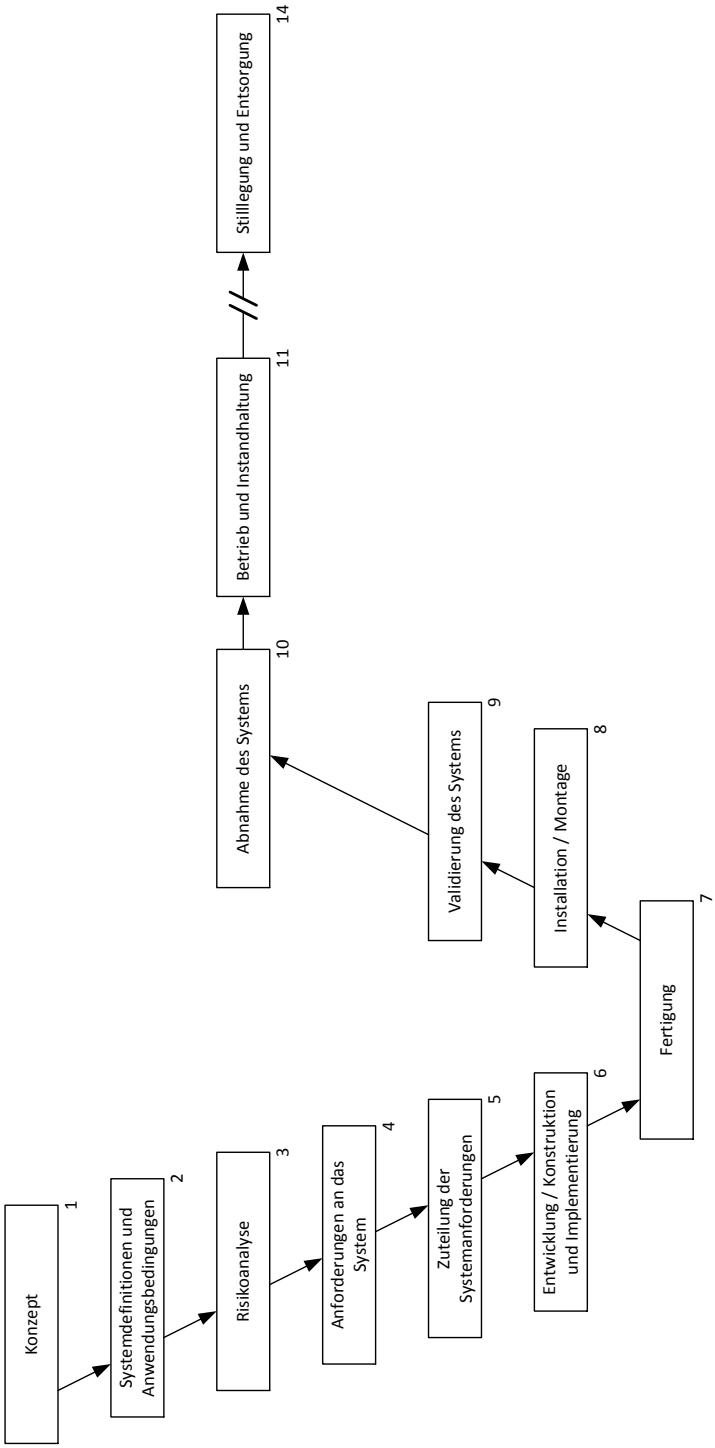

Abbildung 4.1: V-Modell nach EN 50126

des Such-Echostrom-Prinzips und das Verhalten (Schaltvorgänge) der beteiligten Relaisgruppen im Gruppenverbindungsplan ist deshalb möglich, weil damit Prinzipien der objektorientierten Programmierung verwirklicht sind. (siehe Kapitel 3). Folglich kann mit dem Verhalten einer konkreten Relaisgruppe wiederum das Verhalten des passenden Relaisgruppen-Typs allgemeingültig beschrieben werden. Dadurch wird es möglich, das Verhalten der Relaisgruppen im Gruppenverbindungsplan von der Implementierung zu lösen und in Textform zu erfassen. Daraus ergibt sich eine allgemeine, sprachlich formulierte Anforderungsspezifikation.

Desweiteren werden Verallgemeinerungen in der Bezeichnung der Relaisgruppen des Gruppenverbindungsplans getroffen. Die Zugstraßen- und Rangierstraßengruppe werden zu einem *Fahrstraßenelement* zusammengefasst, da die Fahrwegsuche gleichermaßen sowohl bei Zugstraßen, als auch für Rangierstraßen angewendet wird [10, 22]. Die Weichengruppe erhält die Bezeichnung *Weichenelement*. Für die Begriffe Such- und Echostrom wird die abstraktere Bezeichnung Such- und Echosignal gewählt. Der Gruppenverbindungsplan wird fortan als Elementverbindungsplan bezeichnet. Folgende Tabelle fasst die verallgemeinerten Bezeichnungen zusammen:

Fachbegriff	**Verallgemeinerung**
Suchstrom	Suchsignal
Echostrom	Echosignal
Zugstraßengruppe Rangierstraßengruppe	Fahrstraßenelement
Weichengruppe	Weichenelement
Gruppenverbindungsplan	Elementverbindungsplan

An dem nachfolgenden Beispiel ist die Abstraktion von Schaltvorgängen dargestellt, woraus die allgemeinen, sprachlichen Anforderungen hergeleitet sind. Folgende drei Schaltvorgangsbeschreibungen sind [22] entnommen:

> „*Der durch Start- und Zieltastendruck ausgelöste Suchstrom [...] markiert im Fahrweg die Spitze („Zungenseite“) der Weiche 1 [...]* “

> „*Der Echostrom ... schaltet in Weiche 1 das Minuskennzeichen [...] an.* “

> „*Die Verzweigung des Suchstromes wird durch das vom Echostrom angeschaltete Minuskennzeichen der Weiche 1 abgetrennt.*“

Aus diesen Beschreibungen werden folgende allgemeine Anforderungen abgeleitet:

> „Empfängt ein Weichenelement ein Suchsignal aus Richtung der Weichenspitze, so wird das Suchsignal in den linken Strang und in den rechten Strang des Weichenstumpfs weitergeleitet.“

> „Ein empfangenes Echosignal am Weichenelement aus Richtung des Weichenstumpfs (linker Strang) wird nur zur Weichenspitze weitergeleitet und nur dann, wenn an der Weichenspitze vorher ein Suchsignal empfangen wurde. Das Suchsignal an den rechten Strang wird zurückgenommen.“

Alle anderen berücksichtigten Schaltvorgänge sind auf die gleiche Weise verallgemeinert. Eine vollständige Auflistung der Anforderungen in Textform ist im Anhang aufgeführt. Jede Anforderung hat einen eindeutigen Bezeichner, auf den bei der Modellierung jeweils verwiesen wird.

4.2.2 Themenabgrenzung der Fahrwegsuche

Wie in Kapitel 3 angemerkt, ist die Fahrwegsuche der erste und einer von vielen Schritten der Fahrstraßenbildung im Relaisstellwerk. Um diese unabhängig und einzeln betrachtet zu den anderen Schritten modellieren zu können, sind dafür die Anforderungen FA23 und FA24 als Themenabgrenzung zusätzlich definiert.

4.2.3 SysML als Beschreibungsmittel der Anforderungsspezifikation

Eine zentrale Frage ist das zu verwendende Beschreibungsmittel für die modellbasierte Anforderungsspezifikation der Fahrwegsuche. In Kapitel 2 wurde dargelegt, dass die SysML bei der DB Netz AG in Rahmen von „NeuPro“ bereits praktisch eingesetzt wird. Weil die Fahrwegsuche zum gleichen speziellen Themenbereich der Stellwerkstechnik gehört, liegt es nahe, die Fahrwegsuche ebenfalls mit der SysML zu modellieren. Dadurch wird es ebenfalls möglich, den Modellierungsstandard der DB [9] zu verwenden. Eine Spezifikation mit der SysML lässt im Vergleich zur UML zudem offen, ob das System später mittels Hardwarekomponenten oder auf Grundlage von Software implementiert werden kann. Aufgrund der Vorgaben des Modellierungsstandards an die einzusetzenden SysML-Sprachelemente wird deshalb die SysML in der Version 1.2 [16] verwendet. Als Software für die Modellierung und das Zeichnen der SysML-Diagramme wird das Programm *Atego Artisan Studio* in der Version 8.0 eingesetzt.

Die Modellierung setzt sich aus zwei wesentlichen Sichten der Struktur- und Verhaltssicht zusammen, die auch die SysML auf das System bietet [16]. Dabei werden zuerst die einzelnen Komponenten des Elementverbindungsplans modelliert und anschließend ein konkreter Elementverbindungsplan im Zuge der Validierung erstellt (Kapitel 5).

4.3 SysML-Struktursicht

Die Struktursicht der Fahrwegsuche wird mit dem Ziel entwickelt, durch geeignete SysML-Sprachelemente einen Elementverbindungsplan nachzubilden. Dazu sind für das Fahrstraßen- und Weichenelement passende SysML-Sprachelemente anhand der sprachlichen Anforderungen auszuwählen und festzulegen.

Durch die Anforderungen FA01 bis FA08 werden die strukturellen Eigenschaften der Fahrwegsuche beschrieben. Das Fahrstraßen- und Weichenelement sind dabei nach den Prinzipien des Elementverbindungsplans eigenständige Objekte, die miteinander verbunden sind und Informationen austauschen. Dieser Sachverhalt kann mit dem SysML-Sprachelement *Block* abgedeckt werden. Blöcke stellen modulare Einheiten bzw. Elemente in der Systembeschreibung dar, welche über Eigenschaften oder Operationen verfügen [16]. Sie sind ein zentrales Konzept der SysML.

Verbindungspunkte der Fahrstraßen- und Weichenelemente nach außen können über *Flow-Ports* realisiert werden. Flow-Ports sind Interaktionspunkte von Blöcken mit ihrer Umgebung, mit denen z.B. Daten, Energie oder Informationen ausgetauscht werden können [16]. Sie stellen die Schnittstellen eines Blocks dar, wo durch Daten bzw. Informationen in den Block eintreten oder aus dem Block austreten können. In dieser Arbeit werden *atomare* Flow-Ports angewendet, die nur Werte eines einzigen Datentyps übertragen können. Der Modellierungsstandard macht bezüglich der Flow-Ports eine Einschränkung, dass nur solche mit der definierten Richtung „in“ (Eingabe) oder „out“ (Ausgabe) verwendet werden dürfen. Die kombinierte Richtung „inout“ wird nicht angewendet, um insbesondere Such- und Echosignale explizit voneinander zu trennen.

Die Verbindungen der Blöcke untereinander mit Hilfe der Flow-Ports erfolgt über *Konnektoren*, die zwei Flow-Ports miteinander verbinden [16]. Darüber können Informationen von Block zu Block übertragen werden. Die Richtung, in der Daten oder Informationen über einen Konnektor von Block zu Block übertragen werden, wird durch die Richtungen der über einen Konnektor verbundenen Flow-Ports impliziert.

Eine wichtige Frage ist außerdem der zu verwendete Datentyp, der an Flow-Ports und Konnektoren verwendet wird, um Such- und Echosignale auszutauschen. Betrachtet man dazu noch einmal die elektrotechnische Implementierung der Fahrwegsuche im Gruppenverbindungsplan, so wird dabei mit elektrischen Strömen gearbeitet, die entweder fließen oder nicht fließen. Das Fließen bzw. Nichtfließen von Strömen kann passenderweise durch Werte vom Datentyp „Boolean“ dargestellt werden, in dem Sinne, dass ein Suchsignal bzw. Echosignal vorhanden ist oder nicht (Werte „true“ oder „false“). Aus diesem Grunde werden die Flow-Ports, die Such- und Echosignale empfangen oder senden, mit dem Datentyp „Boolean“ definiert.

Zusätzlich fordern die Anforderungen FA06, FA07, FA15 und FA22 eine passende Eingabe bzw. Ausgabe von Informationen an Blöcken. Dies kann ebenfalls mit Flow-Ports realisiert werden, die je nach Einsatzzweck bzw. nach den Anforderungen mit einem anderen Datentypen versehen werden müssen. Hier ist es aufgrund der Anforderungen der Datentyp „String“ für einzelne Zeichen und Zeichenketten. Die einzelnen Blöcke werden nach dem Modellierungsstandard mit internen Blockdiagrammen dargestellt, die die interne Struktur eines Blocks beschreiben [16].

Die nachfolgende Tabelle fasst noch einmal die zu modellierenden Elemente aufgrund der Anforderungen, ihre Entsprechungen in der Implementierung und die dafür verwendeten SysML-Sprachelemente zusammen.

Komponente	Anforderung	SysML-Sprachelement
Spurkabel	FA04	Konnektor
Spurkabelstecker	FA04	Flow-Port
Zugstraßengruppe Rangierstraßengruppe (Fahrstraßenelement)	FA02	Block
Weichengruppe (Weichenelement)	FA03	Block
Stromfluss (Such-/Echosignal)	keine	Datentyp Boolean Wert „true“
kein Stromfluss	keine	Datentyp Boolean Wert „false“

4.3.1 Modell- und Benennungskonventionen des Modellierungsstandards

Der Modellierungsstandard gibt Konventionen für die Benennung von SysML-Sprachelementen der Struktursicht vor [9]. Die Bezeichnungen für Blöcke setzen sich folgendermaßen zusammen:

F_SiE_<Blockname>

Kürzel	Bedeutung
F	Block, der fachliche Anforderungen beschreibt.
SiE	Sicherungsebene des Stellwerks

Für Flow-Ports sind verschiedene Kürzel vorgegeben:

<in|out>_<D|T>_<Portnummer>_<Kd|Md>_<Portname>

Kürzel	Bedeutung
in	Der Flow-Port dient zur Eingabe von Informationen.
out	Der Flow-Port dient zur Ausgabe von Informationen.
D	Flow-Port, an dem Daten kontinuierlich verfügbar sind.
T	Flow-Port, an dem Informationen kurzfristig verfügbar sind. Mit diesem können Ereignisse übermittelt werden.
Kd	An diesem Flow-Port werden Kommandos eingegeben.
Md	An diesem Flow-Port werden Meldungen aufgenommen oder gesendet.

4.3.2 Block Fahrstraßenelement

Das Fahrstraßenelement ist als Block mit der Bezeichnung *F_SiE_Fahrstrassenelement* modelliert und setzt die Anforderung FA02 um. Die Abbildung 4.2 zeigt das interne Blockdiagramm des Fahrstraßenelements.

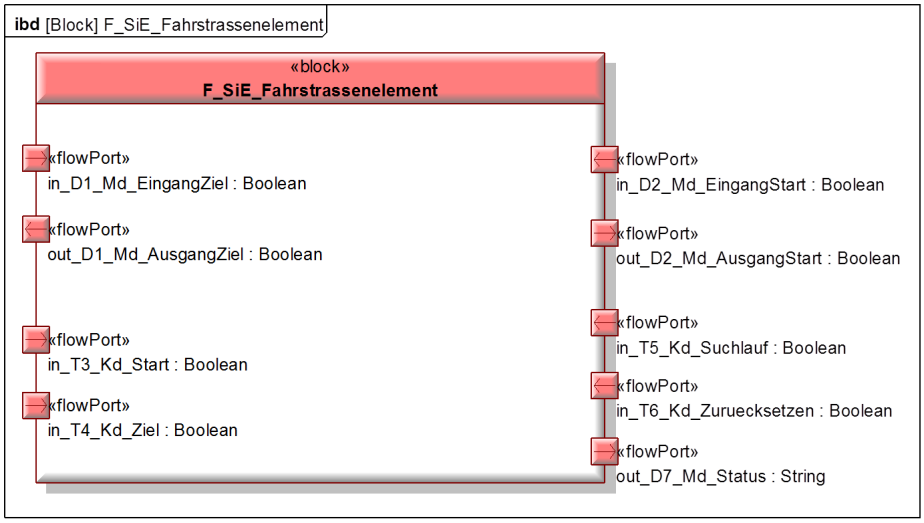

Abbildung 4.2: Internes Blockdiagramm des Blocks F_SiE_Fahrstrassenelement

Tabelle 4.1: Zusammenhang der Anforderungen und der modellierten Flow-Ports des Fahrstraßenelements

Flow-Port	**Anforderung**	**Zweck**
in_D1_Md_EingangZiel	FA04	primär: empfangen von Suchsignalen sekundär: empfangen von Echosignalen entgegen der Geltungsrichtung
out_D1_Md_AusgangZiel	FA04	primär: senden von Echosignalen sekundär: senden von Suchsignalen entgegen der Geltungsrichtung
in_D2_Md_EingangStart	FA04	primär: empfangen von Echosignalen aus Geltungsrichtung sekundär: empfangen von Suchsignalen entgegen der Geltungsrichtung
out_D2_Md_AusgangStart	FA04	primär: senden von Suchsignalen in Geltungsrichtung sekundär: senden von Echosignalen entgegen der Geltungsrichtung
in_T3_Kd_Start	FA06	Eingabe einer Start-Markierung
in_T4_Kd_Ziel	FA07	Eingabe einer Ziel-Markierung
in_T5_Kd_Suchlauf	keine	Starten der Fahrwegsuche
in_T6_Kd_Zuruecksetzen	FA24	Zurücksetzen des Fahrstraßenelements
out_D7_Md_Status	FA15	Ausgabe des Status des Fahrstraßenelements

Tabelle 4.1 zeigt die Bedeutung der einzelnen Flow-Ports und der Zusammenhang mit den Anforderungen des Fahrstraßenelements.
Der Flow-Port „in_T5_Kd_Suchlauf“ entstammt keiner Anforderung, sondern ist ein Detail der SysML-Modellierung. Es wird festgelegt, dass eine Fahrwegsuche explizit über diesen Flow-Port gestartet wird, sobald an diesem der Wert „true“ anliegt.

4.3.3 Block Weichenelement

Der Block *F_SiE_Weichenelement* stellt das Weichenelement nach Anforderung FA03 dar. Das interne Blockdiagramm des Weichenelements zeigt Abbildung 4.3. Das Weichenelement verwendet naheliegenderweise die fachspezifische Terminologie einer Weiche (siehe Kapitel 3).

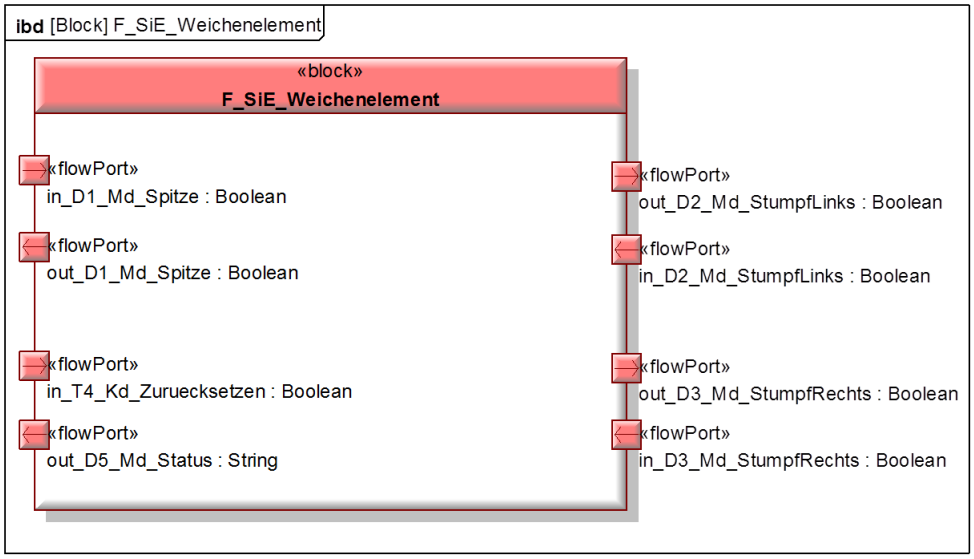

Abbildung 4.3: Internes Blockdiagramm des Blocks F_SiE_Weichenelement

Tabelle 4.2 erläutert den konkreten Zweck der einzelnen Flow-Ports und den Zusammenhang mit den entsprechenden Anforderungen.

4.3.4 Optionaler Block Umfahrstraßenelement

In Kapitel 3 wurde in Zusammenhang mit dem Fahrstraßenbegriff die Umfahrstraße erläutert. Um in einem solchen Szenario vor Beginn der Fahrwegsuche festzulegen, welcher der mehreren möglichen Fahrwege in Betracht kommt, wird ein optionales *Umfahrstraßenelement* benötigt. Ausgangspunkt dafür sind die Anforderungen FA25 bis FA28. Die Funktionalität, die in diesen Anforderungen beschrieben ist, wurde angelehnt an [12]. Prinzipiell handelt es sich dabei um einen Schalter (ähnlich einem elektrischen Schalter), der Such- und Echosignale weiterleiten oder blockieren kann. Je nach gegebenen Gleisplan muss es nicht eingesetzt werden, weshalb es als optionale Funktionalität kategorisiert

Tabelle 4.2: Zusammenhang der Anforderungen und der modellierten Flow-Ports des Weichenelements

Flow-Port	**Anforderung**	**Zweck**
in_D1_Md_Spitze	FA04	empfangen von Such- und Echosignalen an der Weichenspitze
out_D1_Md_Spitze	FA04	senden von Such- und Echosignalen an der Weichenspitze
in_D2_Md_StumpfLinks	FA04	empfangen von Such- und Echosignalen am linken Strang des Weichenstumpfs
out_D2_Md_StumpfLinks	FA04	senden von Such- und Echosignalen am linken Strang des Weichenstumpfs
in_D3_Md_StumpfRechts	FA04	empfangen von Such- und Echosignalen am rechten Strang des Weichenstumpfs
out_D3_Md_StumpfRechts	FA04	senden von Such- und Echosignalen am rechten Strang des Weichenstumpfs
in_T4_Kd_Zuruecksetzen	FA24	Zurücksetzen des Weichenelements
out_D5_Md_Status	FA22	Ausgabe des Status

ist. Es wird im Elementverbindungsplan zwischen die Fahrstraßen- und Weichenelemente eingesetzt (Abbildung 4.5), um die Such- und Echosignale gezielt über die Regelfahrstraße oder über eine Umfahrstraße zu lenken. Das Umfahrstraßenelement ist als Block mit der Bezeichnung *F_SiE_Umfahrstrassenelement* modelliert und ist in Abbildung 4.4 dargestellt. Die Bedeutung der einzelnen Flow-Ports zeigt Tabelle 4.3.

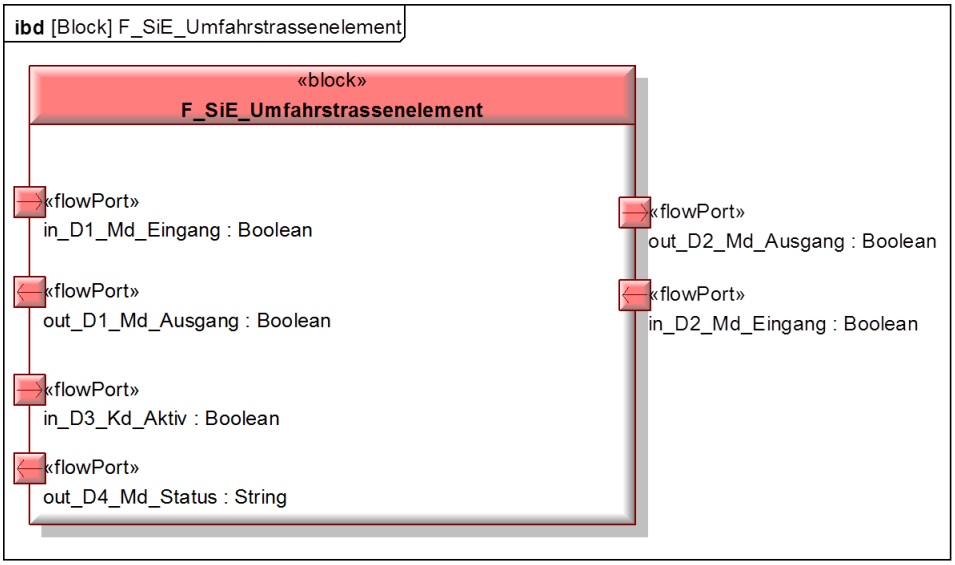

Abbildung 4.4: Internes Blockdiagramm des Blocks F_SiE_Umfahrstrassenelement

Tabelle 4.3: Zusammenhang der Anforderungen und der modellierten Flow-Ports des Umfahrstraßenelements

Flow-Port	**Anforderung**	**Zweck**
in_D1_Md_Eingang	FA25	empfangen von Such- und Echosignalen
out_D1_Md_Ausgang	FA25	senden von Such- und Echosignalen
in_D2_Md_Eingang	FA25	empfangen von Such- und Echosignalen
out_D2_Md_Ausgang	FA25	senden von Such- und Echosignalen
in_D3_Kd_Aktiv	FA27	Eingabe zum Schalten
out_D4_Md_Status	FA28	Ausgabe des Status

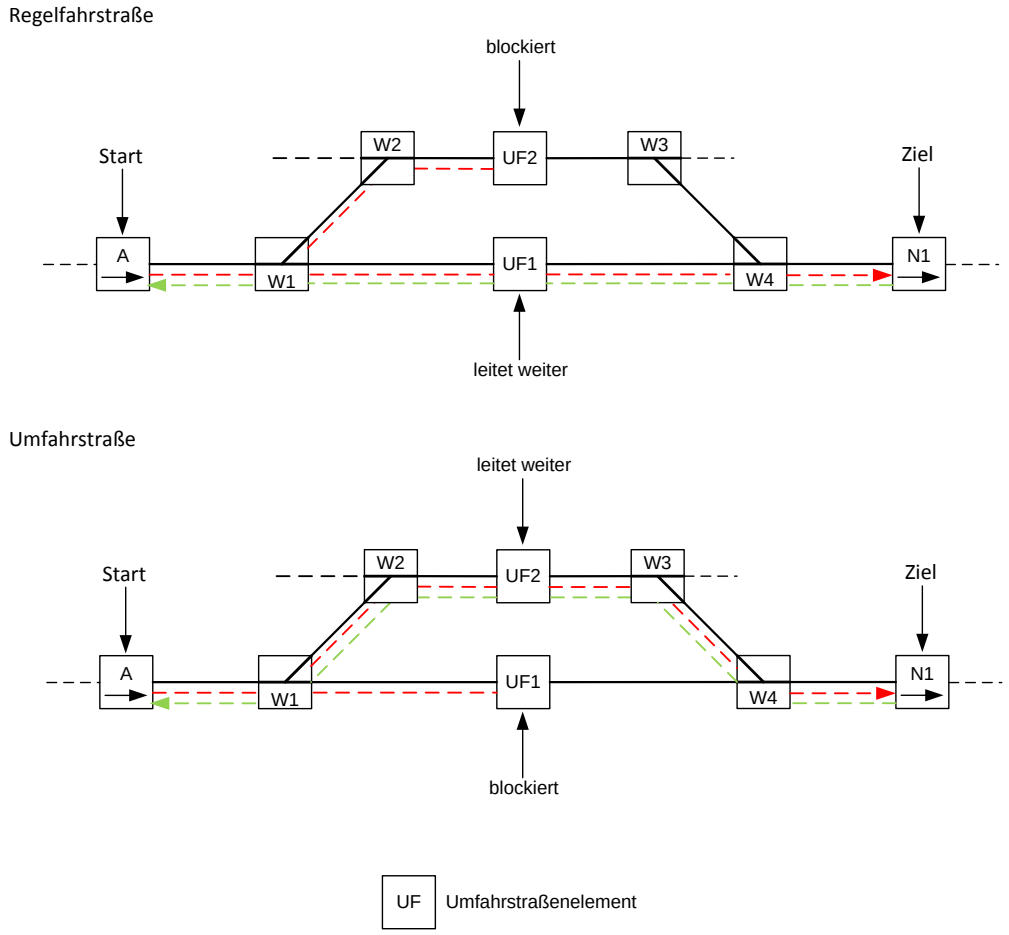

Abbildung 4.5: Demonstration der Wirkung des Umfahrstraßenelements in einem Elementverbindungsplan.

4.4 SysML-Verhaltenssicht

Mit Hilfe der Verhaltenssicht wird das Verhalten der in der Struktursicht definierten Blöcke beschrieben, wobei jeder Block eine eigene Verhaltensbeschreibung erhält. Maßgeblich dafür sind die Anforderungen FA09 bis FA15 (Fahrstraßenelement) und FA16 bis FA22 (Weichenelement). Wesentlicher Aspekt bei der Verhaltensmodellierung ist dabei das Senden und Empfangen von Such- bzw. Echosignalen mittels der Änderung der Werte an den Flow-Ports und das Reagieren auf Änderungen der anliegenden Werte.

Eine naheliegende Möglichkeit der Verhaltensbeschreibung ist die Verwendung von SysML-Zustandsautomaten (*state machines*). Zur Begründung ist es auch hier von Vorteil, die Implementierung im Relaisstellwerk aufzugreifen. Eine zustandsorientierte Beschreibung ist deshalb zielführend, da Relaisgruppen ebenfalls als zustandsorientierte Komponenten klassifizierbar sind. Der Gesamtzustand einer Relaisgruppe ergibt sich als eine Kombination aus den Zuständen der in der Relaisgruppe enthalten Relais. Die Relais, die in der Eisenbahnsicherungstechnik verwendet werden, haben die zwei Zustände „abgefallen“ und „angezogen“ [15]. Der Modellierungsstandard legt ebenfalls die Verwendung von SysML-Zustandsautomaten fest [9]. Daher werden nachfolgend Zustandsautomaten eingesetzt, um die Verhaltenssicht zu beschreiben. Folgende Sprachelemente der SysML für die Verhaltensmodellierung werden in dieser Arbeit berücksichtigt und angewendet [16]:

- initialer Pseudozustand (*inital pseudo state*)
- Zustand (*simple state*)
- Zustand mit Eintrittsaktion / Eintrittsverhalten (*entry activity / entry behaviour*)
- zusammengesetzter Zustand (*composite state*)
- Transition / Zustandsübergang (*transition*)
- Zustandsdiagramm (*state machine diagram*)

Die Transitionen selbst können eine Verhaltensbeschreibung folgender Form nach [16] besitzen:

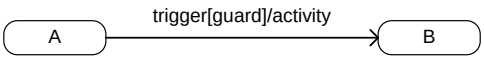

Abbildung 4.6: Transition vom einen Zustand A in einen Zustand B, welche eine Verhaltensbeschreibung enthält.

Die Syntax und Semantik entspricht der SysML- und der UML-Spezifikation. In dieser Arbeit werden ausschließlich Trigger verwendet, die auf Ereignisse des Typs „Change-Event“ nach [17] reagieren. Die Notation dafür lautet:

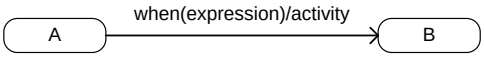

Abbildung 4.7: Verhaltensbeschreibung ohne Guard.

Nachfolgend werden die Trigger, die auf „ChangeEvents“ reagieren als Change-Event-Trigger bezeichnet. Über einen solchen Change-Event-Trigger kann auf die Änderung des Werts an einem Flow-Port reagiert werden. Wenn die Bedingung im Booleschen Ausdruck „*expression*“ von falsch auf wahr wechselt, wird die zugehörige Transition ausgelöst [17]. Der Bestandteil „*guard*“ als Boolesche Bedingung ist optional und wird in dieser Arbeit nicht angewendet. Der Ausdruck „*activity*“ ist ebenfalls optional und kann aus einer Sequenz aufzurufender Aktionen bestehen [17], in dem beispielsweise Operationen oder Anweisungen aufgeführt sind.

Als Aktionssprache für die Change-Event-Trigger kommt die *Atego Structured Action Language* (ASAL) zum Einsatz, die Bestandteil von Artisan Studio ist und die spätere Modellsimulation (Kapitel 5) ermöglicht. Tabelle 4.4 gibt einen Überblick für die Syntax und Semantik der verwendeten Anweisungen der ASAL.

Tabelle 4.4: Verwendete Sprachelemente der Atego Structured Action Language

Ausdruck	**Bedeutung**
port := true;	Weist dem Flow-Port „port“ von Typ Boolean den Wert „true“ zu.
port := false;	Weist dem Flow-Port „port“ von Typ Boolean den Wert „false“ zu.
when(port = true)	Change-Event-Trigger einer Transition. Bei einer Änderung des Wertes am Flow-Port „port“ wird geprüft, ob an diesem der Wert „true“ anliegt.
when(port = false)	Change-Event-Trigger einer Transition. Bei einer Änderung des Wertes am Flow-Port „port“ wird geprüft, ob an diesem der den Wert „false“ anliegt.
when(inport = true)/outport = false;	Change-Event-Trigger einer Transition. Bei einer Änderung des Wertes am Flow-Port „inport“ wird geprüft, ob an diesem der Wert „true“ anliegt. Falls ja, wird „outport“ der Wert „false“ zugewiesen.
port := "A";	Weist dem Flow-Port „port“ von Typ String den Wert „A“ zu.
op();	Ruft die Operation „op“ auf.

4.4.1 Zustandsautomat Fahrstraßenelement

Der Zustandsautomat des Fahrstraßenelements besteht im Wesentlichen aus dem zusammengesetzten Zustand „*Aktiv*“, welcher aus Unterzuständen besteht. Abbildung 4.8 zeigt den vollständigen Zustandsautomaten in einem Zustandsdiagramm. Der Zustand „*Aktiv*“ dient dem Zweck, in jedem Unterzustand das Fahrstraßenelement zurücksetzen zu können kann. Die Bedeutung der einzelnen Unterzustände ist in Tabelle 4.5 dargestellt.

Tabelle 4.5: Zustände im Zustandsautomaten des Fahrstraßenelements

Zustand	**Bedeutung**
NichtMarkiert	Das Fahrstraßenelement hat in diesem Zustand kein Suchsignal und kein Echosignal gesendet oder empfangen und ist derzeit an keiner Fahrwegsuche beteiligt.
AlsStartMarkiert	Das Fahrstraßenelement hat eine Start-Markierung erhalten.
AlsZielMarkiert	Das Fahrstraßenelement hat eine Ziel-Markierung erhalten.
SuchsignalWeitergeleitet	Ein Suchsignal entgegen der Geltungsrichtung wurde empfangen und weitergeleitet.
SuchsignalGesendet	Das Fahrstraßenelement mit einer Start-Markierung hat in Geltungsrichtung ein Suchsignal gesendet.
EchosignalGesendet	Das Fahrstraßenelement mit einer Ziel-Markierung hat ein Echosignal gesendet.
EchosignalWeitergeleitet	Ein Echosignal wurde, passend zum entgegen der Geltungsrichtung empfangenden Suchsignal, weitergeleitet.
EchosignalEmpfangen	Das Fahrstraßenelement mit einer Start-Markierung hat nach dem Senden des Suchsignals das entsprechende Echosignal aus der Geltungsrichtung empfangen.

Der Zustand „NichtMarkiert“ hat ein Eintrittsverhalten in Form eines Operationsaufrufs. Es wird dabei die Operation „init“ aufgerufen, die bei jedem Betreten dieses Zustands alle Flow-Ports für die Ausgabe nach außen zurücksetzt, also den Flow-Ports den Wert „false“ übergibt. Es wird dabei die Anforderung FA24 erfüllt.

Listing 4.1: init-Operation Fahrstraßenelement

```
out_D1_Md_AusgangZiel := false;
out_D2_Md_AusgangStart := false;
out_D7_Md_Status := "";
```

Tabelle 4.6 stellt den genauen Zusammenhang zwischen Anforderungen und Transitionen her. Das in den Anforderungen FA09 bis FA15 beschriebene Verhalten wird durch die Verhaltensbeschreibung der Transitionen realisiert.

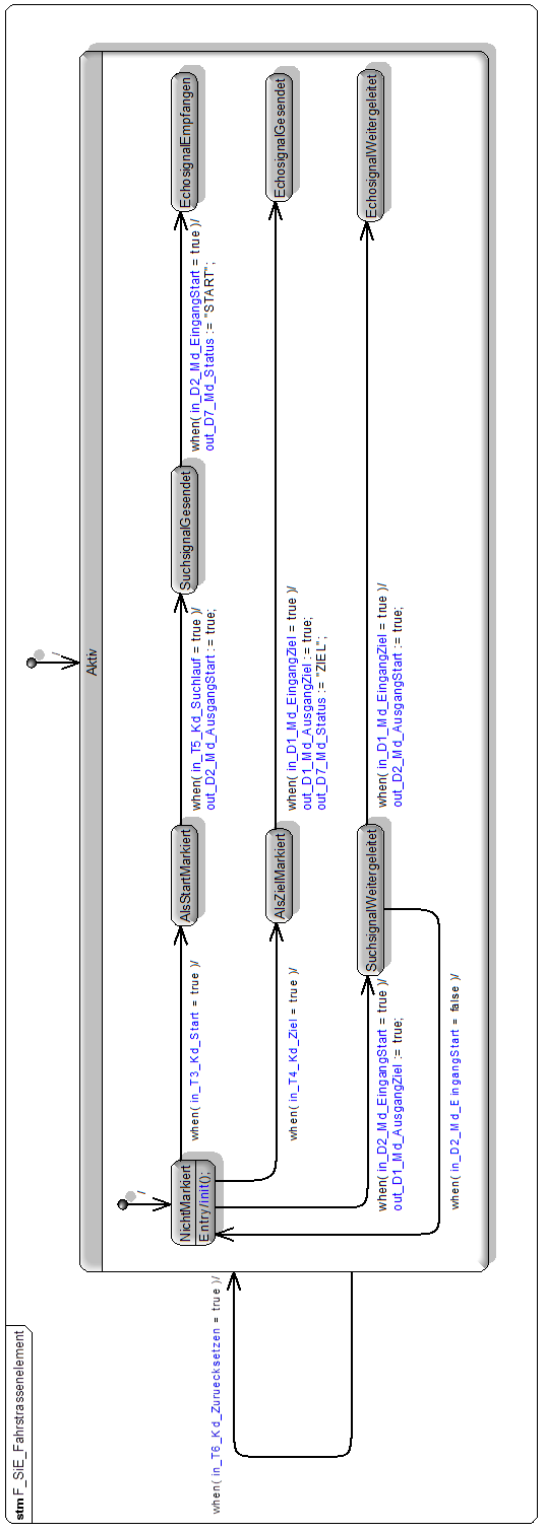

Abbildung 4.8: Zustandsautomat des Blocks F_SiE_Fahrstrassenelement

Tabelle 4.6: Transitionen im Zustandsautomaten des Fahrstraßenelements

Anforderung	**Transition**
FA06	NichtMarkiert⟶ AlsStartMarkiert
FA07	NichtMarkiert⟶ AlsZielMarkiert
FA23	NichtMarkiert⟶ SuchsignalWeitergeleitet
FA10, FA11	AlsStartMarkiert⟶ SuchsignalGesendet
FA15	SuchsignalGesendet⟶ EchosignalEmpfangen
FA12, FA13, FA14	AlsZielMarkiert⟶ EchosignalGesendet
FA23	SuchsignalWeitergeleitet⟶ EchosignalWeitergeleitet
FA24	Aktiv⟶ Aktiv
keine	SuchsignalWeitergeleitet⟶ NichtMarkiert

Die Transition „SuchsignalWeitergeleitet⟶ NichtMarkiert“ wird betreten, wenn ein Suchsignal am betreffenden Fahrstraßenelement zurückgenommen wird, bzw. sich der Wert am Flow-Port „in_D2_Md_EingangStart“ wieder von „true“ auf „false“ ändert. Dies tritt in den Fällen ein, wenn ein benachbartes Weichenelement ein Suchsignal auf dem Strang, auf dem es kein Echosignal erhalten hat, zurückgenommen hat. Genauere Ausführungen zu diesem Sachverhalt finden sich im Abschnitt über den Zustandsautomaten des Weichenelements.

4.4.2 Zustandsautomat Weichenelement

Analog zum Zustandsautomaten des Fahrstraßenelements besteht der Zustandsautomat des Weichenelements ebenfalls aus dem zusammengesetzten Zustand „*Aktiv*“ mit entsprechenden Unterzuständen. In Abbildung 4.9 ist der vollständige Zustandsautomat dargestellt.

Die Bedeutung der einzelnen Unterzustände des Zustandsautomaten des Weichenelements ist in Tabelle 4.7 erläutert. Das in den Anforderungen FA16 bis FA22 definierte Verhalten wird durch die Verhaltensbeschreibung der Transitionen realisiert. Tabelle 4.8 stellt den genauen Zusammenhang zwischen Anforderungen und Transitionen dar.

Das Weichenelement hat im Zustand „NichtMarkiert“ ein Eintrittsverhalten in Form eines Aufrufs der Operation „init“, die bei jeden Betreten des Zustands alle Flow-Ports für die Ausgabe nach außen zurücksetzt. Erfüllt wird die Anforderung FA24.

Listing 4.2: init-Operation Weichenelement

```
out_D1_Md_Spitze := false;
out_D2_Md_StumpfLinks := false;
out_D3_Md_StumpfRechts := false;
out_D5_Md_Status := "";
```

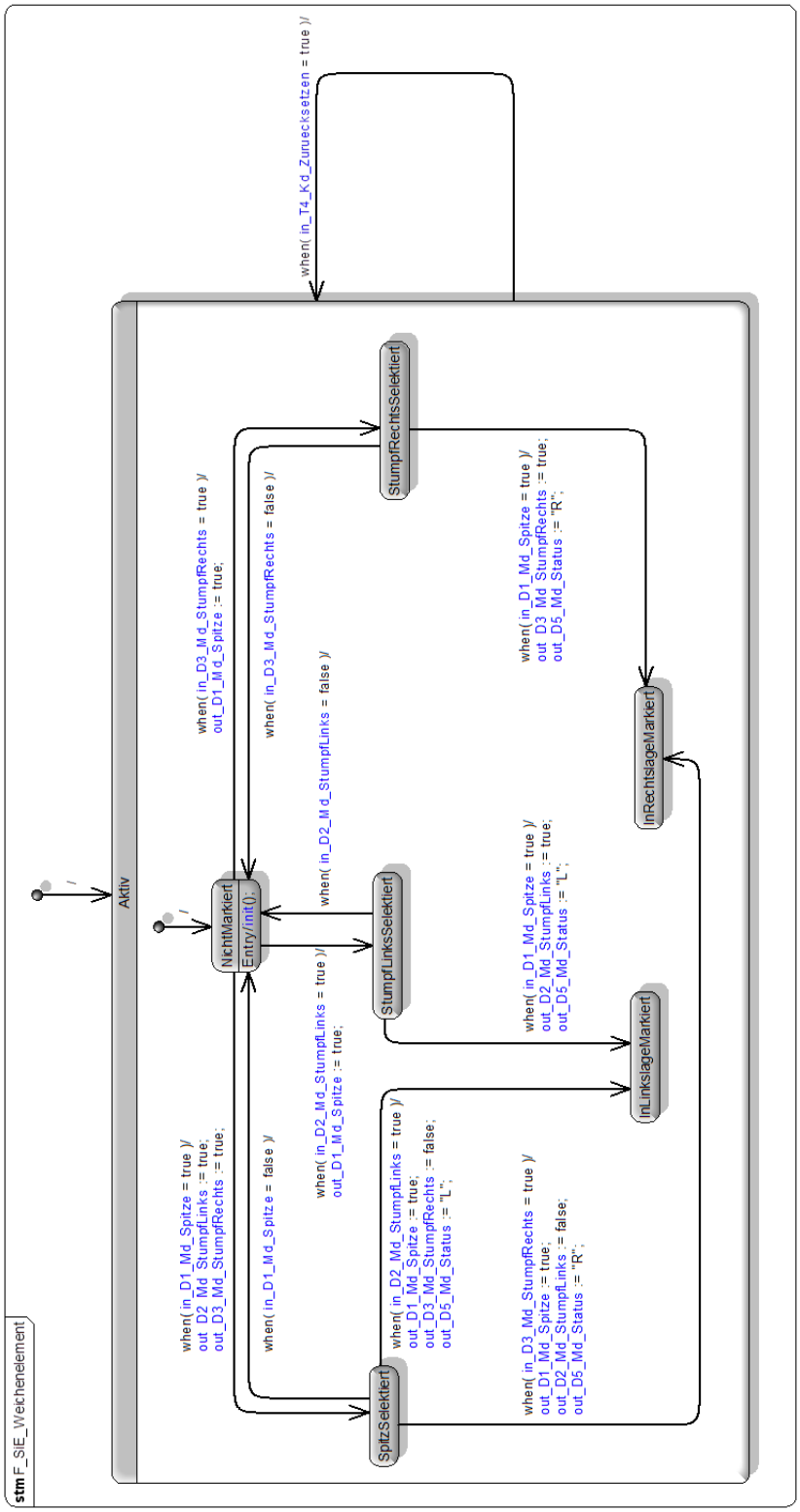

Abbildung 4.9: Zustandsautomat des Blocks F_SiE_Weichenelement

Tabelle 4.7: Zustände im Zustandsautomaten des Weichenelements

Zustand	**Bedeutung**
NichtMarkiert	Es wurde in diesem Zustand kein Suchsignal und kein Echosignal gesendet und empfangen. Das Weichenelement ist derzeit an keiner Fahrwegsuche beteiligt.
SpitzSelektiert	Das Weichenelement hat an der Weichenspitze ein Suchsignal empfangen und es in den linken und rechten Strang weitergeleitet.
StumpfLinksSelektiert	Das Weichenelement hat am Weichenstumpf (linker Strang) ein Suchsignal empfangen und es über die Weichenspitze weitergeleitet.
StumpfRechtsSelektiert	Das Weichenelement hat am Weichenstumpf (rechter Strang) ein Suchsignal empfangen und es über die Weichenspitze weitergeleitet.
InLinkslageMarkiert	Das Weichenelement hat ein Echosignal empfangen und ist in Linkslage festgelegt.
InRechtslageMarkiert	Das Weichenelement hat ein Echosignal empfangen und ist in Rechtslage festgelegt.

Tabelle 4.8: Transitionen im Zustandsautomaten des Weichenelements

Anforderung	**Transition**
FA17	NichtMarkiert $\longrightarrow$ SpitzSelektiert
FA18	NichtMarkiert $\longrightarrow$ StumpfLinksSelektiert NichtMarkiert $\longrightarrow$ StumpfRechtsSelektiert
FA19, FA22	StumpfLinksSelektiert $\longrightarrow$ InLinkslageMarkiert StumpfRechtsSelektiert $\longrightarrow$ InRechtslageMarkiert
FA20, FA22	SpitzSelektiert $\longrightarrow$ InLinkslageMarkiert
FA21, FA22	SpitzSelektiert $\longrightarrow$ InRechtslageMarkiert
FA24	Aktiv $\longrightarrow$ Aktiv
keine	SpitzSelektiert $\longrightarrow$ NichtMarkiert
keine	StumpfLinksSelektiert $\longrightarrow$ NichtMarkiert
keine	StumpfRechtsSelektiert $\longrightarrow$ NichtMarkiert

Ein Weichenelement, welches von einem Suchsignal an der Spitze selektiert wurde und ein Echosignal am linken oder rechten Strang empfängt, nimmt das Suchsignal des Strangs zurück, aus dem kein Echosignal kam. Dies wird durch die Transitionen „SpitzSelektiert $\longrightarrow$ InLinkslageMarkiert“ und „SpitzSelektiert $\longrightarrow$ InRechtslageMarkiert“ realisiert. Bei der Weichengruppe des SpDrS600-Stellwerks bedeutet dies, das der Suchstrom in den nicht mehr benötigten Strang abgetrennt wird [22]. In dem Strang des Weichenelements, wo das Suchsignal zurückgenommen wurde, also der Wert am entsprechenden Flow-Port auf „false“ gesetzt wurde, sollten daher alle benachbarten Fahrstraßen- und Weichenelemente automatisch zurückgesetzt werden, da diese nicht mehr für die Fahrwegsuche benötigt werden. Damit ein anderes, benachbartes Weichenelement auf das Zurücknehmen des Suchsignals reagieren und in den Unterzustand „NichtMarkiert“ zurückgehen kann, sind folgende zusätzliche Transitionen angelegt:

Anforderung	**Transition**
keine	SpitzSelektiert $\longrightarrow$ NichtMarkiert
keine	StumpfLinksSelektiert $\longrightarrow$ NichtMarkiert
keine	StumpfRechtsSelektiert $\longrightarrow$ NichtMarkiert

Dadurch kann die durch ein Suchsignal bewirkte Selektion eines Weichenelements wieder zurückgenommen werden, sobald sich der Wert am Flow-Port, an dem das Suchsignal empfangen wurde, auf „false“ ändert. Zum Verständnis dieses Aspekts ist eine Skizze in Abbildung 4.10 gegeben.

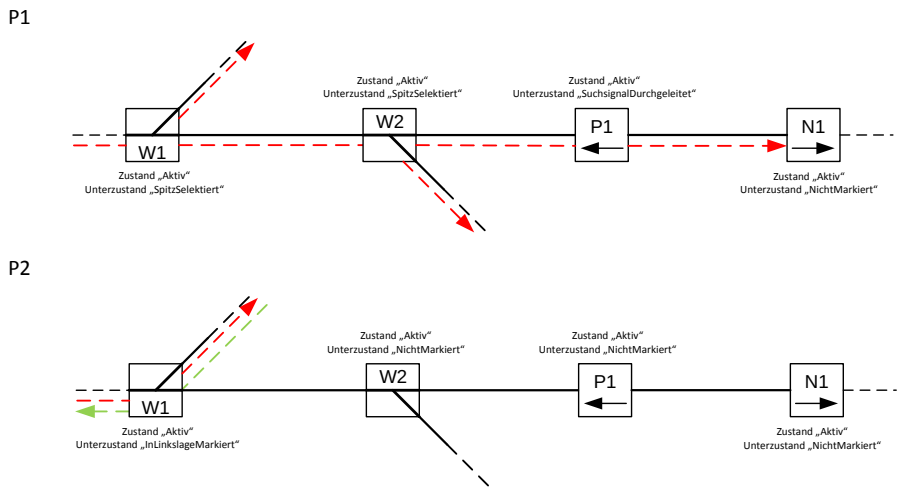

Abbildung 4.10: Zurücknahme des Suchsignals am rechten Strang von W1

In Phase P1 ist neben dem linken auch in den rechten Strang des Weichenelements W1 ein Suchsignal gesendet worden. Da das Fahrstraßenelement N1 aber nicht als Ziel markiert ist, sendet es kein Echosignal. In Phase P2 tritt ein Echosignal am linken Strang von W1 auf. Aufgrund der ausgelösten Transition „SpitzSelektiert $\longrightarrow$ InLinkslageMar-

kiert“ in W1 wird das Echosignal in den rechten Strang zurückgenommen. Es entsteht eine Art Kettenreaktion, bei der Weichenelement W2 in den Zustand „NichtMarkiert“ übergeht und selbst wiederum die verzweigten Suchsignale zurücknimmt. Das Fahrstraßenelement P1 geht ebenfalls vom Zustand „SuchsignalWeitergeleitet“ in den Zustand „NichtMarkiert“ über.

4.4.3 Optionaler Zustandsautomat Umfahrstraßenelement

Das Verhalten des Umfahrstraßenelements wird in dessen Zustandsautomaten durch zwei wesentliche Zustände „*NichtAktiv*“ und „*Aktiv*“ beschrieben. Dabei ist „*Aktiv*“ ähnlich wie beim Fahrstraßen- und Weichenelement ein zusammengesetzter Zustand, in dem Such- und Echosignale weitergeleitet werden. Abbildung 4.11 illustriert den Zustandsautomaten. Tabelle 4.9 zeigt die Bedeutung der einzelnen Zustände.

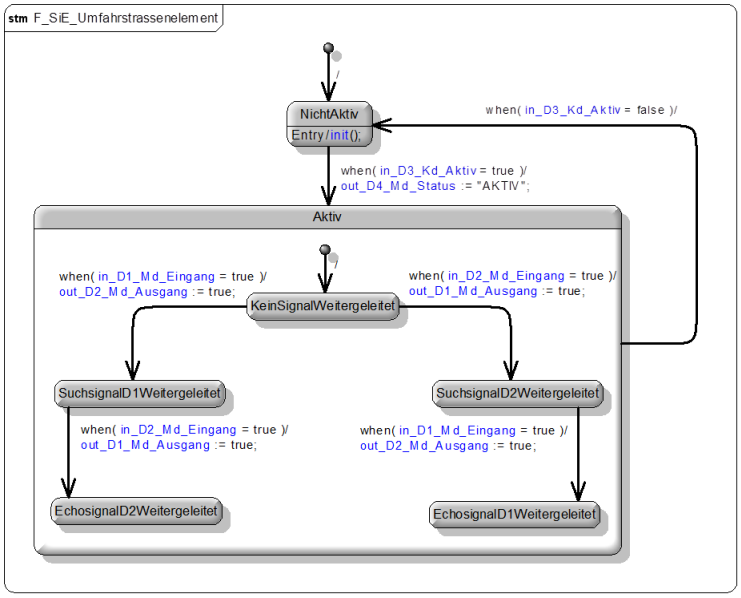

Abbildung 4.11: Zustandsautomat des Blocks F_SiE_Umfahrstrassenelement

Die meisten Transitionen des Umfahrstraßenelements ergeben sich aus der Modellierung und haben keine sprachlichen Anforderungen als Ausgangspunkt (Tabelle 4.10). Das Umfahrstraßenelement hat im Zustand „NichtAktiv“ ein Eintrittsverhalten in Form eines Aufrufs der Operation „init“, welches alle Flow-Ports für die Ausgabe nach außen auf „false“ setzt.

Listing 4.3: init-Operation Umfahrstraßenelement

```
out_D1_Md_Ausgang := false;
out_D2_Md_Ausgang := false;
out_D4_Md_Status := "INAKTIV";
```

Tabelle 4.9: Zustände im Zustandsautomaten des Umfahrstraßenelements

Zustand	**Bedeutung**
NichtAktiv	Es werden keine Such- und Echosignale weitergeleitet.
KeinSignalWeitergeleitet	Es wurde kein Suchsignal zum Weiterleiten empfangen.
SuchsignalD1Weitergeleitet	Es wurde ein Suchsignal empfangen und weitergeleitet.
EchosignalD2Weitergeleitet	Es wurde ein Echosignal empfangen und weitergeleitet.
SuchsignalD2Weitergeleitet	Es wurde ein Suchsignal empfangen und weitergeleitet.
EchosignalD1Weitergeleitet	Es wurde ein Echosignal empfangen und weitergeleitet.

Tabelle 4.10: Transitionen im Zustandsautomaten des Umfahrstraßenelements

Anforderung	**Transition**
keine	KeinSignalWeitergeleitet $\longrightarrow$ SuchsignalD1Weitergeleitet
keine	KeinSignalWeitergeleitet $\longrightarrow$ SuchsignalD2Weitergeleitet
keine	SuchsignalD1Weitergeleitet $\longrightarrow$ EchosignalD2Weitergeleitet
keine	SuchsignalD2Weitergeleitet $\longrightarrow$ EchosignalD1Weitergeleitet
FA27	NichtAktiv $\longrightarrow$ Aktiv Aktiv $\longrightarrow$ NichtAktiv
FA28	NichtAktiv $\longrightarrow$ Aktiv

4.5 Konstruktion eines Elementverbindungplans

Mit dem modellierten Fahrstraßen- und Weichenelement kann nun ein Elementverbindungsplan in SysML zusammengesetzt werden. Dabei geht es an dieser Stelle um das generelle Prinzip der Verbindung der Blöcke untereinander über Konnektoren, da dies zur SysML-Struktursicht gehört. Repräsentativ dafür soll das in Abbildung 4.12 dargestellte Szenario mit der Bezeichnung „Testbahnhof1“ in ein SysML-Modell überführt werden. Dies ist ein beispielhafter Gleisplanausschnitt, erfüllt aber den Zweck zur Demonstration der SysML-Darstellung eines Elementverbindungsplans. Ein konkreter und simulierbarer Elementverbindungsplan wird in Kapitel 5 erstellt.

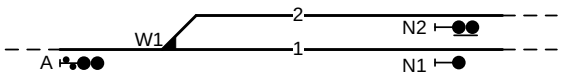

Abbildung 4.12: Lageplanausschnitt Testbahnhof1

Der zum Lageplan zugehörige, allgemeine Elementverbindungsplan ist in Abbildung 4.13 dargestellt.

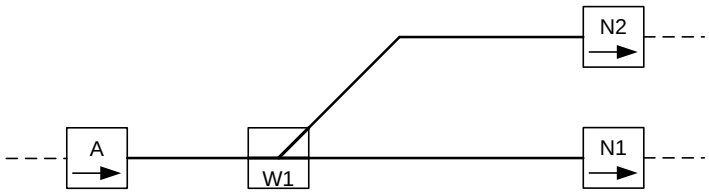

Abbildung 4.13: Elementverbindungsplan Testbahnhof1

Für den SysML-Block „Testbahnhof1“, welcher den oben gezeigten Gleisplanauschnitt und dessen Elementverbindungsplan darstellt, wird ein internes Blockdiagramm angelegt. Innerhalb des Diagramms wird die interne Struktur des Blocks festgelegt, welche aus einer Anordnung von Fahrstraßen- und Weichenelementen besteht. Die Elemente werden mit Hilfe von Konnektoren genau so miteinander verbunden, wie es dem Gleisplanausschnitt entspricht. Das Ergebnis zeigt Abbildung 4.14.

4.5.1 Ablauf einer Fahrwegsuche in SysML

Anhand der Fahrstraße von Fahrstraßenelement A nach Fahrstraßenelement N1 wird nachfolgend der Ablauf einer Fahrwegsuche geschildert. Dabei werden Schritt für Schritt die Änderungen der Werte an den Flow-Ports aufgeführt. Zu Beginn wird für die Fahrstraße A $\longrightarrow$ N1 das Fahrstraßenelement A als Start und Fahrstraßenelement N1 als Ziel markiert. Es wird vorausgesetzt, dass alle Fahrstraßen- und Weichenelemente zurückgesetzt sind. Zu Beginn wird das Start- und Zielelement festgelegt:

- Dem Flow-Port „in_T3_Kd_Start“ des Fahrstraßenelements A wird der Wert „true“ übergeben. Fahrstraßenelement A wechselt vom Zustand „NichtMarkiert“ in den Zustand „AlsStartMarkiert“.
- Dem Flow-Port „in_T4_Kd_Ziel“ des Fahrstraßenelements N1 wird der Wert „true“ übergeben. Fahrstraßenelement N1 wechselt vom Zustand „NichtMarkiert“ in den Zustand „AlsZielMarkiert“.

Die Fahrwegsuche wird durch Übergabe des Wertes „true“ an den Flow-Port „in_T5_Kd_Suchlauf“ vom Fahrstraßenelement A gestartet. Daraus ergibt sich folgendes Ablaufschema:

1. Fahrstraßenelement A wechselt vom Zustand „AlsStartMarkiert“ in den Zustand „SuchsignalGesendet“ und übergibt dem Flow-Port „out_D2_Md_AusgangStart“ den Wert „true“.
2. Weichenelement W1 wechselt vom Zustand „NichtMarkiert“ in den Zustand „SpitzSelektiert“. Den Flow-Ports „out_D2_Md_StumpfLinks“ und „out_D3_Md_StumpfRechts“ wird der Wert „true“ übergeben.

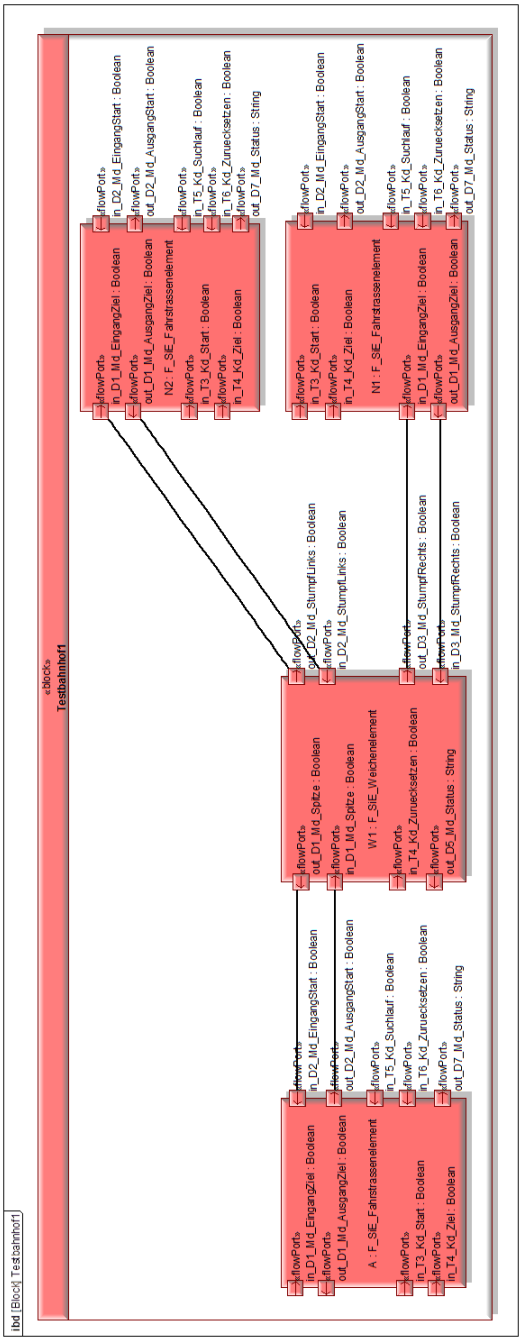

Abbildung 4.14: SysML-Elementverbindungsplan Testbahnhof1

3. Fahrstraßenelement N1 wechselt vom Zustand „AlsZielMarkiert“ in den Zustand „EchosignalGesendet“. Dem Flow-Port „out_D1_Md_AusgangZiel“ wird der Wert „true“ übergeben. Der Flow-Port „out_D7_Md_Status“ erhält den Wert „ZIEL“. Fahrstraßenelement N2 verbleibt im Zustand „NichtMarkiert“.

4. Weichenelement W1 wechselt vom Zustand „SpitzSelektiert“ in den Zustand „InRechtslageMarkiert“. Dem Flow-Port „out_D1_Md_Spitze“ wird der Wert „true“ übergeben. Dem Flow-Port „out_D2_Md_StumpfLinks“ wird der Wert „false“ übergeben und „out_D5_Md_Status“ der Wert „R“.

5. Fahrstraßenelement A wechselt in den Zustand „EchosignalEmpfangen“ und übergibt dem Flow-Port „out_D7_Md_Status“den Wert „START“.

Die Modellierung der Fahrwegsuche ist damit abgeschlossen. Im nächsten Kapitel werden nun die erstellten Blöcke und Zustandsautomaten validiert und überprüft, ob diese die sprachlichen Anforderungen erfüllen.

5 Test des Modells durch Validierung

In diesem Kapitel erfolgt durch das Testen des erstellten SysML-Modells der Fahrwegsuche mittels Validierung, ob es die sprachlichen Anforderungen einhält. Dazu sind aus den Anforderungen zusätzlich Testfälle abgeleitet worden, mit denen die Ein- und Ausgaben an den Flow-Ports der einzelnen Blöcke des Fahrstraßen-, Weichen- und Umfahrstraßenelements überprüft werden. Dabei wird jeder Block zuerst einzeln getestet und anschließend im Verbund mit einem exemplarischen Elementverbindungsplan. Die Validierung bedeutet im Kontext dieser Arbeit, dass untersucht wird, ob die erstellte modellbasierte Anforderungsspezifikation die sprachlich formulierten Anforderungen erfüllt.

Möglich werden die konkreten Tests durch eine Modellsimulation mit dem Zusatzprogramm *SySim* von Artisan Studio, dass die erstellte Struktur- und Verhaltenssicht des Modells in ein ausführbares Programm übersetzt. Für jeden Block wird dabei ein Simulationsszenario in Form eines internen Blockdiagramms erstellt. Anschließend wird das Simulationsszenario durch SySim in Visual Basic-Quellcode überführt. Letzterer kann schließlich kompiliert und ausgeführt werden. Dabei wurde die Entwicklungsumgebung *Visual Basic 2008 Express* verwendet.

Ein Ziel der Validierung ist außerdem möglichst jede Transition eines Zustandsautomaten mindestens einmal auszulösen. Um dies zu prüfen, wird eine spezielle Funktion von Artisan Studio genutzt, die während der Ausführung einer Modellsimulation anzeigt, in welchem Zustand sich ein Zustandsautomat gerade befindet. Diese Funktion protokolliert außerdem jede ausgelöste Transitionen im Zustandsautomaten eines Blocks und gibt diese in Textform in ein Ausgabefenster aus. Dadurch kann nachvollzogen werden, welche und wie viele Transitionen ausgelöst und von den Testfällen abgedeckt wurden.

5.1 Ableitung von Testfällen aus den sprachlichen Anforderungen

Alle Testfälle sind im Anhang ausführlich hinterlegt und mit eindeutigen Bezeichnern definiert. Sie enthalten eine Eingabe und eine erwartete Ausgabe über die Flow-Ports des jeweiligen, zu testenden Blocks. In Tabelle 5.1 sind die Zusammenhänge zwischen Anforderungen und Testfälle für jedes modellierte Element dargestellt. Dabei gibt die Spalte „Voraussetzung“ für jeden Testfall an, wann dieser angewendet werden kann. Einige Testfälle können nur dann angewendet werden, wenn am Block vorher bereits die Eingaben des als Voraussetzung angegebenen Testfalls gemacht wurden.

Tabelle 5.1: Übersicht über den Zusammenhang zwischen Anforderungen und Testfälle für das Fahrstraßen-, Weichen- und Umfahrstraßenelement

Anforderung	**aus Anforderung abgeleiteter Testfall**	**Voraussetzung**
FA06, FA10, FA11	TF01	keine
FA15	TF02	TF01
FA07	TF03	keine
FA12, FA13, FA14, FA15	TF04	TF03
FA23	TF05	keine
FA23	TF06	TF05
FA24	TF07	keine
keine	TF08	TF05
FA17	TF09	keine
FA20, FA22	TF10	TF09
FA21, FA22	TF11	TF09
FA18	TF12	keine
FA19, FA22	TF13	TF12
FA18	TF14	keine
FA19, FA22	TF15	TF14
FA24	TF16	keine
keine	TF17	TF09
keine	TF18	TF12
keine	TF19	TF14
FA26, FA27, FA28	TF20	keine
FA26, FA27	TF21	TF20
FA26, FA27	TF22	TF21, TF20
FA26, FA27	TF23	keine
FA26, FA27	TF24	TF23, TF20
FA26, FA27, FA28	TF25	keine
FA26, FA27	TF26	TF25
FA26, FA27	TF27	TF25

5.2 Erstellung von ausführbaren Modellen mit Simulationsszenarien

Für das Fahrstraßen-, Weichen- und Umfahrstraßenelement wird zunächst jeweils ein einzelnes Simulationsszenario angelegt, um jeden Block unabhängig von den anderen prüfen zu können. Weil dabei noch kein Elementverbindungsplan in SysML zusammengestellt wird, werden die Eingaben und Ausgaben an den Flow-Ports durch geeignete Steuerelemente vorgenommen, die das Programm SySim bereitstellt. Jedes Steuerelement ist dabei selbst ein Block mit Flow-Ports, der entweder Ausgaben aufnehmen und anzeigen oder Eingaben an einen Block übergeben kann. Die Flow-Ports der SySim-Steuerelemente werden dann mit den Flow-Ports des zu testenden Blocks verbunden. Nach der Übersetzung des Simulationsszenarios stehen die Steuerelemente in der Entwicklungsumgebung von Visual Basic als vollwertige Steuerelemente für eine Benutzeroberfläche zur Verfügung, mit denen dann die Simulation gesteuert werden kann. Die folgende Tabelle gibt eine Übersicht über die in dieser Arbeit eingesetzten Steuerelemente.

SySim-Steuerelement	**Darstellung**	**Zweck**
Switch	aus (false)/ an (true)	Dieses Steuerelement besitzt einen Flow-Port mit der Richtung „out“ vom Datentyp Boolean, mit dem Eingaben an einen Flow-Port gemacht werden können. Der gesetzte Wert liegt kontinuierlich an.
PushButton		Dieses Steuerelement besitzt einen Flow-Port mit der Richtung „out“ vom Datentyp Boolean, mit dem Eingaben an einen Flow-Port gemacht werden können. Solange dieses betätigt ist (Maustaste unten), sendet es „true“, andernfalls immer „false“.
LedDisplay		Es dient zur Ausgabe von Werten vom Datentyp String. Es besitzt einen Flow-Port der Richtung „in“.
LedBulb	aus (false) / an (true)	Es dient zur Ausgabe von Werten vom Datentyp Boolean. Es besitzt einen Flow-Port der Richtung „in“.

SySimControlBar		Dieses ist das Steuerelement zum Starten, pausieren und beenden der Simulation und Anzeige der Simulationsszeit.

Der Vorteil des Einzeltests des Fahrstraßen-, Weichen- und Umfahrstraßenelements besteht darin, dass damit die Ein- und Ausgaben an den jeweiligen Flow-Ports unabhängig von einer konkreten Verbindung mit anderen Elementen über Konnektoren getestet werden können. Es handelt sich dabei um einen abstrakteren bzw. allgemeineren Test, als dies mit einem konkreten Elementverbindungsplan möglich wäre. Für die Erstellung einer Modellsimulation wurden folgende Schritte ausgeführt, welche von SySim technisch vorgegeben sind, um die Code-Generierung durchführen zu können:

1. Erstellung eines Simulationsszenarios in Form eines internen Blockdiagramms, welches die dann zu simulierenden Blöcke enthält.
2. Einfügen der zu testenden bzw. zu simulierenden Blöcke und SySim-Steuerelemente in das interne Blockdiagramm.
3. Verbinden der Flow-Ports der zu simulierenden Blöcke mit den Flow-Ports der SySim-Steuerelemente über Konnektoren.
4. Ausführen von SySim und übersetzen des Simulationsszenarios in Visual Basic-Quellcode.
5. Erstellung eines Programmfensters bzw. einer Benutzeroberfläche in der Visual Basic-Entwicklungsumgebung. Dabei werden die SySim-Steuerelemente auf dem Fenster wie gewünscht platziert und angeordnet.
6. Kompilieren des Simulationsprogramms.
7. Starten des Simulationsprogramms.
8. Eingabe von Werten und Auslesen von Werten an den Flow-Ports über die SySim-Steuerelemente.

Das Ergebnis dieser Schritte sind folgende drei Simulationsszenarien für das Fahrstraßen-, Weichen- und Umfahrstraßenelement (Abbildungen 5.1, 5.2 und 5.3). Zu sehen ist jeweils der modellierte und zu testende Block in der Mitte und an dessen Flow-Ports angeschlossene SySim-Steuerelemente.

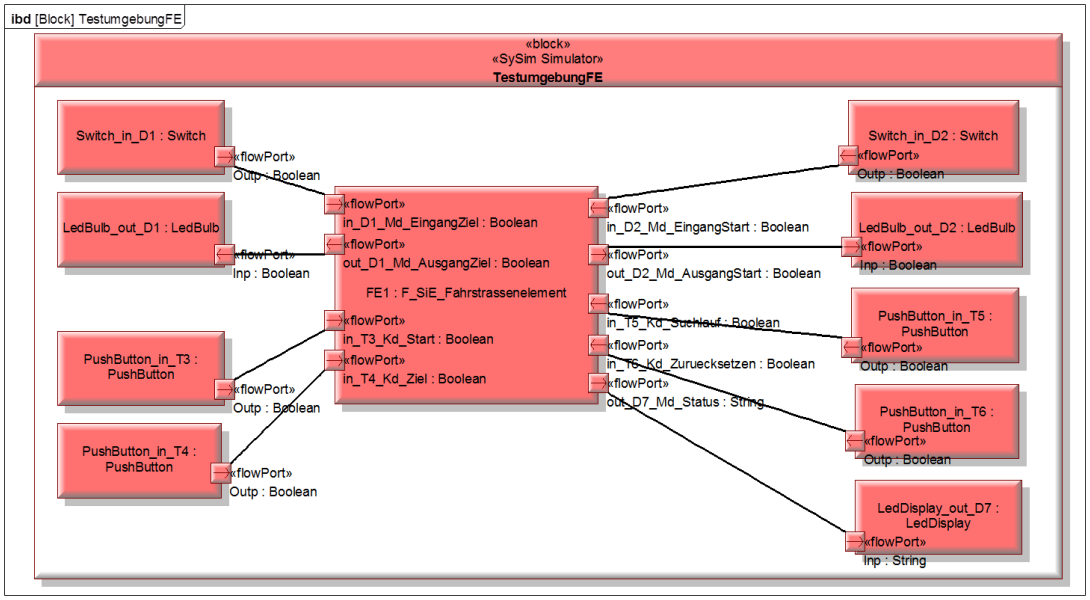

Abbildung 5.1: Internes Blockdiagramm des Simulationsszenarios für das Fahrstraßenelement

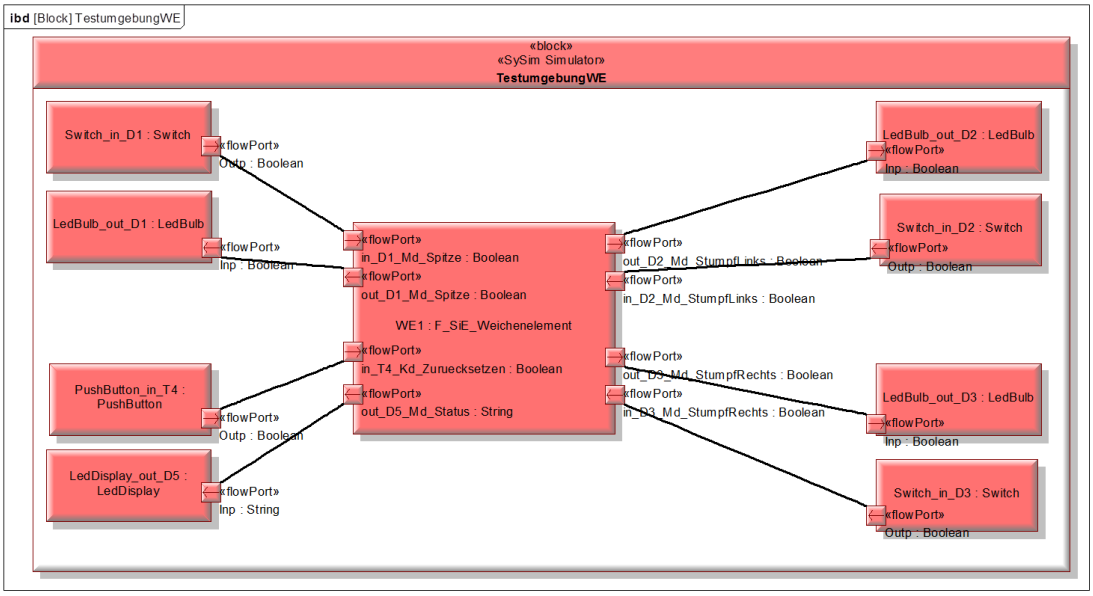

Abbildung 5.2: Internes Blockdiagramm des Simulationsszenarios für das Weichenelement

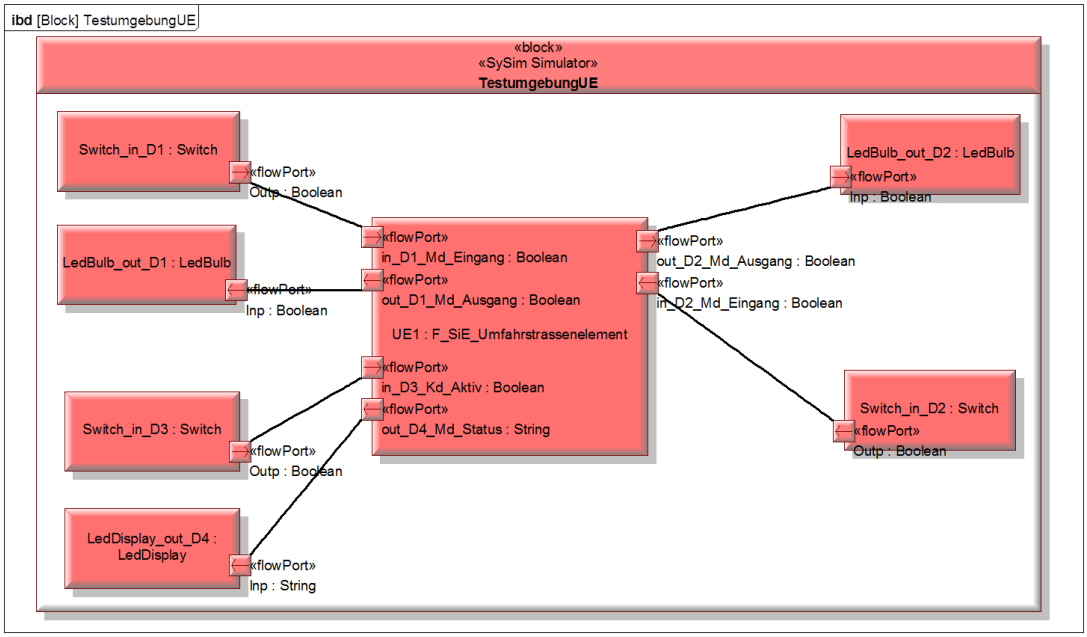

Abbildung 5.3: Internes Blockdiagramm des Simulationsszenarios für das Umfahrstraßenelement

5.3 Validierung Fahrstraßenelement

Zur Validierung des Fahrstraßenelements werden die Testfälle TF01 bis TF08 angewendet, in dem die im Testfall definierten Eingaben an den Flow-Ports über die Steuerelemente gemacht werden. Dabei wurden die Ausgaben der Flow-Ports protokolliert (Tabelle 5.4). Abbildung 5.4 zeigt die Benutzeroberfläche des Simulators des Fahrstraßenelements.

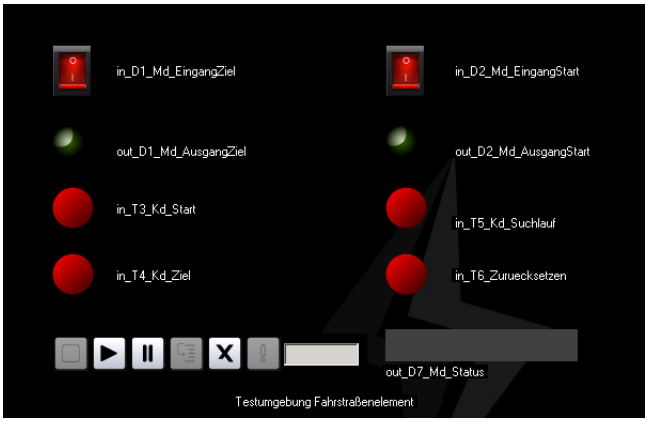

Abbildung 5.4: Benutzeroberfläche des Simulationsszenarios für das Fahrstraßenelement

Tabelle 5.4: Auswertung der Validierung des Fahrstraßenelements

Testfall	**erwartete Ausgabe**	**tatsächliche Ausgabe**
TF01	`out_D2_Md_AusgangStart = true`	`out_D2_Md_AusgangStart = true`
TF02	`out_D7_Md_Status = "START"`	`out_D7_Md_Status = "START"`
TF03	keine	keine
TF04	`out_D1_Md_AusgangZiel = true` `out_D7_Md_Status = "ZIEL"`	`out_D1_Md_AusgangZiel = true` `out_D7_Md_Status = "ZIEL"`
TF05	`out_D1_Md_AusgangZiel = true`	`out_D1_Md_AusgangZiel = true`
TF06	`out_D2_Md_AusgangStart = true`	`out_D2_Md_AusgangStart = true`

TF07	`out_D1_Md_AusgangZiel = false`
	`out_D2_Md_AusgangStart = false`
	`out_D7_Md_Status = ""`
		`out_D1_Md_AusgangZiel = false`
		`out_D2_Md_AusgangStart = false`
		`out_D7_Md_Status = ""`

TF08	`out_D1_Md_AusgangZiel = false`	`out_D1_Md_AusgangZiel = false`

Aus Tabelle 5.4 ergibt sich, dass die tatsächlichen Ausgaben mit den erwarteten übereinstimmen. Folglich erfüllt das Fahrstraßenelement die sprachlichen Anforderungen. Eine ausführliche Ausgabe der ausgelösten Transitionen ist im Anhang hinterlegt. Tabelle 5.5 fasst als Auswertung die ausgelösten Zustandsübergänge für das Fahrstraßenelement zusammen. Jede Transition des Zustandsautomaten des Fahrstraßenelements wurde dabei mindestens einmal ausgelöst.

Tabelle 5.5: Ausgelöste Transitionen im Zustandsautomaten des Fahrstraßenelements

Testfall	**ausgelöste Transition**
TF01	NichtMarkiert $\longrightarrow$ AlsStartMarkiert AlsStartMarkiert $\longrightarrow$ SuchsignalGesendet
TF02	SuchsignalGesendet $\longrightarrow$ EchosignalEmpfangen
TF03	NichtMarkiert $\longrightarrow$ AlsZielMarkiert
TF04	AlsZielMarkiert $\longrightarrow$ EchosignalGesendet
TF05	NichtMarkiert $\longrightarrow$ SuchsignalWeitergeleitet
TF06	SuchsignalWeitergeleitet $\longrightarrow$ EchosignalWeitergeleitet
TF07	Aktiv $\longrightarrow$ Aktiv Im Testfall: NichtMarkiert $\longrightarrow$ NichtMarkiert
TF08	SuchsignalWeitergeleitet $\longrightarrow$ NichtMarkiert

5.4 Validierung Weichenelement

Für die Validierung des Weichenelements werden die Testfälle TF09 bis TF19 angewendet, in dem die im Testfall definierten Eingaben an die Flow-Ports über die angeschlossenen Steuerelemente übergeben werden. Die Ausgaben der Flow-Ports sind in Tabelle 5.6 aufgelistet. Abbildung 5.5 zeigt die Benutzeroberfläche des Simulators des Weichenelements.

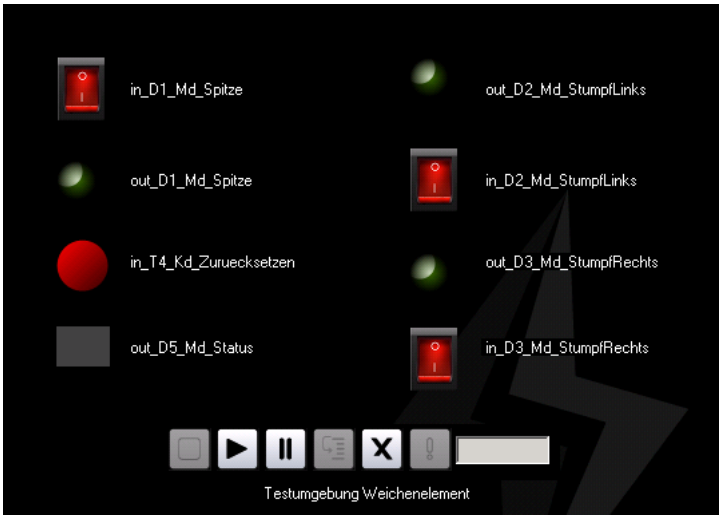

Abbildung 5.5: Benutzeroberfläche des Simulationsszenarios für das Weichenelement

Tabelle 5.6: Auswertung der Validierung des Weichenelements

Testfall	**erwartete Ausgabe**	**tatsächliche Ausgabe**
TF09	`out_D2_Md_StumpfLinks = true` `out_D3_Md_StumpfRechts = true`	`out_D2_Md_StumpfLinks = true` `out_D3_Md_StumpfRechts = true`

TF10	`out_D1_Md_Spitze = true`
	`out_D3_Md_StumpfRechts = false`
	`out_D5_Md_Status = "L"`
		`out_D1_Md_Spitze = true`
		`out_D3_Md_StumpfRechts = false`
		`out_D5_Md_Status := "L"`

TF11	`out_D1_Md_Spitze = true`
	`out_D2_Md_StumpfLinks = false`
	`out_D5_Md_Status = "R"`
		`out_D1_Md_Spitze = true`
		`out_D2_Md_StumpfLinks = false`
		`out_D5_Md_Status = "R"`

TF12	`out_D1_Md_Spitze = true`	`out_D1_Md_Spitze = true`
TF13	`out_D2_Md_StumpfLinks = true` `out_D5_Md_Status = "L"`	`out_D2_Md_StumpfLinks = true` `out_D5_Md_Status = "L"`
TF14	`out_D1_Md_Spitze = true`	`out_D1_Md_Spitze = true`
TF15	`out_D3_Md_StumpfRechts = true` `out_D5_Md_Status = "R"`	`out_D3_Md_StumpfRechts = true` `out_D5_Md_Status = "R"`

TF16	`out_D1_Md_Spitze = false`
	`out_D2_Md_StumpfLinks = false`
	`out_D3_Md_StumpfRechts = false`
	`out_D5_Md_Status = ""`
		`out_D1_Md_Spitze = false`
		`out_D2_Md_StumpfLinks = false`
		`out_D3_Md_StumpfRechts = false`
		`out_D5_Md_Status = ""`

TF17	`out_D2_Md_StumpfLinks = false` `out_D3_Md_StumpfRechts = false`	`out_D2_Md_StumpfLinks = false` `out_D3_Md_StumpfRechts = false`
TF18	`out_D1_Md_Spitze = false`	`out_D1_Md_Spitze = false`
TF19	`out_D1_Md_Spitze = false`	`out_D1_Md_Spitze = false`

Die tatsächlichen Ausgaben stimmen dabei mit den erwarteten überein. Dadurch erfüllt das Weichenelement die sprachlichen Anforderungen. Die Ausgabe der ausgelösten Transitionen ist im Anhang hinterlegt. Die Tabelle 5.8 fasst auswertend die ausgelösten Zustandsübergänge für das Weichenelement zusammen. Jede Transition des Zustandsautomaten wurde mindestens einmal ausgelöst.

Tabelle 5.8: Ausgelöste Transitionen im Zustandsautomaten des Weichenelements

Testfall	**ausgelöste Transition**
TF09	NichtMarkiert $\longrightarrow$ SpitzSelektiert
TF10	SpitzSelektiert $\longrightarrow$ InLinkslageMarkiert
TF11	SpitzSelektiert $\longrightarrow$ InRechtslageMarkiert
TF12	NichtMarkiert $\longrightarrow$ StumpfLinksSelektiert
TF13	StumpfLinksSelektiert $\longrightarrow$ InLinkslageMarkiert
TF14	NichtMarkiert $\longrightarrow$ StumpfRechtsSelektiert
TF15	StumpfRechtsSelektiert $\longrightarrow$ InRechtslageMarkiert
TF16	Aktiv $\longrightarrow$ Aktiv Im Testfall: NichtMarkiert $\longrightarrow$ NichtMarkiert
TF17	SpitzSelektiert $\longrightarrow$ NichtMarkiert
TF18	StumpfLinksSelektiert $\longrightarrow$ NichtMarkiert
TF19	StumpfRechtsSelektiert $\longrightarrow$ NichtMarkiert

5.5 Validierung Umfahrstraßenelement

Zur Validierung des Umfahrstraßenelements werden die Testfälle TF20 bis TF27 angewendet. In Tabelle 5.10 sind die Ergebnisse des Tests dargestellt. Abbildung 5.6 zeigt die Benutzeroberfläche des Simulators des Umfahrstraßenelements.

Tabelle 5.10: Auswertung der Validierung des Umfahrstraßenelements

Testfall	**erwartete Ausgabe**	**tatsächliche Ausgabe**
TF20	`out_D4_Md_Status = "AKTIV"`	`out_D4_Md_Status = "AKTIV"`
TF21	`out_D2_Md_Ausgang = true`	`out_D2_Md_Ausgang = true`
TF22	`out_D1_Md_Ausgang = true`	`out_D1_Md_Ausgang = true`
TF23	`out_D1_Md_Ausgang = true`	`out_D1_Md_Ausgang = true`
TF24	`out_D2_Md_Ausgang = true`	`out_D2_Md_Ausgang = true`
TF25	`out_D4_Md_Status = "INAKTIV"`	`out_D4_Md_Status = "INAKTIV"`
TF26	`out_D2_Md_Ausgang = false`	`out_D2_Md_Ausgang = false`
TF27	`out_D1_Md_Ausgang = false`	`out_D1_Md_Ausgang = false`

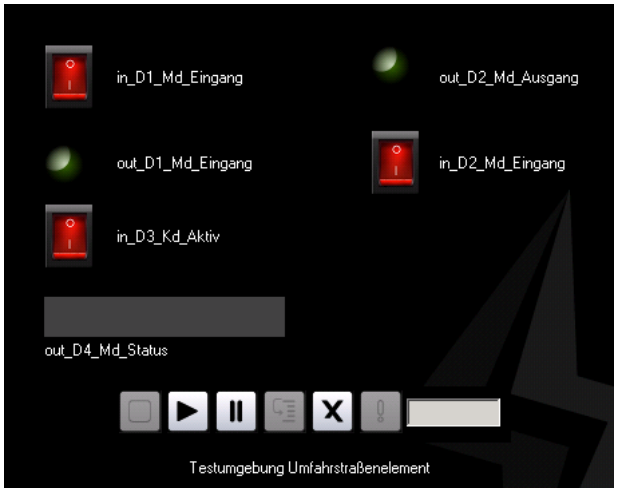

Abbildung 5.6: Benutzeroberfläche des Simulationsszenarios für das Umfahrstraßenelement

Das Umfahrstraßenelement erfüllt die sprachlichen Anforderungen, da alle tatsächlichen Ausgaben den erwarteten entsprechen. Folglich erfüllt das Umfahrstraßenelement die sprachlichen Anforderungen. Die ausführliche Ausgabe der ausgelösten Transitionen ist im Anhang aufgeführt. Tabelle 5.12 fasst als Auswertung die Zustandsübergänge für das Umfahrstraßenelement zusammen. Dabei wurden alle Transitionen im Zustandsautomaten des Umfahrstraßenelements mindestens einmal ausgelöst.

Tabelle 5.12: Ausgelöste Transitionen im Zustandsautomaten des Umfahrstraßenelements

Testfall	**ausgelöste Transitionen**
TF20	NichtAktiv $\longrightarrow$ Aktiv im Testfall: NichtAktiv $\longrightarrow$ KeinSignalWeitergeleitet
TF21	KeinSignalWeitergeleitet $\longrightarrow$ SuchsignalD1Weitergeleitet
TF22	SuchsignalD1Weitergeleitet $\longrightarrow$ EchosignalD2Weitergeleitet
TF23	KeinSignalWeitergeleitet $\longrightarrow$ SuchsignalD2Weitergeleitet
TF24	SuchsignalD2Weitergeleitet $\longrightarrow$ EchosignalD1Weitergeleitet
TF25	Aktiv $\longrightarrow$ NichtAktiv im Testfall: KeinSignalWeitergeleitet $\longrightarrow$ NichtAktiv
TF26	keine
TF27	keine

5.6 Auswertung der Einzelvaliderung

Das Fahrstraßen-, Weichen- und Umfahrstraßenelement wurden jeweils erfolgreich validiert und erfüllen damit die sprachlichen Anforderungen. Es wurden alle Testfälle abgearbeitet und durch Auswertung der Simulationsprotokolle jede Transition jedes Zustandsautomaten mindestens einmal durchlaufen.

5.7 Test der Fahrwegsuche an einem exemplarischen Elementverbindungsplan

Nachdem bei der jeweils einzelnen Validierung des Fahrstraßen-, Weichen- und Umfahrstraßenelements die Erfüllung der sprachlichen Anforderungen nachgewiesen und keine Modellfehler festgestellt wurden, kann nun ein exemplarischer SysML-Elementverbindungsplan als Simulationsszenario erstellt werden, der alle drei getesteten Elemente enthält. An diesem kann nun eine konkrete Fahrwegsuche simuliert und ausgeführt werden. Dieser Elementverbindungsplan mit der Bezeichnung „Testbahnhof2“ stellt einen Gleisplanausschnitt mit einer komplexeren Gleis- und Signalanordnung dar. Aufgrund der Größe dieses internen Blockdiagramms ist dessen Abbildung im Anhang hinterlegt. Zur besseren Übersichtlichkeit wurden in Artisan Studio die Stereotypen und Blocktypen der Elemente in diesem internen Blockdiagramm grafisch ausgeblendet. Der Gleisplan „Testbahnhof2“ ist ein ausführliches Simulationsszenario, dass den Aufbau eines SysML-Elementverbindungsplans mit den angeschlossenen SySim-Steuerelementen beschreibt. Dieser kann als Vorlage für andere zu erstellende Elementverbindungspläne verwendet werden. Insbesondere ist hier die Möglichkeit des Zurücksetzens aller Fahrstraßen- und Weichenelemente zu sehen, bei der es ein Steuerelement von Typ „PushButton“ mit der Bezeichnung „Reset“ gibt. Dieses ist mit allen Fahrstraßen- und Weichenelementen verbunden und kann alle Blöcke gleichzeitig zurücksetzen, nachdem eine Fahrwegsuche ausgeführt und danach eine neue durchgeführt werden soll.
Zum Testen dieses Szenarios sind die Testfälle TF28 bis TF32 aufgestellt, welche Eingaben und erwartete Ausgaben in Form von konkreten Fahrstraßen definieren. Jedoch werden nicht alle möglichen, sondern nur ausgewählte Fahrstraßen getestet, die aber möglichst unterschiedliche Anordnungen von Fahrstraßen-, Weichen- und Umfahrstraßenelementen enthalten. In TF28 und TF29 sind Start- und Zielpunkt der Fahrstraße für den zu ermittelnden Fahrweg gleich, jedoch werden dabei zwei unterschiedliche Fahrwege überprüft (Umfahrstraße). Nachfolgende Tabelle fasst die überprüften Fahrstraßen zusammen, für die ein Fahrweg ermittelt werden soll.

Testfall	**Charakteristik der Fahrstraße**
TF28	Fahrstraße über mindestens eine Weiche und einem aktiven Umfahrstraßenelement (erster möglicher Fahrweg bzw. Regelfahrstraße). Entgegen der Suchsignalrichtung befindet sich ein nicht aktiv beteiligtes Fahrstraßenelement.

TF29	Fahrstraße über mindestens eine Weiche und einem aktiven Umfahrstraßenelement (zweiter möglicher Fahrweg bzw. Umfahrstraße). Entgegen der Suchsignalrichtung befindet sich ein nicht aktiv beteiligtes Fahrstraßenelement.
TF30	Fahrstraße über mindestens eine Weiche. Entgegen der Suchsignalrichtung befindet sich ein nicht aktiv beteiligtes Fahrstraßenelement.
TF31	Fahrstraße über mindestens eine Weiche.
TF32	Fahrstraße ohne Weiche, die direkt von Signal zu Signal verläuft.

In Abbildung 5.7 ist die erstellte Benutzeroberfläche von „Testbahnhof2“ zu sehen. Dabei wurde eine grafische und intuitive Gleisplandarstellung mit Hilfe der Entwicklungsumgebung von Visual Basic erstellt. Tabelle 5.15 zeigt die Auswertung der Tests mit den Testfällen TF28 bis TF32.

Tabelle 5.15: Auswertung der Fahrstraßentests im Simulationsszenario „Testbahnhof2“

Testfall	**erwartete Ausgabe**	**tatsächliche Ausgabe**
TF28	`W1 = "R"`	`W1 = "R"`
	`W5 = "L"`	`W5 = "L"`
	`A = "START"`	`A = "START"`
	`N1 = "ZIEL"`	`N1 = "ZIEL"`
TF29	`W1 = "L"`	`W1 = "L"`
	`W2 = "L"`	`W2 = "L"`
	`W3 = "R"`	`W3 = "R"`
	`W4 = "R"`	`W4 = "R"`
	`W5 = "R"`	`W5 = "R"`
	`A = "START"`	`A = "START"`
	`N1 = "ZIEL"`	`N1 = "ZIEL"`
TF30	`W1 = "L"`	`W1 = "L"`
	`W2 = "L"`	`W2 = "L"`
	`W3 = "R"`	`W3 = "R"`
	`W4 = "L"`	`W4 = "L"`
	`A = "START"`	`A = "START"`
	`N2 = "ZIEL"`	`N2 = "ZIEL"`
TF31	`W3 = "L"`	`W3 = "L"`
	`W2 = "R"`	`W2 = "R"`
	`P3 = "START"`	`P3 = "START"`
	`101 = "ZIEL"`	`101 = "ZIEL"`
TF32	`101 = "START"`	`101 = "START"`
	`103 = "ZIEL"`	`103 = "ZIEL"`

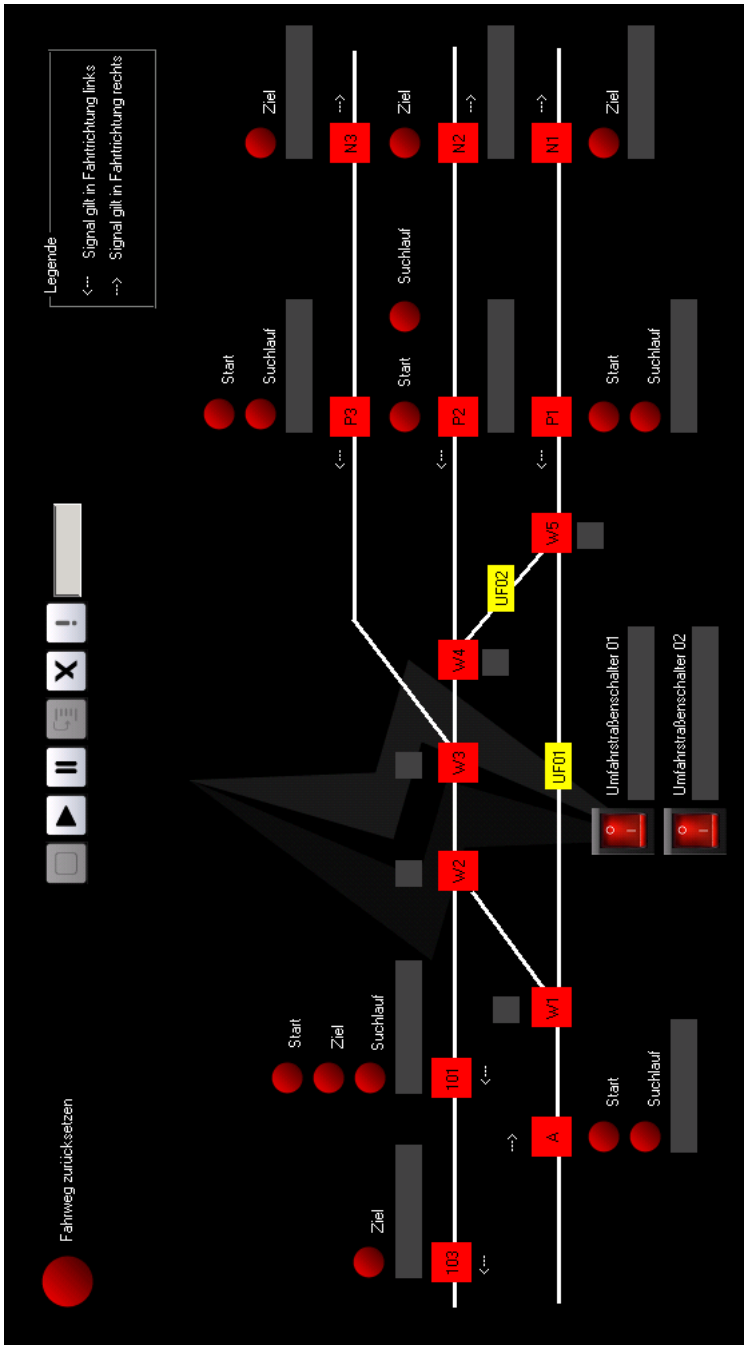

Abbildung 5.7: Benutzeroberfläche des Simulationsszenarios „Testbahnhof2“

5.8 Zusammenfassung der Validierung

Mit dem Einzeltest des Fahrstraßen-, Weichen-, und Umfahrstraßenelements wurde das Verhalten des jeweiligen Blocks anhand der sprachlichen Anforderungen überprüft bzw. validiert. Dabei wurde festgestellt, dass die modellbasierte Anforderungsspezifikation der Fahrwegsuche die sprachlichen Anforderungen erfüllt und frei von Modellfehlern ist. Erreicht wurde dies durch eine Abdeckung und die mindestens einmalige Auslösung aller Transitionen in jedem Zustandsautomaten.

6 Zusammenfassung und Fazit

In dieser Arbeit wurde anhand des Verfahrens der Fahrwegsuche nach dem Spurplanprinzip gezeigt, wie aus sprachlich formulierten Anforderungen eine modellbasierte Anforderungsspezifikation entwickelt wird. Als geeignetes Beschreibungsmittel wurde dazu die SysML verwendet und die Struktur- und Verhaltenssicht auf Basis der sprachlichen Anforderungen beschrieben. Um die einzelnen funktionalen Anforderungen zu erfüllen, wurden passende SysML-Sprachelemente ausgewählt. Die Struktur wurde mit Blöcken und das Verhalten über Zustandsautomaten modelliert. Mit dem entworfenen Fahrstraßen- und Weichenelement sowie dem optionalen Umfahrstraßenelement wurden die Bausteine für die Fahrwegsuche erstellt, mit denen ein Elementverbindungsplan in SysML konstruiert werden kann.

Anschließend wurde das entstandene SysML-Modell mittels Validierung getestet. Dazu wurden aus den sprachlichen Anforderungen Testfälle abgeleitet und für jeden Block ein Simulationsszenario erstellt. Letzteres wurde in Visual Basic-Quellcode übersetzt und ausführbar gemacht. Das in Quellcode übersetzte Szenario wurde mit einer Benutzeroberfläche ausgestattet, die aus SySim-Steuerelementen besteht.

Mittels Validierung wurden dann Fahrstraßen-, Weichen- und Umfahrstraßenelement jeweils in einem Einzeltest simuliert, geprüft und dabei jeder Testfall abgearbeitet. Hier wurde festgestellt, dass jeder Block die sprachlichen Anforderungen der Fahrwegsuche erfüllt. Dies ist damit zu begründen, dass die Transitionen, bzw. deren Verhaltensbeschreibung zu einem großen Teil direkt aus einzelnen Anforderungen abgeleitet werden konnten. Außerdem wurden mit den Testfällen jeweils alle Transitionen in den Zustandsautomaten ausgelöst und im Test abgedeckt. Wesentlicher Vorteil des Einzeltests der Blöcke ist, dass damit das Verhalten unabhängig von einem konkreten Elementverbindungsplan überprüft und Such- bzw. Echosignale durch passende Steuerelemente simuliert werden konnten. Zusätzlich wurde ein weiteres Simulationsszenario erstellt, das einen konkreten Elementverbindungsplan enthält und alle drei modellierten Elemente einsetzt. Dieser dient als Vorlage, wie man in SysML eine funktionierende Simulation der Fahrwegsuche erstellt und durchführt.

Als besonders geeignet hat sich die SysML bei der Darstellung von Elementverbindungsplänen erwiesen. Im Ergebnis ermöglicht die SysML mit ihren gegebenen Sprachelementen eine passende Nachbildung eines Elementverbindungsplans. Zusammenfassend ist die Fahrwegsuche nach dem Spurplanprinzip nun modellbasiert, allgemein und unabhängig von einer Implementierung spezifiziert.

Bei der Erstellung von SysML-Zustandsautomaten in der Verhaltenssicht könnte als Ergänzung außerdem künftig in Betracht gezogen werden, diese mit Hilfe des Minimierungs-

algorithmus aus der theoretischen Informatik in der Anzahl ihrer Zustände zu minimieren. Dazu würden die Zustandsautomaten zuvor in deterministische, endliche Automaten überführt werden. Ein Zweck ist, zu untersuchen, ob der aus sprachlichen Anforderungen entwickelte Zustandsautomat unnötige Zustände besitzt und optimiert werden kann. Zu beachten wäre aber dabei, dass die domänenspezifische Lesbarkeit und die durch Transitionen umgesetzten Anforderungen nicht verloren gehen.

Das SysML-Modell der Fahrwegsuche kann praktisch in einer möglichen, umfangreicheren Spezifikation von Stellwerkstechnik genutzt werden. Es ist denkbar, diese modellbasierte Anforderungsspezifikation im Projekt „NeuPro“ einzubringen, sofern eine Modellierung der Sicherungslogik angestrebt wird.

Literatur

[1] Saeid Arabestani. „Formal verifizierbare objektorientierte Systemspezifikationen mit UML für Eisenbahnsicherungssysteme“. Diss. Technische Universität Braunschweig, 2005.

[2] Ernst Behr. „Sarstedt - erstes SpDrS 60-Stellwerk“. In: *Signal und Draht* 57 (Jan. 1965), S. 1–6.

[3] Randolf Berglehner, Markus Eberhardt und Theodor Staneff. „CENELEC-konformes Requirements Management und Requirements Engineering“. In: *Eisenbahntechnische Rundschau* (Okt. 2008), S. 686–692.

[4] Werner Busch und Dieter Steffens. „Das Spurplanstellwerk SpDrS 600 - Erfahrung und Bewährung“. In: *Eisenbahntechnische Rundschau* 32 (Jan. 1983), S. 23–28.

[5] CENELEC. *DIN EN 50126:2000-03: Bahnanwendungen - Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS)*. März 2000.

[6] CENELEC. *DIN EN 50128:2012-03: Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Software für Eisenbahnsteuerungs- und Überwachungssysteme*. März 2012.

[7] DB Netz AG. *Richtlinie 301 - Signalbuch*. 2014.

[8] DB Netz AG. *Richtlinie 408 - Züge fahren und Rangieren*. 2012.

[9] DB Netz AG (Yuen Man Hon). „Modellierungsstandard für modellbasiertes Vorgehen“. Version 0.3. Feb. 2013.

[10] Dirk. H. Enders und Ferdinand Hein. *Sp Dr 60-Stellwerke bedienen. Der Regelbetrieb*. 4. Aufl. Bahn Fachverlag, Juni 2012.

[11] Jürgen Ernst. *Das SpDrS60-Stellwerk (DB-Fachbuch)*. 3. Aufl. Eisenbahn-Fachverlag Heidelberg-Mainz, 1993.

[12] Wolfgang Fenner, Peter Naumann und Jochen Trinckauf. *Bahnsicherungstechnik*. Hrsg. von Siemens Aktiengesellschaft. 2. Aufl. Publicis Corporate Publishing, 2003.

[13] Yuen Man Hon. „An Engineering-Oriented Formal Framework for Railway Interlocking Systems Requirements Specifications“. Diss. Technische Universität Braunschweig, 2009.

[14] Oliver Lemke. „Modellbasierte Anforderungsspezifikation sicherheitskritischer Systeme im Bahnbereich - Beschreibungsmittel und ihre Anwendung“. Diss. Technische Universität Braunschweig, 2009.

[15] Ulrich Maschek. *Sicherung des Schienenverkehrs*. 2. Aufl. Springer Vieweg, 2013.

[16] *OMG Systems Modeling Language (OMG SysML) Version 1.2*. OMG Document Number: formal/2010-06-01. Object Management Group. 2010-06-01.

[17] *OMG Unified Modeling Language (OMG UML) Version 2.4.1, Superstructure*. OMG Document Number: formal/2011-08-06. Object Management Group. 2011-08-06.

[18] Jörn Pachl. *Systemtechnik des Schienenverkehrs*. 7. Aufl. Springer Vieweg, 2013.

[19] Walter Schmitz. „Das Spurplanstellwerk SpDrS 60“. In: *Signal und Draht* 54 (Feb. 1962), S. 17–39.

[20] Siemens Aktiengesellschaft, Bereich Eisenbahnsignaltechnik. *Zentralstellwerk Hzf Bahnhof Hamburg Hbf*. Stellwerksbroschüre (Nr. F 513/122).

[21] Werner Ulrich und Gerhard Hugel. „Das Zentralstellwerk Nürnberg Hbf“. In: *Die Bundesbahn* 64 (Sep. 1988), S. 849–853.

[22] Horst Walther und Hansjörg Appel. „Das Spurplanstellwerk SpDrS 600 - konstruktiver Aufbau und Schaltung Teil 2“. In: *Signal + Draht* 71 (Okt. 1979), S. 201–209.

[23] Ludwig Wehner. „Das Stellwerk SpDrS 600“. In: *Signal + Draht* 70 (März 1978), S. 43–47.

[24] Otto Wolf. „Inbetriebnahme des Zentralstellwerks München Hbf“. In: *Signal und Draht* 57 (Jan. 1965), S. 8–11.

7 Anhang

7.1 Auflistung der Anforderungen

7.1.1 Allgemeine funktionale Anforderungen

Anforderung	Beschreibung
FA01	Die zur Durchführung einer Fahrwegsuche berücksichtigten Fahrwegelemente sind die Repräsentation eines Haupt- oder Lichtsperrsignals und die Repräsentation einer Weiche.
FA02	Die Repräsentation eines Haupt- oder Sperrsignals wird durch ein Fahrstraßenelement dargestellt, da Haupt- und Lichtsperrsignale Start- oder Zielpunkt einer Fahrstraße sind.
FA03	Die Repräsentation einer Weiche wird durch ein Weichenelement dargestellt.
FA04	Zur Durchführung einer Fahrwegsuche müssen die Fahrstraßen- und Weichenelemente durch einen Elementverbindungsplan nach dem Spurplanprinzip miteinander verbunden werden.
FA05	Eine Fahrwegsuche wird zwischen zwei Fahrstraßenelementen durchgeführt.
FA06	Der Startpunkt eines Fahrweges wird durch eine Start-Markierung des gewünschten Fahrstraßenelements festgelegt.
FA07	Der Zielpunkt eines Fahrweges wird durch eine Ziel-Markierung des gewünschten Fahrstraßenelements festgelegt.
FA08	Die Kommunikation zwischen Fahrstraßen- und Weichenelementen findet mittels Such- und Echosignalen statt.

7.1.2 Funktionale Anforderungen Fahrstraßenelement

Anforderung	Beschreibung
FA09	Such- und Echosignale werden von Fahrstraßenelementen gesendet und empfangen.

FA10	Ein Suchsignal wird nur von dem Fahrstraßenelement aus gesendet, das vorher als Start-Fahrstraßenelement markiert wurde.
FA11	Ein Suchsignal wird von einem Fahrstraßenelement nur in die Richtung gesendet, in der das Fahrstraßenelement Startpunkt einer Fahrstraße sein kann.
FA12	Ein Echosignal wird nur von dem Fahrstraßenelement aus gesendet, das vorher als Ziel-Fahrstraßenelement markiert wurde.
FA13	Ein Echosignal wird von einem Ziel-Fahrstraßenelement nur in die Richtung gesendet, aus der das Suchsignal empfangen wurde.
FA14	Ein Echosignal wird dann von einem Ziel-Fahrstraßenelement aus gesendet, wenn es ein Suchsignal empfangen hat.
FA15	Ein Fahrstraßenelement muss seine Start- oder Zielmarkierung nach dem Empfangen des Echosignals nach außen ausgeben. Die Startmarkierung lautet dabei „START“ und die Zielmarkierung „ZIEL“.

7.1.3 Funktionale Anforderungen Weichenelement

Anforderung	**Beschreibung**
FA16	Weichenelemente leiten Such- und Echosignale weiter.
FA17	Empfängt ein Weichenelement ein Suchsignal aus Richtung der Weichenspitze, so wird das Suchsignal in den linken Strang und in den rechten Strang des Weichenstumpfs weitergeleitet.
FA18	Empfängt ein Weichenelement ein Suchsignal aus Richtung des Weichenstumpfs am linken oder rechten Strang, so wird das Suchsignal nur zur Weichenspitze weitergeleitet.
FA19	Ein empfangenes Echosignal am Weichenelement aus Richtung der Weichenspitze wird nur genau an den Strang (links oder rechts) weitergeleitet, an dem vorher ein Suchsignal empfangen wurde.
FA20	Ein empfangenes Echosignal am Weichenelement aus Richtung des Weichenstumpfs (linker Strang) wird nur zur Weichenspitze weitergeleitet und nur dann, wenn an der Weichenspitze vorher ein Suchsignal empfangen wurde. Das Suchsignal an den rechten Strang wird zurückgenommen.

FA21	Ein empfangenes Echosignal am Weichenelement aus Richtung des Weichenstumpfs (rechter Strang) wird nur zur Weichenspitze weitergeleitet und nur dann, wenn an der Weichenspitze vorher ein Suchsignal empfangen wurde. Das Suchsignal an den linken Strang wird zurückgenommen.
FA22	Ein Weichenelement muss seine Weichenlage nach dem Empfangen des Echosignals nach außen ausgeben. Für die Linkslage wird „L“ ausgegeben und für die Rechtslage „R“.

7.1.4 Zusätzliche funktionale Anforderungen

Diese Anforderungen ergeben sich aus der Themenabgrenzung der reinen Fahrwegsuche von den anderen Schritten der Fahrstraßenbildung.

Anforderung	Beschreibung
FA23	Empfängt ein Fahrstraßenelement, das weder Start- noch Zielpunkt ist, ein Such- oder Echosignal, so muss es das entsprechende Signal ohne weitere Aktion zum nächsten Element weiterleiten. Dieser Fall tritt auf, wenn ein Suchsignal auf ein Fahrstraßenelement trifft und dieses Suchsignal in entgegengesetzter Richtung zur Geltungsrichtung des Fahrstraßenelements verläuft.
FA24	Fahrstraßen- und Weichenelemente müssen zurücksetzbar sein. Zurücksetzbar sein bedeutet, dass nach einer erfolgten Fahrwegsuche alle Elemente wieder in einen Ausgangszustand überführt werden können, damit beliebig viele, auch unterschiedliche Fahrwegsuchvorgänge hintereinander durchgeführt werden können. Das Zurücksetzen erfordert dabei die Löschung der Start-, Ziel- und aller Weichenlagemarkierungen.

7.2 Auflistung der optionalen Anforderungen

7.2.1 Funktionale Anforderungen Umfahrstraßenelement

Anforderung	Beschreibung
FA25	Ein Umfahrstraßenelement ist ein zusätzliches Fahrwegelement, welches Such- und Echosignale blockiert oder weiterleitet.
FA26	Es muss zwei Zustände aufweisen. Diese Zustände legen fest, ob es blockiert oder weiterleitet. In einem Zustand blockiert es, im anderen leitet es Such- und Echosignale weiter.

FA27	Durch eine Eingabe muss zwischen den beiden Zuständen gewechselt werden können.
FA28	Ein Umfahrstraßenelement muss seinen aktuellen Zustand nach außen ausgeben. Für den blockierenden Zustand wird „INAKTIV“ ausgegeben, im nicht-blockierenden Zustand „AKTIV“.

7.3 Auflistung der Testfälle

7.3.1 Testfälle für das Fahrstraßenelement

Testfall	Eingabe	erwartete Ausgabe
TF01	`in_T3_Kd_Start := true;` `in_T5_Kd_Suchlauf := true;`	`out_D2_Md_AusgangStart = true`
TF02	`in_D2_Md_EingangStart := true;`	`out_D7_Md_Status = "START"`
TF03	`in_T4_Kd_Ziel := true;`	keine
TF04	`in_D1_Md_EingangZiel := true;`	`out_D1_Md_AusgangZiel = true` `out_D7_Md_Status = "ZIEL"`
TF05	`in_D2_Md_EingangStart := true;`	`out_D1_Md_AusgangZiel = true`
TF06	`in_D1_Md_EingangZiel := true;`	`out_D2_Md_AusgangStart = true`

TF07	`in_T6_Kd_Zurecksetzen := true;`	`out_D1_Md_AusgangZiel = false`
		`out_D2_Md_AusgangStart = false`
		`out_D7_Md_Status = ""`

TF08	`in_D2_Md_EingangStart := false;`	`out_D1_Md_AusgangZiel = false`

7.3.2 Testfälle für das Weichenelement

Testfall	Eingabe	erwartete Ausgabe
TF09	`in_D1_Md_Spitze := true;`	`out_D2_Md_StumpfLinks = true` `out_D3_Md_StumpfRechts = true`

TF10	`in_D2_Md_StumpfLinks := true;`	`out_D1_Md_Spitze = true`
		`out_D3_Md_StumpfRechts = false`
		`out_D5_Md_Status = "L"`

TF11	`in_D3_Md_StumpfRechts := true;`	`out_D1_Md_Spitze = true`
		`out_D2_Md_StumpfLinks = false`
		`out_D5_Md_Status = "R"`

TF12	`in_D2_Md_StumpfLinks := true;`	`out_D1_Md_Spitze = true`
TF13	`in_D1_Md_Spitze := true;`	`out_D2_Md_StumpfLinks = true` `out_D5_Md_Status = "L"`
TF14	`in_D3_Md_StumpfRechts := true;`	`out_D1_Md_Spitze = true`
TF15	`in_D1_Md_Spitze := true;`	`out_D3_Md_StumpfRechts = true` `out_D5_Md_Status = "R"`

TF16	`in_T4_Kd_Zuruecksetzen := true;`	`out_D1_Md_Spitze = false`
		`out_D2_Md_StumpfLinks = false`
		`out_D3_Md_StumpfRechts = false`
		`out_D5_Md_Status = ""`

TF17	`in_D1_Md_Spitze := false;`	`out_D2_Md_StumpfLinks = false` `out_D3_Md_StumpfRechts = false`
TF18	`in_D2_Md_StumpfLinks := false;`	`out_D1_Md_Spitze = false`
TF19	`in_D3_Md_StumpfRechts := false;`	`out_D1_Md_Spitze = false`

7.3.3 Testfälle für das Umfahrstraßenelement

Testfall	Eingabe	erwartete Ausgabe
TF20	`in_D3_Kd_Aktiv := true;`	`out_D4_Md_Status = "AKTIV"`
TF21	`in_D1_Md_Eingang := true;`	`out_D2_Md_Ausgang = true`
TF22	`in_D2_Md_Eingang := true;`	`out_D1_Md_Ausgang = true`
TF23	`in_D2_Md_Eingang := true;`	`out_D1_Md_Ausgang = true`
TF24	`in_D1_Md_Eingang := true;`	`out_D2_Md_Ausgang = true`
TF25	`in_D3_Kd_Aktiv := false;`	`out_D4_Md_Status = "INAKTIV"`
TF26	`in_D1_Md_Eingang := true;`	`out_D2_Md_Ausgang = false`
TF27	`in_D2_Md_Eingang := true;`	`out_D1_Md_Ausgang = false`

7.3.4 Testfälle Beispiel-Lageplan

Testfall	Fahrstraße	erwartete Ausgabe

TF28	A $\longrightarrow$ N1
	über W1, UF01, W5
		`W1 = "R"`
		`W5 = "L"`
		`A = "START"`
		`N1 = "ZIEL"`

TF29	A $\longrightarrow$ N1
	über W1, W2, W3, W4, UF02, W5
		`W1 = "L"`
		`W2 = "L"`
		`W3 = "R"`
		`W4 = "R"`
		`W5 = "R"`
		`A = "START"`
		`N1 = "ZIEL"`

TF30	A $\longrightarrow$ N2
	über W1, W2, W3, W4
		`W1 = "L"`
		`W2 = "L"`
		`W3 = "R"`
		`W4 = "L"`
		`A = "START"`
		`N2 = "ZIEL"`

TF31	P3 $\longrightarrow$ 101 über W3, W2	W3 = "L" W2 = "R" P3 = "START" 101 = "ZIEL"
TF32	101 $\longrightarrow$ 103	101 = "START" 103 = "ZIEL"

7.4 Ausgabeprotokolle von Artisan Studio der ausgelösten Transitionen in den Tests

Listing 7.1: Ausgabe TF01

```
F_SiE_Fahrstrassenelement[1] : Exit NichtMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_T3_Kd_Start = true )/
F_SiE_Fahrstrassenelement[1] : Enter AlsStartMarkiert
F_SiE_Fahrstrassenelement[1] : Exit AlsStartMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_T5_Kd_Suchlauf = true )/
                               out_T2_Md_AusgangStart := true;
F_SiE_Fahrstrassenelement[1] : Enter SuchsignalGesendet
```

Listing 7.2: Ausgabe TF02

```
F_SiE_Fahrstrassenelement[1] : Exit SuchsignalGesendet
F_SiE_Fahrstrassenelement[1] : Transition when( in_D2_Md_EingangStart = true )/
                               out_T7_Md_Status := "START";
F_SiE_Fahrstrassenelement[1] : Enter EchosignalEmpfangen
```

Listing 7.3: Ausgabe TF03

```
F_SiE_Fahrstrassenelement[1] : Exit NichtMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_T4_Kd_Ziel = true )/
F_SiE_Fahrstrassenelement[1] : Enter AlsZielMarkiert
```

Listing 7.4: Ausgabe TF04

```
F_SiE_Fahrstrassenelement[1] : Exit AlsZielMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_D1_Md_EingangZiel = true )/
                               out_T1_Md_AusgangZiel := true;
                               out_T7_Md_Status := "ZIEL";
F_SiE_Fahrstrassenelement[1] : Enter EchosignalGesendet
```

Listing 7.5: Ausgabe TF05

```
F_SiE_Fahrstrassenelement[1] : Exit NichtMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_D2_Md_EingangStart = true )/
                               out_T1_Md_AusgangZiel := true;
F_SiE_Fahrstrassenelement[1] : Enter SuchsignalWeitergeleitet
```

Listing 7.6: Ausgabe TF06

```
F_SiE_Fahrstrassenelement[1] : Exit SuchsignalWeitergeleitet
F_SiE_Fahrstrassenelement[1] : Transition when( in_D1_Md_EingangZiel = true )/
                               out_T2_Md_AusgangStart := true;
F_SiE_Fahrstrassenelement[1] : Enter EchosignalWeitergeleitet
```

Listing 7.7: Ausgabe TF07

```
F_SiE_Fahrstrassenelement[1] : Enter NichtMarkiert
F_SiE_Fahrstrassenelement[1] : Transition when( in_T6_Kd_Zuruecksetzen )/
F_SiE_Fahrstrassenelement[1] : Enter NichtMarkiert
```

Listing 7.8: Ausgabe TF08

```
F_SiE_Fahrstrassenelement[1] : Exit SuchsignalWeitergeleitet
F_SiE_Fahrstrassenelement[1] : Transition when( in_D2_Md_EingangStart = false )/
F_SiE_Fahrstrassenelement[1] : Enter NichtMarkiert
```

Listing 7.9: Ausgabe TF09

```
F_SiE_Weichenelement[1] : Exit NichtMarkiert
F_SiE_Weichenelement[1] : Transition when( in_D1_Md_Spitze = true )/
                                out_T2_Msg_StumpfGerade := true;
                                out_T3_Msg_StumpfAbzweigend := true;
F_SiE_Weichenelement[1] : Enter SpitzSelektiert
```

Listing 7.10: Ausgabe TF10

```
F_SiE_Weichenelement[1] : Exit SpitzSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D2_Md_StumpfLinks = true )/
                                out_T1_Md_Spitze := true;
                                out_T3_Md_StumpfRechts := false;
                                out_T4_Md_Status := "L";
F_SiE_Weichenelement[1] : Enter InLinkslageMarkiert
```

Listing 7.11: Ausgabe TF11

```
F_SiE_Weichenelement[1] : Exit SpitzSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D3_Md_StumpfRechts = true )/
                                out_T1_Msg_Spitze := true;
                                out_T2_Msg_StumpfGerade := false;
                                out_T4_Msg_Status := "R";
F_SiE_Weichenelement[1] : Enter InRechtslageMarkiert
```

Listing 7.12: Ausgabe TF12

```
F_SiE_Weichenelement[1] : Exit NichtMarkiert
F_SiE_Weichenelement[1] : Transition when( in_D2_Md_StumpfLinks = true )/
                                out_T1_Msg_Spitze := true;
F_SiE_Weichenelement[1] : Enter StumpfLinksSelektiert
```

Listing 7.13: Ausgabe TF13

```
F_SiE_Weichenelement[1] : Exit StumpfLinksSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D1_Md_Spitze = true )/
                                out_T2_Msg_StumpfGerade := true;
                                out_T4_Msg_Status := "L";
F_SiE_Weichenelement[1] : Enter InLinkslageMarkiert
```

Listing 7.14: Ausgabe TF14

```
F_SiE_Weichenelement[1] : Exit NichtMarkiert
F_SiE_Weichenelement[1] : Transition when( in_D3_Md_StumpfRechts = true )/
                                out_T1_Msg_Spitze := true;
F_SiE_Weichenelement[1] : Enter StumpfRechtsSelektiert
```

Listing 7.15: Ausgabe TF15

```
F_SiE_Weichenelement[1] : Exit StumpfRechtsSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D1_Md_Spitze = true )/
                                out_T3_Msg_StumpfAbzweigend := true;
                                out_T4_Msg_Status := "R";
F_SiE_Weichenelement[1] : Enter InRechtslageMarkiert
```

Listing 7.16: Ausgabe TF16

```
F_SiE_Weichenelement[1] : Enter NichtMarkiert
F_SiE_Weichenelement[1] : Transition when( in_T4_Kd_Zuruecksetzen )/
F_SiE_Weichenelement[1] : Enter NichtMarkiert
```

Listing 7.17: Ausgabe TF17

```
F_SiE_Weichenelement[1] : Exit SpitzSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D1_Md_Spitze = false )/
F_SiE_Weichenelement[1] : Enter NichtMarkiert
```

Listing 7.18: Ausgabe TF18

```
F_SiE_Weichenelement[1] : Exit StumpfLinksSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D2_Md_StumpfLinks = false )/
F_SiE_Weichenelement[1] : Enter NichtMarkiert
```

Listing 7.19: Ausgabe TF19

```
F_SiE_Weichenelement[1] : Exit StumpfRechtsSelektiert
F_SiE_Weichenelement[1] : Transition when( in_D3_Md_StumpfRechts = false )/
F_SiE_Weichenelement[1] : Enter NichtMarkiert
```

Listing 7.20: Ausgabe TF20

```
F_SiE_Umfahrstrassenelement[1] : Exit NichtAktiv
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D3_Kd_Aktiv = true )/
                                 out_D4_Md_Status := "AKTIV";
F_SiE_Umfahrstrassenelement[1] : Enter Initial
F_SiE_Umfahrstrassenelement[1] : Exit Initial
F_SiE_Umfahrstrassenelement[1] : Enter Aktiv
F_SiE_Umfahrstrassenelement[1] : Transition /
F_SiE_Umfahrstrassenelement[1] : Enter KeinSignalWeitergeleitet
```

Listing 7.21: Ausgabe TF21

```
F_SiE_Umfahrstrassenelement[1] : Exit KeinSignalWeitergeleitet
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D1_Md_Eingang = true )/
                                 out_T2_Md_Ausgang := true;
F_SiE_Umfahrstrassenelement[1] : Enter SuchsignalD1Weitergeleitet
```

Listing 7.22: Ausgabe TF22

```
F_SiE_Umfahrstrassenelement[1] : Exit SuchsignalD1Weitergeleitet
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D2_Md_Eingang = true )/
                                 out_T1_Md_Ausgang := true;
F_SiE_Umfahrstrassenelement[1] : Enter EchosignalD2Weitergeleitet
```

Listing 7.23: Ausgabe TF23

```
F_SiE_Umfahrstrassenelement[1] : Exit KeinSignalWeitergeleitet
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D2_Md_Eingang = true )/
                                 out_T1_Md_Ausgang := true;
F_SiE_Umfahrstrassenelement[1] : Enter SuchsignalD2Weitergeleitet
```

Listing 7.24: Ausgabe TF24

```
F_SiE_Umfahrstrassenelement[1] : Exit SuchsignalD2Weitergeleitet
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D1_Md_Eingang = true )/
                                 out_D2_Md_Ausgang := true;
F_SiE_Umfahrstrassenelement[1] : Enter EchosignalD1Weitergeleitet
```

Listing 7.25: Ausgabe TF25

```
F_SiE_Umfahrstrassenelement[1] : Exit KeinSignalWeitergeleitet
F_SiE_Umfahrstrassenelement[1] : Transition when( in_D3_Kd_Aktiv = false )/
F_SiE_Umfahrstrassenelement[1] : Enter NichtAktiv
```

7.4.1 Ausgabeprotokolle bei Simulationsstart

Listing 7.26: Ausgabe Fahrstraßenelement

```
F_SiE_Fahrstrassenelement[1] : Instance Created F_SiE_Fahrstrassenelement
F_SiE_Fahrstrassenelement[1] : Enter Initial
F_SiE_Fahrstrassenelement[1] : Exit Initial
F_SiE_Fahrstrassenelement[1] : Transition /
F_SiE_Fahrstrassenelement[1] : Enter Initial
F_SiE_Fahrstrassenelement[1] : Exit Initial
F_SiE_Fahrstrassenelement[1] : Enter Aktiv
F_SiE_Fahrstrassenelement[1] : Transition /
F_SiE_Fahrstrassenelement[1] : Enter NichtMarkiert
```

Listing 7.27: Ausgabe Weichenelement

```
F_SiE_Weichenelement[1] : Instance Created F_SiE_Weichenelement
F_SiE_Weichenelement[1] : Enter Initial
F_SiE_Weichenelement[1] : Exit Initial
F_SiE_Weichenelement[1] : Transition /
F_SiE_Weichenelement[1] : Enter Initial
F_SiE_Weichenelement[1] : Exit Initial
F_SiE_Weichenelement[1] : Enter Aktiv
F_SiE_Weichenelement[1] : Transition /
F_SiE_Weichenelement[1] : Enter NichtMarkiert
```

Listing 7.28: Ausgabe Umfahrstraßenelement

```
F_SiE_Umfahrstrassenelement[1] : Instance Created F_SiE_Umfahrstrassenelement
F_SiE_Umfahrstrassenelement[1] : Enter Initial
F_SiE_Umfahrstrassenelement[1] : Exit Initial
F_SiE_Umfahrstrassenelement[1] : Transition /
F_SiE_Umfahrstrassenelement[1] : Enter NichtAktiv
```

7.5 Abbildung des internen Blockdiagramms des Simulationszenarios Testbahnhof2

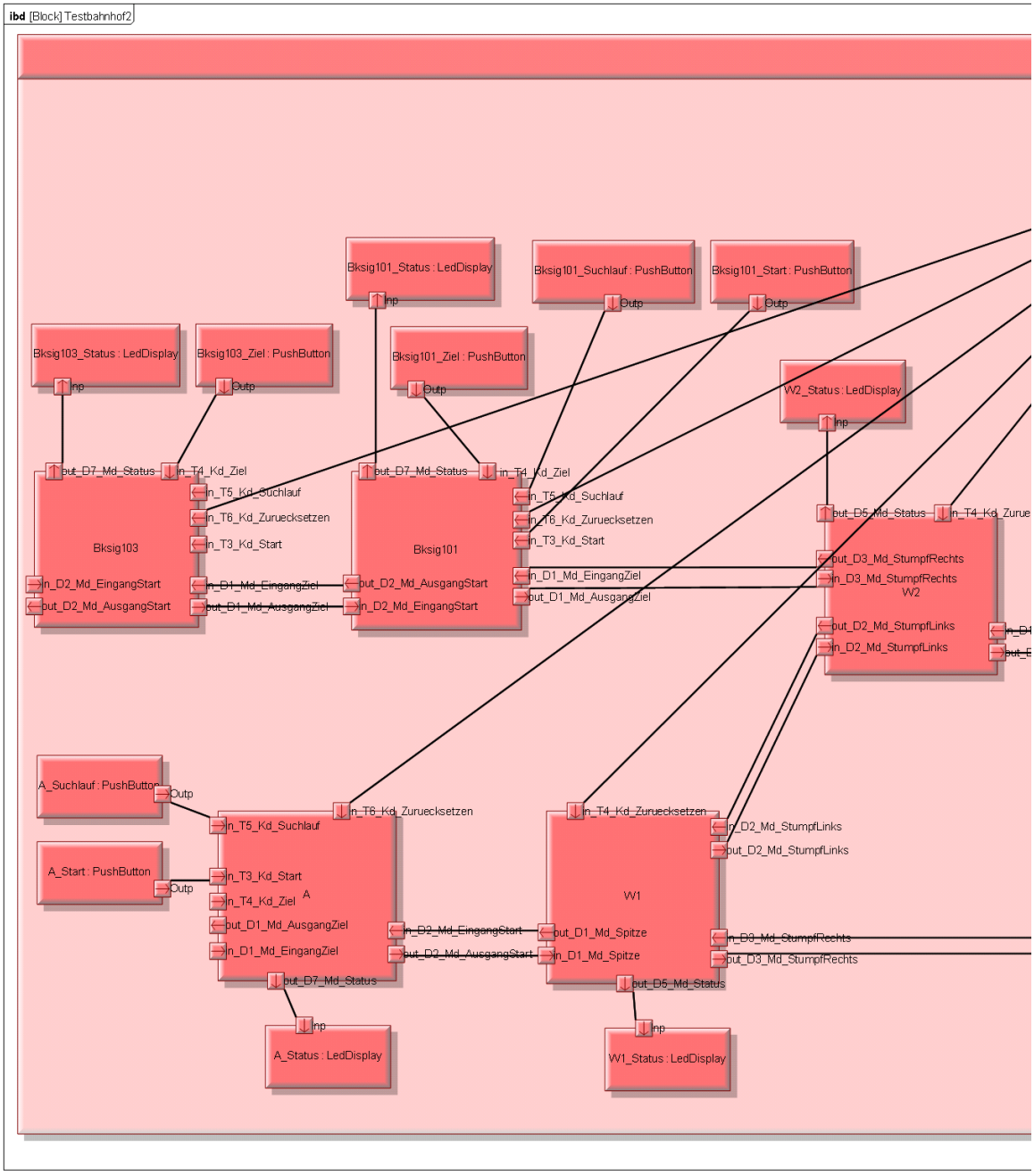

Abbildung 7.1: Internes Blockdiagramm Testbahnhof2 - Teil 1

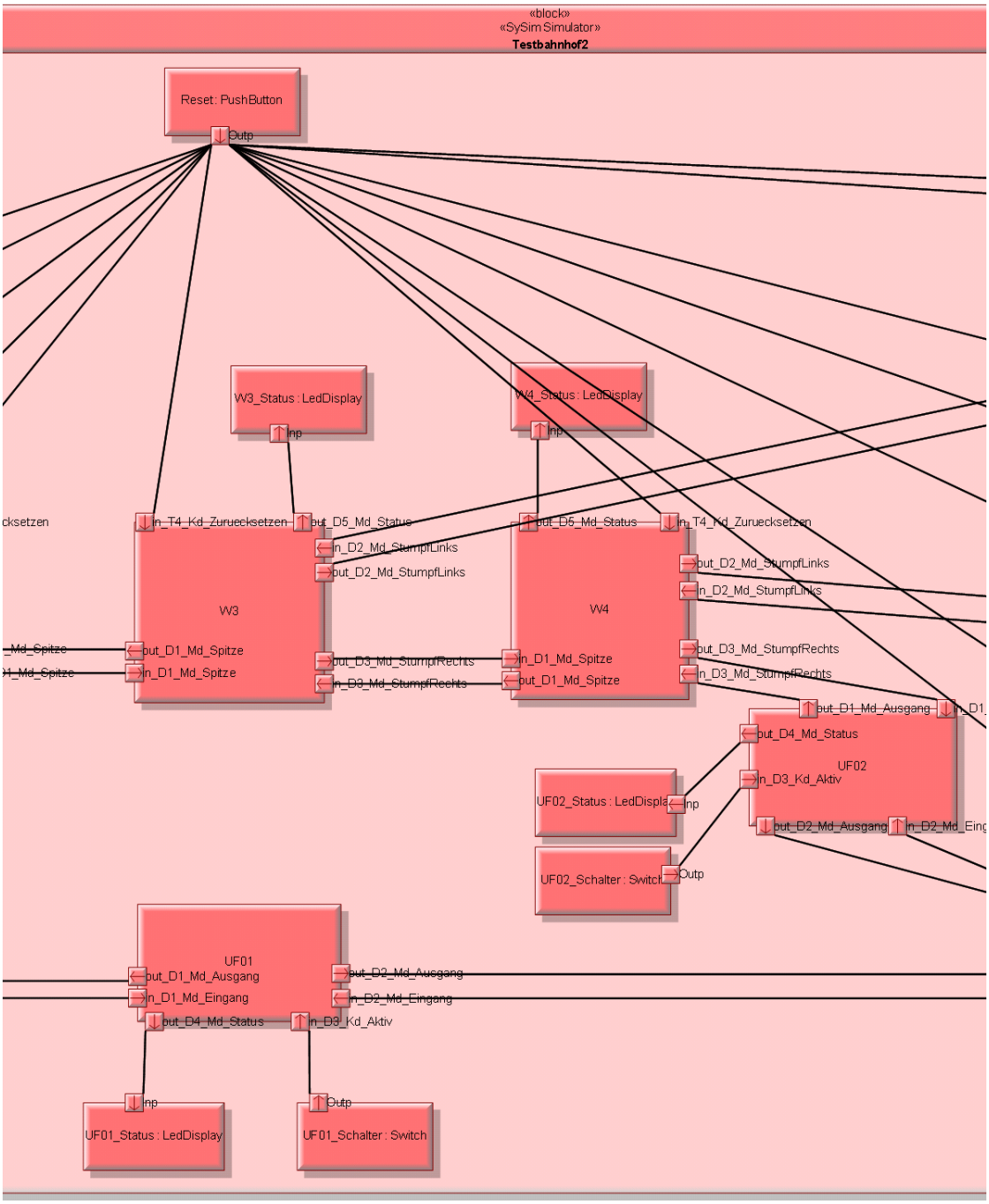

Abbildung 7.2: Internes Blockdiagramm Testbahnhof2 - Teil 2

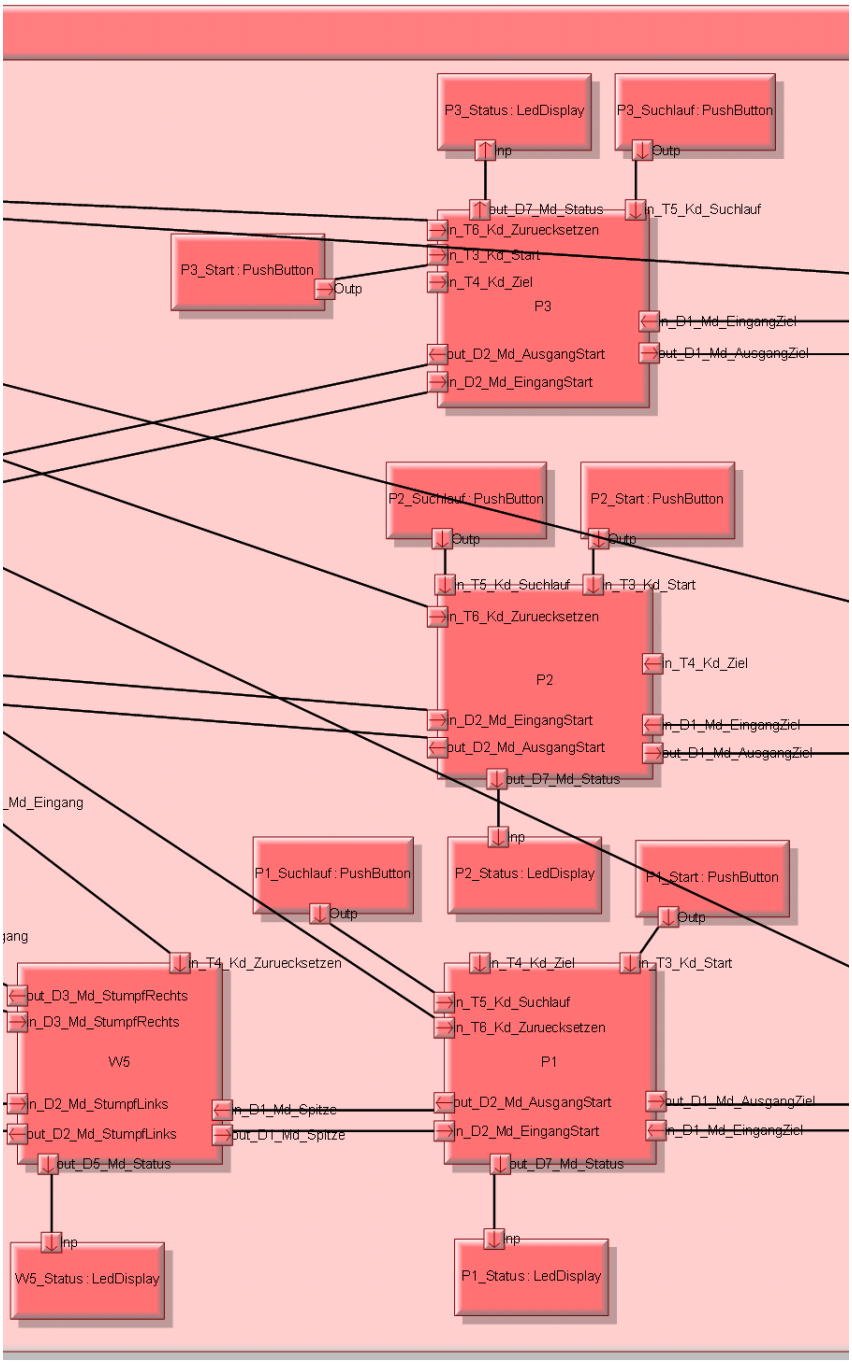

Abbildung 7.3: Internes Blockdiagramm Testbahnhof2 - Teil 3

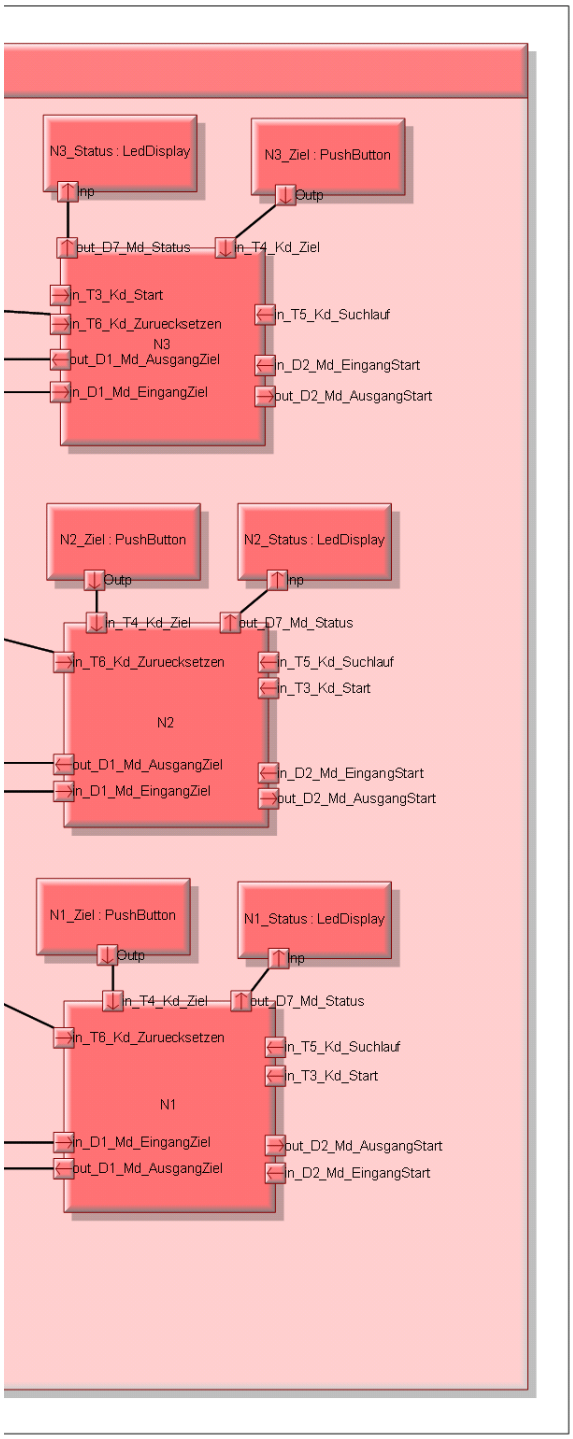

Abbildung 7.4: Internes Blockdiagramm Testbahnhof2 - Teil 4