

Projekt DB 01/97

ETCS-Phase I

Projekt- dokumentation

**Einführung in die Netze,
die graphische Notation und
den Nummernschlüssel**

bearbeitet im Auftrage
der DB-AG München

Projektleitung:
Dipl.-Ing. M. Meyer zu Hörste

Braunschweig, April 1997

**TECHNISCHE
UNIVERSITÄT
BRAUNSCHWEIG**

**Institut für Regelungs- und
Automatisierungstechnik**

Univ.-Prof. Dr.-Ing. E. Schnieder

Langer Kamp 8
38106 Braunschweig

Postfach 3329
38023 Braunschweig

Telefon 0531 / 391-3317
Telefax 0531 / 391-5197

Technische Universität Braunschweig
Institut für Regelungs- und Automatisierungstechnik

| | |
|--------------|---------------------------------------|
| Zeichen: | IfRA / TUBS |
| Dokument: | einf4_1 |
| Stand: | 16.04.98 13:47 |
| Bearbeitung: | Dipl.-Ing. M. Meyer zu Hörste |
| Adresse: | Langer Kamp 8 D-38106 Braunschweig |
| Telefon: | +49 (0) 531 / 391 - 3331 |
| Telefax: | +49 (0) 531 / 391 - 5197 |

Inhaltsverzeichnis

| | |
|--|-----------|
| 1 EINLEITUNG..... | 4 |
| 2 EINFÜHRUNG IN DIE PETRINETZE | 5 |
| 2.1 GRUNDREGELN | 5 |
| 2.2 FARBIGE PETRINETZE | 5 |
| 2.3 HIERARCHISCHE PETRINETZE..... | 6 |
| 2.4 SIMULATION VON PETRINETZEN..... | 8 |
| 2.5 ANALYSE VON PETRINETZEN | 10 |
| 2.6 TESTS AUF DER BASIS VON PETRINETZEN..... | 11 |
| 3 SPEZIFISCHE DEFINITIONEN | 12 |
| 3.1 NETZLOGIK | 12 |
| 3.2 NETZSTRUKTUR..... | 12 |
| 3.3 NETZELEMENTE..... | 14 |
| 3.4 NUMMERNSYSTEM | 16 |
| 4 ANHANG | 19 |
| 4.1 BIBLIOGRAPHIE | 19 |
| 4.2 GLOSSAR | 20 |
| 4.3 ABKÜRZUNGSVERZEICHNIS..... | 22 |

1 Einleitung

Dieses Dokument soll als allgemeine Erläuterung zu den formalen Spezifikationen der Teilsysteme des ETCS dienen. Der dargestellte Ausschnitt aus der vielgestaltigen Welt der Petrinetze ist spezifisch auf die Anforderungen der Modellierung dieser Spezifikation abgestimmt. Für den allgemeinen Einstieg in diese Methode sind einige Werke in dem Literaturverzeichnis aufgeführt, das sich im Anhang befindet [Abel90, Baumgarten90, Reisig90, Schnieder93].

Dieses Dokument ist folgendermaßen strukturiert:

Nach der Einleitung in diesem Kapitel folgt eine Einführung in die Theorie der Petrinetze in Kapitel 2. Im dritten Kapitel finden sich die spezifischen Definitionen für die ERTMS/ETCS Dokumente. Dies sind neben den spezifischen Restriktionen für den Entwurf der Netze auch die Notationen für die Grafik, die keine funktionale Bedeutung im Netz haben. Das Nummernsystem wird ebenfalls in diesem Kapitel erläutert. Im Anhang findet sich ein Literaturverzeichnis, eine Übersicht über die graphischen Elemente sowie der Nummernschlüssel.

2 Einführung in die Petrinetze

2.1 Grundregeln

Ein Petrinetz setzt sich aus Ereignissen und Zuständen zusammen.

Jedes Ereignis setzt als Bedingung für sein Eintreten einen Vorzustand voraus. Entsprechend hat jedes Ereignis einen Nachzustand.

Die Zustandsinformationen befinden sich in Speichern, die als Stellen oder Plätze bezeichnet werden. In der graphischen Notation sind sie durch ein Oval dargestellt. Jeder Platz kann eine oder mehrere Informationen aufnehmen. Diese werden als Marken bezeichnet.

Die elementaren Ereignisse werden als Transitionen bezeichnet und durch ein Rechteck abgebildet.

Die Verbindungen zwischen den Stellen und Transitionen erfolgen durch Kanten, die als Pfeile dargestellt werden.

2.2 Farbige Petrinetze

Die Marken entsprechen in ihrem Typ der Typdeklaration in einer Programmiersprache. Jeder Platz kann nur Marken eines bestimmten Typs aufnehmen. Damit verschiedene Marken gleichen Typs in einer Transition verarbeitet werden können, wird jeder Kante eine Variable zugewiesen.

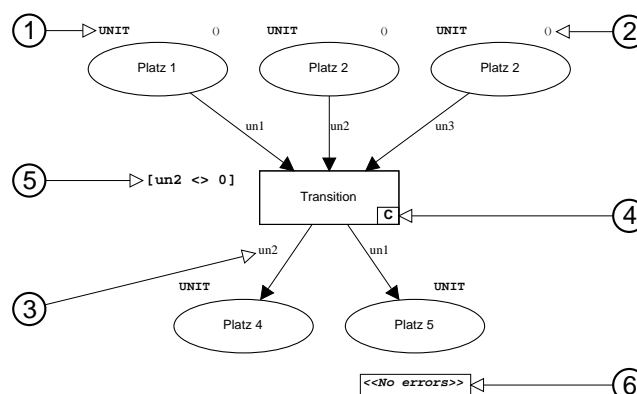


Bild 1: Farbiges Petrinetz

Zu Bild 1: In diesem Netz werden die verschiedenen Regionen gezeigt. In Region (1) ist der Typ der Marken, die auf dem zugehörigen Platz liegen können, aufgeführt; dieser Typ wird als

Colorset bezeichnet. Entsprechend zu diesem Typ gibt es eine Variablenzuweisung zu jeder Kante, die der Identifikation dient (3). Region (2) zeigt Initialmarkierungen. Diese Marken werden vor dem Programmablauf auf die entsprechenden Plätze gelegt und bilden den Ausgangszustand. Region (4) markiert eine „Code-Region“. Dabei handelt es sich um einen Programmtext, der das Ereignis beschreibt, das durch Schalten der Transition eintritt. Grundsätzliche Schaltbedingung für Transitionen ist die Markenbelegung der Vorplätze. Eine Differenzierung der Schaltbedingung kann durch einen „Guard“ erfolgen (Region (5)). Hier können bestimmte geforderte Eigenschaften der eingehenden (abgezogenen) Marken geprüft werden. Region (6) zeigt die „Error-Box“. Hier werden Fehler angezeigt, die beim Syntaxcheck des Programms gefunden wurden.

Die Festlegung der Typen und Variablen, die für ein Netz zur Verfügung stehen, erfolgt in einem speziellen Bereich innerhalb der Netze, dem sogenannten „Global Declaration Node“.

Der Global Declaration Node für das Netz in Bild 1 sieht folgendermaßen aus:

```
(*****)
(* Global Declaration Node *)
(*****)

(*****)
(* Definition der ColorSets *)
(*****)

color UNIT = unit;

(*****)
(* Definition der Variablen *)
(*****)

var un1, un2, un3 : UNIT;

(*****)
```

Bild 2: Global Declaration Node

Im ersten Abschnitt unter der Überschrift „Definition der Colorsets“ werden die Typen für die Plätze definiert. Im folgenden Abschnitt unter der Überschrift „Definition der Variablen“ wird jeder Variablen ein Colorset zugewiesen.

2.3 Hierarchische Petrinetze

Elementare Petrinetze werden ab einer gewissen Größe unübersichtlich. Daher nutzt man die Möglichkeit, Teile der Netzstruktur zusammenzufassen und in Unternetzen abzulegen. Ein Unternetz wird auf dem übergeordneten Netz durch eine Transition repräsentiert. Das Schaltverhalten kann, bedingt durch die innere Struktur, von dem Verhalten elementarer

Transitionen abweichen. In diesem Fall spricht man zur Unterscheidung nicht mehr von einer Transition, sondern von einer Instanz. Das Unternetz wird auch als Verfeinerung bezeichnet. Bild 3 zeigt ein Netz mit einer Instanz, die durch ein Kästchen mit der Inschrift „HS“ für „hierarchical subpage“ gekennzeichnet ist.

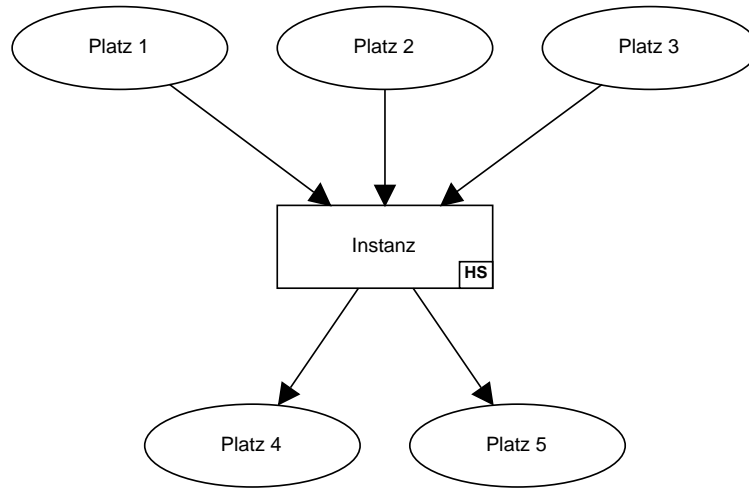


Bild 3: Netz mit Instanz

Der Inhalt einer Instanz läßt sich wiederum als Petrinetz darstellen. Die Bedingung dabei ist, daß alle Plätze, die mit der Instanz verbunden sind, auch im Unternetz erscheinen. Die Plätze, die Marken in das Unternetz hineinführen, werden als „Input-Port-Places“ bezeichnet; entsprechend tragen die Plätze, die Marken aus dem Netz herausleiten, die Bezeichnung „Output-Port-Places“. Das folgende Bild zeigt eine passende Verfeinerung zu der Instanz in Bild 3:

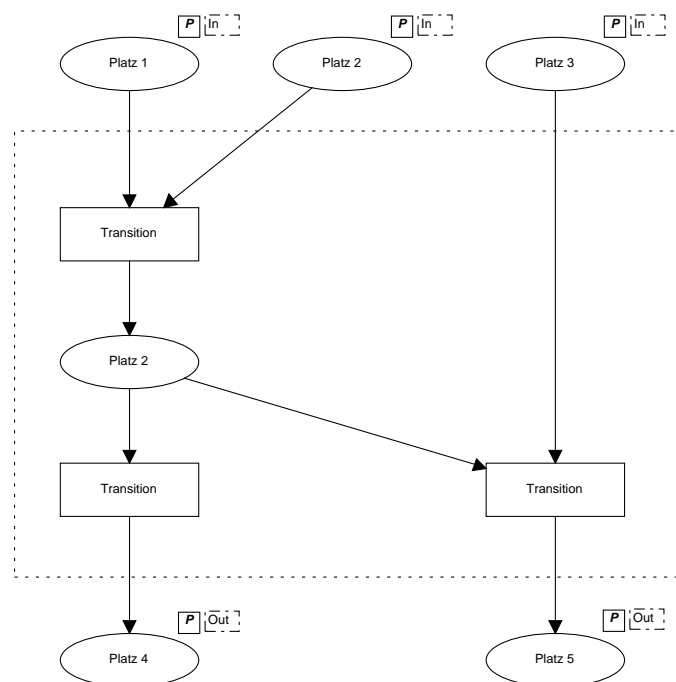


Bild 4: Unternetz zu Bild 5

Bei komplexen hierarchischen Strukturen kann es recht aufwendig werden, Unternetze, die durch mehrere Netzebenen getrennt sind, mit Portplätzen zu verbinden. Abhilfe schafft in diesem Fall ein „Fusion-Place“. Es handelt sich dabei um einen Platz, der beliebig oft innerhalb der Netzstruktur bzw. innerhalb einzelner Netze angebunden werden kann. Wird ein „Fusion-Place“ markiert, steht diese Markierung gleichzeitig an allen Abbildungen dieses Platzes zur Verfügung. Sie werden mit der zusätzlichen Anschrift FG („fusion - global“) im Netz gekennzeichnet.

2.4 Simulation von Petrinetzen

Elementares Schaltverhalten

Das Schaltverhalten einer Transition setzt sich aus zwei Teilschritten zusammen: dem Testen auf Schaltbarkeit und dem eigentlichen Schalten. Eine Transition, die schaltbar ist, wird auch als konzessioniert bezeichnet.

Testen auf Schaltbarkeit (Konzessionierung)

Eine Transition kann schalten, wenn alle ihre Vorplätze besetzt sind und wenn ihre Nachplätze frei sind bzw. wenn die Kapazität der Nachplätze noch nicht ausgeschöpft ist.

Bild 5 zeigt ein einfaches Netz mit einer Transition die drei Vor- und zwei Nachplätze hat.

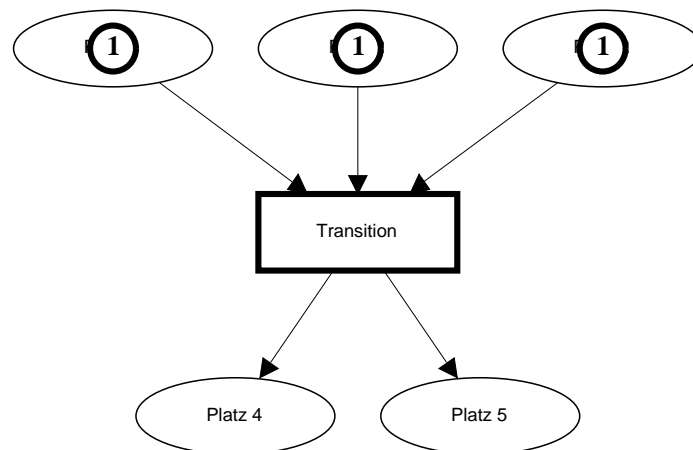


Bild 5: Netz mit einer schaltfähigen Transition

Es sind alle Vorplätze belegt, so daß die Transition schalten kann. Dieser Zustand wird durch den verbreiterten Rahmen der Transition gekennzeichnet.

Schalten einer Transition

Schaltet eine Transition, so wird von jedem Vorplatz eine Marke abgezogen. Die in den Marken enthaltenen Daten können von der Transition verarbeitet werden.

Die Transition erzeugt beim Schalten für jeden Nachplatz eine neue Marke, die innerhalb der Transition neu berechnete Daten aufnimmt. Diese Marken werden zum Abschluß des Schaltvorgangs auf die Nachplätze gelegt.

Bild 6 zeigt das Netz von Bild 5 nach dem Schaltvorgang: die Transition hat geschaltet und damit je eine Marke auf jedem der Nachplätze erzeugt.

Die Ziffer in der Marke gibt an wie viele Marken sich auf dem Platz befinden.

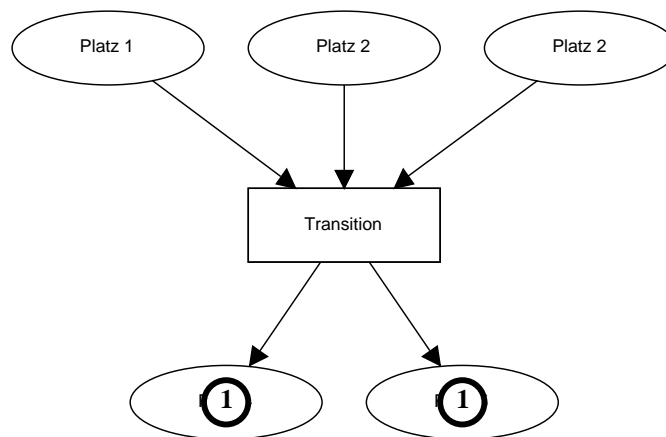


Bild 6: Netz nach dem Schalten einer Transition

2.5 Analyse von Petrinetzen

Für die Untersuchung von Petrinetzen existieren unterschiedliche Verfahren. Sie lassen sich prinzipiell in simulative, algebraische und graphentheoretische Verfahren unterteilen. Bild 7 zeigt die Untersuchungsmöglichkeiten von Petrinetzen auf.

Bei einer simulativen Untersuchung können Fehler im Entwurf bzw. im beschreibenden Modell bei einer systematischen Vorgehensweise u. U. gefunden werden. Hierbei ist es prinzipiell möglich, ein “Gefühl für das Systemverhalten” zu bekommen, daher Grenzen und Wirkung sowie Kausalität innerhalb des Systems zu betrachten. Die Simulation ist jedoch kein Beweis für die Fehlerfreiheit des Systems. Sie ist mit dem Debuggen eines Programms zu vergleichen, wobei bestimmte Fehler entdeckt werden können. Die Simulation schließt das Vorhandensein weiterer Fehler allerdings nicht aus.

Auf der Grundlage des Erreichbarkeitsgraphen von Petrinetzen können dynamische Netzeigenschaften unmittelbar abgelesen werden. Hierbei zählen die Erreichbarkeit einer Markierung und die Existenz partieller und totaler Verklemmungen zu den wichtigsten Analyseergebnissen, da sie eine direkte Aussage über die Funktionsfähigkeit eines (Teil-) Systems liefern können. Wichtig ist hier, daß der Analyse von Erreichbarkeitsgraphen die Graphentheorie, also eine mathematische Basis, zugrunde liegt [Abel90].

Das Problem der graphentheoretischen Analyse ist die Größe der Erreichbarkeitsmenge, die eng mit der Anzahl der Nebenläufigkeiten im System zusammenhängt. Die Größe des Erreichbarkeitsgraphen hat Rückwirkungen einerseits auf die Rechenzeit und andererseits auf den Speicherbedarf für die Erreichbarkeitsmenge. Aus diesem Grund setzt die Erreichbarkeitsanalyse eine begrenzte Anzahl von Systemzuständen und Nebenläufigkeiten voraus. Diese Grenzen lassen sich in Abhängigkeit von Rechen- und Speicherleistung

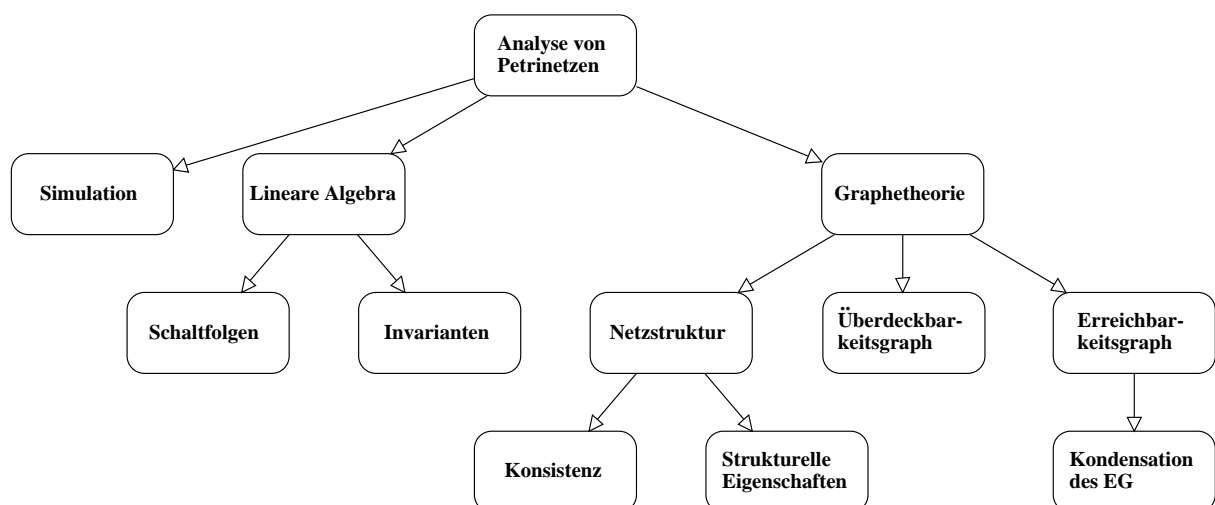


Bild 7: Analysemöglichkeiten

berechnen.

Alternativ zur graphentheoretischen Analyse ermöglichen Methoden der linearen Algebra die Ableitung von Aussagen über das dynamische Verhalten eines Petrinetzes direkt aus der Netzstruktur, d.h. nicht über den Umweg der Konstruktion des Erreichbarkeitsgraphen. Hierbei gibt es grundsätzlich zwei Möglichkeiten: Zum einen die Berechnung von Schaltfolgen und zum anderen die Ermittlung von Invarianten. Nachteil der strukturellen Analyseverfahren ist, daß insbesondere allgemeine Netze nur eingeschränkte Aussagen zulassen. Es ist jedoch prinzipiell möglich, bestimmte Netzstrukturen zu erkennen und deren Verhalten zu interpretieren [Lemmer95].

2.6 Tests auf der Basis von Petrinetzen

Mit Hilfe der Ereignisfolge, die durch das Petrinetz definiert werde ist es möglich funktionale Test für eine Implementierung des Systems zu erstellen. Auf der Basis der Simulation ergibt sich damit ein Soll-Verlauf gegen den ein reales System getestet werden kann. Wenn es Verzweigungen im Petrinetz gibt kann sich eine große Zahl an Testfällen aus einem Netz ergeben.

3 Spezifische Definitionen

3.1 Netzlogik

Im Standard-Layout dieses Projektes befindet sich die Ablauflogik der Netze im zentralen Bereich zwischen den Schnittstellen. Grundsätzlich wird ein Netz von oben nach unten durchlaufen

3.2 Netzstruktur

Die komplexe Struktur des Systems erfordert einen Aufbau des Netzes mit verschiedenen Verfeinerungsebenen. Dieser Aufbau weist drei verschiedene Netztypen auf: Auf den höchsten Ebenen wird der Prozeß im Überblick dargestellt. In den unterlagerten Ebenen werden einzelne Szenarios abgebildet und verknüpft. Innerhalb dieser Szenarios gibt es Funktionen, die so komplex sind, daß für die Modellierung eine weitere Verfeinerung erforderlich ist. Diese unterste Ebene ist die Funktionen-Ebene.

Die Netze einer jeden Ebene haben einen charakteristischen Aufbau:

- Die Netze der Prozeßebene sind in drei Bereiche unterteilt: Im zentralen Bereich (1) befinden sich die Transitionen und Plätze, die die weitere Anwendungslogik enthalten. Rechts und links daneben befinden sich zwei dunkelgraue Kästen (2, 3), in denen die Schnittstellen nach außen liegen. Die Nachrichten, die zu einem Empfänger gesendet werden oder von einem Sender empfangen werden, sind jeweils auf einem Platz zusammengefaßt. Dabei ist auf der Zugseite der rechte und auf der RBC-Seite der linke Kasten nur der Kommunikation zwischen Zug und Leitsystem vorbehalten.

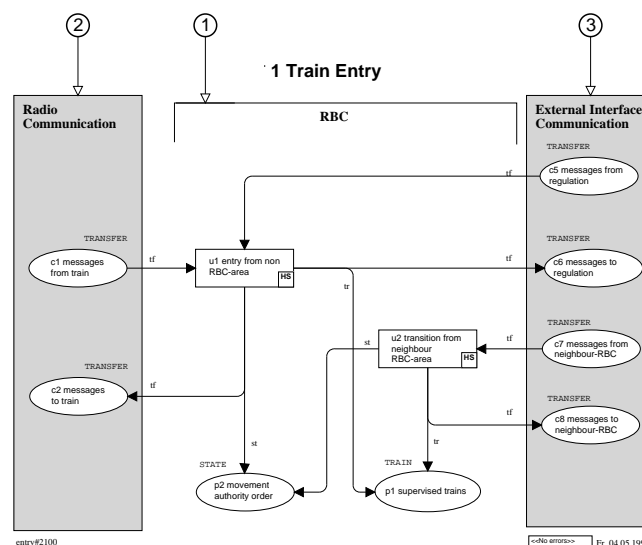


Bild 8: Struktur der Prozeß-Netze

- Auch Szenario-Netze weisen diese drei Bereiche auf. Die Anwendungslogik befindet sich wieder in der Mitte (1), und ganz außen befinden sich die beiden Kästen, die die Schnittstellen enthalten (2, 3). Neben den Schnittstellen befindet sich zur Logik hin auf jeder Seite eine Box mit der Bezeichnung *Radio-Driver* (5), bzw. *Interface-Driver* (4). Alle Transitionen in diesen Boxen beginnen mit *send* oder *receive*; sie enthalten die Empfangs- oder Sendelogik für eine spezifische Nachricht. Daher befindet sich zwischen diesen Boxen und der Anwendungslogik noch einer oder mehrere dunkelgraue Kästen (6, 7), der für jede einzelne Nachricht einen Platz enthält. Die Bezeichnung dieser Kastens beginnt mit *Messages*.

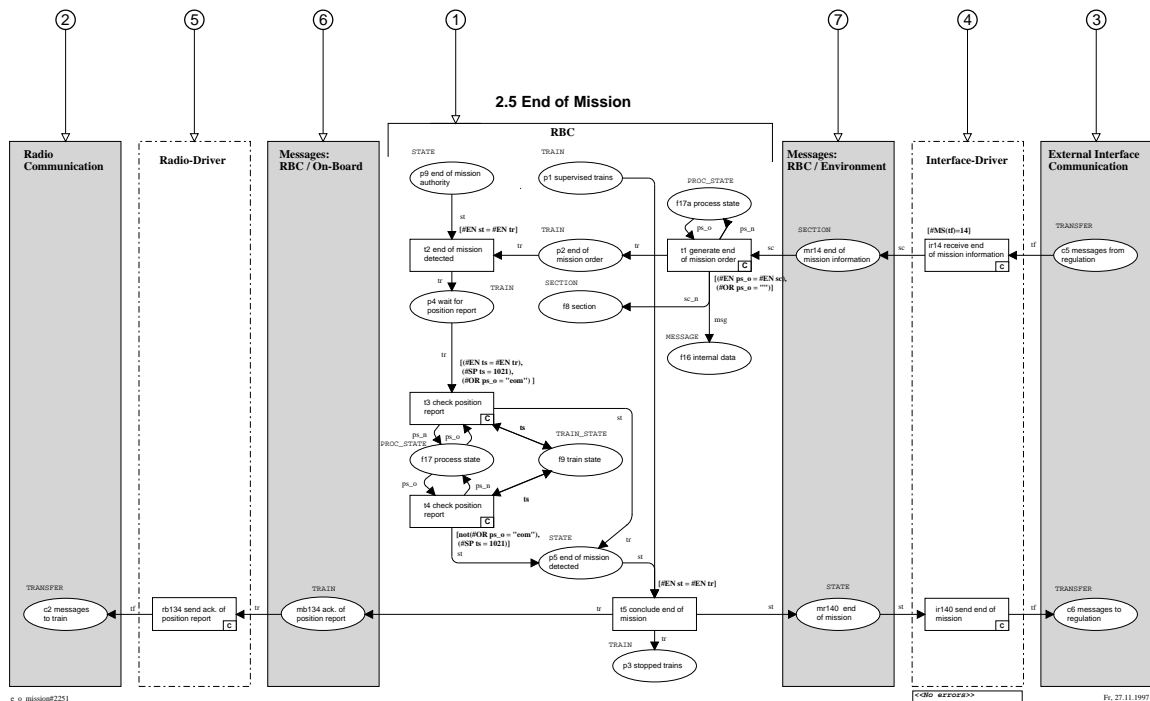


Bild 9: Struktur der Szenario-Netze

- Die Netze der Funktionen-Ebene weisen einen deutlich einfacheren Aufbau auf. Sie stellen nur die Verfeinerung einer Transition dar. Deshalb haben sie keine Schnittstellen nach außen, auch wenn die Transition solche Schnittstellen hat. Die Kommunikation wird von dem übergeordneten Netz der Szenario-Ebene wahrgenommen. Die Logik des Netzes befindet sich innerhalb einer gestrichelten Box (1). Die Ein- und Ausgänge der Transition auf dem übergeordneten Netz befinden sich außerhalb dieser Box (2).

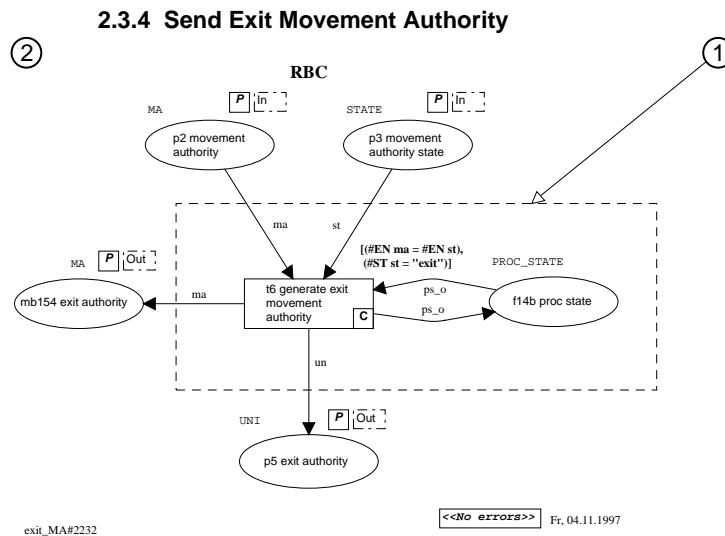


Bild 10: Struktur der Funktions-Netze

3.3 Netzelemente

Innerhalb der Petrinetze befinden sich eine ganze Reihe von verschiedenen Elementen. Die wichtigste Gruppe sind die Elemente der Netze selber. Die Transitionen bilden die Funktionen und Aktivitäten selbst dar. In Bild ??? handelt es sich um die Rechtecke, die mit der Nummer (1) gekennzeichnet sind. Die Transitionen werden durch Plätze (2) verbunden die Informationen und Prozeßzustände tragen. Verfeinerungen werden im Netz durch Instanzen abgebildet, die wie Transitionen dargestellt werden, aber ein abweichendes Schaltverhalten haben können. Um sie von den Transitionen unterscheiden zu können, werden sie mit der Kennung HS (3) versehen. HS steht für hierarchical Subpage. Alle diese Elemente sind durch Kanten (4) verbunden. Wenn eine Transition und ein Platz mit einer Kante sowohl in Hin- als auch in Rückrichtung verbunden sind, wird zwischen zwei Fällen unterschieden. Wenn die Transition den Inhalt der Marke nur liest, aber nicht verändert, handelt es sich um einen rein lesenden Zugriff (5). In diesem Fall werden die beiden Kanten übereinander gelegt. Wenn die Transition den Inhalt der Marke verändert, handelt es sich um einen lesenden und schreibenden Zugriff (6). Hier müssen die Kanten getrennt gelegt werden.

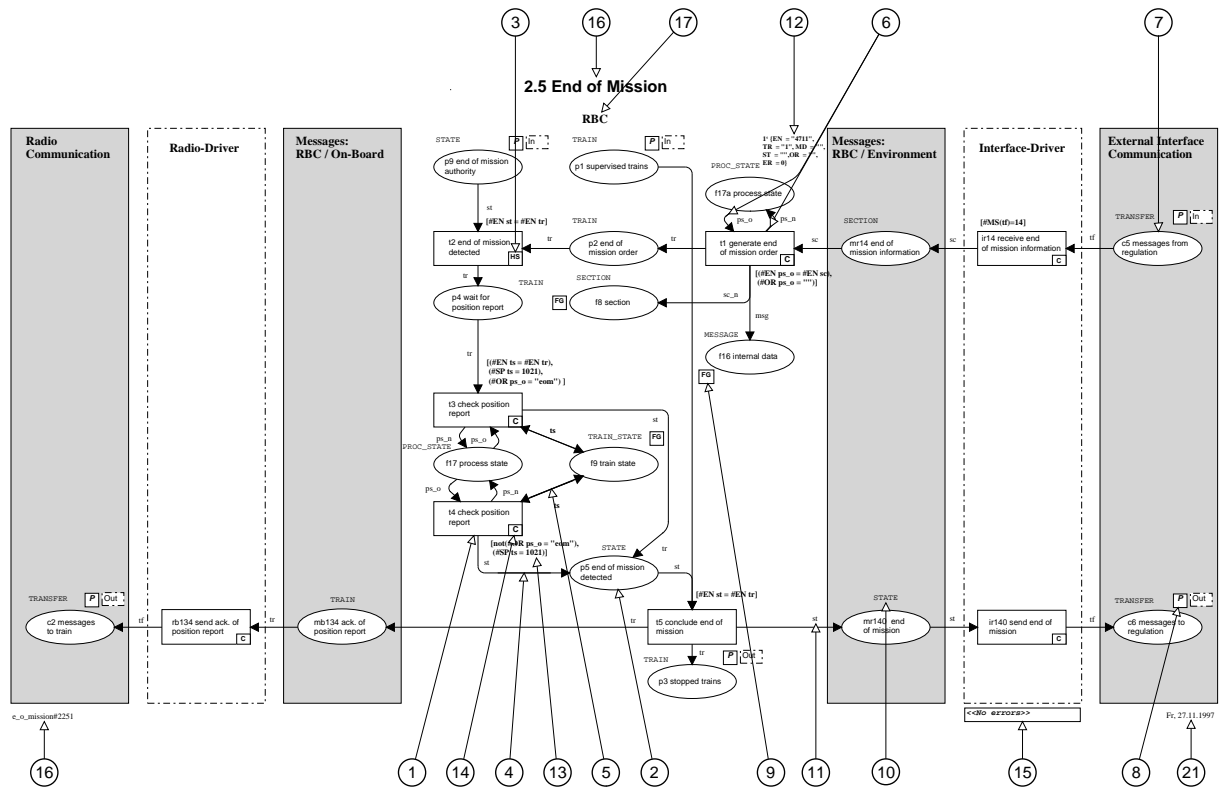


Bild 1: Netzelemente

In hierarchischen Petrinetzen werden die Plätze die eine Verbindung zwischen zwei Netzen bilden auf der unterlagerten Netzebene gekennzeichnet. Es handelt sich um sogenannte Port-Plätze. Diese Plätze werden mit einer entsprechenden Kennzeichnung versehen. Plätze auf denen Marken in das Unternetz hineinkommen werden als Input-Ports (7) bezeichnet; entsprechend heißen Plätze auf denen Marken das Unternetz verlassen Output-Ports (8). Die überall in Netz zugreifbaren Fusion-Plätze tragen, ähnlich wie die Port-Plätze eine Kennzeichnung (9). FG bedeutet, daß es sich um globale, also überall verfügbare, Fusion-Plätze handelt.

Eine Reihe weiterer Elemente kommen durch die Verwendung farbiger Petrinetze hinzu. Den Plätzen müssen definierte Typen zugewiesen werden, die Colorsets (10). Jedem Colorset wird mindestens eine Variable zugewiesen. Diese Variablen dienen der eindeutigen Kennung von Marken, die über Kanten weitergegeben werden (11). Ein Platz kann beim Start des Netzes mit einer initialen Markierung belegt werden (12). Es ist möglich das Schalten einer Transition an eine Bedingung zu knüpfen. Diese Bedingung wird in Form eines sogenannten Guard beschrieben (13). Beim Schalten kann die Transition bestimmte Codes ausführen. Die Anbindung der Codes an die Transition erfolgt mit einer Code-Region (14).

Bei der Ausführung des Syntax-Checks werden in jedem Netz die aufgetretene Fehler angegeben. Für diese Zwecke steht ein besonderer Bereich zur Verfügung (15).

Neben den genannten Elementen der Netze gibt es noch Einträge in Netz, die dem besseren Übersicht und der Versionsverwaltung dienen. Der Name und die Kapitelnummer des Netzes stehen als Überschrift über dem Netz (16). Der Name entspricht der Bezeichnung der Instanz auf dem übergeordneten Netz. Die Kapitelnummer verweist auf das Kapitel in der Dokumentation, in der das Netz beschrieben wird. Das Teilsystem, dessen Logik beschrieben wird, steht über der Applikationslogik und unter der Überschrift (17). Die Blattnummer und die Kurzbezeichnung des Netzes stehen rechts unten, unter dem Netz (18). Mit diesen Informationen kann das Netz auf der Übersichtsseite identifiziert werden. Der Bearbeiter und das Datum der letzten Änderung des Netzes werden auf der anderen Seite des Netzes eingetragen (19).

3.4 Nummernsystem

Die verschiedenen Elemente der Petrinetze sind mit einem Nummernschlüssel gekennzeichnet, um eine eindeutige Identifikation zu ermöglichen:

Die Plätze in den dunkelgrauen Kästen „Radio Communication“ und „External Interface Communication“ tragen den Kennbuchstaben *c*. Sie sind für sämtliche Netze in allen Modellen einheitlich numeriert.

Die Plätze im dunkelgrauen Kasten „Messages: RBC/Onboard“ werden mit dem Buchstaben *m* bezeichnet. Die Nachrichten, die per Funk an den Zug (Radio) weitergegeben werden, tragen die Nummern aus dem Kapitel 8.3.11 der SRS. Die Plätze in den anderen dunkelgrauen Kästen mit der Bezeichnung „Messages: ...“ tragen auch den Kennbuchstaben *m*.

Die Transitionen im Kasten „Radio-Driver“ tragen die Nummer der Nachricht für die sie zuständig sind. Die Nachricht befindet sich im Kasten „Messages: RBC / Onboard“. Die Radio-Driver und Interface-Driver werden allgemein mit dem Buchstaben *r* gekennzeichnet.

Im zentralen Bereich der Netze, in dem sich die Anwendungslogik befindet, werden die Transitionen mit dem Buchstaben *t* und die Plätze mit dem Buchstaben *p* versehen. In jedem Netz werden die Plätze und Transitionen von 1 beginnend neu durchgezählt.

Ein Ausnahme bilden dabei die Fusion-Places, die in mehreren Netzen parallel auftauchen können. Diese haben den Buchstaben *f* und werden jeweils für das RBC-Netz und das Onboard-Netz durchnummeriert.

Die Kennbuchstaben werden also in der folgenden Weise den Bereichen in den Netzen zugeordnet:

| Kenn-Buchst. | Netz-Element: Plätze | Numerierung |
|---------------------|--|--------------------|
| b | Port in einem funktionalen Block, Socket eines funktionalen Blocks | lokal |
| c | Radio Communication und External Interface Communication | global |
| d | MMI (Display) | global |
| f | Fusion-Place | global |
| k | Telegramm - Plätze | global |
| m | Nachrichten (Messages) externer Schnittstellen | global |
| n | Interne Schnittstelle | global |
| p | Plätze der Anwendungslogik, Port-Plätze normaler Unternetze, Sockets normaler Unternetze | lokal |

Tabelle 1: Übersicht der Kennbuchstaben der Plätze.

| Kenn-Buchst. | Netz-Element: Transitionen / Instanzen | Numerierung |
|---------------------|---|--------------------|
| i | Interface-Driver | global |
| o | Funktionaler Block | global |
| t | Transitionen | lokal |
| u | Normales Unternetz | lokal |

Tabelle 2: Übersicht der Kennbuchstaben der Transitionen und Instanzen:

| Kenn-Buchst. | Teilsystem | Erläuterung |
|---------------------|------------------------------------|-----------------------------|
| a | MMI | MMI (Anzeige) |
| b | Onboard | Zuggerät |
| d | Trackside Device | Gleisseitige Einrichtungen |
| f | Track Free Reporting Device | Gleisfreimeldeeinrichtungen |
| g | Diagnostic Recordings | Diagnose Aufzeichnungen |
| i | Interlocking | Stellwerk |
| j | Juridical Recordings | Juristischen Aufzeichnungen |
| k | Train Position Tracking Module | |
| m | Route Map Generator | Streckendatenbank |
| o | Track Occupancy Manager | |
| p | Train Path Supervision | Zuglaufverfolgung (ZLV) |
| r | Regulation | Disposition |
| s | Specific Transmission Module (STM) | |
| t | RBC (Trackside) | |
| u | Train Interface Unit (TIU) | |
| z | Automatic Route Setting | Zuglenkung (ZL) |

Tabelle 3: Übersicht der Kennbuchstaben der Teilsysteme.

| Kenn-Buchst. | Schnittstelle | Erläuterung |
|---------------------|----------------------|--------------------|
| b | Balise | Balise |
| l | Loop | Leiterschleife |
| r | Radio | Funk |

Tabelle 4: Übersicht der Kennbuchstaben der Schnittstellen.

Wenn ein Fusion-Place mehrfach in einem Netz auftritt, tragen die verschiedenen Exemplare des Platzes hinter der Nummer noch einen Buchstaben, der fortlaufend vergeben wird. Ein Sonderfall besteht darin, wenn der Fusion-Place in der Schnittstelle liegt. In diesem Fall trägt er hinter dem *f* den Buchstaben des ETCS-Systems, das mit dieser Schnittstelle angebunden wird.

4 Anhang

4.1 Bibliographie

- [Abel90] Abel, Dirk: Petri - Netze für Ingenieure, Modellbildung und Analyse diskret gesteuerter Systeme. Springer, 1990
- [ARS6.0] EEIG ERTMS Users Group: ERTMS/ETCS Architecture Requirements, Version 6.00, Doc.-Ref. 96E0146, Brüssel, 1997
- [Baumgarten90] Baumgarten, Bernd: Petri-Netze, Grundlagen und Anwendungen, BI-Wissenschaftsverlag, 1990
- [CPN3.0] Design/CPN Reference Manual for X-Windows, Version 2.0, Aarhus 1997
- [FRS4.0] ERRI A200: Functional Requirements Specification, Version 4.00, Utrecht, 1996
- [Jensen92] Jensen, Kurt: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Band 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 1992
- [Jensen94] Jensen, Kurt: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Band 2, Analysis Methods. Monographs in Theoretical Computer Science, Springer-Verlag, 1994
- [Lemmer95] Lemmer, Karsten: Diagnose diskret modellierter Systeme mit Petrinetzen. VDI-Verlag, Düsseldorf, 1995
- [Reisig90] Reisig, Wolfgang: Petrinetze. Springer; 1990
- [Scen2.0] EEIG ERTMS Users Group: Scenarios, Version 2.00, Doc.-Ref. 96E2364, Brüssel, 1997
- [SRS4.0] EEIG ERTMS Users Group: System Requirements Specification, Version 4.00, Doc.-Ref. 96E2364, Brüssel, 1997
- [Schnieder93] Schnieder, Eckehard: Prozeßinformatik. Vieweg, Wiesbaden; 1993
- [Starke90] Starke, Peter H.: Analyse von Petri - Netz- Modellen. Teubner, Stuttgart; 1990

4.2 Glossar

In diesem Glossar werden die deutschen und englischen Begriffe, die in diesem Dokument verwendet werden, einander gegenüber gestellt.

| Deutscher Begriff | Englischer Begriff | Erläuterung | Bemerkung / Ref |
|-------------------------------------|----------------------------|---------------------------------|----------------------|
| Luftschnittstelle | Air Gap | [SRS4.0] | SRS Kap 7 und 8.3 |
| Kante | Arc | [CPN3.0] | Jensen |
| Markentyp | Colorset | [CPN3.0] | |
| Schaltbedingung | Guard | Schaltbedingung [CPN3.0] | |
| Platz | Place | [CPN3.0] | |
| Teilsystem | Subsystem | | |
| Telegramm | Telegram | | |
| Marke | Token | [CPN3.0] | |
| Transition | Transition | [CPN3.0] | |
| Instanz | Instance | Verfeinerte Transition [CPN3.0] | |
| Betriebsart | Operational Mode | | |
| Odometer | Odometer | | |
| Zugstärken | Joining | | |
| Zuschwächen | Splitting | | |
| Rangieren | Shunting | | |
| Infill-Balise | Infill Balise | | |
| Haupt (signal-) Balise | Main Balise | | |
| Positionsmeldung | Position Report | | |
| Bestätigung | Confirmation | | |
| Quittung | Acknowledgement | | |
| (Balisen-)Kopplungs- information | Linking Information | | |
| Fahrzeug-Gerät | Onboard System | | |
| Fahrauftrag (MA) | Movement Authority | | |
| Triebfahrzeugführer | Driver | | |
| Fusion-Platz | Fusion-Place | | |
| Port-Platz | Port-Place | | |
| Luftschnittstelle | Air Gap | | |
| Deklarationsseite | Global Declaration Node | | |

In der Spalte „Erläuterung“ wird in eckigen Klammern angegeben, in welchen Kontext der Begriff gehört:

[CPN3.0] verweist auf einen begriff aus der Petrinetz Theorie hin, der einem Element in der Werkzeug DesignCPN (Version 3.0) entspricht.

[SRS4.0] weist auf einen Begriff hin, der in der System Requirements Specification (Version 4.0) definiert worden ist [vergl. General Glossary 96E399].

4.3 Abkürzungsverzeichnis

| Abk. | Bedeutung | Bemerkung / Kommentar |
|------------------|--|------------------------------|
| ARS | Architecture Requirements | |
| BA | Banking / Nachschieben | Betriebsart lt. [Scen2.0] |
| BTM | Balise Transmission Module | Lt. [FRS4.0] |
| C | Code Region | Lt. [CPN3.0] |
| CPN | Coloured Petri Net (Farbiges Petrinetz) | Lt. [CPN3.0] |
| DB-AG | Deutsche Bahn - AG | |
| DE | Data Entry / Dateneingabe | Betriebsart lt. [Scen2.0] |
| ERTMS | European Rail Traffic Management System | |
| ETCS | European Train Control System | |
| Fdl | Fahrdienstleiter | |
| FFFIS | Form Fit Funtional Interface Specification | Lt. [ARS6.0] |
| FFFS | Form Fit Funtional Specification | Lt. [ARS6.0] |
| FG | Globaler Fusion-Platz | Lt. [CPN3.0] |
| FIS | Functional Interface Specification | Lt. [ARS6.0] |
| FMS | Functional Module Specification | Lt. [ARS6.0] |
| FRS | Functional Requirments Specification | Lt. [ARS6.0] |
| FS | Ferrovie dello Stato/ Nationale italienische Eisenbahngesellschaft | |
| FS | Full Supervision / Vollüberwachung | Betriebsart lt. [Scen2.0] |
| GDN | Global Declaration Node / Deklarationsseite | |
| GSM | Groupe Spéciale Mobile / Global System for Mobile communications | |
| GSM-R | GSM-Rail | |
| IfRA | Institut für Regelungs- und Automatisierungstechnik der Technischen Universität Braunschweig | |
| Int. | Interlocking (Stellwerk) | Lt. [ARS6.0] |
| IS | Isolation / Isoliert | Betriebsart lt. [Scen2.0] |
| MA | Movement Authority (Fahrauftrag) | Lt. [SRS4.0] |
| MMI | Man Machine Interface / Mensch Maschine Interface | Lt. [SRS4.0] |
| NL | Non Leading / Nicht führend, Tandem | Betriebsart lt. [Scen2.0] |
| OS | On Sight / Auf Sicht, Teilüberwachung | Betriebsart lt. [Scen2.0] |
| P _{in} | Input Port-Platz | Lt. [CPN3.0] |
| PO | Power Off / In Ruhe | Betriebsart lt. [Scen2.0] |
| P _{out} | Output Port-Platz | Lt. [CPN3.0] |
| PT | Post Trip / Zwangsgebremst | Betriebsart lt. [Scen2.0] |

| Abk. | Bedeutung | Bemerkung / Kommentar |
|-------------|--|------------------------------|
| RBC | Radio Block Center / Funkblocksystem | |
| Reg. | Regulation (Disposition) | Lt. [ARS6.0] |
| RMG | Route Map Generator | Lt. [ARS6.0] |
| RSDS | Rolling Stock Data System | Lt. [ARS6.0] |
| RT | Real Time | Lt. [ARS6.0] |
| SF | System Failure / Systemstörung | Betriebsart lt. [Scen2.0] |
| SH | Shunting / Rangieren | Betriebsart lt. [Scen2.0] |
| SL | Sleeping / Mehrfachtraktion | Betriebsart lt. [Scen2.0] |
| SNCF | Société Nationale des Chemins de Fer Français / Nationale französische Eisenbahn- gesellschaft | |
| SR | Staff Responsible/ Triebfahrzeugführer- verantwortung | Betriebsart lt. [Scen2.0] |
| SRS | System Requirements Specification | |
| STM | Specific Transmission Module | LT. [SRS4.0] |
| SU | Start Up and Initial Self Test / Aktivierungsselbsttest | Betriebsart lt. [Scen2.0] |
| Tf | Triebfahrzeugführer | |
| Tfz | Triebfahrzeug | |
| TIU | Train Interface Unit / Schnittstelle zum Zug | |
| UN | Unfitted / Nicht ausgerüstet | Betriebsart lt. [Scen2.0] |