

Untersuchung von Service Interaction in Intelligenten Netzen mittels formaler Methoden

Vier Formalisierungen des Basic Call State Model nach CS-2
von ITU und deren Beziehungen zueinander
(Teil 1 und 2)

Diplomarbeit im Studiengang
Informatik
der
Johann Wolfgang Goethe-Universität in Frankfurt am Main

Erstellt von

Astrid Link
Kressenbacher Str. 9
36381 Schlüchtern

und

Armin Noll
Spessartstr. 36
63628 Bad Soden-Salmünster

In Zusammenarbeit mit dem
Institut für Telekooperationstechnik der GMD
Rheinstraße 75
64295 Darmstadt

Betreuer: Prof. Dr. E. Raubold

DANKSAGUNG

Wir danken Prof. Dr. E. Raubold für sein konstruktives Interesse, das der Qualität der Arbeit stets zuträglich war.

Unser besonderer Dank gilt Dr. Peter Ochsenschläger, Ulrich Nitsche und Jürgen Repp von der GMD für ihre großartige Unterstützung und stete Hilfsbereitschaft, die wesentlich zum Gelingen dieser Arbeit beigetragen haben.

Weiterhin danken wir Carla Capellmann und Uwe Herzog, die unsere Arbeit seitens der Deutschen Telekom AG betreuten.

Astrid Link

Armin Noll

Zudem gilt mein Dank meinen Eltern, die mir mein Studium ermöglichten, sowie Armin Noll für die äußerst gute Zusammenarbeit bei der Erstellung dieser Diplomarbeit.

Astrid Link

Außerdem bedanke ich mich besonders bei meinen Eltern, ohne deren Unterstützung mein Studium nicht möglich gewesen wäre, und bei Astrid Link, mit der ich stets sehr gut zusammenarbeiten konnte.

Armin Noll

EIDESSTATTLICHE ERKLÄRUNGEN

Ich erkläre hiermit, daß ich die vorliegende Arbeit nur mit den angegebenen Quellen und Hilfsmitteln erstellt habe.

Breitenbach, den 28. Januar 1997

Astrid Link

Ich erkläre hiermit, daß ich die vorliegende Arbeit nur mit den angegebenen Quellen und Hilfsmitteln erstellt habe.

Hausen, den 28. Januar 1997

Armin Noll

AUFTEILUNG DER DIPLOMARBEIT

Den Kern der Arbeit bilden drei Spezifikationen des Basic Call State Model nach CS-2 von ITU (BCSM CS-2) unter verteilter Sichtweise und eine Spezifikation des BCSM CS-2 unter globaler Sichtweise sowie eine Dienstlogik für die verteilten Modelle und eine Dienstlogik für das globale Modell.

Folgende Teile der Diplomarbeit wurden vornehmlich von Astrid Link angefertigt:

- allgemeine Erläuterungen zur Modellierung des BCSMs (Abschnitt 3.1.1.)
- Dienstlogik zu den drei verteilten Modellen (Abschnitt 3.1.2.)
- verteiltes Modell mit Handshake-Lösung (Abschnitt 3.2.)
- verteiltes Modell ohne strikte Synchronisation (Abschnitt 3.3.)

Folgende Teile der Diplomarbeit wurden vornehmlich von Armin Noll angefertigt:

- optimiertes, verteiltes Modell ohne strikte Synchronisation (Abschnitt 3.4.)
- globales Modell (Abschnitt 4.)
- Dienstlogik zum globalen Modell (Abschnitt 4.3.)

Die Abschnitte *Übersicht*, *Theoretischer Hintergrund*, *Aufgabenstellung* und *Zusammenfassung und Ausblick* wurden gemeinsam erstellt.

INHALTSVERZEICHNIS

ÜBERSICHT	1
1. THEORETISCHER HINTERGRUND	4
1.1. Produktnetze	4
1.1.1. Einleitung	4
1.1.2. Schaltfolgenhomomorphismen	4
1.1.3. Die Produktnetzmaschine und das SH-Verifikation Tool	6
1.2. Das Konzept des Intelligenten Netzes	6
1.2.1. Das konzeptionelle IN-Modell	8
1.2.2. BCSM	9
2. AUFGABENSTELLUNG	10
3. MODELLIERUNGEN UNTER VERTEILTER SICHTWEISE	16
3.1. Allgemeine Erläuterung zur Modellierung der IN-Komponenten	16
3.1.1. Basic Call State Modell (BCSM)	16
3.1.1.1. Points in Call (PICs)	16
3.1.1.2. Detection Points (DPs)	17
3.1.2. Dienstlogik (SCF)	19
3.1.2.1. Service Control Function Input	20
3.1.2.2. Service Control Function Output	21
3.1.2.3. Dienste	21
3.1.2.3.1. Call Hold (CH)	21
3.1.2.3.2. Call Waiting (CW)	23
3.1.2.4. Service Features	26
3.1.2.4.1. Call Hold (CH)	27
3.1.2.4.2. Call Retrieve (CRET)	28
3.1.2.4.3. Call Toggle (CTOG)	29
3.1.2.4.4. Call Waiting (CW)	30
3.2. Erste Spezifikation	32
3.2.1. Aspekte der Modellierung	32
3.2.1.1. Einführung von Prozeß-Ids	32
3.2.1.2. Beschreibung ausgezeichneter Stellen	32
3.2.1.3. Unterscheidung zwischen Auflegen und aufgetretenem Fehler beim Angerufenen	33
3.2.1.4. Sonderbehandlung von O_T_abandon	33
3.2.1.5. Unendlicher Zustandsraum durch Ansammeln von Nachrichten auf den Schnittstellen	33
3.2.2. Analyse und Verifikation	35
3.2.2.1. Anrufer	37
3.2.2.1.1. Anrufer ohne abonnierten Dienst	37
3.2.2.1.2. Anrufer mit abonniertem Call Hold-Dienst	38
3.2.2.2. Angerufener	40
3.2.2.3. O-BCSM	41
3.2.2.4. T-BCSM	44
3.2.2.5. Kommunikation zwischen O- und T-BCSM	46
3.2.2.6. Phasentrennung und Lebendigkeitseigenschaft	48
3.2.2.7. Verhalten bei aktiviertem Call Hold-Dienst	50
3.2.2.8. Besetzt-Verhalten	51
3.2.3. Diskussion der ersten Spezifikation	53
3.3. Zweite Spezifikation	54
3.3.1. Ziel dieser Spezifikation	54

3.3.2. Mögliche Vorgehensweisen	54
3.3.2.1. Verzögerungen beim Nachrichtentransport	54
3.3.2.2. Vermeidung der Synchronisation	55
3.3.3. Konkrete Modellierung	55
3.3.3.1. Alternating Bit-Mechanismus	55
3.3.3.2. Schnittstellen als überschreibbare Puffer	57
3.3.4. Analyse und Verifikation	58
3.3.4.1. Anrufer	58
3.3.4.2. Angerufener	59
3.3.4.3. O-BCSM	59
3.3.4.4. T-BCSM	60
3.3.4.5. Phasentrennung und Lebendigkeitseigenschaft	61
3.3.4.6. Besetzt-Verhalten	62
3.3.4.7. Gesamtverhalten	63
3.3.5. Diskussion der zweiten Spezifikation	64
3.4. Dritte Spezifikation	67
3.4.1. Ziel dieser Spezifikation	67
3.4.2. Vorgehensweise	67
3.4.3. Konkrete Modellierung	67
3.4.4. Analyse und Verifikation	68
3.4.4.1. Anrufer	68
3.4.4.1.1. Anrufer ohne abonnierten Dienst	68
3.4.4.1.2. Anrufer mit abonniertem Call Hold-Dienst	69
3.4.4.2. Angerufener	70
3.4.4.2.1. Angerufener ohne abonnierten Dienst	70
3.4.4.2.2. Angerufener mit abonniertem Call Waiting-Dienst	70
3.4.4.3. O-BCSM	72
3.4.4.4. T-BCSM	73
3.4.4.5. Phasentrennung und Lebendigkeitseigenschaft	73
3.4.4.6. Verhalten bei abonniertem Call Hold-/Call Waiting-Dienst	74
3.4.4.7. Verhalten bei aktiviertem Call Hold-Dienst	76
3.4.4.8. Verhalten bei aktiviertem Call Waiting-Dienst	78
3.4.4.9. Besetzt-Verhalten	79
3.4.4.10. Verbindungsaufbau	79
3.4.4.11. Verbindungsabbau	83
3.4.4.12. Gesamtverhalten	84
4. MODELLIERUNG UNTER GLOBALER SICHTWEISE	86
4.1. Entwicklung eines globalen Modells	86
4.2. Konkrete Modellierung des Basismodells	87
4.2.1. Änderungen der DPs	88
4.3. Anbindung der Dienstlogik	89
4.3.1. Änderungen bezüglich Service Control Function Input	90
4.3.2. Änderungen bezüglich Service Control Function Output	90
4.3.3. Änderungen bezüglich der Modellierung der Service Features	91
4.3.4. Änderungen bezüglich des Rücksprungs zum BCSM	92
4.4. Analyse und Verifikation	93
4.4.1. Verhalten beim Verbindungsaufbau	94
4.4.2. Verhalten beim Verbindungsabbau	97
4.4.3. Anrufer	98
4.4.3.1. Anrufer ohne abonnierten Dienst	98
4.4.3.2. Anrufer mit abonniertem Call Hold-Dienst	99
4.4.4. Angerufener	100
4.4.4.1. Angerufener ohne abonnierten Dienst	100
4.4.4.2. Angerufener mit abonniertem Call Waiting-Dienst	101
4.4.5. Phasentrennung und Lebendigkeitseigenschaft	103
4.4.6. Besetzt-Verhalten	104

4.4.7. Globales Verhalten bei abonniertem Call Hold-Dienst	106
4.4.7.1. Analyse 1	106
4.4.7.2. Analyse 2	107
4.4.7.3. Analyse 3	110
4.4.8. Globales Verhalten bei abonniertem Call Waiting-Dienst	112
4.4.9. Verhalten bei abonniertem Call Hold- und Call Waiting-Dienst	113
ZUSAMMENFASSUNG UND AUSBLICK	117
LITERATURVERZEICHNIS	119
ABKÜRZUNGSVERZEICHNIS	120
ANHANG A	A-1
ANHANG B	B-1
ANHANG C	C-1
ANHANG D	D-1

Übersicht

Die vorliegende Arbeit entstand im Rahmen eines gemeinsamen Forschungsprojektes des Instituts für Telekooperationstechnik der GMD in Darmstadt und des Technologiezentrums der Deutschen Telekom AG in Darmstadt.

Gegenstand der Untersuchung sind die Abläufe beim Auf- und Abbau einer Fernsprechverbindung und die Wechselwirkungen zwischen Diensten im Fernsprechnet (Service Interaction). Dienste wie beispielsweise Anklopfen oder Rückfragen können von einem Teilnehmer bei dem für ihn zuständigen Netzbetreiber abonniert werden. Die steigende Nachfrage seitens der Teilnehmer erforderte ein neues Konzept zur schnelleren und flexibleren Entwicklung, Bereitstellung und Änderung von Diensten. Ein solcher Ansatz ist der des Intelligenten Netzes (IN). Er befindet sich im Telecommunication Standardization Sector der International Telecommunication Union (ITU-T) in der Standardisierungsphase. Im IN wird die grundlegende Vermittlungsfunktion des Fernsprechnetes von dienstspezifischen Abläufen entkoppelt, um die Einführung von neuen Diensten zentralisiert, von Vermittlungsstellen unabhängig möglich zu machen.

Grundlage dieser Arbeit ist der Entwurf von ITU-T zum Standard Q.1224, der auf der zweiten Stufe des achtstufigen IN-Standardisierungsverfahrens angesiedelt ist. Das Hauptaugenmerk liegt auf der Erstellung und Analyse eines Modells für die Abläufe beim Auf- und Abbau einer Fernsprechverbindung, deren Endteilnehmer keine Dienste abonniert haben (Basisanruf).

Die Sichtweise des IN-Konzepts legt den Gedanken nahe, eine Spezifikation des Basisanrufs zu erstellen, die man diensteunabhängig analysieren und auf Sicherheits- und Lebendigkeitseigenschaften hin überprüfen kann. Ausgehend von dieser analysierten und verifizierten Spezifikation des Basismodells können ohne Modifikationen Dienste integriert und im Hinblick auf Service Interaction untersucht werden.

Es hat sich in der Praxis gezeigt, daß eine vollständige Analyse eines solchen Modells sehr umfangreich ist. Daher ist es notwendig, sie mit einem rechnergestützten Werkzeug durchzuführen. Die von der GMD zur Spezifikation und Analyse von verteilten Systemen entwickelte Produktnetzmaschine und das SH-Verifikation Tool, die in dieser Arbeit benutzt werden, sind ein solches Werkzeug.

Im Rahmen dieser Arbeit hat sich ergeben, daß die Kommunikation zwischen den beiden Verbindungsaufbauenden Endsystemen in einigen Fällen dazu führt, daß die Nachrichtenpuffer unkontrolliert gefüllt werden. Ein solches Verhalten unterstellt Puffer unendlicher Größe und verhindert die vollständige Analyse.

In einem ersten Modell besteht die Lösung dieses Problems in der Einführung eines Handshake-Mechanismus. Das bedeutet, daß der Sender eine neue Nachricht erst dann

schicken kann, wenn der Empfänger eine Quittung zur vorherigen Nachricht gesendet hat. Damit wird eine strikte Synchronisation der Endteilnehmer vorausgesetzt und es liegt globale Flußkontrolle vor.

Außerdem wird in diesem Modell davon ausgegangen, daß beim Nachrichtentransport keine zeitlichen Verzögerungen entstehen.

Mit den Analysemöglichkeiten der Produktnetzmaschine und des SH-Verifikation Tools wird gezeigt, daß die Teilnehmer dadurch in ihrer Handlungsfreiheit eingeschränkt sind und die Ergebnisse der Untersuchung von Wechselwirkungen zwischen Diensten verfälscht werden können.

Ein zweites Modell verzichtet auf die strikte Synchronisation der Teilnehmer und die globale Flußkontrolle. Auch die Annahme über das sofortige Vorliegen gesendeter Nachrichten beim Empfänger ist nicht enthalten. Damit es nicht zu einem unkontrollierten Füllen der Nachrichtenpuffer kommt, werden diese als überschreibbare Puffer modelliert.

Ein Alternating Bit-Mechanismus wird benutzt, um Nachrichten zu einer alten Verbindung zwischen zwei Teilnehmern von Nachrichten zu einer neuen Verbindung zwischen denselben Teilnehmern unterscheiden zu können und damit Phasentrennung zu gewährleisten.

An dem jeder Nachricht angehängten Alternating Bit kann ein Teilnehmer erkennen, welche Nachrichten für ihn maßgeblich sind und welche nicht mehr benötigt werden.

Selbst bei Puffern der Größe eins entsteht eine große Zahl an Kombinationen von verschiedenen alten und neuen Nachrichten, die in den Puffern liegen können. Dadurch wird der Zustandsraum des Modells so groß, daß die vorhandenen Rechenkapazitäten eine Analyse mit angekoppelten Diensten nicht erlauben.

Durch Optimierungen beim Nachrichtentransport wird versucht, den Zustandsraum auf ein handhabbares Maß zu reduzieren. Dies führt zu einem dritten Modell.

Es wird von einem intelligenten Transportsystem ausgegangen, das vollständigen Zugriff auf die Nachrichtenpuffer hat. Das bedeutet, daß das Transportsystem Nachrichten an den Empfänger nicht weiterleitet, wenn dieser sich in einem Zustand befindet, in dem er sie nicht mehr benötigt.

Dadurch wird der Zustandsraum zwar erheblich kleiner, für eine Analyse von Diensten mit realistischen Konfigurationen ist er jedoch noch zu groß. Es wird gezeigt, daß eine knappere Modellierung bei den gegebenen Handlungsfreiheiten der Teilnehmer und der gewählten Granularität nicht möglich ist.

Aus diesem Grund wird in einem vierten Modell von der verteilten Sicht auf Anrufer und Angerufenen weitgehend abstrahiert und die Abläufe werden global betrachtet. Dadurch kann auf die explizite Modellierung der Kommunikation verzichtet werden.

Beim Beenden einer Verbindung wird noch zwischen Anrufer und Angerufenem unterschieden, um das unabhängige Agieren der Teilnehmer nicht vollends durch die Abstraktion zu verdecken.

Die geringere Komplexität dieses Modells ermöglicht es, Dienste anzukoppeln und zu analysieren. Dies geschieht exemplarisch mit den Diensten Anklopfen (Call Waiting) und Rückfragen (Call Hold).

Diese Arbeit zeigt, daß von der detaillierten Betrachtungsweise von ITU-T bei der Standardisierung des IN-Konzepts abstrahiert werden muß, um Wechselwirkungen zwischen Diensten mit Produktnetzen zu ermöglichen und zu fundierten Aussagen kommen zu können.

1. Theoretischer Hintergrund

1.1. Produktnetze

1.1.1. Einleitung

Produktnetze wurden bei der GMD in Darmstadt als formales Mittel für die Konzeption, formale Spezifikation und Verifikation von verteilten Systemen entwickelt. Es handelt sich dabei um beschriftete Petrinetze mit individuellen Marken.

Die Ausdrucksstärke der Produktnetze beruht auf der Möglichkeit, komplexe Mengen und Funktionen zu ihrer Beschriftung im sogenannten Vorspann definieren zu können. Des Weiteren werden in Produktnetzen zwei neue Kantenarten gegenüber den klassischen Petrinetzen eingeführt: Verbots- und Abräumkanten. Es ist gezeigt worden, daß Produktnetze (auch ohne Abräumkanten) in ihrer Ausdrucksstärke äquivalent zu Turingmaschinen sind, d.h. den vollständigen Berechenbarkeitsbegriff abdecken.

Dr. P. Ochsenschläger von der GMD entwickelte Analyse- und Verifikationsmethoden ([Oc1, Oc2, Oc3, Oc4, Oc5, Oc6, Oc7]), die zu einem Teil in der Produktnetzmaschine und im SH-Verifikation Tool instrumentalisiert sind und zu einem Teil noch realisiert werden sollen. Dieses Werkzeug unterstützt den Entwurf von Produktnetzen mit Hilfe eines graphischen Editors und ermöglicht sowohl die Analyse des dynamischen Verhaltens von Produktnetzen durch Simulation und vollständige Erreichbarkeitsanalyse als auch die Verifikation von beispielsweise Sicherheits- und Lebendigkeitseigenschaften.

1.1.2. Schaltfolgenhomomorphismen

Der Erreichbarkeitsgraph eines Produktnetzes kann sehr groß werden und ist deshalb zur direkten Betrachtung spezieller Eigenschaften der Dynamik des Netzes im allgemeinen ungeeignet.

Durch die Anwendung von Schaltfolgenhomomorphismen erhält man "vergrößerte Sichtweisen" des dynamischen Verhaltens eines Produktnetzes.

Ein Schaltschritt ist ein Tripel der Form

(Markierung, aktivierte Transition, Nachfolgemarkierung).

Eine Folge von Schaltschritten (M_1, T_1, M_1') , (M_2, T_2, M_2') , ..., (M_k, T_k, M_k') heißt Schaltfolge, wenn für alle $i \in \{1, \dots, k-1\}$ gilt: $M_i' = M_{i+1}$.

Die Menge aller Schaltfolgen, deren Startmarkierung gleich der Anfangsmarkierung des Netzes ist, bildet eine formale Sprache, die man Schaltfolgensprache nennt.

Ein Schaltfolgenhomomorphismus ist eine reihenfolgeerhaltende Abbildung auf der Schaltfolgensprache.

Mittels Schaltfolgenhomomorphismen können Worte, die uninteressante Teile des dynamischen Verhaltens des Netzes repräsentieren, ausgeblendet werden. So lassen sich u.a. Sicherheits- und Lebendigkeitseigenschaften¹ nachweisen.

Sicherheitseigenschaften beziehen sich auf das bereits abgelaufene Verhalten, wohingegen Lebendigkeitseigenschaften sich auf das zukünftige Verhalten beziehen.

Gelten die Sicherheitseigenschaften in der durch einen Schaltfolgenhomomorphismus erzeugten vergrößernde Sichtweise, so kann man diese Eigenschaften direkt auf den zugrundeliegenden Erreichbarkeitsgraphen übertragen.

Dies gilt für die Lebendigkeitseigenschaften nicht. Durch die Anwendung eines Schaltfolgenhomomorphismus kann es dazu kommen, daß im homomorphen Bild ein Zweig des Erreichbarkeitsgraphen einen anderen überdeckt. Repräsentiert nun die überdeckte Verhaltensweise einen Fall, der dazu führt, daß das zukünftige Verhalten eingeschränkt wird und somit die Lebendigkeitseigenschaft nicht mehr gilt, so ist dies unter der vergrößernden Sichtweise nicht mehr zu erkennen (siehe auch [OP]).

Deshalb wurde in [Oc4] der Begriff der "schlichten Homomorphismen" eingeführt. Besitzt eine vergrößernde Sichtweise die Eigenschaft der Schlichkeit, so ist sichergestellt, daß die Lebendigkeitseigenschaften des homomorphen Bildes auf den Erreichbarkeitsgraphen übertragen werden können, wobei man vom approximativen Erfüllen von Eigenschaften ausgeht (siehe auch [NO95]).

Nach [OP] gilt folgender Satz:

Sei L eine Sprache, die von einem endlichen Automaten A erkannt wird, und h ein Homomorphismus auf L . Falls es zu jedem $x \in L$ eine Fortsetzung $y \in L$ gibt, die im Automaten A in eine tote Zusammenhangskomponente führt, so daß jedes $y' \in L$ mit $h(y) = h(y')$ in A ebenfalls in *diese* tote Zusammenhangskomponente führt, dann ist h schlicht auf L .

Die Bedingung des Satzes ist z.B. dann erfüllt, wenn es in jeder toten Zusammenhangskomponente eine Kantenbeschriftung a mit $h(a) \neq \varepsilon$ gibt, so daß es außerhalb dieser toten Zusammenhangskomponente keine Kantenbeschriftung mit dem gleichen homomorphen Bild $h(a)$ gibt.

Besitzt ein Erreichbarkeitsgraph nur eine starke Zusammenhangskomponente, so folgt aus obigem Satz, daß jeder auf diesem Erreichbarkeitsgraph definierte Homomorphismus schlicht ist. Die starken Zusammenhangskomponenten eines Erreichbarkeitsgraphen können von der Produktnetzmaschine berechnet werden.

¹siehe auch [ALP85]

1.1.3. Die Produktnetzmaschine und das SH-Verifikation Tool

Die Produktnetzmaschine und das SH-Verifikation Tool sind von der GMD entwickelte Werkzeuge zur formalen Spezifikation und Analyse von Produktnetzen. Sie bestehen aus

- einem graphischen Editor, der das Zeichnen und Beschriften von Produktnetzen ermöglicht,
- einem Parser, der die syntaktische Korrektheit eines Produktnetzes überprüft und dem Analysator ein syntaktisch korrektes Produktnetz übergibt,
- einem Vorspann, bestehend aus einem Vorspann-Editor zur Eingabe von Mengen- und Funktionsdefinitionen und einem dazugehörigen Parser, der die syntaktische Korrektheit der Definitionen überprüft,
- einem Analysator, der den Erreichbarkeitsgraphen eines Produktnetzes berechnet und einige Hilfsprogramme enthält,
- einem Schaltfolgenhomomorphismus-Editor, der die Definition von Schaltfolgenhomomorphismen ermöglicht und aus dem vorher berechneten Erreichbarkeitsgraphen den sogenannten Minimalautomat, das homomorphe Bild, berechnet (Untersuchung auf Schlichtheit wird unterstützt) sowie
- einem Modul zur Verwaltung von Projekten mit mehreren Teilnetzen.

Zur Spezifikation von Produktnetzen und deren Analyse werden in dieser Arbeit ausschließlich die Produktnetzmaschine und das SH-Verifikation Tool benutzt.

1.2. Das Konzept des Intelligenten Netzes

Durch die Forderung des Marktes im Bereich Telekommunikation nach immer schnellerer Bereitstellung von immer flexibleren Diensten wurde ein erweiterter Ansatz zur Dienstebereitstellung erforderlich, der es ermöglicht, neue Dienste schnell, wirtschaftlich und möglichst unabhängig von den Herstellern der Netzkomponenten anzubieten. Der sich herauskristallisierende Lösungsansatz wird heute weltweit als *Intelligentes Netz* (IN) bezeichnet.

Das IN ist ein Konzept für eine Fernsprechnetzaarchitektur, die sich durch verteilte Rufbearbeitung und modulare Dienstbausteine auszeichnet. Diese besteht aus einer funktionalen und einer physischen Architektur. Im weiteren wird nur auf die funktionale Architektur eingegangen, da diese den Rahmen für diese Arbeit bildet. Eine detailliertere Beschreibung des IN-Konzepts findet sich in [Di95].

Im IN-Konzept enthalten ist die zentralisierte Verwaltung von Daten und Programmen, die für das Anbieten von Diensten im weiteren Sinne benötigt werden. Dadurch ist das Hinzufügen

neuer Dienste bzw. die Änderung bereits bestehender Dienste mit geringen Eingriffen in die Software bestehender Vermittlungsstellen möglich.

Um diesen Ansatz realisieren zu können, mußte nach Wegen gesucht werden, die grundlegenden Vermittlungsfunktionen von den dienstspezifischen Abläufen abzukoppeln. Dazu wurden in der funktionalen Architektur die folgenden Funktionen unterschieden:

- CCAF Call Control Agent Function (Netzzugangsfunktion)
- CCF Call Control Function (grundlegende Vermittlungsfunktion)
- SSF Service Switching Function (Dienstzugriffsfunktion)
- SCF Service Control Function (Dienststeuerungsfunktion)
- SDF Service Data Function (Dienstdatenbankfunktion)
- SRF Service Resource Function (Funktion für besondere Ressourcen, z.B. für Ansagen und Spracherkennung)
- SMF Service Management Function (Dienstmanagementfunktion)
- SMAF Service Management Access Function (Dienstmanagementzugriffsfunktion)

Diese Funktionen können in zwei Gruppen eingeteilt werden, wobei eine die grundlegende Vermittlungsfunktionen und die andere die Funktionen zur Dienststeuerung enthält. Grundlegende Vermittlungsfunktionen sind CCAF und CCF. Die Funktionen SCF, SDF, SRF, SMAF und SMF sind zur Dienststeuerung erforderlich. Die SSF übernimmt die Schnittstellenfunktion zwischen beiden Gruppen.

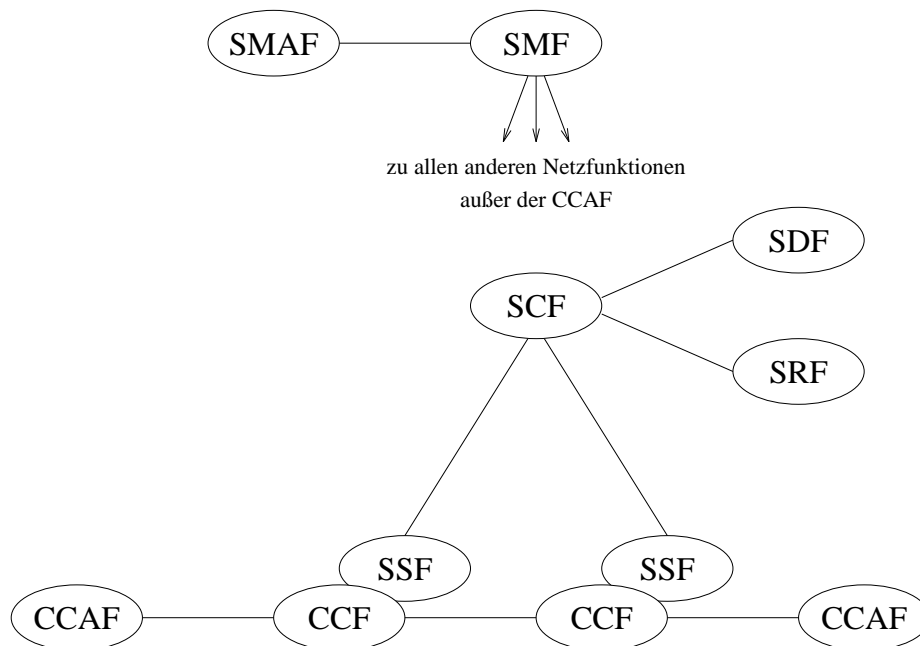


Abb.1: Funktionale IN-Architektur

Das IN-Konzept befindet sich zur Zeit noch in der Standardisierungsphase. Ein Interesse an der Standardisierung besteht in erster Linie seitens der IN-Netzbetreiber. Durch die Möglichkeit zur Kombination von IN-Komponenten verschiedener Hersteller soll eine größere Unabhängigkeit gegenüber den Herstellerfirmen erreicht werden. Gleichzeitig bietet sich die Möglichkeit, Dienste netzübergreifend und damit auch weltweit anbieten zu können. Für den Dienstonutzer bringt die Standardisierung von IN-Diensten den Vorteil, daß die Abläufe zur Benutzung dieser Dienste in allen Netzen gleich sind und damit die Nutzung erleichtert wird.

Die Standardisierung erfolgt bei ITU-T. Die konkrete Definition erfolgt in acht Phasen. Man beginnt mit einer relativ einfachen Struktur mit geringen Fähigkeiten und einigen wenigen Diensten. Die Struktur wird dann in späteren Phasen weiterentwickelt. Das hat den Vorteil, daß Standards schneller verabschiedet werden können. Gleichzeitig kann damit flexibel auf neue technische Möglichkeiten und einen veränderten Bedarf an Diensten reagiert werden. Die Ergebnisse der einzelnen Phasen werden in Capability Sets (CS) zusammengefaßt. Als Standard existiert bisher lediglich CS-1. CS-2 stand bei der Erstellung dieser Arbeit als Entwurf zur Verfügung.

1.2.1. Das konzeptionelle IN-Modell

Das konzeptionelle IN-Modell (IN Conceptual Model, INCM) dient als Rahmen für die Gestaltung und Beschreibung der IN-Architektur. Das INCM besteht aus vier Ebenen. Jede Ebene bietet eine andere abstrakte Sicht auf die Möglichkeiten bzw. Funktionen, die ein IN bereitstellt, und zwar bezüglich der Dienstaspekte, der globalen Funktionalität, der verteilten Funktionalität und der physikalischen Aspekte. Entsprechend dieser Sichtweisen werden die vier Ebenen bezeichnet als

- Service Plane (SP),
- Global Functional Plane (GFP),
- Distributed Functional Plane (DFP) und
- Physical Plane (PP).

Die *Service Plane* bietet eine ausschließlich dienstorientierte Sicht. Ein Dienst (Service) wird aus einem oder mehreren Dienstmerkmalen (Service Features) zusammengesetzt. Ein Service Feature ist das kleinste Element in der Service Plane. Von der Realisierung des Dienstes im Netz wird vollständig abstrahiert. Man beschränkt sich auf die Beschreibung des Netzverhaltens in der Art und Weise, wie sie der Dienstonutzer wahrnimmt.

Die *Global Functional Plane* modelliert das IN aus einer globalen Sicht. Das IN wird als eine einzige Instanz betrachtet, unter Abstraktion von seiner funktionalen und räumlichen Verteiltheit.

In der *Distributed Functional Plane* wird die funktionale Verteiltheit einbezogen. Hier werden auszuführende Aktionen auf funktionale Module abgebildet. Der Basisanruf wird durch ein Modell-Paar, das als Basic Call State Model (BCSM) bezeichnet wird, realisiert. Es besteht aus Ursprungs- (Originating-) und Ziel- (Terminating-) BCSM. Das O-BCSM modelliert den Vermittlungsvorgang in der Vermittlungsstelle des rufenden Teilnehmers, das T-BCSM den in der Vermittlungsstelle des gerufenen Teilnehmers.

Die *Physical Plane* betrachtet die physikalischen Aspekte eines IN-strukturierten Netzes. In ihr wird definiert, wie die funktionalen Module räumlich verteilt werden.

1.2.2. BCSM

Das BCSM umfaßt weitestgehend die Call Control Function (CCF) und die Service Switching Function (SSF). Es ist ein endlicher Automat, mit dem die Vermittlungsvorgänge des Basisnetzes so beschrieben werden, wie sie von der IN-Logik gesehen werden. Die einzelnen Zustände im BCSM, die als Point in Call (PIC) bezeichnet werden, modellieren den Verlauf eines Anrufs vom Abheben bis zum Auflegen des Hörers. Für Anrufer und Angerufenen existieren separate Modelle (O-BCSM, T-BCSM), die miteinander kommunizieren. Zusätzlich zu den PICs existieren im BCSM Stellen, bei deren Erreichen die Dienstelogik aufgerufen werden kann. Diese Stellen werden als Detection Points (DPs) bezeichnet. Nur an diesen kann die Rufbehandlung unterbrochen und die Dienstelogik aufgerufen werden. Ebenso findet der Rücksprung ins BCSM immer zu einem DP statt, wobei sich die Auswahl des DPs nach dem ausgeführten Dienst richtet.

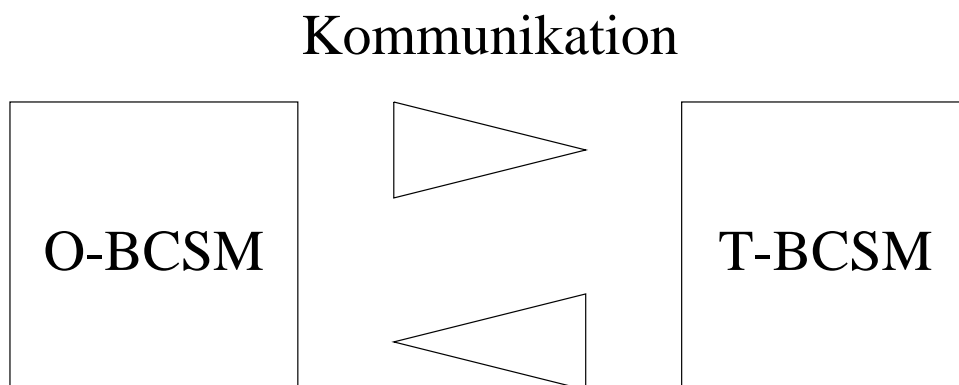


Abb. 2: Die Struktur des BCSMs

2. Aufgabenstellung

Das Ziel dieser Arbeit ist es, das ITU-T-Modell eines Basisanrufs (Basic Call State Model - BCSM) in der Standardisierungsstufe CS-2 zu spezifizieren und zu verifizieren. Exemplarisch sollen zwei CS-2-spezifische Dienste integriert und auf Service Interaction hin untersucht werden.

Die Grundlage der Modellierung des BCSM ist [Q.1224] sowie das bereits bestehende Produktnetzmodell der Standardisierungsstufe CS-1, zu finden in [CS-1]. Zur Dienstmodellierung steht [Q.1221] zur Verfügung.

Um die Unterschiede zwischen BCSM (CS-1) und BCSM (CS-2) zu verdeutlichen, werden im folgenden O- und T-BCSM beider Standardisierungsstufen gegenübergestellt.

O-BCSM

In den Abbildungen 3 und 4 sind die Zustandsübergangsdigramme der beiden O-BCSMs aus [Q.1214] und [Q.1224] zu sehen. Startzustand ist beim O-BCSM (CS-1) *O_Null & Authorize_Origination_Attempt* und beim O-BCSM (CS-2) *O_Null*.

Änderungen bezüglich der Points in Call (PICs):

- PIC *O_Null & Authorize_Origination_Attempt* wurde aufgespalten in PIC *O_Null* (Grundzustand) und PIC *Authorize_Origination_Attempt* (Autorisierungsphase)
- PIC *Routing & Alerting* wurde aufgespalten in die PICs *Select_Route* (Route auswählen), *Authorize_Call_Setup* (Berechtigung für den speziellen Anruf überprüfen), *Send_Call* (Verbindungsaufbauwunsch an T-BCSM schicken) und *O_Alerting* (nachdem ein Freizeichen erhalten wurde, auf Reaktion von T-BCSM warten)
- neuer PIC *O_Suspended* (bei Auflegen des Angerufenen kommt das Token der Verbindung im O-BCSM in diesen PIC, wenn vom T-BCSM innerhalb einer vorgegebenen Zeit das Reanswer-Primitiv geschickt wird, wird die Verbindung wieder aufgenommen und beide treten erneut in die aktive Phase ein)

Neue Detection Points (DPs):

- *Origination_Attempt DP*
- *O_Term_Seized DP*
- *O_Suspend DP*
- *O_Re_answer DP*

Neue Transitionen:

Außer den Transitionen, die zum Einbinden der neuen PICs und DPs benötigt werden, sind folgende Transitionen hinzugekommen:

Transition	von	nach
route_busy	Select_Route PIC	Analyse_Information PIC
route_failure	Send_Call PIC	Select_Route PIC
"	O_Alerting PIC	"
	Send_Call PIC	O_Mid_Call DP
	O_Alerting PIC	"

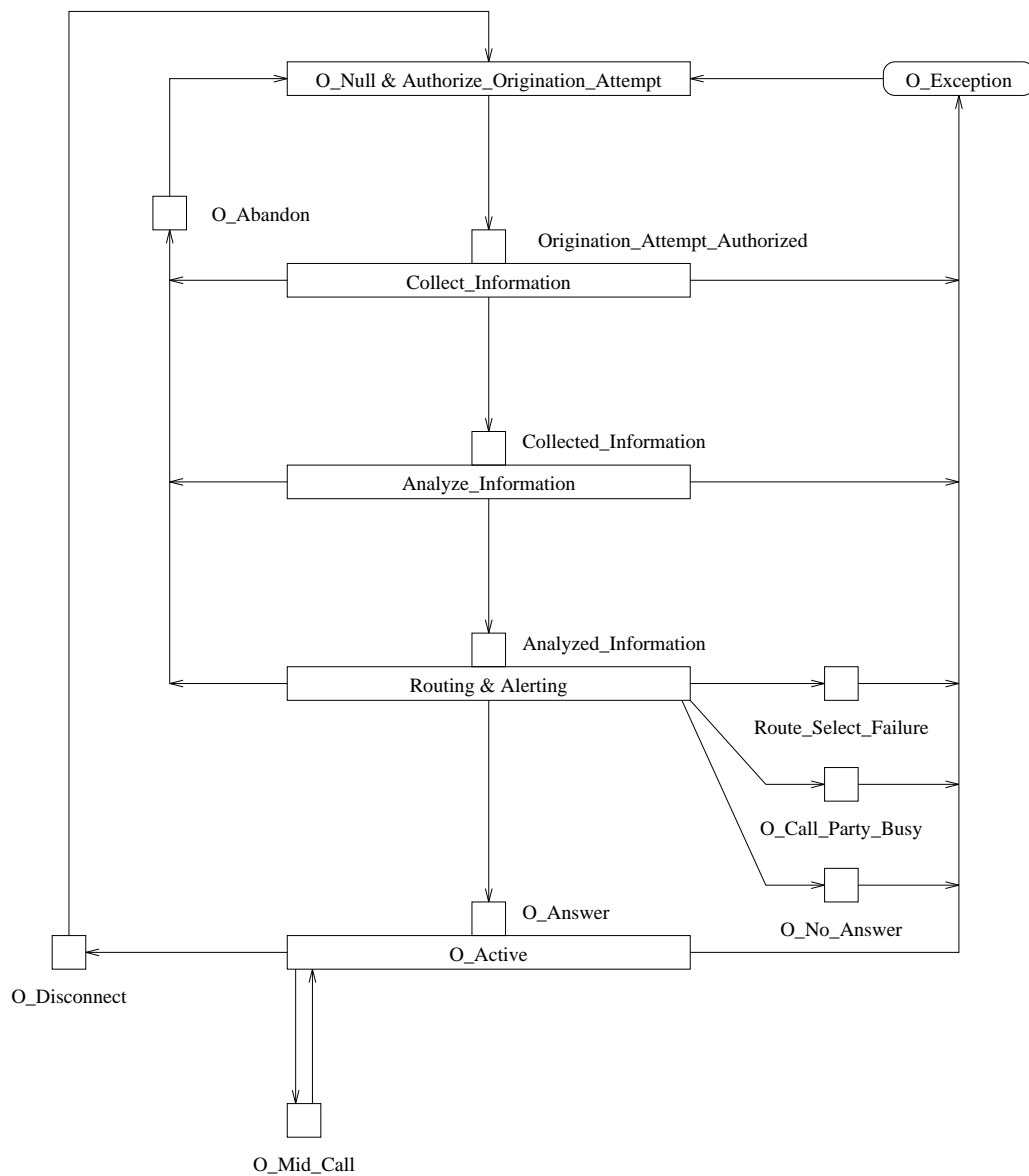


Abb. 3: O-BCSM (CS-1)

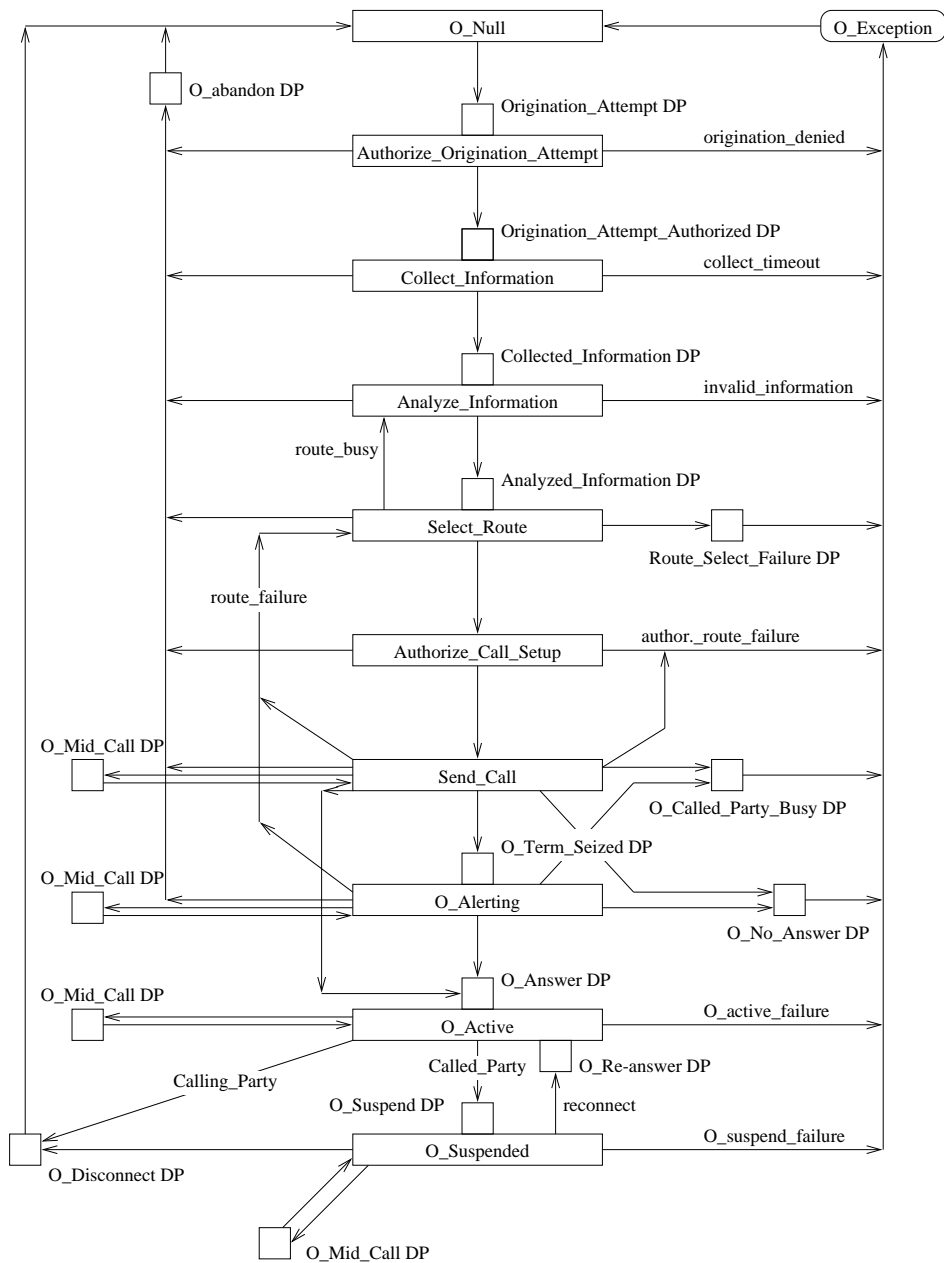


Abb. 4: O-BCSM (CS-2)

T-BCSM

In den Abbildungen 5 und 6 sind die Zustandsübergangsdiagramme der beiden T-BCSMs aus [Q.1214] und [Q.1224] zu sehen. Startzustand ist beim T-BCSM (CS-1) *T_Null* & *Authorize_Termination_Attempt* und beim T-BCSM (CS-2) *T_Null*.

Änderungen bezüglich der Points in Call (PICs):

- PIC *T_Null* & *Authorize_Termination_Attempt* wurde aufgespalten in PIC *T_Null* (Grundzustand) und PIC *Authorize_Termination_Attempt* (Autorisierungsphase)
- PIC *Select_Facility* & *Present_Call* wurde aufgespalten in PIC *Select_Facility* (Prüfen, ob Angerufener besetzt ist) und PIC *Present_Call* (Unterrichten des Angerufenen von eingehendem Anruf)
- neuer PIC *T_Suspended* (Falls der Angerufene auflegt, kommt das Token des Anrufs im T-BCSM in diesen PIC. Der Angerufene hat die Möglichkeit, innerhalb einer vorgegebenen Zeitspanne den Anruf wieder aufzunehmen.)

Neue Detection Points (DPs):

- *T_Abandon DP* (Dieser DP ist bereits im CS-1-Standard enthalten, im CS-1-Produktnetzmodell jedoch noch nicht realisiert.)
- *Call_Accepted DP*
- *T_Suspend DP*
- *T_Re_answer DP*
- *Termination_Attempt DP*
- *Facility_Selected_and_Available DP*

Neue Transitionen:

Außer den Transitionen, die zum Einbinden der neuen PICs und DPs notwendig sind, sind folgende Transitionen hinzugekommen:

Transition	von	nach
SS7_failure	Present_Call PIC	Select_Facility PIC
presentation_failure	Present_Call PIC	T_Exception PIC
call_rejected	T_Alerting PIC	T_Exception PIC
abandon	Authorize_Termination_Attempt PIC	T_Abandon DP
abandon	Select_Facility PIC	T_Abandon DP
abandon	Present_Call PIC	T_Abandon DP
abandon	T_Alerting PIC	T_Abandon DP

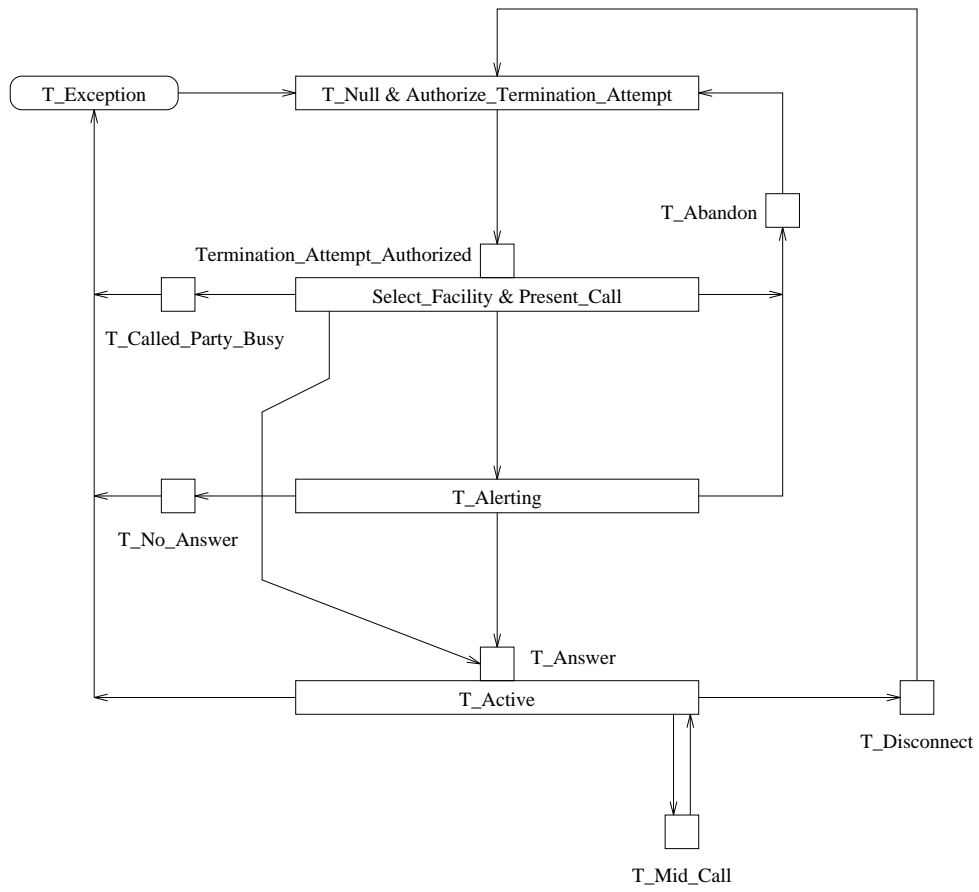


Abb. 5: T-BCSM (CS-1)

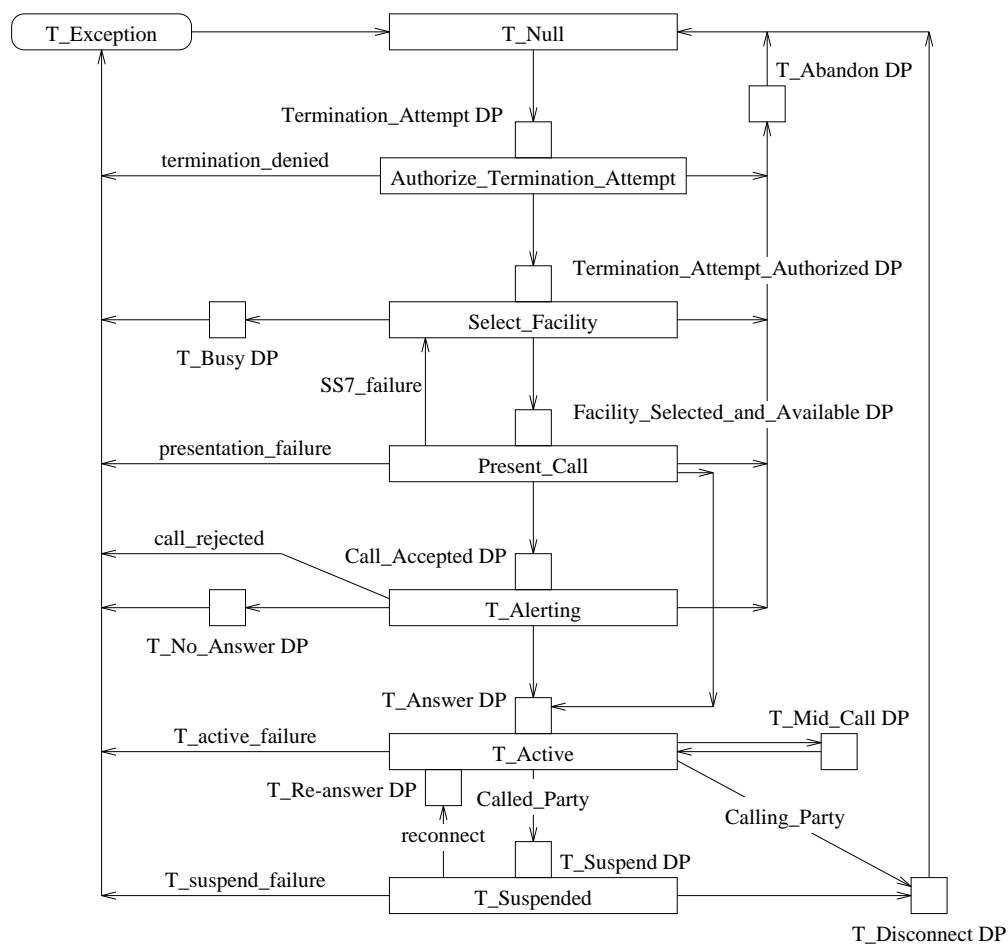


Abb. 6: T-BCSM (CS-2)

Bemerkung:

Die Transitionen vom *O_Alerting PIC* nach *O_No_Answer DP* bzw. *O_Called_Party_Busy DP* werden in dieser Arbeit nicht mitmodelliert. Der Grund dafür ist, daß sich der Angerufene im T-BCSM nach dem Schicken von *T_O_alerted* im *T_Answer DP* befindet und es von dort keine Transition zum *T_No_Answer DP* bzw. *T_Called_Party_Busy DP* gibt. Das wäre aber notwendig, damit von dort *T_O_no_answer* bzw. *T_O_busy* geschickt werden könnte. Daher kann im O-BCSM nach dem Empfangen von *T_O_alerted* kein *T_O_no_answer* oder *T_O_busy* mehr erhalten werden, was aber für den Übergang zu *O_No_Answer DP* bzw. *O_Called_Party_Busy DP* notwendig wäre.

3. Modellierungen unter verteilter Sichtweise

Wie in der Übersicht erläutert, werden in dieser Arbeit drei Modelle des BCSMs unter verteilter Sichtweise vorgestellt. Im Abschnitt 3.1. wird erläutert, wie die Komponenten des BCSMs (PICs und DPs) und die SCF in diesen drei Spezifikationen - basierend auf [Q.1221] und [Q.1224] - modelliert werden. Aspekte, die in der Beschreibung des BCSMs in [Q.1224] nicht berücksichtigt sind, werden in den drei Modellen unterschiedlich umgesetzt. In den Abschnitten 3.2. bis 3.4. werden diese Besonderheiten der einzelnen Spezifikationen und die Analyse der Modelle beschrieben. Die Vorspanne und die Produktnetze der drei Spezifikationen befinden sich in den Anhängen A bis C.

3.1. Allgemeine Erläuterung zur Modellierung der IN-Komponenten

3.1.1. Basic Call State Modell (BCSM)

3.1.1.1. Points in Call (PICs)

Die Points in Call sind funktionale Einheiten. Es gibt sogenannte Entry Events, deren Eintreten notwendig für die Berechnung der Funktion innerhalb des PICs ist. Das Eintreten eines Entry Events wird durch eine Marke auf der Stelle $Entry_PIC_y$, $y \in \{1, 1b, 2, 3, 4, 4b, 4c, 4d, 5, 5b, 6, 7, 7b, 8, 8b, 9, 10, 10b, 11\}$, angezeigt. In Abhängigkeit vom berechneten Funktionswert treten bestimmte Exit Events auf. Jedes Exit Event wird durch eine Transition realisiert. In Abhängigkeit vom Funktionswert (Schalten der Transition, die dieser Funktionswert bedingt) wird eine Marke auf den folgenden PIC oder DP gelegt. Das Ablegen einer Marke auf der Stelle $Entry_x_y$ wobei x angibt, ob es sich um einen PIC oder einen DP handelt, und y die entsprechende Nummer des PICs bzw. DPs angibt ($(x,y) \in \{PIC\} \times \{1, 1b, 2, 3, 4, 4b, 4c, 4d, 5, 5b, 6, 7, 7b, 8, 8b, 9, 10, 10b, 11\} \cup \{DP\} \times \{1, 1b, 2, \dots, 26\}$), entspricht einem Entry Event des entsprechenden PICs bzw. DPs.

Im Standard ist nicht angegeben, was in einem PIC passiert, d.h., wie die Funktionen definiert sind, die die Werte berechnen. Dementsprechend wird die Funktionsberechnung nicht modelliert. Auf der Grundlage des entworfenen Modells können jedoch alle möglichen Resultate der Funktionen betrachtet werden. Der Erreichbarkeitsgraph, der in Abhängigkeit von einer Anfangsmarkierung und vom Modell berechnet wird, enthält implizit alle möglichen Folgen von Funktionswerten. Diese Folgen können durch einen endlichen Automaten dargestellt werden. Diesen Automaten kann man durch die Anwendung eines geeigneten Homomorphismus auf den Erreichbarkeitsgraphen (bzw. auf die Schaltfolgensprache, die der Graph repräsentiert) erhalten. Soll eine bestimmte Folge von Funktionswerten betrachtet werden, so genügt eine Betrachtung des entsprechenden Pfades im Automaten. In jedem Zustand, der mehrere Ausgangskanten besitzt, kann nichtdeterministisch ein nächster Funktionswert ausgewählt werden.

Folgende Tabelle enthält eine kurze Beschreibung der Funktionalität der einzelnen PICs:

<i>PIC</i>	<i>Nr.</i>	<i>Kurzbeschreibung der Funktionalität</i>
O_Null	1	Nullphase im O-BCSM, kein Anruf existiert
Authorize_Origination_Attempt	1b	Autorisierungsphase des Anrufers
Collect_Information	2	Information sammeln (z.B. gewählte Nummer)
Analyze_Information	3	Analysieren der Information
Select_Route	4	Auswählen der Route
Authorize_Call_Setup	4b	Autorisierungsphase für das Absetzen eines Anrufs
Send_Call	4c	Absetzen des Anrufwunsches
O_Alerting	4d	Empfangen des Freizeichens
O_Active	5	Aktive Phase
O_Suspended	5b	Angerufener hat aufgelegt, die Verbindung bleibt aber noch für einen fest vorgegebenen Zeitraum bestehen und kann vom Angerufenen wieder aufgenommen werden
O_Exception	6	Fehlerbehandlung im O-BCSM
T_Null	7	Nullphase im T-BCSM (kein Anruf existiert)
Authorize_Termination_Attempt	7b	Autorisierung zur Weiterleitung eines eingehenden Anrufs an den Angerufenen
Select_Facility	8	Bewertung des Zustands des Angerufenen (besetzt, frei)
Present_Call	8b	Klingeln beim Angerufenen
T_Alerting	9	Übermitteln des Freizeichens an den Anrufer
T_Active	10	Aktive Phase
T_Suspended	10b	Angerufener hat aufgelegt, die Verbindung bleibt aber noch für einen fest vorgegebenen Zeitraum bestehen und kann wieder aufgenommen werden
T_Exception	11	Fehlerbehandlung im T-BCSM

3.1.1.2. Detection Points (DPs)

Die Detection Points modellieren die Service Switching Function (SSF) und damit den Übergang vom BCSM (Call Control Function, CCF) zur Dienstlogik (Service Control Function, SCF). Nur in diesen Punkten ist der Sprung vom BCSM zur SCF gestattet. Umgekehrt kann von der SCF nur zu einem DP des BCSM zurückgesprungen werden. Die einzige Ausnahme von dieser Regelung bilden die PICs *O_Exception* und *T_Exception*. Auch dort kann zur Dienstlogik gesprungen werden, da die Dienstlogik informiert werden muß, falls eine Störung auftritt.

Es bestehen zwei unterschiedliche Mechanismen, die einen Sprung vom BCSM zur SCF ermöglichen. Zum einen gibt es die *statische Armierung*. Hierbei ist im BCSM ein Detection Point für einen Teilnehmer fest armiert. Sobald ein Teilnehmer aktiv oder passiv an einem Anruf beteiligt ist, ist dieser Detection Point automatisch armiert, und es findet ein Sprung zur Dienstlogik statt. Der andere Mechanismus ist die *dynamische Armierung*. Diese Armierung kann nur in der Dienstlogik gesetzt werden. Ein Detection Point wird dann nur für den jeweiligen Anruf armiert. Die Liste der dynamisch armierten Detection Points wird im BCSM ausgewertet, d.h. für den weiteren Ablauf des Gesprächs erfolgt in den dynamisch armierten DPs ein Sprung zur Dienstlogik.

Die Umsetzung dieser Funktionalitäten eines Detection Points wird im folgenden am Beispiel des Detection Point 3 erläutert.

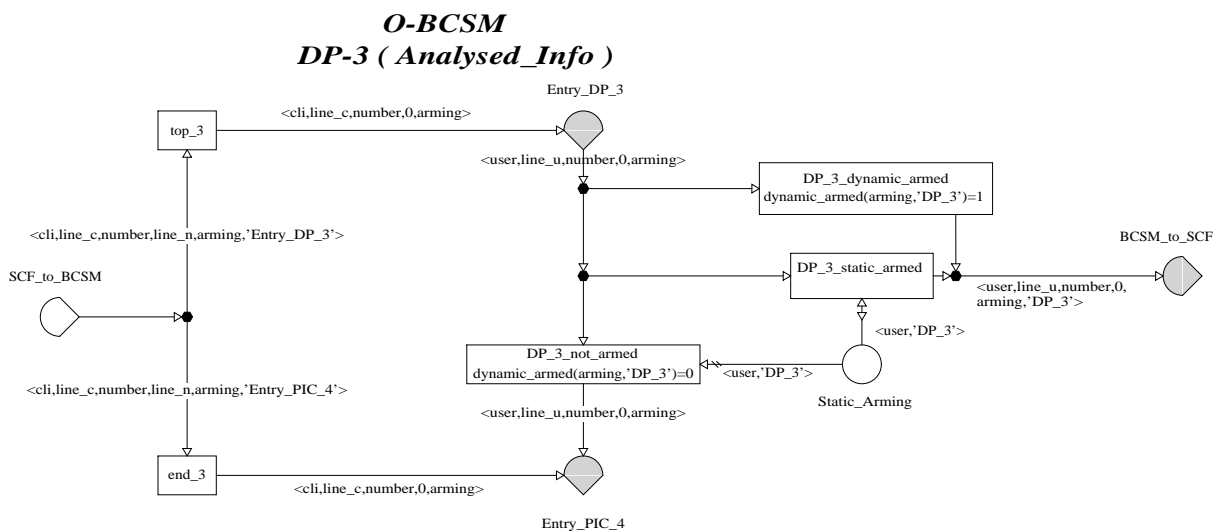


Abb. 7: Detection Point 3

Die Transitionen *DP_3_dynamic_armed* und *DP_3_static_armed* sind für den Sprung vom Detection Point bzw. vom BCSM zur Dienstlogik verantwortlich. Sie repräsentieren die Mechanismen für die statische und dynamische Armierung.

Die Transitionen für den Armierungsmechanismus:

- *DP_3_static_armed*
Diese Transition ist für die statische Armierung zuständig. Es wird anhand der Anfangsmarkierung der Stelle *Static_Arming* erkannt, ob ein bestimmter Detection Point für einen Teilnehmer bzw. Abonnenten statisch armiert ist. Aufgrund der Abfrage dieser Stelle wird entschieden, ob zur Dienstlogik gesprungen wird oder nicht.
- *DP_3_dynamic_armed*
Diese Transition schaltet, wenn dieser Detection Point für eine Verbindung dynamisch armiert ist. Ob das zutrifft, wird mittels des Funktionsaufrufs

$dynamic_armed(arming, 'DP_3')=1$ entschieden. Nach diesem Prinzip wird in jedem DP ermittelt, ob dieser für die jeweilige Verbindung dynamisch armiert ist.

Der Sprung zur Dienstelogik erfolgt in jedem Detection Point über die Stelle *BCSM_to_SCF*. Falls keine der oben genannten Transitionen aktiviert ist, ist dieser Detection Point für diesen Anruf nicht armiert. Der Anruf wird dann mit der Transition *DP_3_not_armed* im BCSM fortgesetzt.

Die Transitionen für den Rücksprung-Mechanismus:

Die beiden Transitionen *top_3* und *end_3* sind für den Rücksprung ins BCSM zuständig. Nachdem die Dienstelogik ihre modifizierte Nachricht an das BCSM gesendet hat, wird in dem jeweiligen Detection Point anhand der letzten Komponente der Nachricht entschieden, ob man an seinen Anfang oder an sein Ende springt. Diese Entscheidung ist vom jeweiligen Dienst abhängig und muß bei der Modellierung von Dienst zu Dienst neu getroffen werden.

3.1.2. Dienstelogik (SCF)

Abb. 8 zeigt schematisch die Struktur der Dienstelogik und das Zusammenspiel von BCSM und Dienstelogik.

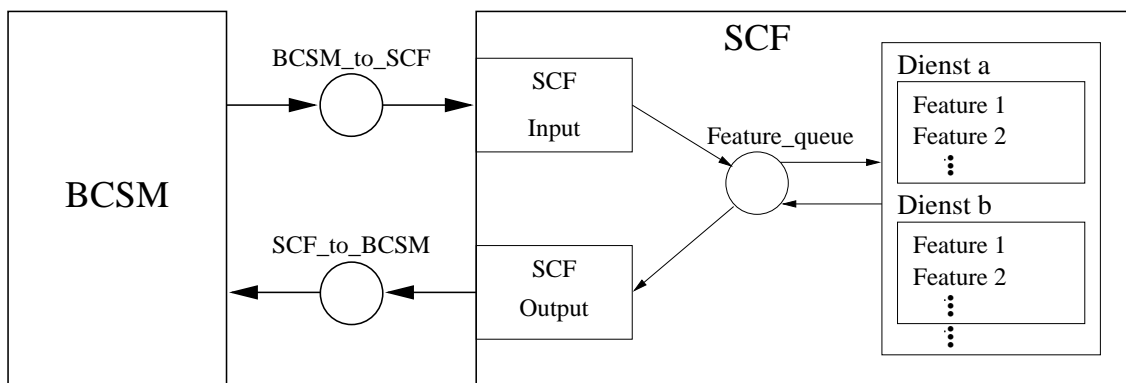


Abb. 8: Struktur der SCF und Anbindung an das BCSM

Die Dienstelogik führt den Dienst aus, der zu einem Dienstewunsch gehört, der vom BCSM an die SCF gesendet wird. Es ist die Aufgabe der SCF, diesen Dienst bzw. diese Nachricht genauer zu analysieren. Dies bedeutet zum einen, daß die Liste der Service Features gelesen wird, aus denen sich der Dienst zusammensetzt, zum anderen, daß genau diese Service Features abgearbeitet werden. Zur Bearbeitung der Dienste sind oftmals dienstspezifische Daten notwendig. Diese für den Dienst bzw. die Service Features notwendigen Daten befinden sich ebenfalls in der Dienstelogik.

Die Modellierung der Dienstlogik besteht aus mehreren Komponenten, die im folgenden beschrieben werden.

3.1.2.1. Service Control Function Input

Abbildung 9 zeigt das Produktnetz der Service Control Function Input. In der Stelle *Service_structure* sind pro Dienstnutzer die von ihm abonnierten Dienste mit den dazugehörigen Service Features und den entsprechenden Ein- und Ausprungadressen bezüglich des BCSMs gespeichert. Die Komponenten der Marken dieser Stelle sind folgende:

- Dienstnutzer
- Service
- Service Access Code
- Service Feature-Liste
- dynamische Armierungsliste
- Einsprungspunkt
- Rücksprungspunkt

Erhält die SCF eine Nachricht vom BCSM - Token auf der Stelle *BCSM_to_SCF* -, so wird, abhängig vom Anrufer (Transition *cli_ServiceFeatures*) bzw. Angerufenen (Transition *number_ServiceFeatures*) und dem DP, aus dem zur SCF gesprungen wurde, ein dazu passendes Token von der Stelle *Service_structure* gelesen. Das bedeutet, daß ein Dienst ausgewählt wird, den der Teilnehmer von diesem Einsprung-DP aus ausführen kann. Dieser Dienst wird mit der dazugehörigen Service Feature-Liste zur Bearbeitung an die Stelle *Feature_queue* weitergegeben.

Es kann vorkommen, daß ein Dienst nach einer Bearbeitung in der Dienstlogik nicht abschließend ausgeführt ist. Um dann einen erneuten Sprung zur Dienstlogik im späteren Verlauf des Gesprächs zu ermöglichen, wird der Nachricht, die an das BCSM zurückgegeben wird, die dynamische Armierungsliste der passenden Marke von der Stelle *Service_structure* mitgegeben. Es ist aber auch möglich, daß die dynamische Armierungsliste, mit der zur Dienstlogik gesprungen wird, noch benötigt wird. Daher wird diese in der Stelle *Arming_alt* gesichert, bevor die Service Feature-Liste zur Bearbeitung an die Stelle *Feature_queue* weitergegeben wird. Sie muß beim Rücksprung zum BCSM wieder übernommen werden, falls der gewünschte Dienst nicht ausgeführt werden kann.

Service Control Function (SCF) Input

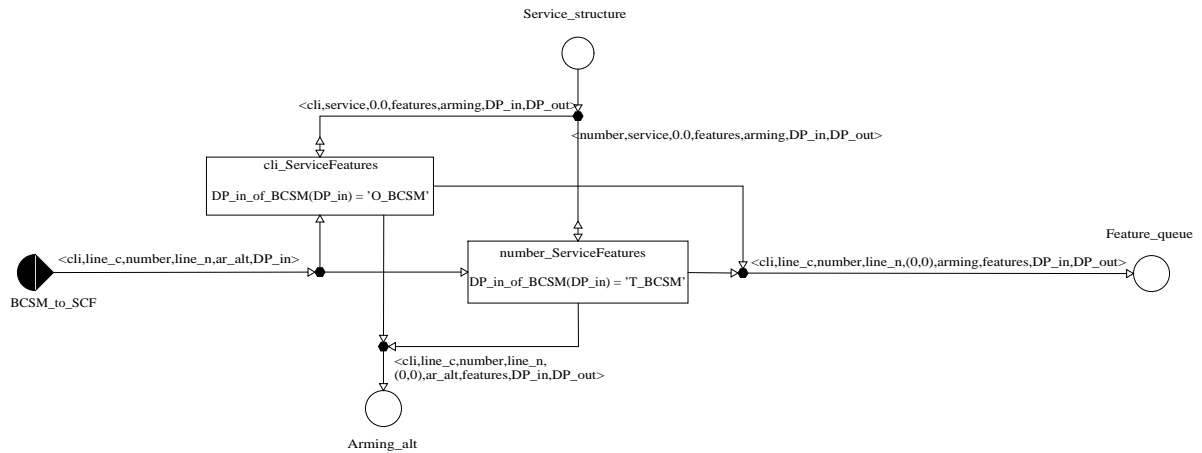


Abb. 9: Service Control Function Input

3.1.2.2. Service Control Function Output

Abbildung 10 zeigt das Produktnetz der Service Control Function Output. Die Transition *Result* erkennt anhand der leeren Liste von auszuführenden Service Features, daß ein Dienst abschließend bearbeitet wurde und gibt das Resultat über die Stelle *SCF_to_BCSM* an das BCSM zurück.

Service Control Function (SCF) Output

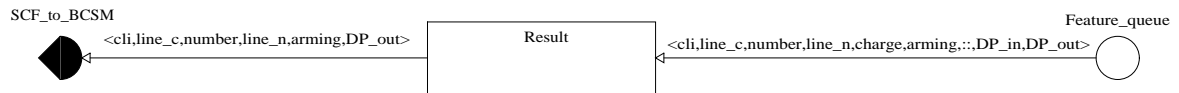


Abb. 10: Service Control Function Output

3.1.2.3. Dienste

Als Grundlage für die Implementierung der Dienste dient [Q.1221]. Im folgenden werden die im Rahmen dieser Arbeit modellierten Dienste beschrieben.

3.1.2.3.1. Call Hold (CH)

Der Dienst Call Hold ermöglicht es dem Dienstinutzer, eine bestehende Verbindung auf Warteposition zu setzen und einen neuen Anruf zu initiieren. Nach Beendigung des neuen Anrufs muß der Dienstinutzer dann den alten Anruf wieder aufnehmen.

Liste der zugehörigen Service Features

Feature	Aussprungadresse	Rücksprungadresse
Call Hold (CH)	DP-8 (O-BCSM) bzw. DP-16 (T-BCSM)	Entry_DP_1b
Call Retrieve (CRET)	DP-9 DP-10 PIC-6	Entry_PIC_5 (O-BCSM) bzw. Entry_PIC_10_after_accept (T-BCSM)

Szenarien

1. Dienstinutzer ist im ersten Gespräch Anrufer

Bei einem Dienstinutzer des Dienstes CH ist der Detection Point 8 in seinem O-BCSM statisch armiert. Sobald der Dienstinutzer mit einem Gespräch in die aktive Phase kommt, hat er damit die Möglichkeit über den Detection Point 8 in die Dienstlogik zu springen. Dort wird die vom BCSM stammende Nachricht interpretiert und in eine Featureliste umgewandelt. Von *DP-8* kommend besteht die Featureliste nur aus dem Service Feature Call Hold.

Dieses Feature legt das Token zur aktuellen Verbindung auf eine in der Dienstlogik eingeführte Stelle *On_hold*, falls der Dienstinutzer nicht bereits eine andere Verbindung auf Warteposition gesetzt hat. Weiterhin werden dann die Detection Points *DP-9*, *DP-10* und der *PIC-6* dynamisch armiert. Der Rücksprung erfolgt zu *Entry_DP_1b*.

Hat der Dienstinutzer bereits eine andere Verbindung auf Warteposition gesetzt, so besteht diese Möglichkeit nun nicht mehr und der Rücksprung erfolgt ohne dynamische Armierung und mit dem Token zu der Verbindung, mit der der Sprung zur Dienstlogik erfolgte, zurück zu *Entry_PIC_5*.

Erfolgt der Rücksprung aber zu *Entry_DP_1b*, so kann der Dienstinutzer nun einen neuen Anruf initiieren. Gelangt er im O-BCSM zu *DP-9*, *DP-10* oder *PIC-6* - das bedeutet auch, daß der neue Anruf beendet ist -, so sind diese für den Dienstinutzer dynamisch armiert. Es erfolgt ein Sprung in die Dienstlogik, wobei die Featureliste jetzt aus dem Feature Call Retrieve besteht. Dieses Feature nimmt das Token für die auf Warteposition gesetzte Verbindung von der Stelle *On_hold*. Mit diesem Token erfolgt dann der Rücksprung zur Stelle *Entry_PIC_5*.

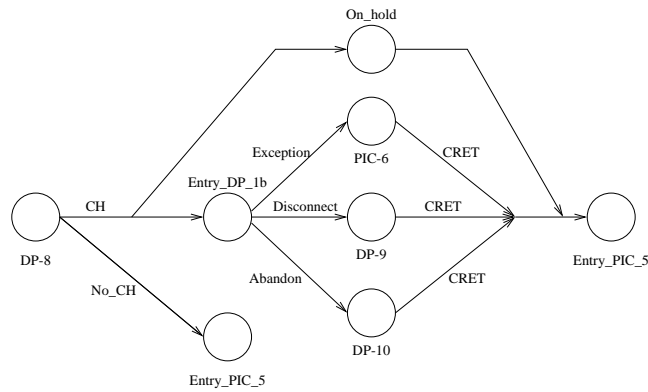


Abb. 11: Dienstaussführung Call Hold (Anrufer)

2. Dienstinutzer ist im ersten Gespräch Angerufener

Hier gilt im Prinzip der gleiche Ablauf wie im ersten Fall. Der Unterschied besteht nur darin, daß für diesen Dienstinutzer anstelle des DP-8 im O-BCSM der DP-16 im T-BCSM statisch armiert ist, und somit der erste Sprung zur SCF von dort erfolgt. Dementsprechend erfolgt der letzte Rücksprung zum BCSM anstatt zu Entry_PIC_5 im O-BCSM zu Entry_PIC_10_after_accept im T-BCSM.

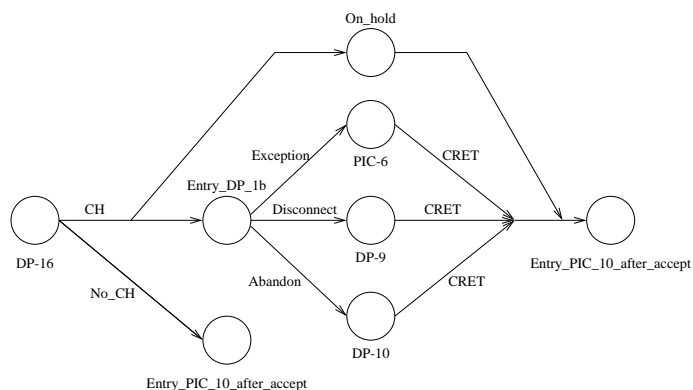


Abb. 12: Dienstaussführung Call Hold (Angerufener)

3.1.2.3.2. Call Waiting (CW)

Der Dienst Call Waiting ermöglicht es dem Dienstinutzer während eines Gesprächs, einen ihm angezeigten neu eingehenden Anruf entgegenzunehmen. Dazu wird der erste Anruf auf Warteposition gesetzt. Der Dienstinutzer hat danach die Möglichkeit, zwischen beiden Gesprächen hin- und herzuschalten.

Liste der zugehörigen Service Features

Feature	Aussprungadresse	Rücksprungadresse
Call Waiting (CW)	DP-13	Entry_PIC_8 SD_13_1 (in DP_13) kein direkter Rücksprung
Call Toggle (CTOG)	DP-8 DP-16	Entry_PIC_5 Entry_PIC_10_after_accept
Call Retrieve (CRET)	DP-17 DP-18 PIC-11 DP-9 DP-10 PIC-6	Entry_PIC_5 Entry_PIC_10_after_accept

Szenarien

Für den Dienstinutzer des Dienstes CW sind, falls er im ersten Gespräch Anrufer ist, im O-BCSM *DP-8*, *DP-9*, *DP-10* und *PIC-6* und im T-BCSM *DP-13*, *DP-17*, *DP-18* und *PIC-11* statisch armiert. Ist er im ersten Gespräch Angerufener, so sind im O-BCSM keine Punkte armiert. Im T-BCSM sind *DP-13*, *DP-17*, *DP-18*, *DP-16* und *PIC-11* statisch armiert.

Sobald die Marke zu einem Anruf im *Busy DP (DP-13)* ankommt, erfolgt ein Sprung zur Dienstelogik, falls der Angerufene den Call Waiting Dienst abonniert hat. In der SCF wird die vom BCSM stammende Nachricht interpretiert und in die Featureliste umgewandelt. Vom *DP-13* kommend besteht diese Featureliste nur aus dem Feature Call Waiting. Dieses Feature entscheidet folgendes: Hat der Dienstinutzer seinen anderen Anruf bereits beendet, so erfolgt der Rücksprung zum BCSM mit dem Token der neuen Verbindung zu *Entry_PIC_8*, wo dann erneut überprüft wird, ob der Angerufene besetzt oder frei ist. Besteht das alte Gespräch hingegen noch, so gibt es zwei Möglichkeiten. Falls der Dienstinutzer bereits einen anderen Anruf auf Warteposition hat, so wird das Token zur neuen Verbindung in die Stelle *SD_13_1* im *DP-13* gelegt, von wo aus diese Verbindung dann das Besetztzeichen erhält. Hat der Dienstinutzer noch keinen anderen Anruf auf Warteposition, so wird der neue Anruf auf Warteposition gesetzt und an den Anrufer wird die Nachricht *T_O_accept* geschickt, womit eine Ansage wie zum Beispiel "Bitte warten" simuliert werden soll.

Da für den Dienstinutzer *DP-8* bzw. *DP-16* statisch armiert ist, hat er, wenn er sich mit einem Anruf in der aktiven Phase befindet, immer die Möglichkeit zur Dienstelogik zu springen und dort das Service Feature Call Toggle auszuführen. Falls ein Anruf auf Warteposition ist, bewirkt dieses Service Feature, daß der Dienstinutzer mit diesem Anruf in die aktive Phase zurückkehrt, indem mit diesem Token zu *Entry_PIC_5*, falls der Dienstinutzer in diesem Anruf Anrufer war, bzw. zu *Entry_PIC_10_after_accept*, falls der Dienstinutzer in diesem Anruf

Angerufener war, zurückgesprungen wird. Gleichzeitig wird der Anruf, mit dem zur Dienstelogik gesprungen wurde, auf Warteposition gesetzt.

Beendet der Dienstanutzer eine Verbindung, während sich noch ein Anruf auf Warteposition befindet, so muß er diesen dann übernehmen. Das ist dadurch gewährleistet, daß er beim Verbindungsabbau über *DP-17*, *DP-18* oder *PIC-11* kommt, falls er im beendeten Anruf Angerufener war, bzw. über *DP-9*, *DP-10* oder *PIC-6*, falls er im beendeten Anruf Anrufer war. Da diese Punkte statisch armiert sind, erfolgt von dort ein Sprung zur SCF, wo das Service Feature Call Retrieve ausgeführt wird. Der Rücksprung zum BCSM erfolgt mit dem Token der auf Warteposition gesetzten Verbindung zu *Entry_PIC_5*, falls der Dienstanutzer in diesem Anruf Anrufer war, bzw. zu *Entry_PIC_10_after_accept*, falls der Dienstanutzer in diesem Anruf Angerufener war.

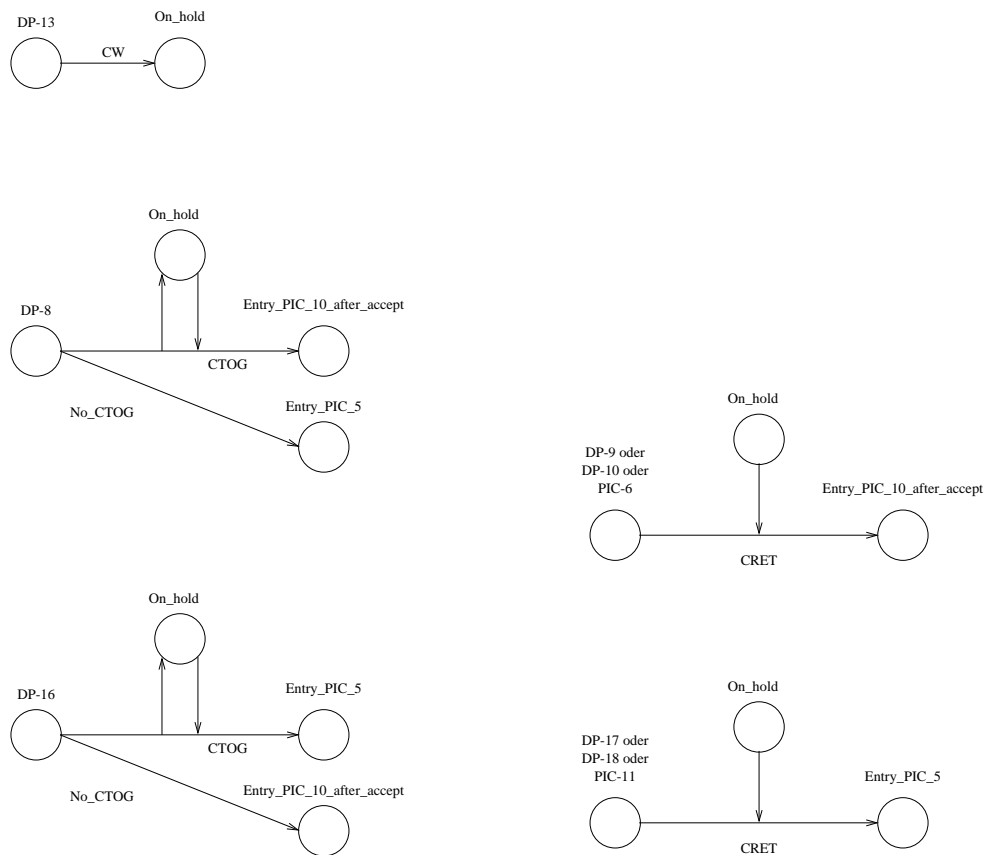


Abb. 13: Dienstausführung Call Waiting (Anrufer)

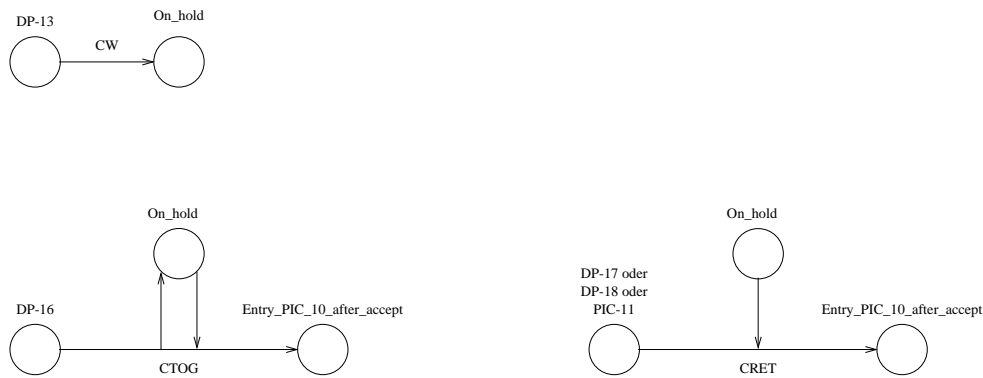


Abb. 14: Dienstausführung Call Waiting (Angerufener)

3.1.2.4. Service Features

Ein Dienst besteht im allgemeinen aus mehreren Service Features. Um einen Dienst zu bearbeiten, ist es notwendig, die einzelnen Service Features, aus denen der Dienst aufgebaut ist, nacheinander abzarbeiten. Dafür wird in der Service Control Function (SCF) die dafür notwendige Übersetzung eines Dienstes in seine zugehörige Featureliste vorgenommen (vgl. Service Control Function Input). Die Featureliste wird zur Bearbeitung auf der Stelle *Feature_queue* abgelegt. Die Marken auf dieser Stelle bestehen aus folgenden Komponenten:

- Anrufer
- Betriebsmittelnummer des Anrufers
- Angerufener
- Betriebsmittelnummer des Angerufenen
- Zahlungsvereinbarung
- dynamische Armierungsliste
- Service Feature Queue
- Einsprungspunkt zur Dienstelogik
- Rücksprungspunkt zum BCSM

Die Service Feature Queue stellt die Liste der Features dar, die den Dienst ausmachen. Im Einsprungspunkt ist gespeichert, von wo in die Dienstelogik gesprungen wurde. Der Rücksprungspunkt gibt an, zu welchem Punkt nach Abarbeitung der Service Feature Queue zurückgesprungen werden muß.

Um die Service Features in der Service Feature Queue in der richtigen Reihenfolge abzarbeiten, wird in den Transitionen der einzelnen Service Features mit der Funktion *seg(features, I, I)* abgefragt, welches das nächste abzarbeitende Service Feature ist. Dadurch kann nur das Service Feature aktiv werden, das an der Reihe ist.

Da die dynamische Armierungsliste, mit der zur Dienstlogik gesprungen wurde, in der Stelle *Arming_alt* zwischengespeichert wurde, müssen die Service Features in je zwei nahezu identische Netze aufgespalten werden. Eines für den Fall, daß die Feature Queue abschließend abgearbeitet wurde und eines, falls noch nachfolgende Features bearbeitet werden müssen. Im ersten Fall wird die Marke mit der alten dynamischen Armierung von der Stelle *Arming_alt* abgezogen und gegebenenfalls wieder übernommen. Im zweiten Fall bleibt die Stelle *Arming_alt* unberührt. In allen übrigen Funktionalitäten stimmen beide Netze überein, daher wird die Modellierung später jeweils nur an einem Netz beschrieben.

Weiterhin wird - falls notwendig - je nach Einsprungspunkt unterschieden, ob der Dienst für einen Dienstanutzer, der Anrufer (O-BCSM) oder Angerufener (T-BCSM) ist, ausgeführt werden soll. Falls erforderlich werden demzufolge die Netze für die einzelnen Service Features wiederum in eines für O-BCSM und eines für T-BCSM aufgespalten.

Nachdem alle Service Features abgearbeitet sind, d.h. die Featureliste leer (:) ist, wird das endgültige Ergebnis an das BCSM zurückgesendet (vgl. Service Control Function Output).

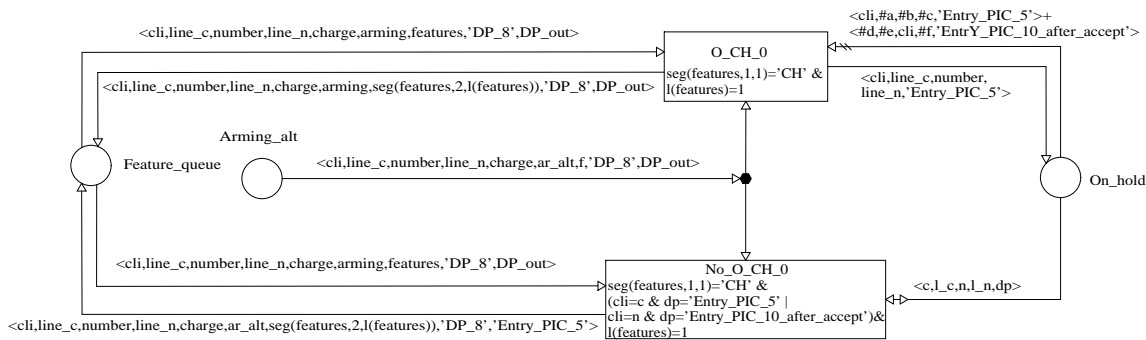
3.1.2.4.1. Call Hold (CH)

Das Service Feature Call Hold setzt eine bestehende Verbindung auf Warteposition.

In diesem Feature wird überprüft, ob derjenige, für den der Dienst ausgeführt wird, bereits eine Verbindung auf Warteposition gesetzt hat. Ist das der Fall, so kann er dies nicht noch einmal tun. Es wird mit der alten Verbindung zu *Entry_PIC_5* (CH für O-BCSM) bzw. *Entry_PIC_10_after_accept* (CH für T-BCSM) zurückgesprungen, wobei auch die alte Armierungsliste wieder übernommen wird (Transitionen *No_O_CH_0*, *No_T_CH_0*).

Hat der Dienstanutzer keine Verbindung auf Warteposition, so wird die aktuelle Verbindung auf Warteposition gesetzt, d.h. das Token für die Verbindung wird mit der zusätzlichen Information, von wo in die Dienstlogik gesprungen wurde, auf die Stelle *On_hold* gelegt. Der Rücksprung zum BCSM folgt jetzt zu *DP_out* (Transitionen *O_CH_0*, *T_CH_0*).

Service Feature: Call Hold (CH) fuer O-BCSM



Service Feature: Call Hold (CH) fuer T-BCSM

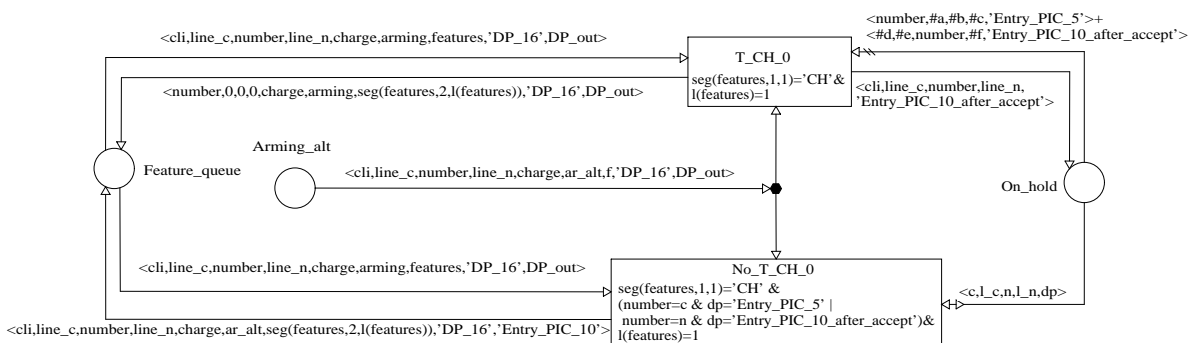


Abb. 15: Service Feature Call Hold

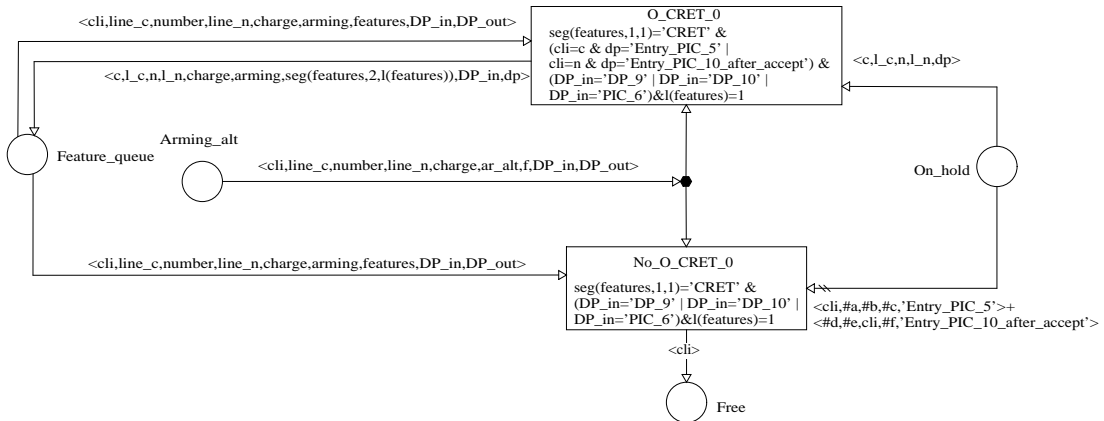
3.1.2.4.2. Call Retrieve (CRET)

Das Service Feature Call Retrieve ermöglicht es, zu einem Gespräch zurückzukehren, das zuvor auf Warteposition gesetzt wurde.

Liegt eine Verbindung auf Warteposition, so wird mit dieser in das BCSM zurückgesprungen (Transitionen *O_CRET_0*, *T_CRET_0*).

Liegt keine Verbindung auf Warteposition, so wird der Dienstnutzer für andere Gespräche freigegeben (Transitionen *No_O_CRET_0*, *No_T_CRET_0*).

Service Feature: Call Retrieve (CRET) fuer O-BCSM



Service Feature: Call Retrieve (CRET) fuer T-BCSM

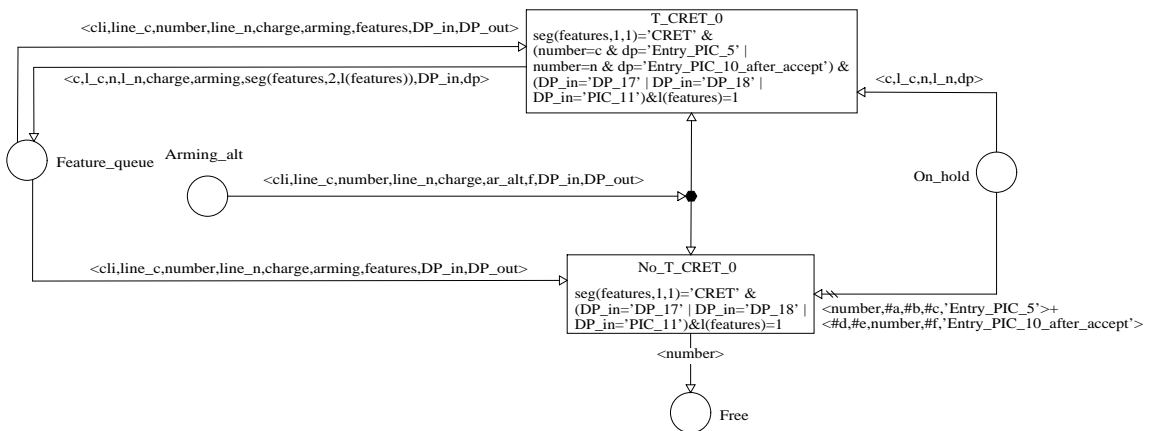


Abb. 16: Service Feature Call Retrieve

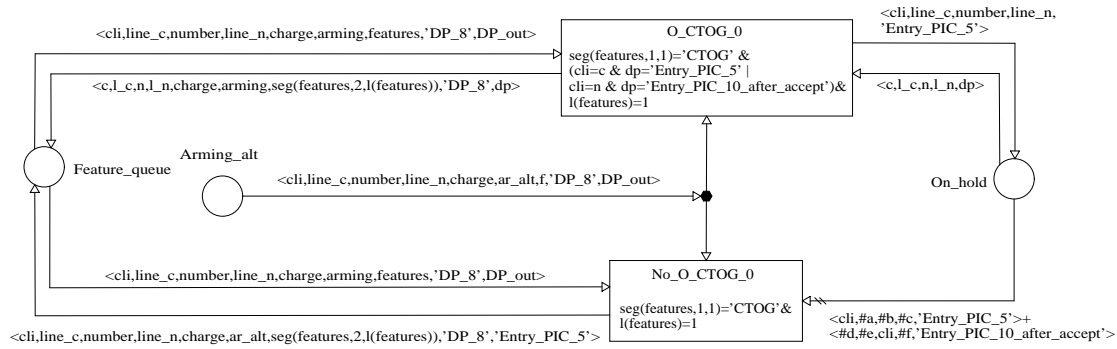
3.1.2.4.3. Call Toggle (CTOG)

Mittels des Service Features Call Toggle kann zwischen einem aktiven und einem wartenden Anruf hin- und hergeschaltet werden.

Liegt eine Verbindung auf Warteposition, so wird mit dieser Verbindung zum BCSM zurückgesprungen. Gleichzeitig wird die Verbindung, mit der zur Dienstlogik gesprungen wurde, auf Warteposition gesetzt, d.h. das Token mit der Verbindung wird mit der zusätzlichen Information, von wo in die Dienstlogik gesprungen wurde, auf die Stelle *On_hold* gelegt.

Liegt keine Verbindung auf Warteposition, so wird mit der alten Verbindung zu *Entry_PIC_5* (CTOG für O-BCSM) bzw. *Entry_PIC_10_after_accept* (CTOG für T-BCSM) zurückgesprungen, wobei auch die alte Armierungsliste wieder übernommen wird (Transitionen *No_O_CTOG_0*, *No_T_CTOG_0*).

Service Feature: Call Toggle (CTOG) fuer O-BCSM



Service Feature: Call Toggle (CTOG) fuer T-BCSM

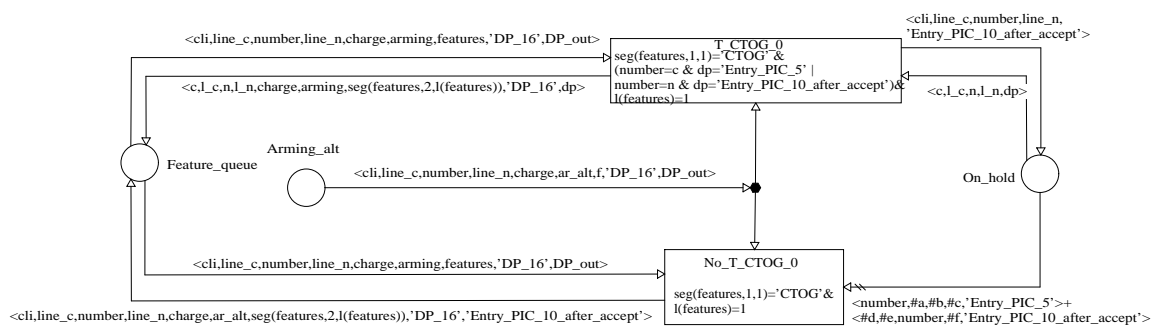


Abb. 17: Service Feature Call Toggle

3.1.2.4.4. Call Waiting (CW)

Ein Dienstnutzer, der an einer bestehenden Verbindung teilnimmt, wird durch das Service Feature Call Waiting darüber informiert, daß jemand Verbindung zu ihm aufnehmen will.

Falls der Dienstnutzer das zuvor bestehende Gespräch bereits beendet hat, wird mit der alten Armierungsliste zu *Entry_PIC_8* zurückgesprungen (Transition *CW_0_2*).

Falls der Dienstnutzer noch an einer bestehenden Verbindung beteiligt ist, gibt es die folgenden Möglichkeiten:

Liegt bereits eine Verbindung für den Dienstnutzer auf Warteposition, so kann keine mehr hinzugefügt werden, und es wird mit der alten Armierungsliste zu *SD_13_1* in *DP-13* zurückgesprungen (Transition *No_CW_0*).

Ist dies nicht der Fall, so wird die aktuelle Verbindung auf Warteposition gesetzt, d.h. das Token der Verbindung wird mit der zusätzlichen Information, von wo in die Dienstlogik gesprungen wurde, auf die Stelle *On_hold* gelegt. Es erfolgt kein direkter Rücksprung zum BCSM. Weiterhin wird eine *T_O_accept* Nachricht an den Anrufer geschickt, was eine Ansage wie "Bitte warten" simuliert (Transition *CW_0_1*).

Service Feature: Call Waiting (CW)

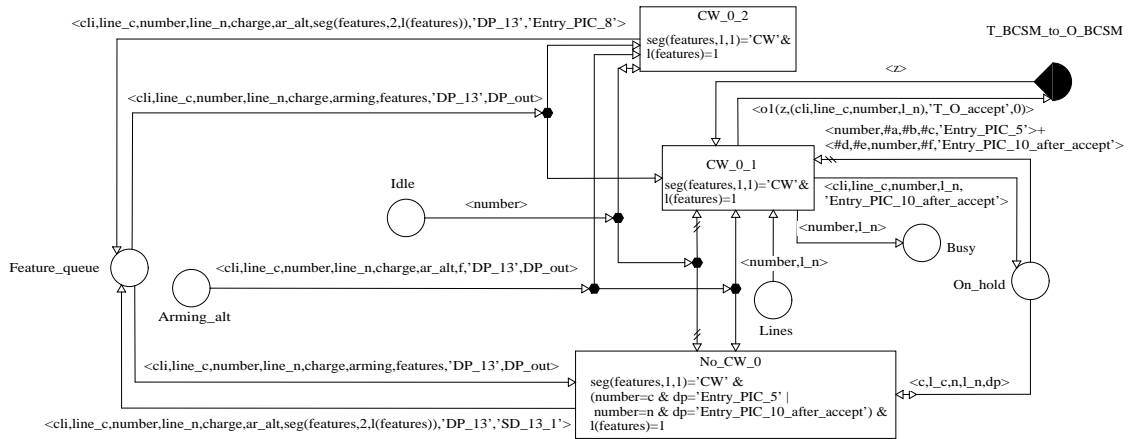


Abb. 18: Service Feature Call Waiting

3.2. Erste Spezifikation

Der Vorspann und die Produktnetze zu dieser Spezifikation befinden sich im Anhang A.

3.2.1. Aspekte der Modellierung

In diesem Abschnitt werden Aspekte der Modellierung erläutert, die in der Beschreibung des BCSMs im Standard [Q.1224] nicht berücksichtigt sind.

3.2.1.1. Einführung von Prozeß-Ids

Da ein Teilnehmer in CS-2 zwei Verbindungen parallel haben darf - eine aktive und eine auf Warteposition gesetzte -, wird die Einführung von Prozeß-Ids zur eindeutigen Identifikation eines Gesprächs notwendig. Als Prozeß-Ids werden Tupel der Form (Nummer des Anrufers, Betriebsmittelnummer des Anrufers, Nummer des Angerufenen, Betriebsmittelnummer des Angerufenen) gewählt. Durch die Vergabe von maximal zwei sich unterscheidenden Betriebsmittelnummern pro Teilnehmer kann jede Verbindung eindeutig identifiziert werden, und es ist sichergestellt, daß ein Teilnehmer an höchstens zwei Anrufen beteiligt ist.

3.2.1.2. Beschreibung ausgezeichneter Stellen

- *Busy*
Die Stelle *Busy* zeigt an, ob ein Teilnehmer telefoniert und welches Betriebsmittel er dazu benutzt. Telefoniert ein Teilnehmer, so liegt auf der Stelle *Busy* eine Marke der Form (Nummer des Teilnehmers, Betriebsmittelnummer).
- *Idle*
Die Stelle *Idle* ist Indikator dafür, ob ein Teilnehmer aus dem Grundzustand heraus eine Verbindung aufbauen darf bzw. für neue nicht Anrufe besetzt ist. Darf ein Teilnehmer eine Verbindung aufbauen bzw. ist er für neue Anrufe nicht besetzt, so liegt auf der Stelle *Idle* eine Marke, die die Nummer des Teilnehmers repräsentiert.
- *Systems*
Mit der Stelle *Systems* wird die Kapazität des O-BCSMs festgelegt. Eine Marke '*system_ready*' zeigt an, daß ein ausgehendes Gespräch vermittelt werden kann.
- *Entry_PIC_7*
Mit der Stelle *Entry_PIC_7* wird die Kapazität des T-BCSMs festgelegt. Eine Marke '*system_ready*' zeigt an, daß ein eingehendes Gespräch vermittelt werden kann.
- *Entry_PIC_1*
Die Stelle *Entry_PIC_1* enthält als Marken die Nummern aller Teilnehmer, die einen Anruf initiieren können.

- *Number_Pool*

Die Stelle *Number_Pool* enthält als Marken die Nummern aller Teilnehmer, die angerufen werden können.

3.2.1.3. Unterscheidung zwischen Auflegen und aufgetretenem Fehler beim Angerufenen

Tritt beim Angerufenen im Zustand *T_Active* ein Fehler auf, erfolgt gemäß dem Standard ein Übergang in den *T_Exception PIC*. Wird dabei, wie beim Auflegen, nur ein *T_O_disconnect* geschickt, wechselt der Anrufer in den Zustand *O_Suspended* und wartet vergeblich auf ein weiteres *T_O_disconnect*, um in den Zustand *O_Disconnect* übergehen zu können. Deshalb werden in diesem Fall im Zustand *T_Active* zwei *T_O_disconnects* an den Anrufer geschickt. Dadurch kann er nach dem Wechsel in den *O_Suspended PIC* über den *O_Disconnect DP* in den Grundzustand gelangen.

3.2.1.4. Sonderbehandlung von *O_T_abandon*

Befinden sich sowohl O- als auch T-BCSM bezüglich einer bestimmten Verbindung im Grundzustand, so wird davon ausgegangen, daß beide Schnittstellen für diese Verbindung keine Marken enthalten. Um dies im Fall eines *O_T_abandon* nach dem Schicken eines *O_T_initiate* sicherstellen zu können, wird dem Abnehmen der Nachricht *O_T_abandon* auf der Seite des T-BCSMs Vorrang gegenüber allen anderen Aktionen eingeräumt. Dies wird erreicht, indem Verbotskanten mit der Anschrift *O_T_abandon* von der Schnittstelle *O_BCSM_to_T_BCSM* zu allen anderen eventuell nebenläufig aktivierten Transitionen eingeführt werden.

Im T-BCSM gibt es nach [Q.1224] einen DP mit der Bezeichnung *T_Abandon*. Dies und die dazugehörenden Transitionen legen nahe, daß der Angerufene nach dem Empfang der Nachricht *O_T_abandon* über diesen DP in den Grundzustand gelangt. Das ist jedoch nicht in allen Fällen möglich. Es wurde daher folgende Vorgehensweise gewählt:

- Wird *O_T_abandon* im *T_Busy DP* oder im *T_No_Answer DP* entgegengenommen, so geht der Angerufene über den *T_Exception PIC* in den Grundzustand.
- Erhält der Angerufene die Nachricht *O_T_abandon* erst im *T_Active PIC*, so gelangt er über den *T_Disconnect DP* in den Grundzustand.

3.2.1.5. Unendlicher Zustandsraum durch Ansammeln von Nachrichten auf den Schnittstellen

Da eine Verbindung nicht sofort beendet ist, wenn der Angerufene auflegt (Schicken von *T_O_disconnect*), entstehen Verhaltensweisen, die im Standard nicht berücksichtigt sind. Der Angerufene kann die Verbindung reaktivieren, indem er innerhalb einer gewissen Zeitspanne wieder abhebt (Schicken von *T_O_reanswer*). Dadurch gelangt er in den *T_Active PIC* zurück

und hat damit die Möglichkeit, durch erneutes Auflegen, Abheben, ... $T_O_disconnect$, $T_O_reanswer$, ... zu schicken. Da der Anrufer diese Nachrichten nicht abnehmen muß, können sie sich auf der Schnittstelle $T_BCSM_to_O_BCSM$ ansammeln.

Das gleiche Problem tritt auf, wenn der Anrufer nach dem Abschicken eines Verbindungsaufbauwunsches ($O_T_initiate$) sofort auflegt (Schicken von $O_T_abandon$). Er gelangt wieder in den Grundzustand und hat damit erneut die Möglichkeit, einen Aufbauwunsch zu schicken und wieder aufzulegen. Es können sich somit beliebig viele Nachrichten $O_T_initiate$ und $O_T_abandon$ auf der Schnittstelle $O_BCSM_to_T_BCSM$ ansammeln, denn der Angerufene ist nicht gezwungen, auf eine dieser Nachrichten zu reagieren.

In beiden Fällen wird der Zustandsraum unendlich groß und somit eine vollständige Analyse unmöglich.

In der Beschreibung des BCSMs ist keine Aussage über ein Transportsystem und den "End-zu-End-Schlupf" gemacht, weshalb man sich für eine geeignete Modellierungsweise entscheiden muß. Um den Zustandsraum klein zu halten, wird zur Lösung des Problems eine möglichst einfache Modifizierung des Modells gewählt. Diese besteht darin, Handshakes für das Schicken der Nachrichten $O_T_abandon$, $T_O_disconnect$ und $T_O_reanswer$ an den entsprechenden Stellen einzubauen.

Abb. 19 zeigt das Schema eines symmetrischen Handshakes.

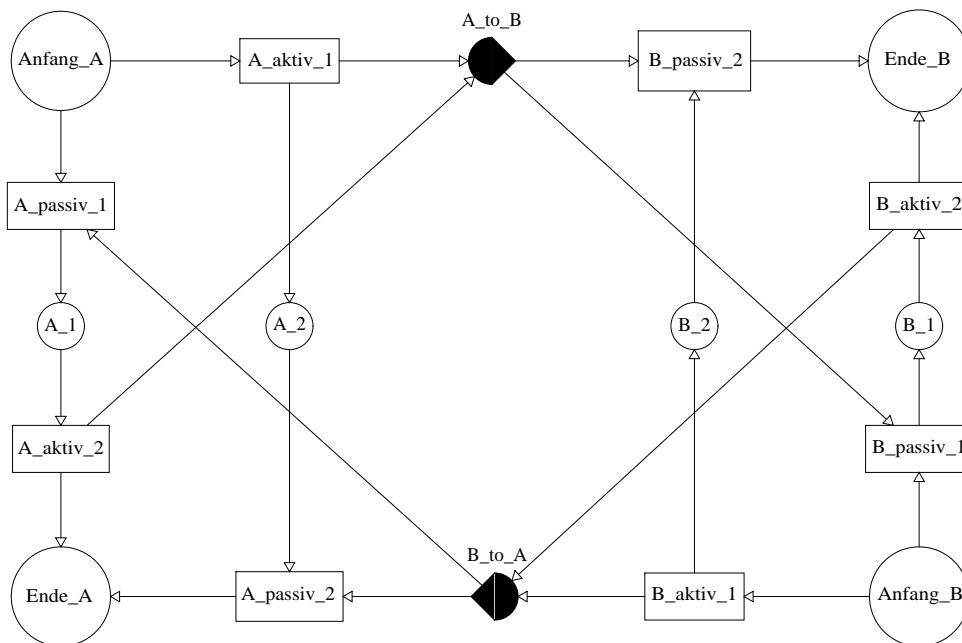


Abb. 19: Schema eines symmetrischen Handshakes

Schickt ein Teilnehmer eine Nachricht (aktives Geschehen, A_aktiv_1), so geht er in einen Zwischenzustand und wartet auf die Quittung des anderen Teilnehmers (A_2). Erhält er diese (A_passiv_2), ist der Nachrichtenaustausch beendet ($Ende_A$). Erhält ein Teilnehmer eine

Nachricht (passives Geschehen, A_{passiv_1}), so geht er in einen Zwischenzustand, in dem er eine Quittung an den anderen Teilnehmer schicken muß (A_1). Schickt er diese (A_{aktiv_2}), ist der Nachrichtenaustausch beendet ($Ende_A$).

Schicken beide Teilnehmer gleichzeitig eine Nachricht (A_{aktiv_1} und B_{aktiv_1}), so nehmen sie die Nachricht des jeweils anderen Teilnehmers als Quittung ihrer eigenen Nachricht an (A_{passiv_2} und B_{passiv_2}). Der Nachrichtenaustausch ist dann beendet ($Ende_A$, $Ende_B$).

Mit der Einführung der Handshakes (End-zu-End-Handshakes) sind folgende Prämissen verbunden:

- strikte Synchronisation bei den durch Handshake kontrollierten Kommunikationsprimitiven
- dadurch globale Flußkontrolle

Diese Modellierung entspricht einem Transportsystem mit nicht überschreibbarem Puffer der Größe eins und macht keine Voraussetzungen bezüglich der Reihenfolgeerhaltung der Nachrichten.

Da das Auflegen der Teilnehmer nicht symmetrisch ist, wenn sich beide in der aktiven Phase befinden - beendet der Angerufene die Verbindung, so besteht sie noch einige Zeit, in der er die Möglichkeit hat, sie wieder aufzunehmen -, kann dieses Schema nicht einfach übernommen werden. Denn für die weitere Verarbeitung muß bekannt sein, von wem das Beenden ausgeht. Dieses Problem wird mit Hilfe von Verbotskanten mit Disconnect-Anschrift gelöst. Diese führen von der jeweiligen Schnittstelle zu der Transition, die ein aktives Disconnect auslöst. Dadurch wird verhindert, daß ein Teilnehmer eine Verbindung aktiv beendet, wenn bereits ein Disconnect vom anderen Teilnehmer geschickt wurde.

In den *Suspended PICs* ist das Beenden symmetrisch und obiges Schema kann übernommen werden. Da hier auch andere Primitive möglich sind, benötigt man aber eine Verbotskante mit der Anschrift der jeweils anderen Primitive von der Schnittstelle zu den Transitionen, von denen eine Nachricht an den anderen Teilnehmer geschickt wird. Andernfalls ist es zum Beispiel möglich, daß der Angerufene ein $T_O_reanswer$ schickt und der Anrufer gleichzeitig ein $O_T_disconnect$. Beide warten dann auf den Handshake des anderen und es entsteht eine Deadlocksituation.

3.2.2. Analyse und Verifikation

Zur Analyse und Verifikation dieser ersten Spezifikation werden für verschiedene Anfangsmarkierungen die Erreichbarkeitsgraphen berechnet. Anschließend werden jeweils Homomorphismen zur Betrachtung spezieller Verhaltensweisen definiert. Wie in Abschnitt 1.1.2. erläutert, werden die Homomorphismen benutzt, um bezüglich einer Eigenschaft

irrelevante Aktionen auszublenden oder gleichartige Aktionen zusammenzufassen. Alle Aktionen werden durch das Schalten von Transitionen (Kanten im Erreichbarkeitsgraphen) repräsentiert. Bei der Definition der Homomorphismen werden zunächst Blöcke in Abhängigkeit der zu verifizierenden Eigenschaft gebildet. Alle Transitionen werden auf die Blöcke verteilt. Der Block, welcher die Transitionen enthält, die die irrelevanten Aktionen repräsentieren, wird auf ε abgebildet. Dies bedeutet, daß die entsprechenden Kanten im Erreichbarkeitsgraphen ausgeblendet werden. Die anderen Blöcke werden nicht auf ε abgebildet. Für diese besteht zusätzlich die Möglichkeit, bei den Markierungen bzw. Folgemarkierungen nur bestimmte Stellen zu berücksichtigen (Q-Markierung).

Der so aus einem Erreichbarkeitsgraph entstehende nichtdeterministische endliche Automat mit ε -Übergängen wird anschließend zu einem deterministischen endlichen Automat umgeformt und minimiert. Dieser Minimalautomat repräsentiert dann die Verhaltensweise, die unter der Sicht des Homomorphismus in der Dynamik des Modells enthalten ist.

Im folgenden wird, wenn nichts anders gesagt wird, davon ausgegangen, daß in den Homomorphismen keine Stellenbelegungen berücksichtigt werden. Dadurch werden unter einem Schaltfolgenhomomorphismus im Bild eines Wortes aus der Schaltfolgensprache die erste und dritte Komponente uninteressant und bleiben, wenn nicht besonders erwähnt, bei der Berechnung des Minimalautomaten unberücksichtigt (vgl. Abschnitt 1.1.2.).

Für die Übertragung einiger durch Homomorphismen aufgezeigter Eigenschaften auf die Dynamik des Modells ist die Eigenschaft der Schlichtheit des entsprechenden Homomorphismus, wie bereits im Abschnitt 1.1.2. erläutert, von wesentlicher Bedeutung. Sie gilt, wenn der Erreichbarkeitsgraph nur eine starke Zusammenhangskomponente besitzt. Um diese Eigenschaft zu überprüfen, wird zu jedem Erreichbarkeitsgraphen der Faktorgraph berechnet.

Die folgenden Verhaltensweisen werden mittels Homomorphismen betrachtet:

- lokale Verhaltensweisen
 1. Anrufer ohne abonnierten Dienst bzw. mit abonniertem Call Hold-Dienst
 2. Angerufener
 3. O-BCSM
 4. T-BCSM
- globale Verhaltensweisen
 5. Kommunikation zwischen O- und T-BCSM
 6. Phasentrennung und Lebendigkeitseigenschaft
- vergleichende Homomorphismen
 7. Verhalten bei abonniertem Call Hold-Service
 8. Besetzt-Verhalten

Die Grundlage für die Verhaltensverifikation von O-BCSM und T-BCSM ist [Q.1224]. Bei der Verifikation des Verhaltens von Anrufer, Angerufenem und der Kommunikation wird nur berücksichtigt, ob ein "vernünftiges" Verhalten zustandekommt.

Die vergleichenden Homomorphismen werden definiert, um Unterschiede zwischen den Verhaltensweisen der ersten Spezifikation und weiterer Spezifikationen verdeutlichen zu können.

3.2.2.1. Anrufer

3.2.2.1.1. Anrufer ohne abonnierten Dienst

Der Untersuchung liegt folgende Erreichbarkeitsanalyse zugrunde:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
SYSTEMS: 1<SYSTEM_READY>
```

Erreichbarkeitsgraph bcsms2c

```
614 Markierungen berechnet.
(0 Totmarkierungen)
(0 Pseudoschaltsschritte)
(1393 Schaltsschritte)
```

```
Faktorgraph von Erreichbarkeitsgraph bcsms2c
(1 Zusammenhangskomponente gefunden)
(1 Randkomponente)
```

Da der Erreichbarkeitsgraph nur eine Zusammenhangskomponente besitzt, ist jeder darauf definierte Homomorphismus schlicht.

Um ein "vernünftiges" Verhalten eines Anrufers zu überprüfen, werden die drei Aktionen *abheben*, *Nummer wählen* und *auflegen* betrachtet. Der Anrufer sollte nach dem Abheben sofort wieder auflegen oder eine Nummer wählen können. Nach dem Wählen einer Nummer sollte mit dem Auflegen wieder der Startzustand erreicht werden.

Der Homomorphismus für den Anrufer besteht also - außer aus dem Block für die auszublendenden Transitionen - aus drei weiteren Blöcken, denen die Transitionen gemäß folgender Tabelle zugeordnet werden:

Block	Transitionen
abheben	attempt_1a
Nummer_waehlen	collected_info_2
auflegen	TP_6_a, TD_9_a, TD_10_a
Epsilon	alle restlichen Transitionen

Dabei wird die Transition *attempt_1a* dem Block *abheben* zugeordnet, da nach ihrem Schalten das Kriterium für "besetzt" erfüllt ist, was nach dem Abheben des Hörers ja ebenfalls der Fall ist.

Die Transition *collected_info_2* korrespondiert mit *Nummer wählen*, da bei ihrem Schalten die Nummer des Anzurufenden aus der Stelle *Number_Pool* gelesen wird.

Die Transitionen im Block *auflegen* werden so gewählt, daß bei ihrem Schalten die von einer Verbindung belegten Betriebsmittel wieder freigegeben werden.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bcsms2c
3 Zustände berechnet.
(4 Kanten, 3 Anschriften)
```

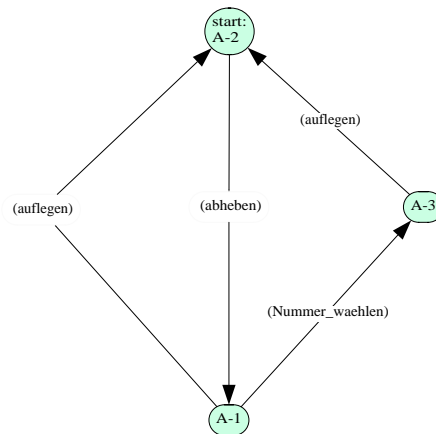


Abb. 20: Minimalautomat "Anrufer"

Der resultierende Minimalautomat in Abb. 20 zeigt, daß in diesem Modell das erwartete Verhalten eines Anrufers enthalten ist.

3.2.2.1.2. Anrufer mit abonniertem Call Hold-Dienst

Im folgenden wird das Verhalten eines Anrufers mit abonniertem Call Hold-Dienst untersucht. Neben seinem normalen Verhalten sollte der Anrufer zwischen Nummer wählen und Auflegen die aktive Verbindung auf Warteposition setzen können. Danach muß er gleich wieder auflegen können, wonach er die erste Verbindung reaktivieren muß (Feature Call Retrieve), oder erneut eine Nummer wählen können. Hierauf sollte der Anrufer nur auflegen können, worauf ein Call Retrieve folgen müßte.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:
ENTRY_PIC_1: 1<1>
```

```

ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<1,2> + 1<2,1>
NUMBER_POOL: 1<2>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9.PIC_6.DP_10,DP_8,DP_1B> +
    1<1,CH,0.0,CRET,::,DP_10,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,PIC_6,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8>
SYSTEMS: 2<SYSTEM_READY>

```

Reachability Graph bcsms2c

```

2785 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(8163 Occurrence Steps)

```

```

Connected Components of Reachability Graph bcsms2c
(1 Component found)
(1 Leaf Component)

```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	ATTEMPT_1A
Nummer_waehlen	COLLECTED_INFO_2
auflegen	TD_10_D, TD_9_C, TD_10_C, EXCEPTION_6_A, EXCEPTION_6B
Call_Hold	O_CH_0
Call_Retrieve	O_CRET_0
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c

```

6 States computed.
(9 Edges, 5 Labels)

```

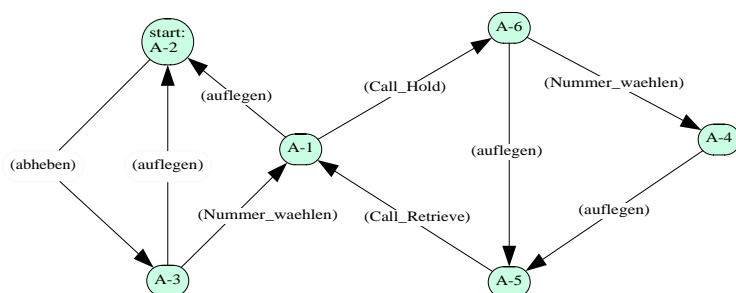


Abb. 21: Minimalautomat "Anrufer mit Call Hold-Dienst"

Der Anrufer hat nach dem Wählen der Nummer die Möglichkeit aufzulegen oder das Feature Call Hold auszuführen, d.h. er setzt die Verbindung auf Warteposition. Danach kann er eine

zweite Verbindung aufbauen, repräsentiert durch den Block *Nummer_waehlen*, oder auflegen (Block *auflegen*). In diesem Fall wie auch nach dem Beenden der zweiten Verbindung muß der Anrufer zur ersten Verbindung zurückkehren (Service Feature Call Retrieve). Danach befindet er sich mit dieser Verbindung wieder in der aktiven Phase und kann erneut das Call Hold-Feature ausführen oder auflegen.

Als Fazit läßt sich feststellen, daß das Ergebnis mit den Erwartungen übereinstimmt.

3.2.2.2. Angerufener

Analog zum Anrufer werden hier die Aktionen *abheben* und *auflegen* betrachtet. Nach dem Abheben kann der Anrufer nur auflegen, wodurch wieder der Startzustand erreicht wird.

Der Untersuchung liegt dieselbe Erreichbarkeitsanalyse wie im Abschnitt 3.2.2.1.1. zugrunde.

Den Blöcken werden bei der Definition des Homomorphismus Transitionen entsprechend der folgenden Tabelle zugeordnet:

Block	Transitionen
abheben	answer_8b, answer_9
auflegen	TD_17_1, T_active_failure_10a, T_suspend_failure_10b
Epsilon	alle restlichen Transitionen

Im Block *abheben* sind die Transitionen enthalten, deren Schalten nur dann möglich ist, wenn ein Anruf nicht mit "besetzt" beantwortet oder gänzlich zurückgewiesen (Schicken von *T_O_no_answer*) wird.

Dem Block *auflegen* werden die Transitionen zugeordnet, die (alternativ) schalten, wenn der Angerufene die aktive Phase endgültig verläßt.

Die Berechnung des Minimalautomaten liefert folgendes:

```
Minimalautomat bcsms2c
```

```
2 Zustände berechnet.  
(2 Kanten, 2 Anschriften)
```

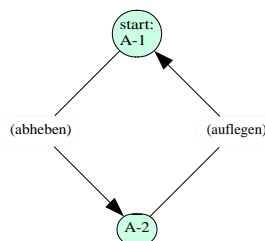


Abb. 22: Minimalautomat "Angerufener"

Der Minimalautomat in Abb. 22 zeigt, daß in diesem Modell das erwartete Verhalten eines Angerufenen enthalten ist.

3.2.2.3. O-BCSM

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<1> + 1<2>
SYSTEMS: 1<SYSTEM_READY>
```

Erreichbarkeitsgraph bcsms2c

4447 Markierungen berechnet.
 (0 Totmarkierungen)
 (0 Pseudoschaltsschritte)
 (13859 Schaltschritte)

Faktorgraph von Erreichbarkeitsgraph bcsms2c
 (1 Zusammenhangskomponente gefunden)
 (1 Randkomponente)

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
o_ok	ATTEMPT_1A
DP_1_not_armed	DP_1_NOT_ARMED
O_abandon	ABANDON_4C_1, ABANDON_4C_2, ABANDON_1B, ABANDON_2, ABANDON_3, ABANDON_4A, ABANDON_4B, ABANDON_4D
O_collected_info	COLLECTED_INFO_2
O_origination_denied	ORIGINATION_DENIED_1B
DP_2_not_armed	DP_2_NOT_ARMED
O_analysed_info	ANALYSED_INFO_3
DP_3_not_armed	DP_3_NOT_ARMED
Route_select_Failure	ROUTE_FAILURE_4A
O_interpret	ROUTE_SELECTED_4A
O_authorize_route_failure	AUTHORIZE_ROUTE_FAILURE_4C, AUTHORIZE_ROUTE_FAILURE_4B
O_authorized_for_number	CALL_SETUP_4B
O_answer	ANSWER_4D, T_4C
O_busy	BUSY_4C, BUSY_4D
O_no_answer	NO_ANSWER_4D, NO_ANSWER_4C
O_alert	ALERTED_4C
DP_7_not_armed	DP_7_NOT_ARMED
O_Mid_Call_5	MID_CALL_5

O_Mid_Call_5b	MID_CALL_5B
O_Disconnect	TP_5B_2, TP_5B_1, TP_5_2
DP_8_not_armed	DP_8_NOT_ARMED_1, DP_8_NOT_ARMED_2
O_Return_DP9_to_Start	TD_9_C
O_Return_DP10_to_Start	TD_10_C
O_Return_PIC6_to_Start	EXCEPTION_6_A, EXCEPTION_6B
O_Exception_DP4	DP_4_NOT_ARMED
O_Exception_DP5	DP_5_NOT_ARMED
O_Exception_DP6	DP_6_NOT_ARMED
O_authorized	AUTHORIZED_1B
O_re_answer	TP_5B_4
DP_20_not_armed	DP_20_NOT_ARMED
O_Suspend	TP_5_1
DP_19_not_armed	DP_19_NOT_ARMED
DP_1b_not_armed	DP_1B_NOT_ARMED
DP_25_not_armed	DP_25_NOT_ARMED
O_route_busy	ROUTE_BUSY_4A
O_route_failure_4c	ROUTE_FAILURE_4C
O_Mid_Call_4c	MID_CALL_4C
O_Mid_Call_4d	MID_CALL_4D
O_active_failure_5	O_ACTIVE_FAILURE_5
O_suspend_failure_5b	O_SUSPEND_FAILURE_5B
O_invalid_information	INVALID_INFORMATION_3
O_collect_timeout	COLLECT_TIMEOUT_2
O_route_failure_4d	ROUTE_FAILURE_4D
Epsilon	alle restlichen Transitionen

Der Homomorphismus enthält alle Transitionen des O-BCSMs.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bcsms2c
```

```
28 Zustände berechnet.  
(55 Kanten, 43 Anschriften)
```

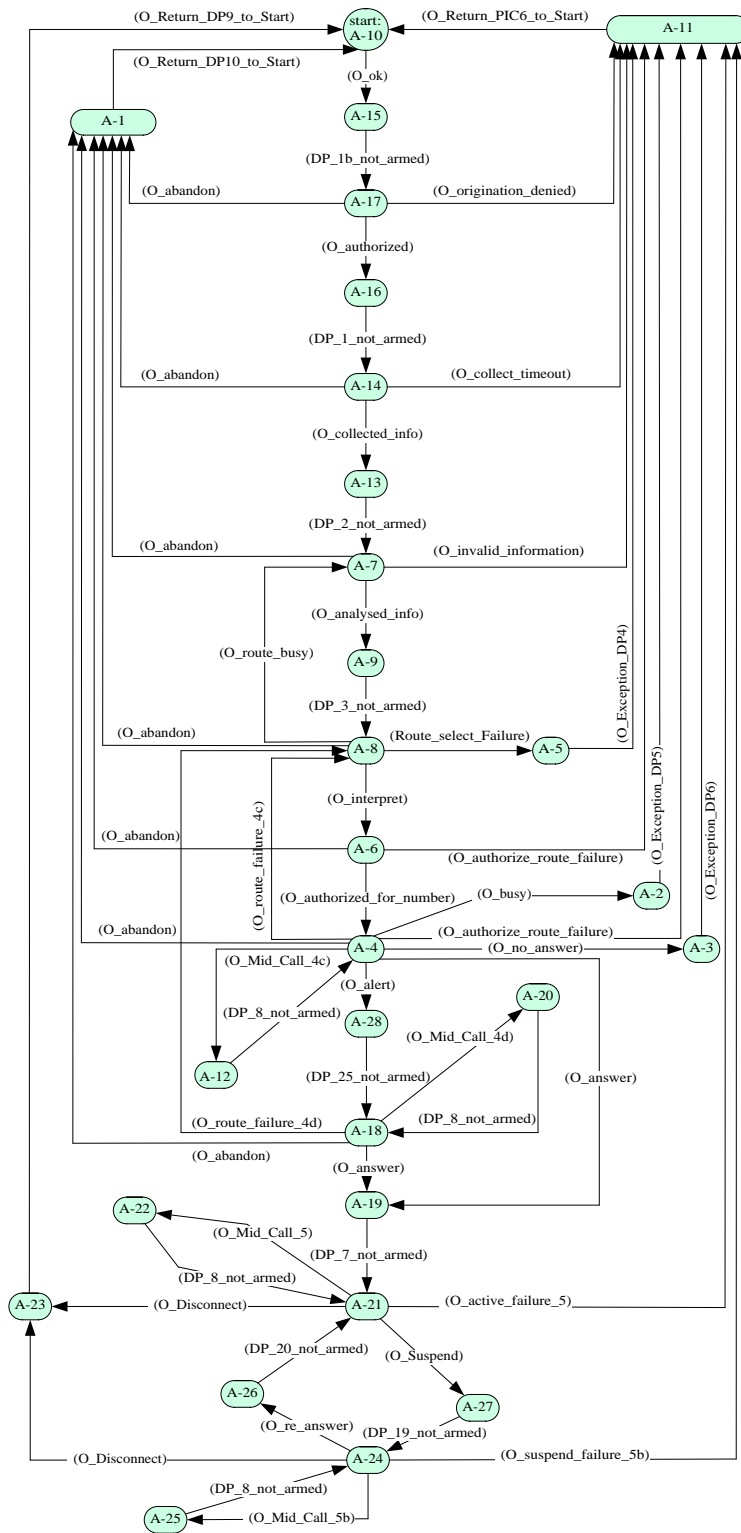


Abb. 23: Minimalautomat "O-BCSM"

Vergleicht man das Ergebnis aus Abb. 23 mit der graphischen Darstellung des O-BCSM von ITU-T (siehe Abb. 4), ist zu erkennen, daß hier lediglich die beiden Kanten fehlen, die aus den erwähnten Gründen in der Modellierung nicht enthalten sind. Die lokale Verhaltensweise des O-BCSM in diesem Modell entspricht somit den Vorgaben von ITU-T.

3.2.2.4. T-BCSM

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1> + 1<2>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
SYSTEMS: 2<SYSTEM_READY>
```

Erreichbarkeitsgraph bcsms2c

4996 Markierungen berechnet.
 (0 Totmarkierungen)
 (0 Pseudoschaltsschritte)
 (16512 Schaltsschritte)

Faktorgraph von Erreichbarkeitsgraph bcsms2c
 (1 Zusammenhangskomponente gefunden)
 (1 Randkomponente)

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
T_get_initiate	GET_INITIATE_7A
T_authorized	AUTHORIZED_7B
T_termination_denied	TERMINATION_DENIED_7B
DP_12_not_armed	DP_12_NOT_ARMED
DP_21_not_armed	DP_21_NOT_ARMED
T_Called_Party_Busy	BUSY_8A
T_Called_Party_Alert	ALERT_8B
T_Called_Party_Answer	ANSWER_8B
DP_13_not_armed	DP_13_NOT_ARMED_1
DP_22_not_armed	DP_22_NOT_ARMED
DP_24_not_armed	DP_24_NOT_ARMED
DP_23_not_armed	DP_23_NOT_ARMED
T_no_answer	NO_ANSWER_9
T_answer	ANSWER_9
DP_14_not_armed	DP_14_NOT_ARMED
DP_15_not_armed	DP_15_NOT_ARMED
T_active_failure_10	T_ACTIVE_FAILURE_10A
T_suspend_failure_10b	T_SUSPEND_FAILURE_10B
mid_call_10	MID_CALL_10A
T_Disconnect_oder_Abandon_10	ABANDON_10, OTDISCONNECT_10

T_disconnect_10b	TIMEOUT_10B, OTDISCONNECT_10B
DP_16_not_armed	DP_16_NOT_ARMED
T_Return_DP17_To_Start	TD_17_1A, TD_17_1B
T_Return_PIC11_To_Start	EXCEPTION_11_A, EXCEPTION_11_B
T_suspend	TODISCONNECT_10
T_re_answer	REANSWER_10B
T_Called_Party_Not_Busy	NOT_BUSY_8A
DP_26_not_armed	DP_26_NOT_ARMED
T_call_rejected_9	CALL_REJECTED_9
SS7_failure	SS7_FAILURE_8B
presentation_failure_8b	PRESENTATION_FAILURE_8B
T_abandon	ABANDON_7B, ABANDON_8B, ABANDON_8A, ABANDON_9
DP_18_not_armed	DP_18_NOT_ARMED_2, DP_18_NOT_ARMED_1
T_abandon_in_no_answer_DP	DP_14_NOT_ARMED_2
T_abandon_in_busy_DP	DP_13_NOT_ARMED_2
Epsilon	alle restlichen Transitionen

Der Homomorphismus enthält alle Transitionen des T-BCSMs.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bcsms2c
20 Zustände berechnet.
(38 Kanten, 35 Anschriften)
```

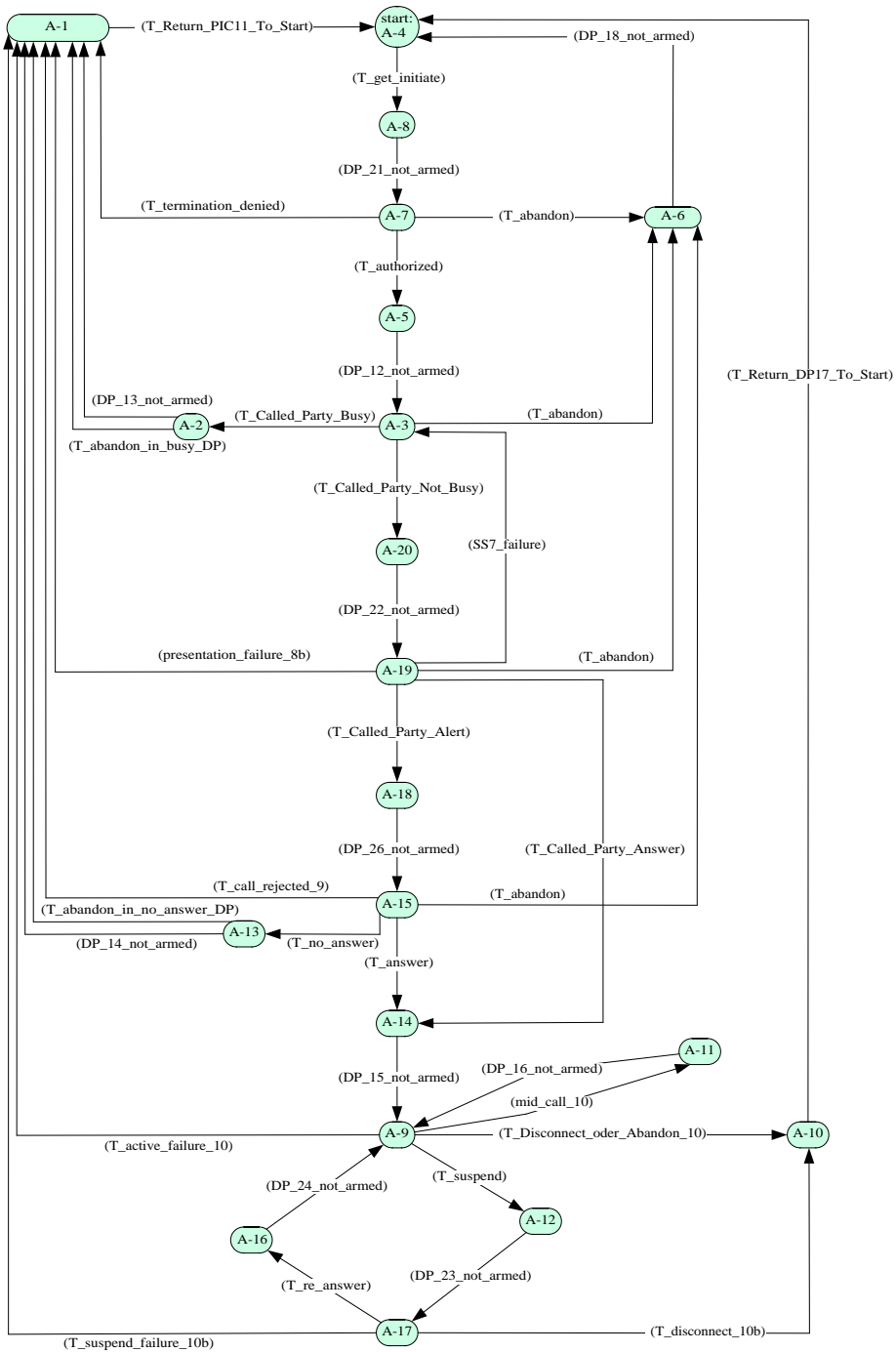


Abb. 24: Minimalautomat "T-BCSM"

Wie bereits beim O-BCSM stimmt auch das gefundene lokale Verhalten des T-BCSMs mit der Automaten­darstellung von ITU-T (Abb. 6) überein.

3.2.2.5. Kommunikation zwischen O- und T-BCSM

Der Untersuchung liegt die Erreichbarkeitsanalyse aus Abschnitt 3.2.2.3. zugrunde.

Auch die Kommunikation zwischen O- und T-BCSM ist in den uns zur Verfügung stehenden Unterlagen nicht beschrieben. Somit kann lediglich überprüft werden, ob die im Modell stattfindende Kommunikation sinnvoll abläuft, sie kann nicht verifiziert werden.

Dazu wird folgender Homomorphismus definiert:

Block	Transitionen
O_send_to_T_initiate	initiate_4c
T_send_to_O_busy	TD_13_a
T_send_to_O_R_no_answer	DP_14_not_armed_1
T_send_to_O_V_no_answer	presentation_failure_8b, call_rejected_9, termination_denied_7b
T_send_to_O_alerted	answer_9
T_send_to_O_accept	accept_10
T_exception_T_send_to_O_disconnect	T_suspend_failure_10b
O_exception_O_send_to_T_disconnect	O_active_failure_5, O_suspend_failure_5b
T_exception_T_send_to_O_2disconnect	T_active_failure_10a
T_send_to_O_disconnect	TP_10a_1, TP_10b_1, TDisconnect_10, timeout_10b
O_send_to_T_disconnect	OTdisconnect_5, OTdisconnect_5b, TP_5_1, TP_5b_1
T_send_to_O_reanswer	reanswer_10b
O_send_to_T_reanswer	TP_5b_4
O_send_to_T_abandon	abandon_4c_2, abandon_4d, route_failure_4d
T_send_to_O_abandon	abandon_7b, abandon_8a, abandon_8b, abandon_9, TD_13_b, DP_13_not_armed_2, abandon_10, DP_14_not_armed_2
Epsilon	alle restlichen Transitionen

Damit der resultierende Minimalautomat nicht zu unübersichtlich wird, ordnet man nicht jede Transition, die eine Nachricht schickt, einem eigenen Block zu.

Im O-BCSM werden folgende Transitionen zu jeweils einem Block zusammengefaßt:

- Transitionen, die ein Auflegen vor dem Zustandekommen der Datenphase repräsentieren (Block *O_send_to_T_abandon*)
- Transitionen, die ein normales Beenden einer Verbindung nach Durchlaufen der Datenphase widerspiegeln (Block *O_send_to_T_disconnect*)
- Transitionen, die schalten, wenn die Datenphase aufgrund eines aufgetretenen Fehlers beendet wird (Block *O_exception_O_send_to_T_disconnect*)

Im T-BCSM werden Nachrichten schickende Transitionen wie folgt zusammengefaßt:

- Transitionen, die aufgrund eines anliegenden *O_T_abandon* eine Quittung, ein *T_O_abandon* schicken (Block *T_send_to_O_abandon*)
- Transitionen, die ein normales Beenden einer Verbindung nach durchlaufener Datenphase widerspiegeln (Block *T_send_to_O_disconnect*)

- Transitionen, die ein $T_O_no_answer$ schicken und einen Ausnahmefall repräsentieren (Block $T_send_to_O_V_no_answer$)

Die Berechnung des Minimalautomaten liefert folgendes:

```
Minimalautomat bcsms2c
12 Zustaeude berechnet.
(27 Kanten, 15 Anschriften)
```

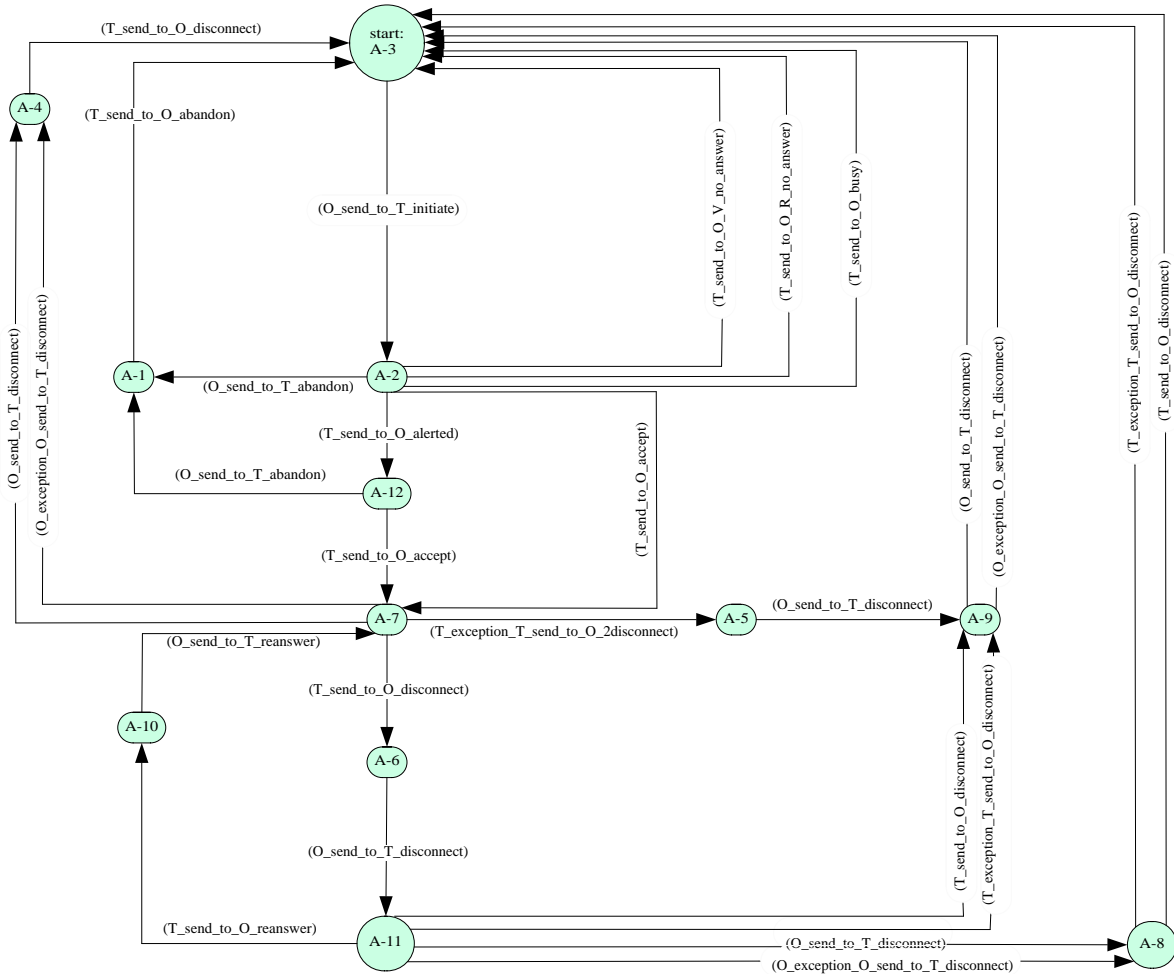


Abb. 25: Minimalautomat "Kommunikation"

Am Minimalautomaten aus Abb. 25 ist nachzuvollziehen, daß die Kommunikation zwischen O- und T-BCSM sinnvoll abläuft.

3.2.2.6. Phasentrennung und Lebendigkeitseigenschaft

Phasentrennung bedeutet, daß von zwei Kommunikationspartnern einer erst in eine neue Datenphase eintreten kann, wenn er selbst und der andere die vorherige verlassen haben. Als Lebendigkeitseigenschaft wird überprüft, daß aus jedem Systemzustand heraus immer wieder eine Datenphase erreichbar ist. Um bei Lebendigkeitseigenschaften vom Verhalten in

einer vergrößerten Sichtweise, erzeugt durch einen Homomorphismus, auf das Verhalten im Erreichbarkeitsgraphen schließen zu können, muß der Homomorphismus, wie bereits im Abschnitt 1.1.2. erläutert, die Eigenschaft der Schlichtheit besitzen.

Der Untersuchung liegt die Erreichbarkeitsanalyse aus Abschnitt 3.2.2.1.1. zugrunde.

Zur Betrachtung dieser globalen Verhaltensweisen wird folgender Homomorphismus definiert:

Block	Transitionen
T_Eintritt_in_Datenphase	accept_10
T_Austritt_aus_Datenphase	OTdisconnect_10, OTdisconnect_10b, timeout_10b, T_active_failure_10a, T_suspend_failure_10b
O_Eintritt_in_Datenphase	answer_4c, answer_4d
O_Austritt_aus_Datenphase	OTdisconnect_5, OTdisconnect_5b, O_active_failure_5, O_suspend_failure_5b, Tdisconnect_5b
Epsilon	alle restlichen Transitionen

Der Eintritt in die Datenphase erfolgt also beim Schicken bzw. Empfangen von T_O_accept . Der Austritt aus der Datenphase ist mit den Transitionen verknüpft, die alternativ eine Datenphase, ob normal oder aufgrund eines Fehlers, beenden können.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bcsms2c
6 Zustände berechnet.
(8 Kanten, 4 Anschriften)
```

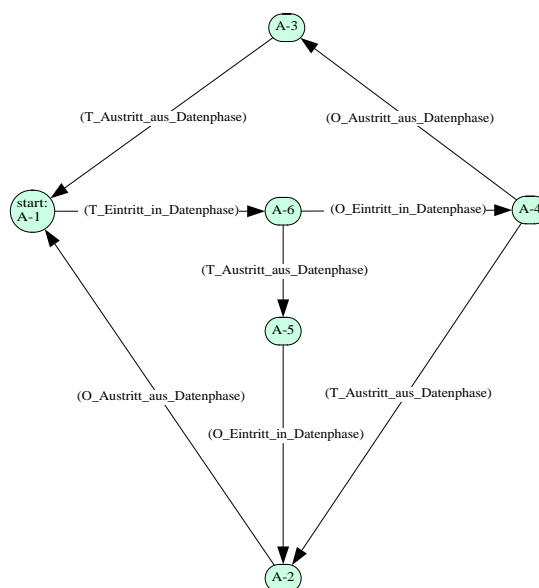


Abb. 26: Minimalautomat "Phasentrennung und Lebendigkeitseigenschaft"

Am Minimalautomaten in Abb. 26 ist zu sehen, daß die Phasentrennung gilt: Zuerst muß T in die Datenphase eintreten. Um erneut eintreten zu können, muß ein Weg durchlaufen werden, auf dem T wieder aus der Datenphase austritt und O eine komplette Datenphase durchläuft, d.h. auch aus der Datenphase austritt. Ebenso kann O nach einem ersten Eintreten in eine Datenphase erst erneut eintreten, wenn es aus der alten wieder ausgetreten ist. Da auf jedem Weg zwischen einem Eintreten von O in die Datenphase und dem nächsten Eintreten eine Kante *T_Eintritt_in_Datenphase* liegt, kann unter Berücksichtigung der Tatsache, daß eine neue Datenphase nur von T begonnen werden kann, O nie in eine Datenphase eintreten, solange T eine alte noch nicht verlassen hat.

Daß auch die genannte Lebendigkeitseigenschaft, also die Eigenschaft einer immer wieder möglichen Datenphase, gilt, ist daran zu erkennen, daß es aus jedem Zustand einen Weg zu einem anderen Zustand gibt, aus dem einer der beiden Teilnehmer in die Datenphase eintreten kann. Außerdem ist der benutzte Homomorphismus schlicht, da der Erreichbarkeitsgraph nur aus einer Zusammenhangskomponente besteht.

An diesem Minimalautomat ist auch die starke Synchronisation der beiden Teilnehmer zu erkennen. Tritt nämlich T in die Datenphase ein und gleich darauf wieder aus, so muß auch O ein- und austreten, bevor T eine neue Phase beginnen kann.

3.2.2.7. Verhalten bei aktiviertem Call Hold-Dienst

Diese Untersuchung wird durchgeführt, um Unterschiede zwischen der Dynamik dieses Modells und der nachfolgender aufzuzeigen.

Der Untersuchung liegt dieselbe Erreichbarkeitsanalyse wie im Abschnitt 3.2.2.1.2. zugrunde.

Zur Definition des Homomorphismus wird ein Block mit nur einer Transition gebildet:

Block	Transitionen
T_accept	accept_10
Epsilon	alle restlichen Transitionen

Zusätzlich werden die beiden Stellen *Entry_PIC_10_after_accept* und *On_hold* eingeblendet. Außerdem sind wegen der nun zu erwartenden Q-Markierungen auch die erste und dritte Komponente der Zustandsübergänge im Minimalautomaten wichtig und werden deshalb nicht ausgeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bcsms2c
1 Zustand berechnet.
(1 Kante, 1 Anschrift)
```

Q-Markierungen : bcsms2c

Q-::

Q-1

ENTRY_PIC_10_AFTER_ACCEPT: 1<1,1,2,1,::>

Ende Q-Markierungen : bcsms2c

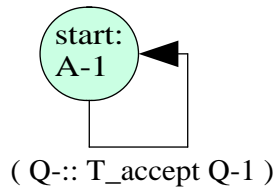


Abb. 27: Minimalautomat "Call Hold-Dienst"

An den Q-Markierungen ist zu erkennen, daß immer, wenn *accept_10* schaltet, auf der Stelle *On_hold* keine Marke liegt. Das bedeutet, daß, wenn der Anrufer eine erste Verbindung zum Angerufenen auf Warteposition setzt und ihn erneut anzurufen versucht, diese zweite Verbindung nie in die aktive Phase gelangen kann. Dazu wäre das Schalten von *accept_10* notwendig und dies geschieht - wie an obigem Ergebnis zu erkennen - nie, wenn eine Verbindung auf Warteposition liegt.

Die Bedeutung dieser Verhaltensweise wird erst später bei der Untersuchung eines weniger strikten Modells deutlich.

3.2.2.8. Besetzt-Verhalten

Auch das Besetzt-Verhalten wird untersucht, um Unterschiede zwischen der Dynamik dieses Modells und der anderer aufzuzeigen.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
SYSTEMS: 2<SYSTEM_READY>
```

Erreichbarkeitsgraph bcsms2c

```
2029 Markierungen berechnet.
(0 Totmarkierungen)
(0 Pseudoschaltsschritte)
(5823 Schaltsschritte)
```

```
Faktorgraph von Erreichbarkeitsgraph bcsms2c
(1 Zusammenhangskomponente gefunden)
(1 Randkomponente)
```

Zur Definition des Homomorphismus wird ein Block mit nur einer Transition gebildet:

Block	Transitionen
Verbindung_1_2_busy	busy_8a
Epsilon	alle restlichen Transitionen

Die Stelle *Entry_PIC_10_after_accept* und alle drei Komponenten der Zustandsübergänge im Minimalautomaten werden eingeblendet.

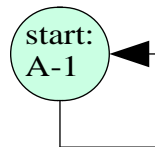
Die Berechnung des Minimalautomaten ergibt folgendes Ergebnis:

```
Minimalautomat bcsms2c
1 Zustand berechnet.
(1 Kante, 1 Anschrift)

Q-Markierungen : bcsms2c

Q-::

Ende Q-Markierungen : bcsms2c
```



(Q-:: Verbindung_1_2_busy Q-::)

Abb. 28: Minimalautomat "Besetzt-Verhalten"

Das Ergebnis zeigt, daß bei der gewählten Konstellation von Anrufer und Angerufenem die Transition *busy_8a*, die ein Schicken von *T_O_busy* zur Folge hat, nur schalten kann, wenn keine Verbindung zwischen Anrufer und Angerufenem im T-BCSM aktiv ist. Dies scheint widersinnig, ist jedoch einfach zu erklären.

Damit der Anrufer eine erste Verbindung zum Angerufenen beenden und einen neuen Aufbauwunsch schicken kann, muß der Angerufene wegen der strikten Synchronisation der Teilnehmer die aktive Datenphase bereits verlassen haben (keine Marke mehr auf der Stelle *Entry_PIC_10_after_accept*). Dies muß er tun, um dem Anrufer eine Quittung schicken zu können, woraufhin dieser in den Grundzustand gelangt, einen neuen Aufbauwunsch sendet und als Antwort *T_O_busy* erhält.

3.2.3. Diskussion der ersten Spezifikation

In der oben vorgestellten Spezifikation des BCSMs sind folgende das Transportsystem bzw. den "End-zu-End-Schlupf" betreffende Prämissen enthalten:

- Der Einbau von End-zu-End-Handshakes bedingt eine strikte Synchronisation bei dem durch Handshake kontrollierten Nachrichtenaustausch. Das bedeutet, daß globale Flußkontrolle vorliegt.
- Da das Transportsystem nur aus den beiden Schnittstellen *O_BCSM_to_T_BCSM* und *T_BCSM_to_O_BCSM* besteht und mit Verbotskanten darauf zugegriffen wird, ist implizit vorausgesetzt, daß eine abgeschickte Nachricht sofort beim Empfänger anliegt.
- Durch den End-zu-End-Handshake und das sofortige Anliegen gesendeter Nachrichten ist sichergestellt, daß bei einem Verbindungsaufbau keine alten Nachrichten mehr auf den Schnittstellen liegen.

Da in der Standardisierung keine Aussage über ein Transportsystem oder den "End-zu-End-Schlupf" gemacht ist, wurde zunächst dieses möglichst einfache Modell gewählt, d.h. ein Transportsystem ohne Intelligenz, bestehend aus den Stellen *O_BCSM_to_T_BCSM* und *T_BCSM_to_O_BCSM*.

Um untersuchen zu können, wie weit sich die strikte Synchronisation der Teilnehmer auf das Verhalten des BCSMs und eingebaute Dienste auswirkt, wird im folgenden ein weniger striktes Modell entwickelt.

3.3. Zweite Spezifikation

Bei der Beschreibung des Modells wird nur auf die Modifikationen gegenüber der ersten Spezifikation eingegangen. Der Vorspann und die Produktnetze dieser Spezifikation befinden sich im Anhang B.

3.3.1. Ziel dieser Spezifikation

Ziel dieser Spezifikation ist es, auf die Prämissen des sofortigen Anliegens gesendeter Nachrichten beim Empfänger und strikte Synchronisation bzw. globale Flußkontrolle zu verzichten.

3.3.2. Mögliche Vorgehensweisen

3.3.2.1. Verzögerungen beim Nachrichtentransport

Liegen gesendete Nachrichten nicht sofort beim Empfänger an und wird auf strikte Synchronisation verzichtet, wird eine Folge sein, daß vor einem neuen Verbindungsaufbau noch vorhandene Nachrichten zu der alten Verbindung zwischen denselben Teilnehmern von den Schnittstellen entfernt werden müssen. Man muß also einen Mechanismus finden, der alte Nachrichten zu einer Verbindung von neuen unterscheiden kann, damit die Phasentrennung gewährleistet werden kann.

Eine Möglichkeit wäre, die beiden Teilnehmer bei einem Verbindungsabbau einen Disconnect Handshake mit dem Transportsystem durchführen zu lassen. Damit hätte das Transportsystem die Information, wann alte Nachrichten nicht mehr benötigt werden und könnte somit die Schnittstellen richtig bereinigen. Der Nachteil dieser Lösung besteht darin, daß bei der expliziten Modellierung eines Transportsystems der Zustandsraum des Modells sehr stark anwachsen würde.

Ein anderer Lösungsvorschlag ist, zu jeweils einer Verbindung einen Alternating Bit-Mechanismus einzuführen. Das bedeutet, daß jeder Nachricht zu einer Verbindung ein Bit angehängt ist. Man hängt zum Beispiel den Nachrichten zur ersten Verbindung zwischen zwei bestimmten Teilnehmern das Bit 0, der zweiten das Bit 1, der dritten wieder das Bit 0 an usw. Dadurch kann jeder Teilnehmer selbst feststellen, welche der für ihn bestimmten Nachrichten er nicht mehr benötigt. Das ist dann der Fall, wenn er die erste Nachricht mit einem anderen Bit, das sei o.B.d.A. 1, empfängt. Zu diesem Zeitpunkt kann er alle für ihn bestimmten Nachrichten zu dieser Verbindung, die Bit 0 haben, von seiner Schnittstelle entfernen. Auch mit diesem Mechanismus ist also die Phasentrennung gewährleistet. Da der Zustandsraum weniger als bei der ersten aufgeführten Möglichkeit anwächst - er verdoppelt sich -, wird bei der zweiten Spezifikation diese Lösung gewählt.

3.3.2.2. Vermeidung der Synchronisation

Bei der ersten Spezifikation liegt durch den End-zu-End-Handshake eine strikte Synchronisation der Endteilnehmer vor und damit verbunden auch eine globale Flußkontrolle. Der End-zu-End-Handshake ist notwendig, um das unkontrollierte Ansammeln von Nachrichten auf den Schnittstellen zu unterbinden.

Um die strikte Synchronisation zu vermeiden, könnte man die Schnittstelle als nicht überschreibbaren Puffer beliebiger, fester Größe N implementieren. Das bedeutet, daß eine Synchronisation nicht sofort, sondern erst nach N Schritten stattfindet. Der Sender kann damit N Nachrichten in den Puffer stellen, ohne daß der Empfänger etwas herausnimmt. Ist der Puffer voll, wird der Sender gezwungen, auf eine Aktion des Empfängers zu warten. Damit liegt auch hier globale Flußkontrolle vor.

Eine andere Lösungsmöglichkeit ist, die Schnittstellen als überschreibbare Puffer mit beliebiger, fester Größe N zu realisieren. D.h., schickt ein Teilnehmer die $(N+1)$ -te Nachricht eines bestimmten Typs, so wird eine Nachricht dieses Typs und eine eventuell dazugehörige (siehe z.B. *T_O_disconnect*, *T_O_reanswer* bzw. *O_T_initiate*, *O_T_abandon*) gelöscht. Damit hat man keinerlei Synchronisation und dadurch auch keine globale Flußkontrolle. Um die Korrektheit dieser Vorgehensweise sicherzustellen, könnte man einen Induktionsbeweis über die Idempotenz führen.

Für die zweite Spezifikation wird diese Lösungsmöglichkeit gewählt, da die Teilnehmer hierbei völlig unabhängig voneinander agieren können und keinerlei globalen Beschränkungen unterliegen. Da der Zustandsraum des Erreichbarkeitsgraphen möglichst klein gehalten werden soll, werden überschreibbare Puffer der Größe eins gewählt.

3.3.3. Konkrete Modellierung

3.3.3.1. Alternating Bit-Mechanismus

Wie oben erläutert, wird ein Alternating Bit-Mechanismus benutzt, um Phasentrennung zu gewährleisten. Dieser liefert dem Teilnehmer ein Erkennungsmerkmal für "alte" eingehende Nachrichten, die er in einer neuen Phase selbst löscht.

Um diesen Mechanismus zu realisieren, wird im O-BCSM bzw. im T-BCSM eine Stelle *O_Alt_Bit* bzw. *T_Alt_Bit* eingeführt, in der pro Tripel (*user*, *line_u*, *number*) ein Bit abgelegt wird.

Wird im O-BCSM ein neuer Verbindungsaufbauwunsch zu einem Tripel (*user*, *line_u*, *number*) geschickt, so wird das Bit von der Stelle *O_Alt_Bit* entfernt und $(Bit+1) \text{ modulo } 2$ sowohl an die Verbindungsaufbauwunschnachricht angehängt, als auch auf der Stelle *O_Alt_Bit* abgelegt. Damit steht in *O_Alt_Bit* immer das Bit zur aktuellen Verbindung zur Verfügung. Erhält der Anrufer die Antwort auf seinen neuen Verbindungsaufbauwunsch, so kann er alle für sich bestimmten Nachrichten zu der gleichen

alten Verbindung von der Schnittstelle entfernen. Daß es sich um alte Nachrichten handelt, erkennt er am anderen Bit. Weiterhin ist er auch sicher, daß keine Nachrichten zur alten Verbindung mehr nachkommen, da er bereits die erste Nachricht zur neuen Verbindung vom Angerufenen erhalten hat.

Wird im T-BCSM ein neuer Verbindungsaufbauwunsch zu einem Tripel (*user*, *line_u*, *number*) empfangen, so wird das zugehörige Bit aus *T_Alt_Bit* entfernt und $(Bit+1) \text{ modulo } 2$ zurückgelegt. Ein Verbindungsaufbauwunsch kann nur entgegengenommen werden, falls das modifizierte Bit dem der Nachricht angehängten entspricht. Somit können dann alle alten, zu dieser Verbindung gehörenden Nachrichten, die für den Angerufenen (*number*) bestimmt sind, von der Schnittstelle entfernt werden. Erkennungsmerkmal ist auch hier das Bit. Weiterhin ist man dabei sicher, daß keine Nachrichten zur alten Verbindung mehr nachkommen können, da der Anrufer (*user*) bereits einen neuen Verbindungsaufbauwunsch geschickt hat, was impliziert, daß er die alte Verbindung beendet haben muß.

Im Fall, daß der Anrufer zwischen dem Abschicken eines Aufbauwunsches und dem Eintreten in die aktive Phase aufliegt und denselben Teilnehmer erneut anruft, während der Angerufene den ersten Aufbauwunsch akzeptiert und als Antwort auf den zweiten Aufbauwunsch ein "besetzt" geschickt wird, reicht der "normale" Alternating Bit-Mechanismus nicht aus, um eine vollständige Trennung von Nachrichten zu zwei aufeinanderfolgenden Verbindungen zwischen denselben Teilnehmern gewährleisten zu können.

Der Angerufene ist nicht gezwungen, auf das Auflegen des Anrufers zu reagieren, und kann zu dieser, für ihn noch bestehenden Verbindung weiter Nachrichten schicken. Beendet der Anrufer die zweite Verbindung und versucht ein drittes Mal denselben Teilnehmer anzurufen, so befindet er sich in derselben Phase wie beim ersten Mal. Er kann nun ein "besetzt" zu dieser Verbindung nicht von Nachrichten zur ersten Verbindung unterscheiden, da die Nachrichten dasselbe Bit enthalten. Der Anrufer könnte also fälschlicherweise statt des "besetzt" das "akzeptiert" zu seinem ersten Aufbauwunsch als Antwort auf seinen dritten Aufbauwunsch interpretieren.

Um dieses Fehlverhalten zu vermeiden, wird der Anrufer beim Schicken einer Antwort auf seinen Aufbauwunsch dazu veranlaßt, die Phase nicht zu wechseln, falls der Angerufene eine ältere Verbindung zum selben Anrufer noch nicht beendet hat.

Dies wird realisiert, indem auf der neuen Stelle *Active* im T-BCSM die kennzeichnenden Marken der Verbindungen abgelegt werden, die in die aktive Phase gelangen. Beim Beenden der aktiven Phase wird die entsprechende Marke wieder von der Stelle *Active* abgezogen.

Dem Anrufer wird die Beibehaltung der Phase mitgeteilt, indem der Antwort auf seinen Aufbauwunsch ein Flag mit dem Wert 1 angehängt wird, falls zur selben Verbindung auf der Stelle *Active* eine Marke liegt.

3.3.3.2. Schnittstellen als überschreibbare Puffer

Um eine völlige Unabhängigkeit von Anrufer und Angerufenem zu erreichen, wird bei der Art der Schnittstellen, wie oben erwähnt, ein überschreibbarer Puffer der Größe eins gewählt. Da die Nachrichten auf der Schnittstelle nach gewissen Kriterien durchsucht werden müssen, können sie nicht mehr als einzelne Token dort liegen. Sie sind daher in einer Liste zusammengefaßt.

Die Notwendigkeit, einen Mechanismus wie den überschreibbaren Puffer einzuführen, entsteht dadurch, daß sich auf den Schnittstellen in folgenden Fällen Nachrichten ansammeln:

1. Anrufer: schickt Verbindungsaufbauwunsch, legt auf, schickt Verbindungsaufbauwunsch,... Das entspricht einem Ansammeln von $O_T_initiate$, $O_T_abandon$, $O_T_initiate$... auf der Schnittstelle $O_BCSM_to_T_BCSM$. In diesem Fall wird $O_T_initiate + O_T_abandon$ zu ϵ überschrieben. Zu beachten ist hierbei, daß der Anrufer bei einem erneuten Verbindungsaufbauwunsch dann nicht die Phase, d.h. das Alternating Bit, wechseln darf, da der Angerufene keine Kenntnis von diesem aktuellen Verbindungsaufbauwunsch erhält und sein Bit nicht ändert.
2. Angerufener: legt auf, hebt innerhalb eines gewissen Zeitrahmens wieder ab, legt auf ... Das entspricht einem Ansammeln von $T_O_disconnect$, $T_O_reanswer$, $T_O_disconnect$... auf der Schnittstelle $T_BCSM_to_O_BCSM$. Hier wird $T_O_disconnect + T_O_reanswer$ zu ϵ und auch $T_O_reanswer + T_O_disconnect$ zu ϵ überschrieben.

Bemerkungen

1. Sowohl im ersten wie auch im zweiten Fall wurde zunächst eine weniger strikte Vorgehensweise gewählt. Es wurde nur $O_T_initiate.O_T_abandon + O_T_initiate$ zu $O_T_initiate$ und $T_O_disconnect.T_O_reanswer + T_O_disconnect$ zu $T_O_disconnect$ überschrieben. Durch die jetzige, striktere Vorgehensweise erhält man aber etwa ein Drittel weniger Markierungen im Erreichbarkeitsgraphen.
2. Weitere Maßnahmen zur Verkleinerung des Zustandsraums:
 - Festlegung einer lexikographischen Sortierung der Schnittstellenliste
 - Überschreibung von $T_O_reanswer + T_O_disconnect.T_O_disconnect$ zu $T_O_disconnect$ (im Fall eines $T_active_failure$)
 - Überschreiben von $T_O_alerted + T_O_accept$ zu T_O_accept
3. Alle diese Maßnahmen schränken die Möglichkeiten von Anrufer bzw. Angerufenem nicht ein, da eine vollständige Analyse gemacht wird und damit die erste Nachricht jeweils auch schon abgenommen worden sein kann und die zweite Nachricht so kein Überschreiben zur Folge hat. Zu bemerken ist aber, daß diese Maßnahmen den Erreichbarkeitsgraphen von ursprünglich 9385 Markierungen auf 4903 Markierungen verkleinerten.
4. Für den Zugriff auf die Schnittstellen sind nun Funktionen notwendig. Diese haben Aufgaben wie Durchsuchen einer Schnittstelle nach bestimmten Nachrichten, sortiertes

Einfügen neuer Nachrichten, Löschen einzelner oder mehrerer Nachrichten und Überschreiben bestimmter Nachrichten durch andere. Die Funktionen sind im Vorspann definiert und bezüglich ihrer Aufgabe kommentiert.

3.3.4. Analyse und Verifikation

Zur Analyse und Verifikation dieses Modells werden, wie schon für die erste Spezifikation, zu mehreren Anfangsmarkierungen die Erreichbarkeitsgraphen berechnet. Anschließend werden Homomorphismen definiert und jeweils der zugehörige Minimalautomat berechnet. Dadurch können folgende Verhaltensweisen aus der Dynamik des Modells extrahiert werden:

- lokale Verhaltensweisen
 1. Anrufer
 2. Angerufener
 3. O-BCSM
 4. T-BCSM
- globale Verhaltensweisen
 5. Phasentrennung und Lebendigkeitseigenschaft
- vergleichende Homomorphismen
 6. Besetzt-Verhalten
 7. Gesamtverhalten

3.3.4.1. Anrufer

Grundlage für die Untersuchung des Verhaltens eines Anrufers ist folgende Erreichbarkeitsanalyse:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_ALT_BIT: 1<1,1,2,0>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_ALT_BIT: 1<1,1,2,0>
T_BCSM_TO_O_BCSM: 1<::>
```

Erreichbarkeitsgraph bcsms2c-s1

```
4903 Markierungen berechnet.
(0 Totmarkierungen)
(0 Pseudoschaltsschritte)
(13144 Schaltsschritte)
```

```
Faktorgraph von Erreichbarkeitsgraph bcsms2c-s1
(1 Zusammenhangskomponente gefunden)
(1 Randkomponente)
```

Der Homomorphismus ist genauso definiert wie beim ersten Modell und auch das Ergebnis der Berechnung des Minimalautomaten ist - wie erwartet - das gleiche (vgl. Abschnitt 3.2.2.1.1.).

Aufgrund der großen Komplexität dieses Modells ist es - im Gegensatz zum ersten Modell - nicht möglich, das Verhalten eines Anrufers mit abonniertem Call Hold-Dienst in annehmbarer Zeit zu untersuchen.

3.3.4.2. Angerufener

Der Untersuchung liegt die Erreichbarkeitsanalyse aus dem Abschnitt 3.3.4.1. zugrunde. Der Homomorphismus wird folgendermaßen definiert:

Block	Transitionen
abheben	answer_8b, answer_9
auflegen	DP_17_not_armed, T_active_failure_10a, T_suspend_failure_10b
Epsilon	alle restlichen Transitionen

Wie erwartet, ist das Ergebnis identisch mit dem im ersten Modell (vgl. Abschnitt 3.2.2.2.).

3.3.4.3. O-BCSM

Der Untersuchung des Verhaltens des O-BCSMs liegt die Erreichbarkeitsanalyse aus dem Abschnitt 3.3.4.1. zugrunde. Aus Komplexitätsgründen ist es nicht möglich, eine Erreichbarkeitsanalyse mit derselben Konfiguration wie im ersten Modell durchzuführen. Es wird die einfachste Konstellation von Anrufer und Angerufenem - ein Anrufer und ein Angerufener mit jeweils einem *'system_ready'* im O- und T-BCSM - gewählt. Dadurch kann es nicht vorkommen, daß ein Aufbauwunsch mit "besetzt" beantwortet wird. Im Minimalautomaten wird deshalb kein dem *Busy DP* entsprechender Zustand enthalten sein.

Der Homomorphismus ist genauso definiert wie im ersten Modell (vgl. Abschnitt 3.2.2.3.). Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimalautomat bscms2c-s1  
  
27 Zustände berechnet.  
(53 Kanten, 41 Anschriften)
```

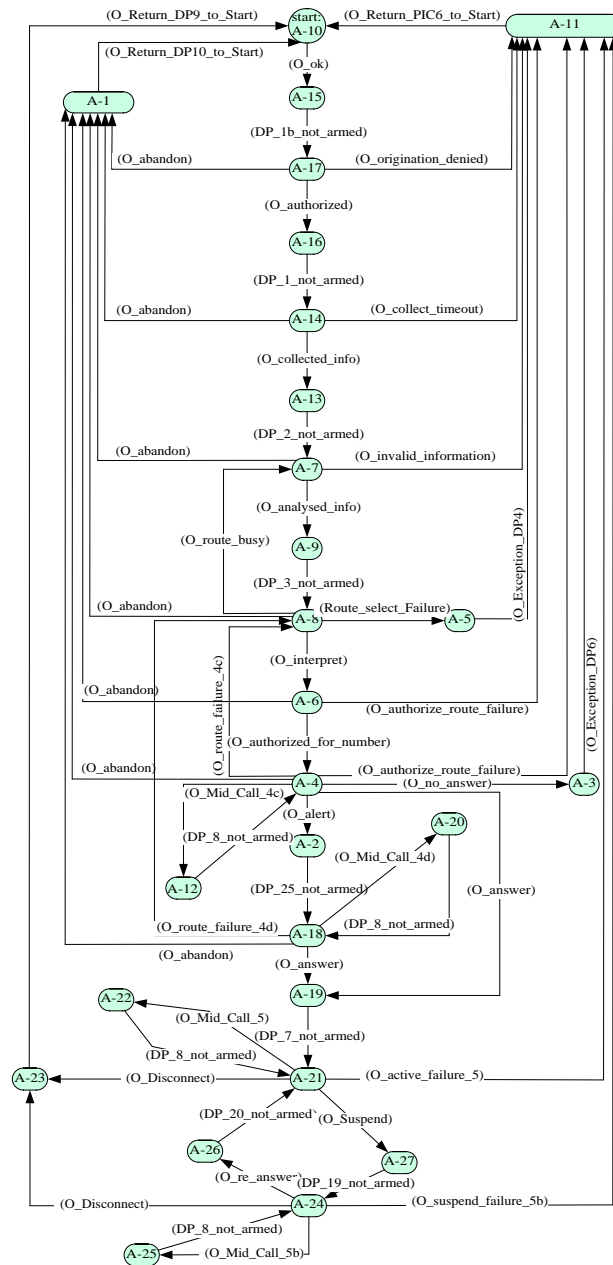


Abb. 29: Minimalautomat "O-BCSM"

Der in Abb. 29 dargestellte Minimalautomat stimmt bis auf das erwähnte Fehlen eines Zustands für den *Busy DP* und der entsprechenden Kanten mit dem im ersten Modell gefundenen überein (vgl. Abschnitt 3.2.2.3.).

3.3.4.4. T-BCSM

Der Untersuchung liegt die Erreichbarkeitsanalyse aus dem Abschnitt 3.3.4.1. zugrunde. Wie bei der Untersuchung des O-BCSMs, muß auch diese Analyse aus Komplexitätsgründen mit der einfachsten Konfiguration durchgeführt werden.

Der Homomorphismus ist genauso definiert wie im ersten Modell (vgl. Abschnitt 3.2.2.4.).

Die Berechnung des Minimalautomaten ergibt folgendes:

Minimalautomat bcsms2c-s1

19 Zustände berechnet.
(35 Kanten, 32 Anschriften)

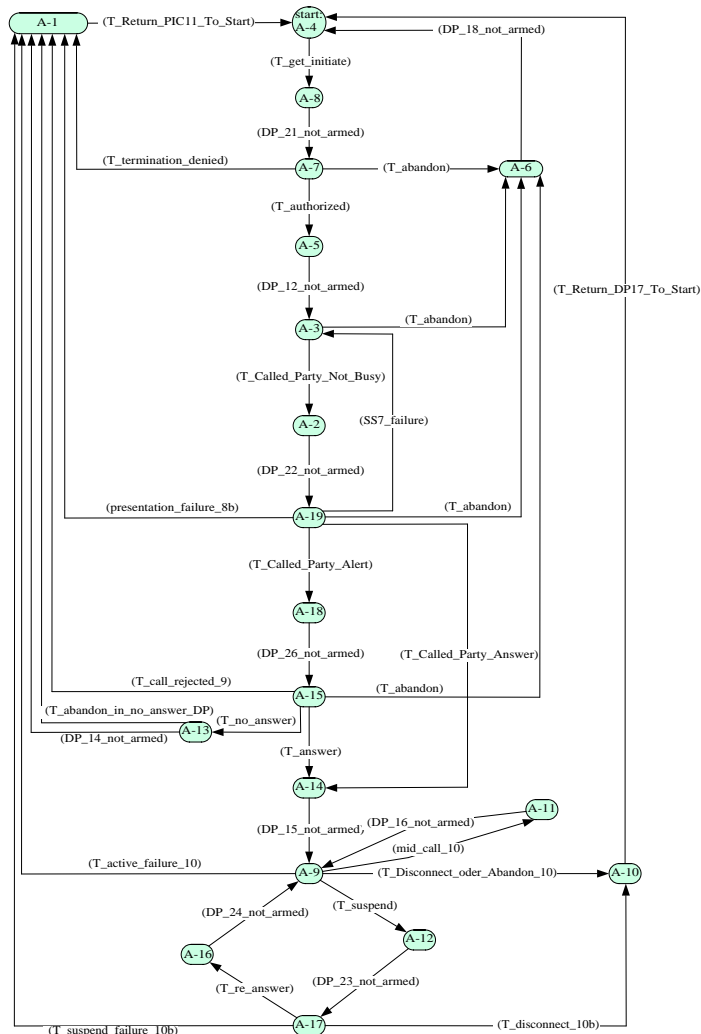


Abb. 30: Minimalautomat "T-BCSM"

Wie bereits beim O-BCSM stimmt der hier gefundene Minimalautomat bis auf das Fehlen des *Busy DPs* mit dem im ersten Modell überein (vgl. Abschnitt 3.2.2.4.).

3.3.4.5. Phasentrennung und Lebendigkeitseigenschaft

Der Untersuchung liegt die Erreichbarkeitsanalyse aus dem Abschnitt 3.3.4.1. zugrunde.

Der Homomorphismus ist genauso definiert wie im ersten Modell (vgl. Abschnitt 3.2.2.6.).

Die Berechnung des Minimalautomaten ergibt folgendes:

Minimalautomat bcsms2c-s1

6 Zustände berechnet.
 (9 Kanten, 4 Anschriften)

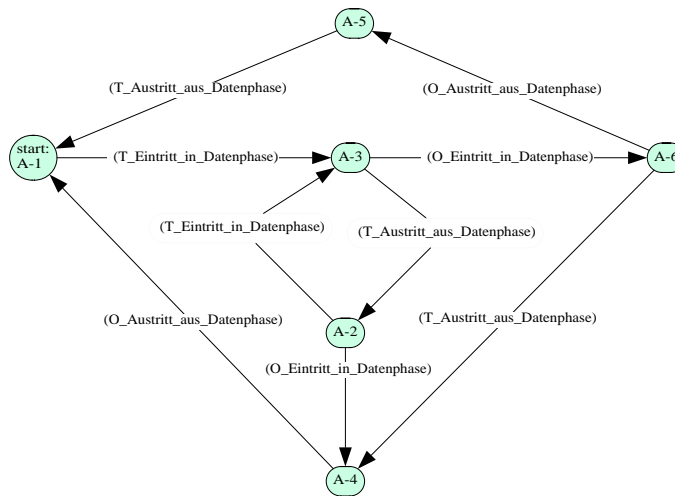


Abb. 31: Minimalautomat "Phasentrennung und Lebendigkeitseigenschaft"

Der Minimalautomat unterscheidet sich in einer zusätzlichen Kante von dem im ersten Modell. T kann nun beliebig oft in eine Datenphase ein- und wieder aus ihr austreten, ohne daß O in eine Datenphase eintreten muß. Dies ist ein wesentlicher Unterschied zum Verhalten des ersten Modells. Die starke Synchronisation der beiden Teilnehmer ist damit entfallen, so daß das Modell die Realität erheblich besser abbildet als ein Modell mit enger Koppelung.

Es ist auch hier mit der gleichen Argumentation wie im ersten Modell einzusehen, daß sowohl Phasentrennung als auch die Lebendigkeitseigenschaft gelten (vgl. Abschnitt 3.2.2.6.).

3.3.4.6. Besetzt-Verhalten

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

neue Anfangsmarkierung:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_ALT_BIT: 1<1,1,2,0>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 2<SYSTEM_READY>
T_ALT_BIT: 1<1,1,2,0>
T_BCSM_TO_O_BCSM: 1<::>
```

Erreichbarkeitsgraph bcsms2c-s1

```
29841 Markierungen berechnet.
(0 Totmarkierungen)
(0 Pseudoschaltsschritte)
(98936 Schaltsschritte)
```

```

Faktorgraph von Erreichbarkeitsgraph bcsms2c-s1
(1 Zusammenhangskomponente gefunden)
(1 Randkomponente)

```

Der Homomorphismus ist genauso definiert wie im ersten Modell (vgl. Abschnitt 3.2.2.8.).

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```

Minimalautomat bcsms2c-s1
1 Zustand berechnet.
(2 Kanten, 2 Anschriften)

Q-Markierungen : bcsms2c-s1

Q-::

Q-1
ENTRY_PIC_10_AFTER_ACCEPT: 1<1,1,2,1,::>

Ende Q-Markierungen : bcsms2c-s1

```

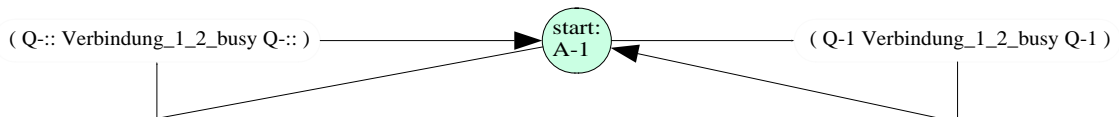


Abb. 32: Minimalautomat "Besetzt-Verhalten"

An den Q-Markierungen und dem Minimalautomaten ist zu erkennen, daß, im Gegensatz zum ersten Modell, das Schalten der Transition *busy_8a* und damit das Schicken eines *T_O_busy* auch auftreten kann, wenn für den Angerufenen noch eine alte Verbindung zu demselben Anrufer besteht (Marke auf der Stelle *Entry_PIC_10_after_accept*). Dieser hat die Verbindung bereits beendet und versucht, eine neue aufzubauen, während der Angerufene auf das Beenden durch den Anrufer nicht reagieren muß.

Hierin drückt sich, wie schon bei der Phasentrennung und Lebendigkeit, der größere End-zu-End-Schlupf zwischen den Teilnehmern aus.

3.3.4.7. Gesamtverhalten

Um abschätzen zu können, wie nahe dieses Modell bezüglich der Modellierung des vorgegebenen End-zu-End-Schlupfs dem Optimum kommt, wird ein Homomorphismus definiert, in dem alle Mechanismen, mit denen der End-zu-End-Schlupf modelliert wird, ausgeblendet werden. Dies sind die Schnittstellen und alle anderen Stellen, in denen das Alternating Bit zusammen mit den restlichen Daten zu einer Verbindung abgelegt wird. Damit erhält man einen Automaten, der den vorgegebenen Schlupf beinhaltet und diesbezüglich optimal ist.

Der Untersuchung liegt dieselbe Erreichbarkeitsanalyse wie der Untersuchung des Verhaltens des Anrufers (vgl. Abschnitt 3.3.4.1.) zugrunde.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsms2c
```

```
843 States computed.  
(2443 Edges, 2000 Labels)
```

Man erkennt an der Anzahl der Markierungen (4903) im Erreichbarkeitsgraphen und an der Anzahl der Zustände des Minimalautomaten, daß die Modellierung einen Zustandsraum erzeugt, der um ein mehrfaches größer ist als der Zustandsraum eines Modells, das denselben Schlupf beinhaltet und dabei optimal ist. Die Mechanismen zur Modellierung des Schlupfs erzeugen also 4060 Markierungen mehr. Bezogen auf das Optimum von 843 Zuständen sind das 481,6% mehr Markierungen.

Inwieweit sich das zweite Modell durch Modifikationen diesem optimalen Modell annähern läßt, wird die Anwendung eines äquivalenten Homomorphismus auf den entsprechenden Erreichbarkeitsgraphen im dritten Modell zeigen.

3.3.5. Diskussion der zweiten Spezifikation

Aufgrund der Möglichkeit für Anrufer und Angerufenen, weitgehend unabhängig vom jeweils anderen zu agieren und damit auch Nachrichten zu schicken, entsteht eine Vielzahl verschiedener Belegungen der beiden Schnittstellen, die in ihrer Kombination den Zustandsraum im Vergleich zur ersten, strikten Spezifikation erheblich vergrößern.

Der wesentliche Grund für die hohe Zahl der Markierungen (29841) bei Vergabe von jeweils zwei Marken *'system_ready'* ist der, daß eingegangene Nachrichten zu einer alten, bereits beendeten Verbindung erst gelöscht werden, wenn die erste Nachricht zu einer neuen Verbindung zwischen den gleichen Teilnehmern entgegengenommen wird (vgl. Alternating Bit-Mechanismus). Für einen Zustand, der zwischen dem Beenden einer Verbindung und dem Löschen der Nachrichten zu dieser Verbindung erreicht werden kann, wird im Erreichbarkeitsgraphen daher pro möglicher Schnittstellenbelegung mit alten Nachrichten eine eigene Markierung berechnet. Das bedeutet, daß dieser Teil des Erreichbarkeitsgraphen unnötigerweise um den Faktor "Anzahl unterschiedlicher Schnittstellenbelegungen mit alten Nachrichten" vergrößert wird.

Es ist daher in annehmbarer Zeit nicht möglich, die Erreichbarkeitsgraphen für die Anfangsmarkierungen berechnen zu lassen, aus denen per Homomorphismus die lokalen Verhaltensweisen von O- und T-BCSM mit dem Schicken eines *T_O_busy* erhalten werden können.

Ebenfalls ist es aufgrund eines zu großen Zustandsraums nicht möglich, das Verhalten bei aktiviertem Call Hold-Dienst in der Konfiguration wie bei der ersten Spezifikation (vgl. Abschnitt 3.2.2.7.) zu berechnen.

Diese Berechnungen wurden jeweils nach mehreren Stunden bzw. Tagen abgebrochen.

Die Komplexität dieses Modells macht deshalb die Untersuchung von CS-2-Diensten und deren gegenseitigen Wechselwirkungen unmöglich. Der Sinn dieses Modells liegt daher darin, die Unterschiede zum ersten Modell bezüglich der Dynamik bei dem Verzicht auf strenge Synchronisation aufzuzeigen.

Sowohl mit der Datenphasentrennung als auch mit dem Besetzt-Verhalten konnten bereits deutliche Unterschiede zur ersten Spezifikation aufgezeigt werden. Man kann mit einer einfachen Überlegung am Call Hold-Dienst zeigen, daß der Grad der Synchronisation der Teilnehmer auch Einfluß auf das Verhalten der Dienste hat, das bei der Service Interaction Detection untersucht wird.

In Abschnitt 3.2.2.7. war zu erkennen, daß im T-BCSM immer nur die Verbindung, die durch das Tripel ($user=1, line_u=1, number=2$) gekennzeichnet ist, in die aktive Phase eintreten kann. Setzt also der Anrufer, für den Call Hold aktiviert ist, die Verbindung auf Warteposition und versucht einen Verbindungsaufbau mit ($user=1, line_u=2, number=2$), so kann die Antwort, die er vom T-BCSM erhält, kein "akzeptiert" sein. Wie an der Q-Markierung zu erkennen ist, gelangt nie ein Token ($1,2,2,_,_$) im T-BCSM in die aktive Phase.

Dies ist jedoch im zweiten Modell möglich. Während der Anrufer mit ($1,2,2$) eine neue Verbindung aufzubauen versucht, beendet der Angerufene die Verbindung ($1,1,2$), die der Anrufer auf Warteposition gesetzt hatte. Der Angerufene gelangt in den Grundzustand, ist für den Aufbauwunsch ($1,2,2$) nicht besetzt und kann diesen Wunsch mit einem "akzeptiert" beantworten. Daraufhin können beide in die aktive Phase eintreten.

Das unterschiedliche Verhalten des Call Hold-Dienstes in der ersten und der zweiten Spezifikation kommt daher, daß ein Teilnehmer, hier der Angerufene, im nicht synchronisierten Modell eine Verbindung beenden kann, ohne daß der Anrufer davon Kenntnis nimmt und darauf reagiert. Genau dies ist jedoch im ersten Modell aufgrund der eingebauten End-zu-End-Handshakes erforderlich. Der Angerufene kann zwar ein *T_O_disconnect* schicken, aber da der Anrufer gerade versucht, eine neue Verbindung aufzubauen, kann er zu diesem Disconnect nicht die erforderliche Quittung zurückschicken, ohne die der Angerufene nicht in den Grundzustand gelangen kann.

Somit ist erwiesen, daß die Synchronisation der Teilnehmer Auswirkungen auf das Verhalten von Diensten und damit auch auf die Wechselwirkungen zwischen Diensten haben kann.

Aufgrund der Tatsache, daß das zweite, weitgehend unsynchronisierte Modell erheblich realitätsnäher als das erste ist, muß ein Modell, mit dem Dienste und ihre Wechselwirkungen vollständig analysiert werden können, das unabhängige Agieren der Teilnehmer gewährleisten. Außerdem müssen alte Nachrichten sofort von den Schnittstellen entfernt werden, da damit alle

Kombinationsmöglichkeiten von alten und neuen Nachrichten auf den Schnittstellen entfallen.
Der Zustandsraum müßte sich dadurch enorm verkleinern.

3.4. Dritte Spezifikation

Bei der Beschreibung des Modells wird nur auf die Modifikationen gegenüber der zweiten Spezifikation eingegangen. Der Vorspann und die Produktnetze zu dieser Spezifikation befinden sich im Anhang C.

3.4.1. Ziel dieser Spezifikation

Ziel dieser Spezifikation ist es, so früh wie möglich unnötige Nachrichten von den Schnittstellen zu entfernen bzw. nicht mehr dorthin zu legen, damit der Zustandsraum kleiner als in der zweiten Spezifikation wird.

3.4.2. Vorgehensweise

Um dieses Ziel zu erreichen, wird in dieser Spezifikation ein Transportsystem simuliert, das einem Teilnehmer Nachrichten zu einer Verbindung, die dieser bereits beendet hat, nicht mehr übermittelt. Weiterhin löscht jeder Teilnehmer beim Beenden einer Verbindung alle für ihn bestimmten Nachrichten zu dieser Verbindung. Somit ist für einen Teilnehmer sofort beim Beenden einer Verbindung auch die Schnittstelle von Nachrichten zu dieser Verbindung bereinigt. Das Transportsystem stellt sicher, daß zu dieser Verbindung keine Nachrichten mehr nachkommen. Durch dieses Verfahren wird daher auch der Alternating Bit-Mechanismus des zweiten Modells überflüssig.

Diese Lösung beinhaltet *weder* globale Flußkontrolle bzw. irgendeine Synchronisation, *noch* geht sie davon aus, daß gesendete Nachrichten sofort beim Empfänger anliegen.

3.4.3. Konkrete Modellierung

Um den Zustandsraum möglichst klein zu halten, wird das Transportsystem nicht explizit modelliert. Die Funktionalität des Transportsystems wird daher im BCSM minimal mitmodelliert.

Im O-BCSM schickt der Anrufer keine Nachrichten mehr zu einer Verbindung an das T-BCSM, falls der Angerufene diese Verbindung bereits beendet hat. Entsprechendes gilt für das Schicken von Nachrichten durch den Angerufenen.

Man benötigt also einen Indikator dafür, ob ein Teilnehmer eine Verbindung beendet hat. Dieser ist durch die Nachrichten auf den Schnittstellen gegeben. Das Auflegen des Anrufers erkennt man durch das Anliegen von *O_T_abandon* oder *O_T_disconnect* auf der Schnittstelle

O_BCSM_to_T_BCSM. Das Auflegen des Angerufenen erkennt man durch das Anliegen von *T_O_busy*, *T_O_no_answer*, zweimal *T_O_disconnect* bzw. einmal *T_O_disconnect*, falls sich der Anrufer bereits in *O_Suspended_PIC* befindet, auf der Schnittstelle *T_BCSM_to_O_BCSM*.

In der oben beschriebenen Simulation wird ausgenutzt, daß das O-BCSM bzw. das T-BCSM eingehende Nachrichten sofort im Zugriff haben. Das ist aber nur notwendig, um die Funktionalität des Transportsystems modellieren zu können und beinhaltet nicht die Annahme, daß Verzögerungen beim Nachrichtentransport nicht möglich sind.

3.4.4. Analyse und Verifikation

Zur Analyse und Verifikation dieses Modells werden, wie schon für die ersten beiden Spezifikationen, zu mehreren Anfangsmarkierungen die Erreichbarkeitsgraphen berechnet. Anschließend werden Homomorphismen definiert und von der Produktnetzmaschine jeweils der zugehörige Minimalautomat berechnet. Dadurch können folgende Verhaltensweisen aus der Dynamik des Modells extrahiert werden:

- lokale Verhaltensweisen
 1. Anrufer ohne abonnierten Dienst bzw. mit abonniertem Call Hold-Dienst
 2. Angerufener ohne abonnierten Dienst bzw. mit abonniertem Call Waiting-Dienst
 3. O-BCSM
 4. T-BCSM
- globale Verhaltensweisen
 5. Phasentrennung und Lebendigkeitseigenschaft
 6. Verhalten Call Hold-/Call Waiting-Dienst
- vergleichende Homomorphismen
 7. Verhalten bei aktiviertem Call Hold-Dienst
 8. Verhalten bei aktiviertem Call Waiting-Dienst
 9. Besetzt-Verhalten
 10. Verbindungsaufbau
 11. Verbindungsabbau
 12. Gesamtverhalten

3.4.4.1. Anrufer

3.4.4.1.1. Anrufer ohne abonnierten Dienst

Grundlage für die Untersuchung des Verhaltens eines Anrufers ohne abonnierte Dienste ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

Reachability Graph bcsms2c-s1

```
887 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(2570 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsms2c-s1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist wie in den ersten beiden Modellen definiert und auch das Ergebnis entspricht dem von oben (vgl. Abschnitt 3.2.2.1.1.).

3.4.4.1.2. Anrufer mit abonniertem Call Hold-Dienst

Im folgenden wird das Verhalten eines Anrufers mit abonniertem Call Hold-Dienst untersucht. Neben seinem normalen Verhalten sollte der Anrufer zwischen Nummer wählen und Auflegen die aktive Verbindung auf Warteposition setzen können. Danach muß er erneut eine Nummer wählen oder auflegen können, woraufhin er die erste Verbindung reaktivieren muß (Feature Call Retrieve). Hierauf sollte der Anrufer nur auflegen können, was gefolgt wird von einem Call Retrieve.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<1,2> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9.PIC_6.DP_10,DP_8,DP_1B> +
    1<1,CH,0.0,CRET,::,DP_10,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,PIC_6,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

Reachability Graph bcsms2c-s1

```
9211 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(33461 Occurrence Steps)
```

```

Connected Components of Reachability Graph bcsms2c-s1
(1 Component found)
(1 Leaf Component)

```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	ATTEMPT_1A
Nummer_waehlen	COLLECTED_INFO_2
auflegen	TD_9_C, TD_10_C, TD_10_D, EXCEPTION_6_A, EXCEPTION_6B
Call_Hold	O_CH_0
Call_Retrieve	O_CRET_0
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```

Minimal Automaton bcsms2c-s1

6 States computed.
(9 Edges, 5 Labels)

```

Das Ergebnis stimmt mit dem erwarteten überein und entspricht dem im ersten Modell gefundenen (vgl. Abschnitt 3.2.2.1.2.).

3.4.4.2. Angerufener

3.4.4.2.1. Angerufener ohne abonnierten Dienst

Der Untersuchung des Verhaltens eines Angerufenen ohne abonnierte Dienste liegt die Erreichbarkeitsanalyse aus Abschnitt 3.4.4.1.1. (Anrufer ohne abonnierte Dienste) zugrunde.

Der Homomorphismus ist definiert wie im zweiten Modell (vgl. Abschnitt 3.3.4.2.) und auch das Ergebnis stimmt mit dem aus den vorhergehenden Modellen überein (vgl. Abschnitt 3.2.2.2.).

3.4.4.2.2. Angerufener mit abonniertem Call Waiting-Dienst

Im folgenden wird das Verhalten eines Angerufenen, der den Call Waiting-Dienst abonniert hat, untersucht.

Neben dem Abheben und Auflegen sollte bei diesem Angerufenen nach dem Abheben ein Call Waiting möglich sein. Danach muß er beliebig oft zwischen seinen zwei Verbindungen hin- und herschalten können. Nach dem Auflegen muß der Angerufene mittels Call Retrieve einen noch wartenden Anruf aktivieren, bevor er endgültig auflegen kann oder ein erneutes Call Waiting möglich ist.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```

ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<1,2> + 1<2,1> + 1<2,2>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SERVICE_STRUCTURE: 1<2,CW,0.0,CRET,::,DP_17,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,DP_18,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,PIC_11,ENTRY_PIC_1> +
    1<2,CW,0.0,CTOG,::,DP_16,ENTRY_PIC_1> +
    1<2,CW,0.0,CW,::,DP_13,ENTRY_PIC_1>
STATIC_ARMING: 1<2,DP_13> + 1<2,DP_16> + 1<2,DP_17> +
    1<2,DP_18> + 1<2,PIC_11>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>

```

Reachability Graph bcsms2c-s1

```

18604 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(60833 Occurrence Steps)

```

```

Connected Components of Reachability Graph bcsms2c-s1
(1 Component found)
(1 Leaf Component)

```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	NOT_BUSY_8A
auflegen	ABANDON_8B, ABANDON_9, CALL_REJECTED_9, NO_ANSWER_9, SS7_FAILURE_8B, PRESENTATION_FAILURE_8B, T_ACTIVE_FAILURE_10A, T_SUSPEND_FAILURE_10B, TD_17_1B, TD_17_1A, DP_17_DYNAMIC_ARMED, DP_17_STATIC_ARMED
Call_Toggle	T_CTOG_0, T_CTOG
Call_Waiting	CW_0_1
Call_Retrieve	T_CRET_0, T_CRET
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-s1

```

5 States computed.
(8 Edges, 5 Labels)

```

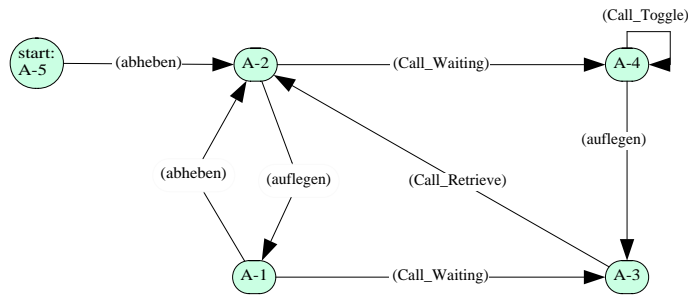


Abb. 33: Minimalautomat "Angerufener mit Call Waiting-Dienst"

Im Startzustand kann der Angerufene abheben und danach auflegen (Übergang von A-2 nach A-1) oder er wird erneut angerufen und es kommt zur Ausführung des Call Waiting-Dienstes, beginnend mit dem Feature Call Waiting (Übergang von A-2 nach A-4).

Erfolgt ein Anruf während der Prozedur des Auflegens, noch bevor der Angerufene in den Grundzustand zurückgekehrt ist, ist er für den Anrufer besetzt und es wird ebenfalls das Feature Call Waiting ausgeführt (Übergang von A-1 nach A-3).

Im Zustand A-4, der für den Angerufenen durch eine aktive Verbindung und eine Verbindung auf Warteposition charakterisiert ist, kann mit dem Feature Call Toggle zwischen den beiden Verbindungen hin- und hergeschaltet werden. Dabei wird der Zustand nicht gewechselt. Schließlich wird die gerade aktive Verbindung beendet (Übergang von A-4 nach A-3). Im Zustand A-3 hat der Angerufene aufgelegt, und es liegt noch eine Verbindung auf Warteposition. Der Angerufene ist nun gezwungen, diese Verbindung mit dem Feature Call Retrieve zu reaktivieren. So gelangt er zurück in den Zustand A-2.

Das Ergebnis zeigt, daß die Erwartungen an den Call Waiting-Dienst erfüllt werden.

3.4.4.3. O-BCSM

Zur Untersuchung des Verhaltens des O-BCSMs wird folgende Erreichbarkeitsanalyse gewählt:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<1> + 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

Reachability Graph bcsmc2c-s1

```
6622 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(25000 Occurrence Steps)
```



```
Connected Components of Reachability Graph bcsmc2c-s1
(1 Component found)
(1 Leaf Component)
```

Aufgrund der kleineren Komplexität dieses Modells gegenüber der des zweiten Modells ist hier die Analyse des O-BCSMs in der Konfiguration möglich, bei der ein "besetzt" vorkommen kann.

Die Definition des Homomorphismus und das Ergebnis der Berechnung des Minimalautomaten stimmen mit dem ersten Modell überein (vgl. Abschnitt 3.2.2.3.).

3.4.4.4. T-BCSM

Grundlage für die Untersuchung des Verhaltens des T-BCSMs ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:
```

```
ENTRY_PIC_1: 1<1> + 1<2>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

```
Reachability Graph bcsmc2c-s1
```

```
6211 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(22460 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsmc2c-s1
(1 Component found)
(1 Leaf Component)
```

Wie beim O-BCSM kann auch bei der Analyse des T-BCSMs die Konfiguration so gewählt werden, daß der Angerufene besetzt sein kann.

Die Definition des Homomorphismus und das Ergebnis der Berechnung des Minimalautomaten stimmen mit dem ersten Modell überein (vgl. Abschnitt 3.2.2.4.).

3.4.4.5. Phasentrennung und Lebendigkeitseigenschaft

Der Untersuchung liegt folgende Erreichbarkeitsanalyse zugrunde:

```
New Initial Marking:
```

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
```

```

NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>

```

```

Reachability Graph bcsmc2c-s1

```

```

3082 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(10927 Occurrence Steps)

```

```

Connected Components of Reachability Graph bcsmc2c-s1
(1 Component found)
(1 Leaf Component)

```

Die Definition des Homomorphismus erfolgt analog der Definition im zweiten Modell und auch das berechnete Ergebnis stimmt mit dem bereits oben besprochenen überein (vgl. Abschnitt 3.3.4.5.).

Derselbe Homomorphismus liefert dasselbe Ergebnis, wenn man ihn auf den Erreichbarkeitsgraphen aus Abschnitt 3.4.4.1.1. anwendet.

3.4.4.6. Verhalten bei abonniertem Call Hold-/Call Waiting-Dienst

Mit dieser Analyse soll das Zusammenspiel vom Call Hold- und Call Waiting-Dienst untersucht werden. Der Untersuchung liegt folgende Erreichbarkeitsanalyse zugrunde, die von einem Anrufer mit abonniertem Call Hold-Dienst und einem Angerufenen mit abonniertem Call Waiting-Dienst ausgeht:

```

New Initial Marking:

```

```

ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<1,2> + 1<2,1> + 1<2,2>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9.PIC_6.DP_10,DP_8,DP_1B> +
    1<1,CH,0.0,CRET,::,DP_10,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,PIC_6,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,DP_17,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,DP_18,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,PIC_11,ENTRY_PIC_1> +
    1<2,CW,0.0,CTOG,::,DP_16,ENTRY_PIC_1> +
    1<2,CW,0.0,CW,::,DP_13,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8> + 1<2,DP_13> + 1<2,DP_16> +
    1<2,DP_17> + 1<2,DP_18> + 1<2,PIC_11>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>

```

```

Reachability Graph bcsmc2c-s1

```

```

117079 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(377053 Occurrence Steps)

```

```

Connected Components of Reachability Graph bcsms2c-s1
(1 Component found)
(1 Leaf Component)

```

Um zu zeigen, welche Verbindung ohne Zutun eines Dienstes zustandekommt und welche vom Call Waiting-Dienst auf Warteposition gesetzte Verbindungen der Anrufer mittels Call Toggle und Call Retrieve aktivieren kann, wird folgender Homomorphismus definiert:

Block	Prädikat	Transitionen
T_Call_retrieve	(ON_HOLD=<1,1,2,1,ENTRY_PIC_5>+ <1,2,2,2,ENTRY_PIC_10_AFTER_ACCEPT>,, ON_HOLD=<1,1,2,1,ENTRY_PIC_5>);	T_CRET, T_CRET_0
T_accept	(~ON_HOLD,,~ON_HOLD & ENTRY_PIC_10_AFTER_ACCEPT=<1,1,2,1,::>);	ACCEPT_10
T_Call_toggle	(ON_HOLD=<1,1,2,1,ENTRY_PIC_5>+ <1,2,2,2,ENTRY_PIC_10_AFTER_ACCEPT>,, ON_HOLD=<1,1,2,1,ENTRY_PIC_5>+ <1,1,2,1,ENTRY_PIC_10_AFTER_ACCEPT>);	T_CTOG, T_CTOG_0
Epsilon		alle restlichen Transitionen

Es werden die Stellen *On_hold* und *Entry_PIC_10_after_accept* sowie alle drei Komponenten der Zustandsübergänge im Minimalautomaten eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```

Minimal Automaton bcsms2c-s1
1 State computed.
(3 Edges, 3 Labels)

Q-Markings : bcsms2c-s1

Q-::

Q-1
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_5>

Q-2
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_5> +
1<1,2,2,2,ENTRY_PIC_10_AFTER_ACCEPT>

Q-3
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_10_AFTER_ACCEPT> +
1<1,1,2,1,ENTRY_PIC_5>

End Q-Markings : bcsms2c-s1

```

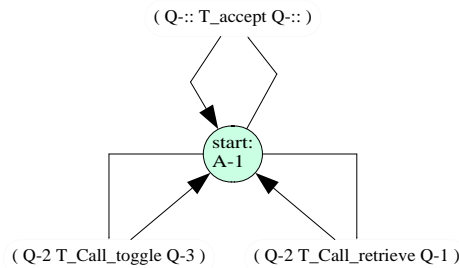


Abb. 34: Minimalautomat "Verhalten Call Hold-/Call Waiting-Service"

Der Minimalautomat in Abb. 34 zeigt, daß zwischen Anrufer und Angerufenem ohne Beteiligung eines Dienstes eine Verbindung immer wieder zustandekommen kann (Schleife ($Q-:: T_accept Q-::$)). Außerdem kann der Angerufene mittels Call Toggle eine durch den Call Waiting-Dienst auf Warteposition gesetzte neue Verbindung zum Anrufer aktivieren und die andere dabei auf Warteposition setzen (Schleife ($Q-2 T_Call_toggle Q-3$)). Dabei hat der Anrufer diese zweite Verbindung initiiert, indem er die erste mit dem Call Hold-Dienst auf Warteposition gesetzt hat (erkennbar an der Marke $\langle 1,1,2,1,ENTRY_PIC_5 \rangle$ auf der Stelle *On_hold* sowohl in der Q-Markierung $Q-2$ als auch in $Q-3$).

Durch Beenden der ersten Verbindung zum Anrufer kann der Angerufene das Feature Call Retrieve ausführen und den wartenden Anruf aktivieren (Schleife ($Q-2 T_Call_retrieve Q-1$)). Dies entspricht unter Berücksichtigung der Definition des Homomorphismus dem erwarteten Verhalten.

3.4.4.7. Verhalten bei aktiviertem Call Hold-Dienst

Im Gegensatz zum zweiten Modell, in dem aus Komplexitätsgründen diese Analyse praktisch nicht berechenbar ist, kann sie an diesem Modell durchgeführt werden.

Der Untersuchung liegt folgende Erreichbarkeitsanalyse zugrunde:

New Initial Marking:

```

ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 2<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<1,2> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9.PIC_6.DP_10,DP_8,DP_1B> +
    1<1,CH,0.0,CRET,::,DP_10,ENTRY_PIC_1>+
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1> +
    1<1,CH,0.0,CRET,::,PIC_6,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8>
SYSTEMS: 2<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>

```

Reachability Graph bcsmc2c-s1

```

9211 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(33461 Occurrence Steps)

```

```

Connected Components of Reachability Graph bcsms2c-s1
(1 Component found)
(1 Leaf Component)

```

Es wird derselbe Homomorphismus wie im ersten Modell benutzt (vgl. Abschnitt 3.2.2.7.). Auch hier werden die Stellen *Entry_PIC_10_after_accept* und *On_hold* sowie alle drei Komponenten im Bildbereich des Homomorphismus eingeblendet. Die Auswertung des Homomorphismus liefert folgendes Resultat:

```

Minimal Automaton bcsms2c-s1

2 States computed.
(4 Edges, 3 Labels)

Q-Markings : bcsms2c-s1

Q-::

Q-1
ENTRY_PIC_10_AFTER_ACCEPT: 1<1,2,2,1,::>

Q-2
ENTRY_PIC_10_AFTER_ACCEPT: 1<1,2,2,1,::>
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_5>

Q-3
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_5>

Q-4
ENTRY_PIC_10_AFTER_ACCEPT: 1<1,1,2,1,::>

End Q-Markings : bcsms2c-s1

```

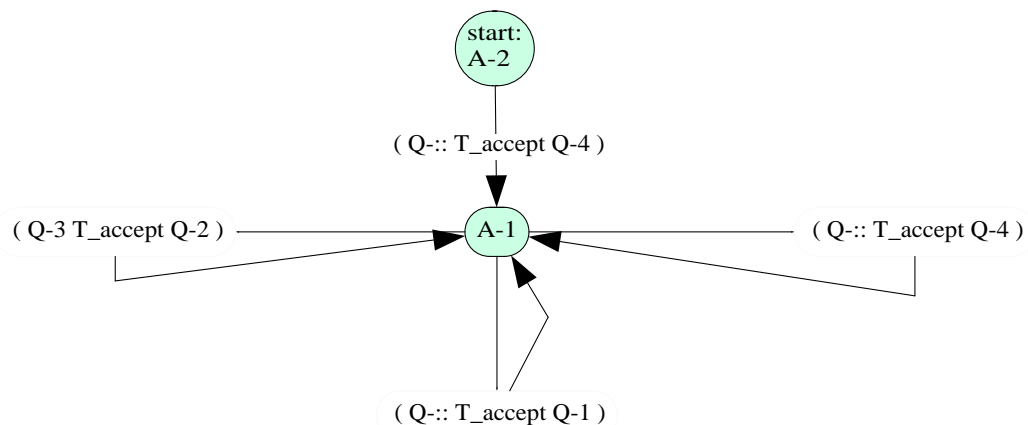


Abb. 35: Minimalautomat "Call Hold-Dienst"

Es ist zu erkennen, daß das Modell folgende Verhaltensweisen enthält:

1. Verbindung (1,1,2) wird vom Angerufenen akzeptiert (Kante $(Q-:: T_accept\ Q-4)$ von A-2 zu A-1)

2.
 - Verbindung (1,1,2) wird vom Angerufenen akzeptiert
 - Anrufer setzt sie auf Warteposition
 - Angerufener beendet die Verbindung (1,1,2); man erhält Q-Markierung $Q-3$
 - Anrufer initiiert Verbindung (1,2,2)
 - Angerufener akzeptiert die Verbindung (1,2,2) (Schleife ($Q-3 T_accept Q-2$) um $A-1$)
3.
 - Verbindung (1,1,2) wird vom Angerufenen akzeptiert
 - Verbindung (1,1,2) wird beendet; man erhält Q-Markierung $Q-::$ und ist in Zustand $A-1$
 - Verbindung (1,1,2) wird erneut aufgebaut (Schleife ($Q-:: T_accept Q-4$) um $A-1$)
4.
 - Verbindung (1,1,2) wird vom Angerufenen akzeptiert
 - Anrufer setzt sie auf Warteposition
 - Anrufer initiiert Verbindung (1,2,2)
 - Angerufener beendet Verbindung (1,1,2)
 - Verbindungsaufbauwunsch zu (1,2,2) wird von Angerufenem mit $T_O_alerted$ beantwortet
 - Anrufer beendet Verbindung (1,2,2) und kehrt zu der auf Warteposition gesetzten Verbindung (1,1,2) zurück
 - Angerufener akzeptiert Verbindung (1,2,2) (Schleife ($Q-:: T_accept Q-1$) um $A-1$)

Man sieht, daß die Handlungsmöglichkeiten in diesem Modell im Gegensatz zum ersten (vgl. Abschnitt 3.2.2.7.) erweitert sind und die Teilnehmer flexibler agieren können. Dies konnte für das zweite Modell aus Komplexitätsgründen zwar nicht gezeigt werden, doch wurde der End-zu-End-Schlupf im dritten Modell gegenüber dem zweiten nicht verändert, so daß keine zusätzlichen Freiräume hinzugekommen sind. Also sind die hier gefundenen Verhaltensweisen bereits im zweiten Modell enthalten.

3.4.4.8. Verhalten bei aktiviertem Call Waiting-Dienst

Hier wird untersucht, zwischen welchen Verbindungen ein Angerufener mit abonniertem Call Waiting-Dienst bei nur einem Anrufer hin- und herschalten kann. Es sollte ihm möglich sein, zwischen zwei Verbindungen zum Anrufer wechseln zu können

Der Untersuchung liegt dieselbe Erreichbarkeitsanalyse wie in Abschnitt 3.4.4.2.2. zugrunde.

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
T_toggle	T_CTOG, T_CTOG_0
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsmcs2c-s1
1 State computed.
(2 Edges, 2 Labels)

Q-Markings : bcsmcs2c-s1

Q-1
ON_HOLD: 1<1,1,2,2,ENTRY_PIC_10_AFTER_ACCEPT>

Q-2
ON_HOLD: 1<1,1,2,1,ENTRY_PIC_10_AFTER_ACCEPT>

End Q-Markings : bcsmcs2c-s1
```

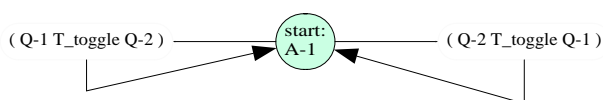


Abb. 36: Minimalautomat "Call Waiting-Dienst"

Der Minimalautomat zeigt, daß der Angerufene zwischen zwei durch die *Lines* 1 und 2 gekennzeichneten Verbindungen zum Anrufer hin- und herschalten kann. Der Anrufer muß eine der Verbindungen bereits beendet haben, um ein zweites Mal anrufen zu können. Das gefundene Resultat entspricht dem erwarteten Verhalten.

3.4.4.9. Besetzt-Verhalten

Der Untersuchung des Besetzt-Verhaltens liegt die Erreichbarkeitsanalyse aus Abschnitt 3.4.4.5. zugrunde. Der Homomorphismus wird wie im zweiten Modell definiert und auch das Ergebnis stimmt mit dem dort gefundenen überein (vgl. Abschnitt 3.3.4.6.). Dies ist korrekt, denn am End-zu-End-Schlupf wurde gegenüber dem zweiten Modell nichts geändert.

3.4.4.10. Verbindungsaufbau

Der Verbindungsaufbau, aus Sicht des O-BCSMs, wird betrachtet, um ihn mit dem Verhalten beim Verbindungsaufbau in einem abstrakteren Modell vergleichen zu können.

Um ein übersichtliches und doch aussagekräftiges Ergebnis zu erhalten, werden einige Netze so manipuliert, daß die Analyse nach jedem auftretenden Fehler oder vorzeitigem Auflegen, spätestens nach dem Erreichen der aktiven Phase in einer Totmarkierung endet. Dadurch beschränkt sich der Erreichbarkeitsgraph auf den hier zu untersuchenden Verbindungsaufbau.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<1> + 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

Reachability Graph bcsmc2c-s1

```
447 Markings computed.
(2 Dead Markings)
(0 Pseudo Occurrence Steps)
(1176 Occurrence Steps)
```

2 Dead Markings bcsmc2c-s1

```
M-337 TD_17_1A M-243 DP_7_NOT_ARMED M-242 TP_11_A
M-288
+++ dead +++
```

```
BUSY: 1<1,1>
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<2>
LINES: 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_BUSY>
T_BCSM_TO_O_BCSM:
    1<(1,1,T_O_DISCONNECT,2,1,0).(1,1,T_O_DISCONNECT,2,1,0)>
```

```
M-298 TD_17_1A M-215 TD_18_A M-214 TP_11_A M-56 DP_18_NOT_ARMED_2 M-
55 EXCEPTION_11_B M-11 TP_6_A M-10
TD_10_A
M-13
+++ dead +++
```

```
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

End 2 Dead Markings bcsmc2c-s1

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
ok	ATTEMPT_1A
DP_1_not_armed	DP_1_NOT_ARMED
collected_info	COLLECTED_INFO_2
DP_2_not_armed	DP_2_NOT_ARMED
analysed_info	ANALYSED_INFO_3
DP_3_not_armed	DP_3_NOT_ARMED
Route_select_Failure	ROUTE_FAILURE_4A
interpret	ROUTE_SELECTED_4A
authorized_for_number	CALL_SETUP_4B
answer	ANSWER_4C, ANSWER_4D
busy	BUSY_4D, BUSY_4C
no_answer	NO_ANSWER_4D, NO_ANSWER_4C
alert	ALERTED_4C
DP_7_not_armed	DP_7_NOT_ARMED
DP_8_not_armed	DP_8_NOT_ARMED_1, DP_8_NOT_ARMED_2
Exception_DP4	DP_4_NOT_ARMED
Exception_DP5	DP_5_NOT_ARMED
Exception_DP6	DP_6_NOT_ARMED
authorized	AUTHORIZED_1B
DP_1b_not_armed	DP_1B_NOT_ARMED
DP_25_not_armed	DP_25_NOT_ARMED
route_busy	ROUTE_BUSY_4A
route_failure_4c	ROUTE_FAILURE_4C
Mid_Call_4c	MID_CALL_4C
Mid_Call_4d	MID_CALL_4D
route_failure_4d	ROUTE_FAILURE_4D
DP9/10/6_not_armed	PIC_6_NOT_ARMED, DP_10_NOT_ARMED, TD_9_A, TD_9_B
O_Disconnect	AUTHORIZE_ROUTE_FAILURE_4C, ABANDON_4C_1, ORIGINATION_DENIED_1B, ABANDON_4D, ABANDON_4C_3, AUTHORIZE_ROUTE_FAILURE_4B, ABANDON_3, COLLECT_TIMEOUT_2, INVALID_INFORMATION_3, ABANDON_4C_2, ABANDON_1B, ABANDON_2, ABANDON_4A, ABANNDON_4B,
Epsilon	alle restlichen Transitionen

Der Homomorphismus beinhaltet alle bis zum Erreichen der aktiven Phase möglichen Zustandsübergänge.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-s1

20 States computed.
 (65 critical Markings)
 (34 Edges, 26 Labels)

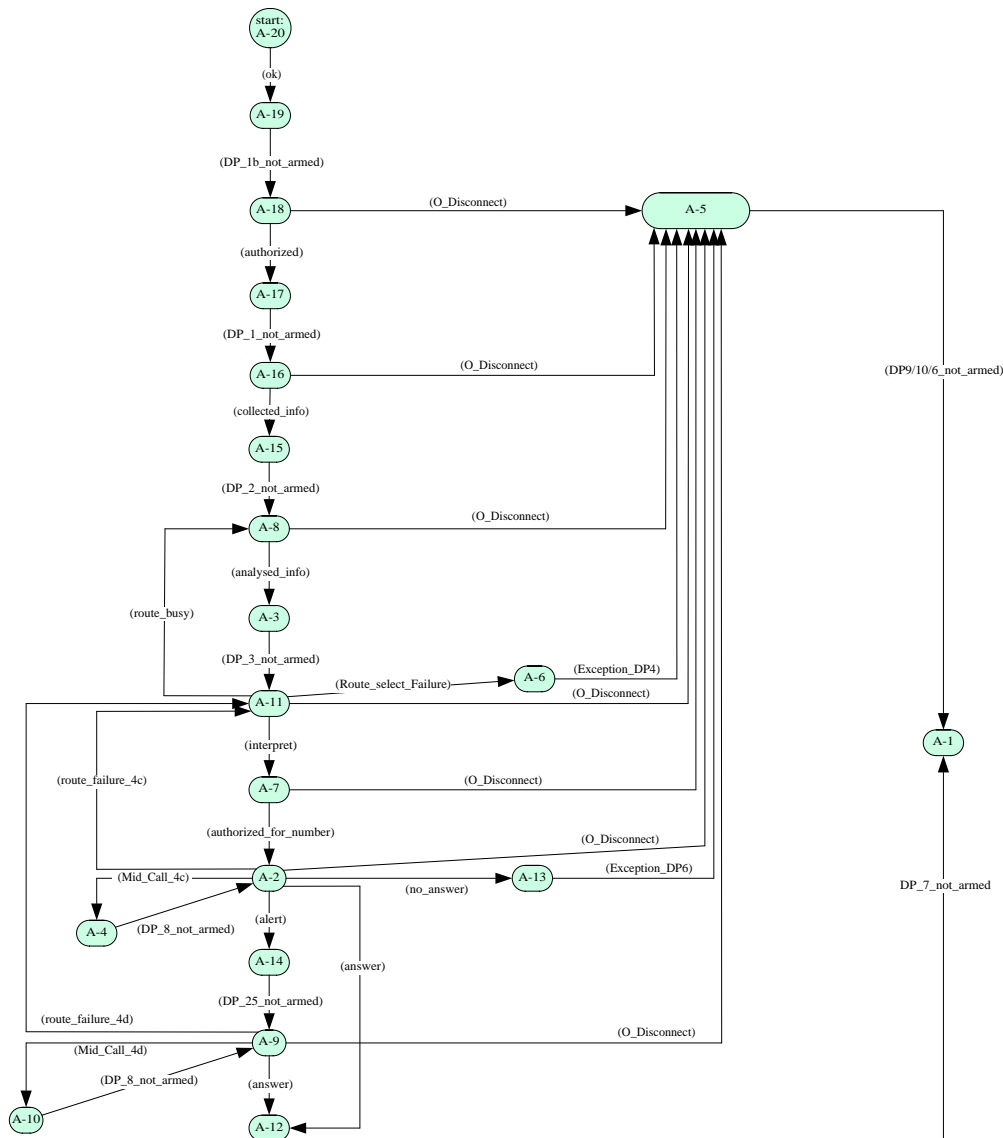


Abb. 37: Minimalautomat "Verbindungsaufbau"

Bemerkung: Dieses Ergebnis wird bei der Analyse eines globalen Modells zu Vergleichszwecken herangezogen.

3.4.4.11. Verbindungsabbau

Die Untersuchung des Verbindungsabbaus wird, wie auch die des Verbindungsaufbaus, bei der Analyse eines Modells unter globaler Sichtweise zu Vergleichen herangezogen und nur deshalb am dritten Modell durchgeführt.

Um ein übersichtliches und doch aussagekräftiges Ergebnis zu erhalten, werden einige Netze so verändert, daß nach dem Ende des Verbindungsabbaus keine neue Verbindung mehr aufgebaut werden kann, sondern die Analyse in einer Totmarkierung endet. Die Anfangsmarkierung wird so gewählt, daß die Analyse im aktiven Zustand der Verbindung, also unmittelbar vor dem Beginn des Verbindungsabbaus, startet.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:
```

```
BUSY: 1<1,1> + 1<1,2>
ENTRY_PIC_1: 1<1>
ENTRY_PIC_10_AFTER_ACCEPT: 1<1,1,2,1,::>
ENTRY_PIC_5: 1<1,1,2,1,::>
ENTRY_PIC_7: 1<SYSTEM_BUSY>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_BUSY>
T_BCSM_TO_O_BCSM: 1<::>
```

```
Reachability Graph bcsms2c-s1
```

```
179 Markings computed.
(1 Dead Marking)
(0 Pseudo Occurrence Steps)
(467 Occurrence Steps)
```

```
1 Dead Marking bcsms2c-s1
```

```
M-157 TD_17_1A M-156 TD_9_B M-155 TP_11_A M-154 TP_6_B
M-168
+++ dead +++
```

```
BUSY: 1<1,2>
ENTRY_PIC_1: 1<1>
ENTRY_PIC_7: 1<SYSTEM_READY>
IDLE: 1<2>
LINES: 1<1,1> + 1<2,1>
NUMBER_POOL: 1<2>
O_BCSM_TO_T_BCSM: 1<::>
SYSTEMS: 1<SYSTEM_READY>
T_BCSM_TO_O_BCSM: 1<::>
```

```
End 1 Dead Marking bcsms2c-s1
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
O_disconnect	OTDISCONNECT_5B, O_SUSPEND_FAILURE_5B, TODISCONNECT_5B, O_ACTIVE_FAILURE_5, OTDISCONNECT_5
T_disconnect	OTDISCONNECT_10, OTDISCONNECT_10B, TIMEOUT_10B, T_SUSPEND_FAILURE_10B, T_ACTIVE_FAILURE_10A
DP_9_10_6_not_armed	PIC_6_NOT_ARMED, DP_10_NOT_ARMED, TD_9_A, TD_9_B
DP_17_11_not_armed	PIC_11_NOT_ARMED, TD_17_1A, TD_17_1B
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-s1

9 States computed.
 (4 critical Markings)
 (12 Edges, 4 Labels)

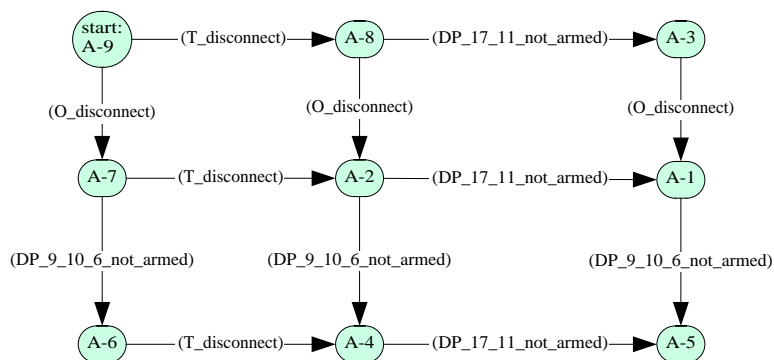


Abb. 38: Minimalautomat "Verbindungsabbau"

3.4.4.12. Gesamtverhalten

Um abschätzen zu können, wie nahe dieses Modell bezüglich der Modellierung des vorgegebenen End-zu-End-Schlupfs dem Optimum kommt, wird ein Homomorphismus definiert, in dem alle Mechanismen, mit denen der End-zu-End-Schlupf modelliert wird, ausgeblendet werden. Damit erhält man einen Automaten, der den vorgegebenen Schlupf beinhaltet und diesbezüglich optimal ist.

Dieser Untersuchung liegt dieselbe Erreichbarkeitsanalyse zugrunde wie der Untersuchung des Verhaltens eines Anrufers ohne abonnierte Dienste (vgl. Abschnitt 3.4.4.1.1.). Bei der Definition des Homomorphismus wird für jede Transition ein eigener Block geschaffen und keine ausgeblendet. Außerdem werden lediglich die (Schnitt-)Stellen $O_BCSM_to_T_BCSM$ und $T_BCSM_to_O_BCSM$ nicht berücksichtigt, da in diesem Modell zur Modellierung des

End-zu-End-Schlupfs außer den Schnittstellen und der Art des Zugriffs darauf keine weiteren Mechanismen benötigt werden.

Die Berechnung des Minimalautomaten ergibt folgendes Resultat:

```
Minimal Automaton bcsmcs2c-s1
```

```
843 States computed.  
(2471 Edges, 116 Labels)
```

Das Ergebnis von 843 Zuständen gegenüber 887 Markierungen im Erreichbarkeitsgraphen zeigt, daß durch die unterschiedlichen Schnittstellenbelegungen lediglich 44 Markierungen erzeugt werden. Bezogen auf das Optimum von 843 sind das lediglich 5,2% mehr Markierungen (vgl. 481,6% beim zweiten Modell, Abschnitt 3.3.4.7.).

Somit liegt dieses Modell nahe am Optimum. Die Erreichbarkeitsanalyse eines Modells unter verteilter Sichtweise und mit diesem End-zu-End-Schlupf wird daher immer obige Größenordnung haben. Hieraus folgt, daß man zur Untersuchung von Service Interaction mit realistischen Testfällen zu einer Modellierung unter einer abstrakteren Sichtweise übergehen muß.

4. Modellierung unter globaler Sichtweise

4.1. Entwicklung eines globalen Modells

Um ein Modell mit realistischen Testkonfigurationen auf Service Interaction hin untersuchen zu können, muß man - wie in Abschnitt 3.4.4.12. bereits erläutert - von der verteilten Sichtweise zu einer abstrakteren, globalen Sichtweise übergehen.

In der globalen Sichtweise muß man sich auf die wesentlichen Handlungsmöglichkeiten der Teilnehmer beschränken, da sonst wieder die Komplexität des verteilten Modells erreicht würde.

In dem hier vorgestellten Modell wird der Zustand eines Anrufs bis zur aktiven Phase an die Handlungen bzw. den Zustand des Anrufers gekoppelt. Die Aktivitäten des Angerufenen werden bis zu diesem Zeitpunkt durch die Reaktionen des Anrufers sichtbar (Beispiel: Falls der Angerufene bereits telefoniert, kommt die Marke des Anrufs in einen globalen *Busy DP*). Das globale Modell stimmt daher bis zur aktiven Phase weitgehend mit dem O-BCSM der verteilten Modellierung überein.

In dem Teil des Modells, der den Verbindungsabbau betrifft, wird jedoch zwischen den Aktivitäten von Anrufer und Angerufenem unterschieden, damit die Unabhängigkeit der Teilnehmer gewährleistet ist. Wie man an den Vergleichen zwischen dem ersten und dem zweiten bzw. dritten Modell gesehen hat, ist dies notwendig, um den Teilnehmern alle realistischen Verhaltenweisen zu ermöglichen.

Derjenige Teilnehmer, der zuerst beendet, kehrt in den Grundzustand zurück, bevor der andere Teilnehmer seinerseits beenden kann.

Bei dieser Vorgehensweise beim Verbindungsabbau kann ein Teilnehmer, sobald er eine Verbindung beendet hat - unabhängig davon, in welchem Zustand sich sein Gesprächspartner befindet -, einen neuen Anruf initiieren bzw. ein eingehendes Gespräch annehmen.

Einerseits ist so die Unabhängigkeit der Teilnehmer beim Verbindungsabbau gegeben, andererseits vergrößert dieses Vorgehen nicht unnötig den Zustandsraum durch eine Verzahnung der Aktionsfolgen beider Teilnehmer beim Beenden.

Bei dieser Vorgehensweise erhält man folgenden Zustandsgraphen eines Basisanrufs:

Angerufenen (*t_bit*) eingeführt. Ein gesetztes Bit bedeutet, daß der entsprechende Teilnehmer die Verbindung beendet hat. In diesem Fall wird er für weitere Aktionen bezüglich dieser Verbindung gesperrt.

Ein weiteres Bit für Anrufer (*o_oh*) und Angerufenen (*t_oh*) wird benötigt, um zu kennzeichnen, ob ein Teilnehmer eine Verbindung auf Warteposition gesetzt hat. Solange das entsprechende Bit auf 1 gesetzt ist, wird der Teilnehmer für Handlungen bezüglich dieser Verbindung gesperrt.

Um gewährleisten zu können, daß nach dem Beenden einer Verbindung durch einen Teilnehmer diese für den anderen noch besteht und dieser somit beispielsweise einen Dienst zur Ausführung bringen oder selbst beenden kann, wird ein neuer PIC (*PIC-12*) eingeführt. Dieser repräsentiert den Zustand nach einseitigem Beenden einer Verbindung durch einen der Teilnehmer.

4.2.1. Änderungen der DPs

In einem globalen Modell muß an bestimmten Punkten sowohl für den Anrufer als auch für den Angerufenen der Sprung zur SCF ermöglicht werden. Das ist bei folgenden DPs der Fall, da diese im verteilten Modell im O-BCSM und im T-BCSM auftreten: *Busy DP*, *No_Answer DP*, *Mid_Call DP*, *Reanswer DP* und *Suspended DP*.

Um beiden Teilnehmern die Möglichkeit zum Sprung zur SCF zu geben, müssen diese DPs im globalen Modell anders modelliert werden als im verteilten Modell. Die Änderungen werden exemplarisch am *Mid_Call DP* (Abb. 40) erläutert.

Änderungen bezüglich der Armierungsmechanismen

- **Dynamische Armierung**
Damit im Fall der dynamischen Armierung zwischen der Armierung für den Anrufer und der für den Angerufenen unterschieden werden kann, enthält das Verbindungstoken im BCSM in den letzten beiden Komponenten die Armierungsliste des Anrufers (*u_ar*), bzw. die des Angerufenen (*n_ar*). Die Transition *DP_8_dynamic_armed_1* schaltet, falls der *DP-8* in der Armierungsliste des Anrufers auftritt und dieser die Verbindung, die durch das Verbindungstoken repräsentiert wird, weder beendet noch auf Warteposition gesetzt hat, was an den Bits *o_oh* und *o_bit* im Verbindungstoken erkannt wird. Die Transition *DP_8_dynamic_armed_2* schaltet, falls die entsprechenden Bedingungen für den Angerufenen erfüllt sind.
- **Statische Armierung**
Die Transition *DP_8_static_armed* schaltet, wenn der *DP-8* für den Anrufer oder den

Angerufenen statisch armiert ist, und derjenige, für den dies der Fall ist, die Verbindung, die durch das Verbindungstoken repräsentiert wird, weder beendet noch auf Warteposition gesetzt hat.

Sowohl für die dynamische wie auch für die statische Armierung wird zusätzlich überprüft, ob die Verbindung aus *PIC-5* (aktive Phase) oder *PIC-12* (ein Teilnehmer hat die Verbindung beendet) zum *DP-8* kommt, denn nur dann soll ein Sprung zur Dienstlogik möglich sein.

Damit in der Dienstlogik erkannt werden kann, von wem - Anrufer oder Angerufenem - ein Dienst aufgerufen wird, enthält das Token, das zur Dienstauführung auf die Stelle *BCSM_to_SCF* gelegt wird, in der ersten Komponente den Dienstnutzer. In der verteilten Modellierung ist dies nicht notwendig, da dort an der Komponente *DP_in* erkannt werden kann, ob der Dienstaufwurf aus dem O-BCSM oder dem T-BCSM kam.

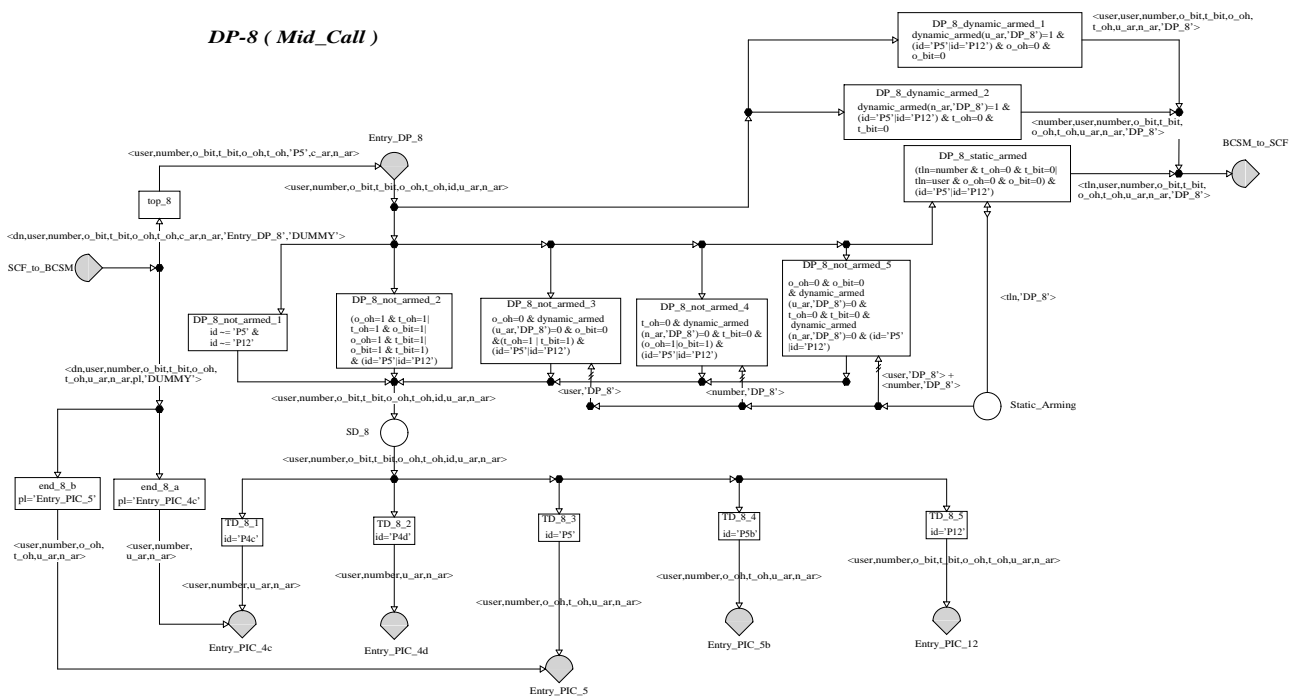


Abb. 40: *Mid_Call DP*

4.3. Anbindung der Dienstlogik

Die Modellierung der Dienstlogik bzw. deren Anbindung an das BCSM entspricht im wesentlichen der des verteilten Modells. Wichtige Unterschiede werden im folgenden an aussagekräftigen Beispielen erläutert.

4.3.1. Änderungen bezüglich Service Control Function Input

Die Modellierung von Service Control Function Input (Abb. 41) bleibt im Vergleich zu der des verteilten Modells im wesentlichen unverändert. Die einzigen Unterschiede bestehen darin, daß

- das Token, das auf die Stelle *Feature_queue* gelegt wird, als erste Komponente den Dienstnutzer enthält und
- die Transitionen *cli_ServiceFeatures* bzw. *number_ServiceFeatures* erst dann einen neuen Dienstwunsch abnehmen, wenn kein anderer Dienst mehr in Bearbeitung ist. Diese Maßnahme wird eingeführt, um den Erreichbarkeitsgraphen nicht unnötig zu vergrößern.

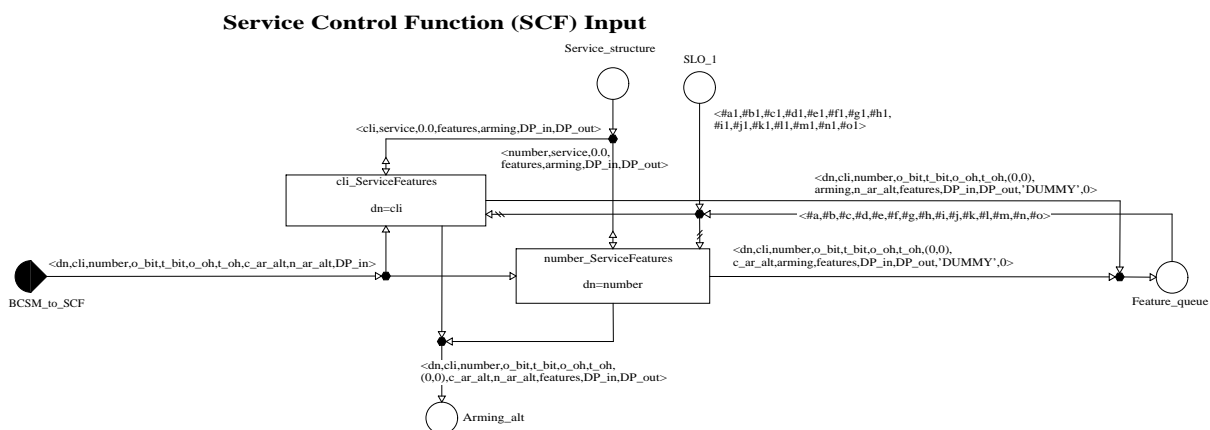


Abb. 41: Service Control Function Input

4.3.2. Änderungen bezüglich Service Control Function Output

In der Dienstmodellierung für das globale Modell ist es oft nötig, daß mit der Ausführung eines Dienstes zwei Token in das BCSM zurückgelegt werden. Durch folgende Modellierung geschieht dies immer in einer festgelegten Reihenfolge, so daß der Zustandsraum des Erreichbarkeitsgraphen nicht unnötig vergrößert wird.

Das Netz Service Control Function Output (Abb. 42) wird gegenüber dem verteilten Modell dahingehend geändert, daß, falls für einen Dienstnutzer zwei Token auf der *Feature_queue* liegen, zuerst das Token, das als letzte Komponente eine 0 enthält, an das BCSM übergeben wird und erst danach das Token, das als letzte Komponente eine 1 enthält. Dazu wird mit dem Schalten von *Result* irgendein Token von der *Feature_queue* genommen. Liegt für den gleichen Dienstnutzer noch ein Token mit letzter Komponente 0 auf der *Feature_queue*, so wird dieses mit der Transition *Result_b* auf die Stelle *SCF_to_BCSM* gelegt, und das Token, das zuerst genommen wurde, wird auf die *Feature_queue* zurückgelegt. Ist dies nicht der Fall, so wird das erste Token mit der Transition *Result_a* auf die Stelle *SCF_to_BCSM* gelegt.

Weiterhin können alle Result-Transitionen nur schalten, falls kein Token mehr auf der Stelle *SCF_to_BCSM* liegt.

Service Control Function (SCF) Output

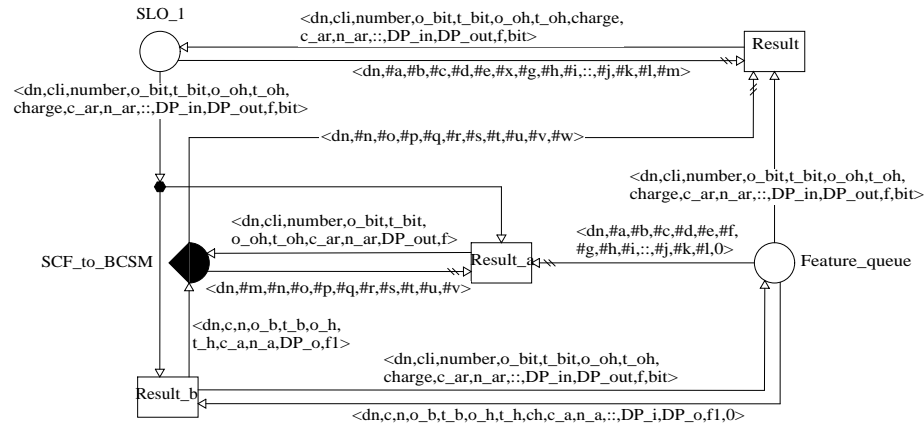


Abb. 42: Service Control Function Output

4.3.3. Änderungen bezüglich der Modellierung der Service Features

Die notwendigen Änderungen in der Modellierung der Service Features werden exemplarisch an dem Service Feature Call Hold (Abb. 43) erläutert. Die Netze der restlichen Service Features befinden sich im Anhang D.

Bei der Modellierung der Service Features für das globale Modell gibt es nicht mehr zwei getrennte Netze für O-BCSM und T-BCSM. Die Funktionalitäten dieser beiden Netze des verteilten Modells sind im globalen Modell in einem Netz enthalten. Es wird je nach Dienstnutzer (erste Komponente des Tokens auf der *Feature_queue*) entschieden, ob das Feature für den Anrufer oder den Angerufenen ausgeführt werden soll.

Wie im verteilten Modell wird auch hier bei der Ausführung des Service Features Call Hold zunächst überprüft, ob der Dienstnutzer bereits eine Verbindung auf Warteposition gesetzt hat. Ist dies der Fall, so kann das Servicefeature Call Hold nicht mehr ausgeführt werden, und es wird mit den alten Armierungslisten zu dem Punkt zurückgesprungen, aus dem die Dienstlogik aufgerufen wurde (Transition: *No_CH_0*).

Hat der Dienstnutzer keine Verbindung auf Warteposition, wird die aktuelle Verbindung auf Warteposition gesetzt, d.h. auf der Stelle *On_hold* gespeichert. Der Rücksprung zum BCSM erfolgt für den Dienstnutzer auch weiterhin zu *Entry_DP_1b*.

Soweit unterscheidet sich die Modellierung nicht von der der verteilten Modelle. Da im globalen Modell pro Verbindung nur ein Token vorhanden ist, muß zusätzlich mit dem Token der Verbindung, mit der zur Dienstlogik gesprungen wurde, zu dem Punkt ins BCSM zurückgesprungen werden, aus dem die Dienstlogik aufgerufen wurde (Transition: *CH_0*). Das Token muß das für den Dienstanutzer gesetzte Bit, das anzeigt, ob dieser die Verbindung auf Warteposition gesetzt hat (*oh_bit*), und die alten Armierungslisten enthalten.

In der letzten Komponente des Tokens, das in die *Feature_queue* zurückgelegt wird, hat man die Möglichkeit, die Priorität des Rücksprungs zum BCSM zu bestimmen (0 entspricht hoher Priorität, 1 entspricht niedriger Priorität). In der hier vorgestellten Modellierung hat das Token, mit dem zuvor zur Dienstlogik gesprungen wurde, höhere Priorität als das Token, das neu in das BCSM kommt.

Service Feature: Call Hold (CH)

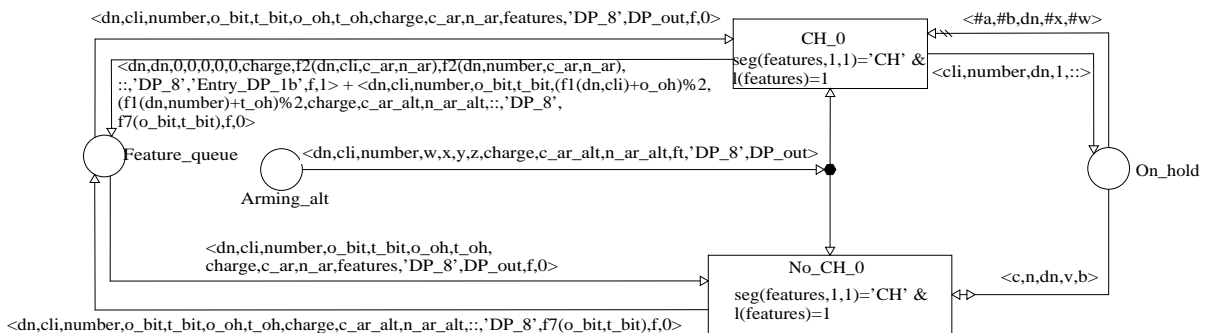


Abb. 43: Service Feature Call Hold

4.3.4. Änderungen bezüglich des Rücksprungs zum BCSM

Die Service Features Call Toggle und Call Retrieve bieten die Möglichkeit, eine zuvor auf Warteposition gesetzte Verbindung wieder zu aktivieren. In dem globalen Modell liegt für auf Warteposition gesetzte Verbindungen immer ein Token im BCSM, bei dem das *oh_bit* des Dienstanutzers auf 1 gesetzt ist. Wird nun eines der Service Features Call Toggle bzw. Call Retrieve ausgeführt, so wird nicht wie im verteilten Modell eine zusätzliche Marke ins BCSM gelegt, sondern die Marke, die bereits im BCSM liegt, wird aktualisiert. D.h. das *oh_bit* des Dienstanutzers wird auf 0 zurückgesetzt. Diese Aktualisierung führt das in Abb. 44 gezeigte Teilnetz durch.

Ruecksprungmechanismus fuer Service Features Call Toggle und Call Retrieve

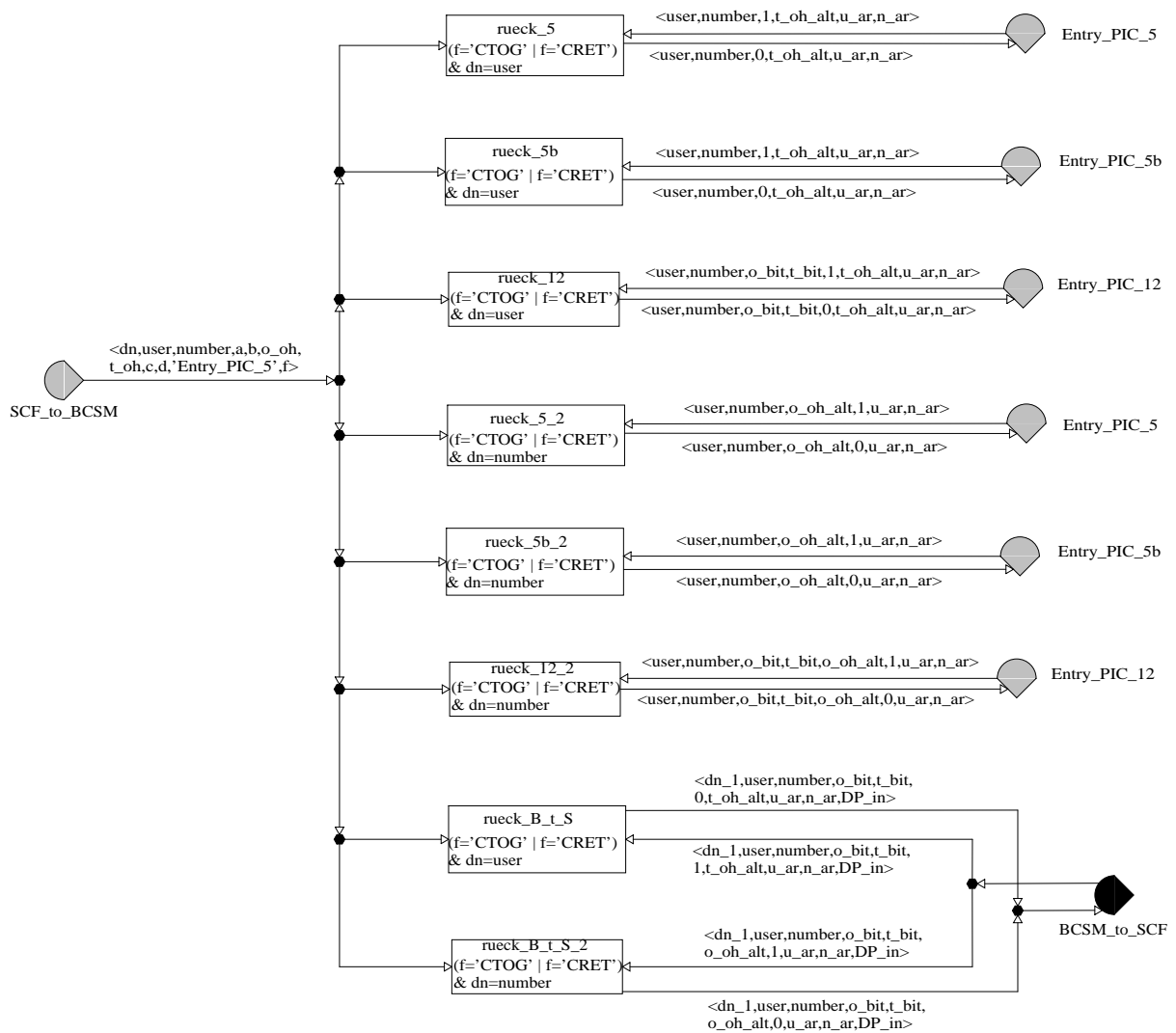


Abb. 44: Zusätzlicher Rücksprungmechanismus

4.4. Analyse und Verifikation

Folgende Untersuchungen werden an diesem Modell durchgeführt:

1. Verhalten beim Verbindungsaufbau
2. Verhalten beim Verbindungsabbau
3. Verhalten eines Anrufers ohne abonnierten Dienst bzw. mit abonniertem Call Hold-Dienst
4. Verhalten eines Angerufenen ohne abonnierten Dienst bzw. mit abonniertem Call Waiting-Dienst
5. Phasentrennung und Lebendigkeitseigenschaft
6. Besetzt-Verhalten
7. globales Verhalten bei abonniertem Call Hold-Dienst

8. globales Verhalten bei abonniertem Call Waiting-Dienst
9. globales Verhalten bei abonniertem Call Hold- und Call Waiting-Dienst

4.4.1. Verhalten beim Verbindungsaufbau

Das Verhalten beim Verbindungsaufbau wird hier untersucht, um zeigen zu können, daß die diesen Teil betreffenden Verhaltensweisen des globalen Modells bereits im dritten Modell enthalten sind.

Um ein übersichtliches und dabei doch aussagekräftiges Ergebnis zu erhalten, muß durch Änderungen an einigen Netzen sichergestellt werden, daß die Analyse nach jedem auftretenden Fehler oder vorzeitigem Auflegen, spätestens jedoch nach dem Erreichen der aktiven Phase in einer Totmarkierung endet. Dadurch beschränkt sich der Erreichbarkeitsgraph auf den hier zu untersuchenden Verbindungsaufbau.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

```

New Initial Marking:

ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2>
NUMBER_POOL: 1<2>

Reachability Graph bcsms2c-gls1

26 Markings computed.
(2 Dead Markings)
(0 Pseudo Occurrence Steps)
(42 Occurrence Steps)

2 Dead Markings bcsms2c-gls1

M-21 DP_7_NOT_ARMED
M-24
+++ dead +++

ENTRY_PIC_1: 1<1>
NUMBER_POOL: 1<2>

M-17 TD_9_A M-6 TD_9_A
M-8
+++ dead +++

ENTRY_PIC_1: 1<1>
IDLE: 1<2>
NUMBER_POOL: 1<2>

End 2 Dead Markings bcsms2c-gls1

```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
ok	ATTEMPT_1A
DP_1b_not_armed	DP_1B_NOT_ARMED
authorized	AUTHORIZED_1B
DP_1_not_armed	DP_1_NOT_ARMED
collected_info	COLLECTED_INFO_2
DP_2_not_armed	DP_2_NOT_ARMED
analysed_info	ANALYSED_INFO_3
DP_3_not_armed	DP_3_NOT_ARMED
interpret	ROUTE_SELECTED_4A
authorized_for_number	CALL_SETUP_4B
alert	ALERTED_4C
DP_25_not_armed	DP_25_NOT_ARMED
answer	ANSWER_4C, ANSWER_4D
DP_7_not_armed	DP_7_NOT_ARMED
O_Disconnect	OTDISCONNECT_12, ABANDON_4D, ABANDON_4B, ABANDON_4A, ABANDON_3, ABANDON_2, ABANDON_1B, ABANDON_4C_2, ABANDON_4C_1, OTDISCONNECT_5, OTDISCONNECT_5B
DP9/10/6_not_armed	TD_9_A, TD_9_B
route_busy	ROUTE_BUSY_4A
Route_select_Failure	ROUTE_FAILURE_4A
Exception_DP4	DP_4_NOT_ARMED
no_answer	NO_ANSWER_4D, NO_ANSWER_4C
Exception_DP6	DP_6_NOT_ARMED
route_failure_4c	ROUTE_FAILURE_4C
Mid_Call_4c	MID_CALL_4C
DP_8_not_armed	DP_8_NOT_ARMED_1, DP_8_NOT_ARMED_2, DP_8_NOT_ARMED_3, DP_8_NOT_ARMED_4, DP_8_NOT_ARMED_5
route_failure_4d	ROUTE_FAILURE_4D
Mid_Call_4d	MID_CALL_4D
Epsilon	alle restlichen Transitionen

Obiger Homomorphismus beinhaltet alle möglichen Zustandsübergänge bis zum Erreichen der aktiven Phase.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-gls1

20 States computed.
 (2 critical Markings)
 (35 Edges, 26 Labels)

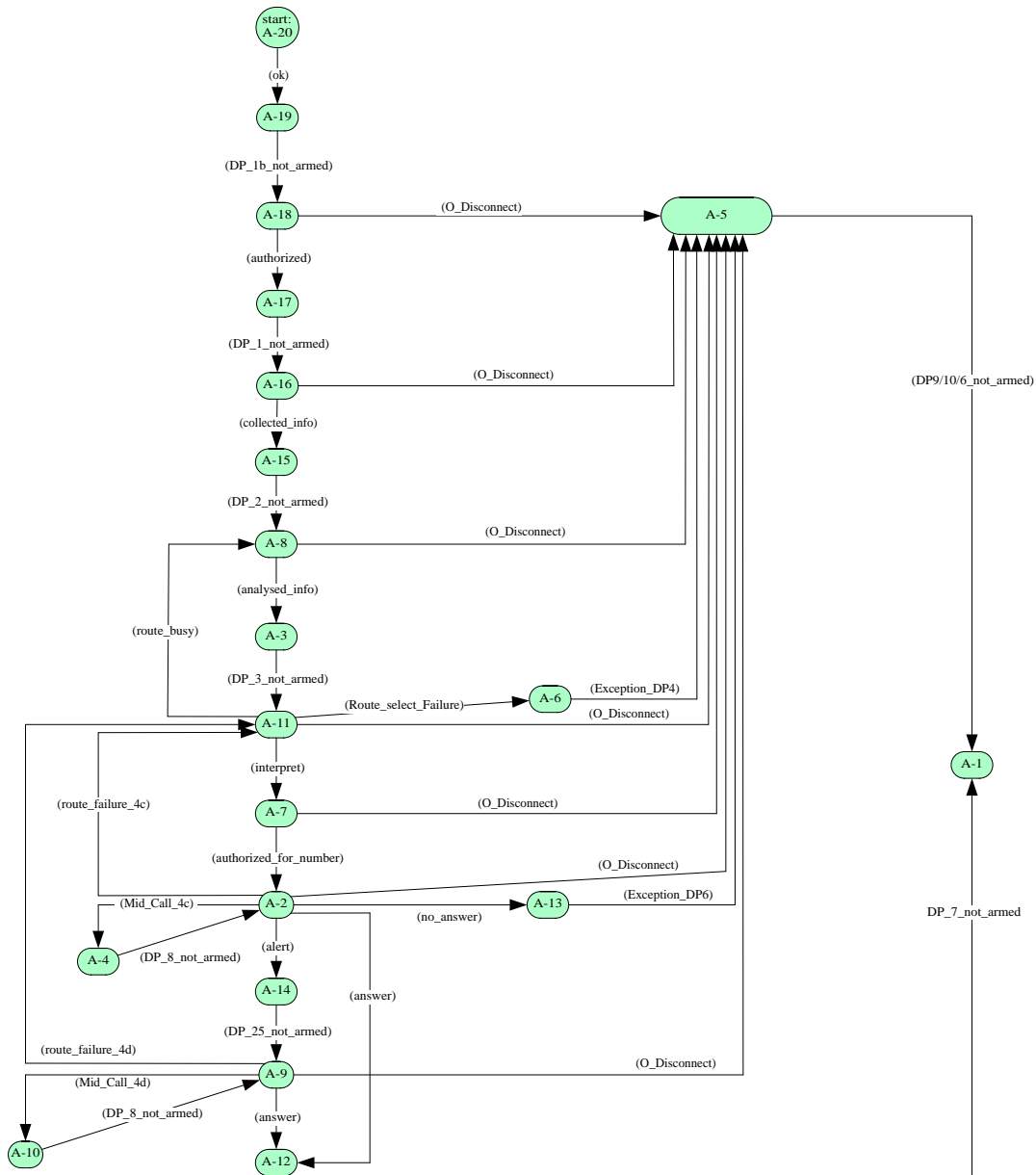


Abb. 45: Minimalautomat "Verbindungsaufbau"

Der in Abb. 45 dargestellte Minimalautomat stimmt mit dem im dritten Modell überein (vgl. Abschnitt 3.4.4.10.). Auch hier existiert kein dem *Busy DP* entsprechender Zustand, da durch die oben genannten Modifikationen einiger Netze kein zweiter Verbindungsaufbau initiiert werden kann.

Somit gilt, daß das Verhalten des globalen Modells beim Verbindungsaufbau im dritten Modell bereits enthalten ist. Durch die Abstraktion werden also keine neuen Verhaltensweisen hinzugefügt.

4.4.2. Verhalten beim Verbindungsabbau

Wie beim Verbindungsaufbau wird die Analyse des Verbindungsabbaus durchgeführt, um zeigen zu können, daß das diesbezügliche Verhalten des globalen Modells auch im dritten Modell enthalten ist.

Um ein übersichtliches und doch aussagekräftiges Ergebnis zu erhalten, wird das Zurücklegen der Marke für den Anrufer auf die Stelle *Idle* nach Beenden der Verbindung durch Änderungen an einigen Netzen verhindert. Durch eine entsprechende Anfangsmarkierung wird festgelegt, daß die Analyse erst mit der aktiven Phase der Verbindung beginnt. Dadurch ist der komplette Verbindungsaufbau im Erreichbarkeitsgraphen nicht enthalten.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:

ENTRY_PIC_1: 1<1>
ENTRY_PIC_5: 1<1,2,0,0,::,::>
NUMBER_POOL: 1<2>

Reachability Graph bcsms2c-gls1

22 Markings computed.
(1 Dead Marking)
(0 Pseudo Occurrence Steps)
(29 Occurrence Steps)

1 Dead Marking bcsms2c-gls1

M-23 TD_9_A M-18 TD_17_1A
M-21
+++ dead +++

ENTRY_PIC_1: 1<1>
NUMBER_POOL: 1<2>

End 1 Dead Marking bcsms2c-gls1
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
O_Disconnect	OTDISCONNECT_12, OTDISCONNECT_5, OTDISCONNECT_5B
T_Disconnect	TODISCONNECT_5B, TODISCONNECT_12
DP_9_not_armed	TD_9_A, TD_9_B
DP_17_not_armed	TD_17_1A, TD_17_1B
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-gls1

8 States computed.
 (1 critical Markings)
 (8 Edges, 4 Labels)

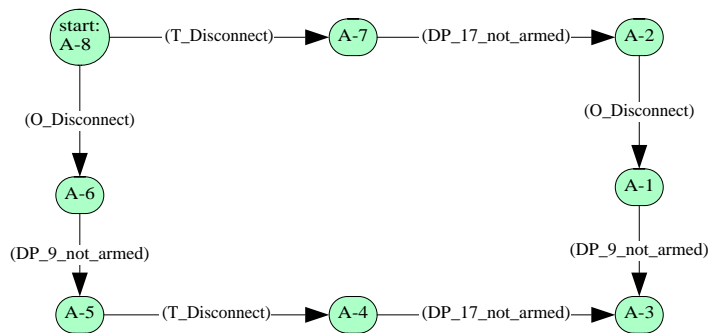


Abb. 46: Minimalautomat "Verbindungsabbau"

Vergleicht man obigen Minimalautomaten mit dem im dritten Modell gefundenen (vgl. Abschnitt 3.4.4.11.), so sieht man, daß im globalen Modell nur Zustandsübergänge existieren, die es im dritten Modell auch gibt. Alle Aktionsfolgen, die im Minimalautomaten des dritten Modells vom Startzustand über den Zustand A-2 zum toten Zustand A-5 führen, sind im globalen Modell nicht enthalten.

Durch die Abstraktion werden also den Verbindungsabbau betreffend keine neuen Verhaltensweisen hinzugefügt.

4.4.3. Anrufer

4.4.3.1. Anrufer ohne abonnierten Dienst

Hat der Anrufer keinen Dienst abonniert, sollte sein Verhalten dem in den ersten drei Modellen entsprechen.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2>
NUMBER_POOL: 1<2>
```

Reachability Graph bcsms2c-gls1

```
145 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(355 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	ATTEMPT_1A
Nummer_waehlen	COLLECTED_INFO_2
auflegen	DP_9_DYNAMIC_ARMED, DP_9_STATIC_ARMED, TD_9_B, TD_9_A
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-gls1

```
3 States computed.
(4 Edges, 3 Labels)
```

Dieses Modell enthält das gleiche Verhalten eines Anrufers wie die ersten drei Modelle (vgl. Abschnitt 3.2.2.1.1.).

4.4.3.2. Anrufer mit abonniertem Call Hold-Dienst

Hat der Anrufer den Call Hold-Dienst abonniert, sollte sein Verhalten dem im ersten und dritten Modell gefundenen entsprechen.

Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2>
NUMBER_POOL: 1<2>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9,DP_8,ENTRY_DP_1B> +
1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8>
```

Reachability Graph bcsms2c-gls1

793 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(1999 Occurrence Steps)

Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	ATTEMPT_1A
Nummer_waehlen	COLLECTED_INFO_2
auflegen	DP_9_STATIC_ARMED, TD_9_B, DP_9_DYNAMIC_ARMED, TD_9_A
Call_Hold	CH_0
Call_Retrieve	CRET_0_2, CRET_0_1
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-gls1

6 States computed.
(9 Edges, 5 Labels)

Das Ergebnis stimmt mit dem im ersten und dritten Modell gefundenen überein (vgl. Abschnitt 3.2.2.1.2.).

4.4.4. Angerufener

4.4.4.1. Angerufener ohne abonnierten Dienst

Hat der Angerufene keinen Dienst abonniert, sollte sein Verhalten dem in den ersten drei Modellen entsprechen.

Der Untersuchung liegt die gleiche Erreichbarkeitsanalyse wie bei einem Anrufer ohne abonnierte Dienste (vgl. Abschnitt 4.4.3.1.) zugrunde:

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
abheben	ANSWER_4D, ANSWER_4C
auflegen	TD_17_1B, TD_17_1A, DP_17_DYNAMIC_ARMED, DP_17_STATIC_ARMED
Epsilon	alle restlichen Transitionen

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsms2c-gls1

2 States computed.
(2 Edges, 2 Labels)
```

Wie erwartet, ist in diesem Modell das gleiche Verhalten eines Angerufenen enthalten wie in den ersten drei Modellen (vgl. Abschnitt 3.2.2.2.).

4.4.4.2. Angerufener mit abonniertem Call Waiting-Dienst

Im folgenden wird das Verhalten eines Angerufenen mit abonniertem Call Waiting-Dienst untersucht.

Neben dem Abheben und Auflegen sollte bei diesem Angerufenen nach dem Abheben ein Call Waiting möglich sein. Danach muß er beliebig oft zwischen seinen beiden Verbindungen hin- und herschalten können. Nach dem Auflegen muß der Angerufene mittels Call Retrieve einen noch wartenden Anruf aktivieren, bevor er endgültig auflegen kann oder ein erneutes Call Waiting möglich ist.

Grundlage der Untersuchung des Verhaltens eines Angerufenen, der den Call Waiting-Dienst abonniert hat, ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:

ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2>
NUMBER_POOL: 1<2>
SERVICE_STRUCTURE: 1<2,CW,0.0,CRET,::,DP_17,ENTRY_PIC_5> +
                    1<2,CW,0.0,CTOG,::,DP_8,ENTRY_PIC_5> +
                    1<2,CW,0.0,CW,::,DP_5,ENTRY_PIC_5>
STATIC_ARMING: 1<2,DP_17> + 1<2,DP_5> + 1<2,DP_8>

Reachability Graph bcsms2c-gls1

1289 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(3788 Occurrence Steps)

Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
Klingeln	ALERTED_4C
auflegen_Anrufer	NO_ANSWER_4D, ROUTE_FAILURE_4D, ABANDON_4D
abheben	ANSWER_4D, ANSWER_4C
auflegen	TD_17_1B, TD_17_1A, DP_17_DYNAMIC_ARMED, DP_17_STATIC_ARMED

Call_Waiting	CW_0_1
Call_Toggle	CTOG_0
Call_Retrieve	CRET_0_1, CRET_0_2
Epsilon	alle restlichen Transitionen

Um ein aussagekräftiges Ergebnis zu erhalten, werden außer den dienstespezifischen Blöcken *Call_Waiting*, *Call_Toggle* und *Call_Retrieve* die Blöcke *Klingeln* und *auflegen_Anrufer* zu obigem Homomorphismus für das Verhalten eines Angerufenen ohne abonnierte Dienste hinzugefügt. Der Block *auflegen_Anrufer* enthält lediglich die Transitionen, die ein Beenden der Verbindung auf der Seite des Anrufers während des Klingelns beim Angerufenen repräsentieren.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-g1s1

6 States computed.
(12 Edges, 7 Labels)

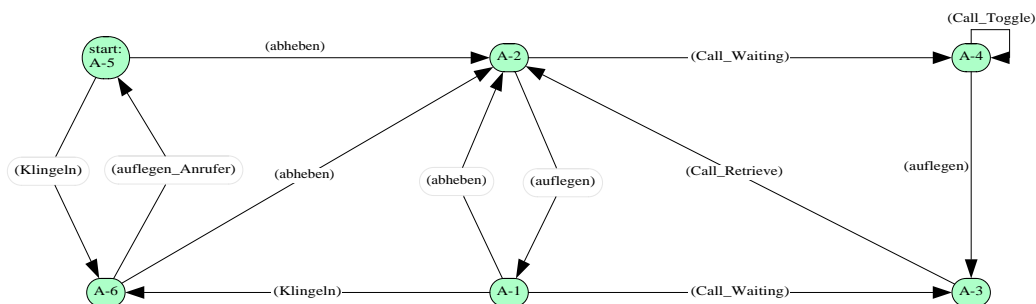


Abb. 47: Minimalautomat "Angerufener mit Call Waiting-Dienst"

Am Minimalautomat in Abb. 47 ist folgendes Verhalten zu erkennen:

Im Startzustand kann der Angerufene abheben oder es klingelt bei ihm. Das Klingeln wird entweder gefolgt vom Auflegen des Anrufers, wodurch der Angerufene in den Startzustand übergeht oder vom Abheben des Angerufenen. Danach kann der Angerufene auflegen (Übergang von A-2 nach A-1) oder er wird erneut angerufen und es kommt zur Ausführung des Call Waiting-Dienstes, beginnend mit dem Feature Call Waiting (Übergang von A-2 nach A-4).

Erfolgt ein Anruf während der Prozedur des Auflegens, noch bevor der Angerufene in den Grundzustand zurückgekehrt ist, so ist er für den Anrufer besetzt und es wird ebenfalls das Feature Call Waiting ausgeführt (Übergang von A-1 nach A-3).

Im Zustand A-4, der für den Angerufenen durch eine aktive Verbindung und eine Verbindung auf Warteposition charakterisiert ist, kann mit dem Feature Call Toggle zwischen den beiden Verbindungen hin- und hergeschaltet werden. Dabei wird der Zustand nicht gewechselt.

Schließlich wird die gerade aktive Verbindung beendet (Übergang von A-4 nach A-3). Im Zustand A-3 hat der Angerufene aufgelegt und es liegt noch eine Verbindung auf Warteposition. Er ist nun gezwungen, diese Verbindung mit dem Feature Call Retrieve zu reaktivieren. So gelangt er zurück in den Zustand A-2.

Man erkennt, daß das gefundene Verhalten mit dem erwarteten und in bezug auf die Dienstausführung auch mit dem des dritten Modells übereinstimmt.

4.4.5. Phasentrennung und Lebendigkeitseigenschaft

Bei der Phasentrennung und der Lebendigkeitseigenschaft ist in diesem Modell zu beachten, daß beim Eintreten in die Datenphase nicht zwischen den Teilnehmern unterschieden wird, diese jedoch die Datenphase getrennt verlassen können. Es wird untersucht, ob beide Teilnehmer beim Betreten einer neuen Datenphase die alte verlassen haben (Phasentrennung) und ob von jedem Systemzustand aus immer wieder eine Datenphase erreichbar ist (Lebendigkeit).

Der Untersuchung liegt die gleiche Erreichbarkeitsanalyse wie bei einem Anrufer ohne abonnierte Dienste (vgl. Abschnitt 4.4.3.1.) zugrunde:

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
O_T_Eintritt_in_Datenphase	DP_7_NOT_ARMED
O_Austritt_aus_Datenphase	OTDISCONNECT_12, OTDISCONNECT_5, OTDISCONNECT_5B
T_Austritt_aus_Datenphase	TODISCONNECT_5B, TODISCONNECT_12
Epsilon	alle restlichen Transitionen

Es gibt, wie oben erwähnt, nur einen Block, der den Eintritt beider Teilnehmer in die Datenphase repräsentiert.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsms2c-g1s1
4 States computed.
(5 Edges, 3 Labels)
```

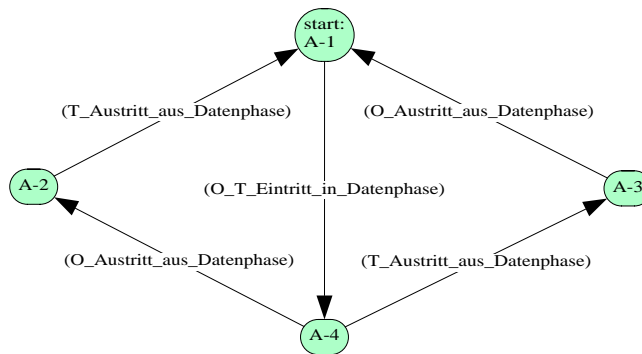


Abb. 48: Minimalautomat "Phasentrennung und Lebendigkeit"

Der Minimalautomat in Abb. 48 zeigt, daß sowohl die Phasentrennung als auch die Lebendigkeitseigenschaft in diesem Modell gelten.

Wird die Datenphase aus dem Startzustand heraus betreten, so kann dies erneut nur geschehen, wenn vorher sowohl O als auch T wieder aus der Datenphase ausgetreten sind (Phasentrennung).

Aus jedem Zustand des Automaten heraus kann das System in den Startzustand gelangen, von dem aus der Eintritt in eine neue Datenphase wieder möglich ist (Lebendigkeitseigenschaft). Da der Erreichbarkeitsgraph nur eine Zusammenhangskomponente besitzt, sind alle Homomorphismen auf ihm schlicht. Die Lebendigkeitseigenschaft gilt also nicht nur in der vergrößernden Sichtweise, sondern auch im Erreichbarkeitsgraphen selbst.

4.4.6. Besetzt-Verhalten

An diesem Modell wird das Besetzt-Verhalten untersucht, um es mit dem der Modelle unter verteilter Sichtweise vergleichen zu können. Es wird erwartet, daß dieses Modell das gleiche Besetzt-Verhalten aufweist wie das zweite und das dritte.

Der Untersuchung liegt die gleiche Erreichbarkeitsanalyse wie bei einem Anrufer ohne abonnierte Dienste (vgl. Abschnitt 4.4.3.1.) zugrunde:

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
Verbindung_1_2_busy	busy_4c
Epsilon	alle restlichen Transitionen

Der Homomorphismus ist definiert wie in den anderen Modellen. Da es die dort benutzte Transition *busy_8a* unter globaler Sichtweise nicht gibt, wird die entsprechende Transition *busy_4c* benutzt, um ein "besetzt" des Angerufenen anzuzeigen. Die Stellen *Entry_PIC_5*,

Entry_PIC_5b und *Entry_PIC_12* sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsmcs2c-gls1
1 State computed.
(2 Edges, 2 Labels)

Q-Markings : bcsmcs2c-gls1

Q-::

Q-1
ENTRY_PIC_12: 1<1,2,1,0,0,0,::,::>

End Q-Markings : bcsmcs2c-gls1
```

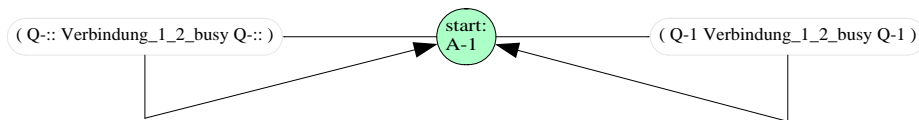


Abb. 49: Minimalautomat "Besetzt-Verhalten"

Der in Abb. 49 gezeigte Minimalautomat und die entsprechenden Q-Markierungen zeigen, daß der Anrufer die Verbindung beenden und immer wieder versuchen kann anzurufen, während der Angerufene noch nicht beendet hat, erkennbar am Bit 0 an der vierten Stelle der Marke auf der Stelle *Entry_PIC_12* in der Q-Markierung *Q-1*.

Der Angerufene kann parallel zu den Aktionen des Anrufers im Zustand *PIC-12* bleiben, versuchen, einen Dienst auszuführen (Schalten von *Mid_Call_12*), oder seinerseits die Verbindung beenden (Schalten von *TODisconnect_12*). Ist er im *PIC-12*, das ist der Zustand, in dem aus seiner Sicht die Verbindung noch besteht - entspricht in den verteilten Modellen einer Marke auf der Stelle *Entry_PIC_10_after_accept* -, wenn der Anrufer das Besetztzeichen erhält (Schalten von *busy_4c*), so erzeugt das in obigem Automaten die Kante (*Q-1 Verbindung_1_2_busy Q-1*). In den anderen Fällen befindet sich die Marke im *DP-8* oder *DP-17*. So erklärt sich die Kante (*Q-:: Verbindung_1_2_busy Q-::*)

Die Q-Markierungen zeigen auch, daß keine Marken auf den Stellen *Entry_PIC_5* und *Entry_PIC_5b* liegen. Dies ist korrekt, denn der Anrufer hat die Verbindung bereits beendet. Das gefundene Ergebnis stimmt somit mit dem Besetzt-Verhalten in den unsynchronisierten verteilten Modellen überein.

4.4.7. Globales Verhalten bei abonniertem Call Hold-Dienst

Um das Verhalten des Modells bei abonniertem Call Hold-Dienst zu untersuchen, werden mittels mehrerer Homomorphismen auf verschiedenen Erreichbarkeitsgraphen vergrößernde Sichtweisen erzeugt.

4.4.7.1. Analyse 1

Die Analyse des Call Hold-Dienstes in der Konstellation, wie sie auch in den Modellen eins und drei möglich ist (ein Anrufer mit Call Hold-Dienst und ein Angerufener ohne Dienste), wird hier zu Vergleichszwecken ebenfalls durchgeführt. Es wird das gleiche Ergebnis wie im dritten, unsynchronisierten Modell erwartet.

Der Untersuchung liegt die dieselbe Erreichbarkeitsanalyse wie bei einem Anrufer mit abonniertem Call Hold-Dienst (vgl. Abschnitt 4.4.3.2.) zugrunde:

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
accept	DP_7_NOT_ARMED
Epsilon	alle restlichen Transitionen

Die Stellen *Entry_PIC_5*, *Entry_PIC_5b*, *Entry_PIC_12* und *On_hold* sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsms2c-gls1

2 States computed.
(3 Edges, 2 Labels)

Q-Markings : bcsms2c-gls1

Q-::

Q-1
ENTRY_PIC_12: 1<1,2,0,1,1,0,::,::>
ENTRY_PIC_5: 1<1,2,0,0,DP_9,::>
ON_HOLD: 1<1,2,1,1,::>

Q-2
ENTRY_PIC_12: 1<1,2,0,1,1,0,::,::>
ON_HOLD: 1<1,2,1,1,::>

Q-3
ENTRY_PIC_5: 1<1,2,0,0,::,::>

End Q-Markings : bcsms2c-gls1
```

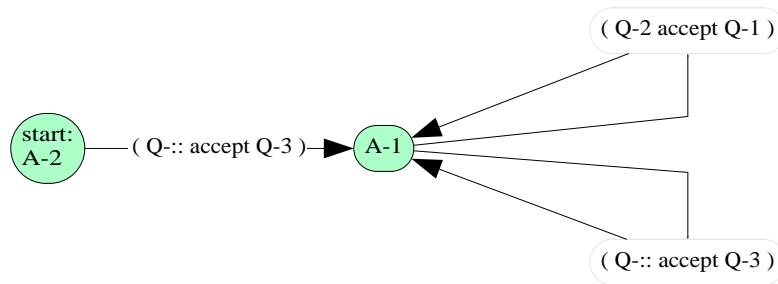


Abb. 50: Minimalautomat "Call Hold-1"

Der Minimalautomat in Abb. 50 zeigt, daß das BCSM vom Startzustand aus zuerst mit der Verbindung zwischen 1 und 2 in die aktive Phase kommen muß (Kante $(Q-:: \text{accept } Q-3)$ von A-2 zu A-1), bevor etwas anderes passieren kann. Beide Teilnehmer können danach die Verbindung beenden und eine neue aufbauen (Schleife $(Q-:: \text{accept } Q-3)$ um A-1).

Der Anrufer kann die Verbindung auf Warteposition setzen und denselben Teilnehmer erneut anrufen. Beendet der Angerufene die erste Verbindung bevor ihn der Aufbauwunsch des Anrufers erreicht, erhält man die Q-Markierung Q-2. Nun ist der Angerufene nicht mehr besetzt und der neue Aufbauwunsch wird akzeptiert, wodurch die Q-Markierung Q-1 erreicht wird. Dieses Verhalten ist nicht im ersten, synchronisierten, wohl aber im dritten, unsynchronisierten Modell enthalten.

Ist der Angerufene für den neuen Aufbauwunsch besetzt, so kann keine neue Verbindung zustandekommen. Der Anrufer muß dann zu der ersten Verbindung zurückkehren, wodurch wieder Q-:: erreicht wird. Da bei diesem Szenario die Transition *accept* nicht schaltet, ist dafür keine weitere Kante im Minimalautomaten enthalten.

Das Ergebnis stimmt mit dem überein, was man von der gewählten Konfiguration erwartet und entspricht dem im unsynchronisierten, dritten Modell.

4.4.7.2. Analyse 2

Die folgende Untersuchung mit einem Anrufer und zwei Angerufenen und dem vom Anrufer abonnierten Call Hold-Dienst ist bei den anderen drei Modellen aus Komplexitätsgründen praktisch nicht möglich.

Die beiden Angerufenen haben identisches Benutzerprofil, weshalb sie auch dieselben Möglichkeiten in ihren Aktionen haben sollten. Es wird außerdem erwartet, daß aus Sicht der Angerufenen dasselbe Verhalten möglich ist, wie es sich in der vorhergehenden Analyse ergeben hat. Zusätzlich sollte der Anrufer mittels des Dienstes je eine vollwertige Verbindung zu jedem der Angerufenen aufbauen und nebeneinander unterhalten können, unabhängig davon, ob einer der beiden eine vom Anrufer auf Warteposition gesetzte Verbindung beendet hat. Das ist in der vorherigen Analyse für das Zustandekommen einer mittels Call Hold initiierten Verbindung notwendig gewesen.

Grundlage dieser Untersuchung ist folgende Erreichbarkeitsanalyse:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2> + 1<3>
NUMBER_POOL: 1<2> + 1<3>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9,DP_8,ENTRY_DP_1B> +
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1>
STATIC_ARMING: 1<1,DP_8>
```

Reachability Graph bcsms2c-gls1

```
23196 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(84982 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
accept	DP_7_NOT_ARMED
Epsilon	alle restlichen Transitionen

Die Stelle *Entry_PIC_5* sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

Minimal Automaton bcsms2c-gls1

```
4 States computed.
(16 Edges, 6 Labels)
```

Q-Markings : bcsms2c-gls1

Q-::

```
Q-1
ENTRY_PIC_5: 1<1,3,0,0,DP_9,::>
```

```
Q-2
ENTRY_PIC_5: 1<1,2,0,0,DP_9,::>
```

```
Q-3
ENTRY_PIC_5: 1<1,3,0,0,::,::>
```

```
Q-4
ENTRY_PIC_5: 1<1,2,0,0,::,::>
```

```
Q-5
ENTRY_PIC_5: 1<1,2,0,0,DP_9,::> + 1<1,3,1,0,::,::>
```

Q-6
 ENTRY_PIC_5: 1<1,3,1,0,::,::>

Q-7
 ENTRY_PIC_5: 1<1,2,1,0,::,::> + 1<1,3,0,0,DP_9,::>

Q-8
 ENTRY_PIC_5: 1<1,2,1,0,::,::>

End Q-Markings : bcsms2c-gls1

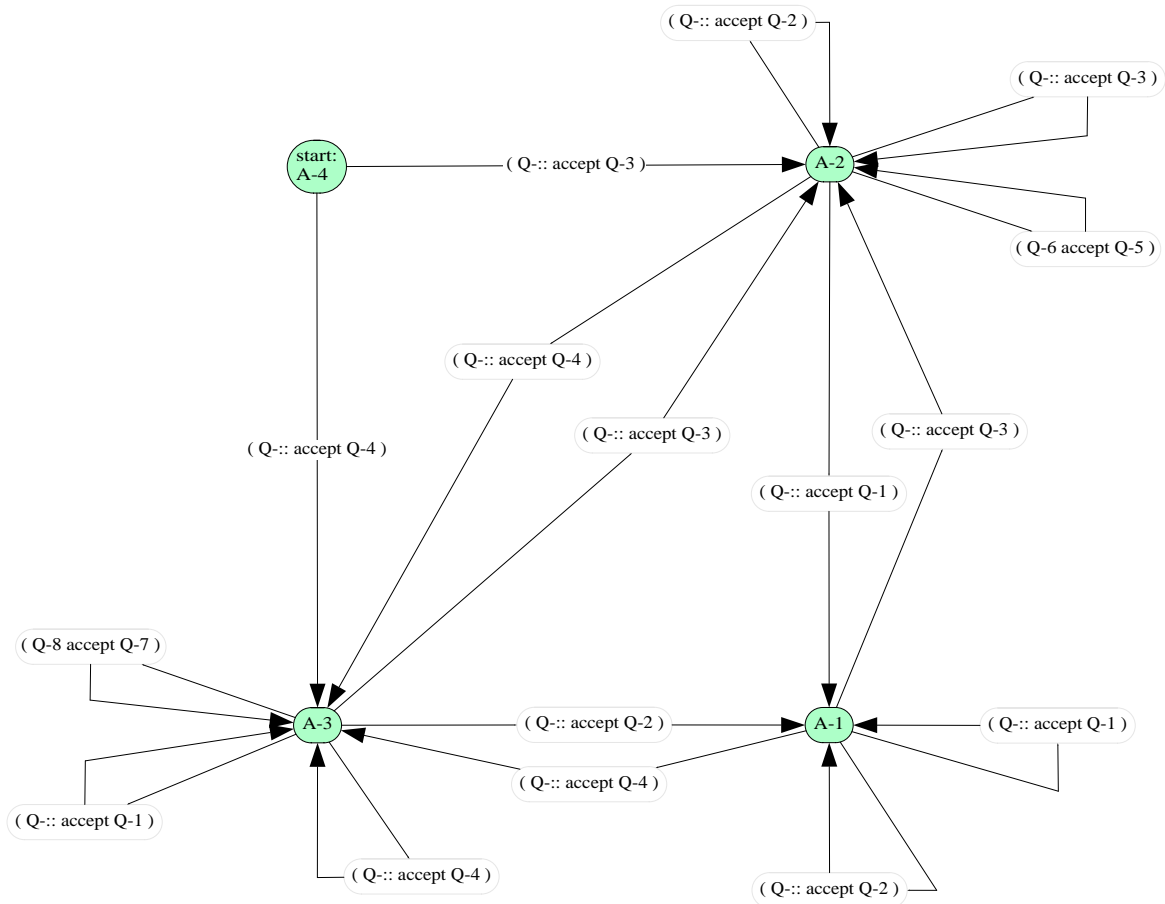


Abb. 51: Minimalautomat "Call Hold-2"

Der Minimalautomat in Abb. 51 ist symmetrisch. Der Anrufer (1) kann zuerst eine Verbindung zu 2 (Kante (Q-:: accept Q-4) von A-4 zu A-3) oder zu 3 (Kante (Q-:: accept Q-3) von A-4 zu A-2) aufbauen. Im Zustand A-3 kann diese Verbindung beendet und immer wieder neu aufgebaut werden (Schleife (Q-:: accept Q-4)). Außerdem kann 1 in diesem Zustand die Verbindung zu 2 auf Warteposition setzen und 3 anrufen (Schleife (Q-8 accept Q-7)). Derselbe Ablauf liegt der Schleife (Q-:: accept Q-1) zugrunde, mit dem Unterschied, daß hier die Marke zur ersten Verbindung durch den Versuch, einen Dienst auf der Seite des Angerufenen auszuführen, nicht auf der Stelle *Entry_PIC_5* liegt, sondern über die Transition *Mid_Call_5* in den *Mid_Call DP* gelangt ist.

Es ist ebenfalls möglich, daß nach dem Beenden der Verbindung zu 2 eine Verbindung zu 3 aufgebaut wird (Kante ($Q:: \text{accept } Q-3$) von $A-3$ zu $A-2$). Schließlich kann 1, nachdem er die Verbindung zu 2 auf Warteposition gesetzt hat, diesen erneut anrufen. Die Verbindung kommt zustande, wenn 2 die erste Verbindung beendet hat. Dies entspricht der Kante ($Q:: \text{accept } Q-2$) von $A-3$ zu $A-1$. Das kann danach beliebig oft geschehen (Schleife ($Q:: \text{accept } Q-2$)). Im Zustand $A-1$ kann 1 nach Beenden aller Verbindungen eine neue zu 3 aufbauen (Kante ($Q:: \text{accept } Q-3$) von $A-1$ zu $A-2$). Dafür ist es nicht von Belang, ob auch 2 die Verbindung zu 1 beendet hat. Außerdem kann 1 nach Wiederaufnahme und erneutem Setzen der ersten Verbindung auf Warteposition 3 anrufen (Schleife ($Q:: \text{accept } Q-1$)). Schließlich kann in $A-1$ eine neue Verbindung zu 2 zustandekommen (Kante ($Q:: \text{accept } Q-4$) von $A-1$ zu $A-3$).

Diese Szenarien nach dem ersten Zustandekommen eine Verbindung zwischen 1 und 2 existieren entsprechend für den Fall, daß zuerst eine Verbindung zwischen 1 und 3 aufgebaut wird.

Das Ergebnis zeigt, daß die Teilnehmer so Verbindungen aufbauen und beenden können, wie es der gewählten Konfiguration entspricht.

4.4.7.3. Analyse 3

Das Ziel der folgenden Analyse ist es, das Modellverhalten bei einem Anrufer und zwei Angerufenen zu untersuchen, wobei ein Angerufener den Call Hold-Dienst abonniert hat.

Es wird erwartet, daß der sich ergebende Automat nicht symmetrisch ist, da die beiden Angerufenen unterschiedliche Benutzerprofile haben. Der Anrufer sollte mit jedem der Angerufenen beliebig oft eine Verbindung aufbauen und wieder beenden können. Außerdem sollte der Dienstabonnent mit Hilfe des Call Hold-Dienstes eine Verbindung mit dem zweiten Angerufenen aufbauen können, während eine Verbindung zum Anrufer besteht.

Grundlage dieser Untersuchung ist folgende Erreichbarkeitsanalyse:

```
New Initial Marking:
```

```
ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2> + 1<3>
NUMBER_POOL: 1<2> + 1<3>
SERVICE_STRUCTURE: 1<2,CH,0.0,CH,DP_9,DP_8,ENTRY_DP_1B> +
    1<2,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1>
STATIC_ARMING: 1<2,DP_8>
```

```
Reachability Graph bcsms2c-gls1
```

```
30882 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(124047 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
accept	DP_7_NOT_ARMED
Epsilon	alle restlichen Transitionen

Die Stelle *Entry_PIC_5* sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```

Minimal Automaton bcsmcs2c-gls1

3 States computed.
(9 Edges, 4 Labels)

Q-Markings : bcsmcs2c-gls1
Q-::

Q-1
ENTRY_PIC_5: 1<1,2,0,0,::,::>

Q-2
ENTRY_PIC_5: 1<1,3,0,0,::,::>

Q-3
ENTRY_PIC_5: 1<2,3,0,0,DP_9,::>

Q-4
ENTRY_PIC_5: 1<1,2,0,1,::,::> + 1<2,3,0,0,DP_9,::>

Q-5
ENTRY_PIC_5: 1<1,2,0,1,::,::>

End Q-Markings : bcsmcs2c-gls1

```

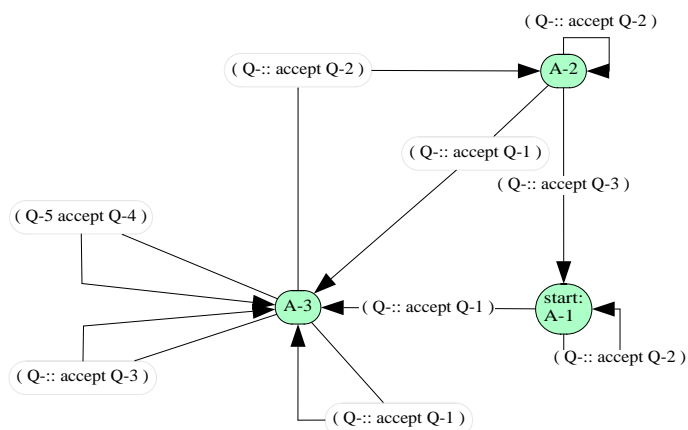


Abb. 52: Minimalautomat "Call Hold-3"

Der Minimalautomat in Abb. 52 zeigt, daß im Startzustand beliebig oft eine Verbindung zwischen dem Anrufer (1) und dem Angerufenen 3 zustandekommen kann (Schleife ($Q-::\text{ accept } Q-2$)).

Das Akzeptieren einer Verbindung zum Angerufenen 2 führt mit der Kante ($Q-::\text{ accept } Q-1$) aus dem Startzustand $A-1$ zum Zustand $A-3$. Das danach beliebig oft mögliche erneute Zustandekommen einer Verbindung zwischen 1 und 2 wird durch die Schleife ($Q-::\text{ accept } Q-1$) um $A-3$ repräsentiert. Der Dienstabonnent 2 kann in diesem Zustand die Verbindung zu 1 auf Warteposition setzen und eine zu 3 aufbauen (Schleife ($Q-::\text{ accept } Q-3$)). Derselbe Ablauf liegt der Schleife ($Q-5\text{ accept } Q-4$) zugrunde, mit dem Unterschied, daß hier die Marke zur Verbindung zwischen 1 und 2 auf der Stelle $Entry_PIC_5$ liegt und nicht wie im vorigen Fall durch den Versuch vom Anrufer 1, einen Dienst auszuführen, über die Transition Mid_Call_5 in den $Mid_Call\ DP$ gelangt ist.

Schließlich kann 1 im Zustand $A-3$ nach Beenden aller Verbindungen wieder eine Verbindung zu 3 aufbauen (Kante ($Q-::\text{ accept } Q-2$) von $A-3$ zu $A-2$). Dies kann danach beliebig oft geschehen (Schleife ($Q-::\text{ accept } Q-2$) um $A-2$). Wird die Verbindung zwischen 1 und 2 im Zustand $A-3$ nur einseitig von 1 beendet, so kann 2 im Zustand $A-2$ diese Verbindung auf Warteposition setzen und eine Verbindung zu 3 aufbauen (Kante ($Q-::\text{ accept } Q-3$) von $A-2$ zu $A-1$).

Eine erneute Verbindung zwischen 1 und 2 führt aus dem Zustand $A-2$ über die Kante ($Q-::\text{ accept } Q-1$) zu $A-3$ zurück.

Abschließend kann man sagen, daß die Verhaltensweisen, die bei der gewählten Konfiguration auftreten, den Erwartungen entsprechen.

4.4.8. Globales Verhalten bei abonniertem Call Waiting-Dienst

Aufgrund der großen Komplexität muß auf die Analyse des Call Waiting-Dienstes mit drei Teilnehmern verzichtet werden. Die Untersuchung wird wie im dritten Modell mit einem Anrufer ohne Dienst und einem Angerufenen mit dem Call Waiting-Dienst durchgeführt.

Der Untersuchung liegt die gleiche Erreichbarkeitsanalyse wie bei einem Angerufenen mit abonniertem Call Waiting-Dienst (vgl. Abschnitt 4.4.4.2.) zugrunde:

Der Homomorphismus ist folgendermaßen definiert:

Block	Transitionen
toggle	CTOG_0
Epsilon	alle restlichen Transitionen

Die Stelle On_hold sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Mit diesem Homomorphismus wird geprüft, zwischen welchen Verbindungen der Abonent mittels Call Toggle hin- und herschalten kann.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsmcs2c-gls1

2 States computed.
(3 Edges, 2 Labels)

Q-Markings : bcsmcs2c-gls1

Q-1
ON_HOLD: 1<1,2,2,1,::>

Q-2
ON_HOLD: 1<1,2,2,0,::>

End Q-Markings : bcsmcs2c-gls1
```

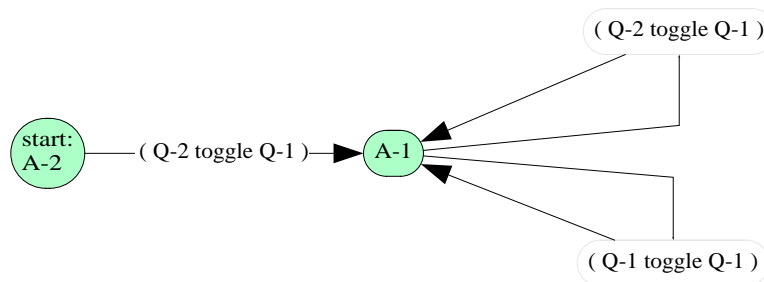


Abb. 53: Minimalautomat "Call Waiting"

Der Minimalautomat in Abb. 53 zeigt, daß der Angerufene zuerst die wartende Verbindung aktivieren muß, erkennbar an der 0 in der vierten Komponente der Marke in $Q-2$. Danach kann dieser Zustand einer aktiven und einer durch Call Waiting wartenden Verbindung beim Angerufenen beliebig oft wieder erreicht werden, wodurch auch wieder das Wechseln zwischen $Q-2$ und $Q-1$ möglich ist (Schleife um $A-1$). Außerdem kann der Angerufene nach dem ersten Wechseln immer wieder zwischen denselben Verbindungen hin- und herschalten, repräsentiert durch die Schleife ($Q-1$ toggle $Q-1$) um $A-1$.

4.4.9. Verhalten bei abonniertem Call Hold- und Call Waiting-Dienst

Analog zum dritten Modell wird hier das Zusammenspiel des Call Hold-Dienstes, abonniert vom Anrufer, und des Call Waiting-Dienstes, abonniert vom Angerufenen, untersucht. Grundlage der Untersuchung ist folgende Erreichbarkeitsanalyse mit je einem Anrufer und Angerufenen in der Anfangsmarkierung:

New Initial Marking:

```
ENTRY_PIC_1: 1<1>
IDLE: 1<1> + 1<2>
NUMBER_POOL: 1<2>
SERVICE_STRUCTURE: 1<1,CH,0.0,CH,DP_9,DP_8,ENTRY_DP_1B> +
    1<1,CH,0.0,CRET,::,DP_9,ENTRY_PIC_1> +
    1<2,CW,0.0,CRET,::,DP_17,ENTRY_PIC_5> +
    1<2,CW,0.0,CTOG,::,DP_8,ENTRY_PIC_5> +
    1<2,CW,0.0,CW,::,DP_5,ENTRY_PIC_5>
STATIC_ARMING: 1<1,DP_8> + 1<2,DP_17> + 1<2,DP_5> + 1<2,DP_8>
```

Reachability Graph bcsms2c-gls1

```
16011 Markings computed.
(0 Dead Markings)
(0 Pseudo Occurrence Steps)
(50750 Occurrence Steps)
```

```
Connected Components of Reachability Graph bcsms2c-gls1
(1 Component found)
(1 Leaf Component)
```

Der Homomorphismus ist folgendermaßen definiert:

Block	Prädikat	Transitionen
accept		DP_7_NOT_ARMED
Call_toggle	(On_hold=<1,2,2,0,::>,)	CTOG_0
Call_retrieve	(On_hold=<1,2,2,0,::>,)	CRET_0_1, CRET_0_2
Epsilon		alle restlichen Transitionen

Für die Blöcke *Call_toggle* und *Call_retrieve* werden Bedingungen angegeben, um das betrachtete Verhalten auf die Fälle einzuschränken, in denen der Angerufene eine durch das Feature Call Waiting auf Warteposition gesetzte Verbindung aktiviert. Das anschließende Umschalten zwischen den Verbindungen durch das Feature Call Toggle bzw. das abschließende Aktivieren einer noch wartenden Verbindung durch Call Retrieve werden dadurch nicht betrachtet.

Die Stellen *Entry_PIC_5* und *On_hold* sowie alle drei Komponenten bei den Zustandsübergängen im Minimalautomaten werden eingeblendet.

Die Berechnung des Minimalautomaten liefert folgendes Ergebnis:

```
Minimal Automaton bcsms2c-gls1
```

```
2 States computed.
(5 Edges, 4 Labels)
```

```
Q-Markings : bcsms2c-gls1
```

```
Q-::
```

Q-1
 ON_HOLD: 1<1,2,2,1,::>

Q-2
 ON_HOLD: 1<1,2,2,0,::>

Q-3
 ENTRY_PIC_5: 1<1,2,0,0,DP_9,::>
 ON_HOLD: 1<1,2,1,1,::>

Q-4
 ON_HOLD: 1<1,2,1,1,::>

Q-5
 ENTRY_PIC_5: 1<1,2,0,0,::,::>

End Q-Markings : bcsms2c-gls1

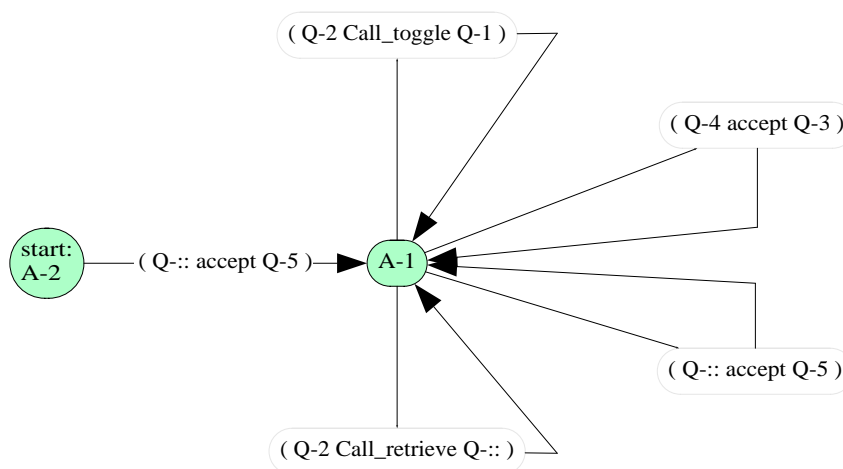


Abb. 54: Minimalautomat "Call Hold-/Call Waiting-Dienst"

Der Minimalautomat in Abb. 54 zeigt, daß zuerst eine Verbindung zwischen 1 und 2 zustandekommen muß (Kante (Q-:: accept Q-5) von A-2 zu A-1). Danach kann die gleiche Verbindung beliebig oft aufgebaut werden (Schleife mit gleicher Anschrift um A-1).

Außerdem kann der Anrufer die Verbindung auf Warteposition setzen, der Angerufene beendet die Verbindung, wird vom Anrufer erneut angerufen und akzeptiert die Verbindung (Schleife (Q-4 accept Q-3) um A-1).

Neben dem Ausführen des Call Hold-Dienstes durch den Anrufer kann es dazu kommen, daß auf der Seite des Angerufenen der Call Waiting-Dienst ausgeführt wird. Dazu setzt der Anrufer die Verbindung auf Warteposition oder beendet sie und ruft wieder an, während der Angerufene die erste Verbindung nicht beendet. Dieser ist dann für den neuen Aufbauwunsch besetzt, weshalb danach der Call Waiting-Dienst, beginnend mit dem Feature Call Waiting, ausgeführt wird. Der Angerufene kann anschließend das Service Feature Call Toggle benutzen, um die neue Verbindung zu aktivieren und die alte auf Warteposition zu setzen (Schleife (Q-2 Call_toggle Q-1) um A-1). Benutzt der Angerufene dieses Feature nicht, so wird er beim

Beenden der ersten Verbindung gezwungen, die wartende Verbindung zu aktivieren (Schleife ($Q-2$ *Call_retrieve* $Q-::$) um $A-1$).

Unter den gemachten Einschränkungen bei der Definition des Homomorphismus durch die Prädikate entspricht das gefundene Verhalten dem erwarteten und dem im unsynchronisierten, dritten Modell gefundenen Verhalten.

Zusammenfassung und Ausblick

Wie einige Analysen in den unsynchronisierten Modellen im Vergleich zu den äquivalenten Analysen im ersten, synchronisierten Modell gezeigt haben, verhindert eine zu starke Synchronisation der Teilnehmer realistische Verhaltensweisen sowohl im BCSM als auch bei der Ausführung von Diensten.

Es hat sich ebenfalls herausgestellt, daß die Modelle unter verteilter Sichtweise - ausgestattet mit dem notwendigen End-zu-End-Schlupf - für die Untersuchung von Diensten bei realistischen Konfigurationen bezüglich Anrufern und Angerufenen zu komplex sind. Die Zahl der Zustände erreicht schnell den sechsstelligen Bereich, wodurch die Berechnung der Zustände selbst sowie die Berechnung der durch Homomorphismen erzeugten vergrößernden Sichtweisen mehrere Stunden bis Tage dauern.

Das vorgestellte weniger granulare Modell unter globaler Sichtweise ist ein Schritt in die richtige Richtung. Es enthält einerseits den Schlupf der unsynchronisierten Modelle unter verteilter Sichtweise, andererseits ist die Komplexität wesentlich geringer.

Folgende Tabelle stellt für einige Konfigurationen die Anzahl der in den einzelnen Modellen berechneten Markierungen gegenüber.

<i>user</i>	1	1,3	1,2	1	1	1	1	1,3
<i>number</i>	2	2	1,2	2,3	2,3	2	2	2
<i># 'system_ready' für user²</i>	2	2	2	2		2	2	
<i># 'system_ready' für number³</i>	2	2	2	2		2	2	
<i>CH</i>					2	1		
<i>CW</i>						2	2	2
globales Modell	145	6431	1818	1499	30882	16011	1289	116474
1. Modell	2029	139962						
2. Modell	29841							
3. Modell	3082	277345	81792	38824		117079	18604	

Einige Analyseergebnisse beim globalen Modell zeigen jedoch, daß auch hier die Anzahl der Zustände bei komplexeren Konfigurationen schnell anwächst. Es ist daher für die Einbindung von weiteren Diensten und die Untersuchung von anderen Dienstkombinationen notwendig, den Zustandsraum durch geeignete Maßnahmen weiter einzuschränken. Dies könnte

² Gilt nur für die Modelle 1 bis 3.

³ Gilt nur für die Modelle 1 bis 3.

beispielsweise durch weitere Abstraktion oder Vermeidung von symmetrischen Fällen geschehen.

Um das globale Modell des BCSMs als Basis für die Untersuchung von Service Interaction im Verlauf der weiteren Standardisierung benutzen zu können, muß es seine Tauglichkeit bei der Anbindung weiterer CS-2-Dienste beweisen. Es könnte dann auch als Vorgabe für die Hersteller von Telekommunikationseinrichtungen dienen.

Literaturverzeichnis

- [ALP85] Alpern Bowen and Schneider Fred: *Defining Liveness*, Information Processing Letters 1985, pp. 182-185
- [CS-1] Ralph Demant, Farhad Fatahi-Vanani, Rafael Galvez-Estrada, Ulrich Nitsche, Peter Ochsenschläger: *Abschlußbericht 1995 zum Forschungsauftrag „Formale Spezifikations- und Verifikationsmethoden zur Behandlung der Service-Interaction-Problematik“*, Darmstadt 1995
- [Di95] Heinz Dibold: *Intelligente Netze*, Vorabdruck aus Bergmann - IN, 1995
- [NO95] U. Nitsche, P. Ochsenschläger: *Approximately Satisfied Properties of Systems and Simple Language Homomorphisms*, Arbeitspapiere der GMD 965, December 1995; erscheint in *Information Processing Letters*
- [Oc1] P. Ochsenschläger: *Projektionen und reduzierte Erreichbarkeitsgraphen*, Arbeitspapiere der GMD 349, 1988
- [Oc2] P. Ochsenschläger: *Modulhomomorphismen*, Arbeitspapiere der GMD 494, 1990
- [Oc3] P. Ochsenschläger: *Die Produktnetzmaschine*, Petri Net Newsletter 39, 1991
- [Oc4] P. Ochsenschläger: *Verifikation kooperierender Systeme mittels schlichter Homomorphismen*, Arbeitspapiere der GMD 688, 1992
- [Oc5] P. Ochsenschläger: *Verifikation verteilter Systeme mit Produktnetzen*, PIK 16 (1993), 42-43
- [Oc6] P. Ochsenschläger: *Verifikation of Cooperating Systems by Simple Homomorphisms Using the Product Net Machine*, Workshop "Algorithmen und Werkzeuge für Petrinetze", Berlin 1994
- [Oc7] P. Ochsenschläger: *Kompositionelle Verifikation kooperierender Systeme*, Arbeitspapiere der GMD 885, 1994
- [OP] P. Ochsenschläger, R. Prinoth: *Modellierung verteilter Systeme - Konzeption, formale Spezifikation und Verifikation mit Produktnetzen*, Vieweg Verlag 1995
- [Q.1214] ITU-T Recommendation Q.1214: *Distributed Functional Plane for Intelligent Network CS-1*
- [Q.1221] ITU-T: *Annex B (to Recommendation Q.1221), CS-2 Benchmark Services AND Features*
- [Q.1224] ITU-T: *Draft Q.1224, new sections 4 to 4.2.3*

Abkürzungsverzeichnis

BCSM	Basic Call State Modell
CCAF	Call Control Agent Function
CCF	Call Control Function
CH	Call Hold
CRET	Call Retrieve
CS	Capability Set
CTOG	Call Toggle
CW	Call Waiting
DFP	Distributed Functional Plane
DP	Detection Point
GFP	Global Functional Plane
IN	Intelligentes Netz
INCM	Intelligent Net Conceptual Model
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
O-BCSM	Originating Basic Call State Model
PIC	Point in Call
PP	Physical Plane
SCF	Service Control Function
SDF	Service Data Function
SMAF	Service Management Access Function
SMF	Service Management Function
SP	Service Plane
SRF	Service Resource Function
SSF	Service Switching Function
T-BCSM	Terminating Basic Call State Model