

Elektronische Fernleihe auf der Grundlage des ILL-Protokolls

Abschlußbericht zu dem von der DFG geförderten
Projekt
der Bibliothek der Universität Konstanz
und
der Universitätsbibliothek Ulm

Konstanz 1996

Bibliothek aktuell : Sonderheft 13
ISBN 3-89318-014-1

Bearbeitung dieses Berichts:

Cornelia Basler
Axel Glatz
Claudia Zirn

Auf den Abdruck des Anhangsteils dieser Diplomarbeit* mußte aus Gründen der Datensicherheit weitgehend verzichtet werden.

*Titel der Arbeit: „Realisierung von Diensten nach dem ILL-Protokoll für das Fernleihsystem der Universitätsbibliothek Konstanz“

Vorwort

Die vorliegende Dokumentation bildet den Abschlußbericht eines von der DFG finanzierten Projektes, das zum Ziel hatte, die Fernleihkommunikation zwischen zwei Bibliotheken auf der Grundlage der ISO-Standards ILL (Interlibrary Loan) und X.400 (elektronische Post) zu automatisieren. Ferner hatte das Projekt zum Ziel, die bislang auf Karteien gestützten Geschäftsgänge zu automatisieren (Fernleihverwaltungssystem) und an bereits vorhandene OPAC-Systeme sowie vorhandene Ausleihverbuchungssysteme anzuschließen.

Projektteilnehmer waren die Bibliothek der Universität Konstanz sowie die Universitätsbibliothek Ulm. Bei der ersten wurde das Projekt geführt, die zweite diente als Partner für die Tests. Daß die UB Ulm nicht in den vollen Genuß der Projektergebnisse kommen kann, ist dadurch bedingt, daß dort - entgegen den ursprünglichen Planungen - nach wie vor kein Ausleihverbuchungssystem existiert, in das man die Programme der Fernleihverwaltung hätte einbinden können. Daraus entstehen organisatorische Probleme, die im Rahmen des Projektes nicht gelöst werden können.

Das Projekt lief seit Ende 1991 mit zwei mehrmonatigen Unterbrechungen. Es war zunächst auf zwei Jahre Laufzeit angelegt; im Verlauf des Projektes stellte sich heraus, daß diese Zeitspanne zu kurz bemessen war, um die grundlegenden Probleme angemessen aufbereiten und lösen zu können. So mußte eine Verlängerung des Projektes um ein weiteres Jahr beantragt werden, was zu einer Unterbrechung führte, während deren Dauer der Projektbearbeiter in einem anderen Projekt mitarbeitete. Gegen Ende der Verlängerung verließ der Projektbearbeiter die Bibliothek, um sich beruflich anderweitig zu orientieren, zumal die Bibliothek ihm keine Aussicht auf eine längere Beschäftigung bieten konnte. Dieser Wechsel führte, nachdem die grundsätzlichen Probleme gelöst und sechs Dienste des ILL-Protokolls programmiert und teilweise dokumentiert waren, zu einer neuerlichen Unterbrechung von mehreren Monaten. Es gelang dann jedoch, drei Diplomanden der Fachhochschule Konstanz, Fachbereich Informatik, zu gewinnen, die als Gruppenarbeit innerhalb eines halben Jahres die restlichen Dienste programmierten und außerdem die vorliegende Dokumentation erstellten.

Bei aller Freude über die Fertigstellung von Software und Dokumentation sowie darüber, daß die Bibliothek der Universität Konstanz im Sommer 1996 ihre Fernleihe umstellen wird, soll nicht verhehlt werden, daß die Entwicklungen auf dem EDV-Sektor und in den bibliothekarischen Auffassungen zur Fernleihe in den vergangenen Jahren so fortgeschritten sind, daß nicht mehr alle Ansätze des Projektes aktuell sind und auch die Lösungen, die wir entwickelt haben, zumindest teilweise nicht mehr gelten dürften. So wurde nach mühsamer Entwicklung die elektronische Post auf der Basis X.400 aufgegeben zugunsten des TCP/IP-Standards. Durch die Diskussion und teilweise durch die Realisierung von Dokumentlieferdiensten wie JASON, DBI-LINK, DBV-OSI (Item Order) usw. ist die Fernleihe als bilaterale Beziehung zwischen zwei Bibliotheken ersetzt worden bzw. soll ersetzt werden durch eine bilaterale Beziehung zwischen Endnutzer und Lieferbibliothek. Das bedeutet, daß etwa bei der Hälfte aller

Fernleihbestellungen, nämlich denen, die sich auf Aufsätze beziehen, die nehmende Bibliothek ihre bisherige Bedeutung verlieren kann.

Wenn man darüber hinaus auch die Bestellung und Lieferung von Monographien auf eine Beziehung zwischen Endnutzer und Lieferbibliothek (z.B. im Rahmen von SUBITO oder DBV-OSI) umstellen will, so braucht man keine standardisierte Kommunikation zwischen den Fernleihsystemen von nehmenden und gebenden Bibliotheken, weil die nehmende Funktion nicht mehr benötigt wird. Sollte sich hingegen herausstellen, daß die gebenden Bibliotheken beim Versand von Monographien weiterhin auf der Belieferung einer nehmenden Bibliothek als Garantin für die Sicherheit der Bücher bestehen, so können die Projektergebnisse zum Zuge kommen, auch dann, wenn ein bestimmtes Softwareprodukt überholt ist.

Die Firmen, die bibliothekarische Software entwickeln, sind die eigentlichen Adressaten unserer Projektergebnisse. Es scheint allerdings, daß sie zur Zeit andere Prioritäten haben als die Automatisierung der Fernleihe. Ihre Prioritäten gehen nach wie vor in Richtung OPAC, Katalogisierung und vor allem Erwerbung.

Schließlich noch eine Erfahrung aus dem Projekt zum Abschluß: nicht alles, was auf dem Markt unter der Bezeichnung „ILL“ angeboten wird, ist auch ILL im Sinne des ISO-OSI-Standards. Mitunter werden selbstentwickelte, nicht standardisierte Fernleihsysteme unter dem Namen „ILL“ angeboten und propagiert. Sie sind keineswegs offen, sondern mit „ILL“ ist schlicht „automatisierte Fernleihe“ gemeint.

Mein Dank gilt denjenigen Mitarbeiterinnen und Mitarbeitern, die dieses Projekt betreut und - soweit es sich um Angehörige unserer Bibliothek handelt - zusätzlich zu ihren eigentlichen Aufgaben die konkrete Arbeit geleistet haben: Erich Friede (ILL und X.400), Cornelia Basler (ILL und Dokumentation), Axel Glatz (ILL und Dokumentation), Claudia Zirn (ILL und Dokumentation), Heinz-Gert ten Brink (Fernleihdatenbank, Fernleihverwaltungssystem, Ausleihsystem; Betreuer der Diplomarbeit), Josef Benz (Ausleihsystem und KOALA-OPAC), Günther Rau (Ausleihsystem, Fernleihe und KOALA-OPAC), Christa Kuon (Fernleihe), Wilfried Lehmler (Fernleihe und Fernleihpolitik), Marlene Nagelsmeier-Linke (Projektleitung bis Frühjahr 1995). Der Deutschen Forschungsgemeinschaft gebührt ein besonderer Dank für die finanzielle Förderung des Projektes bis zum Frühjahr 1995.

Klaus Franken

Zusammenfassung

Die Diplomarbeit hat zum Inhalt, aufbauend auf bereits geleisteten Vorarbeiten, weitere Dienste des OSI-Protokolls ILL (Inter Library Loan) zu programmieren und in die vorhandenen und vorgegebenen Datenstrukturen einzupassen.

Das ILL-Protokoll hat die Aufgabe, die Daten, die im Zusammenhang mit der bibliothekarischen Verwaltung von Fernleihbestellungen anfallen, in eine normgerechte Form umzusetzen, so daß diese auch zwischen Bibliotheken mit verschiedenen Fernleihsystemen (Hard- und Software) ausgetauscht werden können.

Im Rahmen des bisherigen Fernleihprojekts an der Bibliothek der Universität Konstanz wurde eine Fernleihverwaltungsdatenbank entwickelt. Es werden Datenbanksysteme unterstützt, die eine SQL-Schnittstelle bieten. Die Software zur Bearbeitung von Fernleihbestellungen wurde erstellt, die sechs grundlegenden Dienste des ILL-Protokolls wurden programmiert und schließlich wurde ein kommerzielles Mailsystem auf der Basis von X.400 und X.25 installiert. Die Kommunikation mit anderen Systemen kann aber auch wahlweise über TCP/IP erfolgen.

Um zu einer normgerechten Implementierung zu kommen, sollten weitere neun Dienste programmiert, in die Fernleihdatenbank und die Fernleihsoftware eingebunden, sowie in die Bedieneroberfläche integriert werden. Die zu implementierenden Dienste sollten in ihren Datenstrukturen dem funktionalen Standard der IFOBS¹ entsprechen und als Minimum, die nach dem Profil als „obligatorisch“ gekennzeichneten Elemente enthalten.

Die neun zu implementierenden Dienste lauten:

- MESSAGE Austausch von formlosen Nachrichten
- RENEW Antrag auf Leihfristverlängerung
- RENEW-ANSWER Antwort auf Verlängerungsantrag
- OVERDUE Hinweis auf Fristablauf
- RECALL Vorzeitiger Rückruf
- LOST Verlustanzeige
- DAMAGED Beschädigungsanzeige
- CANCEL Stornierung einer Bestellung
- CANCEL-REPLY Antwort auf Stornierung

Gegenstand der Arbeit sollten auch Funktionstests und die Dokumentation des gesamten Fernleihsystems der Universität Konstanz sein.

¹ Die IFOBS (International Forum on Open Bibliographic Systems) hat die Aufsicht über die Entwicklung internationaler Standardprofile für bibliographische OSI-Protokolle.

Inhaltsverzeichnis

	Einleitung.....	1
Kapitel 1	Die manuelle Fernleihe.....	3
	1 Die manuelle Fernleihe	4
	1.1 Begriffe im Fernleihverkehr	4
	1.2 Entwicklung der manuellen Fernleihe	6
Kapitel 2	Die elektronische Fernleihe	9
	2 Entwicklung der elektronischen Fernleihe.....	10
	2.1 Planung der elektronischen Fernleihe	10
	2.2 Ablauf der elektronischen Fernleihe	11
	3 Das Inter-Library-Loan (ILL) - Protokoll	12
	3.1 Entstehung und Funktionalität.....	12
	3.2 Normative Verweise.....	13
	3.3 Rahmenbedingungen und Anwendungsbereich	13
	3.4 Dienstmodell	14
	3.4.1 Dienstbenutzer und Dienstanbieter	14
	3.4.2 ILL-Transaktion, ILL-Subtransaktion und ILL-Transaktionsgruppe	15
	3.4.3 ILL-Transaktionstypen und Topologien	16
	3.4.3.1 Die einfache ILL-Transaktion (Simple).....	17
	3.4.3.2 Die verkettete ILL-Transaktion (Chaining)	17
	3.4.3.3 Die partitionierte ILL-Transaktion (Partitioning)	19
	3.4.3.4 Mischformen von Verkettung und Partitionierung.....	21
	3.4.3.5 Getrennte ILL-Transaktionen	22
	3.4.3.6 Weiterleitung (Forwarding).....	23
	3.4.3.7 Einfache Sterntopologie (Referrals).....	26
	3.4.3.8 Wiederholungen (Retries)	26
	3.4.4 Globale ILL-Transaktionszustände.....	27
	3.4.4.1 Globale ILL-Transaktionszustände des Requesters	28
	3.4.4.2 Globale ILL-Transaktionszustände des Responders	29
	3.4.4.3 Terminierende globale ILL-Transaktionszustände.....	30
	3.4.4.4 Globale ILL-Zustände des Intermediary.....	31
	3.4.4.5 Definition von Zustandsübergängen der globalen ILL-Transaktionszustände	32
	3.4.4.6 Phasen einer ILL-Transaktion	34

3.5	Definition der Dienste des ILL-Protokolls.....	34
3.5.1	Allgemeines.....	34
3.5.2	Fernleihanfrage (ILL-REQUEST).....	36
3.5.3	Weiterleitung einer Fernleihanfrage (FORWARD)	43
3.5.4	Weiterleitungsanzeige (FORWARD-NOTIFICATION).....	44
3.5.5	Versand (SHIPPED).....	44
3.5.6	ILL-Antwort (ILL-ANSWER)	44
3.5.7	Bedingte Antwort (CONDITIONAL REPLY)	45
3.5.8	Stornierung (CANCEL).....	45
3.5.9	Antwort auf eine Stornierung (CANCEL-REPLY)	45
3.5.10	Empfang (RECEIVED)	45
3.5.11	Vorzeitiger Rückruf (RECALL).....	46
3.5.12	Rückgabe (RETURNED).....	46
3.5.13	Eingang (CHECKED-IN)	46
3.5.14	Ablauf der Ausleihfrist (OVERDUE).....	46
3.5.15	Antrag auf Leihfristverlängerung (RENEW).....	47
3.5.16	Antwort auf Leihfristverlängerungsantrag (RENEW-ANSWER)	47
3.5.17	Verlustanzeige (LOST).....	47
3.5.18	Beschädigungsanzeige (DAMAGED).....	47
3.5.19	Formlose Nachricht (MESSAGE)	47
3.5.20	Anfrage zum Zustand einer ILL-Transaktion (STATUS-QUERY).....	48
3.5.21	Zustands- oder Fehlerbericht (STATUS-OR-ERROR-REPORT).....	48
3.5.22	Ablauf der ILL-Transaktion (Expiry).....	48
3.6	ILL-Dienstaufruf-Sequenzen	48
4	Die abstrakte Syntaxnotation 1 (ASN.1).....	49
4.1	Die Beschreibung von Datenstrukturen	49
4.2	Das Grundprinzip von ASN.1	50
4.3	Die abstrakte Syntax der ASN.1-Notation	52
4.4	Die Transfersyntax von ASN.1	56
4.5	Der ASN.1-Compiler der Universität Erlangen	58
4.6	ASN.1 im Fernleihsystem der Universität Konstanz	58
Kapitel 3	Realisierung der elektronischen Fernleihe an der Universität Konstanz.....	59
5	Unterstützte Kommunikationssysteme	60
5.1	Message Handling	60
5.1.1	Das funktionale Modell	61
5.1.2	Der Nachrichtentransferdienst (MTS-Dienst).....	62
5.1.3	Adressierung unter MOTIS	65

5.2	TCP/IP	66
5.2.1	Vergleich der Protokolle mit dem ISO/OSI Standard	66
5.2.2	Internet Protocol.....	68
5.2.3	Transmission Control Protocol.....	71
5.2.3.1	Funktionsweise.....	71
5.2.3.2	TCP-Verbindungsaufbau.....	73
5.2.3.3	TCP-Datenaustausch.....	74
5.2.3.4	TCP-Verbindungsabbau.....	75
5.2.4	Sockets.....	76
6	Client-Server-Architektur.....	81
6.1	Die Server-Programme.....	81
6.1.1	Sendeserver.....	85
6.1.2	Empfangsserver.....	87
6.1.3	Verwaltungsserver.....	89
6.1.4	Dialogserver.....	91
6.1.5	Die Dialogkommandos	94
6.1.5.1	Die Namenskonventionen der Dialogkommandos.....	94
6.1.5.2	Dialogkommandos zum Erfassen eines Vorgangs	95
6.1.5.3	Dialogkommandos zum Ausführen/Aktivieren eines Vorgangs.....	96
6.1.5.4	Dialogkommandos zum Zeigen eines Vorgangs	97
6.1.5.5	Dialogkommandos zum Korrigieren/Updaten eines Vorgangs.....	98
6.2	Der Fernleih-Client.....	100
6.2.1	Die ADIX-Dialoglaufzeit.....	100
6.2.1.1	ADIX-Prozeduren.....	100
6.2.1.2	Client-Server-Dialogerweiterung für ADIX	101
6.2.1.3	Aufbau der BILD-Dateien.....	101
6.2.2	Dialog-Client.....	108
7	Prinzip der Kodierung und Dekodierung der ILL-Daten	113
7.1	Allgemeines Vorgehen.....	113
7.2	Die Kodierung.....	114
7.3	Die Dekodierung	116
8	Die Fernleihdatenbank	119
8.1	Die Tabellen der Fernleihdatenbank.....	119
8.2	Zugriff auf die Tabellen der Fernleihdatenbank.....	121
8.3	Bereitstellung von Daten aus den Tabellen der Fernleihdatenbank	123
9	Lokale Zustände im Fernleihsystem.....	126
9.1	Lokale Zustände der nehmenden Fernleihe.....	128
9.2	Lokale Zustände der gebenden Fernleihe.....	131
9.3	Die Zustandsübergänge	133

Inhaltsverzeichnis

10	Anbindung wichtiger Bibliotheksdienste	136
10.1	Fernleih-Bestelleingabe über WWW-Client	136
10.1.1	Fernleihbestellung einer Monographie	137
10.1.2	Fernleihbestellung eines Zeitschriften-Aufsatzes	138
10.1.3	Expres-Bestellung eines Zeitschriften-Aufsatzes	138
10.1.4	Client-Server-Prinzip bei der Fernleihbestellung im WWW	139
10.2	Fernleih-Bestelleingabe über OPAC-Terminals	140
10.3	Schnittstellen zum Fernleihsystem	142
10.4	Schnittstelle zwischen Fernleihsystem und Ausleihverbuchung	143
11	Bewertung des Fernleihsystems an der Universität Konstanz	145
11.1	Fernleih-Client	145
11.2	Fernleih-Server	145
Kapitel 4	Implementierung der ILL-Dienste	147
12	ILL-REQUEST	148
12.1	Senden und Empfangen des Dienstes ILL-REQUEST	148
12.1.1	Senden des Dienstes ILL-REQUEST durch den Requester	148
12.1.2	Empfangen des Dienstes ILL-REQUEST beim Responder	148
12.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes ILL-REQUEST	149
12.2.1	Zustandsübergänge beim Senden des Dienstes ILL-REQUEST durch den Requester	149
12.2.2	Zustandsübergänge beim Empfangen des Dienstes ILL-REQUEST beim Responder	149
12.3	Die globalen ILL-Transaktionszustände	150
12.4	Menüsteuerung beim Fernleih-Client	150
12.4.1	Nehmende Fernleihe (Requester)	150
12.4.2	Gebende Fernleihe (Responder)	151
13	SHIPPED	152
13.1	Senden und Empfangen des ILL-Dienstes SHIPPED	152
13.1.1	Senden des ILL-Dienstes SHIPPED durch den Responder	152
13.1.2	Empfangen des ILL-Dienstes SHIPPED beim Requester	152

13.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes SHIPPED.....	153
13.2.1	Zustandsübergänge beim Senden des ILL-Dienstes SHIPPED durch den Responder.....	153
13.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes SHIPPED beim Requester.....	154
13.3	Die globalen ILL-Transaktionszustände	155
14	ILL-ANSWER.....	156
14.1	Senden und Empfangen des ILL-Dienstes ILL-ANSWER.....	156
14.1.1	Senden des ILL-Dienstes ILL-ANSWER durch den Responder.....	157
14.1.2	Empfang des ILL-Dienstes ILL-ANSWER beim Requester	157
14.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes ILL-ANSWER.....	158
14.2.1	Zustandsübergänge beim Senden des Dienstes ILL-ANSWER durch den Responder.....	158
14.2.2	Zustandsübergänge beim Empfangen des Dienstes ILL-ANSWER beim Requester.....	159
14.3	Die globalen ILL-Transaktionszustände	160
14.4	Menüsteuerung beim Fernleih-Client.....	160
14.4.1	Gebende Fernleihe.....	160
14.4.2	Nehmende Fernleihe.....	161
15	CANCEL	162
15.1	Senden und Empfangen des ILL-Dienstes CANCEL.....	162
15.1.1	Senden des ILL-Dienstes CANCEL durch den Requester.....	162
15.1.2	Empfang des ILL-Dienstes CANCEL beim Responder	163
15.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes CANCEL	163
15.2.1	Zustandsübergänge beim Senden des Dienstes CANCEL durch den Requester	164
15.2.2	Zustandsübergänge bei Empfang des Dienstes CANCEL beim Responder.....	165
15.3	Die globalen ILL-Transaktionszustände	166
15.4	Menüsteuerung beim Fernleih-Client.....	167
15.4.1	Nehmende Fernleihe.....	167
15.4.2	Gebende Fernleihe.....	168
16	CANCEL-REPLY	169
16.1	Senden und Empfangen des ILL-Dienstes CANCEL-REPLY	169
16.1.1	Senden des ILL-Dienstes CANCEL-REPLY durch den Responder.....	169
16.1.2	Empfang des ILL-Dienstes CANCEL-REPLY beim Requester.....	170

Inhaltsverzeichnis

16.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes CANCEL-REPLY	170
16.2.1	Zustandsübergänge beim Senden des Dienstes CANCEL-REPLY durch den Responder	170
16.2.2	Zustandsübergänge bei Empfang des Dienstes CANCEL-REPLY beim Requester	172
16.3	Die globalen ILL-Transaktionszustände	173
16.4	Menüsteuerung beim Fernleih-Client	174
16.4.1	Nehmende Fernleihe	174
16.4.2	Gebende Fernleihe	175
17	RECEIVED	176
17.1	Senden und Empfangen des ILL-Dienstes RECEIVED	176
17.1.1	Senden des ILL-Dienstes RECEIVED durch den Requester	176
17.1.2	Empfangen des ILL-Dienstes RECEIVED durch den Responder	177
17.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes RECEIVED	177
17.2.1	Zustandsübergänge beim Senden des Dienstes RECEIVED durch den Requester	177
17.2.2	Zustandsübergänge bei Empfang des Dienstes RECEIVED beim Responder	179
17.3	Die globalen ILL-Transaktionszustände	179
18	RECALL	181
18.1	Senden und Empfangen des ILL-Dienstes RECALL	181
18.1.1	Senden des ILL-Dienstes RECALL durch den Responder	181
18.1.2	Empfangen des ILL-Dienstes RECALL beim Requester	182
18.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RECALL	182
18.2.1	Zustandsübergänge beim Senden des ILL-Dienstes RECALL durch den Responder	182
18.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes RECALL beim Requester	183
18.3	Die globalen ILL-Transaktionszustände	184
18.4	Menüsteuerung beim Fernleih-Client	185
18.4.1	Nehmende Fernleihe (Requester)	185
18.4.2	Gebende Fernleihe (Responder)	186
19	RETURNED	187
19.1	Senden und Empfangen des ILL-Dienstes RETURNED	187
19.1.1	Senden des ILL-Dienstes RETURNED durch den Requester	187
19.1.2	Empfangen des ILL-Dienstes RETURNED beim Responder	188

19.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RETURNED	188
19.2.1	Zustandsübergänge beim Senden des ILL-Dienstes RETURNED durch den Requester	188
19.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes RETURNED beim Responder	189
19.3	Die globalen ILL-Transaktionszustände	190
20	CHECKED-IN	191
20.1	Senden und Empfangen des ILL-Dienstes CHECKED-IN	191
20.1.1	Senden des ILL-Dienstes CHECKED-IN durch den Responder	191
20.1.2	Empfangen des ILL-Dienstes CHECKED-IN beim Requester	192
20.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes CHECKED-IN	192
20.2.1	Zustandsübergänge beim Senden des ILL-Dienstes CHECKED-IN durch den Responder	192
20.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes CHECKED-IN beim Requester	193
20.3	Die globalen ILL-Transaktionszustände	193
21	OVERDUE	194
21.1	Senden und Empfangen des ILL-Dienstes OVERDUE	194
21.1.1	Senden des ILL-Dienstes OVERDUE durch den Responder	194
21.1.2	Empfangen des ILL-Dienstes OVERDUE beim Requester	195
21.2	Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes OVERDUE	195
21.2.1	Zustandsübergänge beim Senden des Dienstes OVERDUE durch den Responder	196
21.2.2	Zustandsübergänge beim Empfangen des Dienstes OVERDUE beim Requester	196
21.3	Die globalen ILL-Transaktionszustände	198
21.4	Menüsteuerung beim Fernleih-Client	198
21.4.1	Gebende Fernleihe	198
21.4.2	Nehmende Fernleihe	199
22	RENEW	200
22.1	Senden und Empfangen des ILL-Dienstes RENEW	200
22.1.1	Senden des ILL-Dienstes RENEW durch den Requester	201
22.1.2	Empfangen des ILL-Dienstes RENEW beim Responder	201

Inhaltsverzeichnis

22.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RENEW	201
22.2.1	Zustandsübergänge beim Senden des ILL-Dienstes RENEW durch den Requester	201
22.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes RENEW beim Responder.....	202
22.3	Die globalen ILL-Transaktionszustände	203
22.4	Menüsteuerung beim Fernleih-Client.....	203
22.4.1	Nehmende Fernleihe (Requester).....	204
22.4.2	Gebende Fernleihe (Responder).....	205
23	RENEW-ANSWER.....	206
23.1	Senden und Empfangen des ILL-Dienstes RENEW-ANSWER	206
23.1.1	Senden des ILL-Dienstes RENEW-ANSWER durch den Responder.....	206
23.1.2	Empfangen des ILL-Dienstes RENEW-ANSWER durch den Requester.....	206
23.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RENEW-ANSWER.....	207
23.2.1	Zustandsübergänge beim Senden des ILL-Dienstes RENEW-ANSWER durch den Responder.....	207
23.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes RENEW-ANSWER beim Requester.....	208
23.3	Die globalen ILL-Transaktionszustände	208
23.4	Menüsteuerung beim Fernleih-Client.....	209
23.4.1	Nehmende Fernleihe (Requester).....	209
23.4.2	Gebende Fernleihe (Responder).....	210
24	LOST	211
24.1	Senden und Empfangen des ILL-Dienstes LOST	211
24.1.1	Senden des ILL-Dienstes LOST durch den Requester	211
24.1.2	Empfangen des ILL-Dienstes LOST beim Responder	212
24.1.3	Senden des ILL-Dienstes LOST durch den Responder.....	213
24.1.4	Empfangen des ILL-Dienstes LOST beim Requester	213
24.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes LOST	213
24.2.1	Zustandsübergänge beim Senden des ILL-Dienstes LOST durch den Requester	213
24.2.2	Zustandsübergänge beim Empfangen des Dienstes LOST beim Responder.....	215
24.2.3	Zustandsübergänge beim Senden des Dienstes LOST durch den Responder.....	215
24.2.4	Zustandsübergänge beim Empfangen des Dienstes LOST beim Requester.....	216

24.3	Die globalen ILL-Transaktionszustände	217
24.4	Menüsteuerung beim Fernleih-Client.....	218
24.4.1	Nehmende Fernleihe (Requester)	218
24.4.2	Gebende Fernleihe (Responder).....	219
25	DAMAGED.....	220
25.1	Senden und Empfangen des ILL-Dienstes DAMAGED	220
25.1.1	Senden des ILL-Dienstes DAMAGED durch den Requester.....	221
25.1.2	Empfangen des ILL-Dienstes DAMAGED beim Responder	221
25.1.3	Senden des ILL-Dienstes DAMAGED durch den Responder.....	222
25.1.4	Empfangen des ILL-Dienstes DAMAGED beim Requester	222
25.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes DAMAGED	222
25.2.1	Zustandsübergänge beim Senden des ILL-Dienstes DAMAGED durch den Requester	222
25.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes DAMAGED beim Responder.....	223
25.2.3	Zustandsübergänge beim Senden des ILL-Dienstes DAMAGED durch den Responder	224
25.2.4	Zustandsübergänge beim Empfangen des ILL-Dienstes DAMAGED beim Requester.....	224
25.3	Die globalen ILL-Transaktionszustände	225
25.4	Menüsteuerung beim Fernleih-Client.....	226
25.4.1	Nehmende Fernleihe (Requester)	226
25.4.2	Gebende Fernleihe (Responder).....	227
26	MESSAGE	228
26.1	Senden und Empfangen des ILL-Dienstes MESSAGE.....	228
26.1.1	Senden und Empfangen des ILL-Dienstes MESSAGE durch den Requester	229
26.1.2	Senden und Empfangen des ILL-Dienstes MESSAGE durch den Responder	229
26.2	Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes MESSAGE	229
26.2.1	Zustandsübergänge beim Senden des ILL-Dienstes MESSAGE durch den Requester	229
26.2.2	Zustandsübergänge beim Empfangen des ILL-Dienstes MESSAGE beim Responder.....	230
26.2.3	Zustandsübergänge beim Senden des ILL-Dienstes MESSAGE durch den Responder	231
26.2.4	Zustandsübergänge beim Empfangen des ILL-Dienstes MESSAGE beim Requester.....	232
26.3	Die globalen ILL-Transaktionszustände	232

Abbildungsverzeichnis

Kapitel 1	Manuelle Fernleihe	
	Abb. 1.1	Nehmendes Fernleihaufkommen6
	Abb. 1.2	Gebendes Fernleihaufkommen6
Kapitel 2	Elektronische Fernleihe	
	Abb. 3.1	Einfache ILL-Transaktion17
	Abb. 3.2	Verkettete ILL-Transaktion mit 2 Intermediaries18
	Abb. 3.3	Sterntopologie19
	Abb. 3.4	Partitionierte ILL-Transaktion20
	Abb. 3.5a	Partitionierung nach Verkettung21
	Abb. 3.5b	Verkettung nach Partitionierung22
	Abb. 3.6	Mehrfaches Weiterleiten23
	Abb. 3.7a	Weiterleitung nach Verkettung24
	Abb. 3.7b	Verkettung nach Weiterleitung24
	Abb. 3.8a	Weiterleitung nach Partitionierung25
	Abb. 3.8b	Partitionierung nach Weiterleitung25
	Abb. 3.9	Einfache Sterntopologie26
	Abb. 3.10	ILL-Zustandsübergänge des Requesters bei einem „returnable item“33
	Abb. 4.1	Übertragung einer APDU50
	Abb. 4.2	Abstrakte Syntax in C51
	Abb. 4.3	Kodierung von Typ, Länge und Wert52
	Abb. 4.4	Elementtypen von ASN.153
	Abb. 4.5	Bausteine von ASN.153
	Abb. 4.6	Abstrakte Syntax in C54
	Abb. 4.7	Abstrakte Syntax in ASN.155
	Abb. 4.8	Erstes Byte eines Datenfeldes57
Kapitel 3	Realisierung der elektronischen Fernleihe an der Universität Konstanz	
	Abb. 5.1	Modell eines MHS61
	Abb. 5.2	Die Struktur von X.40062
	Abb. 5.3	Weg und Struktur einer X.400 Nachricht63
	Abb. 5.4	Einige Umschlagfelder64
	Abb. 5.5	X.400 Protokollserie66
	Abb. 5.6	Vergleichende Zuordnung der TCP/IP Protokolle zum OSI-Schichtenmodell67
	Abb. 5.7	IP-Header68
	Abb. 5.8	Kommunikation zwischen Teilnetzen mit Gateway70
	Abb. 5.9	TCP Header72
	Abb. 5.10	Beispiel für einen Verbindungsaufbau73
	Abb. 5.11	Beispiel für einen Datenaustausch über TCP74
	Abb. 5.12	Beispiel für den getrennten Verbindungsabbau75
	Abb. 5.13	Dateiübertragung mit TCP/IP76
	Abb. 5.14	Beispiel für das Einrichten eines passiven Sockets77
	Abb. 5.15	Beispiel für das Warten auf eine Verbindungsanforderung über den Socket78
	Abb. 5.16	Beispiel für die Verbindungsaufnahme mit dem Systemcall connect79
	Abb. 5.17	Beispiel für das Schließen der Verbindung80
	Abb. 6.1	Grobkonzept Client-Server-Architektur83
	Abb. 6.2	Aufbau der Datei cfg.fem84

Abbildungsverzeichnis

Abb. 6.3	Pseudo-C-Code des Sendeservers	86
Abb. 6.4	Pseudo-C-Code des Empfangsservers	87
Abb. 6.5	Die Bearbeitung der eintreffenden ILL-Sendung	89
Abb. 6.6	Pseudo-C-Code des Verwaltungsservers	90
Abb. 6.7	Pseudo-C-Code des Dialogservers	92
Abb. 6.8	Bearbeitung der Kommandos, die vom Socket des Client ankommen	93
Abb. 6.9	Syntax eines Dialogkommandos	94
Abb. 6.10	Beispiel der Kommandos für die Bearbeitung eines Dienstes	95
Abb. 6.11	Die Ausführung des Kommandos „zumn“ stark vereinfacht	99
Abb. 6.12	ADIX und FCADIX Dialoglaufzeit	101
Abb. 6.13	Die Begriffe im Bild, Maske, Feld und Menü	102
Abb. 6.14	Syntax der Bildbeschreibung	103
Abb. 6.15	Syntax der Maskenbeschreibung	104
Abb. 6.16	Syntax der Feldbeschreibung	105
Abb. 6.17	BILD-Datei fchpm.pic	106
Abb. 6.18	Bildschirmanzeige der BILD-Datei fchpm.pic	107
Abb. 6.19	Erweiterte Feldbeschreibung	108
Abb. 7.1	Kodierung und Dekodierung	113
Abb. 7.2	Pseudo-C-Code von encode_apdu(..)	115
Abb. 7.3	Pseudo-C-Code von decode_apdu()	117
Abb. 7.4	Kodierung und Dekodierung im Fernleihsystem der Uni Konstanz	118
Abb. 8.1	Tabelle „partner_tab“ mit Satzpuffer „sp_part“	123
Abb. 8.2	Das Bereitstellen von Daten der Fernleihdatenbank	125
Abb. 9.1	Vergleich eines globalen Zustandswechsels mit lokalen Zustandsübergängen	127
Abb. 9.2	Legende zu den folgenden Zustandsgraphen	133
Abb. 9.3	Zustandsübergänge bei Erfassung einer Fernleihbestellung	133
Abb. 9.4	Zustandswechsel bei Bearbeitung eines Bestelleingangs auf der gebenden Seite	134
Abb. 9.5	Zustandswechsel nach Empfang des Leihobjekts bei der nehmenden Bibliothek	134
Abb. 9.6	Zustandsübergang bei Feststellung der Leihfristüberschreitung auf gebender Seite	135
Abb. 9.7	Zustandswechsel nach Empfang eines ILL-OVERDUE	135
Abb. 9.8	Ende der aktiven Vorgangsbearbeitung	135
Abb. 10.1	Auswahlmenü für die Eingabe von Fernleihbestellungen	136
Abb. 10.2	Fernleihbestellung für eine Monographie	137
Abb. 10.3	Fernleihbestellung eines Zeitschriften-Aufsatzes	138
Abb. 10.4	Expresß-Bestellung eines Zeitschriften-Aufsatzes	139
Abb. 10.5	Client-Server-Prinzip bei der Fernleihbestellung im WWW	140
Abb. 10.6	Verbindung zwischen OPAC-Terminals und Fernleihsystem	141
Abb. 10.7	Schnittstellen zum Fernleihsystem	142

Kapitel 4

Implementierung der ILL-Dienste

Abb. 12.1	Zustandsübergänge beim Senden des ILL-REQUEST (Requester)	149
Abb. 12.2	Zustandsübergänge beim Empfangen des ILL-REQUEST (Responder)	150
Abb. 12.3	Menü ILL-REQUEST (Requester)	151
Abb. 12.4	Menü Anfragen von der Partnerbibliothek (Responder)	151
Abb. 13.1	Zustandsübergänge beim Senden des Dienstes SHIPPED (Responder)	154
Abb. 13.2	Globale ILL-Transaktionszustände	155
Abb. 14.1	Lokale Zustände	157
Abb. 14.2	Zustandswechsel beim Senden der ILL-ANSWER	159
Abb. 14.3	Zustandswechsel bei Empfang einer negativen ILL-ANSWER	159
Abb. 14.4	Globale ILL-Transaktionszustände	160
Abb. 14.5	Menü ILL-ANSWER (Responder)	160

Abb. 14.6	Menü ILL-ANSWER (Requester)	161
Abb. 15.1	Lokale Zustände	162
Abb. 15.2	Zustandsgenerierung beim Senden des CANCEL (Requester)	164
Abb. 15.3	Zustandsgenerierung bei Empfang eines CANCEL (Responder)	165
Abb. 15.4	Globale ILL-Transaktionszustände	166
Abb. 15.5	Menü CANCEL (Requester)	167
Abb. 15.6	Menü CANCEL (Responder)	168
Abb. 16.1	Lokale Zustände	169
Abb. 16.2	Zustandsgenerierung beim Senden des positiven CANCEL-REPLY (Responder)	171
Abb. 16.3	Zustandsgenerierung beim Senden des negativen CANCEL-REPLY (Responder)	171
Abb. 16.4	Zustandsgenerierung bei Empfang eines positiven CANCEL-REPLY (Requester)	172
Abb. 16.5	Zustandsgenerierung bei Empfang eines negativen CANCEL-REPLY (Requester)	173
Abb. 16.6	Globale ILL-Transaktionszustände	173
Abb. 16.7	Menü CANCEL-REPLY (Requester)	174
Abb. 16.8	Menü CANCEL-REPLY (Responder)	175
Abb. 17.1	Lokale Zustände	176
Abb. 17.2	Zustandsgenerierung nach Erhalt eines Objekts zum Verbleib (Requester)	178
Abb. 17.3	Zustandsgenerierung nach Erhalt eines Objekts zur Ausleihe (Requester)	179
Abb. 18.1	Lokale Zustände	181
Abb. 18.2	Zustandsübergänge: Senden RECALL	183
Abb. 18.3	Zustandsübergänge: Empfangen RECALL	183
Abb. 18.4	Globale ILL-Transaktionszustände	184
Abb. 18.5	Menüsteuerung RECALL (Requester)	185
Abb. 18.6	Menüsteuerung RECALL (Responder)	186
Abb. 19.1	Lokale Zustände	187
Abb. 19.2	Zustandsübergänge: Senden RETURN	189
Abb. 19.3	Globale ILL-Transaktionszustände	190
Abb. 20.1	Lokale Zustände	191
Abb. 20.2	Zustandsübergänge: Senden CHECKED-IN	192
Abb. 20.3	Globale ILL-Transaktionszustände	193
Abb. 21.1	Lokale Zustände	194
Abb. 21.2	Zustandsgenerierung bei Senden eines OVERDUE (Responder)	196
Abb. 21.3	Zustandsgenerierung bei Empfang eines OVERDUE (Requester)	197
Abb. 21.4	Globale ILL-Transaktionszustände	198
Abb. 21.5	Menü OVERDUE (Responder)	198
Abb. 21.6	Menü OVERDUE (Requester)	199
Abb. 22.1	Lokale Zustände	200
Abb. 22.2	Zustandsübergänge: Senden RENEW	202
Abb. 22.3	Zustandsübergänge: Empfangen RENEW	202
Abb. 22.4	Globale ILL-Transaktionszustände	203
Abb. 22.5	Menüsteuerung RENEW (Requester)	204
Abb. 22.6	Menüsteuerung RENEW (Responder)	205
Abb. 23.1	Lokale Zustände	206
Abb. 23.2	Zustandsübergänge: Senden RENEW-ANSWER	207
Abb. 23.3	Zustandsübergänge: Empfangen RENEW-ANSWER	208
Abb. 23.4	Globale ILL-Transaktionszustände	209
Abb. 23.5	Menüsteuerung RENEW-ANSWER (Requester)	209
Abb. 23.6	Menüsteuerung RENEW-ANSWER (Responder)	210

Abbildungsverzeichnis

Abb. 24.1	Lokale Zustände.....	211
Abb. 24.2	Zustandsübergänge beim Senden des ILL-Dienstes LOST (Requester).....	214
Abb. 24.3	Zustandsübergänge beim Empfangen des ILL-Dienstes LOST (Responder).....	215
Abb. 24.4	Zustandsübergänge beim Senden des ILL-Dienstes LOST (Responder).....	216
Abb. 24.5	Zustandsübergänge beim Empfangen des ILL-Dienstes LOST (Requester).....	217
Abb. 24.6	Globale ILL-Transaktionszustände.....	217
Abb. 24.7	Menü LOST (Requester)	218
Abb. 24.8	Menü LOST (Responder).....	219
Abb. 25.1	Lokale Zustände.....	220
Abb. 25.2	Zustandsübergänge beim Senden des Dienstes DAMAGED (Requester).....	223
Abb. 25.3	Zustandsübergänge beim Empfangen des Dienstes DAMAGED (Responder)	224
Abb. 25.4	Zustandsübergänge beim Senden des Dienstes DAMAGED (Responder).....	224
Abb. 25.5	Zustandsübergänge beim Empfangen des Dienstes DAMAGED (Requester)	225
Abb. 25.6	Globale ILL-Transaktionszustände.....	225
Abb. 25.7	Menü DAMAGED (Requester)	226
Abb. 25.8	Menü DAMAGED (Responder)	227
Abb. 26.1	Lokale Zustände.....	228
Abb. 26.2	Zustandsübergänge beim Senden des Dienstes MESSAGE (Requester)	230
Abb. 26.3	Zustandsübergänge beim Empfangen des Dienstes MESSAGE (Responder).....	231
Abb. 26.4	Zustandsübergänge beim Senden des Dienstes MESSAGE (Responder)	231
Abb. 26.5	Zustandsübergänge beim Empfangen des Dienstes MESSAGE (Requester).....	232
Abb. 26.6	Globale ILL-Transaktionszustände.....	233
Abb. 26.7	Menü MESSAGE (Requester)	234
Abb. 26.8	Menü MESSAGE (Responder)	235
Abb. 27.1	Ablauf einer Fernleihtransaktion	236
Abb. 28.1	Projektplan.....	239

Einleitung

Die Bibliothek der Universität Konstanz hat zur Zeit einen Bestand von ca. 2 Millionen Büchern. Sollte ein Leser dennoch ein Buch wünschen, das an der Universität Konstanz nicht vorhanden ist, so kann versucht werden, das gewünschte Exemplar bei einer anderen Bibliothek auszuleihen. Dieser Vorgang wird als Fernleihe oder Leihverkehr bezeichnet.

Die im Zusammenhang mit der Fernleihe anfallenden Arbeiten werden in der Universitätsbibliothek Konstanz manuell mit Hilfe von Karteikarten und Fernleihscheinen durchgeführt. Da der Leihverkehr von den Lesern immer häufiger in Anspruch genommen wird, hat der Aufwand an Verwaltungsarbeiten in den letzten Jahren stark zugenommen.

Angesichts dieser Tatsachen hat die Bibliothek der Universität Konstanz ein Projekt beantragt, dessen Ziel es sein soll, Lösungen zu entwickeln und zu erproben, mit denen die Fernleihe durch Nutzung moderner Informations- und Kommunikationstechnologien beschleunigt werden kann. Das Projekt wurde im Mai 1991 von der Deutschen Forschungsgemeinschaft bewilligt.

Seit dieser Zeit wird an der Entwicklung der elektronischen Fernleihe für die Universität Konstanz gearbeitet. Zum Wintersemester 1996 soll das Fernleihsystem stufenweise eingeführt werden.

In der hier vorliegenden Dokumentation wird das Fernleihsystem als Gesamtsystem vorgestellt. Da die Ergebnisse der Arbeit aus unterschiedlichen Gesichtspunkten verschiedenen Personenkreisen zugänglich gemacht werden sollen, wurde versucht, ausführlicher auf die einzelnen Gebiete einzugehen.

Im ersten Kapitel wird eine Einführung in die bibliothekarischen Begriffe des standardisierten Leihverkehrs gegeben. In Kapitel 2 werden die Grundlagen für die elektronische Fernleihe erläutert. Dazu gehören insbesondere die Standards für den Datenaustausch zwischen Bibliotheken. In Kapitel 3 wird ausführlich auf die Realisierung des Fernleihsystems an der Universitätsbibliothek Konstanz eingegangen. Es werden die unterstützten Kommunikationssysteme, die Architektur der Software und des benötigten Datenbanksystems vorgestellt. Außerdem werden speziell auf die Bedürfnisse der Universitätsbibliothek abgestimmte Besonderheiten erläutert. Im vierten Kapitel werden dann die bis jetzt implementierten ILL-Dienste für die Fernleihverwaltung erklärt.

Kapitel 1

Manuelle Fernleihe



1

Die manuelle Fernleihe

1.1

Begriffe im Fernleihverkehr

Fernleihe

Um die Buchbestände der Bibliotheken auch den Lesern zugänglich zu machen, die nicht am Ort einer leistungsfähigen Bibliothek ihren Wohnsitz haben, hat sich ein organisiertes System von Bücherversendungen zwischen Bibliotheken durchgesetzt. Dieser Leihverkehr wird auch als **Fernleihe** (Inter Library Loan - ILL) bezeichnet.

Nehmende und gebende Fernleihe

Der Leihverkehr geht grundsätzlich in beide Richtungen: Wer ausleiht, muß auch verleihen. Somit ist die Fernleihabteilung einer Bibliothek in eine **nehmende** und eine **gebende Fernleihe** unterteilt. Die **nehmende Fernleihe** schickt Anfragen an andere Bibliotheken mit der Bitte um Verleihung des angefragten Objektes. Die **gebende Fernleihe** bearbeitet alle eingetroffenen Anfragen von anderen Bibliotheken und verleiht die gewünschten Objekte, sofern kein Eigenbedarf besteht. Somit kann eine Bibliothek bei einem Vorgang gebende Bibliothek und in einem anderen Vorgang nehmende Bibliothek sein.

Aktive und passive Fernleihe

Weiterhin werden zur Unterscheidung auch die Begriffe **aktive Fernleihe** für die gebende Seite und **passive Fernleihe** für die nehmende Seite benutzt. Die Bezeichnung aktiv und passiv ist etwas irreführend, denn die nehmende Bibliothek darf keineswegs passiv sein, wenn sie ihre Aufgabe im Leihverkehr richtig erfüllen will.

Vermittelnde Bibliothek

Bei der Fernleihe werden dem Besteller die gewünschten Objekte innerhalb bestimmter Fristen leihweise überlassen. Der Leihvorgang erfolgt dabei unter den Bedingungen der gebenden Bibliothek und nach dafür festgelegten Regelungen (Leihverkehrsordnung). Die nehmende Bibliothek spielt dabei die Rolle einer Vermittlerin zwischen dem Leser und der gebenden Bibliothek. Die **vermittelnde Bibliothek** sichert den Leihverkehr durch Maßnahmen, die nach der Leihverkehrsordnung zwischen den mitwirkenden Bibliotheken festgelegt sind, besser als dies der Nutzer im direkten Verkehr mit der Bibliothek tun könnte. Die gebende Bibliothek verleiht also nur indirekt an den Benutzer.

Minimierung des Arbeitsaufwands bei der Fernleihe

Beim Leihverkehr handelt es sich um arbeitsaufwendige Sonderleistungen der Bibliothek für bestimmte Nutzer. Deshalb muß der einzelne Besteller seinen maximal möglichen Beitrag für seinen Leihvorgang übernehmen, um die Verwaltungsarbeiten der Bibliothek auf ein Minimum zu reduzieren. Das bedeutet, daß der Leser seine Recherche gewissenhaft durchzuführen hat, um die Angaben über sein gewünschtes Objekt so exakt wie möglich machen zu können. Die Bibliothek hat daraufhin Sorge zu tragen, daß erst die Beschaffungsmöglichkeiten innerhalb der eigenen Bibliothek oder innerhalb des engsten funktionellen bzw. regionalen Bereichs ausgenutzt worden sind, ehe die Einrichtungen weiter entfernter Bibliotheken in Anspruch genommen werden. Das bedeutet, daß vor Inanspruchnahme des Leihverkehrs die Beschaffungsmöglichkeiten vor Ort und vor Einschaltung des internationalen Leihverkehrs die nationalen Bücherbestände ausgeschöpft werden müssen. Das Heranziehen lokaler, regionaler und nationaler Buchbestände ist auch für die Einhaltung einer angemessenen Beschaffungszeit des gesuchten Buches notwendig.

Fernlehschein	Der Leihverkehrsfall tritt ein, wenn ein Benutzer ein bestimmtes Buch, das er braucht, in der Bibliothek nicht findet. Die Bibliothek versucht dann, das Buch von einer anderen Bibliothek zu besorgen. Die Bibliothek füllt einen entsprechenden Leihschein aus. Oft läßt sie ihn auch vom Benutzer ausfüllen und zwar maschinell.
Bibliographieren	Die Bibliothek überprüft die Angaben auf dem Fernlehschein und ergänzt ihn ggf. bibliographisch. Unter Bibliographieren versteht man die vollständige Recherche eines Objektes und die Ermittlung der identifizierenden Angaben.
Bestellnummer	Bei allen Fernleihvorgängen ist die Bestellnummer wichtig. Sie wird von der den Bestellvorgang einleitenden Bibliothek vergeben und auf dem roten Fernlehschein eingetragen.
Leitweg	Nachdem der Fernlehschein vollständig ausgefüllt ist, legt die nehmende Bibliothek den Leitweg fest. Unter Leitweg versteht man die Stationen (Bibliotheken), die ein Fernlehschein auf seinem Weg bis zur Erledigung durchläuft. Die nehmende Bibliothek legt die erste Station fest, indem sie den Leihschein an die nächste besitzende Bibliothek sendet. Kann diese Bibliothek das Objekt aus irgendwelchen Gründen jedoch nicht verleihen, so leitet sie den Leihschein an die nächste besitzende Bibliothek weiter. Somit kann es passieren, daß ein Leihschein mehrere Stationen durchläuft, bis letztendlich eine Bibliothek das gewünschte Objekt verleihen kann.
Transport	Die gebende Bibliothek schickt das Buch dann an die nehmende Bibliothek und behält den Leihschein als Beleg. Beim Versand wird ein Abschnitt des Leihscheins mit Bestellnummer dem Band beigelegt. Der Transport des Objektes erfolgt entweder per Post, per UPS (United Parcel Services) oder mit dem Bücherauto Baden-Württemberg.
Bücherauto Baden-Württemberg	Es existieren zwei Bücherautos . Ein Auto fährt eine nördliche Schleife und besucht auf seiner Route alle angrenzenden Bibliotheken. Das andere Auto fährt eine südliche Schleife. An einem Schnittpunkt treffen sich die beiden Schleifen, so daß ein Austausch zwischen Norden und Süden stattfinden kann. Befindet sich eine Bibliothek nicht an der Route des Bücherautos, so muß sie die Objekte zur nächstgelegenen Bibliothek bringen, die vom Bücherauto angefahren wird. So muß die Universitätsbibliothek Konstanz ihre Objekte zur Universitätsbibliothek Freiburg bringen, wenn sie vom Bücherauto Gebrauch machen will. Sobald die nehmende Bibliothek das gewünschte Objekt erhalten hat, benachrichtigt sie den Benutzer und stellt das Buch zur Abholung bereit. Wenn der Benutzer das Buch wieder zurückbringt, wird es an die gebende Bibliothek zurückgesandt.
Leihgebühren	Leihgebühren werden nur noch im internationalen Leihverkehr erhoben, für den es einen besonderen Leihschein gibt. Bei der nationalen Fernleihe werden dem Leser lediglich die Benachrichtigungs- und Zustellungsgebühren für das gewünschte Objekt berechnet.
Monographien und Zeitschriften-Aufsätze	Im Leihverkehr unterscheidet man grundsätzlich zwischen zwei Arten von ausleihbaren Objekten, den Monographien und den Zeitschriften-Aufsätzen . Monographien sind ein- oder mehrbändige Bücher, die die gebende Bibliothek verleiht und die nach einer gewissen Leihfrist wieder zurückgegeben werden müssen. Zeitschriften hingegen werden nicht verliehen. Bei Zeitschriften-Aufsätzen werden Kopien statt des Originalbandes geliefert, die der Benutzer behalten darf. Sie werden von der gebenden Bibliothek hergestellt und dem Leser in Rechnung gestellt.

1.2 Entwicklung der manuellen Fernleihe

Nutzung des Fernleihangebots

In der Bibliothek der Universität Konstanz wird das Angebot der Fernleihe vom Nutzer immer mehr in Anspruch genommen. Waren es 1990 noch ca. 22800 Bestellungen in der nehmenden Fernleihe, so sind es 1995 bereits an die 34000 gewesen. Auch die Zahl der Vorgänge in der gebenden Fernleihe stieg in dieser Zeit von 45700 auf 68600.

Die folgende Abbildung zeigt die Jahresergebnisse der nehmenden Fernleihe von 1990 bis 1995:

Abb. 1.1
Nehmendes Fernleihaufkommen

Nehmende Fernleihe					
Jahr	abgesandte Bestellungen	erledigt		erhaltene	
		positiv	negativ	Bände	Kopien
1990	22.834	20.827	2.007	11.004	10.919
1991	27.899	24.451	3.448	12.398	13.777
1992	28.847	25.984	2.863	13.043	14.607
1993	30.779	27.912	2.867	12.244	16.604
1994	30.670	27.414	3.256	12.052	16.192
1995	33.970	30.855	3.115	14.396	17.502

Die folgende Abbildung zeigt die Jahresergebnisse der gebenden Fernleihe von 1990 bis 1995:

Abb. 1.2
Gebendes Fernleihaufkommen

Gebende Fernleihe					
Jahr	eingegangene Bestellungen	erledigt		versandte	
		positiv	negativ	Bände	Kopien
1990	45.671	31.138	14.533	19.477	12.237
1991	49.899	33.747	16.152	20.021	15.532
1992	59.061	39.958	19.103	23.691	17.376
1993	66.489	45.850	20.639	27.730	19.144
1994	69.334	45.371	23.963	27.624	20.047
1995	68.593	45.077	23.516	25.756	20.418

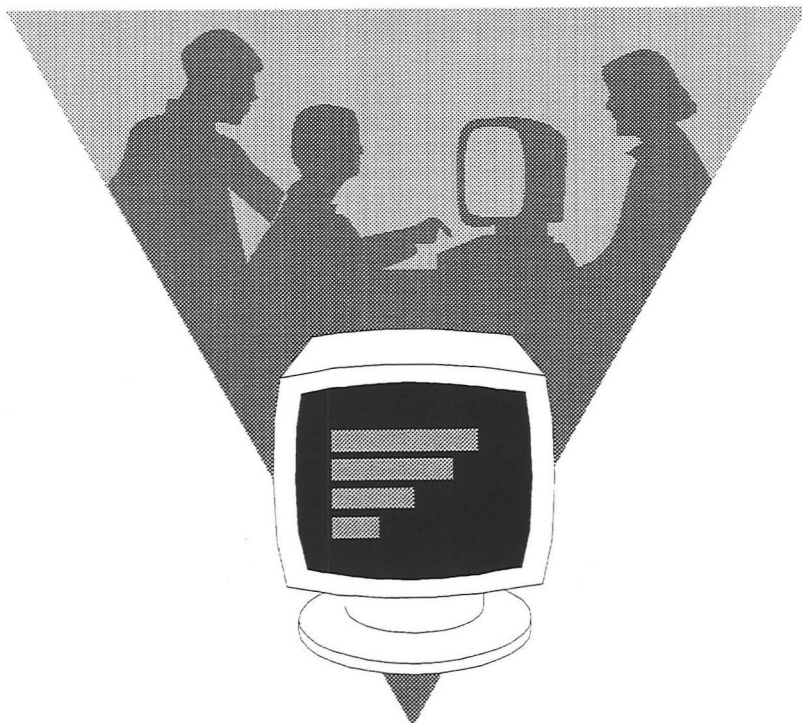
Gegenläufige Tendenz	Die Situation in der manuellen Fernleihe ist durch zwei gegenläufige Tendenzen gekennzeichnet: Zum einen sind die Möglichkeiten der Recherche durch eine Vielzahl neuer Standortnachweise (Zeitschriftendatenbank, Verbundkataloge, Zentralkataloge) nachhaltig verbessert worden, zum anderen ist das Leihscheinauftreten erheblich angestiegen.
Lieferzeiten	Ergebnis dieser gegenläufigen Entwicklung ist, daß die Lieferzeiten in den letzten 20 Jahren stagnieren. Während die Laufzeit einer Fernleihbestellung in den 70er Jahren rund 18 Kalendertage betrug, wurde in der Konstanzer Untersuchung von 1989 und 1990 eine durchschnittliche Laufzeit von 19,7 Kalendertagen ermittelt [Kuon, Lehmler, Mühle, 1991]. Gründe für die langen Laufzeiten können z.B. unzureichende oder falsche Angaben auf dem Leihschein sein oder die besitzende Bibliothek hat das Objekt bereits verliehen, so daß die Bestellung an die nächste besitzende Bibliothek weitergegeben werden muß usw.
Unzufriedenheit der Fernleihbenutzer	Wenn daher im DFG ¹ -Projekt „Benutzerverhalten an deutschen Hochschulbibliotheken“ von 1973 festgestellt wurde, daß 42 % der befragten Fernleihbenutzer erklärten, für ihren Zweck seien die Fernleihbestellungen zu spät gekommen, so darf man wohl davon ausgehen, daß der Prozentsatz unzufriedener Fernleihbenutzer heute eher noch höher anzusetzen ist. Dies vor allem deshalb, weil die Benutzer ständig die Diskrepanz zwischen der schnellen bibliographischen Ermittlung von Literaturnachweisen etwa über Fachdatenbanken und der langsamen Beschaffbarkeit dieser Literatur erfahren.
Fehlende Transparenz	Gravierender noch als die langen Laufzeiten der Fernleihbestellungen ist aber die Tatsache, daß weder der Benutzer noch die bestellende Bibliothek in irgendeiner Weise Einfluß auf den Lauf einer Fernleihbestellung nehmen kann. Der Fernleihschein wird abgeschickt und irgendwann erfolgt im Fall der positiven Erledigung die Zusendung des gewünschten Dokumentes. Der Weg, den der Fernleihschein zwischenzeitlich genommen hat, ist jedoch - sieht man von der anfänglichen Leitwegfestlegung durch die bestellende Bibliothek einmal ab - weder verfolgbar noch steuerbar. Rückfragen seitens des Benutzers sind zwecklos und können somit nicht zur Beschleunigung beitragen, da die konventionellen Fernleihkarteien in der Regel nach Bestellnummern geordnet sind, die dem Benutzer bei Aufgabe seiner Bestellung gar nicht bekannt sind, da sie erst später vergeben werden.
Spaltung der Fernleihe	Folge der langen Laufzeiten im Deutschen Leihverkehr ist eine Spaltung der Fernleihe . Für zahlungskräftige Kunden besteht die Möglichkeit, außerhalb des Deutschen Leihverkehrs gegen entsprechende Gebühren Direktbestellungen bei den zentralen Fachbibliotheken oder Datenbank-Hosts aufzugeben. Bei einer Direktbestellung wird die Bestellung an eine gewünschte Bibliothek gesendet, ungeachtet der Bestimmung, naheliegende Bibliotheken bevorzugt zu behandeln. Auch die Spaltung der Fernleihe in teure Eilt-Bestellungen außerhalb des Deutschen Leihverkehrs und billige, aber langsame Bestellungen innerhalb des Deutschen Leihverkehrs ist eine Entwicklung, die zu Bedenken Anlaß gibt. Da die Bestellungen im Deutschen Leihverkehr so lange Laufzeiten haben, weichen mehr und mehr Benutzer auf die schnellere Fernleihalternative aus. Da gebührenpflichtige Direktbestellungen vorrangig behandelt werden, wird das Tempo im Deutschen Leihverkehr weiter verlangsamt. Das Nachsehen haben jene Fernleihkunden, die sich die teuren Direktbestellungen nicht leisten können.

¹ Deutsche Forschungsgemeinschaft.

Die manuelle Fernleihe

Kapitel 2

Elektronische Fernleihe



2

Entwicklung der elektronischen Fernleihe

2.1 Planung der elektronischen Fernleihe

Elektronische Post	Die mangelnde Transparenz ist ein gravierender Schwachpunkt der herkömmlichen Fernleihabwicklung. Der Einsatz der elektronischen Post kann zu einer ganz erheblichen Verbesserung der Situation beitragen. Die Vorteile der elektronischen Post, insbesondere die Schnelligkeit der Informationsweiterleitung und die weit größere Transparenz, sind entscheidend für Fortschritte in der Fernleihe.
Transparenz	Der Einsatz der elektronischen Post ermöglicht eine vollständige Transparenz eines Fernleihvorgangs. Der Weg einer Fernleihbestellung kann von der bestellenden Bibliothek permanent verfolgt und auch während der Laufzeit beeinflusst werden. So kann auch die räumliche Distanz zwischen dem Benutzer und der für ihn tätig werdenden auswärtigen Bibliothek überwunden werden. Der Benutzer kann sich jederzeit darüber informieren, wo sich seine Bestellung befindet und überfällige Lieferungen über seine Bibliothek reklamieren. Die Möglichkeit, in dringenden Fällen für eine bevorzugte Behandlung der betreffenden Bestellung sorgen, reklamieren oder stormieren zu können, wäre somit ein Gewinn.
Projekt der Uni Konstanz	Angesichts der o.g. Tatsachen hat die Bibliothek der Universität Konstanz ein Projekt beantragt, dessen Ziel es sein soll, Lösungen zu entwickeln und zu erproben, mit denen die Fernleihe durch Nutzung moderner Informations- und Kommunikationstechnologien beschleunigt werden kann. Das Projekt wurde im Mai 1991 von der DFG bewilligt.
X.400 und SMTP/TCP/IP	Zur elektronischen Übermittlung der Fernleihschein und der sonstigen Nachrichten, die zur Abwicklung einer Bestellung ausgetauscht werden müssen, ist der Einsatz der Protokoll-Stacks nach ISO/ITU ² X.400 ³ oder SMTP/TCP/IP ⁴ vorgesehen. X.400 und TCP/IP ermöglichen die Übermittlung von strukturierten Dokumenten und Daten zwischen Sender und Empfänger.
ILL-Protokoll	Für die zu übermittelnden Nachrichten selbst, bedarf es eines ergänzenden Standards. Hierzu bietet sich die Verwendung des ILL-Protokolls ⁵ an. Das ILL-Protokoll ist Teil der Anwendungsschicht im OSI-Referenzmodell. Das ILL-Protokoll definiert eine Menge von Standard-Nachrichten, die speziell an den Bedürfnissen des Leihverkehrs ausgerichtet sind und während eines Fernleihvorgangs ausgetauscht werden. Es sieht insgesamt 21 solcher Standardnachrichten (Dienste) vor. Im Grunde sind die ILL-Standards nichts anderes als die allgemein akzeptierten „Normen“ für das Ausfüllen des konventionellen Leihscheins, ergänzt um standardisierte Mitteilungen, die innerhalb eines Fernleihvorganges ausgetauscht werden.

² International Telecommunication Union (früher CCITT).

³ Siehe Abschnitt 5.1.

⁴ Siehe Abschnitt 5.2.

⁵ Siehe Abschnitt 3.

Integration des Fernleihsystems

Würde sich die elektronische Fernleihe allein auf die elektronische Übermittlung des Fernleihscheines und der sonstigen Korrespondenz beschränken, wäre der zu erwartende Beschleunigungseffekt zweifellos sehr begrenzt, da damit - im Vergleich zur gegenwärtigen konventionellen Abwicklung der Fernleihe - allein die Postlaufzeiten des Fernleihscheins wegfallen würden. Eine wirkliche Rationalisierung dürfte nur dann zu erwarten sein, wenn auch die gesamte Fernleihverwaltung automatisiert wird und dieses System wiederum voll in das lokale Bibliothekssystem integriert wird. Die Verwaltung der Fernleihbestellungen soll zwar in einer gesonderten Datenbank erfolgen, die aber mit der lokalen Ausleihdatenbank vernetzt ist, so daß die Verbuchung von Fernleihvorgängen mit den Routinen des lokalen Ausleihsystems erfolgen kann.

2.2

Ablauf der elektronischen Fernleihe

Fernleih-Bestelleingabe

Der Benutzer gibt seine Bestellung unmittelbar am PC mittels einer speziellen Eingabemaske ein. Dies kann in der Bibliothek geschehen, ebenso aber auch vom Arbeitsplatz des Benutzers in der Universität über das inneruniversitäre Netz oder von Zuhause über das analoge Telefonnetz oder über ISDN etc. Für die Gebühreneinnahme müssen damit zukünftig neuartige Verwaltungslösungen entwickelt werden.

Prüfung

Vom Mitarbeiterplatz in der Fernleihe wird die Bestellung dann überprüft. Ist die Bestellung tatsächlich berechtigt, d.h. das gewünschte Objekt ist in der lokalen Datenbank nicht vorhanden, so werden die Angaben der Bestellung kontrolliert und gegebenenfalls bibliographisch ergänzt. Alternativ zu dieser Vorgehensweise soll auch die Möglichkeit geboten werden, daß der Benutzer eigenständig in Fremddatenbanken bzw. Verbunddatenbanken recherchieren kann und das Ziel seiner Bestellung selbst bestimmt.

Versendung der Leihscheine

Die **Versendung der Leihscheine** mit allen erforderlichen Angaben erfolgt dann über das WiN (Wissenschaftsnetz) gemäß den ISO/ITU- oder Internet-Protokollen. Für die Fälle, in denen Fernleihpartner über das WiN nicht erreicht werden können, soll die Übertragung von Leihscheinen per Fax möglich sein.

Fernleihkorrespondenz

Ist das Dokument in der angegangenen Bibliothek gefunden, können ggf. Sonderkonditionen mitgeteilt und die Zustimmung dazu eingeholt werden. Die Abwicklung dieser **Fernleihkorrespondenz** erfolgt elektronisch gemäß dem ILL-Protokoll.

3

Das Inter-Library-Loan (ILL) - Protokoll

3.1

Entstehung und Funktionalität

Entstehung des Protokolls

Die Kanadische Nationalbibliothek hatte schon Anfang der 80er Jahre die Notwendigkeit eines standardisierten Leihverkehrs zwischen den einzelnen Bibliotheken erkannt und begann in enger Zusammenarbeit mit anderen Bibliotheken und technischen Experten, einen Standard für die Kommunikation zwischen den im Leihverkehr eingebundenen Bibliotheken zu entwickeln.

Inter Library Loan Protokoll

Nach langjähriger Arbeit, vielen Absprachen und Überarbeitungen wurde schließlich 1991 das ILL (Inter Library Loan) - Protokoll von der „International Organisation for Standardization (ISO)“ als internationaler Standard (ISO 10160 / 10161) anerkannt und zertifiziert.

Die aus dieser Arbeit entstandenen Dokumente geben eine generelle, maschinenunabhängige Beschreibung des ILL-Protokolls. Die Dokumente unterstützen auch Programmierer, die eine ILL-Dienst-Umgebung auf Basis dieses Protokolls implementieren wollen.

Open System Interconnection

Das Protokoll basiert auf dem Referenzmodell der Open System Interconnection (OSI). Das generelle Ziel von OSI ist es, mit einem Minimum an technischen Absprachen außerhalb des OSI-Standards, die Verbindung von informationsverarbeitenden Systemen verschiedener Hersteller, verschiedener Komplexitätsgrade und verschiedener Technologien zu ermöglichen.

Das ILL-Protokoll stellt eine Menge von Diensten in der Anwendungsebene (application layer) des OSI - Referenzmodells zur Verfügung. Diese können von Bibliotheken benutzt werden, die alle Fernleihaktivitäten in einer offenen Systemumgebung durchführen wollen. Diese Umgebung ist durch ISO 7498⁶ definiert.

Leistungsumfang ILL-Protokoll

Durch das ILL-Protokoll hat man die Möglichkeit, die Ausleihe sogenannter „returnable objects“, wie z. B. Bücher, oder sogenannter „nonreturnable objects“, wie z. B. Photokopien von Zeitschriftenaufsätzen, zu erbitten. Als „returnable objects“ bezeichnet man bibliographische Objekte, die zurückgegeben werden müssen. Die „nonreturnable objects“ sind entsprechend bibliographische Objekte, die nicht zurückgegeben werden müssen. Verwandte Dienste, wie die Verlängerung einer Ausleihe, vorzeitiger Rückruf eines Objektes und Hinweis auf Leihfristüberschreitung, sind ebenfalls im Leistungsumfang des Protokolls enthalten.

Dem Benutzer wird eine Menge von Diensten durch ein sogenanntes Anwendungsdienstelement (application service element) zur Verfügung gestellt.

Eine gesamte ILL-Transaktion ist dadurch charakterisiert, daß ein Benutzer Dienste des ILL-Protokolls in einer definierten Reihenfolge ausführt.

Mit den zur Verfügung stehenden Diensten sollen einige Ziele erreicht werden.

⁶ Siehe Anhang I.

Diese sind im einzelnen:

- Kontrolle der ILL-Transaktion,
- Zusammenarbeit verschiedener Systeme,
- Minimierung der Fernleihkosten,
- Widerspiegelung der derzeitigen Fernleihpraxis.

Das ILL-Protokoll ist eines aus einer Anzahl verwandter Standards, welche für die Vernetzung von Bibliothekssystemen nützlich sind. Diese Standards können für sich allein oder in kooperativer Art und Weise benutzt werden, um Bibliotheksanwendungen zu unterstützen. Diese Dienste müssen aber nicht zwangsweise bibliotheksspezifisch sein.

Search and Retrieve Protocol

Hier kann man das SR-Protokoll (Search and Retrieve Protocol) anführen, welches den Zugriff auf entfernte bibliographische Datenbanken unterstützt, um Informationen zur Objektidentifikation zu erhalten. Auch dieses Protokoll wurde 1991 als ISO 10162 / 10163 von der „International Organisation for Standardization“ als internationaler Standard anerkannt und kann in Verbindung mit dem ILL-Protokoll angewandt werden.

X.400-Protokoll

Ein weiterer nicht bibliotheksspezifischer Standard für „Message Handling“-Systeme ist das X.400-Protokoll⁷.

File Transfer and Access Management

Der letzte hier anzuführende ISO-Standard ist das „File Transfer and Access Management (FTAM)“-Protokoll, welches zur Übertragung von großen Dateien als Unterstützung der Katalogisierung verwendet wird. Dieser Standard ist ebenfalls nicht bibliotheksspezifisch.

Die Kontrolle und das Management von Interaktionen zwischen solchen bibliographischen Anwendungen gehen aber über den Rahmen des ILL-Standards hinaus.

Die folgende Beschreibung des ILL-Protokolls hält sich eng an die originalen ISO-Dokumente und an Dokumente, die während der Implementierung des Fernleihsystems der Uni Konstanz zu diesem Thema entstanden sind. Die Beschreibung beschränkt sich jedoch nur auf diejenigen Aspekte, die für das allgemeine Verständnis des Protokolls nötig sind.

3.2

Normative Verweise

Verweise auf andere Normen, die in diesem Zusammenhang von Bedeutung sind, sind aus Gründen der Übersichtlichkeit im Anhang I zu finden.

3.3

Rahmenbedingungen und Anwendungsbereich

Rahmenbedingungen

Der ILL-Standard definiert die Dienste für die Fernleihe. Das ILL-Protokoll stellt diese Dienste in Verbindung mit unterstützenden Telekommunikationsdiensten zur Verfügung.

⁷ Siehe Abschnitt 5.1.

Bei diesen handelt es sich um einen „store-and-forward“-Nachrichtendienst, wie er von den MOTIS⁸ Standards (ISO 10021) zur Verfügung gestellt wird, oder um einen direkten verbingungsorientierten Modus, welcher u.a. in den Standards ISO 8822 / 8649⁹ Einsatz findet.

Der ILL-Standard spezifiziert weder individuelle Implementierungen oder Produkte, noch schränkt er die Implementierung von Instanzen (entities) und Schnittstellen innerhalb eines Rechnersystems ein. Die verwendeten Rechnersysteme können von einer Workstation bis zum Mainframerechner reichen.

Anwendungsbereich

Der ILL-Standard ist für die Anwendung durch Bibliotheken, Informationszentren, wie Zentralkataloge, und jedes andere System, das bibliographische Informationen be- und verarbeitet, bestimmt.

Diese Systeme können an einer ILL-Tansaktion in der Rolle eines

Requester oder „nehmende Bibliothek“

- **Requesters,**

Initiator eines Fernleihwunsches. Man spricht in diesem Zusammenhang auch von der „**nehmenden Bibliothek**“.

Responder oder „gebende Bibliothek“

- **Responders** oder eines

Anbieter bibliographischen Materials oder bibliographischer Information. Man spricht in diesem Zusammenhang auch von der „**gebenden Bibliothek**“.

Intermediary oder Vermittler

- **Intermediary**

Vermittler, der im Auftrag eines Requesters handelt, um einen passenden Responder zu finden.

teilnehmen.

Der ILL-Standard unterstützt außerdem verschiedene Netztypologien, die von einfachen bilateralen Interaktionen bis hin zu multilateralen Interaktionen reichen. Die einzelnen Typologien werden im weiteren Verlauf des Kapitels näher beschrieben.

3.4 Dienstmodell

3.4.1 Dienstbenutzer und Dienstanbieter

Anwendungsprozess

Die ILL-Anwendung ist als eine verteilte Menge von **Anwendungsprozessen** (application process) modelliert, wobei jeder von ihnen in einem separaten offenen System, zum Beispiel einem Bibliothekssystem, eingerichtet wird.

Anwendungsinstanz

Innerhalb eines jeden Anwendungsprozesses gibt es zwei Funktionstypen, die lokalen Verarbeitungsfunktionen und die kommunikationsbezogenen Funktionen, d.h. Funktionen unterer Schichten. Die lokalen Verarbeitungsfunktionen kümmern sich um Aktivitäten wie z.B. Datenmanipulation und Reportgenerierung und gehen über den Rahmen des ILL-Standards hinaus. Innerhalb jedes Systems werden die Aspekte des Anwendungsprozesses mit kommunikationsbezogenen Funktionen **Anwendungsinstanz** (application entity) genannt.

⁸ Siehe Abschnitt 5.1.

⁹ Siehe Anhang I.

Anwendungs-
dienstelement Jede Anwendungsinstanz enthält bereits ihrerseits ein oder mehrere **Anwendungsdienstelemente** (application service elements). Diese Anwendungsdienstelemente bieten dem Dienstbenutzer (ILL service user) kommunikationsbezogene Dienste an.

Dabei beteiligen sie sich beim Protokoll austausch mit Partneranwendungsinstanzen, wobei sie sich unterstützende Dienste innerhalb der Anwendungsschicht und darunter liegender Schichten zunutze machen. Beziehungen zu anderen Anwendungsdienstelementen müssen separat beschrieben werden, da diese wiederum über den Rahmen des internationalen Standards hinaus gehen.

ILL-Dienstanbieter Die Menge aller ILL-Anwendungsdienstelemente bildet zusammen mit den Diensten der unteren Schichten aller Systeme den **ILL-Dienstanbieter** (ILL service provider).

Wie bereits in Abschnitt 3.3 erwähnt, kann nun ein Dienstbenutzer, der an einer Fernleihe beteiligt ist, die Rolle eines Requesters, eines Responders oder eines Intermediary einnehmen.

Der Requester generiert ILL-Anfragen. Der Responder empfängt diese und ist der potentielle Lieferant des gewünschten Leihobjektes (item). Der Intermediary hingegen ist ein Responder, der selbst keine ILL-Anfragen erfüllt, und diese im Interesse des Requesters an einen Responder weiterleitet.

Der tatsächliche Lieferant eines gewünschten Leihobjektes ist normalerweise ein Responder. Jedoch können nach dem Dienstmodell auch Institutionen, die selbst keine elektronischen Fernleih Anfragen gemäß dem ILL-Standard empfangen können, angeforderte Objekte liefern. Diese unterstützen die Fernleihe z.B. nur per Telefon oder Post. Wollen solche Institutionen auch an der Möglichkeit der elektronischen Fernleihe teilhaben, können sie sich einer anderen Institution bedienen, die die elektronische Fernleihe unterstützt und im Auftrag als Responder für diese agieren kann.

3.4.2

ILL-Transaktion, ILL-Subtransaktion und ILL-Transaktionsgruppe

ILL-Transaktion,
ILL-Zyklus

Eine **ILL-Transaktion** ist als einzelne, komplette Instanz eines gesamten **ILL-Zyklus** zu bezeichnen, der alle Aktionen, Dienstaufrufe und Nachrichten enthält, die durch die ursprüngliche ILL-Anfrage hervorgerufen wurden. Der Zyklus endet mit der Rückgabe des Leihobjektes.

Dabei gilt Folgendes:

- ILL-Transaktionen können sich zeitlich überlappen, d.h. mehrere ILL-Transaktionen können von einem System gleichzeitig bearbeitet werden.
- Eine ILL-Transaktion kann nur durch einen Requester eingeleitet werden.

ILL-Subtransaktion

Eine **ILL-Subtransaktion** bezeichnet eine Menge von Kommunikationsaktivitäten zwischen einem Intermediary und einem Responder oder einem weiteren Intermediary und ist an eine ILL-Transaktion geknüpft, die von einem Requester eingeleitet wurde.

- Eine ILL-Subtransaktion kann nur von einem Intermediary eingeleitet werden.

Bezieht eine ILL-Transaktion drei oder mehr Fernleihpartner ein, so ist derjenige Requester, der die Fernleihanfrage durch die Generierung eines ILL-REQUEST¹⁰ einleitet und derjenige Fernleihpartner Responder, der den ILL-REQUEST zuletzt erhält.

Mehrere eigenständige ILL-Transaktionen können trotzdem eine Verbindung untereinander aufweisen. Zum Beispiel kann ein Requester eine Folge von Versuchen starten, um direkten Kontakt mit verschiedenen Respondern zu bekommen.

ILL-Transaktions-
gruppe

Solche ILL-Transaktionen werden dann als eine **ILL-Transaktionsgruppe** bezeichnet. Dadurch kann z.B. eine Historie eines Fernleihzykluses erstellt werden, die alle mit diesem Zyklus verwandten Aktionen enthält.

ILL-Transaktions-
Identifizier

Jede einzelne ILL-Transaktion hat einen eindeutigen ILL-Transaktions-Identifizier. Dieser wird benutzt, um alle Informationen (z.B. Status- oder beschreibende Informationen) einer bestimmten ILL-Transaktion zuzuordnen zu können.

Eine ILL-Transaktion beinhaltet u.a. folgende Komponenten:

- **initial-requester-id**
Identifikation des Requesters, der die ILL-Transaktion eingeleitet hat.
- **ILL-transaction-group-qualifier**
unterscheidet eine ILL-Transaktionsgruppe von allen anderen aktiven ILL-Transaktionsgruppen des Requesters.
- **ILL-transaction-qualifier**
unterscheidet eine ILL-Transaktion von allen anderen.

Der ILL-Transaktions-Identifizier jeder ILL-Subtransaktion hat die folgende zusätzliche Komponente, die nur innerhalb der Umgebung eines Intermediary eindeutig ist:

- **sub-transaction-qualifier**
unterscheidet diese Subtransaktion von allen anderen Subtransaktionen innerhalb einer ILL-Transaktion, die durch einen Intermediary eingeleitet wurde.

3.4.3

ILL-Transaktionstypen und Topologien

Es gibt drei ILL-Transaktionstypen:

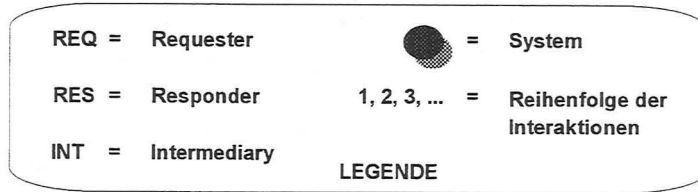
- einfache (simple),
- verkettete (chained),
- partitionierte (partitioned) Transaktion.

Zusätzlich zu den obigen drei Grundtypen können beliebige Kombinationen aus den Grundtypen realisiert werden.

¹⁰ Siehe Abschnitt 3.5.2.

In den folgenden Abschnitten werden die Grundtypen und einige wichtige Kombinationsmöglichkeiten anhand von Abbildungen veranschaulicht. Die in den Abbildungen verwendeten Symbole und Abkürzungen erklären sich mit der folgenden Legende, auf die bei den einzelnen Abbildungen referenziert wird.

Legende zu den folgenden Abbildungen

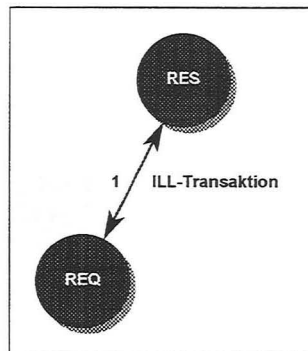


3.4.3.1 Die einfache ILL-Transaktion (Simple)

Punkt-zu-Punkt-Verbindung

Eine einfache ILL-Transaktion bezieht zwei aktive Fernleihpartner mit ein - den Requester und den Responder. Bei diesem Transaktionstyp arbeiten der Requester und der Responder in einer **Punkt-zu-Punkt - Verbindung**.

Abb. 3.1 Einfache ILL-Transaktion



Legende siehe Abschnitt 3.4.3

Alle ILL-Transaktionen, die durch einen Requester eingeleitet wurden, beginnen als einfache ILL-Transaktionen. Jedoch kann ein Requester in seiner Fernleihanfrage (ILL-REQUEST) angeben, daß der Responder die Erlaubnis hat, den ILL-Transaktionstyp in eine verkettete oder partitionierte ILL-Transaktion zu ändern. Wenn der Responder den ILL-Transaktionstyp ändert, wird er automatisch zum Intermediary.

Sollte ein Responder nicht in der Lage sein, eine Fernleihanfrage erfolgreich zu erfüllen, kann er, um den Requester zu unterstützen, seinerseits eine Liste mit weiteren potentiellen Respondern liefern.

3.4.3.2 Die verkettete ILL-Transaktion (Chaining)

Eine verkettete ILL-Transaktion bezieht wenigstens drei Fernleihpartner ein - den Requester, den Responder und einen oder mehrere Intermediaries. Ein Fernleihwunsch wird von einem Intermediary (evtl. über weitere Intermediaries) zum Responder weitergereicht, wobei jeder Intermediary als eine Art Relaisstation für alle ILL-Nachrichten in der Kette zu bezeichnen ist.

Das Inter-Library-Loan (ILL) - Protokoll

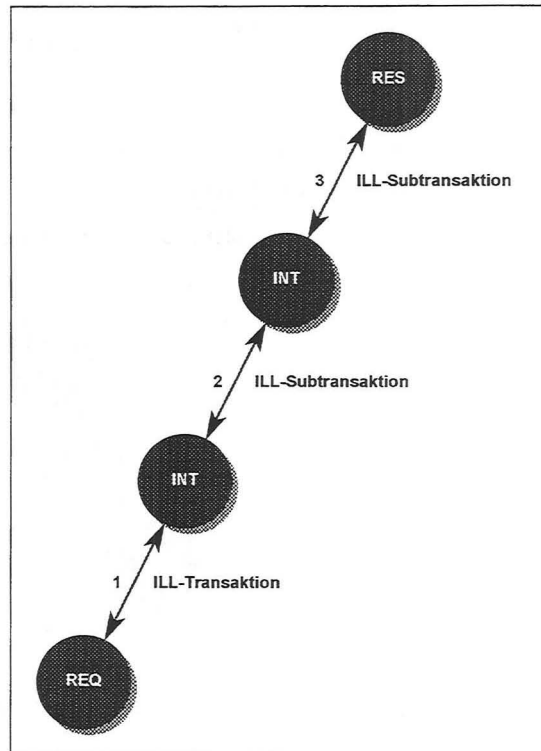
Es besteht in diesem Fall also keine direkte Wechselbeziehung zwischen dem Requester und dem Responder.

ILL-Haupttransaktion

Die Interaktionen zwischen dem Requester und dem ersten Intermediary definieren die **ILL-Haupttransaktion**. Die Menge aller Interaktionen zwischen einem Intermediary und einem Responder bilden eine ILL-Subtransaktion, ebenfalls alle Interaktionen eines jeden Paares dazwischenliegender Intermediaries.

Die folgende Abbildung zeigt eine verkettete ILL-Transaktion mit zwei Intermediaries.

Abb. 3.2
Verkettete ILL-
Transaktion mit 2
Intermediaries



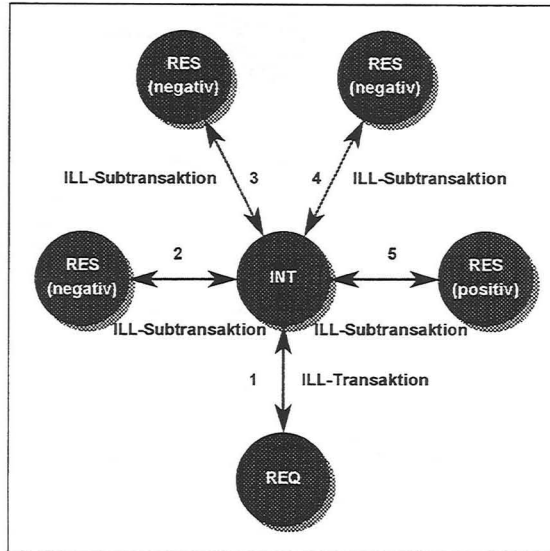
Legende siehe Abschnitt 3.4.3

Führt eine ILL-Subtransaktion nicht zur Erfüllung einer Fernleihanfrage, kann der letzte Intermediary in der Kette eine neue ILL-Subtransaktion mit einem anderen Responder einleiten. Bei jedem weiteren Scheitern kann er nun der Reihe nach versuchen, weitere potentielle Responder anzusprechen.

Sterntopologie

Dieses Vorgehen führt zur Sterntopologie mit einem Intermediary als Mittelpunkt (hub).

Abb. 3.3
Sterntopologie



Legende siehe Abschnitt 3.4.3

Ein Responder kann eine Liste mit potentiellen Respondern an den Intermediary liefern, um diesen in seiner Auswahl zu unterstützen. Das angeforderte Leihobjekt wird entweder direkt an den Requester, an den Kunden des Requesters oder an einen der Vermittler geschickt. Dieser ist dann für die Lieferung an den Requester bzw. an den Kunden des Requesters verantwortlich.

Der Requester kann die Verkettung gestatten oder verbieten und kann, falls dies gewünscht wird, in einer Liste selbst potentielle, eventuell auch präferierte Responder vorschlagen. Zusätzlich hat der Requester die Möglichkeit, in einer weiteren Liste diejenigen Responder an den Intermediary weiterzugeben, bei denen schon ein Leihversuch unternommen wurde. Mit diesem Vorgehen können unnötig doppelte Fernleihanfragen vermieden werden.

3.4.3.3

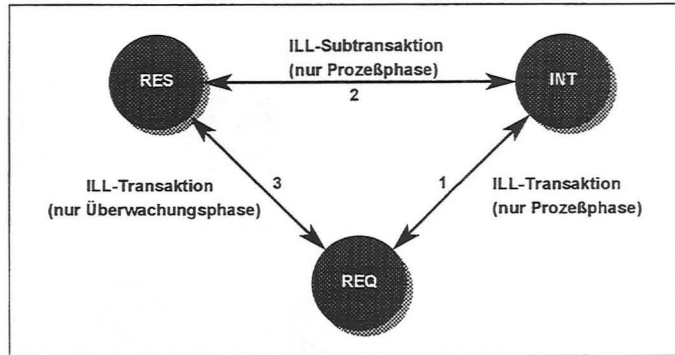
Die partitionierte ILL-Transaktion (Partitioning)

Eine partitionierte ILL-Transaktion bezieht, wie auch die verkettete Variante, wenigstens drei Fernleihpartner mit ein - den Requester, den Responder und einen oder mehrere Intermediaries. Eine Fernleihanfrage wird vom Intermediary zum Responder durchgereicht, der seinerseits dem letzten Intermediary antwortet. Diese Antwort wird ebenfalls durchgereicht, diesmal zurück an den Requester.

Nachdem das gewünschte Leihobjekt vom Responder ausgeliefert wurde und der Requester zusätzlich noch einen Hinweis vom Responder erhalten hat, daß das Leihobjekt versandt wurde, finden alle weiteren Interaktionen nur noch direkt zwischen dem Requester und dem Responder statt. Der Intermediary nimmt nicht länger an der ILL-Transaktion teil.

Die folgende Abbildung zeigt eine partitionierte ILL-Transaktion.

Abb. 3.4
Partitionierte
ILL-Transaktion



Legende siehe Abschnitt 3.4.3

Partitionierte ILL-Transaktionen sind in Situationen nützlich, in denen ein Intermediary im Auftrag eines Requesters handelt, um einen passenden Responder zu finden, selbst aber kein Interesse hat, weiter an der ILL-Transaktion teilzunehmen. Diese Situation trifft man häufig bei Zentralkatalogen an.

Prozeß- und Über-
wachungsphase

Eine partitionierte ILL-Transaktion wird in zwei Phasen aufgeteilt. Die erste Phase, die **Prozeßphase** (processing phase), besteht aus den Interaktionen zwischen dem Requester und dem Responder, denen ein Intermediary zwischengeschaltet ist. Während dieser Phase bildet die Menge aller Interaktionen zwischen den Intermediaries selbst und zwischen dem letzten Intermediary und dem Responder die schon erwähnte ILL-Subtransaktion. Die zweite Phase, die **Überwachungsphase** (tracking phase), besteht aus der direkten Interaktion zwischen dem Requester und dem Responder. Sie wird zur Überwachung der Ausleihe des Objektes verwendet, einschließlich einiger Dienste des ILL-Protokolls¹¹ wie z.B. vorzeitige Rückrufaktionen, Leihfristverlängerungen oder auch Leihfristüberschreitungen.

Das angeforderte Leihobjekt könnte nun direkt an den Requester, an einen Kunden des Requesters oder an einen Intermediary gesendet werden. Dieser ist dann für dessen Auslieferung an den Requester bzw. an den Kunden des Requesters verantwortlich.

Der Requester kann die Partitionierung gestatten oder verbieten und kann, falls dies gewünscht wird, in einer Liste selbst potentielle, eventuell auch präferierte Responder vorschlagen. Zusätzlich hat der Requester die Möglichkeit in einer weiteren Liste diejenigen Responder an den Intermediary weiterzugeben, bei denen schon ein Leihversuch unternommen wurde. Mit diesem Vorgehen können unnötig doppelte Fernleihanfragen vermieden werden.

Sollte ein Responder nicht in der Lage sein, eine Fernleihanfrage erfolgreich zu erfüllen, kann er, um den Requester zu unterstützen, seinerseits eine Liste mit weiteren potentiellen Respondern liefern.

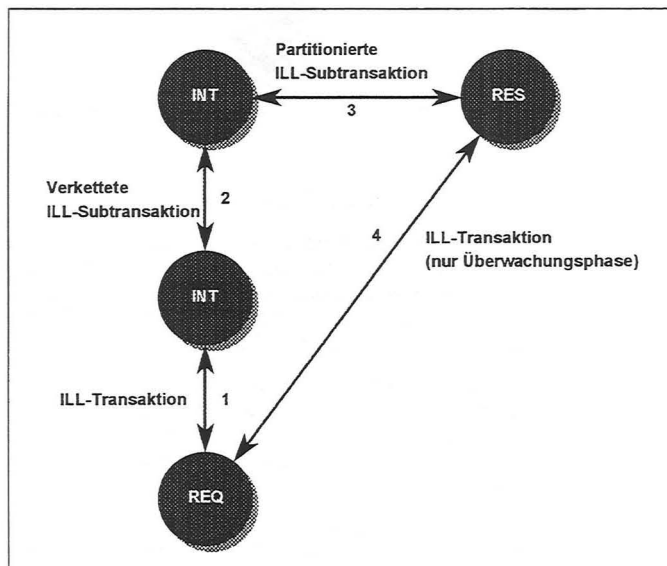
¹¹ Siehe Abschnitt 3.5.

3.4.3.4 Mischformen von Verkettung und Partitionierung

Die Verkettung und die Partitionierung können innerhalb derselben ILL-Transaktion miteinander kombiniert werden. Es ist dabei allerdings Folgendes zu beachten:

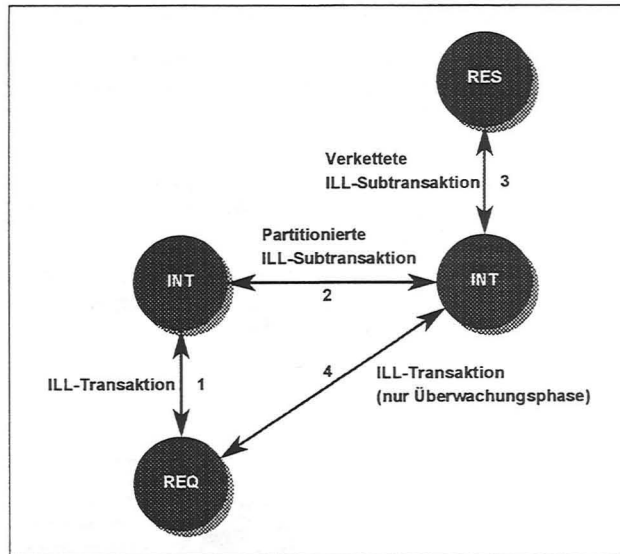
- Wird die Partitionierung nach der Verkettung angewandt, ist die Verkettung außer Kraft gesetzt (siehe Abb. 3.5a). Bei mehrfacher Partitionierung bleibt der Effekt der gleiche.
- Folgt jedoch eine Verkettung auf eine Partitionierung, bleiben die Auswirkungen der Verkettung auf den weiteren Verlauf der ILL-Transaktion gewahrt (siehe Abb. 3.5b).

Abb. 3.5a
Partitionierung nach
Verkettung



Legende siehe Abschnitt 3.4.3

Abb. 3.5b
Verkettung nach
Partitionierung



Legende siehe Abschnitt 3.4.3

3.4.3.5 Getrennte ILL-Transaktionen

Der Vollständigkeit halber ist hier noch eine weitere Art von ILL-Transaktionen erwähnt. Die zuvor beschriebene verkettete oder partitionierte ILL-Transaktion geht davon aus, daß der Intermediary während der Überwachungsphase nur die Rolle eines Relais spielt, d.h. er benutzt keine Dienste des ILL-Protokolls in Eigeninitiative (z.B. Leihfristablauf).

Das ILL-Dienstmodell erlaubt indessen auch einem potentiellen Intermediary eine aktive Rolle während der Prozeßphase zu spielen und die Kontrolle über alle Phasen einer ILL-Transaktion auszuüben, sofern er getrennte ILL-Transaktionen für seine Interaktionen mit den Responder bzw. dem Requester einsetzt.

Getrennte ILL-
Transaktionen,
„dual-role system“

Ein System, welches eine Fernleihanfrage erhält, kann aus der Sicht des ursprünglichen Requesters wie ein endgültiger Responder und in einer zweiten ILL-Transaktion selbst wie ein ursprünglicher Requester agieren. Als solcher nimmt er dann in dieser zweiten ILL-Transaktion Kontakt mit dem endgültigen Responder auf. Diese **getrennten ILL-Transaktionen** erfordern keine gemeinsam geteilte Identifikation der ILL-Transaktion und müssen auch nicht synchron ausgeführt werden. Jegliche Verbindung zwischen Ereignissen bezüglich einer bestimmten ILL-Transaktion und anderen ILL-Transaktionen ist nach freiem Belieben und unter der Kontrolle der beiden miteinander kommunizierenden Partner. Man spricht hier von einem „dual-role system“.

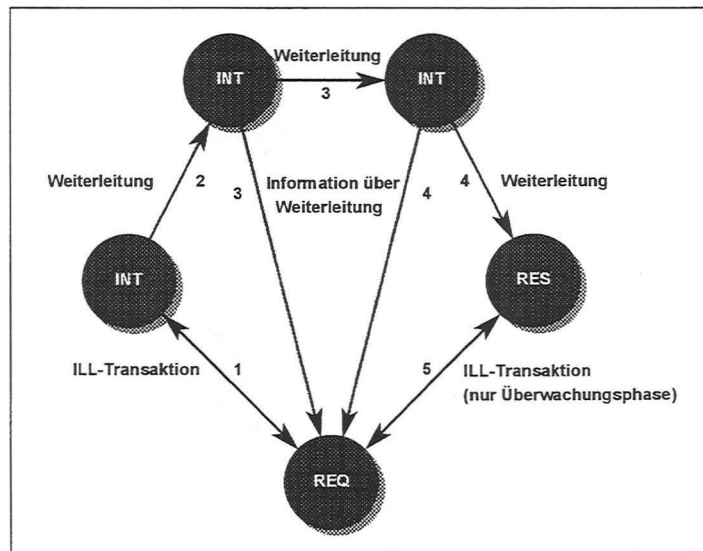
Die einzige Bedingung beim Gebrauch der getrennten ILL-Transaktionen ist, daß alle Leihobjekte über das „dual-role system“ versandt werden (Auslieferung und Rückgabe). Damit ist sichergestellt, daß alle involvierten Partner den Verlauf der zwei getrennten ILL-Transaktionen überwachen und terminierende Zustände¹² erreichen können.

Diese Art von ILL-Transaktion hat aber keinen Einfluss auf das ILL-Protokoll, da es sich um zwei getrennte, einfache ILL-Transaktionen (simple) handelt und wird im ILL-Standard auch nicht weiter beschrieben.

3.4.3.6 Weiterleitung (Forwarding)

Eine weitere Variation der einfachen ILL-Transaktion (simple) beteiligt einen Intermediary, der eine Fernleihanfrage an einen Responder weiterleitet und dann seine aktive Teilnahme an der ILL-Transaktion beendet. Der Responder antwortet dem Requester direkt, sobald er die weitergeleitete Fernleihanfrage erhalten hat. Der Intermediary gibt dem Requester Information darüber, daß er die Fernleihanfrage weitergereicht hat. Das Prinzip des Weiterleitens soll in der folgenden Abbildung anhand einer einfachen ILL-Transaktion (simple) mit drei Intermediaries gezeigt werden.

Abb. 3.6
Mehrfaches
Weiterleiten



Legende siehe Abschnitt 3.4.3

Der Requester kann die Weiterleitung gestatten oder verbieten und kann, falls dies gewünscht wird, in einer Liste selbst potentielle, eventuell auch präferierte Responder vorschlagen.

Zusätzlich hat der Requester die Möglichkeit in einer weiteren Liste diejenigen Responder an den Intermediary weiterzugeben, bei denen schon ein Leihversuch unternommen wurde. Mit diesem Vorgehen können unnötig doppelte Fernleihanfragen vermieden werden.

¹² Siehe Abschnitt 3.4.4.3.

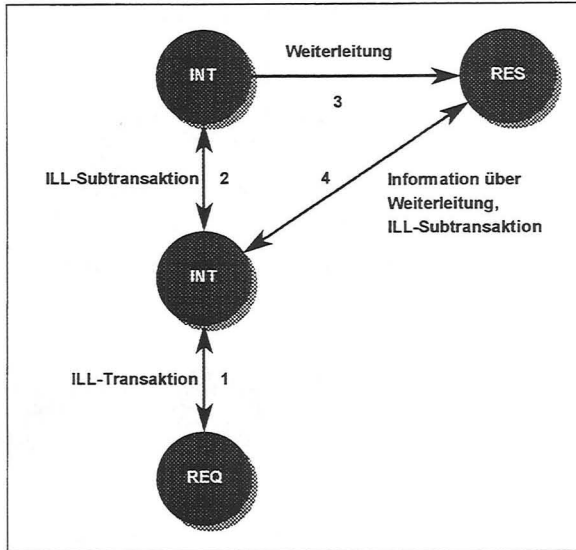
Das Inter-Library-Loan (ILL) - Protokoll

Sollte ein Responder nicht in der Lage sein, eine Fernleihanfrage erfolgreich zu erfüllen, kann er, um den Requester zu unterstützen, seinerseits eine Liste mit weiteren potentiellen Respondern liefern.

Verkettung und Weiterleitung

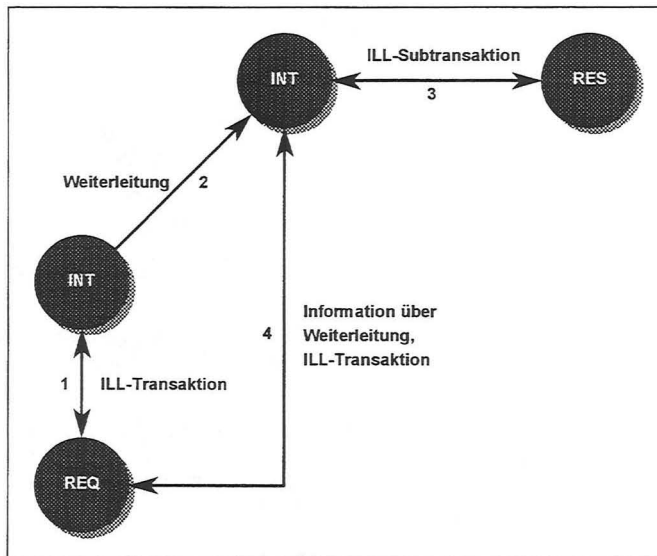
Die Verkettung und die Weiterleitung können innerhalb der gleichen ILL-Transaktion miteinander kombiniert werden. Die Abbildungen zeigen die beiden möglichen Fälle.

Abb. 3.7a
Weiterleitung nach Verkettung



Legende siehe Abschnitt 3.4.3

Abb. 3.7b
Verkettung nach Weiterleitung

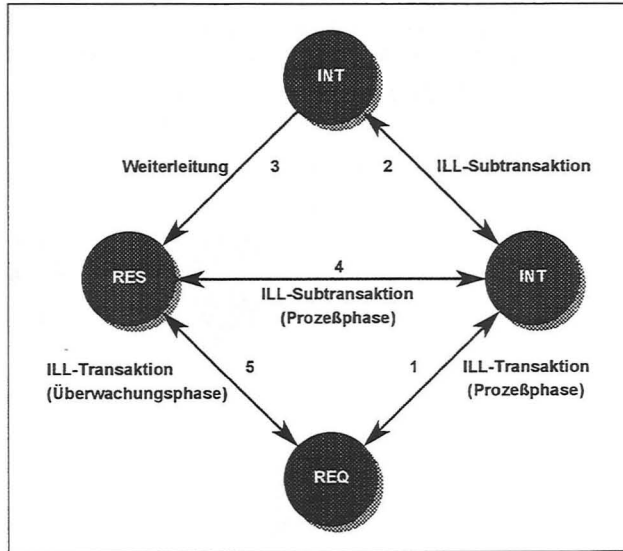


Legende siehe Abschnitt 3.4.3

Partitionierung und Weiterleitung

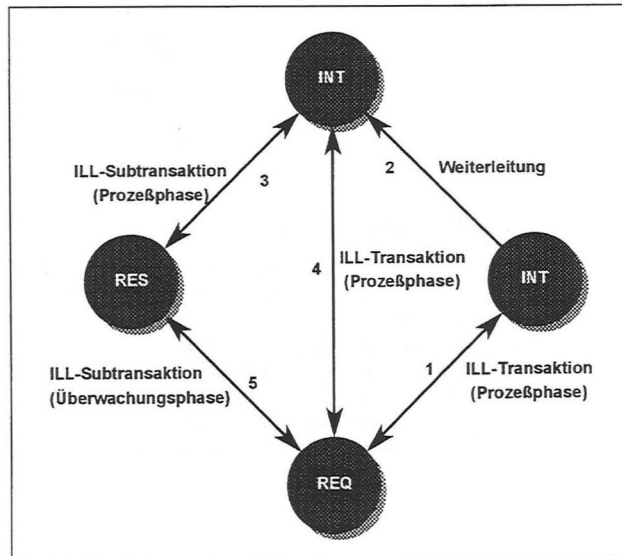
Die Partitionierung und die Weiterleitung können ebenfalls innerhalb der gleichen ILL-Transaktion miteinander kombiniert werden. Die Abbildungen zeigen die beiden möglichen Fälle.

Abb. 3.8a
Weiterleitung nach Partitionierung



Legende siehe Abschnitt 3.4.3

Abb. 3.8b
Partitionierung nach Weiterleitung



Legende siehe Abschnitt 3.4.3

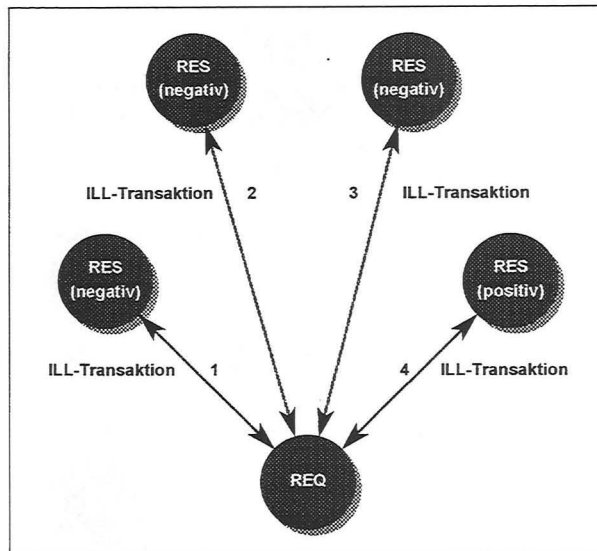
3.4.3.7 Einfache Sterntopologie (Referrals)

Endet eine ILL-Transaktion nicht mit der Erfüllung der Fernleihanfrage, kann der Requester eine neue ILL-Transaktion mit einem neuen Responder einleiten. Der Requester kann so der Reihe nach versuchen, einige potentielle Responder anzusprechen. Dies führt zur einfachen Sterntopologie, bei der der Requester stets Herr des Verfahrens bleibt. In diesem Fall wird jede neue Fernleihanfrage an einen neuen Responder als separate ILL-Transaktion betrachtet. Jede dieser Transaktionen ist Teil der gleichen ILL-Transaktionsgruppe.

Der Wechsel des Fernleihpartners könnte unter dem Gesichtspunkt der Implementierung manuell oder automatisch erfolgen.

Die folgende Abbildung verdeutlicht das Verfahren.

Abb. 3.9
Einfache
Sterntopologie



Legende siehe Abschnitt 3.4.3

3.4.3.8 Wiederholungen (Retries)

Wenn eine Fernleihanfrage mit der Begründung RETRY, ESTIMATE oder LOCATIONS-PROVIDED¹³ unerfüllt bleibt, beendet sich die ILL-Transaktion oder Subtransaktion. Der Requester oder der Intermediary können nun wählen, entweder die ursprüngliche Fernleihanfrage zu einem passenden Zeitpunkt zu wiederholen oder sich anderswo umzusehen. Falls die ursprüngliche Fernleihanfrage wiederholt wird, besitzt diese als eine entsprechende Markierung. Die Wiederholung ist eine neue Transaktion oder Subtransaktion, die der gleichen ILL-Transaktionsgruppe wie die ursprüngliche Fernleihanfrage angehört.

¹³ Siehe Abschnitt 3.5.6.

Für den Requester ist eine Wiederholung eine neue ILL-Transaktion. Deshalb muß sich die ILL-Transaktionsbezeichnung von der originären Fernleihanfrage unterscheiden. Die ILL-Transaktionsgruppenbezeichnung muß aus diesem Grund dieselbe bleiben, so daß es dem Responder oder Intermediary ermöglicht wird, die Wiederholung der vorhergehenden Transaktion zuzuordnen.

Für einen Intermediary ist eine Wiederholung eine neue ILL-Subtransaktion. Deshalb muß sich auch hier die ILL-Subtransaktionsbezeichnung von der der originären Fernleihanfrage unterscheiden. Hier muß allerdings sowohl die ILL-Transaktionsgruppenbezeichnung und die ILL-Subtransaktionsbezeichnung die gleiche sein, so daß es dem Responder oder weiteren Intermediary ermöglicht wird, die Wiederholung der vorhergehenden Subtransaktion zuzuordnen.

3.4.4 Globale ILL-Transaktionszustände

Zu jeder beliebigen Zeit werden die möglichen Interaktionen zwischen einem ILL-Dienstbenutzer (ILL service user) und einem ILL-Dienstanbieter (ILL service provider) durch den Zustand einer ILL-Transaktion genau bestimmt.

Requester-,
Responder- und
Intermediary-Zustand

Der Zustand einer ILL-Transaktion, d.h die Information, die das Stadium einer ILL-Transaktion beschreibt, ist die Kombination des **Requesterzustandes**, des **Responderzustandes** und der **Zustände aller Intermediaries**, die in die ILL-Transaktion mit einbezogen sind. Die Zustände der Requester, Responder und Intermediaries werden durch die Anwendungsinstanzen innerhalb der Endsysteme gehalten.

Damit man Systeme mit eingeschränkter Funktionalität oder Systeme, bei denen Kommunikationskosten minimiert werden müssen, einsetzen kann, beinhaltet das ILL-Protokoll optionale Nachrichten. Dies bedeutet, daß bestimmte Interaktionen, z.B. SHIPPED, RECEIVED, RETURNED und CHECKED-IN¹⁴, optional sind. Diese müssen also nicht zwangsläufig verwendet werden. Deshalb muß ein Dienstereignis in einem System nicht zu einem korrespondierenden Ereignis im anderen System führen.

Aus dem Zustand einer Anwendungsinstanz in einer ILL-Transaktion kann man folglich nicht auf den Zustand der anderen Anwendungsinstanz schließen. Jedoch stehen im ILL-Protokoll Dienste zur Verfügung, um den Zustand der anderen Anwendungsinstanz herauszufinden (STATUS-QUERY und STATUS-OR-ERROR-REPORT¹⁵). Um zu erfahren, welche Aktion gestattet ist, muß eine Anwendungsinstanz aber niemals den Zustand der anderen Anwendungsinstanz kennen. In Wirklichkeit spielen die globalen ILL-Transaktionszustände bei der Kontrolle des Verhaltens eines individuellen Endsystems eine untergeordnete Rolle. Vielmehr ist die lokale Darstellung des ILL-Transaktionszustandes von Bedeutung¹⁶, da diese eine detailliertere Überwachung der ILL-Transaktionen zuläßt. Der Detaillierungsgrad hängt dabei von der jeweiligen Implementierung einer ILL-Anwendung ab.

¹⁴ Siehe Abschnitte 3.5.5, 3.5.10, 3.5.12 und 3.5.13.

¹⁵ Siehe Abschnitte 3.5.20 und 3.5.21.

¹⁶ Siehe Abschnitt 9.

Obwohl der Versand von Nachrichten für einige Dienste optional sein kann, werden die Dienste selbst jedoch ausgeführt, da das lokale System den korrespondierenden Zustandswechsel vornimmt. Wenn z.B. ein Leihobjekt empfangen wurde, geht das lokale System in den Zustand RECEIVED¹⁷ über. Dies ist notwendig, um eine logische Sequenz von Ereignissen für eine ILL-Transaktion festzuhalten und zu kontrollieren.

3.4.4.1

Globale ILL-Transaktionszustände des Requesters

Der Zustand des Requesters ist der Bearbeitungszustand bei der nehmenden Bibliothek. Dieser kann einer der folgenden sein:

- **IDLE**

Die ILL-Transaktion hat noch nicht begonnen.

- **PENDING**

Eine Fernleihanfrage wurde eingeleitet und vom Responder wird nun ein Leihobjekt erwartet. Oder es wurde eine Nachricht empfangen, die angibt, daß ein Leihobjekt sofort geliefert wird, vorgemerkt wurde oder die Fernleihanfrage an eine andere Institution weitergeleitet wurde.

- **NOT-SUPPLIED**

Die ILL-Transaktion hat ein Stadium erreicht, bei dem die Fernleihanfrage durch den Responder nicht erfüllt werden kann.

- **CONDITIONAL**

Die ILL-Transaktion hat ein Stadium erreicht, bei dem der Fernleihwunsch nur erfüllt werden kann, wenn der Requester festgelegten Konditionen zustimmt.

- **CANCEL-PENDING**

Der Requester hat eine Stornierung der ILL-Transaktion eingeleitet, aber bisher vom Responder noch keine Antwort erhalten.

- **CANCELLED**

Die ILL-Transaktion wurde vom Responder storniert.

- **SHIPPED**

Das Leihobjekt wurde an den Requester verschickt.

- **RECEIVED**

Das Leihobjekt wurde vom Responder empfangen.

- **RENEW-PENDING**

Es wurde ein Antrag auf Leihfristverlängerung gestellt.

- **RENEW-OVERDUE**

Es wurde ein Antrag auf Leihfristverlängerung eines Leihobjektes gestellt, dessen Leihfrist schon abgelaufen ist.

¹⁷ Siehe Abschnitt 3.4.4.1.

- **OVERDUE**

Der Requester wurde benachrichtigt, daß die Leihfrist eines Objektes abgelaufen ist.

- **NOT-RECEIVED-OVERDUE**

Der Responder hat für ein Leihobjekt, das bislang noch nicht beim Requester eingetroffen ist, eine Mahnung wegen Ablauf der Leihfrist geschickt.

- **RECALL**

Das Leihobjekt wurde durch den Responder vorzeitig zurückgerufen.

- **RETURNED**

Das Leihobjekt wurde an den Responder zurückgeschickt.

- **LOST**

Das Leihobjekt ging verloren.

3.4.4.2

Globale ILL-Transaktionszustände des Responders

Entsprechend ist der Zustand des Responders der Bearbeitungszustand bei der gebenden Bibliothek. Dieser kann einer der folgenden sein:

- **IDLE**

Der Responder hat noch keine Fernleihanfrage empfangen.

- **IN-PROCESS**

Eine Fernleihanfrage wurde empfangen und wird nun vom Responder bearbeitet. Das Leihobjekt wurde aber noch nicht versandt.

- **FORWARD**

Die Fernleihanfrage wurde an eine andere Institution weitergeleitet.

- **NOT-SUPPLIED**

Der Responder hat auf eine Fernleihanfrage mit einer ILL-ANSWER (RETRY, UNFILLED, LOCATIONS-PROVIDED oder ESTIMATE¹⁸) geantwortet oder die ILL-Transaktion ist abgelaufen (EXPIRY¹⁹).

- **CONDITIONAL**

Die Fernleihanfrage kann nur erfüllt werden, wenn der Requester den angegebenen Bedingungen zustimmt.

- **CANCEL-PENDING**

Der Requester hat eine Stornierung der ILL-Transaktion eingeleitet, der Responder hat darauf aber noch nicht geantwortet.

- **CANCELLED**

Die ILL-Transaktion wurde durch den Responder storniert.

¹⁸ Siehe Abschnitt 3.5.6.

¹⁹ Siehe Abschnitt 3.5.21.

- **SHIPPED**
Das Leihobjekt wurde an den Requester verschickt.
- **RENEW-PENDING**
Es wurde ein Antrag auf Leihfristverlängerung gestellt.
- **RENEW-OVERDUE**
Es wurde ein Antrag auf Leihfristverlängerung eines Leihobjektes gestellt, dessen Leihfrist schon abgelaufen ist.
- **OVERDUE**
Der Responder hat den Requester darüber informiert, daß die Leihfrist eines Objektes abgelaufen ist.
- **RECALL**
Das Leihobjekt wurde durch den Responder vorzeitig zurückgerufen.
- **CHECKED-IN**
Das Leihobjekt wurde vom Requester zurückerhalten.
- **LOST**
Das Leihobjekt ging verloren.

Anmerkung:

Die hier definierten Zustände sind diejenigen, die in einer OSI-Umgebung von Bedeutung sind, d.h. von Bedeutung für mehr als ein Endsystem. Jede Implementierung kann mehr Informationen und Zustände besitzen, als in diesem Standard definiert sind.

Wenn es zum Beispiel ein Responder wünscht, zwischen Leihobjekten, deren Leihfrist noch nicht verlängert wurde, und Leihobjekten, deren Leihfrist schon einmal verlängert wurde, zu unterscheiden, dann könnte er die lokalen Zustände SHIPPED und RENEW, die beide zum Dienstzustand SHIPPED in Beziehung stehen, verwenden. Das System des Requesters bemerkt dabei nicht, daß durch die lokale Implementierung der einfache Zustand SHIPPED zum Zweck einer höheren lokalen Kontrolle der ILL-Transaktion in zwei lokale Zustände aufgeteilt wurde.

Die Zustandsberichte (STATUS-OR-ERROR-REPORT, STATUS-QUERY²⁰) liefern aber nur Zustandswerte, die im ILL-Standard definiert sind.

3.4.4.3

Terminierende
Zustände

Terminierende globale ILL-Transaktionszustände

Für einen Requester, einen Responder oder einen Intermediary gibt es bestimmte Zustände, in denen es keinen weiteren Zustandswechsel mehr geben kann. Solche Zustände heißen **terminierende Zustände**.

Eine ILL-Transaktion wird normalerweise für eine bestimmte Zeitdauer, bevor die Information der ILL-Transaktion für ein Partnersystem unzugänglich ist, in einem terminierenden Zustand bleiben. Andernfalls wird die ILL-Transaktion gleich gelöscht. Die Größe der Zeitdauer ist eine Entscheidung des lokalen Managements oder wird den Abmachungen von Implementierern unterzogen.

²⁰ Siehe Abschnitte 3.5.21 und 3.5.20.

Diese Zeitdauer muß allerdings ausreichend groß sein, damit die Information für die maximale Leihfrist plus einer Leihfristverlängerung und der Lieferzeit erhalten bleibt.

Für Leihobjekte, die nicht zurückgegeben werden müssen, muß die Zeitdauer ausreichend sein, den Requester feststellen zu lassen, daß ein erwartetes Leihobjekt nicht empfangen wurde. Dann müßte nämlich der ILL-Dienst STATUS-QUERY oder LOST²¹ eingeleitet werden.

Terminierende
Zustände Requester

Die möglichen terminierenden Zustände für den Requester sind:

- **NOT-SUPPLIED**
- **CANCELLED**
- **RECEIVED**

Wenn ein Leihobjekt empfangen wurde, das nicht zurückgegeben werden muß.

- **RETURNED**
- **LOST**

Terminierende
Zustände Responder

Die möglichen terminierenden Zustände für den Responder sind:

- **NOT-SUPPLIED**
- **CANCELLED**
- **FORWARD**
- **SHIPPED**

Wenn ein Leihobjekt verschickt wurde, das nicht zurückgegeben werden muß.

- **CHECKED-IN**
- **LOST**

Terminierende
Zustände Inter-
mediary

Die möglichen terminierenden Zustände für den Intermediary sind:

- **NOT-SUPPLIED**
- **FORWARD**
- **SHIPPED**
- **CANCELLED**

Der terminierende Zustand für eine bestimmte ILL-Transaktion wird von seinen Umständen abhängen. Wenn zum Beispiel eine Photokopie bereitgestellt wird, ist für den Responder der Zustand SHIPPED, für den Requester der Zustand RECEIVED, der terminierende Zustand.

3.4.4.4

Globale ILL-Zustände des Intermediary

Ein Intermediary, der in eine verkettete oder partitionierte ILL-Transaktion verwickelt ist, spielt sowohl die Rolle eines Responders (aus der Sicht des Requesters), als auch die Rolle eines Requesters (aus der Sicht des Responders).

²¹ Siehe Abschnitte 3.5.20 und 3.5.17.

Der Intermediary unterhält getrennte Zustandsinformationen für jede dieser getrennten Zustandsmengen. In der Rolle des Requesters ist für einen Intermediary der Zustand SHIPPED ein terminierender Zustand, weil keine weitere Nachricht empfangen wird, um einen Zustandswechsel zu verursachen.

Um sicherzustellen, daß der Zustand SHIPPED innerhalb des Intermediary erreicht wurde, ist die Nachricht SHIPPED²² für verkettete und partitionierte ILL-Transaktionen obligatorisch.

Wenn sich ein Intermediary einer verketteten ILL-Transaktion einmal in einem terminierendem Zustand befindet, gibt er lediglich Nachrichten, die vom Responder empfangen wurden, an den Requester weiter und umgekehrt.

3.4.4.5

Definition von Zustandsübergängen der globalen ILL-Transaktionszustände

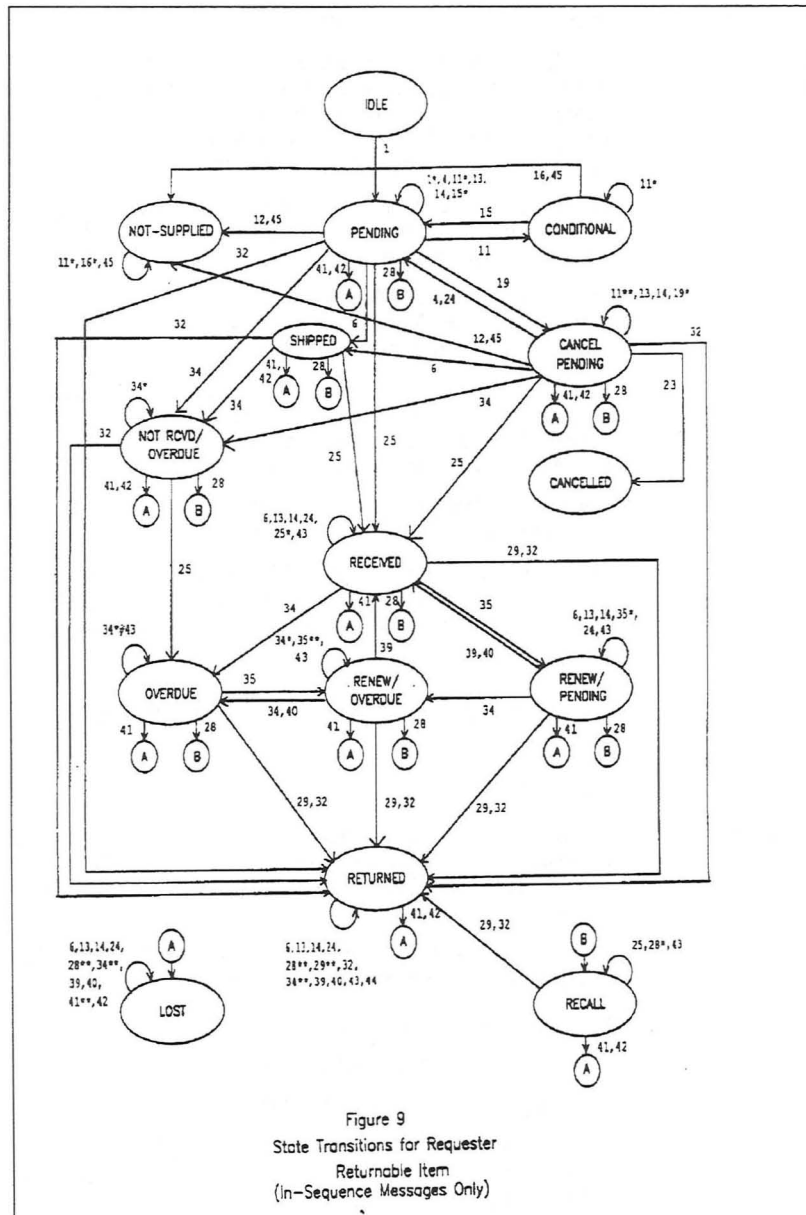
Aufgrund des Umfanges der Diplomarbeit soll an dieser Stelle auf die ISO-Dokumente [ISO 10160, 1991, S. 96 ff.] und [ISO 10161, 1991, S. 111 ff.] verwiesen werden, in denen alle möglichen Zustandswechsel mit den dafür verantwortlichen ILL-Diensten in Form von Zustandsgraphen und Zustandstabellen detailliert aufgeführt sind.

Die folgende Abbildung zeigt als Auszug aus den ISO-Dokumenten beispielhaft einen Zustandsgraph für einen Requester, der die Ausleihe eines „returnable item“ wünscht. Zur Beschreibung der aufgeführten ILL-Dienste und der Unterscheidung beim Dienstaufruf nach „request“ und „indication“ siehe Abschnitt 3.5. Bei diesem Zustandsgraph sind den Zahlen an den Kanten folgende ILL-Dienstaufrufe zugeordnet:

- | | | |
|-----|-----------------------------------|------------------------|
| 1. | ILL-REQUEST request | |
| 4. | FORWARD-NOTIFICATION indication | |
| 6. | SHIPPED indication | |
| 11. | ILL-ANSWER indication | - CONDITIONAL |
| 12. | ILL-ANSWER indication | - UNFILLED |
| | | - ESTIMATE |
| | | - RETRY |
| | | - LOCATIONS-PROVIDED |
| 13. | ILL-ANSWER indication | - WILL-SUPPLY |
| 14. | ILL-ANSWER indication | - HOLD-PLACED |
| 15. | CONDITIONAL-REPLY request | - YES |
| 16. | CONDITIONAL-REPLY request | - NO |
| 19. | CANCEL request | |
| 23. | CANCEL-REPLY indication | - YES |
| 24. | CANCEL-REPLY indication | - NO |
| 25. | Leihobjekt erhalten | |
| | RECEIVED request | |
| 28. | RECALL indication | |
| 29. | Leihobjekt wieder zurückgeschickt | |
| | RETURNED request | |
| 32. | CHECKED-IN indication | 41. LOST request |
| 34. | OVERDUE indication | 42. LOST indication |
| 35. | RENEW request | 43. DAMAGED request |
| 39. | RENEW-ANSWER indication | 44. DAMAGED indicaton |
| 40. | RENEW-ANSWER indication | 45. EXPIRED indication |
| | | - YES |
| | | - NO |

²² Siehe Abschnitt 3.5.5.

Abb. 3.10
ILL-Zustandsübergänge des Requesters bei einem 'returnable item'



Legende zur Abbildung (ISO 10160, 1991, S. 98):

- Eine Zahl ohne zusätzliche Markierung entspricht einem ursprünglicher Dienstaufwurf.
- Eine Zahl mit einem Stern (*) als Markierung entspricht einem wiederholten Dienstaufwurf des gleichen Typs.
- Eine Zahl mit zwei Sternen (**) als Markierung entspricht entweder einem ursprünglichen oder einem wiederholten Dienstaufwurf.

3.4.4.6 Phasen einer ILL-Transaktion

Wie bereits erwähnt, besteht eine ILL-Transaktion aus 2 Phasen - der Prozeßphase (process phase) und der Überwachungsphase (tracking phase). Die Prozeßphase ist für alle ILL-Transaktionen obligatorisch, während die Überwachungsphase nur bei ILL-Transaktionen anwendbar ist, bei denen ein ausgeliefertes Leihobjekt zurückgegeben werden muß (z.B. eine Monographie).

Die Prozeßphase für den Requester enthält alle Ereignisse und Aktionen bis einschließlich des Empfangs des gewünschten Leihobjektes. Diese Phase endet normalerweise im Zustand RECEIVED.

Die Prozeßphase für den Responder enthält alle Ereignisse und Aktionen bis einschließlich des Versandes des gewünschten Leihobjektes. Diese Phase endet normalerweise im Zustand SHIPPED.

Für den Intermediary als Requester enthält die Prozeßphase alle Ereignisse und Aktionen bis einschließlich des Empfangs der SHIPPED-Indikation²³. Für den Intermediary als Responder hingegen enthält die Prozeßphase alle Ereignisse und Aktionen bis einschließlich der Ausgabe des SHIPPED-Request.

Sowohl für den Intermediary als Requester wie auch als Responder endet die Prozeßphase normalerweise im Zustand SHIPPED.

Die Überwachungsphase schließlich enthält alle Ereignisse und Aktionen nach dem Versand und Empfang eines Leihobjektes, das zurückgegeben werden muß, einschließlich Leihfristverlängerung, Ablauf von Leihfristen und Rückgabe des Leihobjektes.

3.5 Definition der Dienste des ILL-Protokolls

3.5.1 Allgemeines

Dienstaufruf

Die Interaktion zwischen den an einer ILL-Transaktion teilnehmenden Anwendungsinstanzen (application entities) wird mit **Dienstaufrufen** beschrieben, welche durch ein ILL-Anwendungsdienstelement (application service element) bereitgestellt werden.

Die Datenfelder der ILL-APDU (Application Protocol Data Unit²⁴) jedes ILL-Dienstes sind von der ISO in folgende Typen unterteilt:

- Pflichtfeld (mandatory oder „M“)
- optionales Feld (user option oder „U“)
- bedingtes Feld (conditional oder „C“)

Diese Felder sind unter gewissen Umständen Pflichtfelder, ansonsten sind es optionale Felder. Die Umstände kommen in der Beschreibung des jeweiligen Dienstes zum Ausdruck.

²³ Siehe Abschnitt 3.5.

²⁴ Siehe Abschnitt 4.2.

- nicht anwendbares Feld (not applicable oder „-“)

Dieser Typ ist notwendig, da man bei einigen ILL-Diensten zusätzlich zu deren normaler Funktionalität Hinweise auf etwas geben kann. Will man beim Dienstaufwurf jedoch keine zusätzlichen Hinweise geben, dann dürfen die Hinweisfelder nicht belegt werden. Bei diesen Feldern handelt es sich dann um nicht anwendbare Felder.

Flexible ILL-Dienste:
„Request“ oder
„Indication“

Um Funktionalität und Flexibilität des ILL-Dienstes zu erhöhen, kann man, wie bereits erwähnt, beim Dienstaufwurf zusätzliche Hinweise geben. Im ILL-Protokoll sind aber auch Dienste definiert, die „nur“ auf etwas hinweisen sollen. Die ISO definiert dafür bei gewissen ILL-Diensten die Aufrufmöglichkeiten „Request“ und „Indication“.

Die Wahl, ob es sich beim Dienstaufwurf um einen der Art „Request“ oder der Art „Indication“ handelt, wird getroffen, indem man entweder alle Datenfelder der dem Dienst zugehörigen ILL-APDU benutzt (indication) oder die Hinweisfelder ausklammert (request). Der Empfänger erkennt somit an den belegten Datenfeldern, um welche Art von Dienstaufwurf es sich gehandelt hat.

Im ISO-Dokument [ISO 10160, 1991] werden alle im ILL-Protokoll vorgesehenen Dienste nach folgendem Schema beschrieben:

- **Funktion**

Verbale Beschreibung, welche Funktionalität der Dienst besitzt.

- **Parameterliste**

Hier werden alle Datenfelder der zum Dienst zugehörigen ILL-APDU in Form einer Tabelle aufgelistet. Die Tabelle beinhaltet Name und o.g. Typ jedes einzelnen Datenfeldes. Der Typ wird dabei nochmals nach „Request“ und „Indication“ gegliedert, falls das für diesen Dienst unterstützt wird. Anschließend werden die Datenfelder der Reihe nach beschrieben. Besteht ein Datenfeld selbst wiederum aus einer Anzahl weiterer Datenfelder (geschachtelte Struktur), werden auch diese beschrieben. Die Beschreibung der Datenfelder beschränkt sich jedoch nur auf Funktion und Inhalt, geht aber nicht auf Datentypen oder ähnliches ein.

Beispielhaft wird in der folgenden Definition der einzelnen ILL-Dienste der Dienst „Fernleihanfrage (ILL-REQUEST)“ gemäß dem ISO-Dokument ISO 10160 ausführlich beschrieben.

Die nachfolgenden Definitionen der im ILL-Protokoll vorgesehenen ILL-Dienste beschränken sich dann aus Gründen des Umfangs der Diplomarbeit auf deren Funktionalität. Für die detailliertere Beschreibung wird wieder auf die ISO-Dokumente [ISO 10160, 1991, S.40 ff.] und [ISO 10161, 1991, S.27 ff.] verwiesen. Weitere ausführlichere Beschreibungen der im Konstanzer Fernleihsystem implementierten ILL-Dienste sind auch in Kapitel 5 enthalten.

3.5.2 Fernleihanfrage (ILL-REQUEST)

Versender:

Requester

Funktion:

Die „Fernleihanfrage“ erlaubt einem Requester, ein Leihobjekt bei einem Responder zu bestellen. Falls das Leihobjekt nicht zur Verfügung gestellt werden kann, hat der Responder die Möglichkeit, negativ zu antworten, die Fernleihanfrage an eine andere Institution weiterzuleiten oder eine verkettete oder partitionierte ILL-Subtransaktion mit einem anderen Responder zu initiieren.

Verkettete oder partitionierte ILL-Subtransaktionen werden durch den Aufruf eines ILL-Requests mit einem Transaktionsidentifizier initiiert, der vom ursprünglichen Transaktions-Identifizier abgeleitet wird.

Die Angaben, die vom Requester geliefert werden, beinhalten Informationen zur Identifizierung des Leihobjektes (z.B. Signatur), sowie Autor und Titel, Angaben dazu, wann und von wem das Leihobjekt gebraucht wird, den Bestimmungsort und ob das Original selbst oder eine Photokopie erwünscht ist.

Eine Fernleihanfrage erlaubt es dem Requester weiter, eine Kostenabschätzung für die Bereitstellung des Leihobjektes, den Standort eines Leihobjektes oder die Vormerkung eines Leihobjektes zu erbitten, falls dieses im Moment nicht verfügbar sein sollte. Man spricht dabei von verschiedenen ILL-Auftragstypen.

Ein potentieller Responder muß zumindest entweder die Ausleihe von Objekten oder die Lieferung von Kopien unterstützen. Die Unterstützung von Standortnachweisen und Kostenabschätzungen hingegen ist optional.

Der Requester zeigt in seiner Fernleihanfrage außerdem an, ob er in der Lage ist, die ILL-Dienste RECEIVED²⁵ und RETURNED²⁶ zu versenden und ob er vom Responder die Versendung der ILL-Dienste SHIPPED²⁷ und CHECKED-IN²⁸ fordert.

²⁵ Siehe Abschnitt 3.5.10.

²⁶ Siehe Abschnitt 3.5.12.

²⁷ Siehe Abschnitt 3.5.5.

²⁸ Siehe Abschnitt 3.5.13.

Parameterliste:

Parametername	Request	Indication
transaction identification	M	M
service date and time	M	M
requester identification	C	C
responder identification	C	C
transaction type	U	U
delivery adress	U	U
delivery service	U	U
billing adress	U	U
ILL service type	M	M
responder-specific service	U	U
requester optional messages	M	M
search type	U	U
supply medium information	U	U
place on hold	U	U
client identification	U	U
item identification	M	M
supplemental item description	U	U
cost information	U	U
copyright compliance	U	U
third party information	C	C
retry flag	U	U
forward flag	-	U
requester note	U	U
forward note	-	U

M = Pflicht, U = optional, C = bedingt, - = nicht anwendbar

Beschreibung der
Datenfelder:

- **transaction identification**

Information, die eine ILL-Transaktion eindeutig identifiziert. Im Umfeld eines Intermediary handelt es sich um eine ILL-Subtransaktion.

Das Datenfeld beinhaltet die folgenden Komponenten:

- „initial-requester identification“,
- einen „ILL-transaction group qualifier“ für verwandte ILL-Transaktionen,
- einen „ILL-transaction qualifier“ für alle Meldungen, die mit der ILL-Transaktion in Verbindung stehen,
- einen „ILL-subtransaction qualifier“, falls es sich um eine ILL-Subtransaktion handelt.

Die „initial-requester identification“, „ILL-transaction group qualifier“ und „ILL-transaction qualifier“ werden vom Requester unterstützt. Die „initial-requester identification“ und der „ILL-subtransaction qualifier“ vom Intermediary.

- **service date and time**

Datum und Uhrzeit, zu der der ILL-Dienst aufgerufen wurde. Das Datenfeld beinhaltet zwei Komponenten:

- Datum und Uhrzeit, zu der der aktuelle Dienst aufgerufen wurde,
- Datum und Uhrzeit, zu der der ursprüngliche Dienst aufgerufen wurde.

Die zweite Komponente wird benutzt, um Wiederholungen des Dienstaufrufs zu identifizieren.

- **requester identification**

Dieses Feld beinhaltet Informationen, die den Requester identifizieren. In einer verketteten oder partitionierten ILL-Transaktion wird dadurch der Intermediary identifiziert, der die ILL-Subtransaktion eingeleitet hat. Das Datenfeld besteht aus einer oder mehreren der folgenden Komponenten:

- „person-or-institution-symbol“,
- „name-of-person-or-institution“.

Diese Identifikation muß dabei innerhalb des Kommunikationsbereiches der Fernleihpartner eindeutig sein, z.B. innerhalb eines Landes.

Die Verwendung dieses Datenfelds ist optional, wenn es sich um eine verbindungsorientierte Kommunikation handelt, und vorgeschrieben, wenn es sich um eine „store-and-forward“-Kommunikation handelt.

- **responder identification**

Dieses Feld beinhaltet Informationen, die den Responder identifizieren. Die Komponenten und die Verwendung dieses Feldes sind dieselben wie beim Datenfeld „requester identification“ (siehe oben).

- **transaction type**

Durch dieses Feld wird der ILL-Transaktionstyp identifiziert und kann dabei folgende symbolische Werte annehmen:

- einfach (Punkt-zu-Punkt),
- verkettet,
- partitioniert.

- **delivery address**

Dieses Feld beschreibt die konventionelle oder die elektronische Adresse, an die das bestellte Leihobjekt ausgeliefert werden soll.

Die konventionelle Adresse besteht dabei aus den folgenden Komponenten:

- „name-of-person-or-institution“,
- „extended-postal-delivery-address“,
- Straße und Nummer,
- Postfach,
- Stadt,
- Region,
- Land,
- Postleitzahl.

Die elektronische Adresse bzw. Systemadresse besteht aus einer oder mehreren der folgenden Komponenten:

- „telecom-service-identifier“,
- „telecom-service-address“.

Der „telecom-service-identifier“ unterscheidet die bei Requester und Responder zu Verfügung stehenden Telekommunikations-Dienste. Er braucht nicht mit einem Wert belegt zu werden, wenn die „telecom-service-address“ nicht eindeutig ist. Die „telecom-service-address“ identifiziert die spezifische Adresse des erkannten Telekommunikations-Dienstes. Sie betrifft die Auslieferung des bestellten Leihobjektes und ist nicht mit der Adresse, die für die ILL-Kommunikation verwendet wird, zu verwechseln.

- **delivery service**

In diesem Feld kann der Name oder ein Code für den Auslieferungs-Dienstleister angegeben werden. Es kann hier aber auch stattdessen die Auslieferungsart angegeben werden (Post, Bücherauto etc.).

- **billing address**

Dieses Feld beinhaltet die konventionelle oder die elektronische Adresse, an die die Rechnung (Fernleihgebühren) geschickt wird. Die Komponenten sind dieselben wie beim Datenfeld „delivery address“.

- **ILL service type**

Durch dieses Feld wird der gewünschte ILL-Auftragstyp festgelegt. Hier kann man auch eine sog. Präferenzliste angeben. Die folgenden ILL-Auftragstypen werden unterstützt:

- Ausleihe,
- nur Photokopien (keine Rückgabe),
- nur Standortnachweise,
- Ausleihkosten-Abschätzung,
- Responder-spezifischer Auftragstyp.

- **responder-specific service**

Hier kann ein Responder-spezifischer Dienst angegeben werden, der von diesem unterstützt wird und nicht im ILL-Standard definiert ist.

- **requester optional messages**

Dieses Datenfeld gibt an, ob der Requester den Versand der ILL-Dienste RECEIVED und RETURNED unterstützt und ob die ILL-Dienste SHIPPED und / oder CHECKED-IN²⁹ erforderlich bzw. vom Responder gewünscht sind.

- **search type**

In diesem Datenfeld können Terminbedingungen für den Erhalt des Leihobjektes angegeben werden. Es besteht aus einer oder mehreren der folgenden Komponenten:

- „level-of-service“,
- „need-before-date“,
- „expiry-flag“,
- „expiry-date“.

- **supply medium information**

Dieses Datenfeld kann man benutzen, um das Medium zu beschreiben, in dem das gewünschte Leihobjekt geliefert werden soll. Es kann auch in Form einer Präferenzliste angegeben werden. Die folgenden Medientypen werden unterstützt:

- gedruckt,
- photokopiert,
- „microform“,
- Film oder Video-Aufnahme,
- Audio-Aufnahme,
- in maschinenlesbarer Form,
- andere.

²⁹ Siehe Abschnitt 3.5.13.

- **place on hold**

Dieses Feld dient als Hinweis des Requesters, daß das gewünschte Leihobjekt beim Responder vorgemerkt werden soll, falls es im Moment nicht ausgeliehen werden kann (Leihobjekt ist schon ausgeliehen oder ist gerade nicht auffindbar).

- **client identification**

Mit diesem Feld wird der Benutzer (Person oder Institution) beschrieben, der das Leihobjekt haben will. Es besteht aus einer oder mehreren der folgenden Komponenten:

- Benutzername,
- Benutzerstatus,
- Benutzer-Identifikation (z.B. Benutzernummer etc.).

- **item identification**

In diesem Datenfeld wird die gesamte bibliographische Information über das Leihobjekt vom Requester eingetragen. Es besteht aus einer oder mehreren der folgenden Komponenten:

- „item type“ (Monographie, Band oder andere),
- „held medium type“ (Medium, in dem der Requester vermutet, daß das Leihobjekt beim Responder zur Verfügung steht.),
- „call number“,
- Autor,
- Titel,
- Untertitel,
- „sponsoring body“,
- Erscheinungsort,
- Verlag,
- Serientitel oder -nummer,
- „volume and issue“,
- „edition“,
- Erscheinungsdatum,
- „publication date of component“,
- Autor des Artikels,
- Titel des Artikels,
- Seitenangabe,
- „national bibliography number“,
- ISBN,
- ISSN,

- „system number“,
- „additional numbers or letters“,
- „verification and / or reference source“.

- **supplemental item description**

In diesem Datenfeld können zusätzliche Informationen angegeben werden, die das Leihobjekt beschreiben. Diese können in einem maschinenlesbaren Format angegeben sein. Das Feld kann vom ursprünglichen Requester belegt werden oder es wird eventuell später von einem Responder oder einem Intermediary nachgetragen (z.B. als Resultat des Bibliographierens).

- **cost information**

Hier können Angaben zu den Fernleihkosten gemacht werden. Das Datenfeld besteht aus einer oder mehreren der folgenden Komponenten:

- „account number of the requester“,
- maximal akzeptierte Fernleihkosten,
- „the existence of reciprocal cost agreement“,
- generelle Bereitschaft, Gebühren zu zahlen,
- Hinweis, daß man auf die Bezahlung von Fernleihkosten gefaßt ist.

- **copyright compliance**

Dieses Datenfeld wird vom Requester benutzt, um anzugeben, an welche Copyright-Bestimmungen bzw. Gesetze er sich zu halten hat.

- **third party information**

In diesem Datenfeld können Informationen angegeben werden, die für solche ILL-Transaktionen relevant sind, an denen mehr als zwei Fernleihpartner teilnehmen. Dies ist z.B. beim Weiterleiten, Verketten oder Partitionieren der Fall. Es besteht aus einer oder mehreren der folgenden Komponenten:

- Erlaubnis weiterzuleiten,
- Erlaubnis zu verketten,
- Erlaubnis zu partitionieren,
- Erlaubnis, die Liste mit potentiellen Respondern zu ändern,
- Adresse des ursprünglichen Requesters,
- Hinweis, ob die Liste mit potentiellen Respondern nach Präferenzen sortiert ist oder nicht,
- Liste mit potentiellen Respondern,
- Liste mit Respondern, bei denen schon ein Leihversuch unternommen wurde.

Die Verwendung dieses Datenfeldes ist vorgeschrieben, wenn der ILL-Dienst ILL-REQUEST für eine partitionierte ILL-Subtransaktion eingeleitet wird.

Es ist auch für den Fall vorgeschrieben, daß ein ILL-REQUEST für eine ILL-Subtransaktion eingeleitet wurde. Dabei muß der empfangene ILL-REQUEST eine Liste mit Respondern enthalten, bei denen schon ein Leihversuch unternommen wurde. Andernfalls ist die Verwendung des Datenfeldes optional.

- **retry flag**

Hier kann der Requester einen Hinweis darauf geben, daß die ILL-Transaktion oder ILL-Subtransaktion ein wiederholter Versuch ist oder daß es der erste Versuch ist.

- **forward flag**

In diesem Datenfeld wird vom ILL-Dienstanbieter angemerkt, ob ein empfangener ILL-REQUEST von einem Intermediary weitergeleitet worden ist.

- **requester note**

Dieses Datenfeld wird vom Requester benutzt, um zusätzliche Informationen einzutragen, die durch keinen anderen ILL-Dienst abgedeckt werden.

- **forward note**

In diesem letzten Datenfeld kann ein Intermediary zusätzliche Informationen unterbringen, wenn dieser den ILL-REQUEST weiterleitet.

Bemerkung:

Der Dienst „ILL-REQUEST“ muß in einer ILL-Anwendung unterstützt werden.

3.5.3

Weiterleitung einer Fernleihanfrage (FORWARD)

Versender:

Responder

Funktion:

Dieser Dienst erlaubt es einem Responder, eine Fernleihanfrage an eine andere Institution, die durch den Responder ausgewählt oder durch den Requester vorgegeben wurde, weiterzuleiten.

Dieser Dienst wird anstelle der ILL-ANSWER³⁰ als Antwort auf eine empfangene Fernleihanfrage verwendet. Ist das Weiterleiten abgeschlossen, nimmt der Responder nicht weiter an der ILL-Transaktion teil. Jedoch können auch dann noch bestimmte Dienste, wie z.B. STATUS-QUERY³¹, STATUS-OR-ERROR-REPORT³² und MESSAGE³³, benutzt werden.

Der Responder, an den eine Fernleihanfrage weitergereicht wird, kann dem Requester antworten oder, falls erlaubt, die Fernleihanfrage an eine andere Institution weiterleiten.

In der Regel pflegt der Responder eine Fernleihanfrage weiterzuleiten, da er dadurch dem Requester einen Gefallen erweist. Beispielsweise könnte eine Bibliothek eine Fernleihanfrage an einen Zentralkatalog (ZK) versenden, der dann den Standort des Leihobjektes bestimmt und die Fernleihanfrage an die besitzende Bibliothek weiterleitet.

³⁰ Siehe Abschnitt 3.5.6.

³¹ Siehe Abschnitt 3.5.20.

³² Siehe Abschnitt 3.5.21.

³³ Siehe Abschnitt 3.5.19.

Das Inter-Library-Loan (ILL) - Protokoll

Der Intermediary, in diesem Fall der ZK, zieht sich dann aus der ILL-Transaktion zurück und alle Interaktionen werden ab diesem Zeitpunkt nur noch zwischen dem ursprünglichen Requester und der besitzenden Bibliothek abgewickelt.

Bemerkung: Dieser Dienst kann, wenn es der Requester so wünscht, auch unterdrückt werden.

3.5.4 Weiterleitungsanzeige (FORWARD-NOTIFICATION)

Versender: Intermediary ⇒ Automatische Generierung beim Weiterleiten einer Fernleihanfrage

Funktion: Ein Requester, der dem Empfänger einer Fernleihanfrage die Erlaubnis zur Weiterleitung des Auftrages erteilt hat, erhält bei jedem Weiterleiten eine Nachricht des Typs FORWARD-NOTIFICATION, in der der Name der weiterleitenden und der Name der nächsten Empfängerbibliothek angezeigt werden. Der Requester weiß somit zu jedem Zeitpunkt, wo sich eine bestimmte Fernleihanfrage befindet.

3.5.5 Versand (SHIPPED)

Versender: Responder

Funktion: Mit diesem Dienst teilt der Responder dem Requester mit, daß das gewünschte Leihobjekt (konventionell oder elektronisch) versendet wurde.

Bemerkung: Der Dienst „SHIPPED“ muß in einer ILL-Anwendung unterstützt werden.

3.5.6 ILL-Antwort (ILL-ANSWER)

Versender: Responder

Funktion: Die „ILL-Antwort“ erlaubt einem Responder, auf eine empfangene Fernleihanfrage zu antworten. Es sind dabei die folgenden Antworten möglich:

- **CONDITIONAL**

Das gewünschte Leihobjekt ist zwar verfügbar, der Fernleihanfrage kann aber nur stattgegeben werden, wenn der Requester bestimmten Ausleihkonditionen, die in der Antwort angegeben werden, zustimmt.

- **RETRY**

Das gewünschte Leihobjekt ist derzeit nicht verfügbar, aber es wird ein Datum angegeben, zu dem das Leihobjekt voraussichtlich verfügbar sein wird.

- **UNFILLED**

Das gewünschte Leihobjekt ist nicht verfügbar.

- **LOCATIONS-PROVIDED**

Institutionen, die das gewünschte Leihobjekt besitzen, werden mitgeteilt.

- **WILL-SUPPLY**

Das gewünschte Leihobjekt ist verfügbar und wird schnellstmöglich verschickt. Ein Datum, wann das Leihobjekt ungefähr verschickt wird, kann optional angegeben werden.

- **HOLD-PLACED**

Das gewünschte Leihobjekt ist nicht sofort verfügbar, wird aber für den Requester gemäß seinen Anweisungen vorgemerkt.

- **ESTIMATE**

Die geschätzten Kosten für die Lieferung des gewünschten Leihobjektes werden als Antwort auf den entsprechenden ILL-Request bereitgestellt.

Bemerkung: Der Dienst „ILL-ANSWER“ muß in einer ILL-Anwendung unterstützt werden.

3.5.7 Bedingte Antwort (CONDITIONAL REPLY)

Versender: Requester

Funktion: Mit der „bedingten Antwort“ akzeptiert der Requester vom Responder mitgeteilte Ausleihkonditionen oder lehnt sie ab. Voraussetzung dafür ist, daß die Fernleihanfrage mit ILL-ANSWER: CONDITIONAL beantwortet wurde. Wenn der Requester den Ausleihkonditionen zustimmt und die für die geforderten Konditionen nötigen Informationen bereitstellt, wird die Fernleihanfrage durch den Responder weiter bearbeitet. Stimmt er jedoch nicht zu, wird die ILL-Transaktion beendet.

Bemerkung: Der Dienst „CONDITIONAL-REPLY“ muß in einer ILL-Anwendung unterstützt werden.

3.5.8 Stornierung (CANCEL)

Versender: Requester

Funktion: Die „Stornierung“ erlaubt einem Requester, eine bereits versendete Fernleihanfrage zu widerrufen.

3.5.9 Antwort auf eine Stornierung (CANCEL-REPLY)

Versender: Responder

Funktion: Mit diesem Dienst kann ein Responder eine empfangene Stornierung akzeptieren oder ablehnen. Akzeptiert es diese, wird die ILL-Transaktion beendet.

Bemerkung: Der Dienst „CANCEL-REPLY“ muß in einer ILL-Anwendung unterstützt werden.

3.5.10 Empfang (RECEIVED)

Versender: Requester

Funktion: Der „Empfang“ erlaubt einem Requester dem Responder anzuzeigen, daß das gewünschte Leihobjekt empfangen wurde.

Bemerkung: Der Dienst „RECEIVED“ muß in einer ILL-Anwendung unterstützt werden.

3.5.11 Vorzeitiger Rückruf (RECALL)

Versender: Responder

Funktion: Der „vorzeitige Rückruf“ wird benutzt, wenn ein Responder dem Requester anzeigen will, daß dieser ein ausgeliehenes Objekt unverzüglich zurücksenden muß. Eine Anfrage des Requesters auf Leihfristverlängerung³⁴ ist dann nicht mehr möglich.

Bemerkung: Der Dienst „RECALL“ muß unterstützt werden, sofern in einer ILL-Anwendung die Ausleihe von Objekten und nicht nur die Lieferung von Kopien unterstützt wird.

3.5.12 Rückgabe (RETURNED)

Versender: Requester

Funktion: Die „Rückgabe“ erlaubt einem Requester dem Responder anzuzeigen, daß ein ausgeliehenes Objekt an den Responder zurückgeschickt worden ist.

Bemerkung: Der Dienst „RETURNED“ muß unterstützt werden, sofern in einer ILL-Anwendung die Ausleihe von Objekten und nicht nur die Lieferung von Kopien unterstützt wird.

3.5.13 Eingang (CHECKED-IN)

Versender: Responder

Funktion: Dieser Dienst erlaubt es einem Responder dem Requester anzuzeigen, daß ein vom Requester ausgeliehenes Objekt zurückerhalten wurde.

Bemerkung: Der Dienst „CHECKED-IN“ muß unterstützt werden, sofern in einer ILL-Anwendung die Ausleihe von Objekten und nicht nur die Lieferung von Kopien unterstützt wird.

3.5.14 Ablauf der Ausleihfrist (OVERDUE)

Versender: Responder

Funktion: Mit dem „Ablauf der Ausleihfrist“ kann ein Responder einem Requester mitteilen, daß ein Objekt überfällig ist, wenn das Fristdatum bereits erreicht ist. Dieser Dienst kann automatisch durch die ILL-Anwendung des Responders oder manuell durch einen Bearbeiter auf Responderseite ausgelöst werden. Vom Requester wird anschließend erwartet, daß er das ausgeliehene Objekt zurückgibt oder eine Leihfristverlängerung beantragt.

Bemerkung: Der Dienst „OVERDUE“ muß unterstützt werden, sofern in einer ILL-Anwendung die Ausleihe von Objekten und nicht nur die Lieferung von Kopien unterstützt wird.

³⁴ Siehe Abschnitt 3.5.15.

3.5.15 Antrag auf Leihfristverlängerung (RENEW)

Versender: Requester

Funktion: Der Requester kann mit diesem Dienst beim Responder die Verlängerung der ursprünglichen Leihfrist beantragen. Er kann dabei ein Wunschkdatum angeben.

3.5.16 Antwort auf Leihfristverlängerungsantrag (RENEW-ANSWER)

Versender: Responder

Funktion: Die „Antwort auf Leihfristverlängerung“ erlaubt dem Responder, einen empfangenen Leihfristverlängerungsantrag zu akzeptieren oder abzulehnen. Akzeptiert er den Antrag, kann er das Wunschkdatum des Requesters übernehmen oder ein anderes Datum angeben, welches dann für den Requester bindend ist.

Bemerkung: Der Dienst „RENEW-ANSWER“ muß unterstützt werden, sofern in einer ILL-Anwendung die Ausleihe von Objekten und nicht nur die Lieferung von Kopien unterstützt wird.

3.5.17 Verlustanzeige (LOST)

Versender: Responder und Requester

Funktion: Die „Verlustanzeige“ wird benutzt, wenn ein Leihobjekt bei der Bearbeitung durch den Responder, auf dem Transportweg, beim Requester oder dessen Kunden verloren gegangen ist - und nur dann. Wenn nur vermutet wird, daß das Leihobjekt verloren gegangen ist, dann sollte der Dienst MESSAGE verwendet werden, um den Fernleihpartner darüber zu informieren.

Bemerkung: Der Dienst „LOST“ muß in einer ILL-Anwendung unterstützt werden.

3.5.18 Beschädigungsanzeige (DAMAGED)

Versender: Responder und Requester

Funktion: Die „Beschädigungsanzeige“ wird benutzt, wenn ein Leihobjekt bei der Bearbeitung durch den Responder, auf dem Transportweg, beim Requester oder dessen Kunden beschädigt worden ist. Derjenige, der die Beschädigung als erster entdeckt, sollte den Fernleihpartner durch die Benutzung dieses Dienstes darauf aufmerksam machen. Hat die Beschädigung des Leihobjektes finanzielle Folgen, müssen diese anschließend außerhalb der ILL-Anwendung geregelt werden.

3.5.19 Formlose Nachricht (MESSAGE)

Versender: Responder und Requester

Funktion: Dieser Dienst erlaubt es, jederzeit formlose Nachrichten zu einer bestehenden ILL-Transaktion zu versenden. Solche Nachrichten können vielfältig genutzt werden, es sollten allerdings keine Informationen durch diesen Dienst ausgetauscht werden, die schon durch andere Dienste abgedeckt sind.

3.5.20 Anfrage zum Zustand einer ILL-Transaktion (STATUS-QUERY)

Versender: Responder und Requester

Funktion: Durch diesen Dienst kann der aktuelle Zustand einer ILL-Transaktion ermittelt werden. Dies ist besonders in Situationen hilfreich, in denen sich der ILL-Transaktionszustand innerhalb eines Systems ändert, ohne eine entsprechende Zustandsänderung in der ILL-Anwendung des Fernleihpartners hervorzurufen. Der Dienst erlaubt es auch, jederzeit den Zustand einer ILL-Transaktion beim Fernleihpartner zu erfragen, um daraus Konsequenzen für weitere Interaktionen zu ziehen.

3.5.21 Zustands- oder Fehlerbericht (STATUS-OR-ERROR-REPORT)

Versender: Responder und Requester

Funktion: Dieser Dienst erlaubt es, Fehler- oder/und Zustandsinformationen an den Fernleihpartner zu übermitteln. Die Zustandsinformation kann zu jeder Zeit oder als Antwort auf eine Anfrage zum Zustand (STATUS-QUERY) zur Verfügung gestellt werden. Sowohl der aktuelle Zustand einer ILL-Transaktion als auch andere sachdienliche Informationen sind im Status-Report enthalten.

Ein Fehler-Report kann durch den Dienstbenutzer oder durch den Dienstanbieter initiiert werden, um eine Fernleihanfrage zurückzuweisen, wenn ein Problem entdeckt wurde.

Bemerkung: Der Dienst „STATUS-OR-ERROR-REPORT“ muß in einer ILL-Anwendung unterstützt werden.

3.5.22 Ablauf der ILL-Transaktion (Expiry)

Versender: ILL-Anwendung

Funktion: Mit dem „Ablauf der ILL-Transaktion“ kann der Dienstanbieter dem Dienstbenutzer mitteilen, daß eine ILL-Transaktion abgelaufen ist.

3.6 ILL-Dienstaufruf-Sequenzen

Durch Zeit-Sequenz-Diagramme kann zum Ausdruck gebracht werden, wie Sequenzen von ILL-Dienstaufrufen zeitlich zueinander in Relation stehen.

Desweiteren zeigen diese Diagramme:

- die Sequenz von Ereignissen jeweils beim Requester und beim Responder
- falls angemessen, die Sequenz von Ereignissen zwischen Requester und Responder.

Beispielhaft werden im ISO-Protokoll [ISO 10160, 1991, S114 ff.] verschiedene ILL-Transaktionen anhand von Zeit-Sequenz-Diagrammen aufgeführt. Dabei wird deutlich, welche (zeitliche) Reihenfolge von obligatorischen und auch optionalen ILL-Dienstaufrufen für die Abhandlung einer bestimmten ILL-Transaktion logisch ist. Es wird dabei auch auf mögliche Kollisionen zwischen einzelnen Diensten hingewiesen.

4

Die abstrakte Syntaxnotation 1 (ASN.1)

4.1 Die Beschreibung von Datenstrukturen

Eine Bibliothek gliedert sich in verschiedene Bereiche mit verschiedenen Aufgabengebieten. So kann es z.B. eine Abteilung geben, die für Fernleihbestellungen zuständig ist, eine andere, die die Verbuchung durchführt und wieder eine andere, die sich mit der Recherche von Büchern beschäftigt.

Daten Jeder Bereich benötigt für die Durchführung seiner Aufgaben bestimmte **Daten**. Die Ausleihverbuchung benötigt die Stammdaten der Benutzer, die Fernleihabteilung benötigt zusätzlich Daten der Partnerbibliotheken, mit denen sie den Leihverkehr praktiziert und bei der Recherche werden die Daten für das gesuchte Objekt benötigt.

Datenstruktur Meist handelt es sich dabei nicht um einfache Datentypen, wie z.B. der Name „Ingo Mustermann“ für einen Benutzer. Vielmehr handelt es sich um komplexe Datentypen, die nicht nur den Namen eines Benutzers, sondern zusätzlich noch seine Benutzernummer, Adresse und Telefonnummer zu einer Struktur zusammenfassen. Eine **Datenstruktur** setzt sich also aus mehreren einzelnen Elementen (Datentypen bzw. Datenfelder) zusammen, die nicht unbedingt vom gleichen Typ sein müssen. Die Datenstruktur Benutzer würde dann z.B. die Datenfelder „Musterbenutzernummer“, „Ingo Mustermann“, „Musterstraße“ und „99999 Musterstadt“ enthalten.

Einheitliche Darstellung und Beschreibung Jeder Bereich verwendet also zahlreiche Datenstrukturen, die für die verschiedenen Aufgaben benötigt werden, und die über die Netzwerke übertragen werden müssen. Damit die Datenstrukturen zwischen verschiedenen Systemen und Programmen und in verschiedenen Programmiersprachen übertragen werden können, ist eine **einheitliche Darstellung und Beschreibung** notwendig.

Notation Bei der Darstellung, Kodierung, Übertragung und Dekodierung von Datenstrukturen kommt daher der Beschreibung von Datenstrukturen, der **Notation**, eine besondere Bedeutung zu. Die Notation muß flexibel genug sein, um in vielen Anwendungen eingesetzt werden zu können und doch soweit standardisiert, daß sie allgemein anerkannt wird.

ASN.1-Notation der OSI Als eine der wichtigsten Beschreibungssprachen bei den Standardisierungsbestrebungen im Rahmen der offenen Kommunikation³⁵ hat sich in den letzten Jahren die **Notation ASN.1** (Abstract Syntax Notation One) der internationalen Standardisierungsorganisation ISO herausgestellt. Diese Beschreibungssprache ist die Grundlage für die Strukturdefinition der neuen (Norm-) Protokollstandards auf der Anwendungs- und Darstellungsschicht.

³⁵ Open Systems Interconnections (OSI).

4.2 Das Grundprinzip von ASN.1

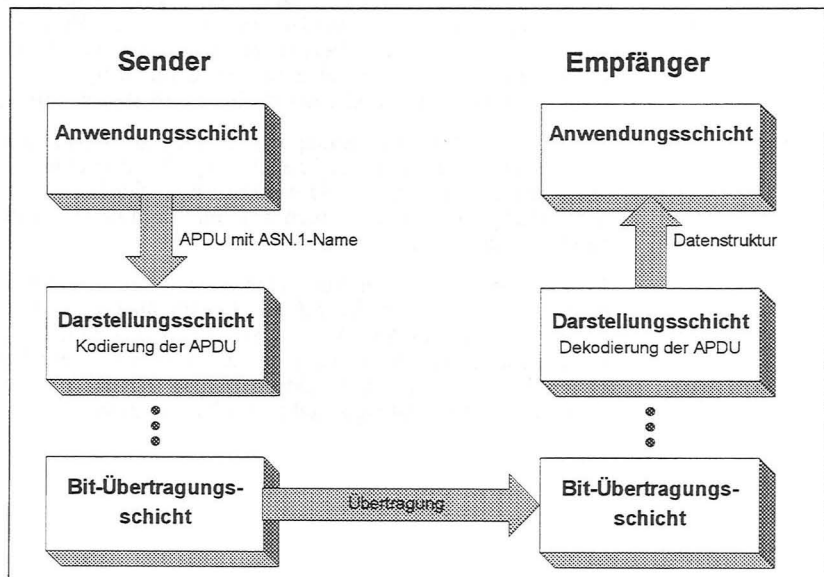
APDU Generell verwendet also jede Anwendung eine Reihe von komplexen Strukturen, die als **APDUs** (Application Protocol Data Units) über das Netzwerk übertragen werden.

Grundidee von ASN.1 Die **Grundidee von ASN.1** besteht darin, daß alle Datenstrukturen (Datentypen), die in verschiedenen Anwendungen eingesetzt werden, in einer einheitlichen Schreibweise, der ASN.1-Notation, definiert und dann in einem Modul (einer Bibliothek) zusammengefaßt werden. Wenn eine Anwendung eine Datenstruktur (d.h. eine APDU) übertragen will, kann sie diese zusammen mit ihrem ASN.1-Namen an die Darstellungsschicht übergeben. Durch den Einsatz der ASN.1-Definition erkennt die Darstellungsschicht, von welchem Typ und wie groß die Felder sind und daher auch, wie sie die Daten für die Übertragung kodieren soll.

Am anderen Ende der Verbindung analysiert die Darstellungsschicht des Empfängers die ASN.1-Kennung der Datenstruktur und erkennt Länge, Typ und Anzahl der Felder. Mit Hilfe dieser Information kann die Darstellungsschicht das externe Format, das für die Übertragung in der Leitung verwendet wurde, in das im Zielrechner benutzte interne Format konvertieren.

Wenn nun eine Anwendung eine Datenstruktur (d.h. eine APDU) übertragen möchte, kann sie in der Darstellungsschicht zwei Zeiger setzen. Der erste zeigt auf die formale Definition des Datentyps in ASN.1, der zweite auf die APDU selbst. Mit Hilfe dieser Information gibt die Kodieroutine einen optimierten, sich selbst identifizierenden Bytestrom aus. Beim Zielrechner wird eine semantisch genau gleiche Datenstruktur aufgebaut, die dann an die Anwendungsschicht weitergegeben wird, vgl. [Tanenbaum, 1992].

Abb. 4.1
Übertragung einer APDU



Das Grundprinzip von ASN.1 wird im folgenden Beispiel nochmals verdeutlicht. Da die abstrakte Syntax von ASN.1 erst im nächsten Abschnitt genauer beschrieben wird, wird für dieses Beispiel eine abstrakte Syntax in der Programmiersprache C verwendet.

In der folgenden Abbildung wird die Datentypdefinition einer Beschädigungsanzeige in einer abstrakten Syntax in C beschrieben. Eine Beschädigungsanzeige wird von einer Bibliothek zur anderen gesendet, um die Beschädigung eines Objektes zu melden:

Abb. 4.2
Abstrakte Syntax
in C

```
typedef struct Beschaedigungsanzeige {

    int Protokollnummer;
    char Absender [25];
    char Empfaenger [25];
    char Beschaedigung [160];
    int Schuldanerkennung;

} DAMAGED;
```

Typ, Länge und Wert

Wird nun eine Beschädigungsanzeige mit der oben definierten Struktur von einem 32-Bit-Rechner der Universität Konstanz an die Universität Ulm geschickt, so ergibt sich der in Abb. 4.3 gezeigte Bytestrom. Da sich jede Datenart in der Leitung selbst identifizieren muß, wird jedes Feld in der Datenstruktur durch **Typ, Länge und Wert** dargestellt.

Datentypen des
Bytestroms

Für dieses Beispiel gelten folgende **Datentypen**:

- 1 für Integer
- 2 steht für Zeichenkette
- 3 für den Satzanfang

Daten des
Bytestroms

Der 57-Byte-Datensatz enthält folgende **Daten** (Werte):

- Protokollnummer: 300
- Absender: Uni Konstanz
- Empfänger: Uni Ulm
- Beschädigung: Die Seiten 354-363 fehlen
- Schuldanerkennung: 0 (für nein)

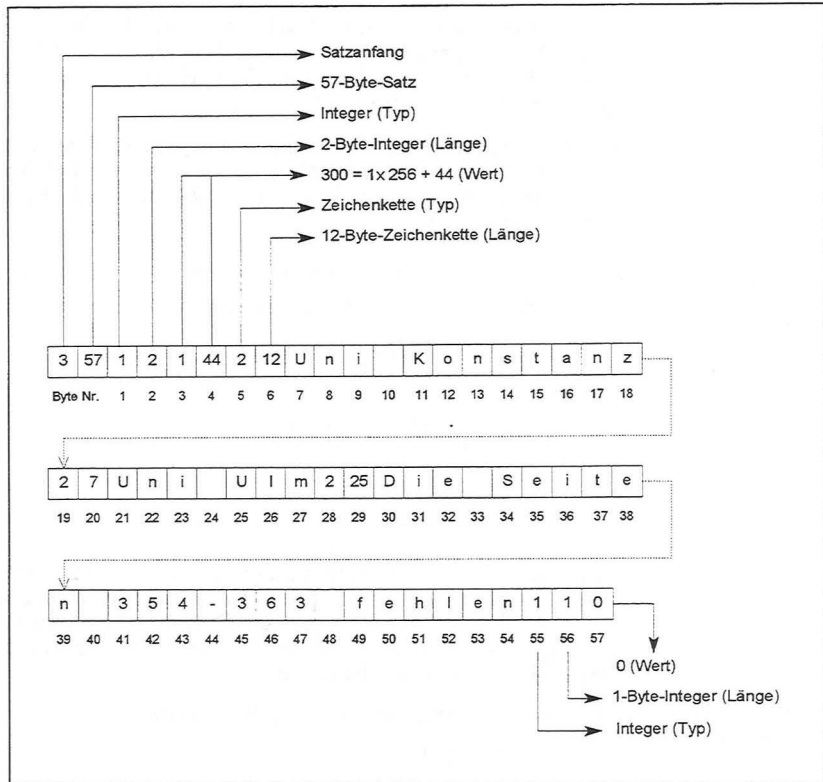
Das anfängliche Byte des Bytestroms enthält den Wert 3, der den Beginn des Datensatzes anzeigt. Das nächste Byte mit dem Wert 57 gibt an, daß der gesamte Bytestrom 57 Bytes lang ist und zwar ab dem darauffolgenden Byte beginnend.

Das darauffolgende Byte mit der Nr. 1 enthält den Wert 1, der aussagt, daß der erste Wert vom Typ Integer ist. Im Byte mit der Nr. 2 wird durch die Zahl 2 angegeben, wie lange der Wert ist; nämlich 2 Byte. Im Byte mit der Nr. 3 folgt schließlich der Wert 300, dargestellt mit 32-Bit $((1 \times 2^8) + (44 \times 2^0))$, was 144 ergibt.

Danach wird immer wieder nach derselben Reihenfolge fortgefahren. Zuerst wird der Datentyp angegeben, dann die Länge der Daten in Byte und schließlich die Daten selbst. So folgt als nächstes eine Zeichenkette, die 12 Bytes lang ist und die Zeichen „Uni Konstanz“ enthält usw.

Die abstrakte Syntaxnotation 1 (ASN.1)

Abb. 4.3
Kodierung von Typ,
Länge und Wert



4.3

Die abstrakte Syntax der ASN.1-Notation

Abstrakte Syntax

Die ASN.1-Beschreibung eines Datentyps wird **abstrakte Syntax** genannt. Abstrakt wird sie deshalb genannt, weil sie keine spezifische Darstellung enthält. Die Definition der Beschädigungsanzeige in Abb. 4.2 kann auch als abstrakte Syntax betrachtet werden. Sie ist lediglich in der Programmiersprache C geschrieben.

Elemente und Bausteine

Die abstrakte Syntax der ASN.1-Notation gliedert sich in verschiedene **Grundelemente**. Aus diesen Elementtypen können **Bausteine** gebildet werden, die dann wiederum zu noch komplexeren Datenstrukturen zusammengesetzt werden können.

Die ASN.1-Grundelementtypen sind in der folgenden Tabelle aufgelistet. Ihre Namen sind reservierte Wörter und werden wie alle anderen reservierten Wörter in ASN.1 in Großbuchstaben geschrieben:

Abb. 4.4
Elementtypen von
ASN.1

Elementtyp	Bedeutung
BOOLEAN	TRUE oder FALSE
INTEGER	ganze Zahl
BIT STRING	Liste mit 0 oder mehr Bits
OCTET STRING	Liste mit 0 oder mehr Bytes
NULL	keiner der Typen
ANY	Vereinigung aller Typen
OBJECT IDENTIFIER	Objektname

BOOLEAN **BOOLEAN** gibt an, ob bestimmte Bedingungen erfüllt sind und wird durch die Werte TRUE oder FALSE abgebildet.

INTEGER **INTEGER** sind Kardinalzahlen, die zum Zählen eingesetzt werden. Der Wertebereich von Integer-Typen kann unterschiedlich definiert und maschinenabhängig sein.

BIT STRING Bitketten (**BIT STRINGS**) sind geordnete Listen von Bits. Jede Bitkette hat 0 oder mehr Bits.

OCTET STRING **OCTET** ist die OSI-Bezeichnung für das, was allgemein unter Byte bekannt ist. Oktettfolgen (**OCTET STRINGS**) sind geordnete Bytelisten. Sie können zur Darstellung von Zeichen oder zur Darstellung von beliebigen byte-orientierten Daten benutzt werden, für die kein anderer Datentyp paßt.

ANY Wird der Typ **ANY** in einem Feld eines Datensatzes verwendet, kann hinterher ein beliebiger gültiger Typ eingesetzt werden. **ANY** ist die Menge aller Typen.

NULL **NULL** bedeutet keine Typangabe. Wird einem Feld in einem Satz der Wert **NULL** zugeteilt, so hat dieses Feld keinen Wert.

OBJECT IDENTIFIER Wird eine Sitzung zwischen Sender und Empfänger aufgebaut, wird ausgehend von der Darstellungsschicht eine Absprache durchgeführt, damit gewährleistet ist, daß beide Partner dieselbe abstrakte Syntax und dieselben Kodierungsregeln und Protokolle für ihre jeweilige Anwendung verwenden. All diese Objekte (Bibliotheken) werden mit Objektkennungen (**OBJECT IDENTIFIER**) gekennzeichnet.

Aus den einzelnen Elementtypen können die folgenden fünf Hauptbausteine erzeugt werden:

Abb. 4.5
Bausteine von
ASN.1

Baustein	Bedeutung
SEQUENCE	geordnete Liste mit verschiedenen Typen
SEQUENCE OF	geordnete Liste mit nur einem Typ
SET	ungeordnete Sammlung mit verschiedenen Typen
SET OF	ungeordnete Sammlung mit nur einem Typ
CHOICE	beliebiger Typ aus einer gegebenen Liste

Die abstrakte Syntaxnotation 1 (ASN.1)

SEQUENCE	In einer SEQUENCE wird eine geordnete Folge von verschiedenen Grundelementtypen zusammengefaßt.
SEQUENCE OF	Der Baustein SEQUENCE OF besteht aus einer geordneten Folge von nur einem Grundelementtyp.
SET	SET ist SEQUENCE ähnlich. Der Unterschied besteht allerdings darin, daß die Reihenfolge der einzelnen Grundelementtypen beliebig angeordnet werden kann.
SET OF	Der Baustein SET OF ist eine Sammlung von Datenstrukturen eines einzelnen Grundelementtyps und vergleichbar mit SEQUENCE OF . Die Elemente unterliegen jedoch keiner bestimmten Reihenfolge.
CHOICE	CHOICE kommt zum Einsatz, wenn in einer Datenstruktur einer von mehreren Typen zur Auswahl steht.
OPTIONAL	Felder vom Typ OPTIONAL können angegeben und übertragen werden, müssen aber nicht.
DEFAULT	Bei Feldern vom Typ DEFAULT kann der Sender ebenfalls wie bei OPTIONAL entscheiden, ob er die Felder angibt und sendet oder nicht. Bei DEFAULT -Feldern kann der Empfänger aber zusätzlich, bei nicht übertragenen Feldern, einen beliebigen Wert nach seinen Wünschen einsetzen.

Die Abb. 4.2 zeigte eine einfache abstrakte Syntax in C. Die folgende Abbildung zeigt ebenfalls eine abstrakte Syntax in C, jedoch in einer komplexeren Form. Hier bestehen einzelne Elemente der Datenstruktur wiederum aus einer Datenstruktur.

Abb. 4.6
Abstrakte Syntax
in C

```
typedef struct damaged {
    int protocol-version-num;
    SYSTEM-ID requester-id;
    SYSTEM-ID responder-id;
    STRING injury;
    int debt-acknowledge;
} DAMAGED;

typedef struct System-Id{
    PERSON-OR-INSTITUTION-SYMBOL person-or-institution-symbol;
    NAME-OF-PERSON-OR-INSTITUTION name-of-person-or-institution;
} SYSTEM-ID;

typedef struct Name-Of-Person-Or-Institution {
    STRING name-of-person;
    STRING name-of-institution;
} NAME-OF-PERSON-OR-INSTITUTION;

typedef struct Person-Or-Institution-Symbol {
    STRING person-symbol;
    STRING institution-symbol;
} PERSON-OR-INSTITUTION-SYMBOL;

typedef struct String {
    unsigned char [160];
} STRING;
```

Zur Erläuterung der zuvor beschriebenen Elementtypen und Bausteine wurde die obige C- Syntax in ASN.1-Syntax übersetzt:

Abb. 4.7
Abstrakte Syntax in
ASN.1

```

Damaged ::= [APPLICATION 16] SEQUENCE {
  protocol-version-num      [0] IMPLICIT INTEGER,
  requester-id              [1] IMPLICIT System-Id OPTIONAL,
  responder-id              [2] IMPLICIT System-Id OPTIONAL,
  injury                    [3] String
  debt-acknowledgement     [4] IMPLICIT INTEGER
}

System-Id ::= SEQUENCE {
  person-or-institution-symbol [0] Person-Or-Institution-
    Symbol OPTIONAL,
  name-of-person-or-institution [1] Name-Of-Person-Or-
    Institution OPTIONAL
}

Name-Of-Person-Or-Institution ::= CHOICE {
  name-of-person            [0] String,
  name-of-institution       [1] String
}

Person-Or-Institution-Symbol ::= CHOICE {
  person-symbol            [0] String,
  institution-symbol [1] String
}

String ::= VisibleString (FROM ("A"|"B"|"C"|...|"Z"|
  "a"|"b"|"c"|...|"z"|"1"|"2"|"3"|...|"9"|"0"|
  " " | "." | "," | "-" | "(" | ")" | "/" | "=" | "|" | "" | "&" |
  "&" | "+" | ";" | "<" | ">" | "'" | "+" | ":" | "?"))

```

Die obige Abbildung zeigt die Applikation Damaged in ASN.1-Notation mit der eindeutig identifizierenden Nummer 16. Diese APDU ist vom Typ SEQUENCE und besteht somit aus einer geordneten Folge von verschiedenen Grundelementtypen. Diese Grundelementtypen sind:

- **protocol-version-num**
ist eine Zahl vom Typ 'IMPLICIT INTEGER'³⁶.
- **requester-id**
ist vom Typ 'System-Id', der wiederum aus einer SEQUENCE besteht. Der Typ OPTIONAL erlaubt es, den Elementtyp 'requester-id' anzugeben oder auch nicht.
- **responder-id**
ist ebenfalls vom Typ 'System-Id OPTIONAL'.
- **injury**
ist vom Typ String und erlaubt eine Menge von numerischen Zeichen, alphanumerischen Zeichen und Sonderzeichen.
- **debt-acknowledge**
ist eine Zahl vom Typ 'IMPLICIT INTEGER'.

³⁶ Der Begriff IMPLICIT wird im nächsten Abschnitt erläutert.

Die Ziffern in eckigen Klammern werden im nächsten Abschnitt erläutert.

4.4 Die Transfersyntax von ASN.1

Die Existenz der Typen OPTIONAL und DEFAULT kann möglicherweise bei der Identifizierung der ankommenden Daten zu Problemen führen. Nimmt man z.B. an, SEQUENCE hat zehn Felder, die alle vom Typ INTEGER und OPTIONAL sind und es werden aber nur drei von ihnen übertragen. Wie weiß nun der Empfänger, welche von den zehn Feldern übertragen wurden?

Tag Dieses Problem kann mit der Einführung von **Tags** (Identifizierungskennzeichen) gelöst werden. ASN.1 läßt identifizierende Tags für Datentypen und Datenfelder zu.

Übergeordnete Datenstruktur Zur Begriffsklärung sei gesagt, daß eine Datenstruktur ein definierter Datentyp ist, der aus mehreren Datenfeldern (einfache Datentypen oder wiederum Datenstrukturen) zusammengesetzt ist. Besitzt eine Datenstruktur ein Feld, das wiederum eine Datenstruktur ist, so ist sie die **übergeordnete Datenstruktur**.

Typen von Tags Es gibt vier verschiedenen Typen von Tags:

- UNIVERSAL
- APPLICATION
- PRIVATE
- kontextspezifisches Tag

Jedes Tag besteht aus einer ganzen Zahl, der eines der reservierten Wörter UNIVERSAL, APPLICATION bzw. PRIVATE voransteht, zum Beispiel [APPLICATION 16]³⁷. Geht ein nicht reserviertes Wort der ganzen Zahl voran oder überhaupt keine Wort handelt es sich um ein kontextspezifisches Tag, z.B. [0]³⁸. Tags werden immer in eckigen Klammern angegeben.

UNIVERSAL Das Tag **UNIVERSAL** ist für die Grundelementtypen reserviert.

APPLICATION Das Tag **APPLICATION** kommt in vielen OSI-Anwendungsprotokollen vor und kennzeichnet die verschiedenen Applikationstypen dieser Protokolle. Beim ILL-Protokoll entspricht dies den einzelnen Diensten (MESSAGE, LOST, DAMAGED usw.)

PRIVATE Das Tag **PRIVATE** ist für Typen reserviert, die der Schreiber der abstrakten Syntax selbst definiert hat.

Kontextspezifische Tags **Kontextspezifische Tags** werden dort eingesetzt, wo zwischen einzelnen Datenfeldern eindeutig unterschieden werden muß. Der Gültigkeitsbereich eines Tags bleibt jedoch nur innerhalb der übergeordneten Datenstruktur bestehen. So kann das Tag [1] in verschiedenen Datentypen (Datenstrukturen) verwendet werden z.B. im Datentyp Transaction-Id und im Datentyp Service-Date-Time. Im Gegensatz zu kontextspezifischen Tags dürfen die anderen Tagtypen nur einmal pro Protokoll vorkommen, z.B. das Tag [APPLICATION 17] darf im ILL-Protokoll nur einmal vorkommen und steht dort für den Dienst ILL-MESSAGE.

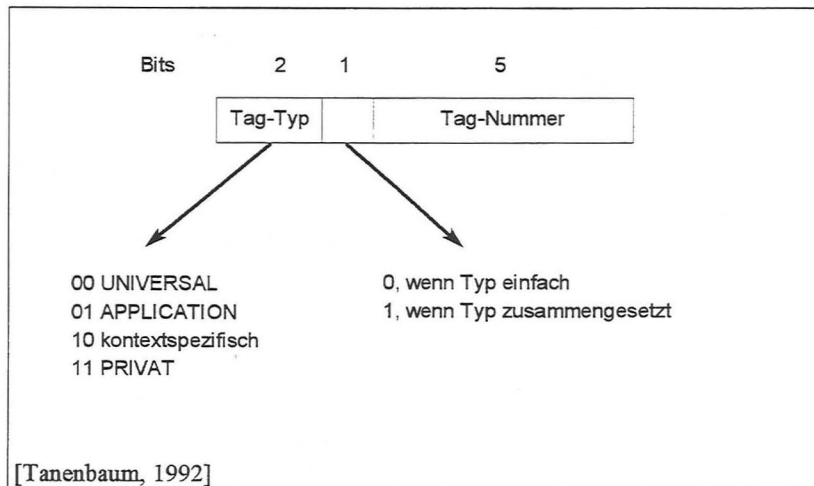
Im ILL-Protokoll werden nur das Tag APPLICATION und kontextspezifische Tags benutzt.

³⁷ Vgl. Abb. 4.7.

³⁸ Vgl. Abb. 4.7.

Ein Tag gliedert sich in drei Felder, die man in Abb. 4.8 sehen kann. Die meisten Tags sind in einem einzigen Byte kodiert. Hierbei geben die 2 höchstwertigen Bits den Tag-Typ an. Die Tag-Bits sind 00, 01, 10 und 11 und repräsentieren UNIVERSAL, APPLICATION, kontextspezifisch und PRIVATE. Das nächste Bit sagt aus, ob es sich um einen einfachen (0) oder zusammengesetzten Typ (1) handelt. Die 5 niedrigsten Bits geben die Tag-Nummer (0,1,2,3,4,46,49 usw.) an:

Abb. 4.8
Erstes Byte eines
Datenfeldes



Identifizierung

In Abschnitt 4.2 wurde bereits erwähnt, daß jedes Feld in der Datenstruktur durch Typ, Länge und Wert dargestellt wird. Hat die Typenbezeichnung oder das Feld ein Tag, so wird dieses zum Zweck der **Identifizierung** bei einer Übertragung mitgeschickt.

Greift man nochmals das o.g. Problem auf: „Wie erkennt der Empfänger bei optionalen Feldern, ob sie übertragen wurden oder nicht?“, so kann dieses Problem nun durch die Verwendung von Tags gelöst werden. Anhand eines Tags kann der Empfänger nämlich jeden Datentyp und jedes Feld genau identifizieren.

IMPLICIT

Wird also ein Tag mitübertragen, so ist die Übertragung des Datentyps überflüssig, da das Tag diese Information auch besitzt. Mit ASN.1 kann daher die Typinformation unterdrückt werden, wenn Typen oder Felder mit Tags übertragen werden. Dazu muß die Person, die die abstrakte Syntax schreibt, das reservierte Wort **IMPLICIT** nach dem Tag anfügen, zum Beispiel so: [1] IMPLICIT INTEGER³⁹. Wenn IMPLICIT nicht nach einem Tag folgt, werden sowohl Tag als auch Typ übertragen

Grundsätzlich gilt also für die ASN.1-Transfersyntax, daß die übertragenen Werte, sowohl die einfachen als auch die zusammengesetzten Werte, aus drei Feldern bestehen:

1. Kennung (Typ oder Tag)
2. Länge des Datenfeldes in Bytes
3. Datenfeld (Wert)

³⁹ Vgl. Abb. 4.7.

Diese drei Felder müssen immer angegeben werden.

4.5

Der ASN.1-Compiler der Universität Erlangen

Automatische
Umsetzung

Eine manuelle Umsetzung der ASN.1-Beschreibung in Strukturen einer Programmiersprache, sowie die Bereitstellung notwendiger Kodierungs- und Dekodierungsroutinen ist aufgrund der Komplexität der Strukturen mühselig und vor allem fehleranfällig. Die **automatische Umsetzung** der Spezifikation in Strukturen einer Programmiersprache, sowie die Generierung von Kodierungs- und Dekodierungsroutinen, ist daher zu bevorzugen. Im Zuge der automatisierten Umsetzung können auch zusätzliche Hilfsfunktionen, z.B. zum Traversieren der Strukturen oder zum Anzeigen der Strukturen, während der Laufzeit des Compilats schnell generiert werden.

ASN.1-Compiler der
Universität Erlangen

Das automatisierte Fernleihsystem der Universität Konstanz wurde in der Programmiersprache C entwickelt und erfordert daher einen ASN.1-Compiler, der die Datenstrukturen des ILL-Protokolls in C umsetzt. Man entschied sich für den **ASN.1-Compiler der Universität Erlangen**.

4.6

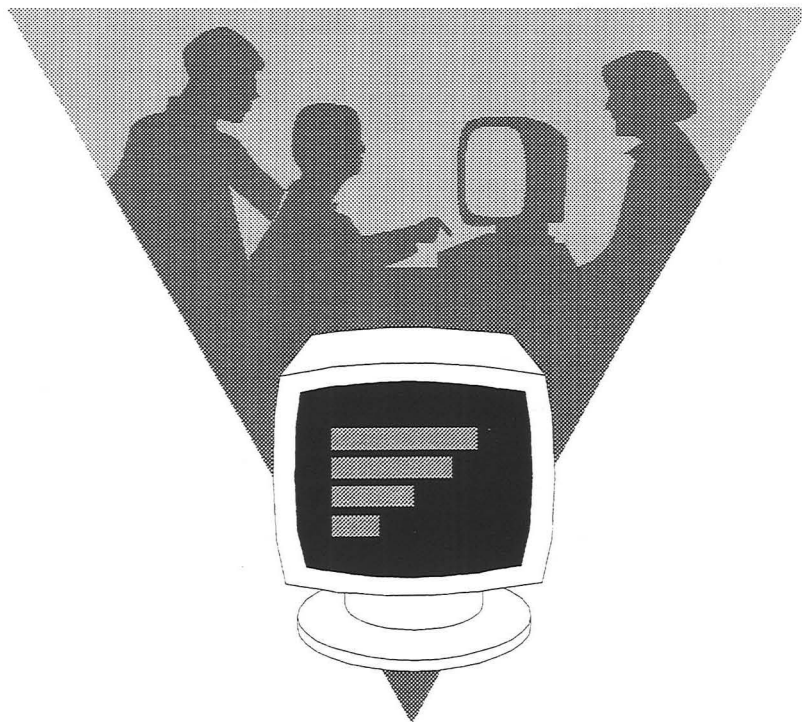
ASN.1 im Fernleihsystem der Universität Konstanz

ISO_10161_IL.h
ISO_10161_IL.c

Beim Konstanzer Fernleihsystem generiert der ASN.1-Compiler, aus dem in ASN.1-Notation dargestellten ILL-Protokoll, die Dateien **ISO_10161_IL.h** und **ISO_10161_IL.c**. Die Headerdatei **ISO_10161_IL.h** enthält die Deklaration der Kodierungs- und Dekodierungsroutinen der ILL-Dienste, sowie die Typdefinitionen der Datenstrukturen des ILL-Protokolls. Die Datei **ISO_10161_IL.c** enthält die entsprechenden Definitionen der Kodierungs- und Dekodierungsroutinen, sowie die notwendigen C-Routinen zur Erzeugung des Bytestroms und zum Schreiben des Bytestroms in eine Datei, die dann über das Netz zum Zielsystem übertragen werden kann.

Kapitel 3

Realisierung der
elektronischen Fernleihe
an der Universität Konstanz



5

Unterstützte Kommunikationssysteme

Zunächst war für das Fernleihprojekt der Bibliothek der Universität Konstanz der Datenaustausch über einen Dienst nach der Empfehlung X.400 für elektronische Postsysteme vorgesehen. Dieser Standard, der zu den Protokollen der Open Systems Interconnection (OSI) gehört, hat sich jedoch nicht in großem Umfang durchgesetzt. Bei der Realisierung des Fernleihsystems gab es Probleme mit der X.400-Mailsystem-Software, die den Datenaustausch ermöglichen sollte. Dennoch sind in sämtlichen Kommunikationsmodulen die Schnittstellen für diesen Informationsaustausch vorgesehen. Deshalb sollen im folgenden Abschnitt die Grundlagen der Protokollserie X.400 nur kurz erläutert werden. Für eine weitere Vertiefung dieses Themas wird auf das Literaturverzeichnis verwiesen.

In der aktuellen Implementierung wird von sämtlichen Modulen der Datenaustausch über die Internet Protokoll Suite mit TCP/IP realisiert. Diese Protokolle sind Bestandteil des UNIX-Betriebssystems und müssen nicht zusätzlich erworben werden. Die immense Marktdurchsetzung und die große Anzahl vorhandener Installationen bietet den Vorteil, eine große Anzahl von Systemen zu erreichen. Die einzelnen Kommunikationsfunktionen, die für den Datenaustausch zwischen Client und Server implementiert wurden, ließen sich durch die Verwendung von TCP/IP über den Aufruf eines Telnet-Client testen, da dieser auf den Protokollen aufbaut. Auch die Grundlagen von TCP/IP sollen im folgenden Abschnitt erläutert werden.

5.1

Message Handling

Geschichtlicher
Hintergrund

1984 entwickelte die CCITT⁴⁰, die heute ITU-T⁴¹ heißt, die Protokollserie X.400 als Empfehlung für Nachrichtenvermittlungssysteme (Message Handling Systems). Die ISO⁴² versuchte, diese Protokolle in die OSI-Anwendungsschicht (Schicht 7 des OSI-Referenzmodells) zu übernehmen.

Die ISO Norm 10021 Absatz 1-7 ist auch unter dem Namen MOTIS (Message-Oriented Text Interchange Systems) bekannt. 1988 glied die CCITT X.400 mit MOTIS ab.

X.400 ermöglicht die Übermittlung von Nachrichten allgemeiner Art. Für die Übermittlung von Informationen, die dem Standard des internationalen Leihverkehrs entspricht, bedarf es einer zusätzlichen Normierung. Dieser entspricht das sogenannte ILL-Protokoll⁴³, welches ebenfalls zur OSI-Anwendungsschicht gehört.

⁴⁰ Comité Consultatif International de Télégraphique et Téléphonique

⁴¹ International Telecommunication Union

⁴² International Organization for Standardization

⁴³ Das Inter-Library-Loan Protokoll wurde in Abschnitt 3 bereits erläutert.

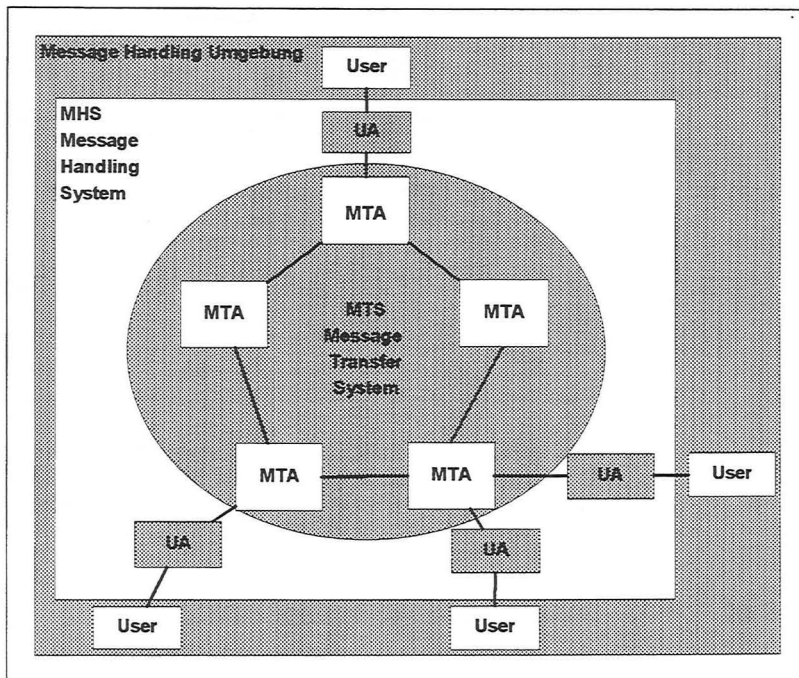
5.1.1 Das funktionale Modell

Ein Message Handling System (MHS) ist ein E-Mail System, das den Austausch von Nachrichten unterstützt. Dabei ist es unerheblich, ob diese Nachrichten zwischen „Personen“ oder Computeranwendungen ausgetauscht werden. Man unterscheidet zwischen Sendersystem und Empfängersystem. Die Übertragung der Nachrichten erfolgt nach dem „Store-and-Forward“ oder auch sogenannten „Teilstrecken-Prinzip“. Dabei ist der Nachrichtentransfer anwendungsunabhängig und inhaltsunabhängig.

Das System läßt sich gliedern in:

- eine Aufbereitungskomponente, den Useragent (UA),
- eine Transferkomponente, das Message Transfer System (MTS).

Abb. 5.1
Modell eines MHS



Useragent

Der **Useragent** bildet die Schnittstelle zwischen Benutzer (Absender und Empfänger) und der eigentlichen Transferkomponente. Er bietet dem Benutzer den sogenannten Interpersonal Messaging Service (IPMS). Dieser ermöglicht es dem Benutzer Post zu erzeugen, zu versenden und Post zu empfangen sowie Mailboxen zu unterhalten.

Message
Transfer Agent

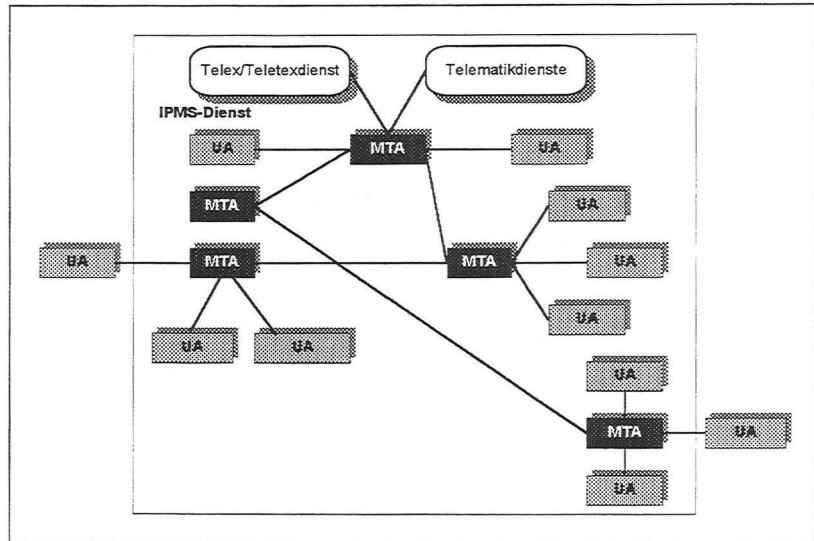
Der Zugriff der Useragents auf das Message Transfer System erfolgt über die **Message Transfer Agents (MTAs)** und umgekehrt. Dies bedeutet, daß der Transferagent die Post vom UA annimmt und dafür verantwortlich ist, daß sie ordnungsgemäß verschickt wird. Wie bei der gewöhnlichen Briefpost kann auch die elektronische Nachricht über mehrere Stationen laufen, bevor sie am Zielort zugestellt wird.

Der MTA ist auch dafür verantwortlich, daß eingehende Post für einen Benutzer in einer Mailbox zwischengespeichert wird, wenn dieser Benutzer momentan nicht erreichbar ist.

Interpersonal Messages

Das in X.400 festgelegte MHS beschränkt sich auf den Austausch sogenannter **Interpersonal Messages (IPM)**, d.h. auf den Austausch von Informationen zwischen Personen. Die Integration erfolgt dadurch, daß spezielle Useragents an das Message Transfer System angeschlossen werden. Diese IPM-UAs bereiten den Inhalt einer Nachricht des sogenannten IPM-Typs auf.

Abb. 5.2
Die Struktur von X.400



Das virtuelle Protokoll, das die Kommunikation zwischen unterschiedlichen UAs ermöglicht, wurde genormt und ist auch unter der Bezeichnung P2 bekannt. Es wird also für die Versendung von Nachrichten zwischen realen Personen verwendet. In der Abb.5.3 wird P2 durch den Nachrichtenkopf und -rumpf definiert.

5.1.2

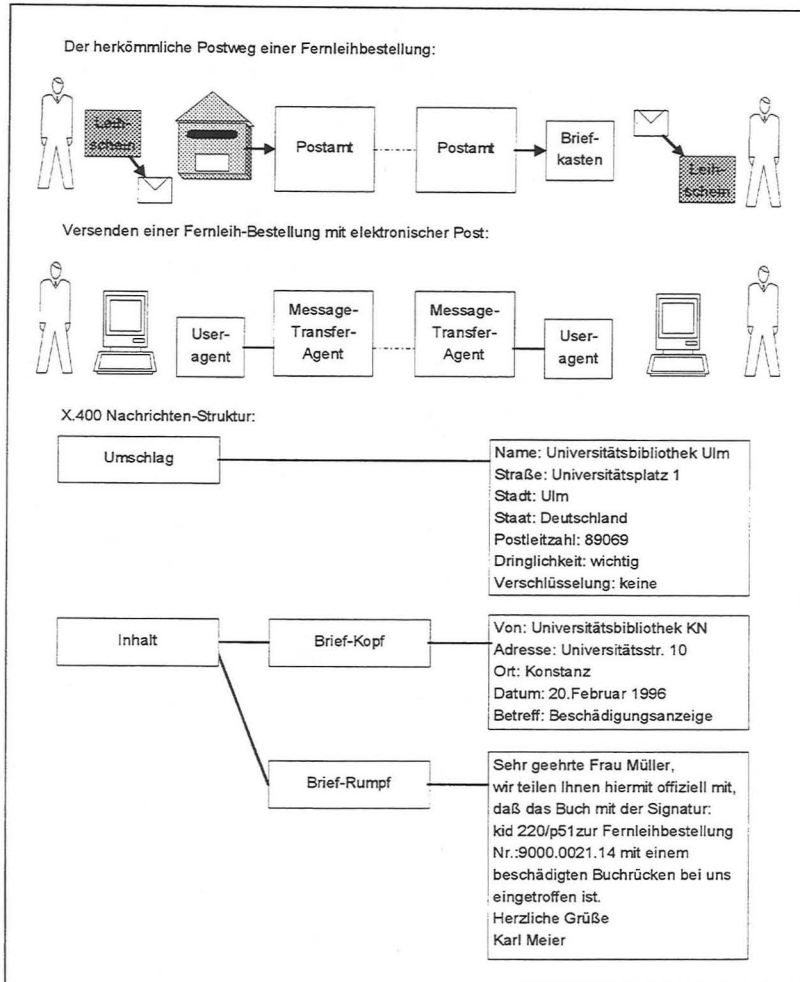
Der Nachrichtentransferdienst (MTS-Dienst)

Das MTS realisiert den Transportdienst. Dieser Dienst ist anwendungsunabhängig. Welche Art von Informationen transportiert werden, ist für das System unwichtig. Der Inhalt der Nachricht bleibt dem System verborgen. Ebenso wie beim herkömmlichen Briefverkehr der Post, ist nur die Adresse auf dem Umschlag von Bedeutung für die Übermittlung. Auch beim elektronischen Mailsystem wird also zwischen Umschlag (Envelope) und dem Inhalt (Content) einer zu übertragenden Nachricht unterschieden.

An der Schnittstelle zwischen UA und MTS werden von UAs erzeugte Nachrichten innerhalb einer bestimmten Zeit transferiert und, falls erforderlich, auch konvertiert. Die Konvertierung ist immer dann nötig, wenn der Empfänger mit einer anderen Ausrüstung als der Absender ausgestattet ist.

Zu den Nachrichtenarten, die verschickt werden, gehören z.B. ASCII-Text, analoges und digitales Fax, digitalisierte Sprache und Bilder, Bildschirmtext u.a.

Abb. 5.3
Weg und Struktur
einer X.400 Nachricht



Ablauf eines Nachrichten-transportes

Der Nachrichtentransports läuft folgendermaßen ab:

Erhält ein MTA eine Nachricht von einem UA, so erfolgt zunächst die Syntax-Prüfung. Ist die Adressierung ungültig, so wird die Nachricht mit einem Kommentar versehen an den Benutzeragenten und somit an den Absender zurückgesendet. Ist die Nachricht gültig, so wird sie mit einem Zeitstempel und einer Kennung versehen.

Zuerst erfolgt die Prüfung, ob die Zieladresse, d.h. der Benutzeragent oder die Mailbox des Empfängers lokal ist. In diesem Fall kann die Sendung sofort zugestellt, in die Zustellschlange eingereiht oder in der Mailbox des Empfängers hinterlegt werden. Wenn es gefordert ist, erfolgt eine Rückmeldung zur Bestätigung der Zustellung an den Absender. Ist die Adresse des Empfängers nicht lokal, wird die Nachricht an den nächsten Transferagenten weitergeleitet. Dabei muß internationale elektronische Post an einer Landesgrenze möglicherweise einen Gateway⁴⁴ passieren.

Ein wichtiger Teil des Transfersystems, der die Weiterleitung einer Nachricht erst ermöglicht, ist der Umschlag. Er enthält vor allem Informationen zu Adressierung, Zustellung, Konvertierung und Datensicherheit.

Abb. 5.4
Einige Umschlagfelder

Feld	Beschreibung
Absenderadresse	Postadresse des Senders
Empfängeradresse	Postadresse des Empfängers
Alternativempfänger erlaubt	Darf die Nachricht an eine weitere Person umgeleitet werden ?
Alternativempfänger	Empfänger zweiter Wahl
Nachrichtenkennung	Kennung der Nachricht
Beförderungsstufe	langsam , normal, schnell
Rückmeldeanforderung	Welche Rückmeldung will der Absender haben ?
MTA-Rückmeldeanforderung	Welche Rückmeldung will der Transferagent haben ?
Zustellung aufgeschoben bis	Darf nicht vor diesem Zeitpunkt zugestellt werden.
letztmögliche Zustellzeit	Darf nicht nach diesem Zeitpunkt zugestellt werden.
Inhalt zurück	Soll der Inhalt der Nachricht bei Nichtzustellung zurückgeschickt werden ?
Informationsart	Text, Fax, digitalisierte Stimme etc.
keine Konvertierung	Konvertierung verboten
informationsmindernde Konvertierung verboten	Es ist nur eine perfekte Konvertierung erlaubt.
Konvertierung nötig	Es muß eine Konvertierung durchgeführt werden.
Verschlüsselungskennung	Kennung für die Verschlüsselungstabelle
Vollständigkeitsprüfung	Prüfsumme des Inhalts
Absenderunterschrift	digitale Unterschrift
Vertraulichkeitsstufe	eingeschrieben, vertraulich, streng vertraulich etc.
Zustellungsbeweis	Unterschrift des Empfängers

⁴⁴ Siehe auch Abschnitt 5.2.2.1.

Die meisten Systeme pflegen ein Logbuch, welches an den Umschlag gehängt wird. Es enthält die Einträge aller Transferagenten, die die Nachricht weitergeleitet haben. Diese Einträge erleichtern die Fehlersuche und die Erkennung eines Schleifendurchlaufs. Enthält das Logbuch den Vermerk des Transferagenten, bei dem es gerade eingetroffen ist, so befindet sich die Message in einer Schleife und es können entsprechende Maßnahmen eingeleitet werden.

5.1.3 Adressierung unter MOTIS

MOTIS benutzt zur Adressierung nicht das NAME@DOMÄNE-System des Internet sondern sogenannte O/R-Namen (Originator/Recipient, Absender/Empfänger). Ein solcher Name ist eine beliebige Kette von Attributen, aus welcher der Name vom Verzeichnisdienst eindeutig herausgefunden werden kann. Der Verzeichnisdienst der Protokollnorm X.500 ermöglicht das Auffinden sämtlicher Adressen eines Landes und entspricht ebenfalls einer ISO-Norm. Er arbeitet wie ein riesiger Telefonbuch-Server. Dieser Dienst wurde durch die CCITT ausgearbeitet, da traditionell die Telefongesellschaften Auskünfte über Telefonanschlüsse erteilen.

X.500 wird an Bedeutung gewinnen; erstens ist es einfacher in der Handhabung als X.400 und zweitens ist momentan keine Alternative für diesen Standard der elektronische Post vorhanden.

Das Verzeichnis der Adressen ist hierarchisch aufgebaut und die Einträge sind anhand von Attributen und Schlüsselattributen zu ermitteln. Damit beim Versenden von internationaler Post sofort das richtige Verzeichnis durchsucht werden kann, muß unbedingt die Angabe des Landes erfolgen. Ein O/R-Name sieht folgendermaßen aus:

Beispiel einer
Adresse

L=D, FIR=UNI-KONSTANZ, ABT=EDV, NAME=HEINZ TENBRINK

Dabei kennzeichnet L das Land, FIR die Firma, ABT die Abteilung und NAME die Person, die die Message erhalten soll.

Der Nachrichtenkopf wird unter MOTIS binär kodiert und der Umschlag als Dienst durch einen ASN.1-Typ gekennzeichnet. Die gesamte Nachricht wird als APDU (Application Protocol Data Unit) verschickt.

Abschließend soll noch eine Übersicht über die unterschiedlichen Bereiche der X.400 Protokollserie gegeben werden.

Abb. 5.5
X.400
Protokollserie

CCITT-Empfehlung	Titel	entsprechende ISO-Norm
X.400	System Model - Service Elements	DIS 10021-1
X.401	Basic Service Elements and Optional User Facilities	
X.408	Encoded Information Type Conversion Rules	DIS 10021-3
X.409	Presentation Transfer Syntax and Notation	
X.410	Remote Operations and Reliable Transfer Server	
X.411	Message Transfer Layer	DIS 10021-4
X.420	Interpersonal Messaging User Agent Layer	DIS 10021-7
X.430	Access Protocol for Teletex Terminals	
(X.500)	The Directory - Overview of Concepts, Models and Services	

5.2 TCP/IP

5.2.1 Vergleich der Protokolle mit dem ISO/OSI Standard

Geschichtlicher Hintergrund

1969 stellte das DoD (United States Department of Defense) Experimentalnetzwerk vor, das ARPANET. Dieses Netz war Ergebnis von Forschungsprojekten der ARPA (heute DARPA), der (Defense) Advanced Research Projects Agency des Amerikanischen Verteidigungsministeriums. Bald entwickelte sich aus diesem Experiment ein landesweites Netz zur Rechner-Rechner-Kommunikation. Mit Hilfe der dort verwendeten Technologie entstanden weitere Netze z.B. das MILNET, ein militärisches Netz mit strengen Sicherheitsvorschriften, und auch das ARPA internet.

Als Protokoll bei Entwurf und Realisierung dieser Netze wurde die Internet Protocol Suite verwendet. Wichtige Protokolle darin sind das Internet Protocol (IP) und das Transmission Control Protocol (TCP). Die Internet Protocol Suite ermöglicht auch die Datenkommunikation in heterogenen Netzen.

Die Internet-Protokoll-Familie basiert ebenso wie OSI auf einem Schichtenmodell. Die Schichten beider Standards lassen sich jedoch nicht eindeutig miteinander vergleichen. Dennoch soll in der folgenden Abbildung eine Zuordnung vorgenommen werden, um zu verdeutlichen, in welcher Ebene TCP und IP angesiedelt sind.

Abb. 5.6
Vergleichende Zu-
ordnung der TCP/IP
Protokolle zum OSI
Schichtenmodell

ISO/OSI-Schichtenmodell	Internet Protokoll Familie		
7. Anwendung	Telnet		
6. Darstellung	FTP	HTTP	
5. Sitzung	SMTP		
4. Transport	TCP	UDP	
3. Netzwerk	IP	ARP	ICMP
2. Sicherung	Ethernet (LAN)		
1. Bitübertragung	oder X.25 (WAN)		

Die TCP/IP-Protokolle werden sowohl im lokalen Bereich (LAN) als auch im Fernbereich (WAN) eingesetzt, um die Kommunikation zwischen Rechnern unterschiedlicher Betriebssysteme zu ermöglichen. Weitere Vorteile dieser Protokollfamilie liegen in der Wirtschaftlichkeit, ihrer Robustheit und dem dezentralen Prinzip.

Das Vermittlungsprotokoll IP arbeitet als bytestrom-orientierter Datagramm-dienst und wurde für die Verbindung zwischen der riesigen Anzahl von lokalen Netzwerken und Weitverkehrsnetzen innerhalb des ARPA Internet entwickelt. Das OSI-Referenzmodell hat sich mit der Kommunikation zwischen unterschiedlichen Netzen anfangs wenig beschäftigt, während bei der Entwicklung von IP darin die Hauptaufgabe bestand.

Der Transportschicht von ISO/OSI entspricht etwa die Zuordnung der beiden Protokolle:

- TCP (Transmission Control Protocol) für den verbindungsorientierten Datentransfer,
- UDP (User Datagramm Protocoll) für den verbindungslosen Datagramm-dienst.

Die folgenden Applikationsprotokolle sind mit den oberen Schichten 5-7 des OSI-Modells vergleichbar.

- FTP und TFTP (File Transfer Protocol und Trivial File Transfer Protocol) dienen dem Dateitransfer,
- TELNET (Telnet Protocol) ermöglicht die Terminalemulation bzw. Remote Login,
- SMTP (Simple Mail Transfer Protocol) ist ein Protokoll zum Senden und Empfangen elektronischer Post und entspricht der Protokollserie X.400,
- HTTP (Hypertext Transfer Protocol) baut auf der TCP/IP Verbindung auf und ermöglicht z.B. die schnelle Suche von Einträgen innerhalb des World Wide Web mit Hilfe des Datenformats HTML (Hypertext Markup Language).

5.2.2 Internet Protocol

Das heute bekannte Vermittlungsprotokoll IP wurde in den frühen 80er Jahren eingeführt und wird in der Regel zusammen mit dem ARPANET-Transportprotokoll TCP genutzt.

Die Transportschicht nimmt zu übertragende Nachrichten auf und unterteilt sie in Datagramme von jeweils bis zu 64 K. Diese Datensegmente werden mit der Internetadresse des gewünschten Zielrechners vom TCP an das IP weitergereicht. Für IP ist der Inhalt der Segmente transparent. Die Segmente werden als Datagramme durch das Internet Protokoll verschickt. Das bedeutet, die Übertragung der Pakete erfolgt verbindungslos und ohne Bestätigung.

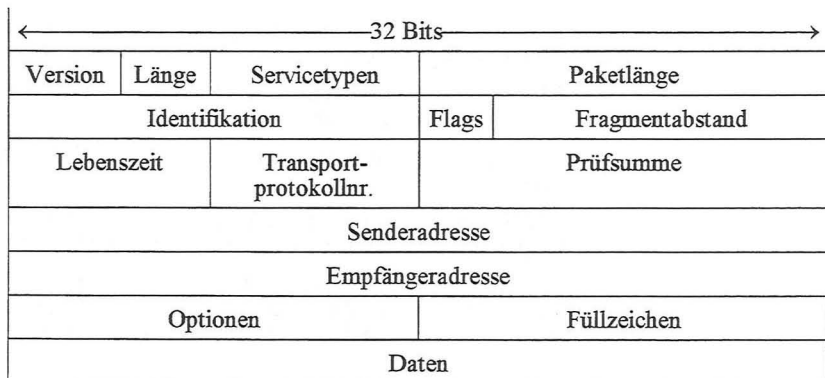
Dabei kommt die Hauptaufgabe des Internet Protokolls der Auswahl der Route zwischen Quellknoten und Zielknoten im Netz zu. Damit ist IP eindeutig der Netzwerk-Schicht zuzuordnen.

IP übernimmt keine Garantie für die Einhaltung der Reihenfolge der Paketversendung, und auch die Ablieferung beim Empfänger wird nicht zugesichert. Da eine Route über unterschiedliche Netze führen kann, benötigen die Übertragungssysteme Informationen, um das Datagramm weiterbefördern zu können. Diese nötigen Angaben sind im IP-Header enthalten, der jedem Segment hinzugefügt wird.

Es kann die Notwendigkeit bestehen, daß Datagramme aufgeteilt werden müssen, wenn sie für die Übertragung wegen bestimmter Hardware- oder Software-Restriktionen zu groß sind. In diesem Fall wird ein Datagramm in Fragmente aufgeteilt. Ein so entstandenes Fragment wird wie ein eigenständiges Datagramm übertragen. Anhand einer Kennung wird am Ziel das Datagramm als Fragment identifiziert und mit den anderen eintreffenden Fragmenten zusammengesetzt.

Das IP-Datagramm besteht aus einem Kopf- und einem Textteil. Der Kopfteil besitzt einen festen Teil mit 20 Bytes und einen optionalen Teil von variabler Länge. Der Header enthält neben den Internetadressen von Sendersystem und Empfängersystem verschiedene andere Felder:

Abb. 5.7
IP-Header



Das Feld **Version** hält fest, welcher Version des Protokolls das Datagramm gehört. Da die Länge des Nachrichtenkopfes nicht konstant ist, wird diese im Feld **Länge** beschrieben. Der minimale Wert der Header-Länge ist 5.

Das Feld **Servicetyp** oder Dienst-Art erlaubt dem Host⁴⁵, den gewünschten Dienst vom Subnet abzurufen. Dabei gibt es die Möglichkeit, verschiedene Kombinationen von Zuverlässigkeit und Geschwindigkeit zu nutzen. So ist für die Übertragung von digitalisierter Sprache oder bewegten Bildern die Geschwindigkeit wichtiger als die Zuverlässigkeit, während beim Datentransfer die Zuverlässigkeit entscheidender ist als die Übertragungsdauer.

In der **Paketlänge** wird die Gesamtlänge des Datagramms angegeben, sie beträgt maximal 65536 Bytes.

Das Feld **Identifikation** erlaubt dem Ziel-Host eine Zuordnung eines neu eingetroffenen Fragments zu einem Datagramm. Alle Fragmente eines Datagramms enthalten den gleichen Identifikationswert.

Die **Flags** bestehen aus einem unbenutzten und zwei genutzten Bits. Diese geben an, ob das Datagramm überhaupt fragmentiert werden darf und ob es sich bei dem übertragenen Paket um ein Fragment handelt, dem weitere folgen.

Der **Fragmentabstand** gibt an, welche Stelle das vorliegende Fragment im Datagramm einnehmen muß. Alle Fragmente eines Datagramms, außer dem letzten, müssen aus einem Vielfachen von 8 Bytes bestehen. Diese 8 Bytes sind die Grundfragmenteinheit.

Die **Lebenszeit** begrenzt die Paketlebensdauer. Die maximale Lebensdauer beträgt 255 Sekunden. Bei jeder Weitervermittlung des Pakets durch einen Netzknoten wird der Zähler dekrementiert. Steht der Zähler auf Null, dann wird das Paket aus dem Netz entfernt.

An der **Protokollnummer** erkennt IP, an welches übergeordnete Transportprotokoll das Segment weitergereicht werden muß.

Die **Prüfsumme** ermöglicht die Kontrolle, ob der IP-Header ordnungsgemäß übertragen wurde. Die IP-Prüfsumme kontrolliert ausschließlich die korrekte Übertragung des IP-Headers und ist sinnvoll, weil sich der Nachrichtenkopf an einem Gateway⁴⁶ ändern könnte.

Das Feld **Optionen** wird für die Sicherung, die Quellen-Leitwegbestimmung, für Fehlermeldungen, für das Debugging, zur Zeitstempelung und für andere Informationen verwendet, die nicht immer gebraucht werden.

Die Verwendung von Router und Gateway

Die kommunizierenden Netze sind über sogenannte Router miteinander verbunden. Router verbinden Netze, in denen dasselbe Übertragungsprotokoll verwendet wird und selektieren die Datenblöcke nach Internet-Adressen. Die Adressierung erfolgt mit Hilfe einer zweiteiligen 32 Bit Integer Größe. Der erste Teil stellt die Netzadresse, der zweite die Rechneradresse dar. Man spricht in diesem Fall von dem NAME@DOMÄNE-System des Internet.

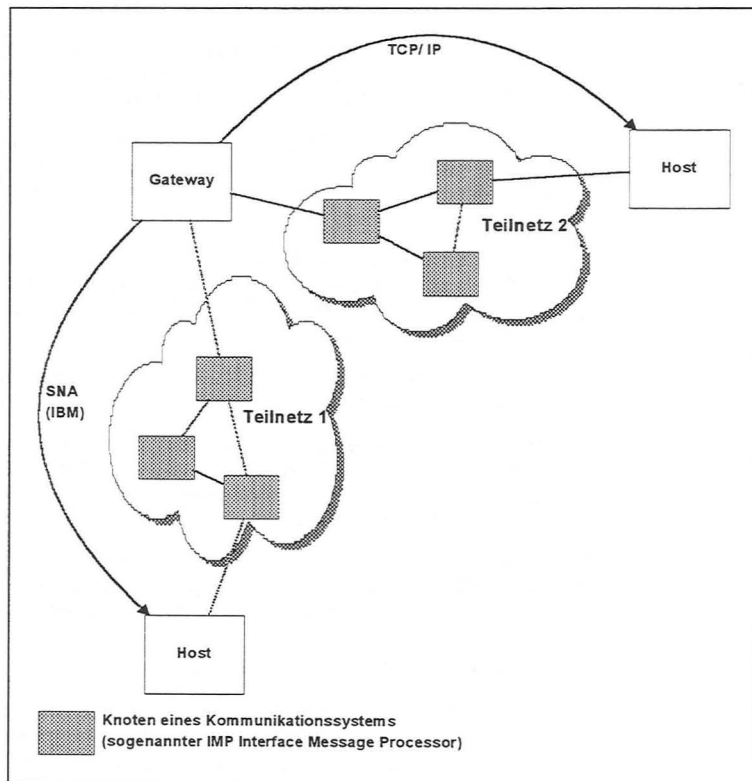
⁴⁵ Unter Host wird hier die Maschine im Netz verstanden, auf der ein Anwendungsprogramm läuft.

⁴⁶ Der Gateway wird im nächsten Abschnitt erläutert.

Gateways ermöglichen die Kommunikation zwischen Netzen mit unterschiedlichen Übertragungsprotokollen auf den Schichten größer oder gleich der Schicht 3.

In den Teilnetzen werden die Datagramme nach den im jeweiligen Subnet gültigen Protokollen übermittelt und erhalten zusätzlich zum IP-Header noch einen Netz-Header (z.B. Ethernet). Muß ein IP-Datagramm in ein anderes Subnet übertragen werden, wird es an einen Gateway⁴⁷ geschickt. Dieses Gateway fungiert im einen Netz als Ziel-Host im anderen als Quell-Host. In diesem Gateway wird der alte Netz-Header entfernt und ein Header für das nächste Teilnetz anhand der Informationen im IP-Header generiert. Der ursprüngliche IP-Header bleibt bestehen. Ist das Datagramm größer, als im nächsten Teilnetz zugelassen, dann wird an dieser Stelle auch noch eine neue Fragmentierung vorgenommen.

Abb. 5.8
Kommunikation zwischen Teilnetzen mit Gateway



Die Überprüfung der Nachrichtenübertragung erfolgt mit Hilfe des ICMP (Internet Control Message Protocol). Dieses Protokoll beschreibt Paketformate für Fehler- und Statusmeldungen, die für das Netzwerkmanagement benötigt werden.

⁴⁷ Es gibt tatsächlich zwei Bedeutungen von Gateway. Unter einem Gateway wird auch einfach der Zugang oder die Verbindung zu einem anderen Netz verstanden.

5.2.3 Transmission Control Protocol

5.2.3.1 Funktionsweise

Da TCP den Datentransport zwischen den jeweiligen Anwendungsprozessen organisiert, ist es der Transport-Schicht zugeordnet. Das Transmission Control Protocol ist ein verbindungsorientiertes End-to-End Protokoll, das die zu übertragenden Daten in nummerierte Datenblöcke, sogenannte Segmente zerlegt.

Die Transportinstanzen des TCP akzeptieren beliebig lange Nachrichten von Benutzerprozessen, die als Bytestrom vorliegen. Die Nachrichten werden in Teile mit einer maximalen Länge von 64 K zerlegt. Ein solcher Teil wird als eigenständiges Datagramm verschickt.

Da die Vermittlungsschicht nicht garantiert, daß die Teile korrekt abgeliefert werden, ist es die Aufgabe des TCPs, die Datagramme bei Verlust erneut zu übertragen und diese nach ihrer Ankunft wieder in der richtigen Reihenfolge zu Nachrichten zusammensetzen.

Die Informationen, die zur Datenübertragung benötigt werden, befinden sich im Header des zu übertragenden Datenblocks. Er ist mindestens 20 Bytes groß und enthält u. a.:

- Quell-Portnummer,
- Ziel-Portnummer,
- Sequenznummer,
- Quittungsnummer (Piggyback acknowledgement),
- Fenstergröße (Window),
- Prüfsumme (Checksum),
- Dringlichkeitszeiger (Urgent Pointer),
- die eigentlich zu übertragenden Daten.

Portnummer

Die **Portnummern** dienen dazu, unterschiedliche Prozesse innerhalb eines Systems anzusprechen, und ermöglichen es, Speicherbereich oder Datenblöcke zu den Verbindungen dieser Prozesse zuzuordnen. Eine Transportverbindung wird durch ihre beiden Endpunkte definiert, die als „Sockets“ bezeichnet werden. Ein solcher Socket⁴⁸ wird durch die Internetadresse des Host und eine dort vergebene Portnummer eindeutig identifiziert. TCP-Verbindungen sind grundsätzlich duplexfähig. Die in jeder Richtung übertragenen Daten werden als kontinuierlicher Bytestrom aufgefaßt.

Sequenznummer

Die **Sequenznummer** oder Folgennummer kennzeichnet daher jedes Datenbyte, das TCP überträgt. Die Sequenznummer des ersten Bytes in einem Datagramm wird in dessen TCP-Header vermerkt. Es wird also nicht die Anzahl der Segmente in der Sequenznummer hinterlegt, sondern die Anzahl bereits übertragener Bytes. Die Sequenznummer ist eine 32 Bit Integer Größe. Sie ermöglicht dem Empfänger, die Segmente wieder in der richtigen Reihenfolge zusammensetzen.

⁴⁸ Siehe Abschnitt 5.2.4.

Unterstützte Kommunikationssysteme

- Quittungsnummer** Der Empfang eines Datenblocks wird durch eine Quittung bestätigt. Dazu dient die **Quittungsnummer**. Ist das Flag ACK gesetzt, so wird das empfangene Segment als Quittung angesehen. Dabei wird nicht die Anzahl oder Nummer der empfangenen Segmente bestätigt, sondern die Anzahl der empfangenen Bytes in der Quittungsnummer übertragen. Erhält ein Sender keine Empfangsbestätigung in einer bestimmten Zeit, wiederholt er die Sendung der bis dahin unquitierten Segmente.
- Fenstergröße** Würde man vor jeder Sendung eines weiteren Segments die Empfangsbestätigung abwarten, könnte dies die Übertragung unnötig verzögern. Wenn dagegen eine große Zahl von Datenblöcken unquitiert gesendet würde, könnte ein schneller sendender Rechner einen langsameren empfangenden Rechner mit Datagrammen überfluten. Mit der **Fenstergröße** wird die maximal zu verschickende Datenmenge vereinbart, die unquitiert gesendet werden darf. Mit der Quittung versendet ein Rechner die Byte-Anzahl, die er aktuell verarbeiten kann, in diesem 16 Bit Feld. Die Flußsteuerung erfolgt also individuell und kann jederzeit angepaßt werden.
- Prüfsumme** Mit der **Prüfsumme** wird kontrolliert, ob das gesamte Segment korrekt übertragen wurde. Der Prüfsummenalgorithmus besteht darin, daß alle Daten als 16-Bit-Wörter addiert werden und dann das Einerkomplement dieser Summe gebildet wird. Stimmt die Prüfsumme bei Sender und Empfänger überein, so gilt die Übertragung als korrekt ausgeführt. Andernfalls muß die Sendung wiederholt werden.
- Dringlichkeitszeiger** Der **Dringlichkeitszeiger** gibt an, im wievielten Byte des Pakets Daten beginnen, die für den Empfänger als besonders dringend gekennzeichnet sind. Das gesetzte Flag URG zeigt an, daß ein Datagramm solche Vorrangdaten enthält.

Abb. 5.9
TCP Header

Quell-Port				Ziel-Port				
Sequenznummer								
Quittungsnummer								
Daten- Abstand	Reser- viert	U R G	A C K	P S H	R S T	S Y N	F I N	Fenstergröße
Prüfsumme				Dringlichkeitszeiger				
Optionen				Füllzeichen				
Daten								

5.2.3.2

TCP-Verbindungsaufbau

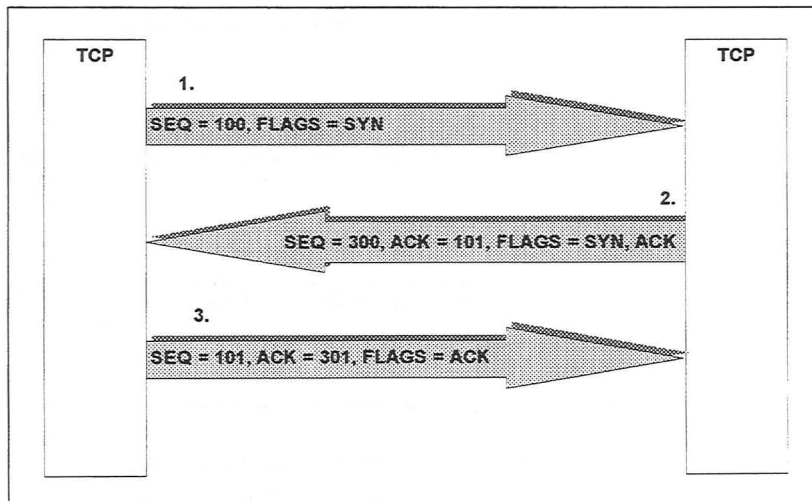
Three Way
Handshake

Der Verbindungsaufbau zwischen zwei TCP-Modulen erfolgt nach dem so bezeichneten **Three Way Handshake**.

1. Der TCP-Client, das ist jenes Modul, welches die Verbindung wünscht, sendet an den TCP-Server ein Segment mit der Sequenznummer (Initial Sequence Number), die zur Kennzeichnung der ersten Antwortdaten benutzt werden soll. Durch das Flag SYN, das im TCP-Header gesetzt wird, wird der Wunsch zur Verbindungsaufnahme deutlich gemacht.
2. Der Server antwortet mit einer Quittung, d.h. durch das Flag ACK wird signalisiert, daß die Quittungsnummer gesetzt ist. Außerdem enthält die Antwort die Sequenznummer, die der Client erwartet, also die oben genannt Initial Sequence Number. Zusätzlich wird auch vom Server eine Initial Sequence Number übertragen, die dann der Client für die Kennzeichnung seines nächsten Segments verwenden soll.
3. Der Client antwortet mit einer weiteren Quittung, in der sich außer der Quittungsnummer, die vom Server erwartete Sequenznummer und die als nächstes vom TCP-Server für eine Antwort zu verwendende Sequenznummer befindet.

Die Verbindungsaufnahme ist damit erfolgreich abgeschlossen. Nun informiert der Client die höheren Schichten seines Systems über die Verbindungsaufnahme. Auch der Server tut dies nach Empfang des Quittungssegments.

Abb. 5.10
Beispiel für einen
Verbindungsaufbau



Im vorliegenden Beispiel kennzeichnet der TCP-Client seinen Verbindungswunsch mit dem Flag SYN und überträgt die Sequenznummer 100.

Diese wird vom TCP-Server mit der Quittungsnummer ACK = 101 und dem gesetzten Flag ACK quittiert. Auch der TCP-Server kennzeichnet seinen Verbindungswunsch mit dem Flag SYN. Seine Sequenznummer, die der Client zuerst quittieren soll, lautet 300.

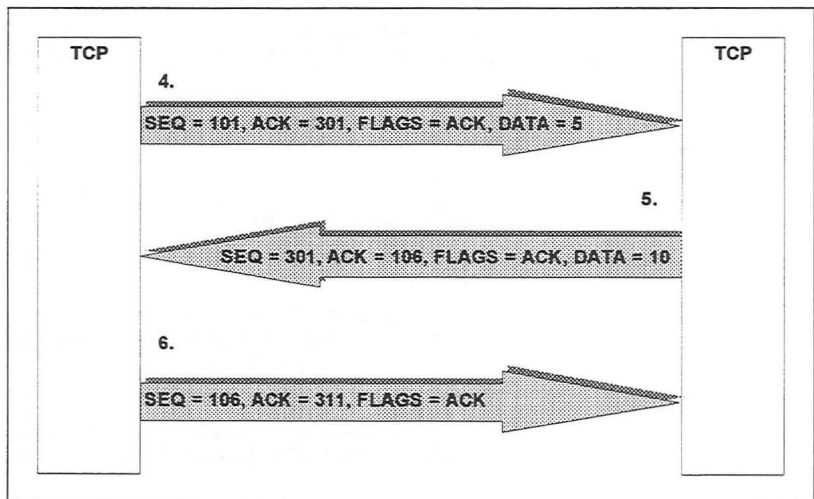
Der TCP Client quittiert die Sequenznummer 300 mit der nächsten erwarteten, also mit ACK = 301, und bestätigt so den Erhalt der Nachricht des TCP-Servers.

5.2.3.3 TCP-Datenaustausch

Wenn ein höheres Protokoll eines Client ein zu versendendes Datenpaket an das TCP-Modul übergibt, verpackt dieses die Daten in ein Segment. Die Sequenznummer bezeichnet dabei die Nummer des ersten Bytes dieses Segments im Datenstrom. Der Server prüft die empfangene Sequenznummer daraufhin, ob diese mit der nächsten erwarteten Sequenznummer übereinstimmt. Damit kann der Verlust von Segmenten erkannt werden. Der Server sendet an den Client die Bestätigung des Datenerhalts, indem er mit dem Flag ACK die Sequenznummer des nächsten erwarteten Bytes überträgt.

Geht eine Quittung verloren, sendet einer der Prozesse nach einer festgelegten Zeitspanne das letzte zu übertragende Segment erneut. Der Empfänger kann anhand der Sequenznummer erkennen, daß er das Original bereits erhalten hat. Die Bestätigung muß er aber nochmal senden.

Abb. 5.11
Beispiel für einen
Datenaustausch über
TCP



In diesem Beispiel wird der Datenaustausch deutlich.

Der TCP-Client sendet an den Server 5 Bytes. Da vom Server keine Daten übertragen wurden, quittiert er nochmals mit der bis dahin gültigen Quittungsnummer ACK = 301.

Jetzt überträgt auch der TCP-Server 10 Bytes und bestätigt mit der Quittungsnummer ACK = 106 (101 + 5 Bytes) den Empfang der Daten des TCP-Client.

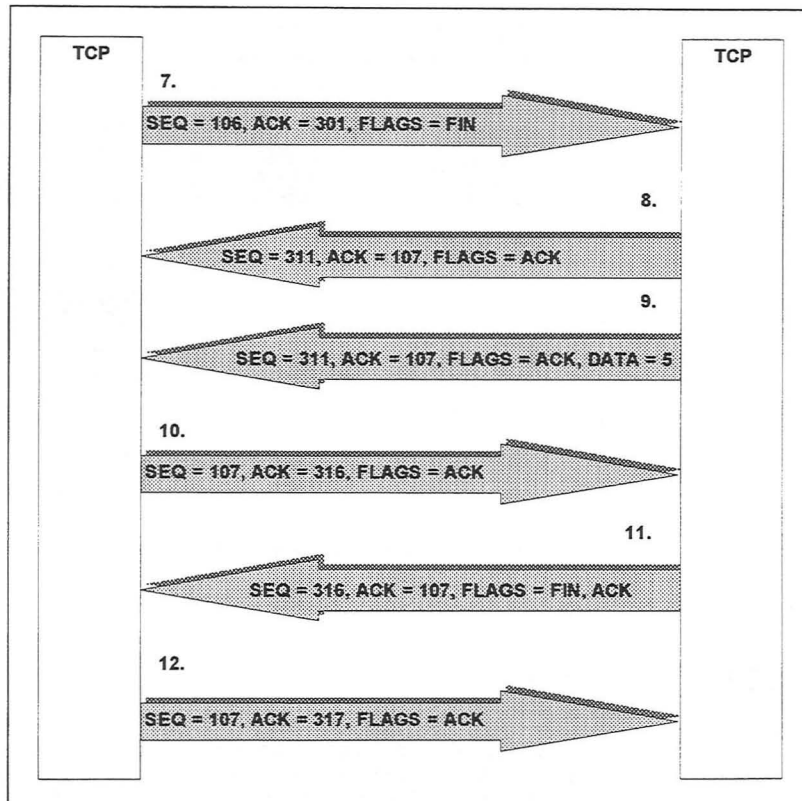
Der TCP-Client quittiert im folgenden den Erhalt der 10 Bytes des Servers mit ACK = 311 (301 + 10 Bytes).

5.2.3.4

TCP-Verbindungsabbau

Die Verbindung wird für beide Übertragungsrichtungen getrennt abgebaut. Das Flag FIN signalisiert den Wunsch zum Verbindungsabbau. Auch das Kommando zum Verbindungsabbau muß quittiert werden.

Abb. 5.12
Beispiel für den
getrennten Verbindungsabbau



In diesem Beispiel soll der Verbindungsabbau erläutert werden.

Der Client meldet den Wunsch auf Verbindungsabbau mit dem Flag FIN an den Server.

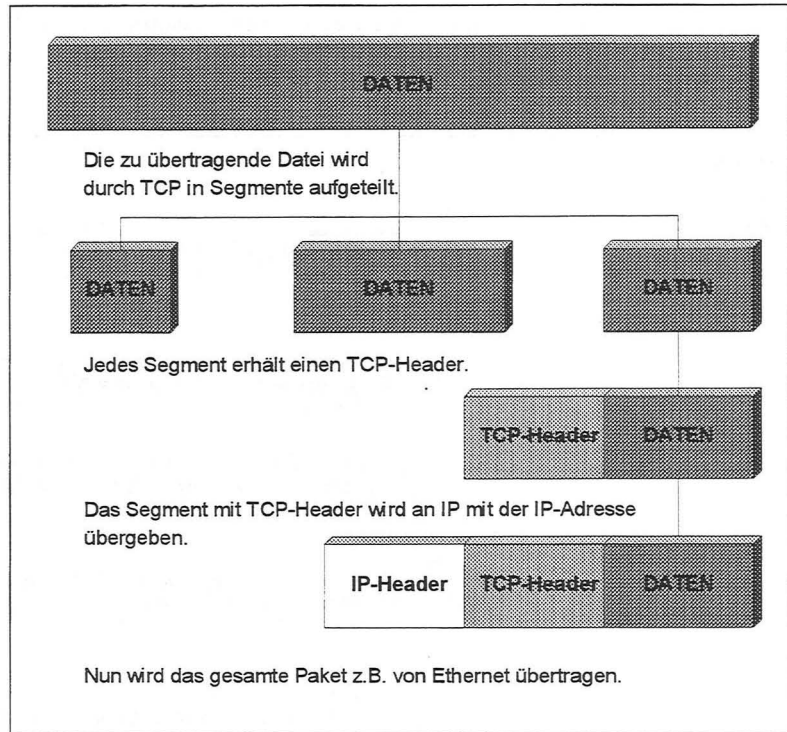
Der TCP-Server bestätigt den Erhalt des Wunsches mit dem Quittungswert ACK = 107. Er kann jedoch weiter Daten übertragen. So werden noch einmal 5 Bytes an den Client übertragen.

Der TCP-Client quittiert den Erhalt der Daten.

Nun sendet auch der TCP-Server das Flag FIN, das den Verbindungsabbau kennzeichnet, an den Client. Der Client bestätigt den Empfang und damit ist die Verbindung auf beiden Seiten geschlossen.

Die folgende Abbildung faßt die Datenübertragung bei der Verwendung von TCP und IP noch einmal zusammen.

Abb. 5.13
Dateiübertragung mit
TCP/IP



5.2.4 Sockets

Sockets sind eine Programmierschnittstelle (API Application Programming Interface), die zur Entwicklung netzfähiger Applikationen verwendet wird. Sie stellen die Verbindung zwischen Betriebssystem und Benutzerprozeß her. Ursprünglich waren Sockets Bestandteil des Berkeley UNIX BSD-Systems, um dort die Kommunikation über TCP/IP Netze zu realisieren. Vorrangig werden sie benutzt, um virtuelle Verbindungen und Client-Server-Modelle zu realisieren. Heute sind sie auch als API für Personal Computer verfügbar (WINSOCK = window sockets).

Sockets behandeln die Interprozeß-Kommunikation als verallgemeinerten Dateizugriff. Der Mechanismus unterstützt verbindungsorientierte und verbindungslose Kommunikationsbeziehungen.

Eine Gruppe von Systemcalls ermöglicht den Zugriff auf den Transportdienst. Diese sollen im folgenden behandelt werden, da sie auch in der Client-Server-Architektur des Fernleihsystems Verwendung finden.

socket

Der Systemcall `socket` erzeugt einen Socket, also eine bestimmte Datenstruktur innerhalb des Betriebssystems. Die Parameter dieses Aufrufs definieren das Adressenformat (die jeweilige Internet-Adresse), den Sockettyp (verbindungsorientiert oder verbindungslos) und das Protokoll (TCP/IP).

bind

Damit ein entfernter Benutzer eine Verbindungsanforderung an den wartenden Socket senden kann, muß dieser einen Namen oder eine Adresse besitzen. Mit dem Systemcall `bind` wird dem Socket ein Name zugewiesen. Dieser Name wird nun bekanntgegeben, damit entfernte Prozesse den Socket adressieren können.

listen

Um eingehende Verbindungsanforderungen speichern zu können, benötigt der Socket Pufferplatz. Dieser wird ihm mit der Anweisung `listen` zugewiesen. Ein Socket, der auf diese Weise Pufferplatz erhält, wird zum passiven Endpunkt. Er wartet nun auf ankommende Verbindungsanforderungen.

Abb. 5.14
Beispiel für das
Einrichten eines
passiven Sockets

```
int einrichten(portnum)
u_short portnum;
{
    char myname[MAXHOSTNAME+1]
    int flag, size, type;

    /* Feststellen, wie der Hostname des aktuellen */
    /* Prozessors lautet */
    gethostname(myname, MAXHOSTNAME);

    /* Feststellen, wie unsere Hostadresse lautet */
    hp=gethostbyname(myname);
    /* Im Fehlerfall Rückgabewert liefern */
    if(hp==NULL)
        return(-1);

    /* Vereinbarungen über die Adressierung treffen */
    sa.sin_family=hp->h_addrtype;
    /* Portnummer ist eine 16 Bit Größe */
    sa.sin_port=htons(portnum);

    /* ein Socket wird bytestrom-orientiert und */
    /* duplexfähig angelegt, Descriptor ist old_soc */
    if((old_soc=socket(AF_INET, SOCK_STREAM, 0))<0)
        return(-1);

    /* Optionen für den Socket setzen: */
    /* - zyklisch soll die Verbindung geprüft werden */
    /* - bei Verbindungsabbruch soll ein bestimmtes */
    /* Signal gesendet werden */
    size=sizeof(int);
    type=1;
    flag=setsockopt(old_soc,SOL_SOCKET, SO_KEEPALIVE,\
                    (char*)&type, size);
    flag=setsockopt(old_soc,SOL_SOCKET, SO_REUSEADDR,\
                    (char*)&type, size);
    if(flag)
        return(-1);

    /* dem Socket werden die Werte aus „sa“ zugewiesen*/
    /* also Hostadresse und Portnummer */
    if(bind(old_soc, &sa, sizeof sa, 0)<0)
        return(-1);

    /* Pufferplatz für 9 Verbindungen wird zugewiesen */
    listen(old_soc, 9);
    return(old_soc);
}
```

Im obigen Programmbeispiel wird ein Socket für einen Port des Fernleihsystems eingerichtet, der Socket wird bekanntgegeben und als passiver Endpunkt eingerichtet, damit er Verbindungsanforderungen von Prozessen auf anderen Systemen annehmen kann. In diesem Fall dürfen maximal 9 Anfragen in eine Warteschlange für den Prozeß gestellt werden.

accept

Der Benutzerprozeß hängt sich an den Socket an und wartet auf eine eingehende Verbindungsanforderung. Dies geschieht mit dem Aufruf des Systemcalls **accept**. Ist bereits eine Anforderung vorhanden, so wird diese aus der Warteschlange des Sockets genommen. Ist noch keine Anforderung eingetroffen, blockiert der Prozeß, bis eine Anforderung ankommt, es sei denn, eine Blockierung ist für diesen Socket ausgeschlossen. Sobald eine Anforderung ansteht, wird ein neuer Socket erzeugt, der wieder als Verbindungsendpunkt dient. So kann ein einziger Anschluß dazu benutzt werden, mehrere Verbindungen aufzubauen.

Abb. 5.15
Beispiel für das
Warten auf eine
Verbindungsanforderung über den
eingeschalteten
Socket

```
int verbinden()
{
    int i, u;
    i = sizeof(isa);

    /* Name des Socket ermitteln, dessen          */
    /* Descriptor old_soc ist                      */
    getsockname(old_soc, &isa, &i);

    /* Warten auf eine Verbindungsanforderung    */
    akt_soc = accept(old_soc, &isa, &i);

    return(akt_soc);
}
```

connect

Um eine Verbindung zu einem entfernten Socket aufzunehmen, wird der Systemcall **connect** aufgerufen. Dabei werden der lokale Socket und der Name des Zielsystems als Parameter angegeben.

In einer verbindungsorientierten Kommunikation spricht man von einem Client, wenn der entsprechende Prozeß eine Verbindung aufbaut. Der Prozeß, der auf eine derartige Anfrage antwortet oder sie nur passiv empfängt, wird als Server bezeichnet. Im folgenden Programmbeispiel wird eine solche Verbindungsanforderung an ein Zielsystem über einen Socket vereinfacht dargestellt.

Abb. 5.16
Beispiel für die Ver-
bindungsaufnahme
mit dem Systemcall
connect

```

int tcp_open(portnr,hoststr) /* TCP-Verbindung öffnen */
int portnr;
unsigned char *hoststr;
{
    int flag;

    /* Feststellen, wie die Hostadresse lautet */
    host_ptr = gethostbyname(hoststr);
    ...
    /* Zieladresse umsetzen */
    dest.sin_family = AF_INET;
    dest.sin_port = htons((u_short)portnr);
    memcpy((Char *)&dest.sin_addr,\
           (char*)host_ptr->h_addr,host_ptr->h_length);

    /* Socket bytestrom-orientiert, duplexfähig */
    /* anlegen */
    sock = socket(AF_INET, SOCK_STREAM, 0);

    /* Ist der Socket in Ordnung? */
    if(sock < 0)
    {
        perror("socket");
        return(12);
    }
    ...
    /* Anwählen der TCP-Verbindung über den Socket */
    flag = connect(sock, (struct sockaddr *) &dest,\
                  sizeof(dest));

    /* Steht die Verbindung? */
    if(flag < 0)
    {
        perror("connect");
        return(13);
    }
    return(0);          /* Verbindung zum Server OK */
}

```

shutdown

Die Verbindung über einen Socket beendet der Systemcall **shutdown**. Die beiden Richtungen einer Duplex-Verbindung können unabhängig voneinander abgeschaltet werden. Im folgenden Programmbeispiel ist der Verbindungsabbau dargestellt.

Abb. 5.17
Beispiel für das
Schließen der
Verbindung

```
int lan_close()
{
    if(old_soc)
    {
        /* Verbindung abbauen      */
        shutdown(old_soc, 0)

        /* Socket schließen        */
        close(old_soc);
    }

    if(akt_soc)
        close(akt_soc);
    akt_soc=0;
    old_soc=0;
}
```

send und recv

Nachrichten werden mit **send** gesendet und mit **recv** empfangen.

select

Mit dem Aufruf **select** kann der Prozeß die Blockierung steuern. Ein hat Prozeß mehrere Verbindungen aufgebaut und möchte den Aufruf **recv** an den Verbindungen ausgeben, an denen Nachrichten für ihn angekommen sind. Da er nicht weiß, an welchen Verbindungen tatsächlich Nachrichten warten, blockiert er möglicherweise bei einer Verbindung, bis eine Anforderung für ihn ankommt und an anderen Verbindungen könnten Nachrichten bearbeitet werden. Mit **select** blockiert er in diesem Fall nur solange, bis an einer anderen Verbindung eine Anforderung für ihn ankommt. Voraussetzung ist, daß in der Parameterliste des Systemcalls die Verbindung aufgeführt ist.

Für die Systemcalls **send**, **recv** und **select** soll an dieser Stelle kein Programmbeispiel gegeben werden. Zum Vergleich mit einem konventionellen Telefonat kann jedoch die folgende Übersicht dienen:

Client		Server	
socket:	Anschluß einrichten	socket:	Anschluß einrichten
		bind:	Anschluß im Telefonbuch anmelden
		listen:	Warten auf Anruf
connect:	wählen	accept:	Hörer abnehmen
send:	sprechen	recv:	hören

6

Client-Server-Architektur

Die Hauptaufgaben, die sich aus der Nutzung des Fernleihsystems der Universitätsbibliothek Konstanz ergeben, sind:

- Dialog des Systems mit dem Benutzer,
- Verwaltung der Fernleihdaten,
- Datenaustausch mit anderen Bibliotheken nach dem ILL-Standard.

Für diese unterschiedlichen Bereiche des Fernleihsystems wurde eine Client-Server-Architektur definiert, um eine hohe Portabilität und Modularität zu gewährleisten. Ein weiterer Vorteil dieses Systemaufbaus ist eine leichte Testbarkeit und Wartbarkeit des Systems. Durch die Modularisierung wird wesentlich die Wiederverwendbarkeit unterstützt. Schnittstellen zu anderen Bibliothekssystemen sind leichter realisierbar.

Bei der Programmierung ist auf leichte Portierbarkeit geachtet worden. Die Implementierung erfolgte unter UNIX komplett in ANSI-C. Für den Benutzerdialog am PC ist alternativ ein DOS-Client realisiert worden.

Die Kommunikation mit anderen Systemen erfolgt in der Regel über TCP/IP Sockets⁴⁹, kann aber auch über das X.400 Message-Handling-System⁵⁰ (MHS) abgewickelt werden. Der Datenaustausch über das MHS wird z.Zt. nicht unterstützt, im Programmablauf wird er aber berücksichtigt.

Die Aktualisierung der Fernleihdaten im lokalen Verbuchungssystem erfolgt mit Hilfe des Verbuchungsservers, der jedoch nicht mehr zum eigentlichen Fernleihsystem gehört.

Die Routinen für Datenbankzugriffe sind ausschließlich in Embedded-SQL realisiert. So kann jede moderne relationale Datenbank eingesetzt werden.

6.1

Module des
Fernleihsystems

Die Server-Programme

Folgende Komponenten wurden für das Fernleihsystem implementiert:

- Monitormodul (fesys.c),
- Dialogserver (fserver.c),
- Verwaltungsserver (zserver.c),
- Sendeserver (sserver.c),
- Empfangsserver (eserver.c),
- Dialog-Client (fclient.c, fclientg.c, fclientn.c, fclientg.c, fclients.c).

Diese jeweils eigenständigen Prozesse sind in sich ebenfalls modular organisiert. Der ursprüngliche Client wurde auf Grund seiner Größe in mehrere Module zerlegt.

⁴⁹ Siehe Abschnitt 5.2.4.

⁵⁰ Siehe Abschnitt 5.1.

Sämtliche Server-Programme werden normalerweise vom Monitorprogramm gestartet und beendet und laufen auf einer Maschine. Um das Testen der einzelnen Module zu erleichtern, wurde jedoch auch ein manueller Start ermöglicht. Wird bei solch einem manuellen Start der Parameter „test“ angegeben, dann wird die Kommunikation zwischen den einzelnen Modulen am Bildschirm ausgegeben. Beispielsweise werden die Kommandos und deren Argumente, die der Client an den Dialogserver übergibt, ebenso angezeigt wie die Rückgabewerte vom Dialogserver zum Client.

Die Programme können einheitlich mit der Tastenkombination „STRG+C“ abgebrochen werden.

Alle wichtigen Aktionen werden für jedes Servermodul separat in einer Logbuchdatei festgehalten. Die Datei wird jeweils beim Start des Programms normiert.

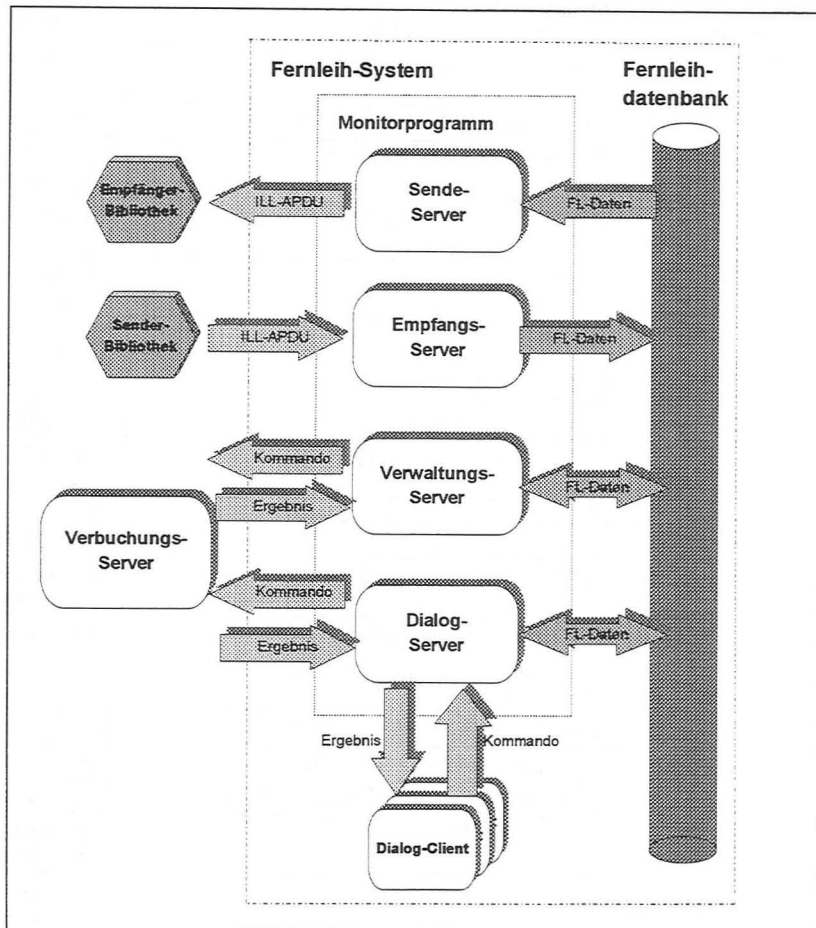
Aufgaben der
System-Module

Die Module haben im wesentlichen folgende Aufgaben:

- Das Monitorprogramm dient dem Starten, Überwachen und Beenden der einzelnen Komponenten des Fernleihsystems.
- Der Dialogserver kommuniziert mit den Clients des Fernleihsystems. Die Datenbankzugriffe für die Benutzer- und Dienstfunktionen erfolgen ebenfalls über diesen Prozeß.
- Die Überwachung der Zustände und Folgezustände⁵¹ der gebenden und nehmenden Fernleihe erfolgt durch den Verwaltungsserver. Dieses Programm fungiert außerdem als Zustands- und Datenbankverwaltungssystem.
- Die Generierung von Sendungen nach dem ILL-Protokoll übernimmt der Sendeserver. Bestellvorgänge, die einen bestimmten Bearbeitungszustand erreicht haben, werden kodiert und an die gewünschte Empfängerbibliothek gesendet.
- Den Empfang von ILL-Nachrichten nach dem ILL-Protokoll und die Dekodierung dieser Nachrichten regelt der Empfangsserver. Eintragungen der Daten der empfangenen ILL-APDUs in die Fernleihdatenbank werden ebenfalls durch dieses Modul vorgenommen.
- Der Dialog-Client übernimmt die Realisierung der Bedieneroberfläche unter UNIX oder DOS.

⁵¹ Die lokalen Zustände werden in Abschnitt 9 erläutert.

Abb. 6.1
Grobkonzept
Client-Server-
Architektur



Fernleihsystem-
steuerung

Zur Steuerung der Prozesse des Fernleihsystems dient das **Monitorprogramm „fesys“ (Fernleihsystemsteuerung)**. Dieses Programm ermöglicht ein geordnetes und übersichtliches Starten und Beenden der einzelnen Prozesse des Fernleihsystems. So muß z.B. der Dialogserver aktiviert sein, bevor die Dialog-Clients gestartet werden und Sendungen werden nur verschickt, wenn der Sende-server aktiv ist etc.

Konfigurationsdatei
für die Server-
Prozesse

Die **Konfigurationsdatei „cfg.fern“** enthält wichtige Angaben zur Steuerung des Systems. Sie wird im Aufrufverzeichnis sämtlicher Server erwartet. Beim Starten des Monitorprogrammes wird zunächst das Vorhandensein dieser Datei geprüft.

Die wichtigsten Konstanten, die in der Konfigurationsdatei enthalten sind, können der folgenden Abbildung entnommen werden. Zusätzlich enthält die Datei „cfg.fern“ die Namen aller temporären Druckdateien.

Abb. 6.2
Aufbau der Datei
cfg.fern

cfg.fern		
DB_PATH	fifh	Datenbank-Bezeichner
EMPF_MODUS	0	0=TCP-Empfang, 1=X.400-Empfang
MHS_ZUGANG	/export/home/fern/.postrc	post_path im Programm
FS_PATH	fserver	Pfad und Name des Dialogservers
SS_PATH	sserver	Pfad und Name des Sendeservers
ES_PATH	eserver	Pfad und Name des Empfangsservers
ZS_PATH	zserver	Pfad und Name des Verwaltungsservers
FLOGBPATH	logb.fserv	Pfad und Dateiname Dialogserver-Logbuch
SLOGBPATH	logb.sserv	Pfad und Dateiname Sendeserver-Logbuch
ELOGBPATH	logb.eserv	Pfad und Dateiname Empfangsserver-Logbuch
ZLOGBPATH	logb.zserv	Pfad und Dateiname Verwaltungsserver-Logbuch
URLADPROT	urlad.prot	Dateiname Urlade-Protokoll
HOMESIGEL	352	Nr. des Heimat-Sigels
HOMEIBNR	07/000011	Nr. der Heimatbibliothek
BNR_LESESAAL	47/000022	Benutzer-Nr. vom Lesesaal
ILLSUBJECT	ISOStandard10161	Subject einer ILL-Sendung
VBPORTNUM	2200	Portnummer des Verbuchungsservers
VBHOSTADR	koala6	Hostname des Verbuchungsservers
FEPORNUM	4711	Portnummer des Fernleih-Dialogservers
ZEIT_DIFF	0	Zeitdifferenz falls Abweichung der Localtime

In den folgenden Abschnitten sollen die einzelnen Server-Prozesse näher erläutert werden.

6.1.1 Sendeserver

Der Sendeserver hat die Aufgabe, anstehende sendefähige Aufträge, dem ILL-Standard entsprechend, an die jeweilige Partnerbibliothek zu verschicken.

Einleitende
Programm-Phase

In der einleitenden Programm-Phase wird zunächst die Interrupt-Routine für den Abbruch des Programms mit „STRG+C“ bekanntgegeben.

Nach der Prüfung, ob das Programm mit dem „test“-Parameter aufgerufen wurde, wird das Vorhandensein der Konfigurationsdatei „cfg.fern“ abgefragt. Anschließend wird die Logbuchdatei „logb.sserv“ für die Protokollierung geöffnet und der Zugang zur Fernleihdatenbank ermöglicht.

Nun können die Daten der Heimatbibliothek mit Hilfe des Home-Sigels⁵², welches in der Konfigurationsdatei eingetragen ist, aus der Partnertabelle⁵³ gelesen werden. Mit diesen Daten werden die jeweils benötigten Satzpuffer für die nehmende und gebende Bearbeitung normiert, d.h. mit den nötigen Standardwerten vorbelegt.

Zyklische Vorgangs-
prüfung

Das Programm prüft zyklisch in der Fernleihdatenbank, ob es sendefähige Aufträge gibt. Solche Aufträge können erstmalig sein (z.B. ILL-REQUEST) oder es sind Folgeaktionen, die im Laufe der Bearbeitung eines Auftrags entstehen (z.B. ILL-SHIPPED). Dabei wird zwischen nehmenden und gebenden Vorgängen unterschieden. Dies wird durch einen Schalter „send“ realisiert, der etwa einer technischen Flip-Flop-Schaltung entspricht. Ist dieser Schalter durch send = NE_ZU auf nehmende Zustände gesetzt, dann werden alle sendefähigen „Requester-Zustände“ in einer Cursorliste abgearbeitet. Verweist der Schalter durch send = GE_ZU auf gebende Zustände, so werden alle sendefähigen „Responder-Zustände“ in der zugehörigen Cursorliste abgearbeitet.

Sendefähige Aufträge der nehmenden Fernleihe haben in der Zustandstabelle der Fernleihdatenbank „nehm zu tab“⁵⁴ als Primärkennung den Wert ‘send_mail_nz’⁵⁵. Sendefähige Aufträge der gebenden Fernleihe werden in der Tabelle „geb zu tab“ durch die Primärkennung ‘send_mail_gz’ gekennzeichnet. Die Sekundärkennung kennzeichnet auf nehmender und auch auf gebender Seite die Art der Sendung, die verschickt werden soll (ILL-APDU), wenn die Primärkennung den Auftrag als „bereit zum Senden“ kennzeichnet.

Die Cursorlisten werden sortiert nach Datum und Zeit abgearbeitet.

Beendete bzw. abgearbeitete Zustände haben in den Zustandstabellen ein Endedatum und eine Endezeit. Im Falle der zu versendenden Aufträge ist noch kein Endedatum und keine Endezeit gesetzt, somit gelten sie als un bearbeitet.

Kodierung der
Sendedaten

Wenn ein Vorgang als sendefähig erkannt ist, dann werden alle relevanten Daten zu diesem Vorgang aus der Datenbank ausgelesen. Diese werden für die Kodierfunktion in speziellen Puffern⁵⁶ bereitgestellt. Anschließend wird in die Routine „encode_apdu(...)“ verzweigt, welche die Kodierung der lokalen Fernleihdaten gemäß dem ILL-Standard durchführt⁵⁷. Als Aufrufparameter wird die ILL-APDU-Nummer und eine eindeutige Transaction-ID übergeben.

⁵² Das Sigel ist eine feststehende eindeutige Abkürzung für die Bibliothek.

⁵³ Tabellen der Fernleihdatenbank werden in Abschnitt 8.1 erläutert.

⁵⁴ Tabellen der Fernleihdatenbank werden in Abschnitt 8.1 erläutert.

⁵⁵ Primär- und Sekundärkennung werden in Abschnitt 9 erklärt.

⁵⁶ Siehe Erläuterung in Abschnitt 8.3.

⁵⁷ Das Prinzip der Kodierung zu versendender ILL-Vorgänge wird in Abschnitt 7 erläutert.

Die Kodierfunktion verzweigt anhand dieser Parameter ihrerseits in das dazugehörige externe Unterprogramm („ei....c“) zur Abhandlung der Kodierung der speziellen ILL-APDU. Diese Funktion liest dann die Daten aus den Satzpuffern dem Standard des ILL-Dienstes entsprechend in die sendefähige Struktur ein.⁵⁸

Sendung

Nach erfolgreicher Verschlüsselung steht die sendefertige ILL-APDU in der temporären Datei 'illsend.tmp'. Durch den Aufruf der Prozedur „x400 send(...)“ wird diese Datei dem X.400 Mailsystem zum Versenden übergeben oder das Versenden erfolgt direkt über Hostname und Portnummer im Internet auf der Basis der TCP/IP-Sockets.

Die X.400-Empfängeradresse wird der Partnertabelle der Fernleihdatenbank entnommen. Alle potentiellen Empfängerbibliotheken müssen dort mit ihrer X.400-Adresse eingetragen sein. Bei Kommunikation über Hostname und Portnummer müssen lediglich die entsprechenden Felder 'Host' und 'Port' in der Partnertabelle belegt sein.

Zustandswechsel
oder Zustands-
generierung

Abgearbeitete bzw. versendete Zustände werden nun aktualisiert, indem entweder der alte Zustand beendet und ein neuer Zustand, der dem Dienst entspricht, generiert wird. Oder es wird einfach nur der alte Zustand abgeschlossen⁵⁹. Eine stark vereinfachte Darstellung des Programmablaufs kann der folgenden Abbildung entnommen werden.

Abb. 6.3
Pseudo-C-Code des
Sendeservers

```
sserver()
{
  ...
  Einleitende Programmphase;
  ...
  for(;;)
  {
    if (send = NE_ZU)
      alle sendefähigen nehmenden Vorgänge ermitteln;
    if (send = GE_ZU)
      alle sendefähigen gebenden Vorgänge ermitteln;

    while(Sind noch Vorgänge vorhanden?)
    {
      nächsten zu sendenden Vorgang ermitteln;
      alle benötigten Daten in Puffer lesen;
      Daten kodieren;
      kodierte Daten versenden;
      Zustände des versendeten Vorgangs aktualisieren;
    }
    if (send = NE_ZU)
      send = GE_ZU;
    else
      send = NE_ZU;
  }
}
```

⁵⁸ Siehe genauere Erläuterungen in Abschnitt 7.

⁵⁹ Siehe Abschnitt 9.

6.1.2 Empfangsserver

Der Empfangsserver erwartet die ankommenden Sendungen der Partnerbibliotheken und wandelt diese in die von der eigenen Bibliothek benötigten Daten um, damit sie von anderen Modulen weiter bearbeitet werden können. Das bedeutet beispielsweise, daß mit Hilfe der eintreffenden Sendungen lokale Zustände zu bestehenden Vorgängen abgeschlossen werden können oder neue Folgezustände generiert werden müssen, die eine weitere Bearbeitung der Fernleihvorgänge erst ermöglichen⁶⁰. Werden Folgezustände generiert, müssen Sätze in der Datenbank angelegt werden, die dem ILL-Dienst entsprechen, aber lokalen Anforderungen genügen⁶¹.

Einleitende
Programm-Phase

In der einleitenden Programm-Phase wird zunächst die Interrupt-Routine für den Abbruch des Programms mit „STRG+C“ bekanntgegeben.

Nach der Prüfung, ob das Programm mit dem „test“-Parameter aufgerufen wurde, wird die Existenz der Konfigurationsdatei abgefragt. Anschließend wird die Logbuchdatei „logb.eserv“ für die Protokollierung geöffnet.

Beim Empfang der ankommenden Daten wird zwischen TCP/IP-Modus und X.400-Modus unterschieden, wobei die weitere Bearbeitung der Daten gleich ist.

Abb. 6.4
Pseudo-C-Code des
Empfangsservers

```

eserver()
{
...
  Einleitende Programmphase;
  ...
  if( TCP-Empfang )
  {
    Socket einrichten;
    for(;;)
    {
      Verbindung aufnehmen;
      ...
      Sohnprozeß erzeugen;
      ...
      switch (Prüfung der Verbindung über Sohn)
      {
        case (Fehler):
          {Socket schließen;
           Programm beenden;}
        case (alles in Ordnung):
          {Bearbeitung der
           eintreffenden Daten;
           Socket schließen;
           Programm beenden;}
        default:
          ...
      }
      Socket schließen;
    }
  }
  if( X.400-Empfang )
  { Bearbeitung der eintreffenden Daten; }
}

```

⁶⁰ Siehe Abschnitt 9.

⁶¹ Siehe Abschnitt 9.

TCP-Empfang	Der obigen Abbildung ist zu entnehmen, daß für den TCP-Empfang zunächst ein Socket eingerichtet und anschließend die Verbindung aufgebaut wird. Es wird ein Sohnprozeß erzeugt, mit dem die Bearbeitung der eintreffenden Daten erfolgt. So übernimmt jeweils ein Sohn die weitere Verarbeitung der Daten. Die eintreffenden Zeichen werden zunächst in einen Empfangspuffer geleitet. Ist die Übertragung beendet oder der Empfangspuffer gefüllt, wird der gesamte Pufferinhalt in die Datei „illrecv.tmp“ geschrieben und der Puffer geleert. Diese Datei muß prozeßspezifisch angelegt werden. Der Socket wird nach Abschluß der Bearbeitung geschlossen.
X.400-Empfang	Der zweite Programmzweig der Abbildung des Pseudo-Codes des Empfangsservers entspricht dem X.400-Empfang. In diesem Fall wird mit Hilfe der Prozedur „check_x400“ das Eintreffen von Daten abgefragt. Hierbei wird mit Hilfe der X.400-Shell-Tools geprüft, ob eine Sendung mit dem ILL-Subject 'ISOStandard10161' empfangen wurde. Die empfangene Mail-Datei wird ebenfalls in die Datei „illrecv.tmp“ konvertiert. Für die Bearbeitung der Daten wird der Zugang zur Datenbank ermöglicht und die Daten der eigenen Bibliothek zur Kontrolle abgefragt.
Dekodierung der Empfangsdaten	Durch den Aufruf der Dekodieringsroutine „decode_apdu()“ wird der Inhalt der Datei „illrecv.tmp“ mit Hilfe der einzelnen Dekodierungsfunktion („di....c“) für den jeweiligen ILL-Dienst entschlüsselt. ⁶² Dabei ist es beim Dekodieren unwichtig, welcher Empfangsmodus vorher gesetzt war, da sowohl mit TCP als auch mit X.400 empfangene Daten in die Datei „illrecv.tmp“ geleitet werden.
Zustandswechsel oder Zustands-generierung	Die für die Fernleihe relevanten Daten werden in speziellen Pufferspeichern ⁶³ bereitgestellt und anschließend vom Empfangsserver in die Fernleihdatenbank geschrieben. D.h. in den Zustandstabellen der gebenden und nehmenden Bearbeitung werden, dem jeweiligen ILL-Dienst entsprechend die Zustände aktualisiert. Beispielsweise werden die auf „Empfang“ wartenden Zustände abgeschlossen und die Folgezustände aktiviert oder neue parallele Zustände generiert. ⁶⁴ Die Abbildung auf der nächsten Seite zeigt den Ablauf der Bearbeitung von empfangenen Daten.

⁶² Siehe genauere Erläuterungen in Abschnitt 7.

⁶³ Siehe Erläuterung in Abschnitt 8.3.

⁶⁴ Siehe Abschnitt 9.

Abb. 6.5
Die Bearbeitung
der eintreffenden ILL-
Sendung

```

bearbeitung()
{
...
Datenbank öffnen;
die eintreffenden Daten je nach Empfangsmodus in Datei
„illrecv.tmp“ leiten;
...
Dekodierung
...
switch( eingetroffene ILL-APDU )
{
case (send_request):
    neuen Bestellsatz in DB aufnehmen;
    break;
case (send_received):
    Status „RECEIVED“ in DB setzen;
    break;
case ...
}
Datenbank schließen;
...
}

```

6.1.3 Verwaltungsserver

Der Verwaltungsserver soll die Überwachung von Transaktionszuständen der einzelnen Fernleihdienste ermöglichen und dient als Schnittstelle zum Verbuchungssystem.

Einleitende
Programm-Phase

In der einleitenden Programm-Phase wird zunächst die Interrupt-Routine für den Abbruch des Programms mit „STRG+C“ bekanntgegeben.

Nach der Prüfung, ob das Programm mit dem „test“-Parameter aufgerufen wurde, wird das Vorhandensein der Konfigurationsdatei abgefragt. Anschließend wird die Logbuchdatei „logb.zserv“ für die Protokollierung geöffnet und der Zugang zur Fernleihdatenbank realisiert.

Negative Rückmel-
dungen bearbeiten

Der Verwaltungsserver durchsucht die Fernleihdatenbank zyklisch nach Rückmeldungen von Partnerbibliotheken, die anzeigen, daß die Bestellung für eine Monographie oder eine Zeitschrift nicht erfüllt werden kann.

Wurde eine negative Rückmeldung empfangen, so aktiviert der Verwaltungsserver die aktuelle Bestellung mit Hilfe des nächsten in der Datenbank eingetragenen Leitweges neu. Der abgearbeitete Leitweg wird in der Zustandstabelle mit einem negativen Status versehen. Anschließend wird ein neuer sendefähiger Vorgang mit dem nächsten zu verwendenden Leitweg generiert. Das bedeutet, die Bestellung wird nun an eine andere Partnerbibliothek verschickt. Ist kein weiterer Leitweg vorhanden, wird eine Benutzerbenachrichtigung veranlaßt.

Benachrichtigung
des Benutzers
veranlassen

Zyklisch wird die Fernleihdatenbank nach Sätzen durchsucht, die einen Zustand erreicht haben, in dem der Benutzer eine Benachrichtigung erhalten soll. Der Zustand zur Benachrichtigung des Benutzers wird abgeschlossen.

Schnittstelle zum
Verbuchungssystem

Mit Hilfe der externen Funktion „fdgbri“, die eine Schnittstelle zum lokalen Verbuchungssystem realisiert, wird entweder direkt eine Nachricht an das Verbuchungssystem gesendet oder eine Nachricht in einer temporären Druckdatei hinterlegt. Die Kommunikationsroutinen zum Verbuchungssystem sind alle in der Datei „ferflup.c“ implementiert.⁶⁵

Um für wartende Zustände die Dauer des Wartens einzugrenzen, soll eine Prüfung dieses Zeitlimits durch den Verwaltungsserver möglich werden. Diese Funktion ist jedoch noch nicht realisiert worden.

Es gehört ebenfalls zu den zukünftigen Aufgaben des Verwaltungsservers, aus abgearbeiteten Fernleihvorgängen eine Statistik zu erstellen. Beendete Vorgänge sollen gelöscht und die ermittelten Statistikdaten in einer separaten Datenbank erfaßt werden.

Der Abbildung kann man entnehmen, daß bis jetzt nur die Suche nach negativen Bestellrückmeldungen und die Suche nach anstehenden Benutzerbenachrichtigungen im Programmablauf berücksichtigt wurden.

Abb. 6.6
Pseudo-C-Code des
Verwaltungsservers

```
zserver()
{
  ...
  Einleitende Programmphase;
  ...
  for(;;)
  {
    alle negativen Bestellrückmeldungen suchen;
    while( negative Rückmeldung vorhanden )
    {
      alle Leitwege zum Fernleihvorgang ermitteln;
      while( Ist noch ein Leitweg vorhanden? )
      {
        if( Ist der Leitweg noch unbearbeitet? )
        {
          Bestellung über diesen Leitweg anstoßen;
        }
        else if( Leitweg schon abgearbeitet )
        {
          Bestellung negativ beenden;
          Benutzerbenachrichtigung veranlassen;
        }
      }
    }
    Vorgänge, die Benutzernachricht erwarten, suchen;
    while( Vorgang mit Benachrichtigungszustand vorhanden )
    {
      Benachrichtigungszustand beenden;
      Benachrichtigung durch Verbuchung veranlassen;
    }
  }
  ...
}
```

⁶⁵ Die Schnittstelle zur Ausleihverbuchung wird in Abschnitt 10.3 beschrieben.

6.1.4 Dialogserver

Der als „fserver“ bezeichnete Prozess hat die Aufgabe, die Dialogwünsche der Clients zu realisieren. Die Kommunikation erfolgt nach dem Client-Server-Prinzip mit Hilfe der Socket-Schnittstelle via TCP/IP. Die Kommunikation erfolgt im Kommandomodus. So können auch „einfache“ Clients (z.B. TELNET) Dialogkommandos absetzen und die Ergebnisse anzeigen. Mit dem zugehörigen Client des Fernleihsystems erfolgt eine interne Umsetzung der Kommandos, so daß die Bedieneroberfläche maskenorientiert dargestellt werden kann.

Der Dialogserver wird normalerweise vom Monitorprogramm gestartet und beendet. Ein manueller Start ist durch den Aufruf von „fserver“ oder „fserver test“ möglich. Wird der Parameter „test“ angegeben, dann erfolgt eine Anzeige der Kommunikation mit dem Client am Bildschirm. Dies bedeutet, daß sämtliche Kommandos mit ihren Argumenten, die vom Client an den Server übergeben werden, ebenso angezeigt werden, wie die Rückgabewerte des Servers an den Client.

Einleitende
Programm-Phase

In der einleitenden Programm-Phase wird zunächst die Interrupt-Routine für den Abbruch des Programms mit „STRG+C“ bekanntgegeben.

Nach der Prüfung, ob das Programm mit dem „test“-Parameter aufgerufen wurde, wird das Vorhandensein der Konfigurationsdatei „cfg.fern“ abgefragt. Die Datei enthält unter anderem auch die Portnummer, unter der sich die Clients beim Dialogserver anmelden. Anschließend wird die Logbuchdatei „logb.fserv“ für die Protokollierung geöffnet. Bei der Protokollierung wird ebenso verfahren wie beim Programmablauf des Servers mit dem „test“-Parameter, nur daß die Ausgabe der Kommandos und der Rückgabewerte nicht über den Bildschirm erfolgt, sondern in die Logbuchdatei geschrieben wird.

Verbindungsaufbau

Es wird ein Socket für einen eigenen Port eingerichtet. Für jede anliegende Anfrage auf Kontaktaufnahme wird die Verbindung über diesen Socket hergestellt. Für jede Verbindung wird mit dem Systemcall „fork()“ ein Sohnprozeß generiert, der die ankommenden Anfragen eines Clients weiterbearbeitet. Die Anzahl der Clients ist nicht begrenzt.

Für die Bearbeitung wird die Fernleihdatenbank geöffnet und die Verbindung zum Verbuchungsserver aufgenommen, wenn die Portnummer des Hostes in der Konfigurationsdatei eingetragen wurde. Ansonsten erfolgt nur die Eintragung ins Logbuch des Servers, daß die Verbindung zum Verbuchungssystem nicht verwirklicht werden konnte.

Nach der Verbindungsaufnahme zwischen Dialog-Client und Server wird zuerst ein Begrüßungsbild vom Server zum Client geschickt. Der Text für dieses Bild wird der Datei „flgruss.pic“ entnommen.

Der Client sendet seine Anforderungen als Kommandos über den Socket zum Server. Der Ablauf des Serverprogramms für den Dialog mit dem Client ist in der folgenden Abbildung noch einmal übersichtlich in Pseudo-C-Code dargestellt.

Abb. 6.7
Pseudo-C-Code des
Dialogservers

```
fserver()
{...
  Einleitende Programmphase;
  ...
  Socket einrichten;
  for(;;)
  {
    Verbindung aufnehmen;
    ...
    Sohnprozeß erzeugen;
    ...
    switch (Prüfung der Verbindung über Sohn)
    {
      case (Fehler):
      {
        Socket schließen;
        Programm beenden;
      }
      case (alles in Ordnung):
      {
        bearbeitung();
        Socket schließen;
        Programm beenden;
      }
      default: ...
    }
  }
  Socket schließen;
}
```

Einlesen der empfangenen Zeichen in einen Puffer

Vom aktuellen Socket werden die ankommenden Zeichen durch den Server nacheinander in einen Empfangspuffer gelesen, wobei Steuerzeichen übergangen werden. Das Server-Programm prüft nun zunächst, ob das erste Zeichen im Empfangspuffer signalisiert, daß die Verbindung zum Client abgebaut werden soll. Der Client sendet in diesem Fall eine Escape-Sequenz oder ein „q/Q“. Bleibt die Verbindung bestehen, werden die ersten vier Zeichen untersucht.

Verzweigung in eine Kommandoroutine

Diese entsprechen einem Schlüsselwort. Jedes Schlüsselwort stellt ein Kommando mit sogenannten Aspektzuweisungen⁶⁶ dar, mit dem bestimmte Datenbankzugriffe durchgeführt werden können. Mit den Aspektzuweisungen werden Werte übergeben, die für die Ausführung des Kommandos benötigt werden (wie z.B. die Fernleih-Bestellnummer für den Zugriff auf einen Bestellsatz in der Fernleihdatenbank). Der Server vergleicht also die eingetroffenen Zeichen mit den Kommandos. Entspricht eine Zeichenkette, die vom Client übermittelt wurde, einer solchen Basis-Kommunikationsfunktion, so verzweigt der Dialogserver in die zugehörige externe Kommandoroutine. Als Parameter wird immer der gesamte Inhalt des Empfangspuffers, die Adresse des Sendepuffers, die Länge der Empfangsdaten und die maximale Länge der Sendedaten an die Kommandoroutine weitergereicht.

⁶⁶ Siehe Erläuterung in Abschnitt 6.1.5.1.

Die Routine durchsucht den übergebenen Empfangspuffer nach vorhandenen Argumenten für das auszuführende Kommando. Mit den übergebenen Argumentwerten führt sie das Kommando aus und nach erfolgreicher Bearbeitung schreibt sie das Ergebnis in den Sendepuffer.

Als Rückgabe liefert die Routine einen ganzzahligen numerischen Wert an den Server. Ist dieser Wert „Null“, also das Kommando erfolgreich ausgeführt worden, dann überträgt der Server den Inhalt des Sendepuffers, so wie er von der Kommandoroutine gefüllt wurde, über den Socket zum Client.

Ist der Wert ungleich „Null“, so bezeichnet er einen bestimmten Fehler. In diesem Fall wird mit Hilfe einer Sprunganweisung ein normierter, dem Rückgabewert des Kommandos entsprechender Fehlertext vom Server-Programm in den Sendepuffer geschrieben. Erst danach überträgt der Dialogserver den Inhalt des Sendepuffers an den Client.

Verbindungsabbau

Nach Bearbeitungsende wird die Datenbank und der Socket zum Client vom Dialogserver geschlossen.

Der Ablauf der Bearbeitung von Kommandos, die durch den Dialogserver erfolgt, wird im folgenden Pseudo-C-Code noch einmal vereinfacht beschrieben.

Abb. 6.8
Bearbeitung der
Kommandos, die
vom Socket des
Client ankommen

```

bearbeitung()
{
    ...
    Fernleihdatenbank öffnen;
    Verbindung zum Verbuchungsserver öffnen;
    Begrüßungsbild übertragen;
    for(;;)
    {
        vom Socket ankommende Zeichen ohne Steuerzeichen in
        den Empfangspuffer lesen;

        for(;;)
        {
            if((erstes Zeichen im Empfangspuffer == 'g') ||
                (erstes Zeichen im Empfangspuffer == 'Q'))
                {Fülle Sendepuffer mit („Bye, Bye!“); break;}

            if(erste 4 Zeichen im Empfangspuffer == „keln“)
                {rc = fekeln(...); break;}
            if(erste 4 Zeichen im Empfangspuffer == „ebes“)
                {rc = feebes(...); break;}
            ...
        }

        switch (rc)
        {
            case 0: Fülle Sendepuffer mit Fehlertext 1; break;
            case 1: Fülle Sendepuffer mit Fehlertext 2; break;
            ...
        }

        send(Sendepuffer);
    }
    Datenbank schließen;
    Verbuchungsverbindung schließen;
    return(0);
}

```

6.1.5 Die Dialogkommandos

6.1.5.1 Die Namenskonventionen der Dialogkommandos

Kommando-Syntax

Die Kommandos, mit denen die Kommunikation zwischen Dialogserver und Client oder Dialogserver und Verbuchungsserver⁶⁷ realisiert wurde, sind nach einer einheitlichen Syntax aufgebaut. Jedes Kommando beginnt mit einem 4-stelligen Schlüsselwort. Es folgt danach eine variable Anzahl von Aspektzuweisungen. Eine Aspektzuweisung besteht aus einem 2-stelligen Aspektkey, einem Gleichheitszeichen und dem Aspektwert.

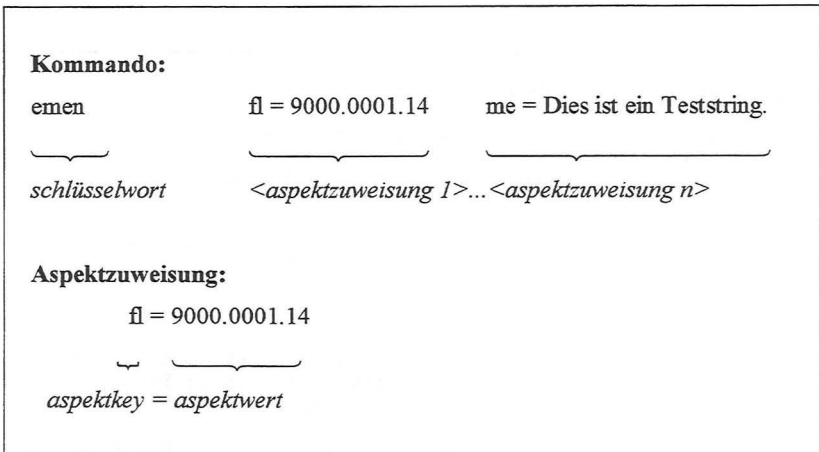
kommando ::= schlüsselwort <aspektzuweisung 1> ... <aspektzuweisung n>

aspektzuweisung ::= aspektkey = aspektwert

Beispiel: emen fl = 9000.0001.14 me = Dies ist ein Teststring.

Erfasse die Meldung „Dies ist ein Teststring.“, die zum Vorgang der nehmenden Bearbeitung mit der Fernleih-Bestellnummer 9000.0001.14 gehört, in der Fernleihdatenbank.

Abb. 6.9
Syntax eines
Dialogkommandos



Die jeweilige Kommandoroutine wurde in der Datei „fexxxx.c“ in unserem Beispiel also in der Datei „feemen.c“ implementiert.

Es gibt weitere Namenskonventionen, die nicht wie die Anzahl der Zeichen eines Kommandos einen programmtechnischen Hintergrund haben. Ein Bestandteil des Namens kennzeichnet die Zuordnung zum jeweiligen Dienst:

Kennzeichnung
des Dienstes

<u>Kommando</u>	<u>ILL-DIENST</u>
emen..., amel...	MESSAGE
erec..., arec...	RECALL
eove..., zove...	OVERDUE

⁶⁷ Die Schnittstelle zum Verbuchungssystem wird in Abschnitt 10.3 erläutert.

Zugriff auf Daten der nehmenden oder gebenden Fernleihe

Eine Gruppierung erfolgt auch nach dem Zugriff auf Daten der gebenden oder nehmenden Fernleihe. Bei einigen Service-Dienst-Kommandos ist dies an der Endung 'n' für nehmend und 'g' für gebend zu erkennen. Eine Meldung der gebenden Fernleihe kann der Client also mit dem Kommando „zmeg...“ vom Dialogserver anfordern, während für die nehmende Fernleihe „zmen...“ aufgerufen werden muß.

Art der Datenbearbeitung

Weiter wird nach Art der Zugriffe auf die Datenbasis unterschieden, je nachdem, ob ein Vorgang gedruckt (aktiviert/ausgeführt), erfaßt, korrigiert oder gezeigt werden soll. Beispiele können der folgenden Abbildung entnommen werden.

Abb. 6.10
Beispiel der Kommandos für die Bearbeitung eines Dienstes

Kommando	Quelldatei	Bedeutung
amel...	feamel.c	aktivieren/ausführen
emen...	feemen.c	erfassen (nehmende FL)
kmeg...	fekmeg.c	korrigieren/updaten (gebende FL)
zmen...	fezmen.c	zeigen einzelnen Vorgang (nehmende FL)
zumg...	fezumg.c	zeigen ungelesener Vorgänge (gebende FL)

Beim Aufruf eines Kommandos gibt es zwingend vorgegebene und optionale Aspektzuweisungen.

Bei der Ausführung einer Kommandoroutine erfolgt zuerst die Prüfung, ob zwingend vorgegebene Aspektzuweisungen beim Aufruf des Kommandos vorhanden sind. Die Kommandoroutine durchsucht dabei den vom Server-Programm übergebenen Empfangspuffer nach dem entsprechend vorgegebenen Aspektkey. Wird dieser gefunden, kann sie den zugehörigen Aspektwert einlesen.

Die Kommandoroutine gibt einen ganzzahligen numerischen Wert an den Dialogserver zurück. Ist dieser Wert „ungleich Null“, kennzeichnet er eine Textzeile, die einer Fehlermeldung entspricht. Der Server kann also sofort diese Fehlermeldung an den Client zurückliefern oder die ordnungsgemäße Erfassung bestätigen.

Eine Auflistung aller Dialogkommandos ist im Anhang II zu finden!

6.1.5.2

Dialogkommandos zum Erfassen eines Vorgangs

Die Kommandoroutinen, die der Client an den Dialogserver zum Erfassen eines neuen Fernleihvorganges übermittelt, sind alle ähnlich aufgebaut. Dabei kennzeichnet der Name der Kommandoroutine, die ausgeführt werden soll, den jeweiligen Dienst, für den ein Vorgang erfaßt werden soll.

Unter Vorgang verstehen wir im folgenden einen oder mehrere Datensätze, die in den Tabellen der Fernleihdatenbank (z.B.: „geb_zu_tab“ oder „nehm_zu_tab“) angelegt werden. In diesen Sätzen sind dann sämtliche Informationen enthalten, die den aktuellen Zustand des jeweiligen Dienstes kennzeichnen.⁶⁸

⁶⁸ Zur Erläuterung des Tabellenaufbaus siehe auch Abschnitt 8.1.

Dazu gehören neben der Fernleih-Bestellnummer vor allem die Primär- und Sekundärkennung.⁶⁹ Erst diese Informationen ermöglichen die Bearbeitung der Daten, die für die Ausführung der Fernleihe benötigt werden.

Ein zwingend benötigter Aspektwert bei der Erfassung ist die Fernleih-Bestellnummer sowohl bei der gebenden Bearbeitung als auch bei der Erfassung eines nehmenden Vorgangs.

Eine Prüfung erfolgt auf die Ausführbarkeit der Erfassung. D.h. ein neuer Vorgang darf nur dann angelegt werden, wenn die globalen und lokalen Zustände der ILL-Transaktion⁷⁰ dies für den zu bearbeitenden Dienst auch gestatten. Voraussetzung ist natürlich, daß ein erster Datensatz mit einem Anfangsstatus und der vorliegenden Fernleih-Bestellnummer existiert.

Beispiel:

Eine Meldung auf Ablauf der Leihfrist (OVERDUE) darf nur dann an die Partnerbibliothek geschickt (also erfaßt) werden, wenn:

- die Bibliothek das Objekt von uns erhalten hat,
- die Leihfrist tatsächlich abgelaufen ist,
- die Leihfrist nicht zwischenzeitlich durch eine Anfrage auf Leihfrist-Verlängerung auf ein späteres Datum verschoben wurde,
- das Objekt nicht bereits zur Rückbuchung bei uns eingetroffen ist,
- das Objekt nicht bereits bei uns zurückgebucht wurde.

Alle diese Punkte müssen vor der Erfassung der Mahnung geprüft werden.

Sind alle für den zu bearbeitenden Dienst notwendigen Überprüfungen positiv ausgefallen, dann werden neue Datensätze in den Tabellen der nehmenden oder gebenden Bearbeitung angelegt.

6.1.5.3

Dialogkommandos zum Ausführen/Aktivieren eines Vorgangs

Diese Dialogkommandos stoßen vor allem bei den speziellen Servicediensten den Ausdruck von vordefinierten Listen an. Da dies die überwiegende Zahl der Kommandos ist, soll hier speziell auf diesen Listendruck eingegangen werden.

Der Client kann dabei mit der Übergabe der Aspektzuweisungen steuern, welche Informationen gedruckt werden sollen.

Druck eines einzelnen Vorgangs

Wird eine Fernleih-Bestellnummer übergeben, so wird nur der gewünschte Bestellvorgang gedruckt.

Beispiel: Drucke alle ungelesenen Meldungen zum Fernleihvorgang mit der Nr.: 9000.0047.14, der zur nehmenden Bearbeitung gehört.

Kommando: „amel fl=9000.0047.14 ty=n“

⁶⁹ Die Erläuterung der lokalen Zustände erfolgt in Abschnitt 9.

⁷⁰ Informationen zu den globalen ILL-Zuständen finden Sie in Abschnitt 3 und zu den lokalen Zuständen in Abschnitt 9.

Druck aller unbearbeiteten Vorgänge

Wird eine Liste zum ersten Mal gedruckt, so werden nur die unbearbeiteten Vorgänge auf dieser Liste erscheinen. Bei diesem Aufruf wird der Aspektkey „da“, der das Datum für einen Wiederholungsdruck kennzeichnet, nicht gesetzt. Man erkennt unbearbeitete Vorgänge an dem nicht eingetragenen Enddatum und der noch nicht gesetzten Endezeit. Mit dem Ausdruck gelten diese Vorgänge als abgeschlossen, d.h. sie erhalten eine Endezeit und ein Enddatum. Diese Vorgehensweise ermöglicht eine zügige Bearbeitung aller anstehenden Vorgänge durch die Fernleihabteilung. Am Morgen kann durch einen Batch-Aufruf der Druck sämtlicher Listen angestoßen werden, die dann z.B. beim Bibliographieren manuell weiterbearbeitet werden müssen.

Beispiel: Drucke alle empfangenen noch nicht zur Kenntnis genommenen Meldungen über Leihfristablauf aus.

Kommando: „aove“

Wiederholungsdruck

Sollten Listen verlorengehen oder die Bearbeitung von Vorgängen eines bestimmten Tages überprüft werden müssen, so ermöglicht dies der Wiederholungsdruck. Dieser wird durch die Übergabe des jeweiligen Datums mit einer Aspektzuweisung ermöglicht. So können alle Vorgänge, die zum jeweiligen ILL-Dienst an einem bestimmten Tag gedruckt worden sind, anhand des gesetzten Enddatums ermittelt und erneut ausgedruckt werden.

Beispiel: Drucke alle Meldungen über Leihfristablauf erneut aus, die bereits am 12.12.1995 bearbeitet wurden.

Kommando: „aove da=12.12.1995“

Alle Namen für die temporären Druckdateien sind in der Konfigurationsdatei „cfg.fern“ enthalten.

6.1.5.4

Dialogkommandos zum Zeigen eines Vorgangs

Bei diesen Kommandos unterscheidet man zwischen dem Anzeigen eines einzelnen Vorgangs und dem Auflisten aller unbearbeiteten Vorgänge des jeweiligen Dienstes.

Anzeige einer Vorgangsliste

Die Vorgänge werden im Listenformat am Bildschirm dargestellt. Eine Zeile einer solchen Liste enthält vordefinierte Werte wie z.B.: Fernleih-Bestellnummer, Anfangs- und Endezeit bzw. Datum.

Der Aufbau einer Zeile erfolgt bei der nehmenden Bearbeitung über den Aufruf der Funktion „fenlis“ und auf gebender Seite über die Funktion „feglis“. Diese Funktionen werden in den Kommandoroutinen aufgerufen. Sie generieren jeweils eine Zeile für die Darstellung am Bildschirm und füllen damit den Sendepuffer. Da in der Bildschirmdarstellung geblättert werden kann, übernimmt diese Funktion auch die Auswahl der entsprechenden Sätze für die jeweiligen Zeilen.

Die unbearbeiteten Vorgänge werden dabei wieder anhand der nicht gesetzten Endezeit aus der Datenbank ausgelesen.

Anzeige eines Vorgangs

Das Anzeigen eines einzelnen Vorgangs wird immer dann notwendig, wenn spezielle Informationen, die nicht in Listenform dargestellt werden können, zugänglich gemacht werden sollen. Dies ist z.B. der Fall, wenn die Bibliothek zu einer ILL-Transaktion eine Meldung oder eine Beschädigungsanzeige empfangen hat.

Wählt der Benutzer am Bildschirm mit RETURN eine Zeile aus der dargestellten Liste aus, so ruft der Client ein Kommando auf, das die jeweilige Hintergrundinformation zum entsprechenden Datensatz liefert.

Eindeutiger Key eines solchen in einer Bildschirmzeile dargestellten Satzes ist die Identnummer. Diese Identnummer wurde zuvor mit den anderen Informationen im Sendepuffer durch die Funktionen „fenlis“ oder „feglis“ übermittelt. Der Client übergibt diese Identnummer als Aspektwert beim Aufruf des Dialogkommandos zur Anzeige des einzelnen Vorgangs.

6.1.5.5

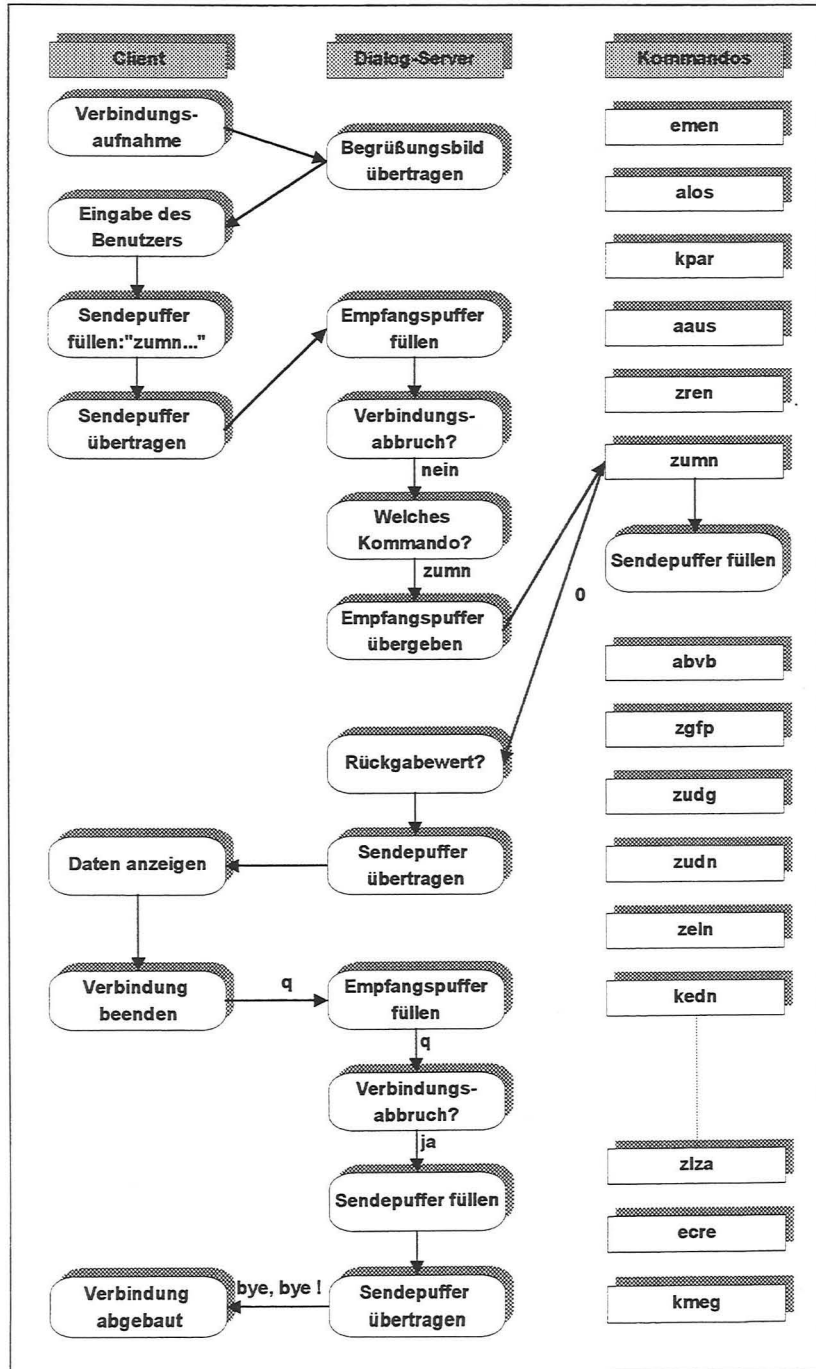
Dialogkommandos zum Korrigieren/Updaten eines Vorgangs

Diese Kommandos ermöglichen die Korrektur eines Fernleihdatensatzes. In der Regel erfolgt dieser Update vor allem dann, wenn ein Satz vom unbearbeiteten in den bearbeiteten Status wechselt. Das bedeutet, der Client ruft dieses Kommando automatisch auf, ohne daß eine spezielle Eingabe des Benutzers erwartet wird.

Wurde beispielsweise ein einzelner Vorgang am Bildschirm angezeigt, dann gilt er nicht mehr als unbearbeitet. In diesem Fall wird er mit einem eindeutigen Key, dies ist in der Regel nur die Identnummer, gelesen. Er erhält dann eine Endezeit und ein Endedatum, die kennzeichnen, wann der Vorgang abgeschlossen wurde.

Die folgende Abbildung stellt eine Übersicht über den Ablauf einer Kommunikation zwischen Server und Client und den Aufruf eines Kommandos dar.

Abb. 6.11
Die Ausführung des
Kommandos „zumn“
stark vereinfacht



6.2 Der Fernleih-Client

6.2.1 Die ADIX-Dialoglaufzeit

Der Begriff ADIX steht für Allgemeines Dialog-Interface für UNIX und DOS. Mit ADIX, auch ADIX-Dialoglaufzeit genannt, kann ein Dialog in der Sprache C programmiert werden. Die ADIX-Laufzeit führt selbständig alle wesentlichen Funktionen zum Aufbau eines Bildschirmdialoges aus und ermöglicht eine einheitliche Benutzeroberfläche.

6.2.1.1 ADIX-Prozeduren

Die ADIX-Prozeduren basieren auf CURSES-Bildschirmroutinen von UNIX bzw. auf Grafikroutinen unter MS DOS (curses.h/graph.h). Aufbauend auf diesen Basisroutinen enthält die ADIX-Laufzeit die Abhandlung der numerischen und alphanumerischen Feldeingabe, die Cursorsteuerung in Eingabefeldern, sowie die Feldeauswahl in Abhängigkeit von der betätigten Auslösetaste. Zusätzlich können nach einer Feldeingabe entsprechende Prüfprozeduren aufgerufen werden, die spezifische Abprüfungen realisieren.

Eingabefelder

Bei alphanumerischen **Eingabefeldern** kann neben der Standard-Feldeingabe auch eine Feldeingabe durch Menüauswahl erfolgen. Dabei wird von der ADIX-Dialoglaufzeit eine Auswahl an möglichen Eingabewerten angezeigt. Durch die Cursortasten kann ein Menübalken bewegt werden und somit die Selektion eines Wertes erfolgen.

Funktionsarten

Der Dialogablauf wird über ADIX-Funktionen gesteuert. Diese werden in vier **Funktionsarten** gegliedert:

- **Funktionen zur Initialisierung bzw. Verwaltung des Dialoges:**

Am Anfang eines Dialoges muß der Aufruf einer Funktion zur Initialisierung erfolgen. Sie konfiguriert die Darstellung für die jeweilige Umgebung automatisch - Farbe oder Monochrom. Ebenso muß am Ende des Dialoges die Dialoglaufzeit wieder geschlossen und beendet werden.

- **Funktionen, die mit der externen Bildbeschreibung aus einer BILD-Datei⁷¹ arbeiten:**

Mit diesen Funktionen können BILD-Dateien geöffnet und geschlossen werden, Rahmen um Bilder gezeichnet werden, die Felder und Masken eines Bildes ein- oder ausgeblendet werden, die Eingabefelder eines Bildes zur Eingabe freigegeben werden, der Cursor auf ein bestimmtes Feld positioniert werden usw.

- **Funktionen, die keine externe Bildbeschreibung benötigen:**

Dazu gehören Funktionen um den Cursor zu positionieren, um einen String auf der aktuellen Position auszugeben, um den Bildschirm zu löschen usw.

Im Anhang III sind sämtliche ADIX-Funktionen aufgelistet.

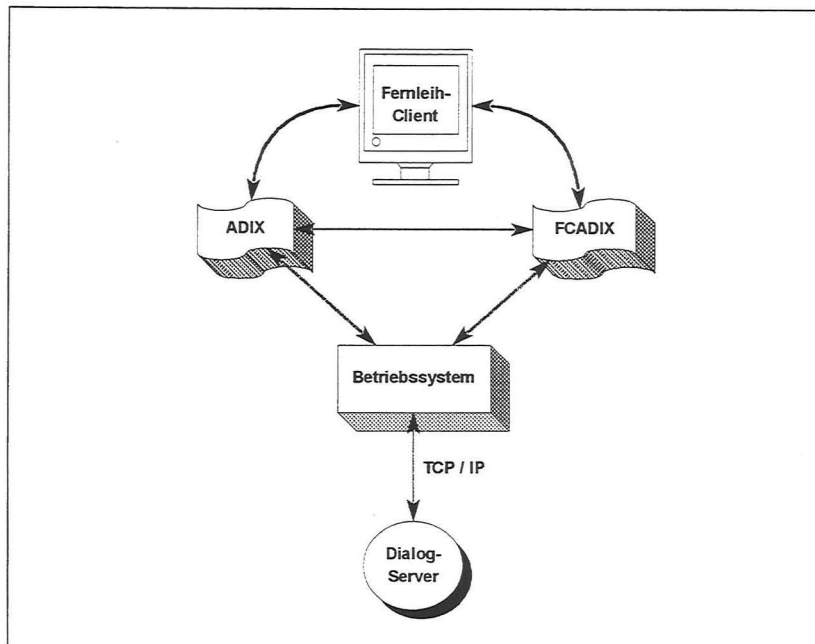
⁷¹ Siehe Abschnitt 6.2.1.3.

6.2.1.2 Client-Server-Dialogerweiterung für ADIX

FCADIX

Um auch den Dialog bei der Client-Server-Programmierung auch unterstützen zu können, wurde ADIX um das Modul **FCADIX** erweitert. Diese Unterstützung kann allerdings nur dann erfolgen, wenn die Kommunikation zwischen Client und Server dem Verbuchungsstandard⁷² (VB-Protokoll) der Universitätsbibliothek Konstanz entspricht. Das FCADIX-Modul enthält Funktionen, die im Client das Senden und Empfangen von Kommandos auf der Basis von TCP/IP ermöglichen.

Abb. 6.12
ADIX und
FCADIX
Dialoglaufzeit



6.2.1.3

Aufbau der BILD-Dateien

BILD-Dateien

Die ADIX-Dialoglaufzeit arbeitet mit sogenannten Maskendateien, auch **BILD-Dateien** genannt. Mit diesen Dateien werden die Bilder eines Dialoges beschrieben. Sie enthalten Angaben über Größe und Position des Bildes auf dem Bildschirm. Alle Komponenten des Bildes z.B. Menüs, Masken, Anzeigefelder oder Eingabefelder werden in der BILD-Datei beschrieben und können unabhängig vom Programm geändert werden.

Bild, Maske, Feld
und Menü

Ein **Bild** beinhaltet die Beschreibung aller Masken, Felder und Menüs. Eine **Maske** ist nur ein Teil des gesamten Bildes, z.B. eine Zeile. Besteht ein Bild nur aus einer Maske, die über mehrere Zeilen geht, so sind die am Bildschirm angezeigten Zeilen fix. Wird dagegen jede Zeile durch eine eigene Maske beschrieben, so können individuell Zeilen ein- oder ausgeblendet werden. Somit ist die

⁷² Siehe Abschnitt 6.1.5.1: Die Namenskonventionen der Dialogkommandos.

Anzeige variabel und ein Bild kann für mehrere Dialoganzeigen, die ähnlich sind, verwendet werden. Felder dienen zur Anzeige oder Eingabe von Werten. Ein Menü beschreibt die Masken, die durch einen Menübalken ausgewählt werden können.

Zur Verdeutlichung der Begriffe Bild, Maske, Feld und Menü dient die folgende Grafik:

Abb. 6.13
Die Begriffe Bild,
Maske, Feld und
Menü

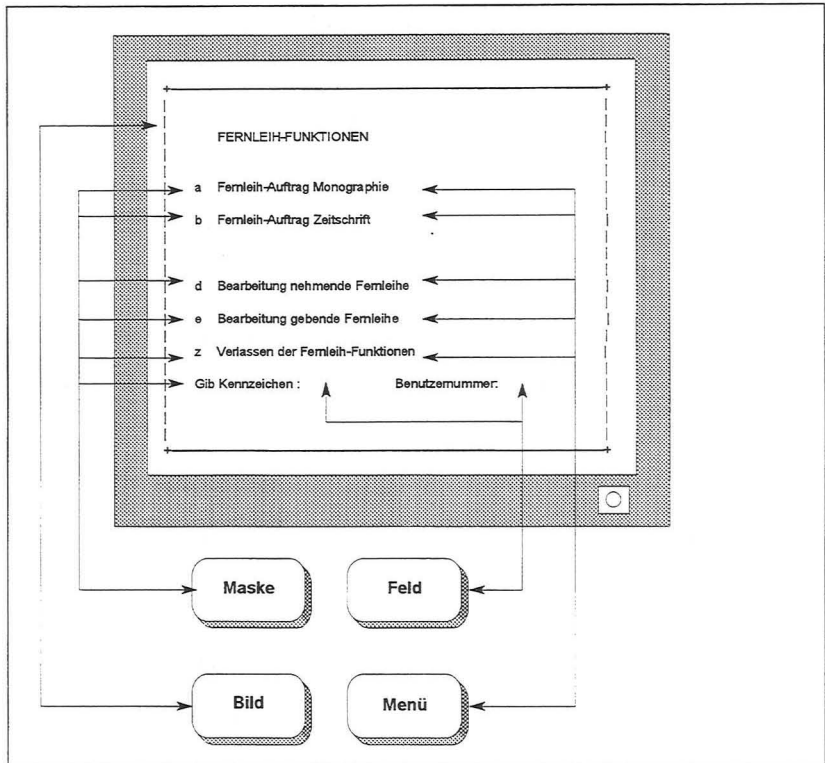


Bild-, Masken- und
Feldattribute

Bilder, Masken und Felder können mit unterschiedlichen Schriftarten bzw. Farben dargestellt werden. Die Einstellung erfolgt über **Bildattribute** (batr), **Maskenattribute** (matr) bzw. **Feldattribute** (fatr). Die Attribute haben folgenden einheitlichen Aufbau:

- Die Attribute bestehen aus fünf einzelnen Zeichen z.B. „NNNNN“
- Das erste Zeichen beschreibt den Schrifttyp beim Monochrom-Bildschirm bzw. die entsprechende Vordergrundfarbe am Farbbildschirm:

N = NORMAL	Standardschrifttyp oder Standardfarbe
U = UNDERLINE	Unterstreichung oder entsprechende Farbe
B = BOLD	Fett oder entsprechende Farbe
S = STANDOUT	besondere Hervorhebung oder entsprechende Farbe
R = REVERSE	Umkehrung oder entsprechende Farbe
B = BLINK	blinkend oder entsprechende Farbe
D = DIM	halbe Helligkeit oder entsprechende Farbe

Bei Anwendungen mit Farbbildschirm können auch individuelle Farben über eine Nummer (0..7) eingetragen werden.

- Das fünfte Zeichen beschreibt die Hintergrundfarbe von Bild, Maske oder Feld, falls ein Farbbildschirm benutzt wird. Standardeinstellung ist „N“ für Monochrom- und Farbbildschirm. Individuelle Farben sind:

0 = schwarz	3 = braun	6 = cyan
1 = rot	4 = blau	7 = weiß
2 = grün	5 = magenta	

Zusammensetzung einer BILD-Datei

Eine BILD-Datei besteht maximal aus 3 Teilen:

- Bildbeschreibungsteil,
- Maskenbeschreibungsteil,
- Feldbeschreibungsteil.

Bildbeschreibungsteil

Im **Bildbeschreibungsteil** sind folgende Beschreibungsmerkmale eines Bildes zusammengefaßt:

- Bildname,
- Bildposition (Zeile, Spalte, linker oberer Eckpunkt),
- Größe (Anzahl Zeilen, Anzahl Spalten),
- Attribute (Rahmen, Farbe, Füllmuster).

Abb. 6.14
Syntax der
Bildbeschreibung

BILD= *name oz ls az as batr*

BILD= Schlüsselwort zur Einleitung der Bildbeschreibung
name Name der Bilddatei max. 20 Zeichen (z.B. fcbild.pic)
oz oberste Zeile des Bildes (1..24)
ls linke Spalte des Bildes (1..80)
az Anzahl der Zeilen für dieses Bild
as Anzahl der Spalten für dieses Bild
batr Bildattribute (5 Zeichen hintereinander)
 batr(1): X= mit Rahmen N = Ohne Rahmen
 batr(2): frei N = Normal
 batr(3): frei N = Normal
 batr(4): frei N = Normal
 batr(5): Hintergrundfarbe N = Normal

Maskenbeschreibungsteil

Im **Maskenbeschreibungsteil** können beliebig viele Masken mit jeweils folgenden Merkmalen zusammengefaßt sein:

- Maskennummer (zum expliziten Anzeigen/Löschen),
- Maskenposition (Zeile, Spalte),
- Maskenstatus (sichtbar/unsichtbar),
- Attribute (Darstellungstyp, Farbe),
- Maskentext.

Abb. 6.15
Syntax der Masken-
beschreibung

```

MASKEN=  anzm
MASK=    nrmask mz ms mstat matr \mtext\
MASK=

MASKEN=  Schlüsselwort zur Einleitung des Maskenbeschreibungsteils
anzm    Anzahl der einzelnen Maskenbeschreibungen
MASK=    Schlüsselwort zur Einleitung einer Maskenbeschreibung
nrmask  Maskennummer (1...999)
          Die erste Maske kann eine sog. Hintergrundmaske sein. Ihr
          Maskentext darf beliebig lang sein. Alle anderen Masken
          texte dürfen maximal 80 Zeichen haben.
mz     Zeilennummer für die Maske (1...24)
ms     Spaltennummer für den Anfang der Maske (1...80)
mstat  Maskenstatus: 0 = keine Anzeige, 1 = wird angezeigt
matr  Maskenattribute (5 Zeichen hintereinander)
          matr(1): Farbe      N = Normal (Vordergrundfarbe)
          matr(2): frei       N = Normal
          matr(3): frei       N = Normal
          matr(4): frei       N = Normal
          matr(5): Farbe      N = Normal (Hintergrundfarbe)
mtext  Maskentext
          Der Maskentext darf Blanks enthalten. Am Anfang und am
          Ende muß ein '\ ' (Backslash) stehen.
    
```

Feldbeschreibungs-
teil

Im **Feldbeschreibungsteil** können einzelne Feldbeschreibungen mit jeweils folgenden Merkmalen zusammengefaßt sein:

- Feldnummer (zum expliziten Anzeigen/Löschen),
- Feldposition (Zeile, Spalte),
- Feldstatus (Eingabefeld/Anzeigefeld/unsichtbar),
- Attribute (Darstellung, Feldgrenze, Länge, Vorlöschung, Farbe),
- Datentyp (menüfeld, char, uc, short, int, long, float),
- Eingabelänge/Anzeigelänge (Anzahl Zeichen),
- Nummer des ersten Menüelementes und die Anzahl der Menüelemente bei Feldern vom Datentyp 'menüfeld'.

Abb. 6.16
Syntax der
Feldbeschreibung

```

FELDER=  anzf
FELD=    nrfeld zf sf fstat fatr ftyp flgt <MENUE= fmenü am>
FELD=

```

FELDER= Schlüsselwort zur Einleitung des Feldbeschreibungsteiles
anzf Anzahl der einzelnen Feldbeschreibungen
FELD= Schlüsselwort zur Einleitung einer Feldbeschreibung
nrfeld Nummer des Feldes (= Parameternummer 1...99)
zf Zeilennummer für das Feld (1...24)
sf Spaltennummer für den Anfang des Feldes (1...80)
fstat Feldstatus 0 = keine Anzeige 1 = nur Anzeige
 2 = Anzeige + Eingabe -1 = ausgeblendet
fatr Feldattribute (5 Zeichen hintereinander)
fatr(1): Farbe/Darstellung oder Vordergrundfarbe
fatr(2): Feldbegrenzung
 N = keine Grenze L = Limit
 L = Eingabe als Großbuchstabe L = Kleinbuchstabe
fatr(3): Vorangestellte Nullen vor einem Wert
 N = keine Vornullen Z = Vornullen
fatr(4): vorherige Löschung der Feldinhalte
 N = keine Löschung C = Clear
fatr(5): Farbe oder Hintergrundfarbe
ftyp Datentyp der zugehörigen Variablen
 -1 = zu diesem Eingabefeld gehört ein Maskenmenü
 -2 = zu diesem Eingabefeld gehört ein Feldmenü
 0 = char-string mit Platz für String-Endekennung
 1 = numerisch 1 Byte (unsigned char)
 2 = numerisch 2 Byte (short)
 3 = numerisch 2 oder 4 Byte
 4 = numerisch 4 Byte
 ly = Gleitkomma (float), y = Nachkommastellen
 Eingabefelder, denen ein Menü zugeordnet ist müssen den
 Datentyp 'char-string' haben.
flgt Feldlänge am Schirm (Anzeigelänge = Eingabelänge)
 Die Feldlänge ist inklusive Vorzeichen, Dezimalzeichen und
 Nachkommastellen.
MENUE= Schlüsselwort zur optionalen Einleitung der Menüzuordnung
 für ein Feld.
fmenü Erstes Menüelement für dieses Feld
am Anzahl der Menüelemente für dieses Feld

In der folgenden Abbildung ist ein Beispiel für eine BILD-Datei mit einem Menüfeld und vier zugeordneten Menümasken. Die BILD-Datei heißt fchpm.pic und bildet das Hauptmenü des Konstanzer Fernleihsystems:

Abb. 6.17
BILD-Datei fchpm.pic

```

BILD=      fchpm.pic  1 1 23 79 XNNNN

MASKEN=    9
MASK=      1 4 15 1 NNNNN \FERNLEIH-FUNKTIONEN\
MASK=      2 7 6 1 NNNNN \a Fernleih-Auftrag Monographie\
MASK=      3 9 6 1 NNNNN \b Fernleih-Auftrag Zeitschrift\
MASK=      4 11 6 0 NNNNN \c Zustandskontrolle eines Auftrags\
MASK=      5 13 6 1 NNNNN \d Bearbeitung nehmende Fernleihe\
MASK=      6 15 6 1 NNNNN \e Bearbeitung gebende Fernleihe\
MASK=      7 17 6 1 NNNNN \z Verlassen der Fernleih-Funktionen\
MASK=      8 19 6 1 NNNNN \Gib Kennzeichen : \
MASK=      9 19 40 1 NNNNN \Benutzernummer: \

FELDER=    2
FELD=      1 19 24 2>NNLNN -1 1  MENUE= 2 6
FELD=      2 19 55 2>NLNNN 0 9
    
```

Bildbeschreibungsteil

In der ersten Zeile der BILD-Datei wird die Bildbeschreibung mit dem Schlüsselwort „BILD=“ eingeleitet. Anschließend folgt der Name des Bildes (fchpm.pic) und die Koordinaten des Bildes. Die linke obere Ecke des Bildes beginnt in der ersten Zeile und in der ersten Spalte des Bildschirms. Das Bild hat insgesamt 23 Zeilen und 79 Spalte. Die Attribute „XNNNN“ geben an, daß das Bild einen Rahmen haben soll und die Standardschriftart beim Monochrom-Bildschirm bzw. die Standard-Vordergrundfarbe beim Farbbildschirm angezeigt werden soll. Bei der Hintergrundfarbe soll ebenfalls die Standardeinstellung verwendet werden.

Maskenbeschreibungsteil

Das Bild besteht insgesamt aus neun Masken. Die Masken sind von 1 bis 9 durchnummeriert. Die Nummer der Maske steht direkt hinter dem Schlüsselwort „MASK=“. Hinter der Maskennummer stehen die Koordinaten jeder einzelnen Maske und der Maskenstatus. Die Maske Nummer 1 hat die Angaben „ 4 15 1“, d.h. die Maske beginnt in Zeile 4 und Spalte 15 und soll standardmäßig angezeigt werden.

Feldbeschreibungsteil

Das Bild hat auch zwei Felder, die ebenfalls je eine Nummer haben. Das Feld mit der Nummer 1 beginnt in der Zeile 19 und Spalte 24. Die Zahl 2 gibt an, daß das Feld angezeigt wird und zugleich eine Eingabe in dieses Feld möglich ist. Die Attribute „>NNLNN“ sagen aus, daß die Darstellung beim Monochrom-Bildschirm oder die Vordergrundfarbe beim Farbbildschirm der Standardeinstellung entspricht. Ebenso ist das Feld nicht begrenzt, besitzt keine Vornull und Vorlöschung und die Hintergrundfarbe entspricht der Standardeinstellung.

Abb. 6.19
Erweiterte
Feldbeschreibung

```
FELD=nrfeld zf sf fstat fatr ftyp flgt <MENUE=fmenü am> ek sk
```

ek Aspektkey für den Empfang eines Wertes im Client
sk Aspektkey für das Senden eines Wertes zum Server

Beispiel:

```
FELDER= 2
FELD=   1 19 24 2>NNLNN 0 25   sigel   si
FELD=   2 19 55 2>NLNNN 0 9    bibnr   bn
```

Die Aspekte „sigel“ und „bibnr“ sind die Variablen für das Empfangen und „si“ und „bn“ für das Senden.

Senden Vor dem Senden werden im Sendepuffer alle Daten entsprechend dem VB-Protokoll aufgebaut. Alle Felder eines Bildes, bei denen eine Feldeingabe erfolgte, kommen mit ihrem Sendeky und dem zugehörigen Wert in den Sendepuffer und werden dann als Basiskommando zum Server geschickt.

Empfangen Empfängt der Client einen Datenblock vom Server, so wird geprüft, ob in diesem Datenblock die Empfangs-Aspektkeys eines bestimmten Bildes enthalten sind. Bei Übereinstimmung wird den entsprechenden Feldern des Bildes der zugehörige Wert aus dem empfangenen Datenblock zugeordnet und angezeigt.

6.2.2 Dialog-Client

TELNET-Programm Für die Entwicklungs- und Testphase wurde auf der Client-Seite unter UNIX und DOS lediglich ein **TELNET-Programm** eingesetzt. Damit war es möglich, ohne komplexe Dialogoberfläche alle Kommandos im Linemode zu testen und die Logik des gesamten Systems zu kontrollieren.

ADIX Mit Hilfe des Dialogsystems **ADIX** wurde anschließend eine zeichenorientierte Dialogoberfläche erstellt. Dieser Dialog-Client benutzt intern die Kommando-Funktionen wie sie beim Dialogserver aufgelistet sind. Die Bildbeschreibungen der ADIX-Dialoglaufzeit wurden für den Client-Server-Dialog, wie im vorhergehenden Abschnitt erläutert, ergänzt. Aufgrund dieser Erweiterung ist es möglich, jedem Feld eines Bildes einen Sende- und einen Empfangsidentifikator zuzuordnen. Diese Identifikatoren ermöglichen bei der Client-Server-Kommunikation die Zuordnung der Sende- und Empfangsinformationen zu den Feldern eines Bildes.

Aufruf Der **Aufruf** des DOS- oder UNIX-Clients erfolgt mit dem Kommando
*fc*lient *portnummer* *hostname*

Der Parameter „*portnummer*“ kennzeichnet den Port, unter dem der Server erreichbar ist und der Parameter „*hostname*“ ist der logische Name des entsprechenden Hosts.

Verbindungsaufbau Nach erfolgreichem **Verbindungsaufbau** wird auf dem Bildschirm des Clients das Begrüßungsbild des Servers angezeigt. Die Quittierung dieses Bildes führt dann zum Hauptmenü des Fernleih-Clients.

Hauptmenü

Das **Hauptmenü** besteht aus den Menüpunkten

- Fernleih-Auftrag Monographie,
- Fernleih-Auftrag Zeitschrift,
- Bearbeitung nehmende Fernleihe,
- Bearbeitung gebende Fernleihe.

Das Untermenü „**Fernleih-Auftrag Monographie**“ enthält eine Maske zur Eingabe aller Daten, die für die Bestellung einer Monographie relevant sind. Nachdem die Bestellaangaben gemacht worden sind, können diese gespeichert werden und es wird für diesen Auftrag eine Transaktionsnummer generiert. Anhand dieser Transaktionsnummer kann der Auftrag dann im Menüpunkt „**Bearbeitung nehmende Fernleihe**“ weiterbearbeitet werden.

Das Untermenü „**Fernleih-Auftrag Zeitschrift**“ ist analog zum Untermenü „**Fernleih-Auftrag Monographie**“ aufgebaut. Unterschiede bestehen lediglich in den relevanten Bestellaangaben.

Das Untermenü „**Bearbeitung nehmende Fernleihe**“ ist in mehrere Untermenüs unterteilt. Diese Untermenüs entsprechen den einzelnen Schritten, die bei der Bearbeitung eines nehmenden Fernleihvorganges durchgeführt werden müssen. Für jeden Bearbeitungsschritt gibt es eine Maske, die zur Ausführung der notwendigen Arbeiten dient.

Das Untermenü „**Bearbeitung gebende Fernleihe**“ ist ebenfalls in die entsprechenden Arbeitsschritte der gebenden Fernleihe gegliedert.

Die einzelnen Menüpunkte sind meist durch eine eigene Dialogfunktion realisiert. Ausnahmen bilden lediglich die Menüpunkte „Auswahl von nehmenden Vorgängen“ und „Auswahl von gebenden Vorgängen“. Diese sind zusammen mit ihren direkt untergeordneten Menüpunkten in einer gemeinsamen Dialogfunktion implementiert.

Jede Dialogfunktion ruft eine BILD-Datei auf. Die Namen der Dateien lauten „fc*.pic“. Die ersten beiden Buchstaben stehen für Fernleih-Client.

Wird in einer Dialogfunktion eine Anfrage an den Server gestellt, so wird das entsprechende Kommando zum Server gesendet. Die zur Verfügung stehenden Kommandos⁷³ für den Client-Server-Dialog und die Abhandlung der ankommenden Kommandos im Server wurden bereits beschrieben.

Baumstruktur

Die Verzweigungen der Menüs in Untermenüs und somit die Aueinanderfolge der einzelnen Dialogfunktionen kann in einer **Baumstruktur** dargestellt werden. Diese Baumstruktur ist sehr komplex und daher im Anhang IX zu finden. Die Darstellung der Reihenfolge in Tabellenform ist anschließend aufgelistet. Jede Dialogfunktion ist einer Ebene zugeordnet. Es gibt fünf Ebenen von A bis E. Eine Dialogfunktion der Ebene C wird aus einer Dialogfunktion der Ebene B aufgerufen und alle Untermenüs der Dialogfunktion der Ebene C sind in Ebene D. So wird z.B. der Menüpunkt C3.13.1 aus dem Menü B3.13 aufgerufen und die Untermenüs von C 3.13.1 lauten D3.13.1.1, D3.13.1.2 und D3.13.1.3.

⁷³ Siehe Abschnitt 6.1.5.

Client-Server-Architektur

Menüpunkt	Dialog-Funktion	Beschreibung der Dialogfunktion	BILD-Datei	Kom.
	dia_main	Hauptmenü	fchpm	
	dia_benu	Benutzernummer und Paßwort prüfen		cben
A1	dia_mbes	Fernleih-Auftrag Monographie	fcmbs	
A2	dia_zbes	Fernleih-Auftrag Zeitschrift	fczbes	
A3	dia_nehm	Bearbeitung nehmende Fernleihe	fcnehm	
A4	dia_gebe	Bearbeitung gebende Fernleihe	fcgebe	
B3.1	dia_bebearb	Auswahl von nehmenden Vorgängen	fcntyp	
B3.2	dia_best	Benutzer-Bestellung sichten/korrig.	fcbest	zbes kbes
B3.3	dia_leit	Leitwege zur Bestell. erfassen/korrig.	fcleit	zlei elei dlei
B3.4	dia_eink	Eingangsbearbeitung der Objekte	fceink	zein badr abvb abau
B3.5	dia_abenu	Ausleihe/Ausgabe an Benutzer	fcaben	zzur aaus
B3.6	dia_rbenu	Rückruf an Benutzer absetzen	fcrben	zzur arck
B3.7	dia_rcall	Rücksendung an Partnerbibliothek	fcrcall	zzur arca
B3.8	dia_rlief	Alle nehmenden Listen drucken	fcrлие	zzur azur
B3.9	dia_nedr	Fernleihscheine drucken	fcnedr	abib
B3.10	dia_fldr	Benutzernachricht aktivieren/bearb.	fcfldr	adrs
B3.11	dia_nabenu	FL-Vorgänge eines Benutzers anzeig.	fcnabe	zbes
B3.12	dia_lvor	Verbuchte FL einer Partnerbibliothek	fclvor	znfb znfp zgif
B3.13	dia_sndi	Auswahl spezielle Servicedienste	fcsndi	
B4.1	dia_part	Partnerbibliotheken bearbeiten	fcpart	zpar epar kpar dpar
B4.2	dia_anbearb	Auswahl von gebenden Vorgängen	fcgtyp	
B4.3	dia_anfr	Eine FL-Anfrage sichten/korrigieren	fcanfr	zanf eanf kanf
B4.4	dia_auspart	Ausleihe/Auslieferung bearbeiten	fcapar	zaus avbl avba
B4.5	dia_ruepart	Rückgabe von Partnerbibliothek	fcrpar	zaus avbr
B4.6	dia_negpart	negative Rückmeldung versenden	fcnpar	zaus aneg
B4.7	dia_gedr	Alle gebenden Listen drucken	fcgedr	ahol

B4.8	dia_lvor	Gebende Ausleihe Partnerbib. anz.	fcldvor	znfb znfp zgfp
B4.9	dia_sgdi	Auswahl spezielle Servicedienste	fcsgdi	
C3.1.1		Zustandsübersicht aller Vorgänge	fcnlis	zlzb
C3.1.2		neue unbearbeitete Bestellungen	fcnlis	znzu
C3.1.3		Bestellungen beim Bibliographieren	fcnlis	znzu
C3.1.4		Bestellungen mit zugeord. Leitwegen	fcnlis	znzu
C3.1.5		gesendete Bestellungen	fcnlis	znzu
C3.1.6		bereitgestellte Bestell. für Benutzer	fcnlis	znzu
C3.1.7		In Ausleihe befindliche Bestellungen	fcnlis	znzu
C3.1.8		Zur Rücksendung bereitst. Bestell.	fcnlis	znzu
C3.1.9		vollständig beendete Bestellvorgänge	fcnlis	znzu
C3.1.10		negative Bestellvorgänge	fcnlis	znzu
C3.13.1	dia_message	MESSAGE	fcsdme	
C3.13.2	dia_lost	LOST	fcldlo	
C3.13.3	dia_renew	RENEW/RENEW-ANSWER	fcldrm	
C3.13.4	dia_recall	RECALL	fcldrc	
C3.13.5	dia_damaged	DAMAGED	fcldbe	
C3.13.6	dia_overdue	OVERDUE	fcldov	
C3.13.7	dia_cancel	CANCEL/CANCEL-REPLY	fcldca	
C4.2.1		Akt. Zustandsübersicht aller Vorg.	fcglis	zlza
C4.2.2		neue unbearbeitete Anfragen	fcglis	zgzu
C4.2.3		Anfragen, die z. Z. b. Holdienst sind	fcglis	zgzu
C4.2.4		Anfragen, die verfügbar sind	fcglis	zgzu
C4.2.5		sendefähige negative Antworten	fcglis	zgzu
C4.2.6		verbuchte Ausleihen	fcglis	zgzu
C4.2.7		beendete Vorgänge	fcglis	zgzu
C4.9.1	dia_message	MESSAGE	fcldme	
C4.9.2	dia_lost	LOST	fcldlo	
C4.9.3	dia_renewans	RENEW/RENEW-ANSWER	fcldra	
C4.9.4	dia_erecall	RECALL	fcldrc	erec
C4.9.5	dia_damaged	DAMAGED	fcldbe	
C4.9.6	doa_eoverdue	OVERDUE	fcldov	eove
C4.9.7	dia_can_reply	CANCEL/CANCEL-REPLY	fcldcr	
D3.13.1.1	dia_emen	MESSAGE senden	fcldmel	emen
D3.13.1.2	dia_zumn	Ungelesene MESSAGE anzeigen	fcnlis	zumn
D3.13.1.3	dia_medr	Ungelesene MESSAGE drucken	fcldmedr	amel
D3.13.2.1	dia_elon	LOST senden	fcldmel	elon
D3.13.2.2	dia_zuhn	Ungelesene LOST anzeigen	fcnlis	zuhn
D3.13.2.3	dia_losdr	Ungelesene LOST drucken	fcldlosdr	alos
D3.13.3.1	dia_eren	RENEW senden	fclderen	eren
D3.13.3.2	dia_zrenew_ans	Pos. RENEW-ANSWER anzeigen	fcnlis	zran
D3.13.3.3	dia_drrenew_ans	Positive RENEW-ANSWER drucken	fcldrenadr	aran
D3.13.3.4	dia_zrenew_ans	Neg. RENEW-ANSWER anzeigen	fcnlis	zran
D3.13.3.5	dia_drrenew_ans	Neg. RENEW-ANSWER drucken	fcldrenadr	aran
D3.13.4.1	dia_zrecall	Empfangene RECALL anzeigen	fcnlis	zrec
D3.13.4.2	dia_drrecall	Empfangene RECALL drucken	fcldrecdr	arec
D3.13.5.1	dia_edan	DAMAGED senden	fcldmel	edan

Client-Server-Architektur

D3.13.5.2	dia_zudn	Empfangene DAMAGED anzeigen	fcnlis	zudn
D3.13.5.3	dia_adam	Empfangene DAMAGED drucken	fcdamdr	adam
D3.13.6.1	dia_zoverdue	Empfangene OVERDUE anzeigen	fcnlis	zove
D3.13.6.2	dia_droverdue	Empfangene OVERDUE drucken	foovedr	aove
D3.13.7.1	dia_ecancel	CANCEL senden	fccan	ecan
D3.13.7.2	dia_zcan_rpn	nicht erfüllbare CANCEL anzeigen	fcnlis	zcn
D3.13.7.3	dia_drcan_rpn	nicht erfüllbare CANCEL drucken	fccrpndr	acrn
D4.9.1.1	dia_emeg	MESSAGE senden	fcmel	emec
D4.9.1.2	dia_zumg	Ungelesene MESSAGE anzeigen	fcglis	zumg
D4.9.1.3	dia_medr	Ungelesene MESSAGE drucken	fcmedr	amel
D4.9.2.1	dia_elog	LOST senden	fcmel	alog
D4.9.2.2	dia_zulg	Empfangene LOST anzeigen	fcglis	zulg
D4.9.2.3	dia_losdr	Empfangene LOST drucken	fclosdr	alos
D4.9.3.1	dia_zrenew	Neu empfangene RENEW zeigen	fcglis	zren
D4.9.3.2	dia_drrenew	Neu empfangene RENEW drucken	fcrendr	aren
D4.9.3.3	dia_erna	RENEW-ANSWER senden	fcerna	erna
D4.9.3.4	dia_zrenew_ags	Pos. RENEW-ANSWER anzeigen	fcglis	zrag
D4.9.3.5	dia_drrenew_ans_g	Positive RENEW-ANSWER drucken	fcrenadr	arag
D4.9.3.6	dia_zrenew_ags	Neg. RENEW-ANSWER anzeigen	fcglis	zrag
D4.9.3.7	dia_drrenew_ans_g	Neg. RENEW-ANSWER drucken	fcrenadr	arag
D4.9.5.1	dia_edag	DAMAGED senden	fcmel	edag
D4.9.5.2	dia_zudg	Empfangene DAMAGED anzeigen	fcglis	zudg
D4.9.5.3	dia_adam	Empfangene DAMAGED drucken	fcdamdr	adam
D4.9.7.1	dia_zcancel	Neu empfangene CANCEL anzeigen	fcglis	zcan
D4.9.7.2	dia_drcancel	Neu empfangene CANCEL drucken	fccandr	acan
D4.9.7.3	dia_ecan_rep	CANCEL-REPLY senden	fcacre	ecre
D4.9.7.4	dia_zcan_rep	Nicht genehmigte CANCEL anzeigen	fcglis	zcrg
D4.9.7.5	dia_drcan_rpg	Nicht genehmigte CANCEL drucken	fccrpndr	acrg
E3.13.1.2.1	dia_zmen	Eine einzelne MESSAGE anzeigen	fcmel	zmen kmen
E3.13.2.2.1	dia_zeln	Einen einzelnen LOST anzeigen	fcmel	zeln keln
E3.13.5.2.1	dia_zedn	Einen einzelnen DAMAGED anz.	fcmel	zedn kedn
E4.9.1.2.1	dia_zmeg	Eine einzelne MESSAGE anzeigen	fcmel	zmeg kmeg
E4.9.2.2.1	dia_zelg	Einen einzelnen LOST anzeigen	fcmel	zelg kelg
E4.9.5.2.1	dia_zedg	Einen einzelnen DAMAGED anz.	fcmel	zedg kedg

7

Prinzip der Kodierung und Dekodierung der ILL-Daten

7.1 Allgemeines Vorgehen

Kodierung

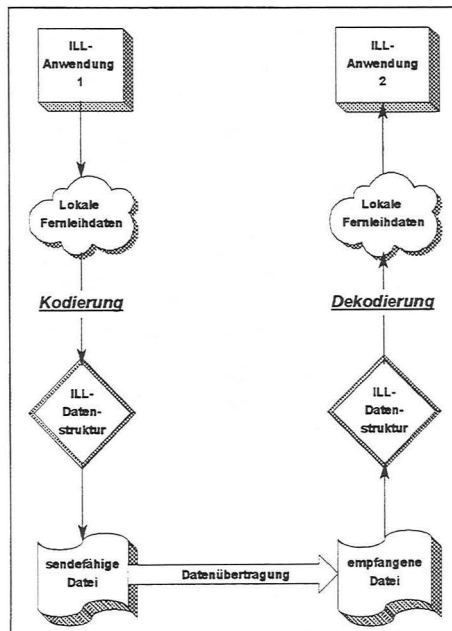
Wie bereits in Kapitel 4.3 beschrieben, müssen alle obligatorischen und optionalen lokalen Daten der ILL-Anwendung beim Versenden eines ILL-Dienstes an den richtigen Stellen in die komplette Datenstruktur des ILL-Protokolls eingefügt werden. Dieses Vorgehen wird als **Kodierung** bezeichnet. Die Datenstruktur wird im weiteren Verlauf als ILL-Datenstruktur bezeichnet. Die gefüllte ILL-Datenstruktur wird anschließend als Bytestrom in eine Datei geschrieben. Diese Datei wird dann mittels den in Kapitel 5.1 bzw. 5.2 unterstützten Kommunikationssystemen zum Partnersystem übermittelt.

Dekodierung

Der Empfang eines ILL-Dienstes geht genau den umgekehrten Weg. Dabei wird die über das jeweilige Kommunikationssystem empfangene Datei ausgelesen und die Inhalte als Bytestrom direkt in eine leere ILL-Datenstruktur geschrieben. Schließlich müssen nun die übertragenen Daten aus der Datenstruktur wieder in die lokalen Datenformate des Partnersystems gebracht werden. Diesen Vorgang bezeichnet man entsprechend als **Dekodierung**.

Die folgende Abbildung soll das allgemeine Vorgehen verdeutlichen.

Abb. 7.1
Kodierung und
Dekodierung



7.2 Die Kodierung

Sendeserver Im Folgenden soll nun die Realisierung der Kodierung im Fernleihsystem der Uni Konstanz beschrieben werden. Wie bereits in Abschnitt 6.1.1 erläutert, hat der Sendeserver die Aufgabe, zyklisch in einer Zustandstabelle⁷⁴ der Datenbank nachzusehen, ob ein sendefähiger ILL-Dienst vorhanden sind. Ist dies der Fall, liest der Sendeserver aus der Datenbank die für den ILL-Dienst notwendigen Nutzdaten aus und stellt sie im Hauptspeicher in Form sogenannter Satzpuffer⁷⁵ bereit. Diese sind in ihrem Gültigkeitsbereich global für alle Server-Prozesse des Fernleihsystems. Schließlich verzweigt der Sendeserver in eine Routine zur Kodierung der im Hauptspeicher bereitstehenden Daten.

encode_apdu(...) Diese Routine hat den Namen „**encode_apdu(...)**“ und hat als Parameter

- die ILL-Dienst-Nummer des zu versendenden ILL-Dienstes und
- eine ihm zugehörige, eindeutige ILL-Transaktions-Id.

Die ILL-Transaktions-Id wurde bei der Eröffnung der ILL-Transaktion von der ILL-Anwendung generiert.

eimain.c Die Routine ist als externe Funktion realisiert, wird also bei der Erstellung des ausführbaren Server-Programms hinzugebunden - die zugehörige Datei heißt **eimain.c**.

Die Funktion ist wie eine Art Sprungverteiler, der in Abhängigkeit des jeweiligen zu kodierenden ILL-Dienstes in weitere externe Funktionen verzweigt. Diese haben dann das Füllen der Datenstruktur mit den dienstbezogenen Werten aus den globalen Satzpuffern zur Aufgabe. Bevor jedoch in diese Funktionen verzweigt wird, wird als erstes die Information, welcher ILL-Dienst übertragen werden soll, in der ILL-Datenstruktur vermerkt. Danach beginnt der eigentliche Sprungverteiler. Dort wird in jedem Zweig der aktuelle globale ILL-Zustand⁷⁶ aus der Datenbank ausgelesen. Der Schlüssel für die Suche ist dabei der zweite übergebene Parameter - die ILL-Transaktions-Id.

Jetzt wird in die dienstabhängigen Kodierungsfunktionen verzweigt, in denen die ILL-Datenstruktur mit den restlichen notwendigen Werten gefüllt wird. Nach dem Füllen wird der globale ILL-Transaktionszustand aktualisiert⁷⁷ und in die Datenbank zurückgeschrieben. Schließlich wird außerhalb des Sprungverteilers eine Datei eröffnet und der gesamte Inhalt der mittlerweile gefüllten ILL-Datenstruktur als Bytestrom in diese Datei geschrieben. Dieses Vorgehen wurde in Kapitel 4.2 ausführlich beschrieben. Das Schreiben in die Datei übernimmt die Funktion „**enc_ILL APDU(...)**“, die eine vom ASN.1 - Compiler automatisch generierte Standardfunktion ist⁷⁸.

illsend.tmp Die Datei hat den Namen „**illsend.tmp**“ und wird im Anschluß an die Kodierung mittels der zur Verfügung stehenden Kommunikationssysteme an den Fernleihpartner verschickt.

⁷⁴ Siehe Abschnitt 8.1.

⁷⁵ Siehe Abschnitt 8.3.

⁷⁶ Siehe Abschnitt 3.4.4.

⁷⁷ Zustandsübergänge siehe Abschnitt 3.4.4.5.

⁷⁸ Siehe Abschnitt 4.5.

Ein Pseudo-C-Code soll die Abläufe in der Funktion „encode_apdu(...)“ veranschaulichen:

Abb. 7.2
Pseudo-C-Code von
encode_apdu(...)

```

encode_apdu(int ILL-Dienst-Nummer, string ILL-Transaktion-Id)
{
    ...
    Setzen der ILL-Dienst-Nummer in der ILL-Datenstruktur;
    ...
    switch(ILL-Dienst-Nummer)
    {
        case ILL-Dienst-1: {
            Lesen des globalen ILL-Zustands
            aus Datenbank(ILL-Transaktion-Id);

            Verzweigung in eine externe Kodie-
            rungsfunktion, die das dienstbezo-
            gene Füllen der ILL-Datenstruktur
            erledigt;

            Aktualisieren des globalen ILL-
            Zustands;

            Zurückschreiben des globalen ILL-
            Zustands in die Datenbank(ILL-
            Transaktion-Id);

            break;
        }

        ...
        ...
        ...

        case ILL-Dienst-22: {
            ...
            break;
        }

    }

    Öffnen der Datei „illsend.tmp“;

    Schreiben der gefüllten ILL-Datenstruktur als
    Bytestrom in die Datei „illsend.tmp“;
}

```

eimain.h

Die Kodierungsfunktionen, welche für das dienstabhängige Füllen der ILL-Datenstruktur zuständig sind, bestehen im Großen und Ganzen aus einfachen Zuweisungen - den Strukturelementen aus der ILL-Datenstruktur werden Werte aus den im Hauptspeicher bereitgestellten Satzpuffern zugewiesen. Für die im ILL-Protokoll möglichen Optionalitäten existiert die Header-Datei „eimain.h“, in der für jede im Protokoll mögliche Optionalität eine Boolean-Variable mit Vorbelegung bereitgestellt ist. In den Kodierungsfunktionen wird die Belegung der zugehörigen Variablen bei jeder Wahlmöglichkeit abgeprüft. Der optionale Wert wird anschließend je nach Vorbelegung der zugehörigen Variable in die ILL-Datenstruktur eingetragen oder auch nicht.

Zu Dokumentationszwecken ist im Anhang IV eine Liste der Namen aller externen Kodierungsfunktionen mit zugehörigem ILL-Dienst und Dateinamen aufgeführt.

7.3 Die Dekodierung

Empfangsserver,
„illrecv.tmp“

Die Dekodierung geht nun, wie in Abschnitt 7.1 erwähnt, genau den umgekehrten Weg. Das für die Dekodierung von empfangenen Daten in lokale Formate des Empfänger-ILL-Systems zuständige Serverprogramm ist im Fernleihsystem der Uni Konstanz der **Empfangsserver**⁷⁹. Die Daten, die über eines der unterstützten Kommunikationssysteme übertragen wurden, werden zunächst vom Empfangsserver in einem Pufferspeicher abgelegt. Ist die Übertragung abgeschlossen, wird der Inhalt des Pufferspeichers in die Datei „illrecv.tmp“ geschrieben.

decode_apdu()

Diese Datei muß mit der in Abschnitt 7.2 beschriebenen Datei „illsend.tmp“ identisch sein, ansonsten ist ein Fehler in der Übertragung aufgetreten. Anschließend verzweigt der Empfangsserver in die Routine zur Dekodierung der empfangenen Daten. Diese Routine hat den Namen „decode_apdu()“.

dimain.c

Auch diese Routine ist extern und ist in Analogie zur Kodierungsroutine unter dem Dateinamen „dimain.c“ zu finden. Da, wie erwähnt, die Dekodierung den umgekehrten Weg geht, wird in dieser Routine als erstes die Datei „illrecv.tmp“ in eine leere ILL-Datenstruktur eingelesen. Das Lesen als Bytestrom aus der Datei übernimmt die Funktion „dec_ILL_APDU(...)“, die ebenfalls eine vom ASN.1 - Compiler automatisch generierte Standardfunktion ist.

Wie bei der Kodierung wird nun die Information über den übertragenen ILL-Dienst lokal gespeichert und im Folgenden je nach übertragenem Dienst in eine weitere externe Dekodierungsfunktionen verzweigt. In dieser werden die Nutzdaten aus der standardisierten ILL-Datenstruktur in lokale Formate und Strukturen der ILL-Anwendung umgewandelt und stehen nach der Dekodierung im Hauptspeicher in den entsprechenden Satzpuffern zur Verfügung. Nun muß noch ein Zustandswechsel für die ILL-Transaktion durchgeführt werden. Dazu wird zunächst der aktuelle Zustand der betroffenen ILL-Transaktion aus der Datenbank ausgelesen. Je nach empfangenem ILL-Dienst wird nun der definierte Zustandswechsel⁸⁰ durchgeführt und die Datenbank bezüglich des Folgezustands aktualisiert.

Wird ein ILL-Dienst empfangen, der im aktuellen Zustand der ILL-Transaktion unzulässig ist, wird als letzte Aktion in jedem Zweig des Sprungverteilers die automatische Generierung eines Error-Report angestoßen. Die automatische Generierung eines STATUS-OR-ERROR-REPORT⁸¹ und das automatische Verschicken an den Versender des gerade empfangenen, unzulässigen ILL-Dienstes ist allerdings zum jetzigen Zeitpunkt noch nicht vollständig implementiert.

⁷⁹ Siehe Abschnitt 6.1.2.

⁸⁰ Zustandsübergänge siehe Abschnitt 3.4.4.5.

⁸¹ Siehe Abschnitt 3.5.21.

Wie schon bei der Kodierung soll auch hier ein Pseudo-C-Code die Abläufe in der Funktion „decode_apdu()“ veranschaulichen:

Abb. 7.3
Pseudo-C-Code von
decode_apdu()

```

decode_apdu()
{
    ...

    Öffnen der Datei „illrecv.tmp“;

    Lesen des Inhalts der Datei „illrecv.tmp“ als Bytestrom
    in eine leere ILL-Datenstruktur;

    Setzen der ILL-Dienst-Nummer aus der ILL-Datenstruktur;
    ...

    switch(ILL-Dienst-Nummer)
    {
        case ILL-Dienst-1: {
            Verzweigung in eine externe Kodie-
            rungsfunktion, die das dienstbezo-
            gene Füllen der lokalen Satzpufer
            aus der ILL-Datenstruktur er-
            ledigt;

            Lesen des globalen ILL-Zustands
            aus Datenbank(ILL-Transaktion-Id);

            Aktualisieren des globalen ILL-
            Zustands;

            Zurückschreiben des globalen ILL-
            Zustands in die Datenbank(ILL-
            Transaktion-Id);

            Generierung eines Error-Reports,
            falls der empfangene ILL-Dienst
            unzulässig war;

            break;
        }

        ...
        ...
        ...

        case ILL-Dienst-22: {
            ...
            break;
        }

    }
}

```

dimain.h

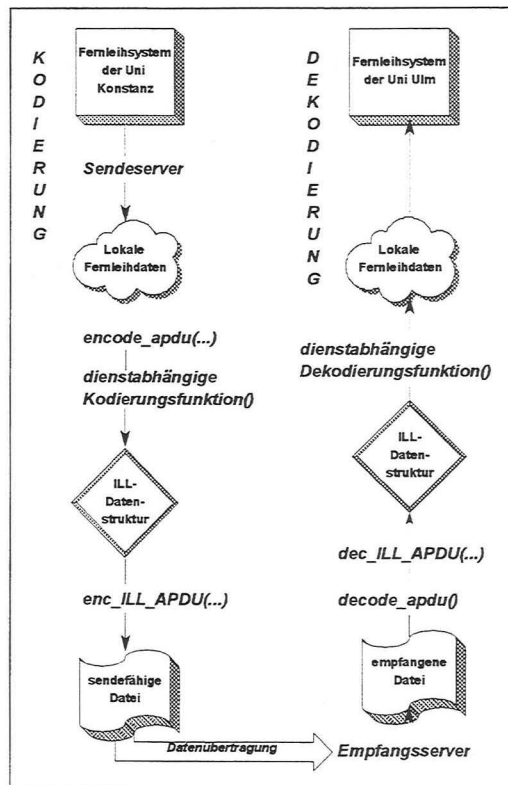
Das Dekodieren aus der ILL-Datenstruktur in die lokalen Satzpufer erfolgt analog zur Kodierung mit einfachen Zuweisungen. Auch die Dekodierung von im ILL-Protokoll als optional definierten Werten geschieht analog zur Kodierung. Die Entscheidung, ob ein Wert dekodiert wird oder nicht, hängt bei der Dekodierung eben von der Belegung der entsprechenden Boolean-Variablen in der ILL-Datenstruktur ab. Hier sind die für die Optionalitäten vorgesehenen Variablen in der Datei „dimain.h“ zu finden.

Prinzip der Kodierung und Dekodierung der ILL-Daten

Zu Dokumentationszwecken ist im Anhang IV eine Liste der Namen aller externen Dekodierungsfunktionen mit zugehörigem ILL-Dienst und Dateinamen aufgeführt. Es sind aber nur die im Fernleihsystem der Uni Konstanz implementierten ILL-Dienste aufgelistet.

Abschließend soll das Prinzip der Kodierung und Dekodierung von ILL-Daten für das Fernleihsystem der Uni Konstanz anhand der folgenden Abbildung veranschaulicht werden. Dabei wird vorausgesetzt, daß die Uni Ulm ebenfalls das von der Uni Konstanz entwickelte Fernleihsystem einsetzt.

Abb. 7.4
Kodierung und
Dekodierung im
Fernleihsystem der
Uni Konstanz



8

Die Fernleihdatenbank

INGRES, ORACLE
oder ADABAS

Die Datenbasis für das Fernleihsystem an der Uni Konstanz bildet wahlweise die netzwerkorientierte INGRES-Datenbank in der Version 6.2, die relationale ORACLE-Datenbank in der Version 7.015 oder die ebenfalls netzwerkorientierte ADABAS-Datenbank. Der Einsatz der ADABAS-Datenbank wurde im Fernleihsystem allerdings bislang noch nicht getestet. Mittels eines Preprozessor-Defines in der Datei „fedb.ora“ kann eine Auswahl zwischen den zur Verfügung stehenden Datenbanksystemen getroffen werden.

8.1

Die Tabellen der Fernleihdatenbank

Das dem Fernleihsystem der Uni Konstanz zugrunde liegende Datenmodell wird in insgesamt 13 Tabellen abgebildet. Im folgenden wird lediglich beschrieben, welche Aufgaben diese Tabellen bei der Abwicklung und lokalen Verwaltung von ILL-Transaktionen haben. Der genaue Aufbau jeder einzelnen Tabelle kann dem Anhang V entnommen werden.

Tabellen der
Fernleihdatenbank

- **ref_ill_tab**

Diese Referenztabelle dient der Zuordnung zwischen der lokal generierten Fernleih-Bestellnummer und der daraus entstandenen ILL-Transaktions-Identnummer.

- **illglobal_tab**

Mit dieser Tabelle werden die globalen Zustände der ILL-Transaktionen dokumentiert und geprüft.

- **partner_tab**

Diese Tabelle beschreibt die Fernleihpartner. Jeder mögliche Partner belegt einen Satz in dieser Tabelle.

- **part_fach_tab**

Die Tabelle enthält die Fachgebietsschwerpunkte der Partner.

- **part_stop_tab**

Hier können Liefereinschränkungen, Schließzeiten der Partner etc. eingetragen sein.

- **part_stat_tab**

Diese Tabelle enthält statistische Daten über Partnerbibliotheken.

- **nehm_zu_tab**

Diese Tabelle enthält die Verwaltungsdaten solcher ILL-Transaktionen, die beim Requester in Bearbeitung sind (nehmende Ausleihe). Jede ILL-Transaktion belegt eine variable Anzahl von Sätzen in dieser Tabelle. Ein Satz kennzeichnet einen bestimmten Zustand (Aktivität) der ILL-Transaktion. Die Summe aller Sätze zu einer bestimmten ILL-Transaktion beschreibt somit deren chronologisch detaillierten Ablauf.

- **nehm_ausl_tab**

Die Tabelle beschreibt die Leihobjekte, die dem Requester von den Partnern zur Ausleihe überlassen wurden.

- **leitweg_tab**

Diese Tabelle enthält die nach dem Bibliographieren in Frage kommenden Zielbibliotheken.

- **nehm_best_tab**

Hier sind die bibliographischen Daten der Fernleihaufträge von den Kunden des Requesters abgelegt.

- **geb_ausl_tab**

Die Tabelle beschreibt die Leihobjekte, die den Fernleihpartnern vom Responder zur Ausleihe überlassen wurden.

- **geb_zu_tab**

Diese Tabelle enthält die Verwaltungsdaten solcher ILL-Transaktionen, die beim Responder in Bearbeitung sind (gebende Ausleihe). Jede ILL-Transaktion belegt eine variable Anzahl von Sätzen in dieser Tabelle. Ein Satz kennzeichnet einen bestimmten Zustand (Aktivität) der ILL-Transaktion. Die Summe aller Sätze zu einer bestimmten ILL-Transaktion beschreibt somit deren chronologisch detaillierten Ablauf.

- **geb_best_tab**

Hier sind die bibliographischen Daten der Fernleihaufträge von Kunden der Requester fremder Bibliotheken abgelegt.

Die Datenbanktabellen werden mit folgenden SQL-Skripten erstellt, gefüllt, modifiziert und auch wieder gelöscht.

- (1) db.crea_fern ↔ Generieren aller 13 Tabellen des Fernleihsystems.
- (2) db.modi_fern ↔ Indizieren einiger Tabellen des Fernleihsystems.
- (3) db.load_fern ↔ Laden der Partnerdaten in die Tabelle 'partner_tab' aus der sequentiellen Datei „benper.ur“. Dabei wird das Programm „fepartur“ verwendet, dessen Gebrauch im folgenden noch beschrieben wird.
- (4) db.drop_fern ↔ Löschen aller 13 Tabellen des Fernleihsystems.

Mit dem letzten Skript werden die Skripte (4), (1), (3) und (2) genau in dieser Reihenfolge ausgeführt. Alle dabei auftretenden Meldungen werden in der Datei „dbgen.prot“.

(5) db.gen_fern ↔ Ausführen der Skripte (1) - (4) in der o.g. Reihenfolge.

Die Skripte (1) und (2) sind in Anhang V und Anhang VI zu finden. Aus dem Skript (1) läßt sich auch der detaillierte Aufbau jeder einzelnen Tabelle mit allen ihren Feldern entnehmen.

Urladen mit
„fepartur *Dateiname*“

Das **Urladen** der Fernleihdatenbank läßt sich außer mit dem SQL-Skript (3) auch noch über das Programm „fepartur“ durchführen. Dieses Programm benötigt als einzigen Parameter eine Datei, in der alle Partnerdaten entsprechend dem Aufbau der Partnertabelle sequentiell enthalten sind.

Der Aufruf ist somit „fepartur *Dateiname*“.

8.2

Zugriff auf die Tabellen der Fernleihdatenbank

Basisprozeduren,
Datenbankunabhängigkeit

Der Zugriff aller Programm-Module des Fernleihsystems auf die Daten in den Tabellen erfolgt grundsätzlich über die sogenannten **Basisprozeduren**. Diese Prozeduren enthalten Embedded-SQL-Anweisungen für den Zugriff auf die Datenbank. Dadurch wird optimale **Datenbankunabhängigkeit** erreicht. Alle Basisprozeduren sind in der Datei „fedb.ora“ enthalten. Nur diese Datei ist somit für den SQL-Precompiler relevant. Die vorhandenen Basisprozeduren wurden in der Entwicklungsphase mit den Datenbanksystemen INGRES und ORACLE getestet.

Die Funktionsnamen der Basisprozeduren haben einen normierten Aufbau - sie bestehen aus drei Teilen, die jeweils mit einem Unterstrich voneinander getrennt sind. Der normierte Aufbau der Funktionsnamen lautet,

„*Zugriffsart_Zugriffsziel_Zugriffskey*“,

wobei die einzelnen Teile folgendes bedeuten:

- **Zugriffsart** ↔ rk = read mit eindeutigem Key
dk = delete mit eindeutigem Key
uk = update mit eindeutigem Key
wr = write eines kompletten Satzes
te = Trefferliste⁸² definieren, der Key ist equal
tg = Trefferliste definieren, der Key ist greater
tl = Trefferliste definieren, der Key ist lower
ts = Trefferliste definieren, ohne Key
- **Zugriffsziel** ↔ Kurzbezeichnung der Tabelle, auf die zugegriffen werden soll.
- **Zugriffskey** ↔ Kurzbezeichnung des Keys in der Tabelle, auf die zugegriffen werden soll.

⁸² Dieser Begriff wird im weiteren Verlauf dieses Abschnitts noch näher erläutert.

Rückgabewerte Alle Basisprozeduren liefern als **Rückgabe** folgende **Werte** vom Typ „integer“ an das aufrufende Programm zurück:

- 0 ↔ Der Zugriff wurde korrekt durchgeführt.
- < 0 ↔ Es ist ein Zugriffsfehler aufgetreten.
- 100 ↔ EOF⁸³ in der Tabelle, d.h. es konnte z.B. bei einem lesenden Zugriff auf eine Tabelle kein dem Suchkriterium (Key) entsprechender Datensatz gefunden werden.

Bei denjenigen Basisprozeduren, bei denen der Datenbankzugriff über einen Key stattfindet, wird dieser Key als Parameter benutzt. Vor dem ersten Zugriff auf eine Tabelle wird die Verbindung zum Datenbanksystem mit folgenden Funktionsaufrufen initialisiert bzw. wieder abgebaut:

- `fe_init(„Datenbankname“);`
- `fe_close(„Datenbankname“);`

Trefferlisten ↔
Cursorlisten

Neben dem einfachen lesenden Zugriffen auf eine Tabelle mittels eines eindeutigen Keys, gibt es, wie in der Beschreibung der Zugriffsart bereits erwähnt, noch die Möglichkeit, eine **Trefferliste** zu definieren. Eine Trefferliste ist eine Untermenge von Datensätzen einer Tabelle, die einem angegebenen Suchkriterium (Key) entsprechen. Der Begriff „Trefferliste“ ist ein Begriff aus der Bibliothekswelt und ist äquivalent zum Begriff der „Cursorliste“, der von SQL her bekannt ist. Im weiteren Verlauf wird der Begriff der „Trefferliste“ verwendet. Es stehen folgende Funktionen für die Abhandlung eines Zugriffs über eine Trefferliste zur Verfügung.

- `cu_open(„Name der Trefferliste“, &Größe_der_Trefferliste);`

In dieser Funktion wird die Trefferliste durch die entsprechenden SQL-Kommandos geöffnet. Der zweite Parameter enthält nach dem Aufruf der Funktion die Anzahl der Datensätze, die dem Suchkriterium entsprechen haben. Der interne Satzzeiger (Cursor) steht auf dem ersten Satz der Trefferliste.

- `cu_read(„Name der Trefferliste“);`

Diese Funktion liest gemäß der aktuellen Position des internen Satzzeigers einen Satz aus der Trefferliste aus. Der interne Satzzeiger wird nach dem Aufruf der Funktion inkrementiert.

- `cu_delete(„Name der Trefferliste“);`

Diese Funktion löscht gemäß der aktuellen Position des internen Satzzeigers einen Satz aus der Trefferliste.

- `cu_close(„Name der Trefferliste“);`

Diese Funktion schließt eine geöffnete Trefferliste und gibt den internen Satzzeiger wieder frei.

⁸³ End of File.

Als Konvention für die Namesgebung der Trefferlisten gilt, daß als „Name der Trefferliste“ der Name der Funktion genommen wird, mit der die Trefferliste definiert worden ist. Die oben aufgeführten Funktionen sind alle in der Datei „fedb.ora“ zu finden.

8.3 Bereitstellung von Daten aus den Tabellen der Fernleihdatenbank

Nach einem lesenden Zugriff auf eine Tabelle der Fernleihdatenbank sollen die entsprechenden Daten im Hauptspeicher zur weiteren Verarbeitung zur Verfügung stehen.

globale
Satzpuffer

Dies geschieht beim Fernleihsystem der Uni Konstanz durch den Einsatz sogenannter globaler Satzpuffer. Diese Satzpuffer sind C-Strukturen, deren Aufbau identisch zu den im SQL-Skript (1) definierten Tabellen ist. Ihr Gültigkeitsbereich ist global für alle Serverprozesse des Fernleihsystems. Auch hier gibt es eine Konvention für die Namensgebung dieser Satzpuffer:

„sp_Kurzbezeichnung der zugehörigen Tabelle“

Die folgende Abbildung zeigt beispielhaft den Aufbau der Tabelle „partner_tab“ mit dem zugehörigen Satzpuffer „sp_part“.

Abb. 8.1
Tabelle „partner_tab“
mit Satzpuffer
„sp_part“

CREATE TABLE partner_tab		STRUCT sp_part {	
(s_sigel	char (25),	char s_sigel[26];	Suchsigel (normierte Schreibweise)
sigel	char (25),	char sigel[26];	Sigelnnummer
bibnr	char (9),	char bibnr[10];	Bibliotheknummer
sname	char (60),	char sname[61];	Suchname (normierte Schreibweise)
bname	char (60),	char bname[61];	Name der Bibliothek
land	char (40),	char land[41];	Land
plz	char (10),	char plz[11];	Postleitzahl
ort	char (40),	char ort[41];	Ort
straße	char (40),	char straße[41];	Straße oder Postfach
email	char (160),	char email[161];	Email-Adresse
xmail	char (160),	char xmail[161];	X.400-Mail-Adresse
host	char (160),	char host [161];	Hostname für tcp direkt
port	char (20),	char port [21];	Portnummer für tcp direkt
fax	char (20),	char fax[21];	Fax-Nr.
telefon	char (20),	char telefon[21];	Tel.-Nr.
partner	char (20),	char partner[21];	Ansprechpartner in der Bibliothek
lief_kond	char (20),	char lief_kond[21];	Lieferkonditionen
best_art	char (2),	char best_art[3];	Schein='s', Tcp 't', X.400='x'
lief_art	char (2),	char lief_art[3];	Auto='a' o. ' ', Post='p'
stop_krit	char (1),	char stop_krit[2];	Wenn != blank, dann part_stop_tab
bemerkung	char (160)	char bemerkung[161];	Bemerkungstext
);		} sp_part;	

fetable.h

Die Definition aller im Fernleihsystem eingesetzten Satzpuffer findet man in der Datei „fetable.h“.

In den derzeitigen Versionen der eingesetzten Datenbanksysteme ist es allerdings nicht möglich, in den Basisprozeduren diese Satzpuffer direkt anzugeben, da der SQL-Precompiler innerhalb der SQL-Kommandos nur Elementarvariablen zulässt und keine Strukturen.

Zwischenvariablen

Aus diesem Grund sind speziell für die Embedded-SQL-Funktionen **Zwischenvariablen** definiert, die alle Elemente jeder einzelnen Satzpufferstruktur abdecken. Diese Zwischenvariablen sind in der Datei „*fedb.ora*“ für jeden Satzpuffer definiert.

schreibender
Datenbankzugriff

Entsprechend zu einem lesenden Zugriff müssen die Daten bei einem **schreibenden Zugriff** vor dem Aufruf der jeweiligen Embedded-SQL-Funktion in dem entsprechenden Satzpuffer bereitstehen. Dafür sorgen die Serverprogramme.

Um das Kopieren der Variablenwerte aus den Satzpuffern in diese Zwischenvariablen (schreibender Datenbankzugriff) bzw. aus den Zwischenvariablen in die Satzpuffer (lesender Datenbankzugriff) zu erleichtern, gibt es Funktionen, die genau diesen Kopiervorgang ausführen. Die Namensgebung für diese Funktionen richtet sich nach folgender Konvention:

- **into_Name des Satzpuffers()**

Die Variablenwerte der Elemente des Satzpuffers werden in die zugehörigen Zwischenvariablen kopiert.

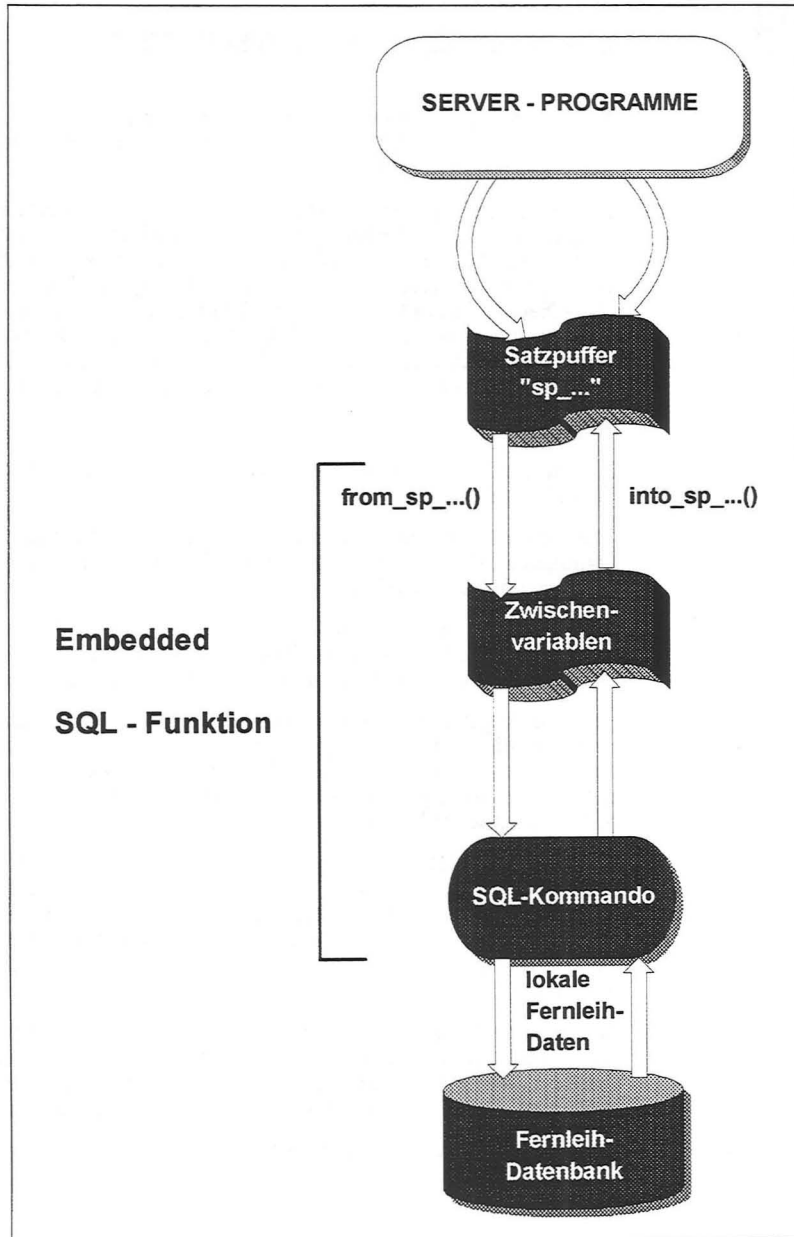
- **from_Name des Satzpuffers()**

Die Werte der Zwischenvariablen werden in die zugehörigen Elemente des Satzpuffers kopiert.

Diese Funktionen findet man ebenfalls in der Datei „*fedb.ora*“.

Zusammenfassend läßt sich das Vorgehen anhand der folgenden Abbildung verdeutlichen:

Abb. 8.2
Das Bereitstellen von
Daten der Fernleih-
datenbank



9

Lokale Zustände im Fernleihsystem

Wie bereits in Abschnitt 3.4.4 erklärt, gibt es durch das ILL-Protokoll vorgeschriebene globale ILL-Transaktionszustände, die den Bearbeitungszustand einer ILL-Transaktion bestimmen.

Mit diesen Transaktionszuständen wird aber nur das Verhältnis der miteinander kommunizierenden Bibliotheken genau bestimmt. Die intern abgehandelten Dienstwege bei einer Fernleihe können bei den jeweiligen Bibliotheken jedoch auch individuell verschieden sein. Wie die Organisation und der Ablauf eines ILL-Dienstes intern bestimmt wird, hängt allein von der entsprechenden Bibliothek ab. Im Fernleihsystem der Universitätsbibliothek Konstanz wurden deshalb sogenannte lokale Zustände eingeführt, die sowohl die vorgeschriebenen Transaktionszustände als auch die internen Bearbeitungszustände abbilden können.

Beispiel:

Ein Student gibt eine Fernleihbestellung in Auftrag. Damit wird der globale ILL-Zustand IDLE aktiviert.

Die Fernleihabteilung prüft die Bestellung des Studenten zunächst auf Vollständigkeit. Mit Hilfe der angegebenen Daten wird dann bibliographiert und die Leitwege für die Bestellung ermittelt.

Gibt es Fehler in den Bestellaangaben des Studenten, dann muß dieser benachrichtigt und die Korrektur der Bestellung durchgeführt werden. Danach erfolgt erneut das Bibliographieren und das Ermitteln von Leitwegen.

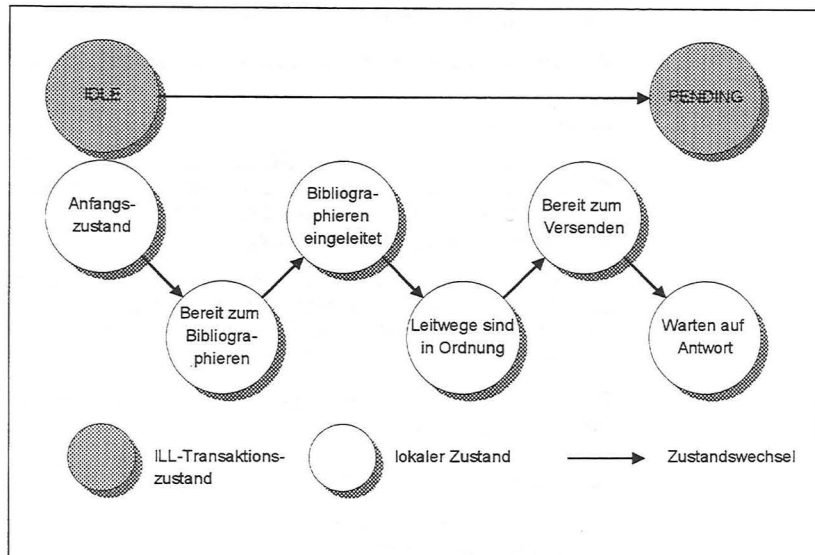
Sind die Angaben für die Fernleihbestellung vollständig, dann werden die Leitwege für die Bestellung eingetragen und erst jetzt erfolgt die eigentliche Fernleihbestellung von einer Bibliothek an die andere. Das bedeutet, erst jetzt wechselt der bis dahin gültige globale ILL-Transaktionszustand IDLE in den Transaktionszustand PENDING.

Durch die globalen Zustände wird der gesamte interne Ablauf der Bestellung durch den Studenten bis zum eigentlichen Versenden des Auftrags an einen Responder nur durch den Zustand IDLE abgebildet. Für die lokale Verwaltung der Fernleihe ist das nicht ausreichend, wenn die Abläufe automatisiert werden sollen. Daher sind im Konstanzer Fernleihsystem lokale Zustände eingeführt worden.

Im vorliegenden Beispiel wechselt der Bestellvorgang im günstigsten Fall vom „Anfangszustand“ in den Zustand „Bereit zum Bibliographieren“, dann in den Zustand „Bibliographieren wurde eingeleitet“ und „Leitwege sind in Ordnung“. Nach dem „Bereit zum Versenden“ folgt das „Warten auf die Antwort“.

Der folgende Zustandsgraph verdeutlicht noch einmal die Unterschiede.

Abb. 9.1
Vergleich eines globalen Zustandswechsels mit den entsprechenden lokalen Zustandsübergängen



Um einen lokalen Zustand näher zu definieren, erhält er eine Primärkennung, die eindeutig ist. Außerdem ist er noch durch die Sekundärkennung genauer beschrieben. Die Kennungen sind auch in den Zustandstabellen „nehm_zu_tab“ und „geb_zu_tab“ der Fernleihdatenbank als Datenfelder „prim_zu“ und „sek_zu“ enthalten.

Beispiel:

Der Zustand der nehmenden Fernleihe „benach_nz“ gibt an, daß ein Benutzer benachrichtigt werden soll. Er besitzt als Primärkennung den Wert 1 und als Sekundärkennung den Wert 1 bis 60.

Hat die Sekundärkennung einen Wert von 1 bis 20, dann wird ein Fernleihvorgang beendet. Beispielsweise soll der Benutzer eine Nachricht erhalten, daß sein Bestellwunsch nicht erfüllt werden kann.

Liegt der Wert der Sekundärkennung im Bereich von 21 bis 40, so ist der Folgezustand ein Wartezustand. Es soll dann auf die Antwort des Benutzers gewartet werden.

Werte zwischen 41 und 60 geben an, daß die Bearbeitung des Vorgangs normal weiterlaufen soll. Mit dem Sekundärwert wird also nicht nur der Zustand genauer bestimmt, sondern auch die Folgezustände, die möglich sind, genau festgelegt.

Die Sekundärkennung ist jedoch nicht in jedem Fall notwendig.

Eine genaue Auflistung aller möglichen Zustände, der Folgezustände und der zugehörigen Primär- und Sekundärkennungen befindet sich im Anhang VII!

9.1 Lokale Zustände der nehmenden Fernleihe

Folgende lokale Zustände kennzeichnen die nehmende Bearbeitung:

anfangs_zu

Ein Fernleihauftrag wird über PC oder ein OPAC-Terminal erfaßt und in die Fernleihdatenbank als unbearbeitete Bestellung aufgenommen. Formale Kontrollen auf Autor, Titel usw. werden vorgenommen. Ein Gebührensatz wird im lokalen Verbuchungssystem angelegt.

Der Vorgang wird zum Bibliographieren bereitgestellt, wenn alle Angaben zur Bestellung vollständig gemacht wurden. Ist die Bestellung ungültig, weil wichtige Informationen fehlen oder falsche Benutzerangaben gemacht wurden, dann wird der Vorgang sofort beendet.

Hat der Benutzer unvollständige Angaben in der Bestellung gemacht, muß er benachrichtigt werden.

benach_nz

Der Benutzer soll benachrichtigt werden. Die Nachrichtenart ist in drei verschiedene Gruppen aufgeteilt:

- Der Bestellwunsch des Benutzers kann nicht erfüllt werden. Der Grund dafür wird angegeben.
- Nach dem Versenden der Nachricht erfolgen weitere Aktionen der Bibliothek erst, nachdem der Benutzer eine Antwort auf die Nachricht gegeben hat.
- Der Benutzer erhält nur eine informative Nachricht ohne Auswirkung auf den normalen Fernleihverkehr.

bereit_bib_nz

Die Bestellung ist zum Bibliographieren bereit.

bib_aktiv_nz

Das Bibliographieren wurde eingeleitet. Die Leitwege sind zu erfassen.

leitweg_ok_nz

Die Bestellung ist sendefähig. Sie kann über den nächsten eingetragenen, noch unbearbeiteten Leitweg verschickt werden.

send_mail_nz

Vorgänge mit diesem Zustand werden vom Sendeserver verschickt. Der Sendeserver kann mit Hilfe der Sekundärkennung unterscheiden, welche ILL-APDU gesendet werden soll, d.h. zu welchem Dienst die Sendung gehört.

wart_illreq_nz

Die Bestellung wurde verschickt. Die nehmende Bibliothek wartet nun auf eine Antwort der gebenden Bibliothek oder das bestellte Leihobjekt. Bei diesem Zustand wird mit Hilfe der Sekundärkennung unterschieden, ob die Bestellung auf elektronischem Weg erfolgte oder konventionell ein Leihschein verschickt wurde.

zeitlimit_nz

Eine systeminterne Wartezeit wurde überschritten, eine Folgeaktion muß angestoßen werden. Die Sekundärkennung kennzeichnet, worauf gewartet wurde.

Es wird unterschieden zwischen dem:

- Warten auf die Antwort zu einer ILL-Sendung,
- Warten auf Quellenangaben durch den Benutzer,
- Warten auf eine Benutzerantwort, die Kostenübernahme betreffend,
- Warten auf Abholung der Bücher, die für die Ausleihe bereitgestellt wurden.

objekt_no_nz

Das Objekt ist nicht vorhanden oder nicht ausleihbar.

wart_benant_nz

Warten auf eine Benutzerantwort.

benu_bereit_nz

Die Lieferung wurde empfangen und für den Benutzer bereitgestellt. Der Sender erhielt eine Quittung über den Empfang des Leihobjekts.

benu_ausl_nz

Das Objekt wurde an den Benutzer ausgeliehen, auf die Rückgabe wird gewartet.

ruecksend_nz

Das Objekt wurde vom Benutzer an die nehmende Bibliothek zurückgegeben. Jetzt kann die Rücksendung an die gebende Bibliothek erfolgen.

quit_neg_nz

Die Leitwege zu einem Bestellvorgang konnten alle nur negativ abgearbeitet werden. Entweder wird jetzt ein neuer Leitweg erfaßt oder der Benutzer muß eine entsprechende Nachricht erhalten und der gesamte Leihvorgang beendet werden.

beendet_nz

Der Vorgang ist abgeschlossen. Nicht benötigte Informationen können gelöscht werden. Daten, die für statistische Zwecke gebraucht werden, müssen zunächst bestehen bleiben.

unread_mail_nz

Eine Nachricht zu einem Fernleihvorgang von einer gebenden Bibliothek wurde empfangen. Die Nachricht wurde noch nicht zur Kenntnis genommen.

read_mail_nz

Eine empfangene Nachricht zu einem Fernleihvorgang wurde zur Kenntnis genommen, d. h. sie wurde entweder gelesen oder ausgedruckt.

wart_ren_ans_nz

Eine Verlängerungsanfrage zu einem Vorgang wurde gestellt. Nun wird auf eine Antwort von der gebenden Bibliothek gewartet.

empf_ren_ans_pos_nz

Es wurde eine positive Antwort auf eine Verlängerungsanfrage zu einem Vorgang von der gebenden Bibliothek empfangen. Die gebende Bibliothek hat entweder das von der nehmenden Bibliothek vorgeschlagene neue Ausleihdatum akzeptiert oder sie hat eine eigene neue Ausleihfrist gesetzt.

empf_ren_ans_neg_nz

Die Anfrage auf Ausleihfristverlängerung wurde abgelehnt. Das Objekt muß zum ursprünglich festgelegten Termin zurückgesendet werden.

empf_lost_nz

Die nehmende Bibliothek hat von der gebenden eine Verlustanzeige erhalten.

empf_recall_nz

Die nehmende Bibliothek hat von der gebenden einen vorzeitigen Rückruf erhalten. Sie muß diesen an den Benutzer weiterleiten und so die Rückgabe des Objekts veranlassen.

empf_damaged_nz

Die nehmende Bibliothek hat von der gebenden die Information erhalten, daß das Leihobjekt beschädigt ist.

empf_overdue_nz

Eine Nachricht über den regulären Ablauf der Leihfrist wurde empfangen. Der Benutzer ist zu benachrichtigen.

wart_can_rep_nz

Es wird auf einen CANCEL-REPLY, also auf eine Antwort auf einen Stornierungswunsch gewartet.

empf_can_rep_pos_nz

Die nehmende Bibliothek hat einen positiven CANCEL-REPLY empfangen.

empf_can_rep_neg_nz

Die nehmende Bibliothek hat einen negativen CANCEL-REPLY empfangen.

9.2

Lokale Zustände der gebenden Fernleihe

Folgende lokale Zustände kennzeichnen die gebende Bearbeitung:

anfangs_gz

Eine Fernleihanfrage einer nehmenden Bibliothek wurde empfangen. Sie wurde noch nicht bearbeitet. Diese Bearbeitung erfordert die Ermittlung der Signatur des Leihobjekts, weiter sind der Ausleihstatus und die Verbuchungsnummer zu ermitteln.

unerfuellbar_gz

Die Anfrage einer Partnerbibliothek kann nicht erfüllt werden. Entweder ist das gewünschte Objekt langfristig ausgeliehen oder es ist z.Zt. unauffindbar.

verfuegbar_gz

Das gewünschte Leihobjekt ist verfügbar, ein Ausdruck der Bestelldaten für den Holdienst soll erfolgen.

ausleih_gz

Das Objekt ist ausleihfähig. Es wurde durch den Holdienst bereitgestellt. Konditionen für die Ausleihe müssen erfaßt werden. Bei der Ausleihverbuchung wird ein Lieferschein gedruckt und das Objekt wird für den Versand bereitgestellt.

send_mail_gz

Vorgänge mit diesem Zustand werden vom Sendeserver verschickt. Der Sendeserver kann durch der Sekundärkennung ermitteln, welche ILL-APDU geschickt werden soll, also zu welchem ILL-Dienst die Sendung gehört.

wart_return_gz

Das Objekt wurde verliehen. Es wird auf die Rücksendung durch die nehmende Bibliothek gewartet.

zeitlimit_gz

Die Wartezeit auf die Rückgabe eines verliehenen Objekts wurde überschritten. Die Zeitkontrolle erfolgt durch das Verbuchungssystem. Eine Mahnung oder ein Rückruf muß angestoßen werden.

erledigt_gz

Der Vorgang gilt als abgearbeitet. Die Daten zur Fernleihanfrage können gelöscht werden, sofern sie nicht für statistische Auswertungen benötigt werden.

unread_mail_gz

Eine Nachricht zu einem Fernleihvorgang von einer nehmenden Bibliothek wurde empfangen. Die Nachricht wurde noch nicht zur Kenntnis genommen.

read_mail_gz

Eine empfangene Nachricht zu einem Fernleihvorgang wurde zur Kenntnis genommen, d. h. sie wurde entweder gelesen oder ausgedruckt.

empf_ren_gz

Ein Wunsch auf Verlängerung der Leihfrist wurde empfangen.

erl_rena_pos_gz

Ein Verlängerungswunsch wurde bearbeitet. Die Verlängerung der Leihfrist wurde gewährt. Entweder wurde das vorgeschlagene neue Ausleihdatum akzeptiert oder ein andere Leihfrist gesetzt.

erl_rena_neg_gz

Der Verlängerungswunsch wurde abgewiesen.

empf_lost_gz

Eine Verlustbenachrichtigung wurde von der nehmenden Bibliothek empfangen. Damit muß der Fernleihvorgang abgeschlossen werden.

empf_damaged_gz

Eine Beschädigungsanzeige wurde von der nehmenden Bibliothek empfangen.

empf_can_gz

Ein Wunsch auf Stornierung eines Auftrags wurde von der nehmenden Bibliothek empfangen und muß beantwortet werden.

erl_can_rep_pos_gz

Der Wunsch auf Stornierung wurde der nehmenden Bibliothek gewährt. Der Fernleihvorgang muß abgeschlossen werden.

erl_can_rep_neg_gz

Der Wunsch auf Stornierung wurde der nehmenden Bibliothek abgelehnt. Der Fernleihvorgang bleibt bestehen.

9.3 Die Zustandsübergänge

Die Zustandsübergänge sind von der Situation bei der Bearbeitung der entsprechenden Fernleihbestellung abhängig. Daher kann der Ablauf bei der Bearbeitung eines ILL-Dienstes unterschiedliche lokale Zustandswechsel zur Folge haben. Die Wechsel unterliegen jedoch in jedem Fall der logischen Konsequenz, die sich aus dem Bearbeitungsablauf eines Dienstes ergibt.

In den folgenden Beispielen wird davon ausgegangen, daß Requester und auch Responder das Fernleihsystem der Bibliothek der Universität Konstanz benutzen. In den Abbildungen sollen noch einmal die Unterschiede bei globalen und lokalen Zustandsübergängen verdeutlicht werden. Als Grundlage dient der normale Ablauf der Bearbeitung einer Fernleihanfrage.

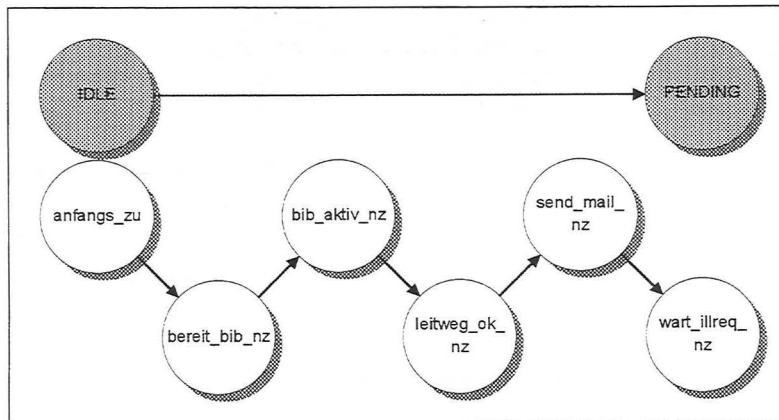
Abb. 9.2
Legende zu den
folgenden Zustands-
graphen



Der Zustandswechsel in den Abbildungen wird immer durch ein Kommando oder ein Serverprogramm durchgeführt.

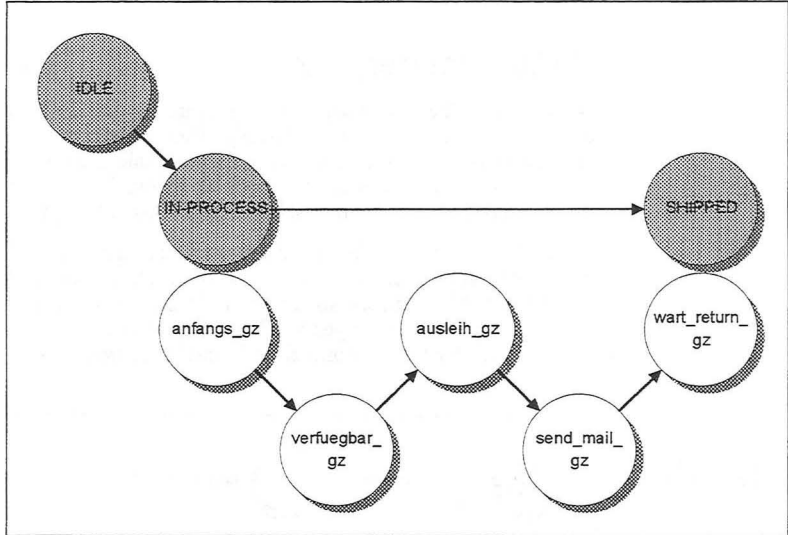
Die Bestellung wird durch den Studenten am PC erfasst. Anschließend wird sie in der Fernleihabteilung kontrolliert, bibliographiert und erhält eine Leitweg-Eintragung. Danach wird sie an die gebende Bibliothek weitergeleitet.

Abb. 9.3
Zustandsübergänge
bei Erfassung einer
Fernleihbestellung



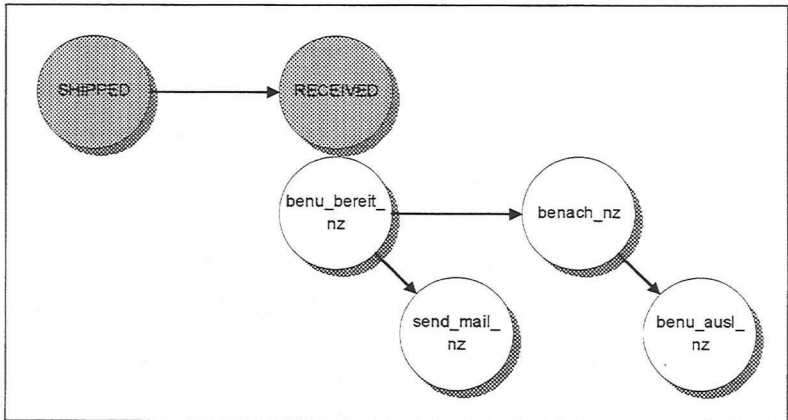
Von dieser wird die Bestellung weiterbearbeitet und die in diesem Fall gewünschte Monographie wird geliefert.

Abb. 9.4
Zustandswechsel bei
Bearbeitung eines
Bestelleingangs auf
der gebenden Seite



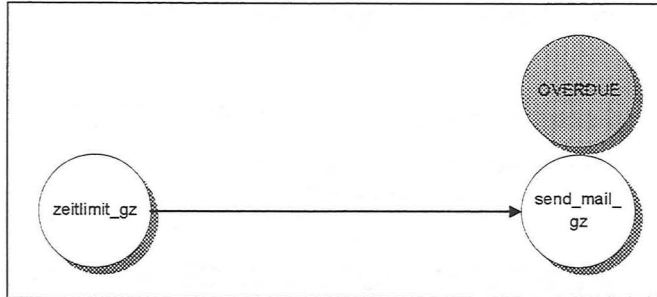
Auf der nehmenden Seite wird nach Erhalt des Buches die Benutzerbenachrichtigung veranlaßt. Das Buch wird ausgeliehen. Die Benachrichtigung des Benutzers ist ebenso wie die Bestätigung, daß das Leihobjekt bei der nehmenden Bibliothek eingetroffen ist (Zustand send_mail_nz(RECEIVED)), optional.

Abb. 9.5
Zustandswechsel
nach Empfang des
Leihobjekts bei der
nehmenden Biblio-
thek



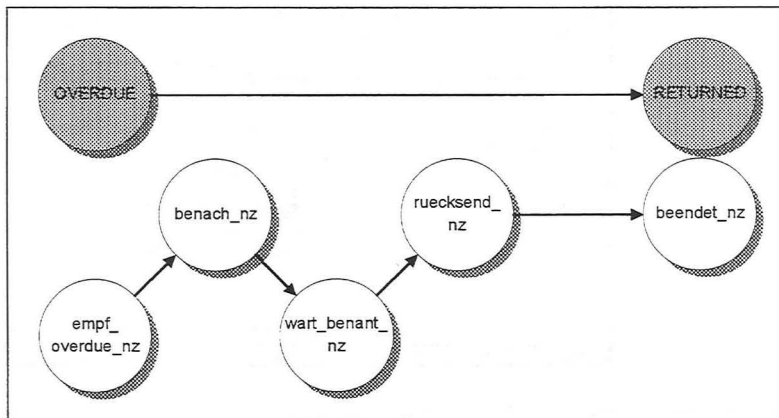
Nach dem Ablauf der Leihfrist sendet die gebende Bibliothek eine Nachricht zur Leihfristüberschreitung.

Abb. 9.6
Zustandsübergang bei Feststellung der Leihfristüberschreitung auf gebender Seite



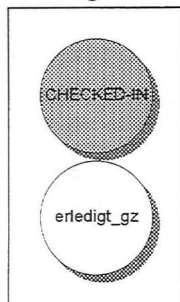
Ist das Buch in diesem Moment noch nicht zurückgegeben worden, erhält der Benutzer von der nehmenden Bibliothek seine Rückgabeaufforderung und bringt das Buch zurück. Die nehmende Bibliothek liefert die Monographie wieder an die gebende aus. Damit ist der Vorgang auf der nehmenden Seite abgeschlossen.

Abb. 9.7
Zustandswechsel nach Empfang eines ILL-OVERDUE



Bei der gebenden Bibliothek erfolgt jetzt noch die Rückbuchung.

Abb. 9.8
Ende der aktiven Vorgangsbearbeitung



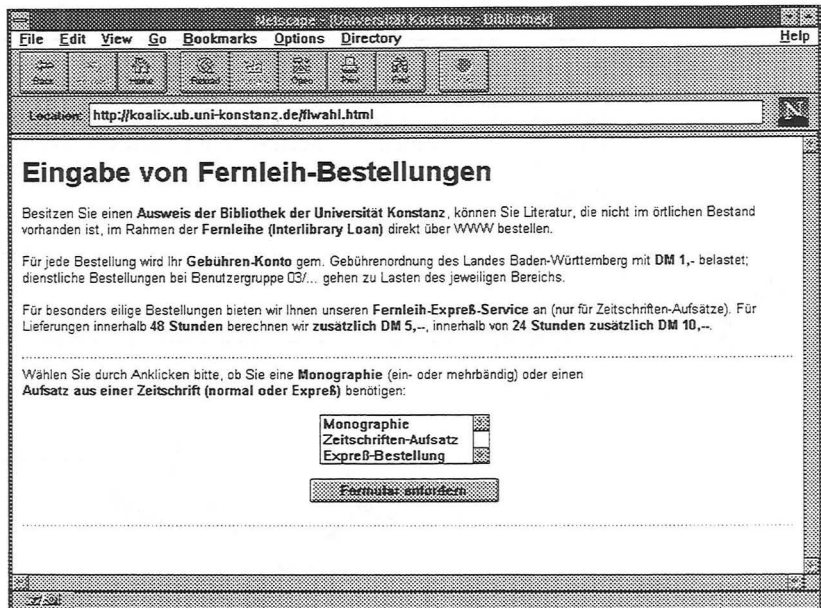
10

Anbindung wichtiger Bibliotheksdienste

10.1 Fernleih-Bestelleingabe über WWW-Client

Eine Fernleihbestellung kann auch im World Wide Web (WWW) eingegeben werden. Auf den Seiten der Bibliothek der Universität Konstanz kann über das Menü „WWW-Service-Funktionen“ folgende Seite ausgewählt werden:

Abb. 10.1
Auswahlmenü für die
Eingabe von
Fernleihbestellungen



Differenzierung zwischen Monographie und Zeitschrift

Bei einer Fernleihbestellung unterscheidet man zunächst zwischen den verschiedenen Objekten, die bestellt werden können. Man differenziert zwischen einer Monographie und einer Zeitschrift. Eine **Monographie** ist ein ein- oder mehrbändiges Buch, das ausgeliehen wird und wieder zurückgegeben werden muß. Eine **Zeitschrift** hingegen kann nicht ausgeliehen werden. Hier werden die gewünschten Seiten kopiert und dem Antragsteller gegen Entgelt überlassen.

Differenzierung zwischen normaler und Express-Bestellung

Desweiteren wird bei einer Fernleihbestellung von Zeitschriften-Aufsätzen zwischen zwei Zustellungsgeschwindigkeiten unterschieden. Es gibt die normale Bestellung und die Express-Bestellung. Bei der **normalen Bestellung** erhält der Auftragsteller die gewünschten Kopien nach einer üblichen Bearbeitungsdauer von ein bis zwei Wochen. Bei einer **Express-Bestellung** wird innerhalb von 24 oder 48 Stunden geliefert.

Bei dem Auswahlmenü in der vorangegangenen Abbildung muß also durch Markieren von einer der Möglichkeiten und Anklicken des Buttons „Formular anfordern“ das entsprechende Eingabeformular verlangt werden. Es existieren drei verschiedene Bestellformulare:

- Formular für die Bestellung einer Monographie.
- Formular für die normale Bestellung eines Zeitschriften-Aufsatzes.
- Formular für die Expres-Bestellung eines Zeitschriften-Aufsatzes.

10.1.1 Fernleihbestellung einer Monographie

Pflichtfelder

Bei der Bestellung einer Monographie muß der Leser zunächst seine Benutzernummer und sein Paßwort eingeben. Die Paßworteingabe ist zwingend notwendig um vor einer versehentlichen oder mißbräuchlichen Bestellung für eine fremde Benutzernummer zu schützen. Desweiteren sind die Felder „dienstlich benötigt“ (nur bei Uni-Mitarbeitern mit den Benutzernummern 03/...), „Mehrkosten-Übernahme“, „Storno-Datum“, „Autor/Herausgeber“ und „Buchtitel“ Pflichtfelder.

Abb. 10.2
Fernleihbestellung
für eine Monographie

The screenshot shows a web browser window with the address bar containing 'http://koalix.ub.uni-konstanz.de/cgi-bin/fwbest'. The main content area is titled 'Fernleih-Bestellung für eine Monographie (TEST)'. The form contains the following elements:

- Benutzername: [input field]
- Passwort: [input field]
- Nur bei Ausw. Nr. 03... dienstlich benötigt? ja nein
- Übernahme von evtl. Mehrkosten bis DM 10, ohne Rückfrage ja nein
- Bestellung stornieren, wenn nicht erfüllt: bis: [input field] (f. rom. g.)
- Autor/Trag: [input field]
- Titel: [input field]
- Untertitel: [input field]
- Kurzgezeichnet: [input field]
- Erach.-Ort: [input field] Verlag: [input field]
- Erach.-Jahr: [input field] Seiten: [input field] Auflage: [input field] ISBN: [input field]
- Serientitel: [input field]
- Bemerkungen: [input field]
- Quelle: [input field]
- Buttons: 'Bestellung absenden' and 'Eingabe löschen'

10.1.2 Fernleihbestellung eines Zeitschriften-Aufsatzes

Pflichtfelder

Bei der Bestellung eines Zeitschriften-Aufsatzes sind wie bei der Bestellung einer Monographie folgende Felder **Pflichtfelder**: „Benutzernummer“, „Paßwort“, „dienstlich benötigt“ (nur bei Uni-Mitarbeitern mit den Benutzernummern 03/...), „Mehrkosten-Übernahme“, „Storno-Datum“, „Name der Zeitschrift“ und „Verfasser“.

Abb. 10.3
Fernleihbestellung
eines Zeitschriften-
Aufsatzes

The screenshot shows a Netscape browser window with the URL <http://koalibx.ub.uni-konstanz.de/cgi-bin/flwbest>. The page title is "Fernleih-Bestellung für Zeitschriften-Aufsatz (TEST)". The form contains the following fields and options:

- Benutzernr.: Passwort: Nur bei Ausweis-Nr. (Gr... dienstlich benötigt? ja nein
- übernahme von evtl. Mehrkosten bis DM 10, ohne Rückfrage ja nein
- Bestellung stornieren, wenn nicht erfüllbar bis: [11.mm.jj]
- Zeitschrift:
- Errech.-Ort: Verlag:
- Bd./Vol./Nr.: Errech.-Jahr: ISSN:
- Gewünschter Aufsatz:
- Seiten:
- Verfasser:
- Aufs.-Titel:
- Bemerkungen:
- Quelle:

Buttons at the bottom: "Bestellung abschicken" and "Eingabe löschen".

10.1.3 Expresß-Bestellung eines Zeitschriften-Aufsatzes

Pflichtfelder

Bei der Expresß-Bestellung eines Zeitschriften-Aufsatzes sind ähnliche Angaben zu machen, wie bei der normalen Bestellung. **Pflichtfelder** sind wieder „Benutzernummer“, „Paßwort“, „dienstlich benötigt“ (nur bei Uni-Mitarbeitern mit den Benutzernummern 03/...), „Name der Zeitschrift“ und „Verfasser“. Hinzu kommen noch die Wahl zwischen „24-Stunden-Lieferung“ oder „48-Stunden-Lieferung“.

Abb. 10.4
Express-Bestellung
eines Zeitschriften-
Aufsatzes

10.1.4

Client-Server-Prinzip bei der Fernleihbestellung im WWW

Auf dem WWW-Server befindet sich das HTML-Skript *flwahl.html*. Dieses Skript bildet das Auswahlmenü aus Abb. 10.1 und wird zum WWW-Client geschickt, sobald ein Benutzer am Client-Terminal diese Seite anfordert. Nachdem der Benutzer dann die Auswahl getroffen hat, werden je nach Wahl die entsprechende Seiten zum Server gesendet. Der WWW-Server realisiert dann einen Link zum CGI-Interface *flwbest*. Dieses Interface schickt je nach eingegangenem Wunsch das zugehörige Skript zum Client zurück, so daß das gewünschte Bestellformular angezeigt wird. Nachdem der Benutzer die Bestellung ausgefüllt und zum Server gesendet hat, führt das CGI-Interface eine Transaktionskontrolle durch und quittiert die Bestellung oder lehnt sie ab.

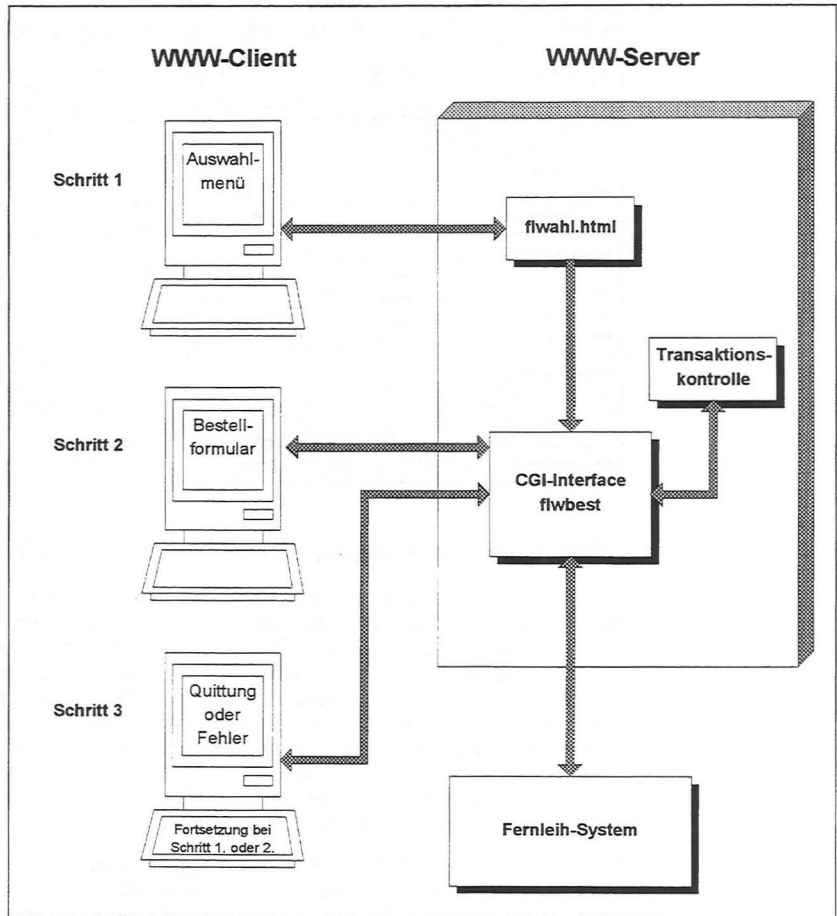
Transaktions-
kontrolle

Jedes an den WWW-Client gesendete Eingabeformular enthält (verdeckt) eine eindeutig identifizierbare Transaktionsnummer, die vom Server verwaltet wird. Beim Absenden einer Bestellung prüft die **Transaktionskontrolle**, ob bereits früher eine Bestellung mit derselben Nummer ausgeführt wurde. Ist dies der Fall, so erscheint die Fehlermeldung: „Aus Ihrem Formular wurde bereits eine Bestellung abgeschickt. Neue Formular-Auswahl machen“. Dadurch wird ausgeschlossen, daß mit einem bereits gesendeten Formular, auf das man mit BACK bzw. GO in der Historie zurückblättern kann, versehentlich oder mißbräuchlich eine weitere Bestellung erzeugt werden kann.

Nachdem eine Bestellung die Transaktionsprüfung durchlaufen hat und akzeptiert wurde, werden die Daten an das Fernleihsystem übergeben und die Bestellung wird am Client-Terminal bestätigt.

In der folgenden Abbildung wird das Client-Server-Prinzip bei einer Fernleihbestellung im WWW nochmals verdeutlicht. Dabei wird der Ablauf einer Bestellung von der Auswahl eines Formulars bis zur Quittierung der Bestellung anhand von 3 Schritten erklärt.

Abb. 10.5
Client-Server-Prinzip
bei der Fernleih-
bestellung im WWW



10.2

Fernleih-Bestelleingabe über OPAC-Terminals

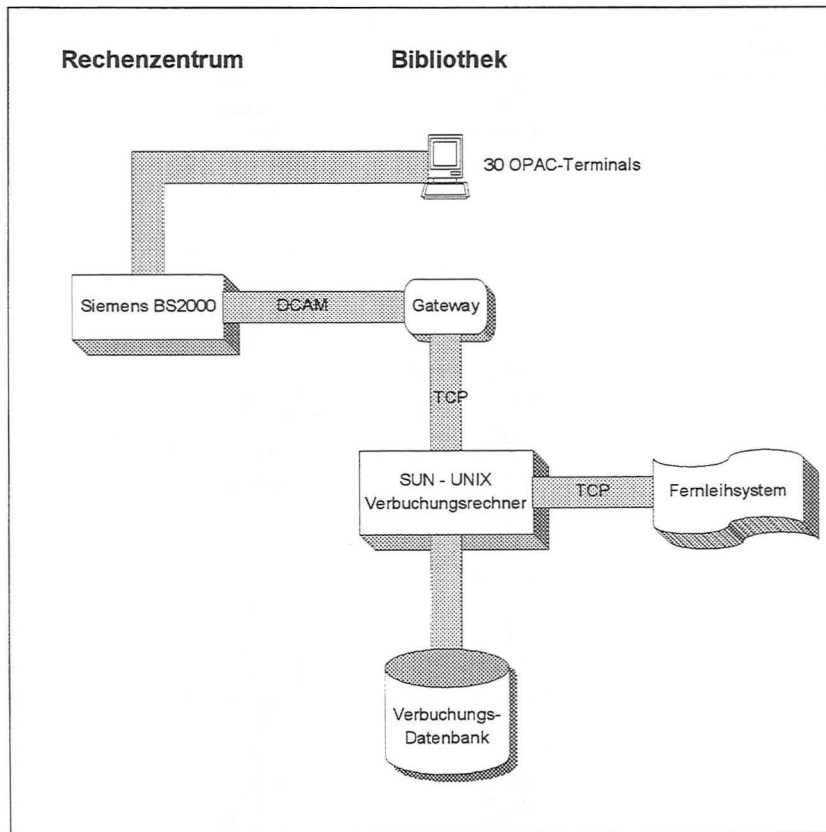
OPAC

Der **OPAC (Online Public Access Catalog)** ist ein öffentliches Recherche-System, das dem Leser ermöglicht über ein Terminal Suchanfragen an einen Zentralkatalog zu stellen. Beim Konstanzer KOALA⁸⁴-OPAC kann der Benutzer zusätzlich noch Briefe/Nachrichten an die Bibliothek schicken, Vormerkungen auf Bücher machen, sein Passwort ändern, eine Fernleihbestellung aufgeben usw.

⁸⁴ Konstanzer Ausleih- und Anfrage-System.

Daten, wie Vormerkungen, Paßwörter, Nachrichten usw. betreffen die Verbuchung. Fernleihbestellungen hingegen müssen zum Fernleihsystem übertragen werden. Sowohl die Daten für das Verbuchungssystem, als auch Fernleihbestellungen werden vom OPAC-Terminal zum Siemens Rechner geschickt. Dort wird das entsprechende Kommando erzeugt und zum SUN UNIX-Rechner gesendet. Die Verbindung läuft über ein Gateway, da an den Siemens-Rechner kein direkter TCP-Anschluß programmierbar ist. Das Gateway wird durch einen PC realisiert, der zwei Schnittstellenkarten beinhaltet: Eine DCAM⁸⁵-Schnittstellenkarte und eine TCP-Schnittstellenkarte. Der Verbuchungsrechner erkennt das ankommende Kommando und verzweigt entweder zum Fernleihsystem oder zum Verbuchungssystem.

Abb. 10.6
Verbindung zwischen
OPAC-Terminals und
Fernleihsystem



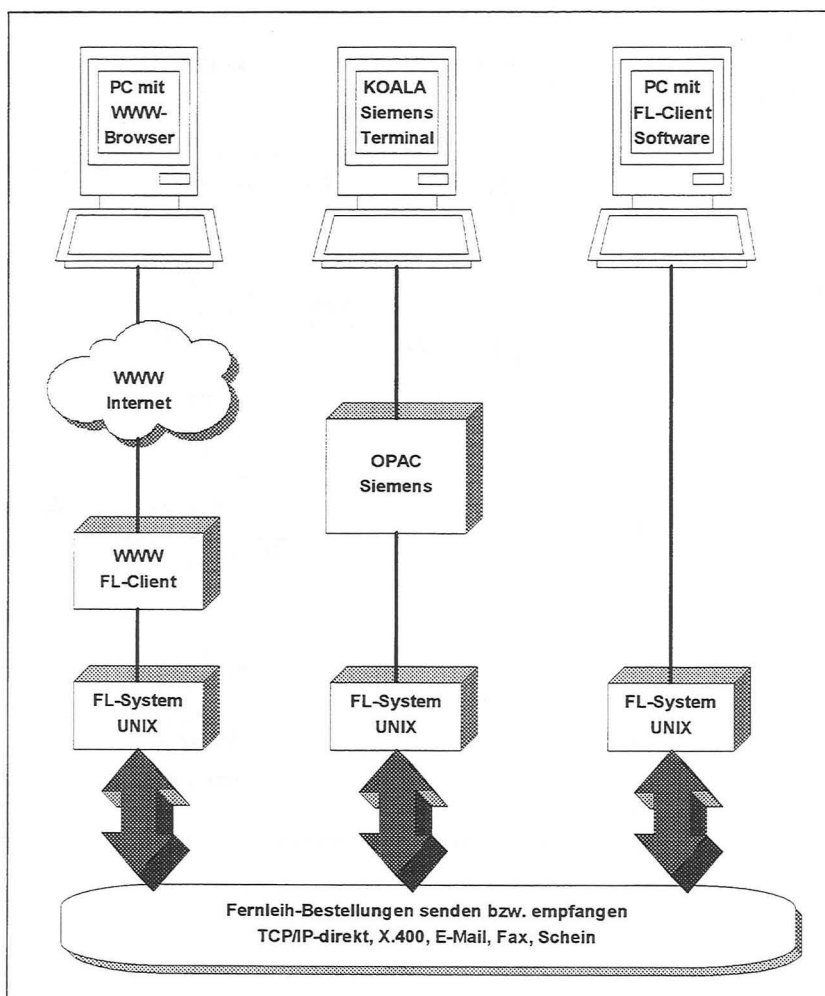
⁸⁵ Dialog Communication Access Method Protocol (Siemens-DFÜ-Protokoll).

10.3 Schnittstellen zum Fernleihsystem

Fernleih-Bestelleingaben können also gemäß Abschnitt 10.1 über einen WWW-Client oder gemäß Abschnitt 10.2 über ein KOALA-Siemens-Terminal gemacht werden. Für die Sachbearbeiter in der Fernleihabteilung steht noch eine weitere Möglichkeit der Eingabe von Fernleihbestellungen zur Verfügung; nämlich die PCs mit der Fernleih-Client Software.

In der folgenden Abbildung sind alle Schnittstellen zum Fernleihsystem in einer Grafik zusammengefasst:

Abb. 10.7
Schnittstellen zum
Fernleihsystem



10.4 Schnittstelle zwischen Fernleihsystem und Ausleihverbuchung

Funktionen	Im Verbuchungssystem (VB-System) sind wesentliche Funktionen enthalten, die vom Fernleihsystem (FL-System) ebenfalls benutzt werden müssen. Es ist nicht sinnvoll, diese Funktionen in beiden Systemen zu haben, da hierdurch ein doppelter Verwaltungs- und Pflegeaufwand entsteht.
Testbetrieb	Bei der Konzeption dieser Schnittstelle wurde davon ausgegangen, daß ein Testbetrieb des FL-Systems auch ohne VB-System möglich sein muß. Ein sinnvoller Produktionsbetrieb ist allerdings nur in Zusammenarbeit mit einem VB-System möglich. Folgende Leistungen des Verbuchungssystems werden vom Fernleihsystem genutzt:
Benutzer-Check	<ul style="list-style-type: none">• Prüfung eines Benutzers anhand von Benutzernummer und Paßwort. Im VB-System erfolgt auch die Prüfung bezüglich Fernleihberechtigung und Benutzersperrung.
Gebührensatz anlegen	<ul style="list-style-type: none">• Anlegen eines Gebührensatzes aufgrund einer getätigten Fernleihbestellung. (Die Gebührenverwaltung mit Erinnerungs- und Mahnwesen ist ein Teil des VB-Systems.)
Briefsatz anlegen	<ul style="list-style-type: none">• Anlegen von Briefsätzen zur Benachrichtigung der Benutzer. Benutzerbenachrichtigungen sind z.B. Abholbenachrichtigungen, Rückrufe, Quellnachweise usw. Der automatische Briefdruck ist eine Komponente im VB-System.
Bereitstellung anlegen	<ul style="list-style-type: none">• Generierung der Bereitstellung eines Buches aufgrund des Eingangs eines Objektes von der gebenden Bibliothek. Mit dem Anlegen eines speziellen Bereitstellungsatzes für die Fernleihe im VB-System ist es möglich, die Ausleihe einer Fernleihbestellung analog der lokalen Ausleihe zu verwalten. Die Kontrolle von Ausleihe, Rückgabe und Ausleihfristen erfolgt im VB-System. Erinnerungs- und Mahnbriefe werden vom VB-System erstellt. Für den Testbetrieb sind im FL-System die Funktionen „Benutzer-Ausleihe“ und „Benutzer-Rückgabe“ in einer vereinfachten Form enthalten.
Ausleihe veranlassen	<ul style="list-style-type: none">• Veranlassen der Ausleihe eines Objektes an die nehmende Partnerbibliothek. Mit dem Anlegen eines Ausleihsatzes für die Fernleihe im VB-System ist es möglich, Ausleihe, Rückgabe und Ausleihfristen zu kontrollieren.
FL-System als Client	In den oben genannten Fällen ist das VB-System „Server“ und das FL-System „Client“. Die Anfragen kommen vom Fernleihsystem und werden im Verbuchungssystem realisiert. Somit werden die Daten im VB-System aktualisiert.
FL-System als Server	Es existiert aber auch der Fall, daß das VB-System „Client“ ist und das FL-System „Server“. Damit im FL-System eine lückenlose Kontrolle von Fernleihvorgängen möglich ist, müssen die Ausleihen und Rückgaben von Fernleihobjekten vom VB-System an das FL-System gemeldet werden:
Ausleihe an Benutzer	<ul style="list-style-type: none">• Rückmeldung des Verbuchungssystems an das nehmende Fernleihsystem über die Ausleihe des Objektes an den Benutzer. Im FL-System wird daraufhin vom Zustand „Bereitstellung des Objektes für den Benutzer“ auf den Zustand „Ausgeliehen an Benutzer“ gewechselt.
Ausleihe an Partnerbibliothek	<ul style="list-style-type: none">• Rückmeldung an die gebende Fernleihe über die Ausleihe des Objektes an die Partnerbibliothek. Im FL-System wird vom Zustand „Objekt ist ausleihfähig“ auf den Zustand „Warten auf Rückgabe des Objektes“ aktualisiert.

Anbindung wichtiger Bibliotheksdienste

Rückgabe von
Partnerbibliothek

- Rückmeldung an die gebende Fernleihe über die Rückgabe des Objektes von der Partnerbibliothek. Im FL-System wird somit der Zustand „Vorgang beendet“ generiert.

Synchronisation

Dadurch, daß das FL-System und das VB-System sowohl als Client, als auch als Server agieren, findet ständig eine Synchronisation zwischen den Daten im Fernleihsystem und im Verbuchungssystem statt.

Im Anhang VIII sind die Kommandos zur Kommunikation zwischen FL-System und VB-System aufgeführt.

11

Bewertung des Fernleihsystems an der Universität Konstanz

Betrachtet man das elektronische Fernleihsystem der Universitätsbibliothek Konstanz kritisch, so kann man einige Vor- und Nachteile des Systems erkennen. In den folgenden beiden Abschnitten sind die Vor- und Nachteile für den Fernleih-Client und die Fernleih-Server gesondert aufgeführt.

11.1 Fernleih-Client

Vorteile:

- Durch die zeichenorientierte Oberflächengestaltung ist nur ein geringer Hauptspeicherbedarf notwendig. Der Fernleih-Client ist somit auf einem 286-PC lauffähig.
- Durch die TELNET-orientierten Kommandos wird ein leichtes Testen der Client-Server-Kommunikation ermöglicht.
- Der Aufbau der Dialogfunktionen erfolgte in Anlehnung an die realen organisatorischen Bearbeitungsschritte der Fernleihe. Somit ist der Ablauf des Dialogs sehr praxisbezogen und benutzerorientiert.

Nachteile:

- Die zeichenorientierte Oberfläche wirkt veraltet und wird bei der Anzeige mehrerer Menüpunkte schnell unübersichtlich.
- Es existiert keine Mausunterstützung.
- Es wurde noch keine Schnittstelle zur Recherche im Verbundkatalog realisiert, die eine Datenübernahme der besitzenden Bibliotheken in das Fernleihsystem ermöglicht.
- Das Client-Programm wurde zu wenig parametrisiert, so daß bei Änderungen von Feldgrößen ein erhöhter Arbeitsaufwand notwendig ist.

11.2 Fernleih-Server

Vorteile:

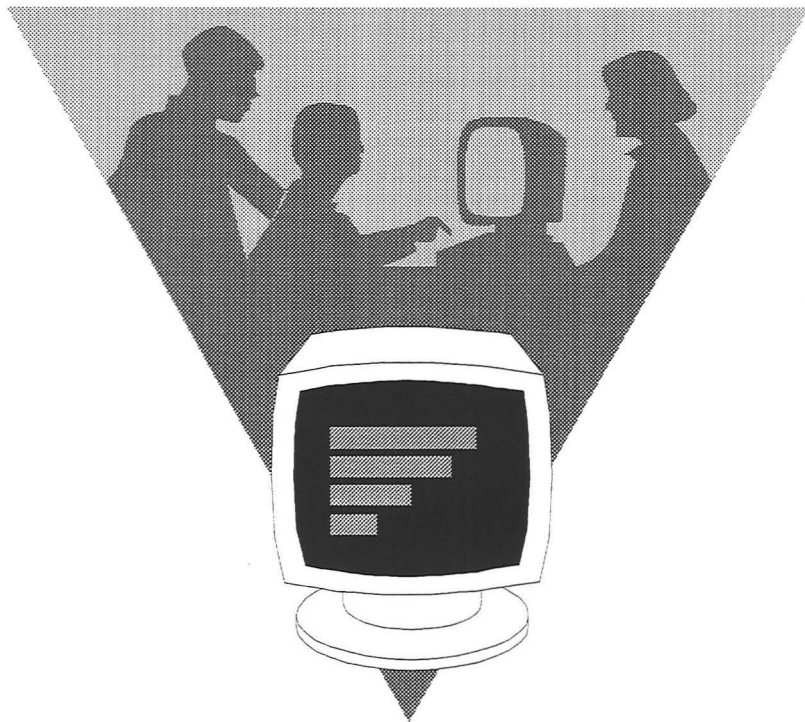
- Durch eine generierte SQL-Schnittstelle zur Datenbank ist ein leichter Wechsel der Datenbank möglich.
- Durch die realisierte Schnittstelle zum lokalen Verbuchungssystem ist die Nutzung der Funktionen des VB-Systems möglich.

Nachteile:

- Die Tabellen und Felder der Datenbank wurden nach den Bedürfnissen der Konstanzer Fernleihe statisch festgelegt und nicht genügend parametrisiert. Umstellungen auf andere Bibliotheken sind daher sehr schwierig.
- Die Anpassung des Fernleihsystems an ein fremdes Verbuchungssystem ist sehr schwierig, da es hierfür keine genormte Schnittstelle gibt.

Kapitel 4

Implementierung der ILL-Dienste



12

ILL-REQUEST

Verwendung des
Dienstes ILL-
REQUEST

Dieser Dienst erlaubt es einem Requester ein Objekt bei einem Responder zu bestellen. Wenn das Objekt nicht zur Verfügung gestellt werden kann, kann die gebende Bibliothek die Anfrage negativ beantworten.

Die Angaben, die vom Requester geliefert werden, beinhalten Informationen zur Identifizierung des Objektes, wie z.B. Autor, Titel, Erscheinungsjahr usw.

Sobald die nehmende Bibliothek den Dienst ILL-REQUEST verschickt hat und die gebende Bibliothek die Fernleihanfrage erhalten hat, existiert eine neue ILL-Transaktion. Jede ILL-Transaktion hat eine eindeutig identifizierende Transaktionsnummer, die bei der Bearbeitung dieses Vorgangs und bei der Nutzung weiterer ILL-Dienste stets angegeben werden muß.

12.1

Senden und Empfangen des Dienstes ILL-REQUEST

Der Dienst ILL-REQUEST wird von der nehmenden Bibliothek gesendet und von der gebenden Bibliothek empfangen. Das Senden und Empfangen einer Fernleihanfrage ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Der Dienst ILL-REQUEST kann von der nehmenden Bibliothek nur im Zustand 'leitweg_ok_nz' gesendet werden. Die gebende Bibliothek kann den Dienst ILL-REQUEST jederzeit empfangen, solange keine ILL-Transaktion mit derselben Transaktionsnummer existiert.

12.1.1

Senden des Dienstes ILL-REQUEST durch den Requester

Lokaler Zustand
leitweg_ok_nz

Der Requester hat die Bestelldaten der Fernleihanfrage erfaßt und das Bibliographieren vollständig abgeschlossen. Danach legt er den Leitweg der Bestellung fest. Der lokale Zustand 'leitweg_ok_nz' ist erreicht, wenn für die im Leitweg angegebene Zielbibliothek eine TCP/IP Adresse (Hostname, Portnummer) bzw. eine X.400 Adresse verfügbar ist. Ab diesem Zeitpunkt kann die Fernleihanfrage an die gebende Bibliothek versendet werden.

12.1.2

Empfangen des Dienstes ILL-REQUEST beim Responder

Lokaler Zustand
anfangs_gz

Der Responder kann jederzeit einen ILL-REQUEST empfangen. Die einzige Bedingung ist, daß noch keine andere ILL-Transaktion mit derselben Transaktionsnummer vorhanden ist. Nachdem die gebende Bibliothek die Fernleihanfrage erhalten hat, befindet sich die ILL-Transaktion im Anfangszustand 'anfangs_gz'. Dies ist der Grundzustand einer unbearbeiteten Anfrage nach Empfang des ILL-REQUEST. Ab diesem Zeitpunkt existiert eine ILL-Transaktion.

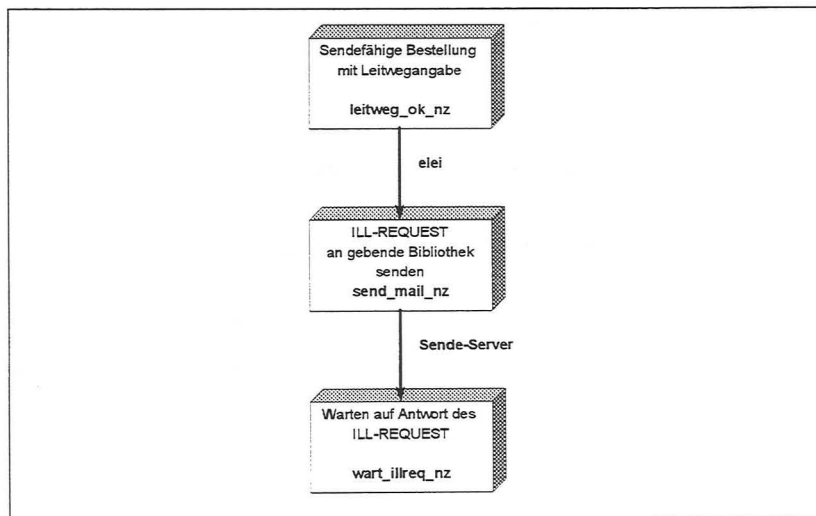
12.2 Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes ILL-REQUEST

Beim Senden des Dienstes ILL-REQUEST findet ein Zustandswechsel statt. Dabei wird der aktuelle Zustand beendet, d.h. der entsprechende Datensatz in der Fernleihdatenbank erhält ein Endedatum und eine Endezeit. Mit Abschluß des aktuellen Zustandes wird sofort der nachfolgende Zustand generiert, bei dem Anfangsdatum und -zeit gesetzt werden, Endedatum und -zeit bis zur Beendigung des Zustandes offen bleiben. Beim Empfangen des Dienstes ILL-REQUEST wird ein Anfangszustand für diese ILL-Transaktion in der Fernleihdatenbank generiert.

12.2.1 Zustandsübergänge beim Senden des Dienstes ILL-REQUEST durch den Requester

Die nehmende Bibliothek kann den Dienst ILL-REQUEST nur senden, wenn sich die Bestellung im Zustand 'leitweg_ok_nz' befindet. Durch das Kommando *eiei* wird der aktuelle Zustand 'leitweg_ok_nz' abgeschlossen und der Folgezustand 'send_mail_nz' mit Sekundärkennung 'send_request' in der Fernleihdatenbank generiert. Sobald der Sendeserver diesen sendefähigen Dienst verschickt hat, beendet er den generierten Satz, indem er ihn mit Endedatum und Endezeit versieht.

Abb. 12.1
Zustandsübergänge
beim Senden des
ILL-REQUEST
(Requester)

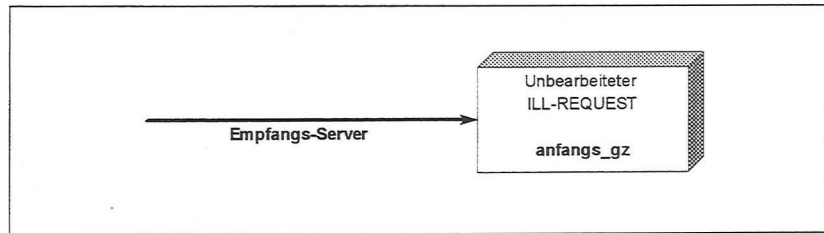


12.2.2 Zustandsübergänge beim Empfangen des Dienstes ILL-REQUEST beim Responder

Der Responder kann den Dienst ILL-REQUEST empfangen, wenn noch keine andere ILL-Transaktion mit derselben Transaktionsnummer existiert. Durch dem Empfangsserver wird der Anfangszustandsatz 'anfangs_gz' in der Fernleihdatenbank generiert.

ILL-REQUEST

Abb. 12.2
Zustandübergänge
beim Empfangen des
ILL-REQUEST
(Responder)



12.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind also für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Dabei spielt im lokalen Prozeß der Zustand der Partnerbibliothek keine Rolle.

Erst beim Senden und Empfangen von Diensten werden die Zustände der Partnerbibliothek wichtig. Daher existieren zusätzlich zu den lokalen Zuständen auch globale ILL-Transaktionszustände⁸⁶. Sie dienen zur Prüfung, ob ein Sender einen Dienst in seinem aktuellen globalen Zustand senden darf und ob ein Empfänger einen Dienst in seinem aktuellen globalen Zustand empfangen darf.

Der Dienst ILL-REQUEST kann vom Requester im globalen Zustand IDLE gesendet und vom Responder im globalen Zustand IDLE empfangen werden. Die globalen Folgezustände sind nach dem Senden der Zustand PENDING und nach dem Empfangen der Zustand IN-PROCESS.

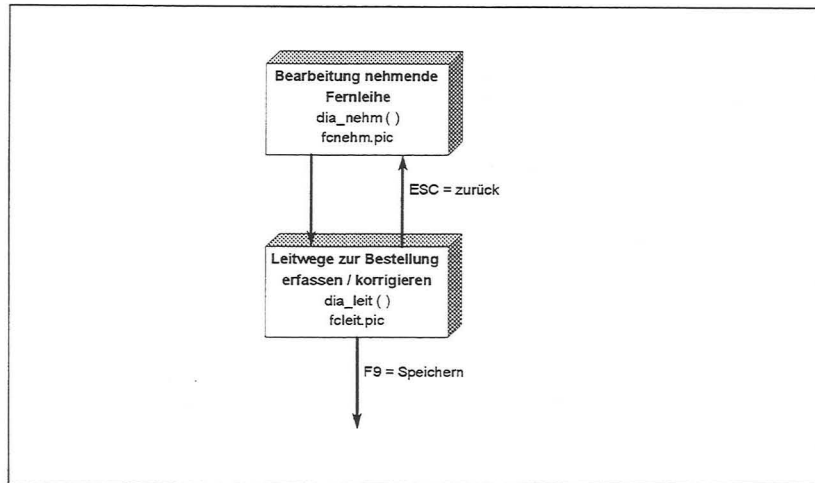
12.4 Menüsteuerung beim Fernleih-Client

12.4.1 Nehmende Fernleihe (Requester)

In der nehmenden Bearbeitung kann man zum Menü „Leitwege zur Bestellung erfassen/korrigieren“ gelangen, indem man im Hauptmenü den Punkt „Bearbeitung nehmende Fernleihe“ wählt. Es können drei Leitwege pro Bestellung angegeben werden, d.h. drei Zielbibliotheken, die in der Reihenfolge ihrer Nennung angefragt werden. Nachdem die Leitwege erfaßt und gespeichert sind kann die Fernleihanfrage sofort oder zu einem späteren Zeitpunkt versendet werden.

⁸⁶ Siehe Abschnitt 3.4.4.

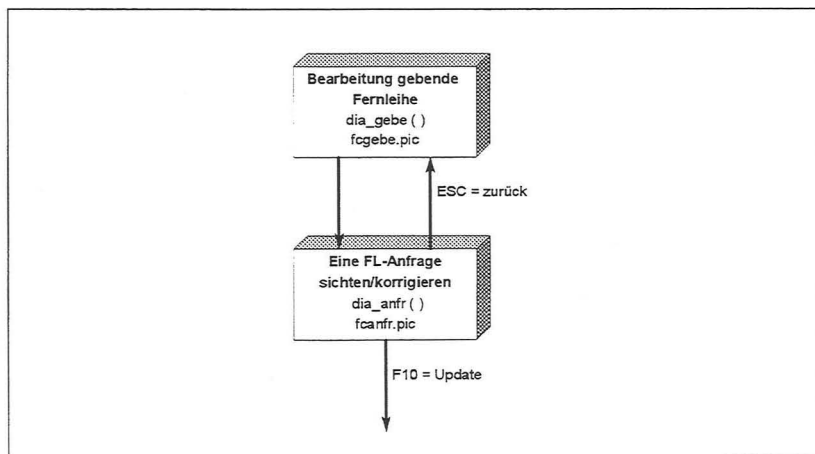
Abb. 12.3
Menü ILL-REQUEST
(Requester)



12.4.2 Gebende Fernleihe (Responder)

In der gebenden Bearbeitung gelangt man zur Bearbeitung von eingetroffenen neuen Fernleihanfragen, indem man im Hauptmenü die Verzweigung „Bearbeitung gebende Fernleihe“ und dann das Untermenü „Eine FL-Anfrage sichten/korrigieren“ wählt. Hier können die Daten der unbearbeiteten Fernleihanfrage gesichtet und gegebenenfalls korrigiert werden.

Abb. 12.4
Menü
Anfragen von der
Partnerbibliothek
(Responder)



13 SHIPPED

Verwendung des ILL-Dienstes SHIPPED

Mit diesem ILL-Dienst teilt der Responder dem Requester mit, daß er das gewünschte Objekt an den Requester versendet hat.

Der ILL-Dienst SHIPPED ist optional, d.h. seine Verwendung kann auch unterlassen werden. Im Konstanzer Fernleihsystem wird er jedoch standardmäßig benutzt, sobald die gebende Bibliothek die Ausleihe an die nehmende Bibliothek ausgeführt hat.

Im Fernleihsystem der Uni Konstanz wird dieser Dienst nach der Ausführung des Bearbeitungsschrittes „Ausleihe/Auslieferung“ im Menü der gebenden Bearbeitung automatisch an den Requester verschickt. Deshalb entfällt bei der Beschreibung dieses Dienstes der Abschnitt „Menüsteuerung im Fernleih-Client“.

13.1 Senden und Empfangen des ILL-Dienstes SHIPPED

Der ILL-Dienst SHIPPED wird von der gebenden Bibliothek gesendet und von der nehmenden Bibliothek empfangen. Das Senden und Empfangen einer Versandnachricht ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Der ILL-Dienst SHIPPED kann von der gebenden Bibliothek nur im Zustand ‘ausleih_gz’ gesendet werden. Die nehmende Bibliothek kann den ILL-Dienst SHIPPED nur im aktuellen Zustand ‘wart_return_gz’ empfangen.

13.1.1 Senden des ILL-Dienstes SHIPPED durch den Responder

Lokaler Zustand
ausleih_gz

Nachdem der Responder eine Fernleihanfrage vom Requester erhalten hat, prüft er die Anfrage auf ihre Erfüllbarkeit. Ist das gewünschte Objekt in der gebenden Bibliothek verfügbar, so wird es als ausleihfähig bezeichnet und die ILL-Transaktion erreicht den lokalen Zustand ‘ausleih_gz’. Der Holdienst wird nun beauftragt das Objekt zu holen. Endet die Suche nach dem gewünschten Objekt in der gebenden Bibliothek erfolgreich, so kann es nun an die nehmende Bibliothek verschickt werden. Durch dem ILL-Dienst SHIPPED wird dem Requester zugleich mitgeteilt, daß das Objekt an ihn abgesendet wurde. Wird das ausleihfähige Objekt in der gebenden Bibliothek nicht gefunden, so erhält der Requester den Dienst ILL-ANSWER mit der Antwort UNFILLED.

13.1.2 Empfangen des ILL-Dienstes SHIPPED beim Requester

Lokaler Zustand
wart_illreq_nz

Der Requester hat an den Responder eine Fernleihanfrage gestellt und wartet nun auf die Antwort zu seiner Bestellung. Die ILL-Transaktion befindet sich nun im Zustand ‘wart_illreq_nz’. In diesem Zustand kann die nehmende Bibliothek den ILL-Dienst SHIPPED empfangen und weiß dadurch, daß sich das gewünschte Objekt auf dem Transportweg befindet.

13.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes SHIPPED

Beim Senden des ILL-Dienstes SHIPPED findet ein Zustandswechsel statt. Dabei wird der aktuelle Zustand beendet, d.h. der entsprechende Datensatz in der Fernleihdatenbank erhält ein Endedatum und eine Endezeit. Mit Abschluß des aktuellen Zustandes wird sofort der nachfolgende Zustand generiert, bei dem Anfangsdatum und -zeit gesetzt werden, Endedatum und -zeit bis zur Beendigung des Zustandes offen bleiben.

Beim Empfangen des ILL-Dienstes SHIPPED findet kein Zustandswechsel und keine Generierung eines neuen Zustandssatzes in der Fernleihdatenbank statt. Der Empfang des Dienstes wird lediglich im aktuellen Zustandsdatensatz vermerkt.

13.2.1 Zustandsübergänge beim Senden des ILL-Dienstes SHIPPED durch den Responder

Die gebende Bibliothek kann den ILL-Dienst SHIPPED nur senden, wenn sich die ILL-Transaktion im Zustand 'ausleih_gz' befindet. Man unterscheidet nun, ob die nehmende Bibliothek eine Monographie oder einen Zeitschriften-Aufsatz bestellt hat:

Monographie

- Eine Monographie wird ausgeliehen und muß wieder zurückgegeben werden. In diesem Fall wird durch das Kommando *avba* der Zustand 'ausleih_gz' abgeschlossen und ein neuer Datensatz in der Fernleihdatenbank für den ILL-Dienst SHIPPED generiert. Dieser Folgezustand heißt 'send_mail_gz' mit Sekundärkennung 'send_shipped_a'. Die Sekundärkennung gibt an, daß es sich um den Versand einer Monographie handelt, die wieder zurückgegeben werden muß.

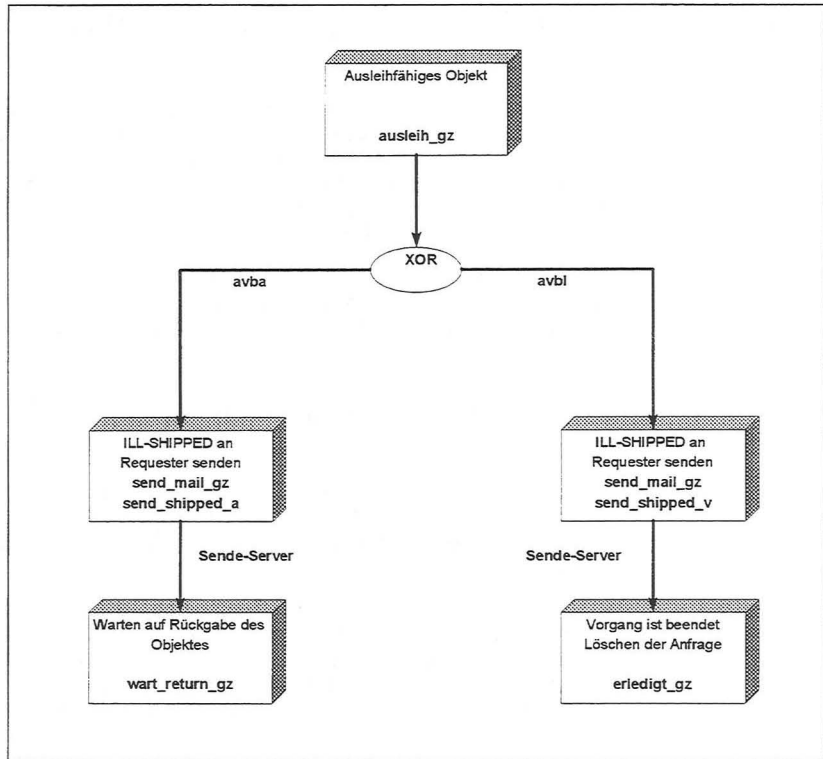
Nachdem der ILL-Dienst SHIPPED versendet wurde wird der Zustand 'send_mail_gz' vom Sendeserver abgeschlossen und der Folgezustand 'wart_return_gz' generiert. Der Responder verbleibt nun in diesem lokalen Zustand bis das Objekt von der nehmenden Bibliothek wieder zurückgeschickt wird.

Zeitschriften-Aufsatz

- Ein Zeitschriften-Aufsatz kann nicht ausgeliehen werden. Er wird kopiert und die Kopien verbleiben gegen Entgelt bei der nehmenden Bibliothek bzw. dem Leser. In diesem Fall wird durch das Kommando *avbl* der Zustand 'ausleih_gz' abgeschlossen und ein neuer Datensatz in der Fernleihdatenbank für den ILL-Dienst SHIPPED generiert. Dieser Folgezustand heißt 'send_mail_gz' mit Sekundärkennung 'send_shipped_v'. Die Sekundärkennung gibt an, daß es sich um den Versand von Kopien handelt, die nicht wieder zurückgegeben werden müssen.

Nachdem der ILL-Dienst SHIPPED versendet wurde wird der Zustand 'send_mail_gz' vom Sendeserver abgeschlossen und der Folgezustand 'erledigt_gz' generiert. Da die Kopien beim Requester verbleiben, ist die ILL-Transaktion auf der gebenden Seite somit abgeschlossen.

Abb. 13.1
Zustandsübergänge
beim Senden des
Dienstes SHIPPED
(Responder)



13.2.2

Zustandsübergänge beim Empfangen des ILL-Dienstes SHIPPED beim Requester

Durch den Empfang des ILL-Dienstes SHIPPED bekommt die ILL-Transaktion lediglich eine Markierung, daß der ILL-Dienst SHIPPED vom Responder gesendet wurde, d.h. daß das ausgeliehene Objekt bereits auf dem Transportweg zum Requester ist. Diese Markierung wird an dem aktuellen Zustandssatz 'wart illreq_nz' der ILL-Transaktion vermerkt und in der Fernleihdatenbank aktualisiert. Es wird folglich kein Zustandsübergang oder eine Zustandsgenerierung durch den Empfang des Dienstes durchgeführt. Somit entfällt für den Empfang des ILL-Dienstes SHIPPED die Abbildung von Zustandsübergängen.

13.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind also für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Dabei spielt im lokalen Prozeß der Zustand der Partnerbibliothek keine Rolle.

Erst beim Senden und Empfangen von ILL-Diensten werden die Zustände der Partnerbibliothek wichtig. Daher existieren zusätzlich zu den lokalen Zuständen auch globale ILL-Transaktionszustände⁸⁷. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen Zustand senden darf und ob ein Empfänger einen ILL-Dienst in seinem aktuellen globalen Zustand empfangen darf.

Die folgende Tabelle zeigt in welchen globalen Zuständen der ILL-Dienst SHIPPED von der gebenden Bibliothek gesendet und von der nehmenden Bibliothek empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 13.2
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
SHIPPED empfangen	SHIPPED senden
	IN-PROCESS
PENDING	
CANCEL-PENDING	
SHIPPED	SHIPPED
RECEIVED	
RENEW/PENDING	
RENEW/OVERDUE	
OVERDUE	
NOT-RECEIVED/OVERDUE	
RECALL	
RETURNED	
LOST	
Folgezustand: SHIPPED	Folgezustand: SHIPPED

⁸⁷ Siehe Abschnitt 3.4.4.

14

ILL-ANSWER

Verwendung von
ILL-ANSWER

Dieser Dienst erlaubt einem Responder, auf einen empfangenen Fernleihwunsch zu antworten. Die folgenden Antworten auf einen Fernleihwunsch sind möglich:

- **CONDITIONAL**: Das gewünschte Objekt ist verfügbar, der Fernleihwunsch kann aber nur zufriedengestellt werden, wenn der Requester bestimmten Ausleihkonditionen, die in der Antwort angegeben werden, zustimmt.
- **RETRY**: Das gewünschte Objekt ist derzeit nicht verfügbar, aber ein Datum, zu dem das Objekt voraussichtlich verfügbar sein kann, wird angegeben.
- **UNFILLED**: Das gewünschte Objekt ist nicht verfügbar.
- **LOCATIONS-PROVIDED**: Besitzende Bibliotheken werden mitgeteilt.
- **WILL-SUPPLY**: Das gewünschte Objekt ist verfügbar und wird sobald wie möglich verschickt. Ein ungefähres Datum, wann das Objekt geliefert werden kann, wird optional spezifiziert.
- **HOLD-PLACED**: Das gewünschte Objekt ist nicht sofort verfügbar, wird aber für den Requester gemäß seinen Anweisungen vorgemerkt.
- **ESTIMATE**: Die geschätzten Kosten für die Lieferung des gewünschten Objektes werden als Antwort auf einen Estimate-Request bereitgestellt.

Bei der Implementierung des Fernleihsystems der Universitätsbibliothek Konstanz wurden die ILL-Standard-Antworten **CONDITIONAL**, **RETRY**, **WILL-SUPPLY**, **LOCATIONS-PROVIDED**, **HOLD-PLACED** und **ESTIMATE** noch nicht realisiert. Im Moment ist nur die Sendung einer negativen Antwort auf eine Bestellanfrage möglich, also die Sendung des Dienstes **ILL-ANSWER-UNFILLED**.

14.1

Senden und Empfangen des ILL-Dienstes ILL-ANSWER

Eine ILL-ANSWER kann nur vom Responder gesendet werden, nachdem er vom Requester eine Fernleihanfrage erhalten und diese geprüft hat.

Der Dienst kann nur vom Requester empfangen werden, wenn er eine Fernleihanfrage an den Responder gestellt hat und diese noch nicht beantwortet wurde.

Folgende lokale Zustände eines Fernleihvorgangs können beim Versenden oder Empfangen des Dienstes ILL-ANSWER bestehen:

Abb. 14.1
Lokale Zustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
ILL-ANSWER senden	ILL-ANSWER empfangen
anfangs_gz	wart_illreq_nz
unerfuellbar_gz	
verfuegbar_gz	
ausleih_gz	

14.1.1 Senden des ILL-Dienstes ILL-ANSWER durch den Responder

Lokaler Zustand
anfangs_gz

Der Responder kann die ILL-ANSWER an die nehmende Bibliothek verschicken, wenn bereits während der Bearbeitung der Bestellung festgestellt wird, daß die Anfrage nicht erfüllt werden kann, weil z.B. das Leihobjekt nicht vorhanden oder stark beschädigt ist. Während der Bearbeitung der Anfrage ist der lokale Zustand der ILL-Transaktion auf der gebenden Seite noch 'anfangs_gz'.

Lokaler Zustand
unerfuellbar_gz

In jedem Fall muß der lokale Zustand 'anfangs_gz', wenn festgestellt wird, daß die Fernleihanfrage nicht zu erfüllen ist, auf 'unerfuellbar_gz' gesetzt werden. Auch jetzt ist es noch möglich, den Dienst ILL-ANSWER zu verschicken. In beiden Fällen wird die Antwort negativ sein.

Lokaler Zustand
verfuegbar_gz

Ist das Leihobjekt für die Fernleihe verfügbar, besteht für den entsprechenden Leihvorgang der lokale Zustand 'verfuegbar_gz'. In diesem Fall wird die Antwort für den Requester wahrscheinlich positiv ausfallen.

Lokaler Zustand
ausleih_gz

Der Requester kann ebenfalls eine Antwort auf seine Anfrage erhalten, wenn beim Responder das Leihobjekt bereits durch den Holddienst zum Versand bereitgestellt wurde und damit der lokale Zustand 'ausleih_gz' gesetzt wurde.

Im Konstanzer Fernleihsystem werden nur die negativen Antworten unterstützt. Im Falle einer Absage wird an den Requester also die Mitteilung gesendet, daß das Objekt nicht verfügbar ist. Die ILL-Standard-Antworten CONDITIONAL, RETRY, LOCATIONS-PROVIDED, WILL-SUPPLY, HOLD-PLACED und ESTIMATE werden nicht erteilt.

14.1.2 Empfang des ILL-Dienstes ILL-ANSWER beim Requester

Lokaler Zustand
wart_illreq_nz

Der Requester kann den Dienst ILL-ANSWER nur empfangen, wenn sich die entsprechende ILL-Transaktion im lokalen Zustand 'wart_illreq_nz' befindet.

Eine Fernleihanfrage wurde durch die nehmende Bibliothek gestellt. Nun wird auf die Lieferung des Objekts oder eine Antwort des Responders gewartet.

Auch beim Empfang wird nur der Erhalt einer negativen Antwort unterstützt. Die Fälle CONDITIONAL, RETRY, LOCATIONS-PROVIDED, WILL-SUPPLY, HOLD-PLACED und ESTIMATE werden nicht abgedeckt, da sie noch nicht implementiert wurden.

14.2 Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes ILL-ANSWER

Beim Senden der ILL-ANSWER wird die ILL-Transaktion auf der Responderseite abgeschlossen, da in diesem Fall beim Fernleihsystem der Universitätsbibliothek Konstanz die negative Antwort gesendet wird. Der Responder kann also die Anfrage des Requesters nicht erfüllen.

Bei Empfang einer negativen Antwort zu einer Fernleihbestellung wird der Zustandsdatensatz des zugehörigen Leitweges mit einer negativen Kennung beendet. Wenn ein weiterer Leitweg vorhanden ist, wird automatisch dieser nächste Leitweg gesetzt. Es wird also der Request für die Bestellung bei der nächsten Partnerbibliothek aktiviert.

Wurden alle Leitwege negativ abgearbeitet, erfolgt eine konventionelle Bestellung oder der Benutzer wird benachrichtigt.

14.2.1 Zustandsübergänge beim Senden des Dienstes ILL-ANSWER durch den Responder

Wie bereits erwähnt, kann in der Implementierung des Konstanzer Fernleihsystems die gebende Seite nur eine negative Antwort senden. In diesem Fall befindet sich die betreffende Fernleihbestellung entweder im Zustand 'anfangs_gz', 'verfuegbar_gz' oder 'ausleih_gz'.

Der Zustand 'verfuegbar_gz' kann auch bestehen, wenn das gewünschte Objekt nicht verliehen werden darf oder kann. Und der Zustand 'ausleih_gz' kann zwar gesetzt worden sein, dennoch besteht die Möglichkeit, daß die gebende Bibliothek das Objekt dringend selbst benötigt und damit eine negative Antwort an den Requester erforderlich ist.

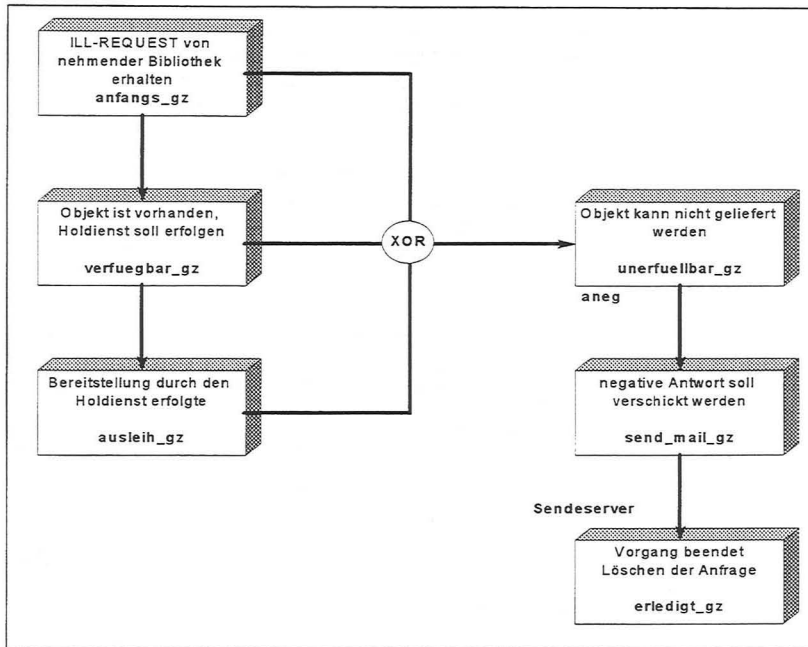
In jedem der drei Fälle wird zunächst durch den Aufruf des Kommandos *aeng* der Zustand 'unerfuellbar_gz' gesetzt, jedoch wird dabei kein neuer Datensatz generiert, sondern der bis dahin bestehende lokale Zustand wird aktualisiert.

Wenn der Zustand 'unerfuellbar_gz' bereits besteht, erfolgt keine Zustandsänderung.

Anschließend wird ebenfalls durch das Kommando *aneg* ein Satz für den Sendeserver generiert, der in der Sekundärkennung beinhaltet, daß es sich um eine negative Antwort handelt. Der lokale Zustand 'unerfuellbar_gz' wird abgeschlossen, indem Endedatum und -zeit gesetzt werden.

Nach dem Versenden wird durch den Sendeserver der neue lokale Zustand 'erledigt_gz' generiert. Damit ist die Vorgangsbearbeitung auf der Responderseite abgeschlossen.

Abb. 14.2
Zustandswechsel
beim Senden der
ILL-ANSWER

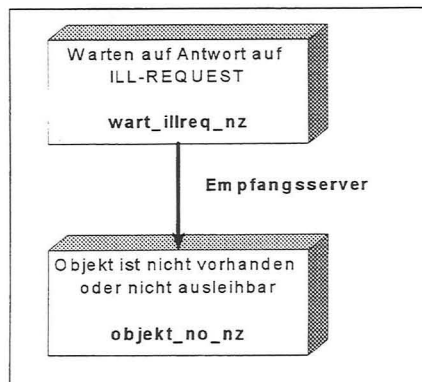


14.2.2 Zustandsübergänge beim Empfangen des Dienstes ILL-ANSWER beim Requester

Bei Empfang einer negativen Antwort wird der lokale Zustand der Fernleihbestellung 'wart_illreq_nz' vom Empfangsserver abgeschlossen und es erfolgt ein Zustandswechsel auf 'objekt_no_nz'. Damit ist die Bearbeitung der entsprechenden Fernleihbestellung jedoch noch nicht beendet.

Die weitere Vorgehensweise wurde bereits erläutert.

Abb. 14.3
Zustandswechsel bei
Empfang einer negativen
ILL-ANSWER



14.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Der Zustand der ILL-Transaktion bei der Partnerbibliothek ist jedoch für den Wechsel zwischen lokalen Zuständen innerhalb eines Systems bedeutungslos.

Erst beim Senden und Empfangen von ILL-Diensten werden die globalen ILL-Transaktionszustände der Partnerbibliothek wichtig. Die Übersicht stellt dar, in welchen globalen ILL-Transaktionszuständen der Dienst ILL-ANSWER ausgeführt und empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist.

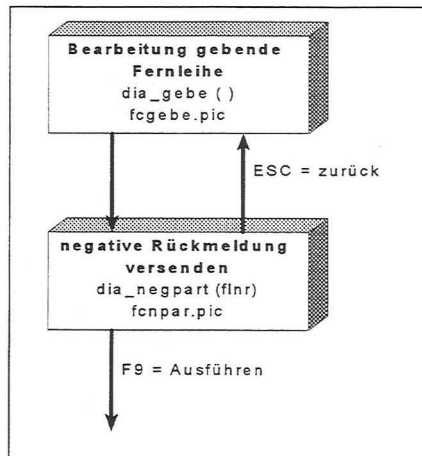
Abb. 14.4
Globale ILL-
Transaktions-
zustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
ILL-ANSWER senden	ILL-ANSWER empfangen
IN-PROCESS	PENDING
	CANCEL-PENDING
NOT-SUPPLIED	NOT-SUPPLIED
<u>FOLGEZUSTAND:</u> (bei ANSWER-UNFILLED) NOT-SUPPLIED	<u>FOLGEZUSTAND:</u> (bei ANSWER-UNFILLED) NOT-SUPPLIED

14.4 Menüsteuerung beim Fernleih-Client

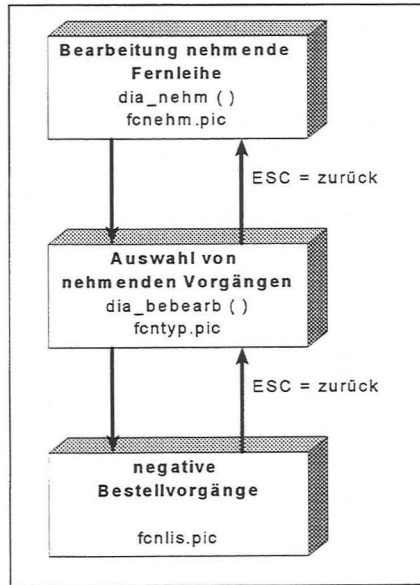
14.4.1 Gebende Fernleihe

Abb. 14.5
Menü ILL-ANSWER
(Responder)



14.4.2 Nehmende Fernleihe

Abb. 14.6
Menü ILL-ANSWER
(Requester)



15

CANCEL

Verwendung von
CANCEL

Dieser Dienst erlaubt einem Requester, die Stornierung einer ILL-Transaktion einzuleiten. Vom Responder wird erwartet, daß er die Stornierung annimmt oder ablehnt. In jedem Fall muß eine Fernleihanfrage durch den Requester erfolgt sein. Mit der Anfrage auf Stornierung wird die weitere Bearbeitung des Fernleihvorgangs gestoppt, bis die Antwort des Responders eintrifft.

15.1

Senden und Empfangen des ILL-Dienstes CANCEL

Ein CANCEL kann nur vom Initiator der Fernleihanfrage also dem Requester gesendet werden. Ebenso kann er nur vom Responder empfangen werden. Sinnvoll ist die Anfrage auf Stornierung jedoch nur, wenn die Anfrage von der nehmenden Bibliothek zwar erfolgte, die Bearbeitung auf der gebenden Seite aber noch nicht so weit fortgeschritten ist, daß das Leihobjekt bereits auf dem Weg oder sogar schon beim Requester eingetroffen ist.

Die folgende Übersicht zeigt, in welchen lokalen Zuständen der ILL-Transaktion der ILL-Dienst von der nehmenden Bibliothek gesendet und in welchen lokalen Zuständen der Vorgangsbearbeitung die gebende Bibliothek den CANCEL ohne Probleme annehmen kann:

Abb. 15.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
CANCEL senden	CANCEL empfangen
wart_illreq_nz	anfangs_gz
	unerfuellbar_gz
	verfuegbar_gz
	ausleih_gz

15.1.1

Senden des ILL-Dienstes CANCEL durch den Requester

Lokaler Zustand
wart_illreq_nz

Der Requester kann einen CANCEL senden, wenn sich die zu stornierende ILL-Transaktion im lokalen Zustand 'wart_illreq_nz' befindet:

Die Fernleihbestellung wurde ordnungsgemäß erfaßt und verschickt. Die nehmende Bibliothek wartet nun auf die Antwort der gebenden Bibliothek oder das bestellte Leihobjekt. Für den Benutzer hat der Erhalt des Objekts keine Bedeutung mehr und er meldet dies der nehmenden Bibliothek, die daraufhin den Vorgang stornieren kann.

15.1.2 Empfang des ILL-Dienstes CANCEL beim Responder

Lokaler Zustand
anfangs_gz

Ein CANCEL kann bei der gebenden Bibliothek empfangen werden, wenn der lokale Zustand **'anfangs_gz'** der ILL-Transaktion gesetzt wurde.

Eine Fernleihanfrage der nehmenden Bibliothek wurde empfangen. Sie wurde noch nicht bearbeitet, d.h. die Signatur wurde noch nicht geprüft, der Ausleihstatus des gewünschten Objekts steht noch nicht fest.

Wird in diesem Stadium der ILL-Transaktion ein Wunsch auf Stornierung empfangen, dann muß die Antwort auf die Stornierung an den Requester gesendet werden. Der Leihvorgang kann beim Responder sofort beendet werden, je nachdem ob die Stornierung akzeptiert wird oder nicht.

Lokaler Zustand
unerfuellbar_gz

Hat die gebende Bibliothek festgestellt, daß sie die Fernleihanfrage des Requesters nicht erfüllen kann, hat diesen aber noch nicht darüber informiert, kann sie ebenfalls einen CANCEL empfangen. Der lokale Zustand der Fernleihbestellung ist in diesem Moment **'unerfuellbar_gz'**.

Auch in diesem lokalen Zustand muß die Anfrage auf Stornierung des Vorgangs beantwortet werden. Sie wird in jedem Fall positiv ausfallen und der Vorgang kann sofort abgeschlossen werden.

Lokaler Zustand
verfuegbar_gz

Erhält die gebende Bibliothek im lokalen Zustand **'verfuegbar_gz'** einen Wunsch auf Stornierung, dann wurde bereits festgestellt, daß das Leihobjekt verliehen werden kann und es soll vom Holddienst bereitgestellt werden.

In diesem Fall kann die Bearbeitung durch den Holddienst abgebrochen und der Vorgang beendet werden, nachdem die gebende Bibliothek die Antwort auf den Stornierwunsch (CANCEL-REPLY) an die nehmende verschickt hat.

Lokaler Zustand
ausleih_gz

Hat der Holddienst das Objekt bereitgestellt, befindet sich der Fernleihvorgang bereits im Lokalzustand **'ausleih_gz'**. Damit steht das Leihobjekt unmittelbar zum Versand bereit.

Trifft nun ein CANCEL von der nehmenden Bibliothek ein, ist es durchaus möglich, daß auf Grund der organisatorischen Abläufe innerhalb der gebenden Bibliothek eine negative Antwort auf den CANCEL (CANCEL-REPLY) an den Requester verschickt werden muß. Eine Antwort sollte aber in jedem Fall erfolgen. Ist sie negativ, dann läuft die Bearbeitung des entsprechenden Vorgangs normal weiter. D.h. das Leihobjekt wird verschickt und der Folgezustand **'wart_return_gz'** wird gesetzt.

Wird der CANCEL akzeptiert, kann die Vorgangsbearbeitung beendet werden.

15.2 Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes CANCEL

Beim Senden des CANCEL wird ein neuer Zustand generiert und auch auf der Responder Seite wird ein neuer paralleler Zustand beim Empfang einer Stornieranfrage generiert. Beim neu generierten Zustand werden ein Anfangsdatum und -zeit gesetzt. Das Endedatum und die Endezeit werden erst bei Sendung des CANCEL bzw. dem Empfang der Antwort auf den CANCEL (CANCEL-REPLY) eingetragen. Mit diesem Setzen der Endedaten werden die Zustände abgeschlossen.

15.2.1 Zustandsübergänge beim Senden des Dienstes CANCEL durch den Requester

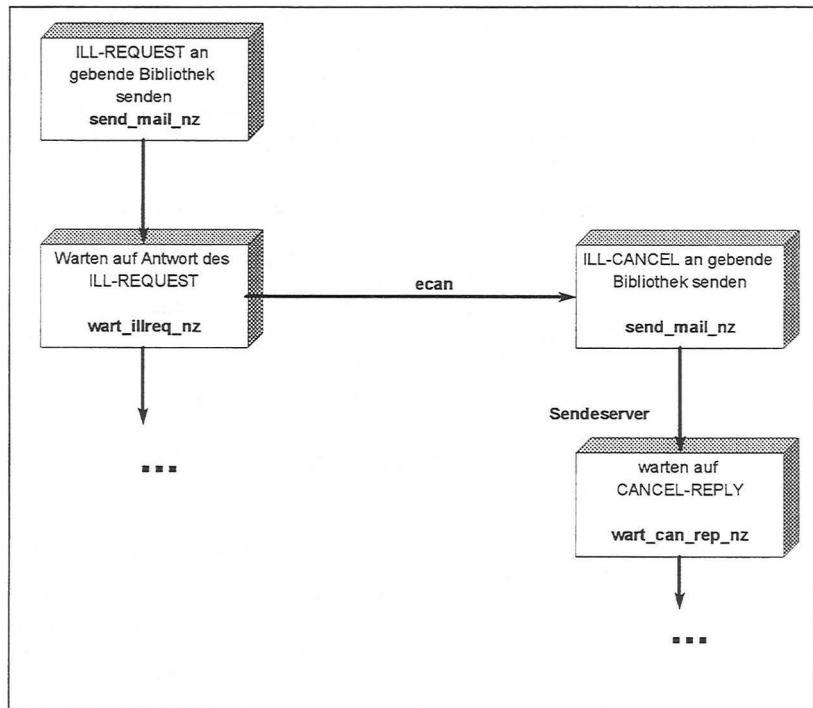
Wie bereits erwähnt, kann die nehmende Bibliothek nur zu einem Vorgang im lokalen Zustand 'wart_illreq_nz' den Dienst für die Stornierung senden. Wünscht ein Fernleih-Client das Versenden eines CANCEL, so übergibt er das Kommando *ecan* an den Dialogserver, der prüft, ob dieser Zustand vorliegt.

Weiterhin erfolgt die Prüfung, ob bereits ein CANCEL verschickt wurde. Dann kann kein neuer CANCEL zum selben Vorgang erfasst werden. Steht die Antwort auf den Stornierungswunsch durch die gebende Bibliothek schon lange aus, kann mit dem Dienst MESSAGE nachgefragt werden, warum die Verzögerung eingetreten ist.

Wenn der CANCEL ordnungsgemäß erfasst worden ist, wird durch das Programm *feecan.c* ein neuer Zustand 'send_mail_nz' generiert. An der Sekundärerkennung 'cancel' erkennt der Sendeserver, daß der Dienst CANCEL zu verschicken ist. Der Sendeserver schließt den Zustand 'send_mail_nz' ab, indem er Endezeit und -datum einträgt. Anschließend generiert er den neuen Zustand 'wart_can_rep_nz'. Dieser Zustand zeigt an, daß eine Stornieranfrage versendet wurde und nun auf die Antwort durch den Responder gewartet wird.

Der aktuelle Zustand 'wart_illreq_nz' bleibt bestehen.

Abb. 15.2
Zustandsgenerierung
beim Senden
des CANCEL
(Requester)



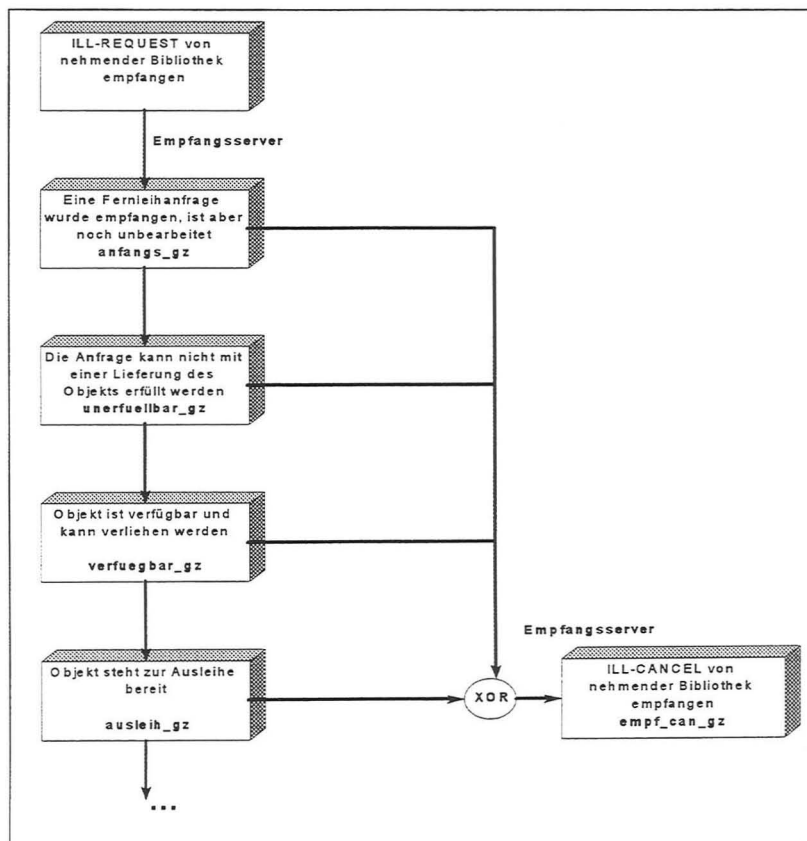
15.2.2 Zustandsübergänge bei Empfang des Dienstes CANCEL beim Responder

Der Responder kann den Dienst CANCEL nur empfangen, wenn sich die betreffende ILL-Transaktion in einem der folgenden lokalen Zustände auf der gebenden Seite befindet:

- 'anfangs_gz',
- 'unerfuellbar_gz',
- 'verfuegbar_gz',
- 'ausleih_gz'.

Der Empfangsserver prüft, ob beim Eintreffen des CANCEL einer dieser Zustände vorliegt. Daraufhin wird der parallele Zustand 'empf_can_gz' generiert. Die weitere Handhabung unterliegt der Fernleihabteilung der Bibliothek. In der Regel wird pro Arbeitstag eine Liste mit den empfangenen Stornierungsanfragen ausgedruckt werden. Diese müssen anschließend beantwortet werden, indem für jede Anfrage eine Antwort in Form des ILL-Dienstes CANCEL-REPLY erfaßt wird.

Abb. 15.3
Zustandsgenerierung
bei Empfang eines
CANCEL
(Responder)



15.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände einer ILL-Transaktion sind nur für die lokalen Zustandswechsel innerhalb eines Systems von Bedeutung. Die globalen ILL-Transaktionszustände dienen der Prüfung, ob die Durchführung eines ILL-Dienstes in diesem Stadium der Transaktion überhaupt möglich und sinnvoll ist.

Die folgende Tabelle verdeutlicht, in welchen globalen Zuständen der Dienst CANCEL von der nehmenden Bibliothek gesendet und von der gebenden Bibliothek empfangen werden kann.

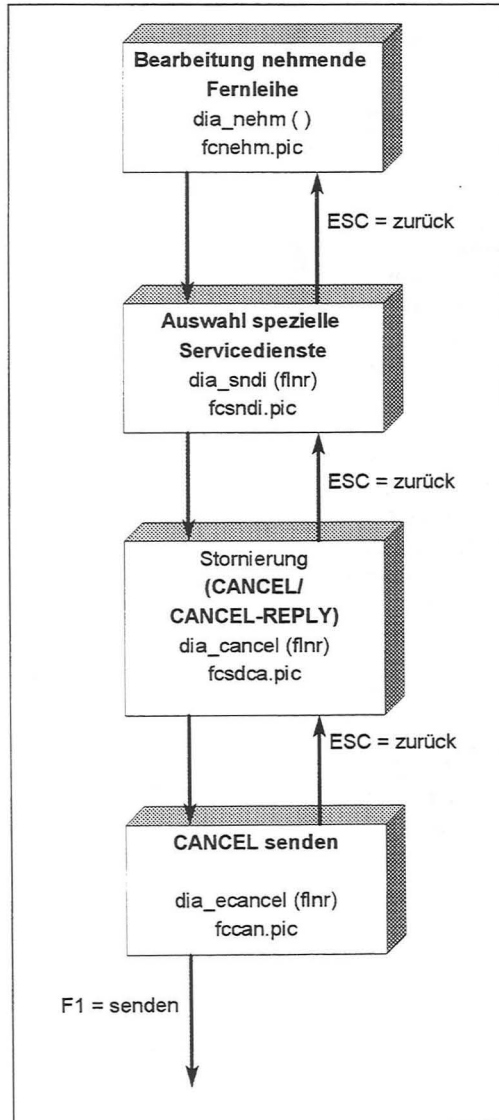
Abb. 15.4
Globale ILL-
Transaktionszu-
stände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
CANCEL senden	CANCEL empfangen
PENDING	IN-PROCESS
FOLGEZUSTAND: CANCEL-PENDING	FOLGEZUSTAND: CANCEL-PENDING

15.4 Menüsteuerung beim Fernleih-Client

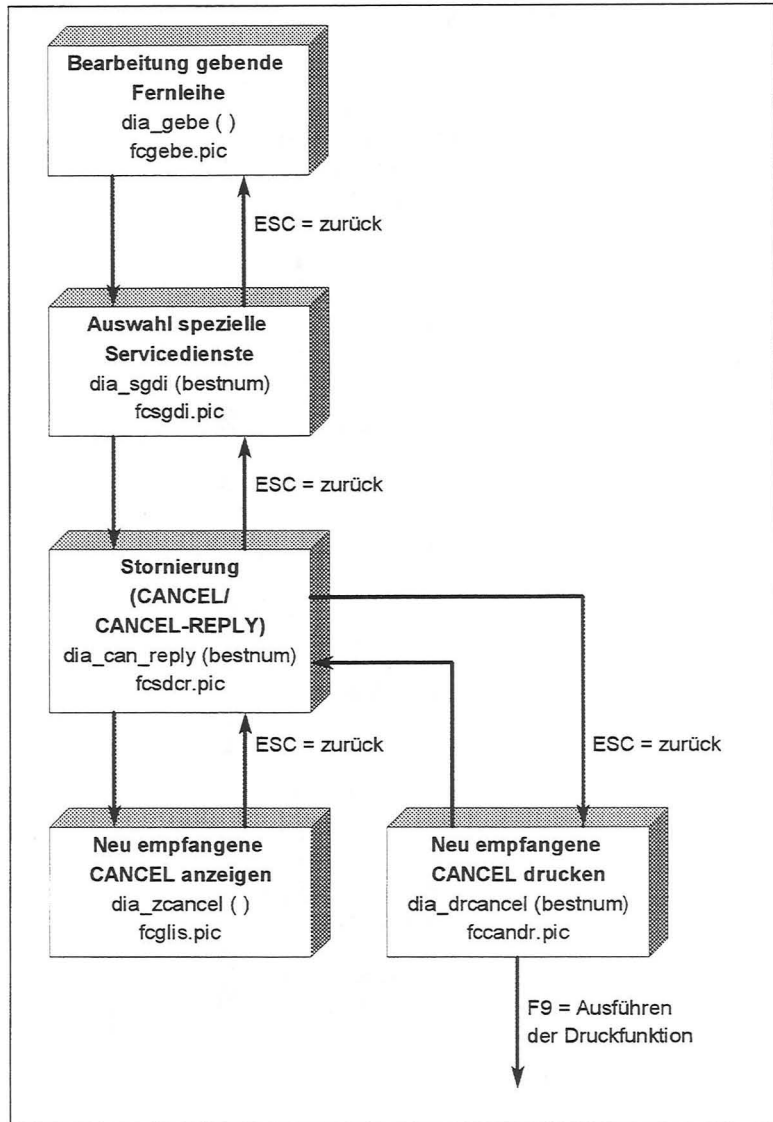
15.4.1 Nehmende Fernleihe

Abb. 15.5
Menü CANCEL
(Requester)



15.4.2 Gebende Fernleihe

Abb. 15.6
Menü CANCEL
(Responder)



16

CANCEL-REPLY

Verwendung von
CANCEL-REPLY

Dieser Dienst erlaubt einem Responder, die Anfrage auf Stornierung einer ILL-Transaktion zu beantworten. Dabei wird erwartet, daß er die Stornierung annimmt oder ablehnt. In jedem Fall aber muß eine Anfrage auf Stornierung durch den Requester erfolgt sein.

Nimmt er die Stornierung an, gilt die Fernleihanfrage als beendet. Nimmt er die Stornierung nicht an, wird die weitere Bearbeitung dem Stand der ILL-Transaktion entsprechend fortgesetzt.

16.1

Senden und Empfangen des ILL-Dienstes CANCEL-REPLY

Ein CANCEL kann nur vom Initiator der Fernleihanfrage, also dem Requester, gesendet werden. Ebenso kann er nur vom Responder empfangen, also auch nur von diesem mit einem CANCEL-REPLY beantwortet werden.

Die folgende Übersicht zeigt, in welchen lokalen Zuständen der ILL-Dienst von der gebenden Bibliothek gesendet werden kann und in welchen lokalen Zuständen die nehmende Bibliothek den CANCEL-REPLY ohne Probleme annehmen kann:

Abb. 16.1
Lokale Zustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
CANCEL-REPLY senden	CANCEL-REPLY empfangen
empf_can_gz	wart_can_rep_nz

16.1.1

Senden des ILL-Dienstes CANCEL-REPLY durch den Responder

Lokaler Zustand
empf_can_gz

Der Responder kann einen CANCEL-REPLY senden, wenn sich die zu stornierende ILL-Transaktion auf der gebenden Seite im lokalen Zustand 'empf_can_gz' befindet:

Die gebende Bibliothek hat also auf eine Fernleihanfrage einen Wunsch auf Stornierung erhalten. Dieser empfangene CANCEL muß beantwortet werden.

Wenn die Stornierung gewährt wird, dann wird die ILL-Transaktion auf der gebenden Seite abgeschlossen. Wird die Stornierung abgelehnt, dann bleibt der Vorgang im aktuellen Zustand bestehen, der vor dem Empfang des Wunsches auf Stornierung vorlag. Die weitere Bearbeitung erfolgt wie bei jedem anderen Vorgang auch.

16.1.2 Empfang des ILL-Dienstes CANCEL-REPLY beim Requester

Lokaler Zustand
wart_can_rep_nz

Ein CANCEL-REPLY kann bei der nehmenden Bibliothek nur empfangen werden, wenn der lokale Zustand 'wart_can_rep_nz' der ILL-Transaktion gesetzt wurde.

Eine Fernleihanfrage wurde durch die nehmende Bibliothek gestellt. Da das Leihobjekt vom Benutzer nicht mehr benötigt wird, wurde ein Wunsch auf Stornierung an die gebende Bibliothek verschickt, die das Objekt noch nicht geliefert hatte. Jetzt wird auf die Antwort des Responders gewartet.

16.2 Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes CANCEL-REPLY

Beim Senden des CANCEL-REPLY wird zunächst der alte lokale Zustand, der den unbeantworteten CANCEL kennzeichnet, abgeschlossen. Das bedeutet, der Datensatz mit der Zustandskennung 'empf_can_gz' erhält eine Endezeit und ein Endedatum. Nach der Erfassung der Antwort auf die Stornierung wird jeweils ein neuer lokaler Zustand generiert, der kennzeichnet, wie die Antwort ausgefallen ist. Auch auf der Empfängerseite des CANCEL-REPLY wird ein neuer Zustand generiert und der alte Zustand abgeschlossen. Die Zustandswechsel werden im folgenden genauer erklärt.

16.2.1 Zustandsübergänge beim Senden des Dienstes CANCEL-REPLY durch den Responder

Wie bereits erwähnt, kann die gebende Bibliothek nur zu einem Vorgang im lokalen Zustand 'empf_can_gz' den Dienst für die Antwort zu einer Stornierung senden. Wünscht ein Fernleih-Client das Versenden eines CANCEL-REPLY, so übergibt er das Kommando *ecre* an den Dialogserver, der prüft, ob dieser Zustand vorliegt.

Wenn der CANCEL-REPLY ordnungsgemäß erfaßt worden ist, wird durch das Programm *feecre.c* der alte Zustand 'empf_can_gz' beendet, d.h. im entsprechenden Datensatz in der Tabelle 'geb_zu_tab' wird die Endezeit und das Endedatum eingetragen. Anschließend wird entweder der neue Zustand 'erl_can_rep_pos_gz' generiert, wenn die Stornierung angenommen wurde oder der neue Zustand 'erl_can_rep_neg_gz', wenn die Stornierung abgelehnt wurde. Es wird überprüft, ob zu diesem Vorgang möglicherweise schon einmal eine Antwort auf Stornierung gegeben wurde. Ist das der Fall, dann wird auch diese abgeschlossen, indem der betreffende Satz eine Endezeit und ein Endedatum erhält. Damit wird gewährleistet, daß immer der aktuelle Stand der Antwort auf einen Stornierwunsch beim Anzeigen der Übersicht zu einer ILL-Transaktion dargestellt wird.

Abschließend wird der Zustand 'send_mail_gz' generiert. An der Sekundärkennung 'cancelrep' erkennt der Sendeserver, daß der Dienst CANCEL-REPLY zu verschicken ist. Der Sendeserver schließt den Zustand 'send_mail_gz' ab, indem er Endezeit und -datum einträgt.

Wurde eine positive Antwort gewünscht, also die Stornierung gewährt, dann werden durch den Sendeserver alle offenen lokalen Zustände beendet und ein abschließender Zustand 'erledigt_gz' generiert. In der Sekundärerkennung dieses Zustands wird mit 'send_cancelrep' vermerkt, daß die Bearbeitung des Fernleihvorgangs durch eine Stornierung beendet wurde.

Bei einer Ablehnung der Stornierung bleiben alle aktuellen Zustände offen und werden regulär weiterbearbeitet.

Abb. 16.2
Zustandsgenerierung
beim Senden
des positiven
CANCEL-REPLY
(Responder)

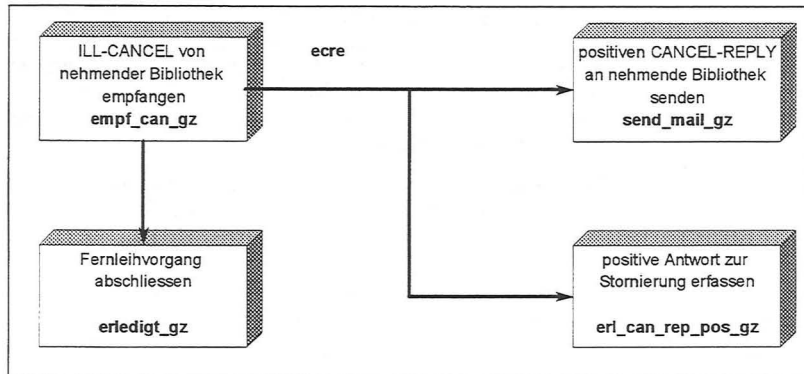
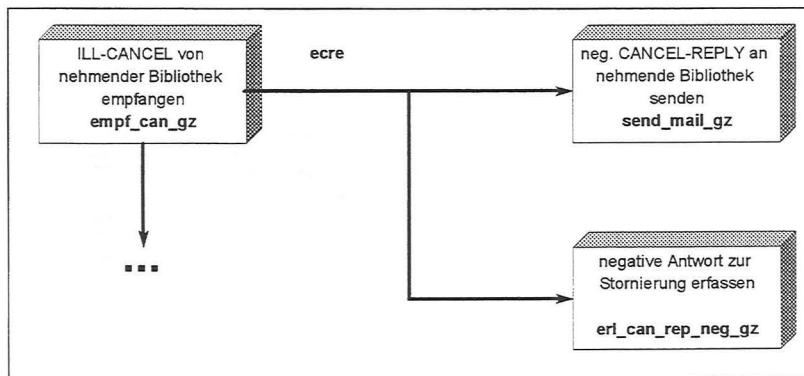


Abb. 16.3
Zustandsgenerierung
beim Senden
des negativen
CANCEL-REPLY
(Responder)



16.2.2 Zustandsübergänge bei Empfang des Dienstes CANCEL-REPLY beim Requester

Der Requester kann den Dienst CANCEL-REPLY nur empfangen, befindet sich die ILL-Transaktion auf der nehmenden Seite im lokalen Zustand 'wart_can_rep_nz'.

Der Empfangsserver prüft, ob beim Eintreffen des CANCEL-REPLY dieser Zustand vorliegt. Der Wartezustand wird in diesem Fall abgeschlossen und ein neuer generiert. Wurde die Stornierung von der gebenden Bibliothek akzeptiert, dann wird der Folgezustand 'empf_can_rep_pos_nz' generiert. Weiterhin wird ein Folgezustand 'beendet_nz' generiert, der kennzeichnet, daß die Vorgangsbearbeitung abgeschlossen ist. Als Sekundärkennung trägt er 'send_cancelrep', woran erkannt werden kann, daß die Vorgangsbearbeitung mit einer akzeptierten Stornierung beendet wurde.

Wurde die Stornierung durch den Responder abgelehnt, dann wird der Folgezustand 'empf_can_rep_neg_nz' generiert. Der aktuelle Zustand, der vor dem Wunsch auf Stornierung bestand, bleibt offen.

Abb. 16.4
Zustandsgenerierung
bei Empfang eines
positiven CANCEL-
REPLY (Requester)

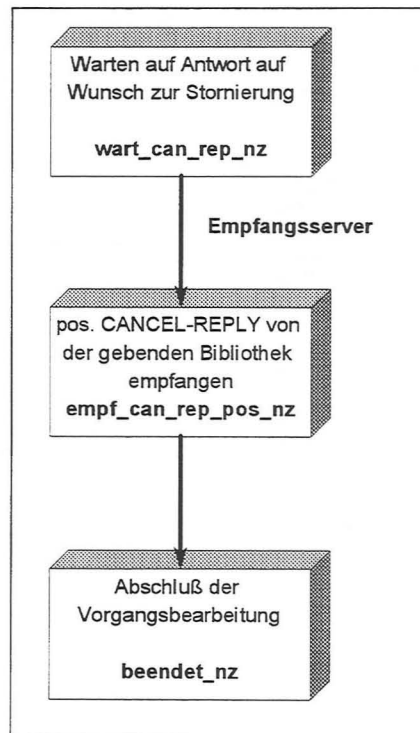
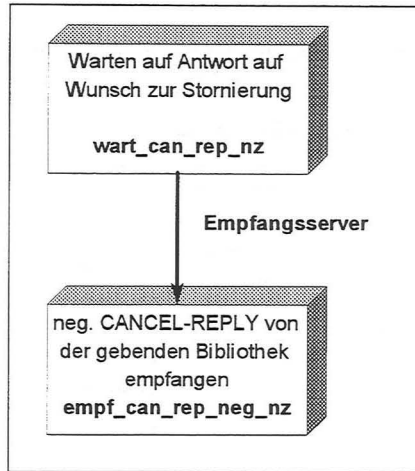


Abb. 16.5
Zustandsgenerierung bei Empfang eines negativen CANCEL-REPLY (Requester)



16.3

Die globalen ILL-Transaktionszustände

Die folgende Tabelle verdeutlicht, in welchen globalen Zuständen der Dienst CANCEL-REPLY von der gebenden Bibliothek gesendet und von der nehmenden Bibliothek empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist.

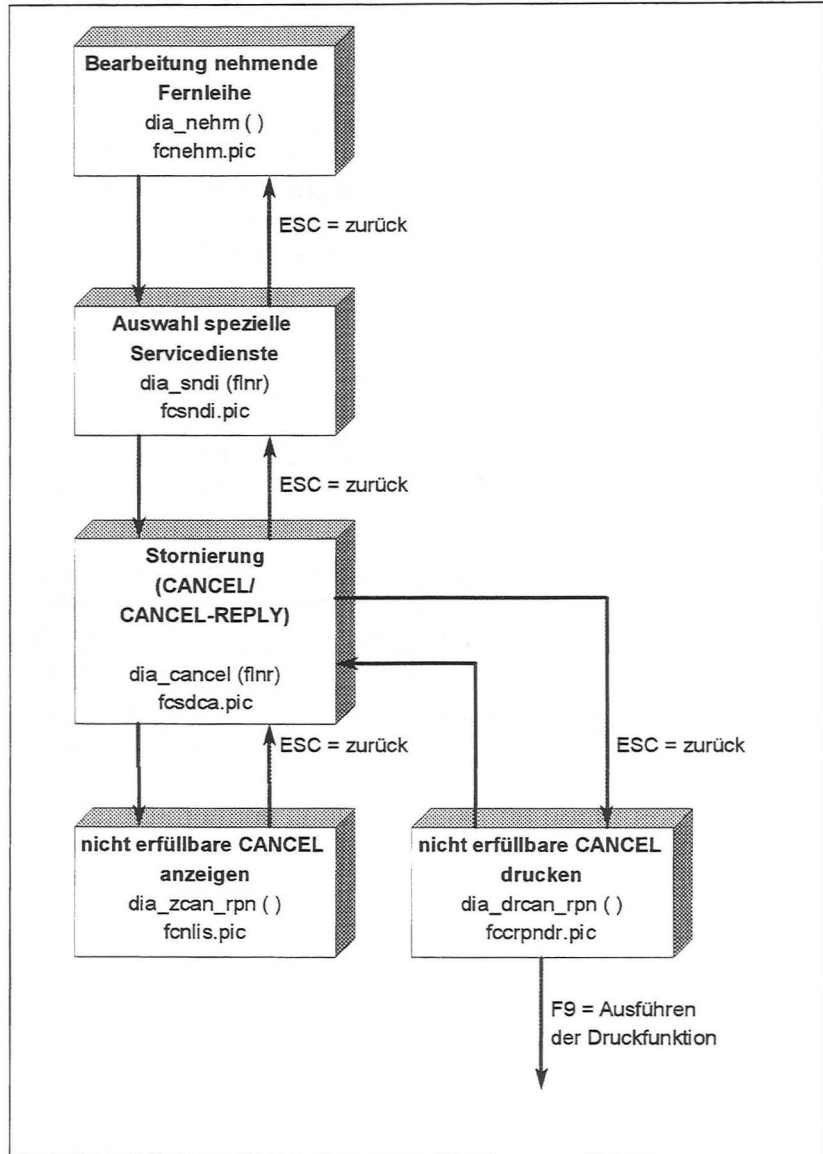
Abb. 16.6
Globale ILL-Transaktionszustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
CANCEL-REPLY senden	CANCEL-REPLY empfangen
CANCEL-PENDING	CANCEL-PENDING
<u>FOLGEZUSTAND:</u> CANCELLED	<u>FOLGEZUSTAND:</u> CANCELLED

16.4 Menüsteuerung beim Fernleih-Client

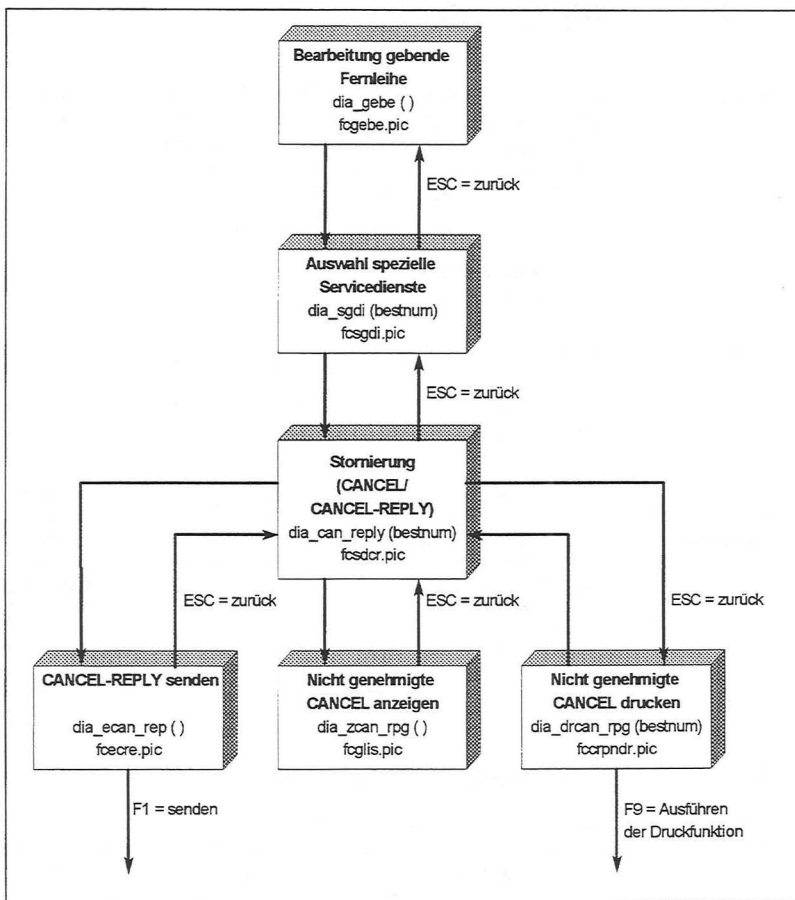
16.4.1 Nehmende Fernleihe

Abb. 16.7
Menü CANCEL-
REPLY (Requester)



16.4.2 Gebende Fernleihe

Abb. 16.8
Menü CANCEL-
REPLY (Responder)



17 RECEIVED

Verwendung von RECEIVED

Dieser ILL-Dienst erlaubt einem Requester anzuzeigen, daß ein gewünschtes Objekt empfangen wurde. Die Benutzung ist optional. Voraussetzung ist, daß auch eine Anfrage gestellt wurde, die vom Responder mit der Lieferung des Objekts erfüllt wurde. Im Fernleihsystem der Universität Konstanz wird der RECEIVED geteilt in:

- Empfang eines Objekts zur Ausleihe,
- Empfang eines Objekts zum Verbleib (z.B. bei Lieferung eines kopierten Zeitschriftenartikels).

17.1 Senden und Empfangen des ILL-Dienstes RECEIVED

Ein RECEIVED kann nur vom Requester gesendet werden. Also kann er auch nur von dem entsprechenden Responder empfangen werden.

Abb. 17.1 Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RECEIVED senden	RECEIVED empfangen
wart_illreq	wart_return_gz
	erledigt_gz

17.1.1 Senden des ILL-Dienstes RECEIVED durch den Requester

Lokaler Zustand
wart_illreq_nz

Der Requester kann den Dienst RECEIVED nur nutzen, wenn sich die entsprechende ILL-Transaktion auf der nehmenden Bearbeitungsseite im lokalen Zustand 'wart_illreq_nz' befindet und er mit Hilfe der Eingangskontrolle feststellt, daß ein von ihm gewünschtes Leihobjekt zu diesem Vorgang eingetroffen ist.

Bei der Eingangskontrolle werden nochmal alle Fernleihdaten für die betreffende Bestellung überprüft.

Im Konstanzer Fernleihsystem wird dabei automatisch ein Satz für den Sender server generiert, der dem Responder anzeigt, daß das Leihobjekt ordnungsgemäß beim Requester eingegangen ist. Automatisch bedeutet hier, bei der Durchführung der Eingangskontrolle wird der RECEIVED ohne Aufruf eines besonderen Menü-Punktes ausgeführt. Für den bearbeitenden Mitarbeiter der Bibliothek ist die Ausführung des ILL-Dienstes RECEIVED also transparent.

17.1.2

Empfangen des ILL-Dienstes RECEIVED durch den Responder

Lokaler Zustand
'wart_return_gz'
oder 'erledigt_gz'

Der Responder kann eine Nachricht über einen RECEIVED beim Requester nur erhalten, wenn er das Leihobjekt auch tatsächlich an diesen abgeschickt hat und sich infolgedessen der Vorgang im Zustand 'wart_return_gz' oder 'erledigt_gz' befindet.

Der Wartezustand ist gesetzt, wenn das Objekt zur Ausleihe verschickt wurde. Dies ist immer der Fall, wenn eine Monographie verliehen wurde, die anschließend von der nehmenden Bibliothek zurückgegeben werden muß.

Der Zustand 'erledigt_gz' besteht bei einer ILL-Transaktion beim Responder, wenn das Objekt zum Verbleib an die nehmende Bibliothek versendet wurde. So muß ein Zeitschriften-Aufsatz nicht zurückgegeben werden. Er wird von der gebenden Bibliothek kopiert und gegen eine Gebühr an die nehmende ausgeliefert. Daher ist eine weitere Bearbeitung durch den Responder nach der Auslieferung zum Verbleib nicht mehr nötig und die ILL-Transaktion gilt als abgeschlossen.

17.2

Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes RECEIVED

Bei der Eingangskontrolle wird vom Requester geprüft, ob die eingetroffenen Objekte bestellt waren, für welche Benutzer sie bereit gestellt werden müssen und ob sie zum Verbleib geliefert wurden oder ob ein Satz in der Ausleih-Verbuchung generiert werden muß. Der Dienst RECEIVED wird dabei automatisch ausgeführt, ohne daß der bearbeitende Bibliothekar einen bestimmten Menü-Punkt wählen muß.

Beim Senden des RECEIVED richtet sich die Generierung von Folgezuständen danach, ob das Objekt zum Verbleib geliefert wurde, was z.B. bei Kopien von Zeitschriften-Aufsätzen der Fall ist, oder ob das Objekt nur verliehen wird. Beim Empfang eines RECEIVED wird sich am lokalen Zustand nichts ändern.

17.2.1

Zustandsübergänge beim Senden des Dienstes RECEIVED durch den Requester

Wird die Eingangskontrolle durchgeführt, werden mehrere Zustände neu generiert. Zuerst wird der lokale Zustand 'wart_illreq_nz' abgeschlossen und der Folgezustand 'benu_bereit_nz' generiert, der kennzeichnet, daß das Objekt für den Benutzer bereitgestellt wurde.

Lieferung zum
Verbleib

Wurde das Leihobjekt zum Verbleib geliefert, wird zunächst ein Satz mit dem Zustand 'send_mail_nz' generiert. Dieser wird durch den Sendeserver verschickt und abgeschlossen. Die Sekundärkennung enthält dabei den Wert 'send_received_v', an welchem der Responder erkennen kann, daß das Objekt zum Verbleib bei der nehmenden Bibliothek eingetroffen ist.

Für die Benachrichtigung des Benutzers wird ein Satz mit dem Zustand 'benach_nz' generiert. Da der Vorgang insgesamt mit dem Eintreffen des Objekts abgeschlossen ist, wenn dieses nicht zurückgeschickt werden muß, wird abschließend noch der Zustand 'beendet_nz' generiert.

RECEIVED

Lieferung zur
Ausleihe

Wenn das Leihobjekt zur Ausleihe geliefert wurde, wird ebenfalls ein Satz mit dem Zustand 'send_mail_nz' generiert, der vom Sendeserver verschickt und abgeschlossen wird. Bei diesem wird aber mit Hilfe der Sekundärkennung 'send_received_a' angezeigt, daß das Objekt beim Requester zur Ausleihe eingetroffen ist.

Auch in diesem Fall wird ein Zustand 'benach_nz' generiert, der kennzeichnet, daß für den Benutzer eine Benachrichtigung verschickt werden soll. Kann das gelieferte Objekt nur im Lesesaal bereitgestellt werden, erhält der Zustand die Sekundärkennung 'nach bereit lese'. Ist das Objekt ausleihfähig, erhält die Sekundärkennung den Wert 'nach_bereit_ausl'.

Ausleihe im
Lesesaal

Wird das Objekt im Lesesaal für den Benutzer zur Verfügung gestellt, dann wird sofort der Zustand 'benu_ausl_nz' generiert. Außerdem wird in der Ausleihverbuchung ein Buchungssatz angelegt, der als Benutzernummer die Nummer des Lesesaals trägt.

Abb. 17.2
Zustandsgenerie-
rung nach Erhalt
eines Objekts zum
Verbleib (Requester)

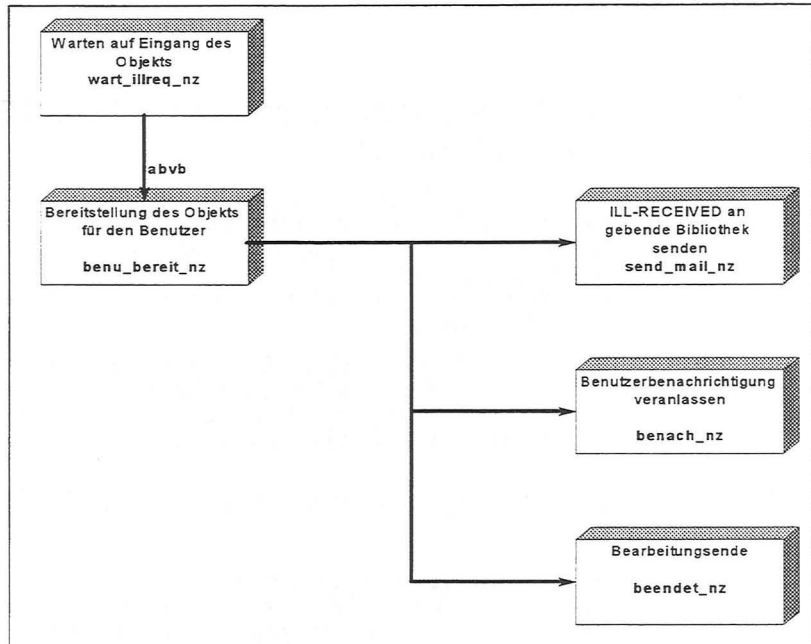
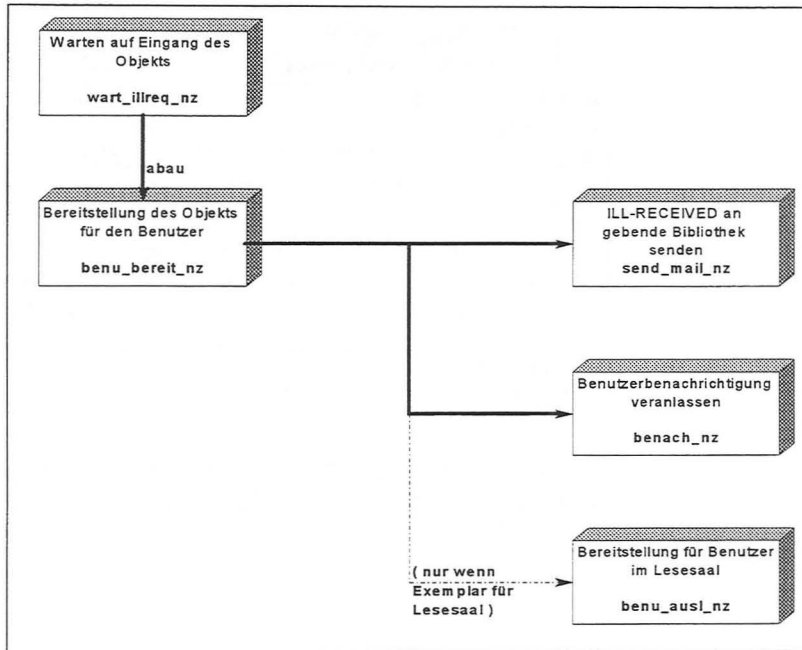


Abb. 17.3
Zustandsgenerierung
nach Erhalt
eines Objekts zur
Ausleihe (Requester)



17.2.2 Zustandsübergänge bei Empfang des Dienstes RECEIVED beim Responder

Beim Empfang eines RECEIVED ändert sich kein lokaler Zustand auf der Responderseite. Der Wartezustand 'wart return_gz' bleibt bestehen, wenn das Objekt zur Ausleihe verschickt wurde. Und auch der Lokalzustand 'erledigt_gz' ändert sich nicht, wenn das Objekt zum Verbleib verschickt wurde.

17.3 Die globalen ILL-Transaktionszustände

Das ILL-Protokoll unterstützt optionale Nachrichten. Die Interaktion RECEIVED-Request ist eine solche optionale Nachricht. Das bedeutet beim Requester muß der globale Transaktionszustand nach dem Empfang eines Leihobjekts auf RECEIVED gesetzt werden. Da die Änderung beim Requester jedoch keinerlei Einfluß auf den Transaktionszustand beim Responder hat, muß diesem diese Zustandsänderung nicht gemeldet werden.

Die folgende Tabelle verdeutlicht, in welchen globalen Transaktionszuständen der Dienst RECEIVED von der nehmenden Bibliothek verschickt und von der gebenden Bibliothek empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist.

RECEIVED

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RECEIVED senden	RECEIVED empfangen
SHIPPED	SHIPPED
PENDING	RENEW-PENDING
CANCEL-PENDING	RENEW-OVERDUE
	OVERDUE
	CHECKED-IN
	LOST
	RECALL
<u>FOLGEZUSTAND:</u> RECEIVED	<u>FOLGEZUSTAND:</u> -

Da der Dienst ILL-RECEIVED automatisch bei der Eingangskontrolle angestoßen wird, soll auf die Darstellung der Menü-Steuerung an dieser Stelle verzichtet werden.

18

RECALL

Verwendung von
RECALL

Dieser Dienst wird vom Responder benutzt, um dem Requester anzuzeigen, daß ein ausgeliehenes Objekt vorzeitig zurückgegeben werden muß. Es wird davon ausgegangen, daß der Requester nach dem Empfang des ILL-Dienstes unverzüglich die Bearbeitung der Rückgabe des Leihobjektes in die Wege leitet. Nachdem der Requester den ILL-Dienst RECALL empfangen hat, darf er keine Anfrage auf Leihfristverlängerung (ILL-Dienst RENEW) mehr an den Responder stellen. Der Responder hingegen darf zu einer ILL-Transaktion nur einmal den ILL-Dienst RECALL versenden.

18.1

Senden und Empfangen des ILL-Dienstes RECALL

Der ILL-Dienst RECALL wird von einem Responder gesendet und von einem Requester empfangen. Das Senden und Empfangen des ILL-Dienstes RECALL ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst RECALL gesendet bzw. empfangen werden kann:

Abb. 18.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RECALL empfangen	RECALL senden
wart_illreq_nz	wart_return_gz
benu_bereit_nz	
benu_ausl_nz	
rücksend_nz	
beendet_nz	

18.1.1

Senden des ILL-Dienstes RECALL durch den Responder

Lokaler Zustand
wart_return_gz

Der Responder hat das vom Requester angeforderte Leihobjekt abgeschickt. Das hat er dem Requester eventuell mit dem optionalen ILL-Dienst SHIPPED angezeigt. Nun wartet er auf die Rückgabe des Leihobjekts. Sollte nun durch gewisse Umstände das Leihobjekt vor Ablauf der vereinbarten Leihfrist vorzeitig zurückgerufen werden, so kann in diesem lokalen Zustand der ILL-Dienst RECALL dazu verwendet werden.

18.1.2 Empfangen des ILL-Dienstes RECALL beim Requester

Der Requester kann den ILL-Dienst RECALL empfangen, wenn er sich in einem der folgenden fünf lokalen Zustände befindet:

Lokaler Zustand wart_illreq_nz	Der Requester hat die Bestellung des Leihobjektes an den Responder verschickt und wartet nun auf den Eingang des Leihobjektes oder auf eine Antwort des Responders. In diesem Zustand kann der Requester den ILL-Dienst RECALL empfangen, damit er vom Rückruf des Responders gleich informiert ist, wenn das Leihobjekt bei ihm eintrifft.
Lokaler Zustand benu_bereit_nz	Der Requester hat das Leihobjekt vom Responder erhalten und die Eingangskontrolle durchgeführt. Der Requester informiert nun den Benutzer, daß sein gewünschtes Objekt eingetroffen ist und zur Abholung bereitsteht. Auch in diesem Zustand ist der Empfang des ILL-Dienstes RECALL zulässig.
Lokaler Zustand benu_ausl_nz	Der Requester hat das Leihobjekt an den Benutzer ausgeliehen und wartet nun auf dessen Rückgabe. In diesem Zustand kann ebenfalls der ILL-Dienst RECALL empfangen werden. In sehr dringenden Fällen kann dann der Benutzer zusätzlich angeschrieben werden, obwohl dessen Leihfrist noch nicht abgelaufen ist.
Lokaler Zustand ruecksend_nz	Der Requester hat das Leihobjekt vom Benutzer zurückerhalten und bearbeitet nun die Rücksendung an den Responder. Auch in diesem Zustand ist der Empfang des ILL-Dienstes RECALL zulässig.
Lokaler Zustand beendet_nz	Der Requester hat das Leihobjekt an den Responder zurückgeschickt. Mit dem Absenden des Leihobjektes an den Responder ist die Transaktion auf der Seite des Requesters abgeschlossen. Selbst in diesem Zustand kann der Requester den ILL-Dienst RECALL empfangen, da der Versand des ILL-Dienstes RETURNED optional ist und der Responder somit eventuell nicht wissen kann, daß das Leihobjekt schon vorzeitig zurückgeschickt wurde. Dies ist z.B. der Fall, wenn der Benutzer das Buch früher als erwartet zurückgibt.

18.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RECALL

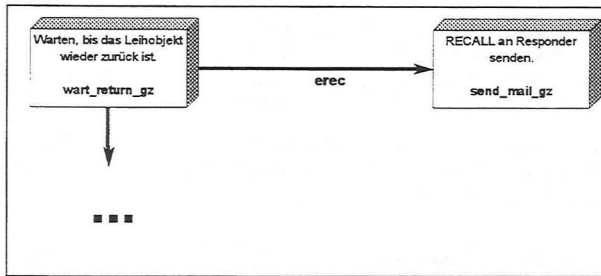
Sowohl beim Senden als auch beim Empfangen des ILL-Dienstes RECALL findet kein Zustandswechsel statt. Es werden lediglich sog. Parallelzustände in der Fernleihdatenbank generiert, die das Versenden bzw. den Empfang des ILL-Dienstes RECALL quittieren.

18.2.1 Zustandsübergänge beim Senden des ILL-Dienstes RECALL durch den Responder

Durch den Aufruf des Kommandos *erec* wird ein neuer Zustandssatz 'send_mail_gz' in der Fernleihdatenbank generiert, der in seiner Sekundärkennung den ILL-Dienst RECALL identifiziert. Dieser wird dann vom Sendeserver verschickt und mit Endedatum und Endezeit abgeschlossen. Der ursprüngliche Zustand der ILL-Transaktion ('wart_return_gz') bleibt also bestehen.

In der folgenden Abbildung wird dieser Vorgang veranschaulicht:

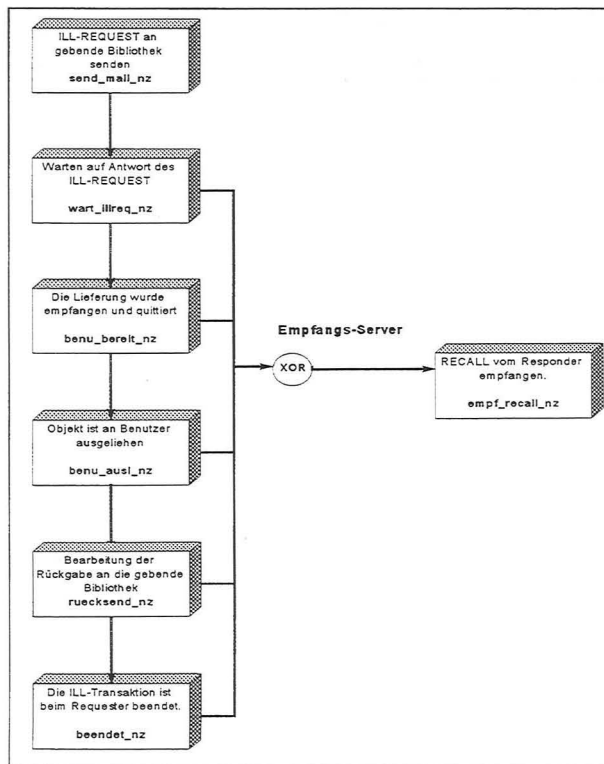
Abb. 18.2
Zustandsübergänge:
Senden RECALL



18.2.2 Zustandsübergänge beim Empfangen des ILL-Dienstes RECALL beim Requester

Durch den Empfang des ILL-Dienstes RECALL bleibt der ursprüngliche Zustand der ILL-Transaktion auf der Seite des Requesters ebenfalls bestehen. Es wird auch hier ein Parallelzustand generiert, der den Empfang des Dienstes anzeigt. Dieser Zustand heißt 'empf_recall_nz'.

Abb. 18.3
Zustandsübergänge:
Empfangen RECALL



18.3 Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände⁸⁸. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand senden darf und ob analog ein Empfänger einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand empfangen darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen der ILL-Dienst RECALL gesendet bzw. empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 18.4
Globale ILL-
Transaktions-
zustände

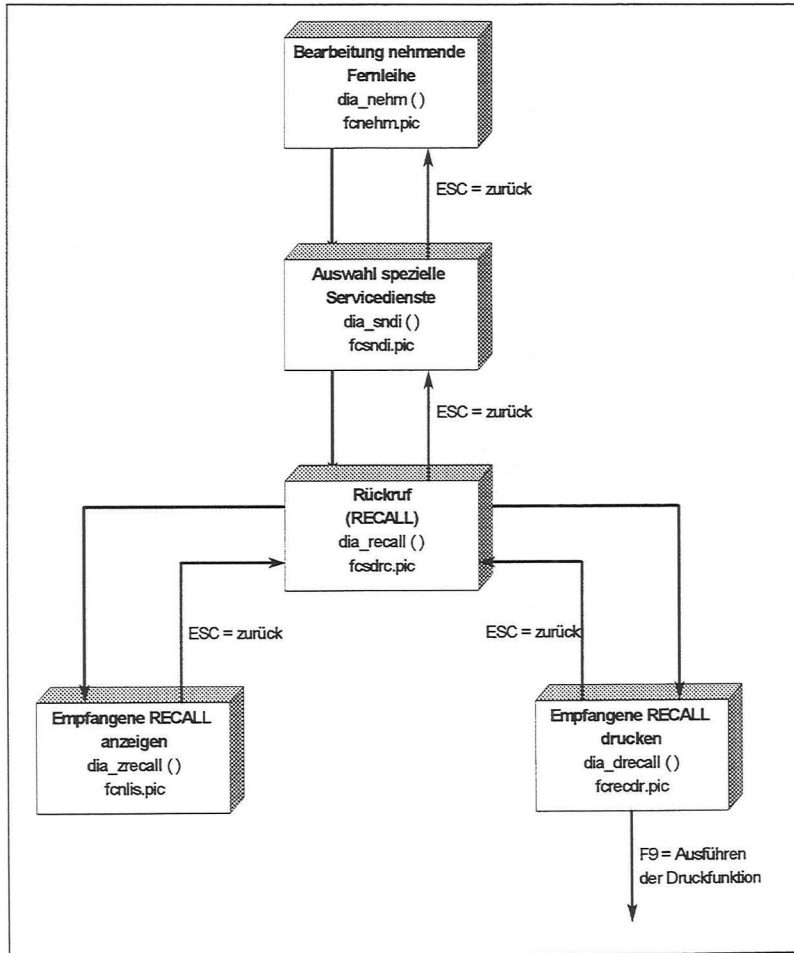
Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RECALL empfangen	RECALL senden
SHIPPED	SHIPPED
RECEIVED	
RENEW / PENDING	RENEW / PENDING
OVERDUE	OVERDUE
RENEW / OVERDUE	RENEW / OVERDUE
NOT-RECEIVED / OVERDUE	
<u>FOLGEZUSTAND:</u> RECALL	<u>FOLGEZUSTAND:</u> RECALL

⁸⁸ Siehe Abschnitt 3.4.4.

18.4 Menüsteuerung beim Fernleih-Client

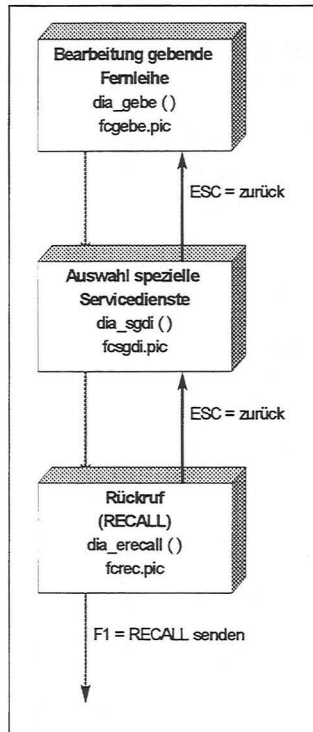
18.4.1 Nehmende Fernleihe (Requester)

Abb. 18.5
Menüsteuerung
RECALL
(Requester)



18.4.2 Gebende Fernleihe (Responder)

Abb. 18.6
Menüsteuerung
RECALL
(Responder)



19

RETURNED

Verwendung von
RETURNED

Dieser optionale ILL-Dienst wird vom Requester benutzt, um dem Responder anzuzeigen, daß ein ausgeliehenes Objekt an ihn zurückgeschickt worden ist. Die ILL-Transaktion ist damit auf der Seite des Requesters vollständig beendet, d.h. der Requester kommt durch das Versenden des Dienstes in einen terminierenden Zustand (lokaler Zustand 'beendet_nz'). Im Fernleihsystem der Uni Konstanz wird dieser Dienst nach der letzten Kontrolle des Leihobjekts durch den Bearbeiter der Rücklieferung (Vollständigkeitskontrolle, Festlegen der Versandart etc.) automatisch an den Responder verschickt. Deshalb entfällt bei der Beschreibung dieses Dienstes der Abschnitt „Menüsteuerung im Fernleih-Client“.

19.1 Senden und Empfangen des ILL-Dienstes RETURNED

Der ILL-Dienst RETURNED wird vom Requester gesendet und vom Responder empfangen. Das Senden und Empfangen des ILL-Dienstes RETURNED ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst RETURNED automatisch gesendet bzw. empfangen werden kann:

Abb. 19.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RETURNED senden	RETURNED empfangen
benu_bereit_nz	wart_return_gz
benu_ausl_nz	
rücksend_nz	

19.1.1 Senden des ILL-Dienstes RETURNED durch den Requester

Lokaler Zustand
benu_bereit_nz

Der Requester hat das Leihobjekt vom Responder erhalten und die Eingangskontrolle durchgeführt. Der Requester informiert nun den Benutzer, daß sein gewünschtes Objekt eingetroffen ist und zur Abholung bereitsteht. Der Benutzer hat in diesem Fall das für ihn bereitgestellte Leihobjekt nicht abgeholt. Nach einer gewissen Zeitspanne wird das Leihobjekt wieder an den Responder zurückgeschickt und mit Abschluß der Bearbeitung der Rücklieferung der ILL-Dienst RETURNED automatisch gesendet.

Lokaler Zustand
benu_ausl_nz

Der Requester hat das Objekt an den Benutzer ausgeliehen und wartet nun auf dessen Rückgabe. Durch die Ausführung der Rückbuchung des Leihobjektes und der Rücklieferungsbearbeitung wird anschließend automatisch der Versand des ILL-Dienstes RETURNED ausgeführt.

RETURNED

Lokaler Zustand
ruecksend_nz

In diesem Zustand ist die Rückbuchung des Leihobjektes im lokalen Verbuchungssystem und im Fernleihsystem schon durchgeführt worden. Nun kann ebenfalls mit dem Abschluß der Rücklieferungsbearbeitung aus diesem Zustand der automatische Versand des Dienstes ausgeführt werden.

19.1.2

Empfangen des ILL-Dienstes RETURNED beim Responder

Lokaler Zustand
wart_return_gz

Der Responder hat das vom Requester angeforderte Leihobjekt abgeschickt. Das hat er dem Requester eventuell mit dem optionalen ILL-Dienst SHIPPED angezeigt. Nun wartet er auf die Rückgabe des Leihobjekts. In diesem lokalen Zustand kann er den ILL-Dienst RETURNED vom Requester empfangen und weiß somit, daß das von ihm ausgeliehene Objekt wieder auch dem Rückweg ist.

19.2

Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RETURNED

Nur beim Senden des ILL-Dienstes RETURNED finden Zustandswechsel statt. Dabei wird der aktuelle Zustand beendet, d.h. Systemdatum und -uhrzeit werden als Endedatum und Endezeit dem Zustand zugeordnet und in die Datenbank eingetragen. Mit dem Abschluß des aktuellen Zustandes wird sofort der nachfolgende Zustand generiert, bei dem Anfangsdatum und -zeit gesetzt werden, Endedatum und -zeit aber bis zu seinem Abschluß offen bleiben.

19.2.1

Zustandsübergänge beim Senden des ILL-Dienstes RETURNED durch den Requester

Ist das Leihobjekt beim Requester eingetroffen, wird eine Eingangskontrolle des empfangenen Leihobjekts durchgeführt. Anschließend muß der Benutzer davon in Kenntnis gesetzt werden, daß das Leihobjekt zum Abholen bereitgestellt ist. Dazu wird in der lokalen Verbuchung ein Bereitstellungssatz generiert, durch den die automatische Benachrichtigung des Benutzers per Post oder per E-Mail erfolgt. Zusätzlich wird im Fernleihsystem ein Bereitstellungssatz angelegt.

Bei der Rückgabe des Leihobjektes durch den Benutzer müssen der angelegte Bereitstellungssatz bzw. der Ausleihsatz, wenn der Benutzer das Leihobjekt auch wirklich abgeholt hat, aus der lokalen Verbuchung wieder gelöscht werden. Momentan bleibt der Bereitstellungs- bzw. Ausleihsatz im Fernleihsystem zu Statistikzwecken noch erhalten. Diesen Vorgang nennt man auch Rückbuchung des Leihobjekts.

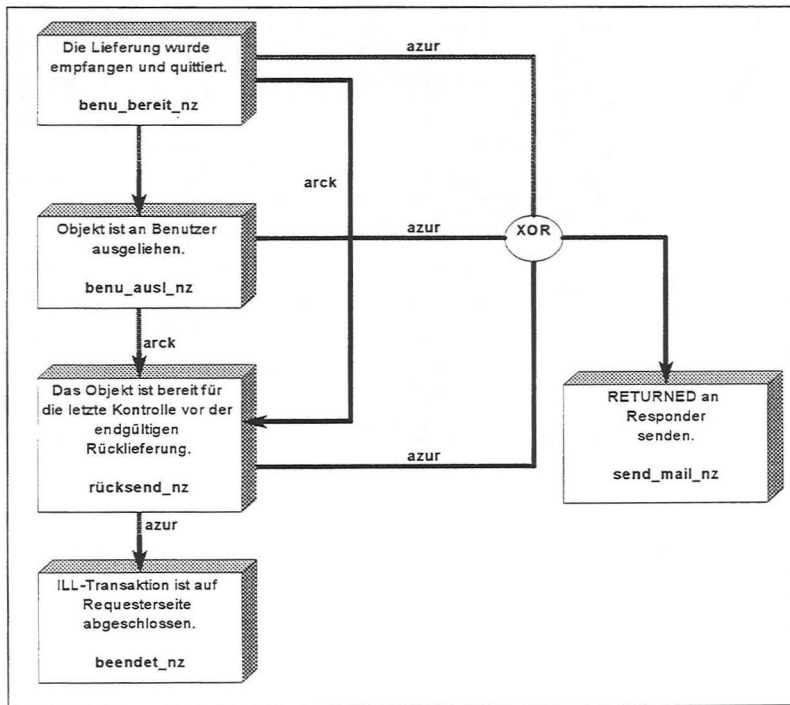
Durch den Aufruf des Kommandos *arck* wird diese Rückbuchung im Fernleihsystem vorgenommen. Ist der lokale Zustand der ILL-Transaktion bei Aufruf des Kommandos 'benu bereit_nz' oder 'benu ausl_nz', wird der Bereitstellungssatz bzw. der Ausleihsatz in der lokalen Verbuchung gelöscht. Die ILL-Transaktion geht anschließend in den Zustand 'ruecksend_nz' über.

Durch den Aufruf des Kommandos *azur* wird schließlich die letzte Kontrolle des Leihobjektes vor dem Zurücksenden an den Responder durchgeführt. Nach dem Durchlaufen dieser Kontrollfunktion wird der Versand des ILL-Dienstes RETURNED an den Responder automatisch ausgeführt. Es wird ein neuer Zustandsatz 'send mail_nz' in der Fernleihdatenbank generiert, der in seiner Sekundärkennung den ILL-Dienst RETURNED identifiziert. Nach der Generierung wechselt die ILL-Transaktion in den Zustand 'beendet_nz'.

Der Sendeserver verschickt anschließend den neu generierten Satz, beendet diesen und außerdem alle noch offenen Zustände zu dieser ILL-Transaktion. Damit ist die ILL-Transaktion auf der Seite des Requesters vollständig abgeschlossen. Der Abschluß der Rücklieferung wird aus dem Kommando *azur* zusätzlich noch in den Zuständen 'benu_bereit_nz' und 'benu_aus_nz' unterstützt, da es Ausnahmefälle gibt, in denen Leihobjekte schon in der Rücklieferungsbearbeitung sind, jedoch noch nicht zurückgebucht wurden. Die Rückbuchung im lokalen Verbuchungssystem wird dann in diesem Kommando ausgeführt.

In der folgenden Abbildung wird dieser Vorgang veranschaulicht:

Abb. 19.2
Zustandsübergänge:
Senden RETURNED



19.2.2

Zustandsübergänge beim Empfangen des ILL-Dienstes RETURNED beim Responder

Durch den Empfang des ILL-Dienstes RETURNED bekommt die ILL-Transaktion lediglich eine Markierung, daß der ILL-Dienst RETURNED vom Requester empfangen wurde, d.h. daß das ausgeliehene Objekt bereits auf dem Rückweg ist. Diese Markierung wird an dem aktuellen Zustandssatz der ILL-Transaktion ('wart_return_gz') vermerkt und in der Fernleihdatenbank aktualisiert. Es werden folglich keine Zustandwechsel durch den Empfang des Dienstes durchgeführt. Somit entfällt für den Empfang des ILL-Dienstes RETURNED die Abbildung der Zustandsübergänge.

19.3 Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände⁸⁹. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand senden darf und ob analog ein Empfänger einen ILL-Dienst im seinem aktuellen globalen ILL-Transaktionszustand empfangen darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen der ILL-Dienst RETURNED gesendet bzw. empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 19.3
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RETURNED senden	RETURNED empfangen
RECEIVED (*)	SHIPPED
RETURNED / PENDING	RETURNED / PENDING
RETURNED / OVERDUE	RETURNED / OVERDUE
OVERDUE	OVERDUE
RECALL	RECALL
	LOST
	CHECKED-IN
<u>FOLGEZUSTAND:</u> RETURNED für (*)	<u>FOLGEZUSTAND:</u> -

⁸⁹ Siehe Abschnitt 3.4.4.

20

CHECKED-IN

Verwendung von
CHECKED-IN

Dieser optionale ILL-Dienst wird vom Responder benutzt, um dem Requester anzuzeigen, daß ein ausgeliehenes Objekt in ordnungsgemäßem Zustand zurückerhalten wurde. Die ILL-Transaktion ist somit auch auf der Seite des Responders vollständig beendet, d.h. der Responder kommt durch das Versenden des Dienstes in einen terminierenden Zustand (lokaler Zustand 'erledigt_gz'). In Ausnahmefällen kann man diesen Dienst auch benutzen, um eine ILL-Transaktion „künstlich“ in einen terminierenden globalen ILL-Transaktionszustand zu bringen, falls Probleme während der Überwachungsphase aufgetreten sind. Dies gilt aber nur für den Fall, daß kein anderer ILL-Dienst mehr zur Behebung der aufgetretenen Probleme verwendet werden kann. Im Fernleihsystem der Uni Konstanz wird der ILL-Dienst CHECKED-IN nach der sogenannten Rückbuchung des Leihobjekts durch den Bearbeiter automatisch an den Requester verschickt. Deshalb entfällt bei der Beschreibung dieses Dienstes der Abschnitt „Menüsteuerung im Fernleih-Client“.

20.1

Senden und Empfangen des ILL-Dienstes CHECKED-IN

Der ILL-Dienst CHECKED-IN wird vom Responder gesendet und vom Requester empfangen. Das Senden und Empfangen des ILL-Dienstes CHECKED-IN ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst CHECKED-IN automatisch gesendet bzw. empfangen werden kann:

Abb. 20.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
CHECKED-IN empfangen	CHECKED-IN senden
vollständig durch globale ILL-Transaktionszustände abgedeckt ⁹⁰	wart_return_gz

20.1.1

Senden des ILL-Dienstes CHECKED-IN durch den Responder

Lokaler Zustand
wart_return_gz

Der Responder hat das vom Requester angeforderte Leihobjekt abgeschickt. Das hat er dem Requester eventuell mit dem optionalen ILL-Dienst SHIPPED angezeigt. Nun wartet er auf die Rückgabe des Leihobjekts. Nach Erhalt des Leihobjekts vom Requester führt er die Eingangskontrolle durch. Ist das Leihobjekt vollständig und in ordnungsgemäßem Zustand zurückgekommen, kann der Responder dies mit dem ILL-Dienst CHECKED-IN bestätigen.

⁹⁰ Siehe Abschnitt 20.3.

20.1.2 Empfangen des ILL-Dienstes CHECKED-IN beim Requester

Für das Empfangen des ILL-Dienstes CHECKED-IN sind keine zusätzlichen Zustände für die lokale Verwaltung der ILL-Transaktion notwendig. Die im ILL-Protokoll vorgesehenen globalen Zustände reichen dabei aus. Deshalb werden an dieser Stelle keine zusätzlichen lokalen Zustandskontrollen durchgeführt.

20.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes CHECKED-IN

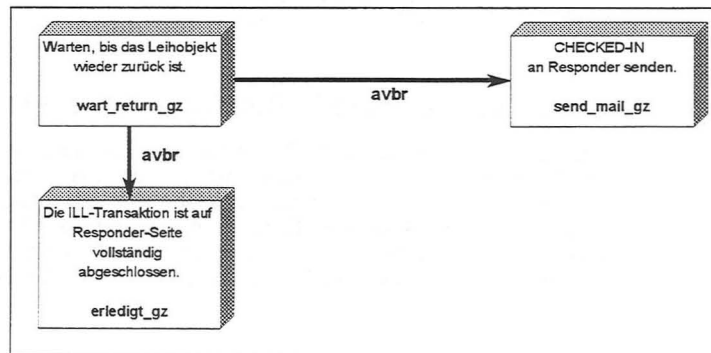
Nur beim Senden des ILL-Dienstes CHECKED-IN finden Zustandswechsel statt. Dabei wird der aktuelle Zustand beendet, d.h. Systemdatum und -uhrzeit werden als Endedatum und Endezeit dem Zustand zugeordnet und in die Datenbank eingetragen. Mit dem Abschluß des aktuellen Zustandes wird sofort der nachfolgende Zustand generiert, bei dem Anfangsdatum und -zeit gesetzt werden, Endedatum und -zeit aber bis zu seinem Abschluß offen bleiben.

20.2.1 Zustandsübergänge beim Senden des ILL-Dienstes CHECKED-IN durch den Responder

Ist das Leihobjekt wieder beim Responder eingetroffen, wird eine Eingangskontrolle des empfangenen Leihobjekts durchgeführt. Neben der bereits erwähnten Vollständigkeitskontrolle (Anzahl der Bände etc.) und der Kontrolle auf eventuelle Beschädigungen am Leihobjekt muß das erhaltene Leihobjekt wieder im lokalen Verbuchungssystem und im Fernleihsystem zurückgebucht werden. Die Ausführung dieser Kontrollen geschieht durch das Kommando *avbr*. In diesem Kommando wird auch der Zustandswechsel von 'wart_return_gz' auf den terminierenden Zustand 'erledigt_gz' durchgeführt. Damit ist die ILL-Transaktion auch auf der Seite des Responders vollständig abgeschlossen. Anschließend wird ein neuer Satz 'send_mail_gz' in der Fernleihdatenbank generiert, der in seiner Sekundärkennung den ILL-Dienst CHECKED-IN identifiziert. Der Sendeserver verschickt nun diesen eben generierten Satz, schließt diesen und außerdem noch alle offenen Zustandssätze der ILL-Transaktion ab.

In der folgenden Abbildung wird dieser Vorgang veranschaulicht:

Abb. 20.2
Zustandsübergänge:
Senden CHECKED-
IN



20.2.2 Zustandsübergänge beim Empfangen des ILL-Dienstes CHECKED-IN beim Requester

Durch den Empfang des ILL-Dienstes CHECKED-IN bekommt die ILL-Transaktion lediglich eine Markierung, daß der ILL-Dienst CHECKED-IN vom Responder empfangen wurde, d.h. daß der Responder das Leihobjekt wieder erhalten hat. Diese Markierung wird an dem Zustandssatz der ILL-Transaktion in der ILL-Transaktionstabelle der Fernleihdatenbank vermerkt und dort aktualisiert. Es werden folglich keine Zustandwechsel durch den Empfang des Dienstes durchgeführt. Somit entfällt für den Empfang des ILL-Dienstes CHECKED-IN die Abbildung der Zustandsübergänge.

20.3 Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände⁹¹. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand senden darf und ob analog ein Empfänger einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand empfangen darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen der ILL-Dienst CHECKED-IN gesendet bzw. empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 20.3
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
CHECKED-IN empfangen	CHECKED-IN senden
PENDING	
CANCEL / PENDING	
RECEIVED	
RENEW / PENDING	RENEW / PENDING
RENEW / OVERDUE	RENEW / OVERDUE
SHIPPED	
NOT-RECEIVED-OVERDUE	
OVERDUE	OVERDUE
RETURNED	
RECALL	RECALL
<u>FOLGEZUSTAND:</u> RETURNED	<u>FOLGEZUSTAND:</u> CHECKED-IN

⁹¹ Siehe Abschnitt 3.4.4.

21 OVERDUE

Verwendung von OVERDUE

Dieser Dienst erlaubt es einem Responder, dem Requester mitzuteilen, daß ein Objekt überfällig ist, wenn das Fälligkeitsdatum für dieses Objekt erreicht ist. Diese Aktion kann automatisch durch das System des Responders oder manuell durch eine Person auf Responderseite ausgelöst werden. Vom Requester wird erwartet, daß er das geliehene Objekt zurückgibt oder eine Verlängerung (RENEW) beantragt.

21.1 Senden und Empfangen des ILL-Dienstes OVERDUE

Ein OVERDUE kann nur von einer gebenden Bibliothek gesendet und nur von einer nehmenden Bibliothek empfangen werden. Das Senden und Empfangen eines OVERDUE ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst OVERDUE von der gebenden Bibliothek gesendet und von der nehmenden empfangen werden kann:

Abb. 21.1
Lokale Zustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
OVERDUE senden	OVERDUE empfangen
wart_return_gz	wart_illreq_nz
	benu_bereit_nz
	benu_ausl_nz
	ruecksend_nz
	beendet_nz

21.1.1 Senden des ILL-Dienstes OVERDUE durch den Responder

Lokaler Zustand
wart_return_gz

Der Responder kann einen OVERDUE senden, wenn die betreffende ILL-Transaktion sich im lokalen Zustand 'wart_return_gz' befindet:

Der Responder hat vom Requester eine Fernleihanfrage mit dem Dienst ILL-REQUEST erhalten. Er hat die Anfrage bearbeitet und das gewünschte Objekt zur Ausleihe an den Requester versendet und wartet jetzt auf die Rückgabe des verliehenen Objekts.

Durch die lokale Verbuchung wird gemeldet, daß die Ausleihfrist abgelaufen ist. Nun wird entweder manuell oder bei elektronischer Kopplung mit dem Verbuchungssystem automatisch die Erfassung des OVERDUE angestoßen.

21.1.2 Empfangen des ILL-Dienstes OVERDUE beim Requester

Lokaler Zustand
wart_illreq_nz

Der Requester kann den ILL-Dienst OVERDUE empfangen, wenn sich die zugehörige ILL-Transaktion im lokalen Zustand 'wart_illreq_nz' befindet.

Der Requester hat eine Fernleihanfrage an den Responder geschickt und wartet noch auf eine Antwort der gebenden Bibliothek oder auf das gewünschte Leihobjekt. Wenn in diesem Zustand ein OVERDUE empfangen wird, dann ist das Leihobjekt entweder noch auf dem Transportweg vom Responder zum Requester, wenn es zu Verzögerungen gekommen ist oder das Objekt ist möglicherweise verlorengegangen. In diesem Fall würde als logische Konsequenz nach dem Erhalt des Dienstes OVERDUE eine Nachricht mit dem ILL-Dienst MESSAGE an die gebende Bibliothek gesendet werden, um den Sachverhalt zu klären.

Lokaler Zustand
benu_bereit_nz

Befindet sich die ILL-Transaktion im lokalen Zustand 'benu_bereit_nz', dann wurde das Leihobjekt für den Benutzer zur Ausleihe bereitgestellt. Wurde dabei ein vorgegebenes Zeitlimit für die Abholung des Objekts noch nicht überschritten, dann kann der Benutzer benachrichtigt werden. Dieser hat die Möglichkeit, da es beim Transport des Objekts offenbar zu Verzögerungen gekommen ist, durch die nehmende Bibliothek einen Antrag auf Leihfristverlängerung (ILL-Dienst RENEW) zu stellen.

Lokaler Zustand
benu_ausl_nz

Das Leihobjekt befindet sich bereits beim Benutzer, wenn der Vorgang auf nehmender Seite den lokalen Zustand 'benu_ausl_nz' erreicht hat.

Es muß in diesem Fall eine Benutzernachricht veranlaßt werden. Entweder gibt der Leser das Objekt zurück oder es muß auch in diesem Fall eine Ausleihfristverlängerung (ILL-Dienst RENEW) beantragt werden.

Lokaler Zustand
ruecksend_nz

Im lokalen Zustand 'ruecksend_nz' hat der Benutzer das Leihobjekt bereits zurückgebracht. Es steht nun zur Rücksendung an die gebende Bibliothek bereit, daher sind keine besonderen Aktionen in Verbindung mit dem Empfang des OVERDUE erforderlich.

Lokaler Zustand
beendet_nz

Wurde das betreffende Objekt bereits von der nehmenden an die gebende Bibliothek zurückgegeben, dann befindet sich die ILL-Transaktion im lokalen Zustand 'beendet_nz'. In diesem Fall ist keine weitere Aktion durch den Requester mehr nötig.

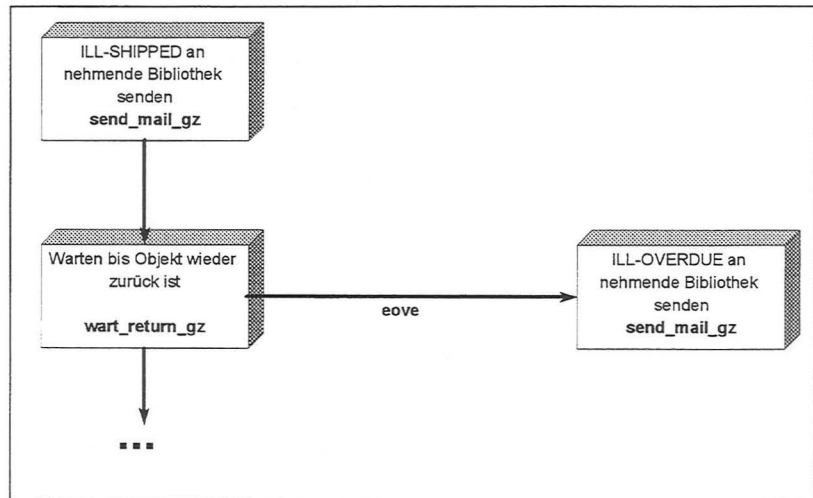
21.2 Lokale Zustandsübergänge beim Senden und Empfangen des Dienstes OVERDUE

Sowohl beim Senden als auch beim Empfangen des Dienstes OVERDUE findet kein Zustandswechsel statt. Der aktuelle Zustand bleibt bestehen und es wird ein paralleler Zustand generiert, bei dem Anfangsdatum und -zeit gesetzt werden. Enddatum und -zeit aber bleiben bis zu seinem regulären Abschluß offen.

21.2.1 Zustandsübergänge beim Senden des Dienstes OVERDUE durch den Responder

Wie bereits erwähnt, kann die gebende Bibliothek nur zu einem Vorgang im lokalen Zustand 'wart_return_gz' den Dienst zur Leihfristüberschreitung senden. Wünscht ein Fernleih-Client das Versenden eines OVERDUE, so übergibt er das Kommando *eove* an den Dialogserver, der prüft, ob dieser Zustand vorliegt. Nur wenn dies der Fall ist, wird durch das Programm *feove.c* der neue Zustand 'send_mail_gz' zum Versenden des OVERDUE generiert. Sobald der Sendeserver diesen sendefähigen Dienst verschickt hat, beendet er den Zustand 'send_mail_gz' durch das Setzen von Endezeit und Endedatum. Ein neuer Zustand wird nicht generiert, der lokale Zustand 'wart_return_gz' bleibt bestehen. Wenn das Leihobjekt vom Requester zurückkehrt, wird der Vorgang regulär abgeschlossen.

Abb. 21.2
Zustandsgenerierung bei Senden eines OVERDUE (Responder)



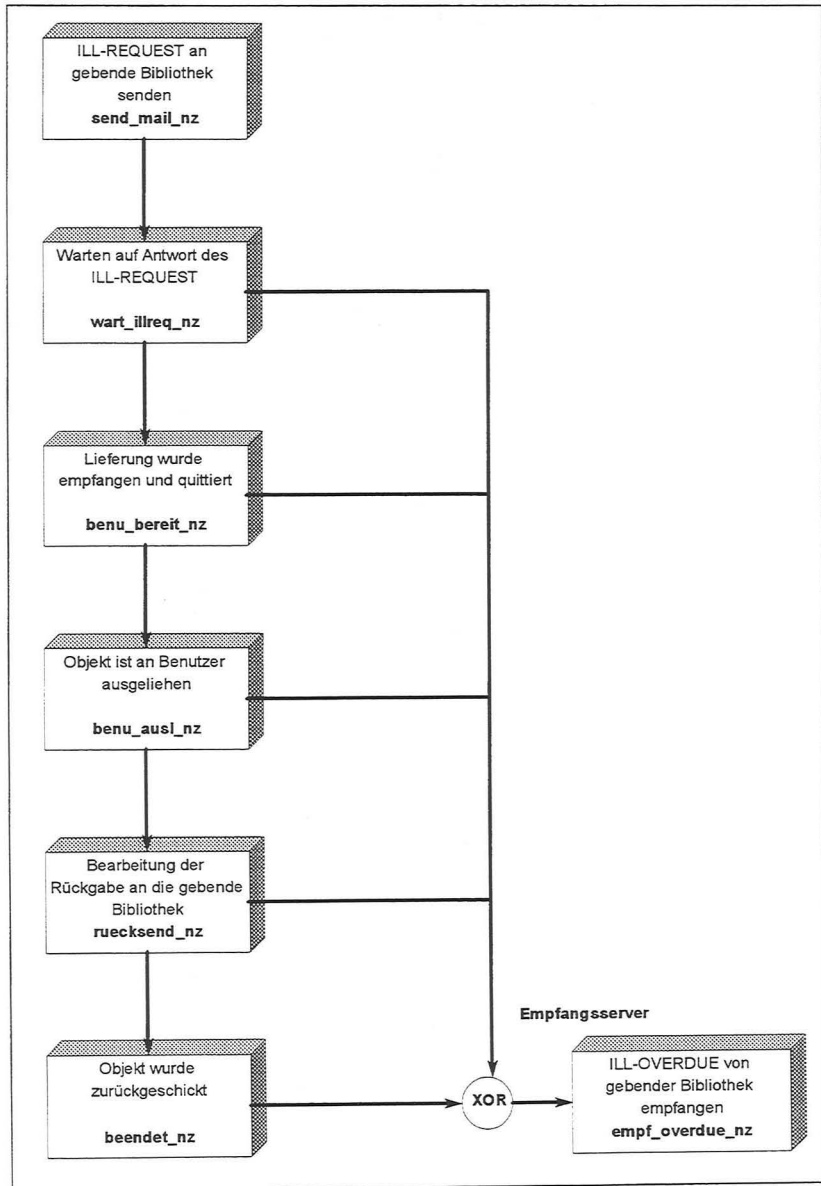
21.2.2 Zustandsübergänge beim Empfangen des Dienstes OVERDUE beim Requester

Der Requester kann den Dienst OVERDUE nur empfangen, wenn sich der betreffende Vorgang in einem der folgenden Zustände befindet:

- 'wart_illreq_nz',
- 'benu_bereit_nz',
- 'benu_ausl_nz',
- 'ruecksend_nz' oder
- 'beendet_nz'.

Der Empfangs-Server überprüft, ob beim Eintreffen eines OVERDUE einer dieser Zustände vorliegt. Der Server generiert darauf den parallelen Zustand 'empf_overdue_nz'. Die Handhabung des weiteren Vorgehens erfolgt momentan getrennt vom Fernleihprogramm.

Abb. 21.3
Zustandsgenerierung
bei Empfang eines OVERDUE
(Requester)



21.3 Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände.⁹² Sie dienen zur Prüfung, ob ein ILL-Dienst im aktuellen globalen ILL-Transaktionszustand von der Senderseite verschickt und ein ILL-Dienst im aktuellen globalen ILL-Transaktionszustand auf Empfängerseite erhalten werden darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen ein OVERDUE von der gebenden Bibliothek gesendet und von der nehmenden empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist.

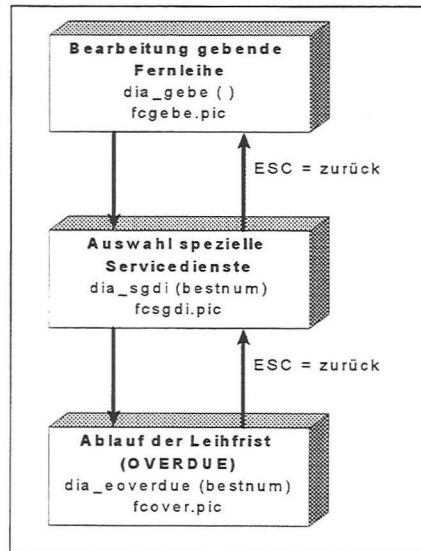
Abb. 21.4
Globale ILL-
Transaktions-
zustände

Responder (gebende Bibliothek)	Requester (nehmende Bibliothek)
OVERDUE senden	OVERDUE empfangen
SHIPPED	SHIPPED
RENEW/PENDING	RENEW/PENDING
RENEW/OVERDUE	RENEW/OVERDUE
	NOT-RECEIVED/OVERDUE
	RECEIVED
<u>FOLGEZUSTAND:</u> OVERDUE	<u>FOLGEZUSTAND:</u> OVERDUE

21.4 Menüsteuerung beim Fernleih-Client

21.4.1 Gebende Fernleihe

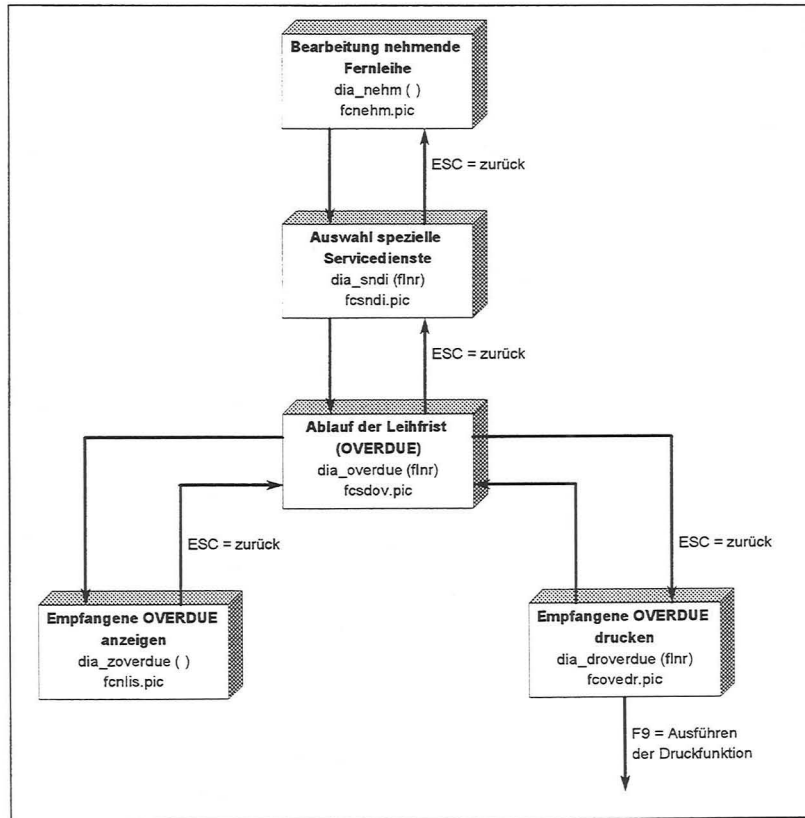
Abb. 21.5
Menü OVERDUE
(Responder)



⁹² Siehe Abschnitt 3.4.4.

21.4.2 Nehmende Fernleihe

Abb. 21.6
Menü OVERDUE
(Requester)



22

RENEW

Verwendung von
RENEW

Dieser Dienst wird vom Requester benutzt, um beim Responder eine Verlängerung der Leihfrist eines ausgeliehenen Objekts zu beantragen. Er kann dabei ein von ihm gewünschtes neues Leihfristdatum angeben. Der Antrag auf Leihfristverlängerung muß aber nicht zwangsläufig am Ende der regulären Leihfrist geschehen. Dieser ILL-Dienst kann benutzt werden, sobald der Verlängerungswunsch besteht. Der ILL-Dienst kann nicht verwendet werden, wenn eine der folgenden Situationen vorliegt:

- Der Requester hat zuvor vom Responder den ILL-Dienst RECALL empfangen.
- Der Requester hat schon einmal einen Leihfristverlängerungs-Antrag zur selben ILL-Transaktion an den Responder gesendet, hat jedoch von diesem noch keine Antwort erhalten.
- Der Requester hat zur selben ILL-Transaktion auf einen Leihfristverlängerungs-Antrag bereits eine negative Antwort vom Responder erhalten.
- Dem Requester wurde zur selben ILL-Transaktion bereits eine Leihfristverlängerung vom Responder gewährt, jedoch mit dem Vermerk, daß keine weiteren Leihfristverlängerungs-Anträge mehr zulässig sind.
- Zwischen Requester und Responder wurde vereinbart (Nutzkonditionen), daß zur selben ILL-Transaktion generell keine Anträge auf Leihfristverlängerung zulässig sind.

22.1

Senden und Empfangen des ILL-Dienstes RENEW

Der ILL-Dienst RENEW wird vom Requester gesendet und vom Responder empfangen. Das Senden und Empfangen des ILL-Dienstes RENEW ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst RENEW gesendet bzw. empfangen werden kann:

Abb. 22.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RENEW senden	RENEW empfangen
benu_bereit_nz	wart_return_gz
benu_ausl_nz	

22.1.1 Senden des ILL-Dienstes RENEW durch den Requester

Lokaler Zustand
benu_bereit_nz

Der Requester hat das Leihobjekt vom Responder erhalten und die Eingangskontrolle durchgeführt. Der Requester informiert nun den Benutzer, daß sein gewünschtes Objekt eingetroffen ist und zur Abholung bereitsteht. In diesem Zustand kann der Benutzer schon den Wunsch äußern, daß er das Objekt gerne länger ausgeliehen hätte als ursprünglich angegeben. In diesem Zustand ist das Senden des ILL-Dienstes RENEW zulässig.

Lokaler Zustand
benu_ausl_nz

Der Requester hat das Leihobjekt an den Benutzer ausgeliehen und wartet nun auf dessen Rückgabe. Der Benutzer merkt nun beispielsweise, daß er das Leihobjekt doch länger benötigt als ursprünglich gedacht und kann den Wunsch auf Leihfristverlängerung äußern. In diesem Zustand kann ebenfalls der ILL-Dienst RENEW gesendet werden.

22.1.2 Empfangen des ILL-Dienstes RENEW beim Responder

Lokaler Zustand
wart_return_gz

Der Responder hat das vom Requester angeforderte Leihobjekt abgeschickt. Das hat er dem Requester eventuell mit dem optionalen ILL-Dienst SHIPPED angezeigt. Nun wartet er auf die Rückgabe des Leihobjekts. In diesem lokalen Zustand kann er den ILL-Dienst RENEW vom Requester empfangen.

22.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RENEW

Sowohl beim Senden als auch beim Empfangen des ILL-Dienstes RENEW findet kein Zustandswechsel statt. Es werden lediglich sog. Parallelzustände in der Fernleihdatenbank generiert, die zur Kontrolle und Steuerung der Abhandlung des Leihfristverlängerungs-Antrages benutzt werden.

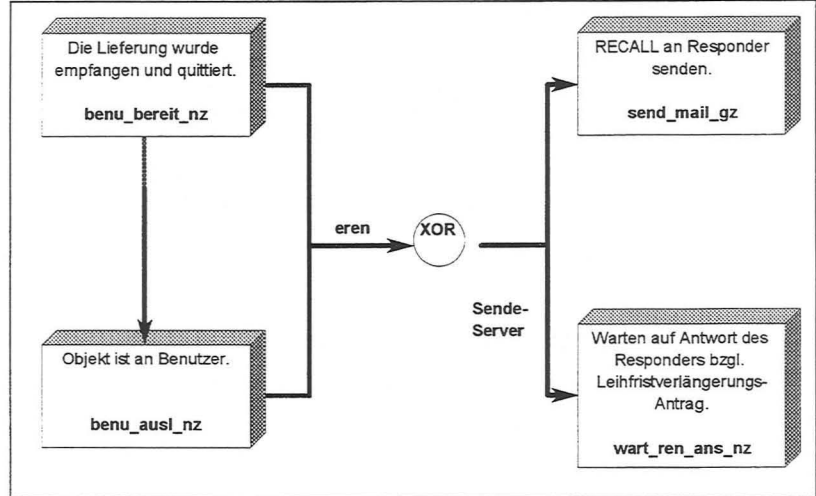
22.2.1 Zustandsübergänge beim Senden des ILL-Dienstes RENEW durch den Requester

Durch den Aufruf des Kommandos *eren* wird ein neuer Zustandssatz 'send_mail_nz' in der Fernleihdatenbank generiert, der in seiner Sekundärkennung den ILL-Dienst RENEW identifiziert. In diesem Satz ist auch das vom Requester gewünschte neue Leihfristdatum enthalten. Dieser wird dann vom Sendeserver verschickt und mit Endedatum und Endezeit abgeschlossen. Zusätzlich wird ein weiterer Parallelzustand generiert, der signalisiert, daß ein Leihfristverlängerungs-Antrag an den Responder versendet wurde und nun auf eine Antwort (ILL-Dienst RENEW-ANSWER⁹³) von diesem gewartet wird. Dieser Parallelzustand heißt 'wart_ren_ans_nz' und bleibt solange offen (ohne Endedatum und Endezeit), bis eine Antwort vom Responder erhalten wurde. Der ursprüngliche Zustand der ILL-Transaktion ('benu_bereit_nz' oder 'benu_ausl_nz') bleibt dabei aber bestehen. Ebenso wird der weitere Verlauf der ILL-Transaktion durch die Generierung dieser Parallelzustände nicht beeinflusst.

⁹³ Siehe Abschnitt 23.

In der folgenden Abbildung wird dieser Vorgang veranschaulicht:

Abb. 22.2
Zustandsübergänge:
Senden RENEW

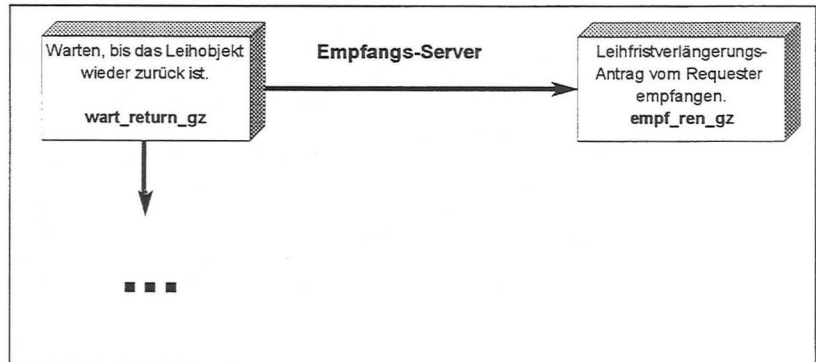


22.2.2

Zustandsübergänge beim Empfangen des ILL-Dienstes RENEW beim Responder

Durch den Empfang des ILL-Dienstes RENEW bleibt der ursprüngliche Zustand der ILL-Transaktion auf der Seite des Responders ebenfalls bestehen. Es wird auch hier ein Parallelzustand generiert, der den Empfang des Dienstes anzeigt. Dieser Zustand heißt 'empf_ren_gz' und beinhaltet das vom Requester gewünschte neue Leihfristdatum. Der Responder antwortet nun dem Requester auf seinen Leihfristverlängerungs-Antrag mit dem ILL-Dienst RENEW-ANSWER⁹⁴.

Abb. 22.3
Zustandsübergänge:
Empfangen RENEW



⁹⁴ Siehe Abschnitt 23.

22.3

Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände⁹⁵. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand senden darf und ob analog ein Empfänger einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand empfangen darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen der ILL-Dienst RENEW gesendet bzw. empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 22.4
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RENEW senden	RENEW empfangen
RECEIVED (*)	SHIPPED (*)
RENEW / PENDING	RENEW / PENDING
RENEW / OVERDUE	RENEW / OVERDUE
OVERDUE (**)	OVERDUE (**)
<u>FOLGEZUSTÄNDE:</u> RENEW / PENDING für (*) RENEW / OVERDUE für (**)	<u>FOLGEZUSTÄNDE:</u> RENEW / PENDING für (*) RENEW / OVERDUE für (**)

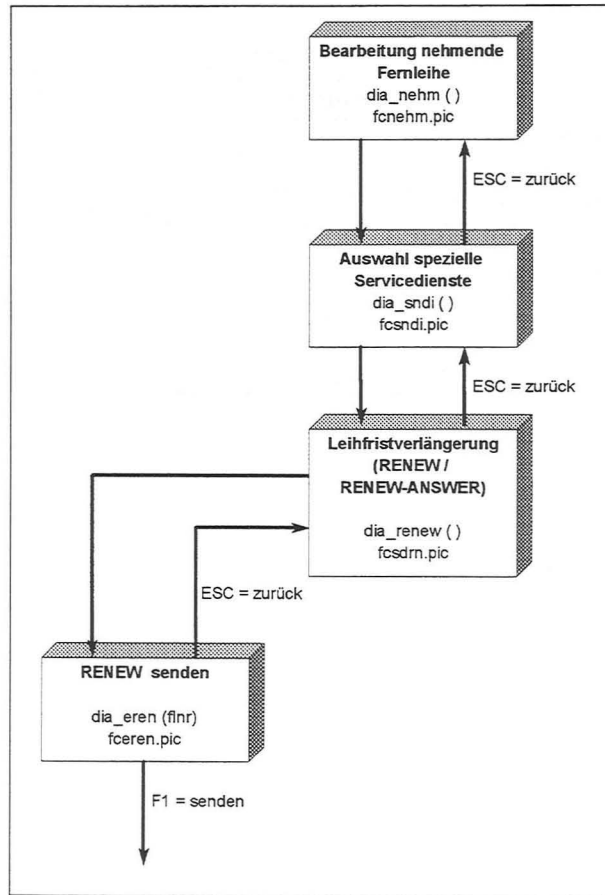
⁹⁵ Siehe Abschnitt 3.4.4.

22.4 Menüsteuerung beim Fernleih-Client

Die ILL-Dienste RENEW und RENEW-ANSWER sind auf Requester- und Responderseite in einem Menüpunkt zusammengefaßt. In der Menüsteuerung werden aber immer nur diejenigen (Unter-)Menüpunkte aufgeführt, die zu dem beschriebenen ILL-Dienst gehören.

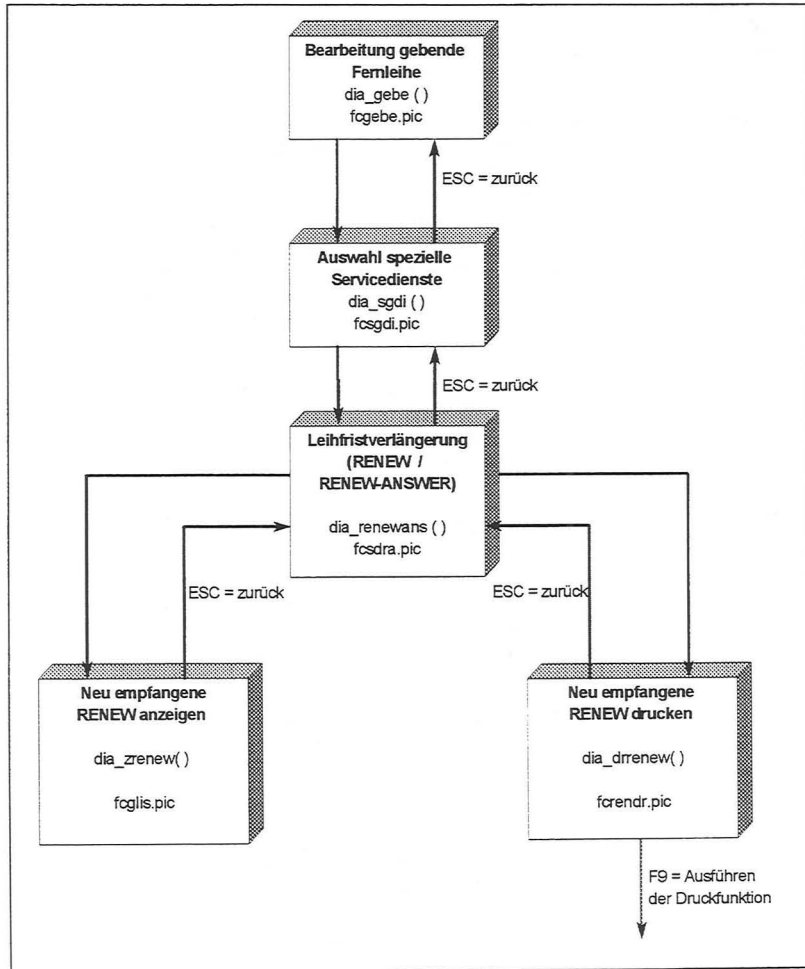
22.4.1 Nehmende Fernleihe (Requester)

Abb. 22.5
Menüsteuerung
RENEW (Requester)



22.4.2 Gebende Fernleihe (Responder)

Abb. 22.6
Menüsteuerung
RENEW (Responder)



23

RENEW-ANSWER

Verwendung von
RENEW-ANSWER

Dieser Dienst wird vom Responder benutzt, um einem Requester auf einen Leihfristverlängerungs-Antrag zu antworten. Beantwortet der Responder den Antrag positiv, muß er in seiner Antwort der Verlängerung der Leihfrist generell zustimmen. Zusätzlich muß er das gewährte neue Leihfristedatum angeben. Er kann dabei entweder das vom Requester in seinem Antrag angegebene Datum akzeptieren oder ein anderes angeben. Desweiteren kann der Responder in seiner Antwort angeben, ob für diese ILL-Transaktion weitere Anträge auf Leihfristverlängerung zugelassen werden sollen oder nicht. Er kann aber auch negativ antworten, indem er den Antrag auf Leihfristverlängerung generell ablehnt.

Als einzige Voraussetzung für die Verwendung des ILL-Dienstes RENEW-ANSWER gilt, daß zuvor ein Antrag auf Leihfristverlängerung für eine bestimmte ILL-Transaktion empfangen worden sein muß, der bislang noch nicht beantwortet wurde.

23.1

Senden und Empfangen des ILL-Dienstes RENEW-ANSWER

Der ILL-Dienst RENEW-ANSWER wird vom Responder gesendet und vom Requester empfangen. Das Senden und Empfangen des ILL-Dienstes RENEW-ANSWER ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt, in welchen lokalen Zuständen der ILL-Dienst RENEW-ANSWER gesendet bzw. empfangen werden kann:

Abb. 23.1
Lokale Zustände

Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RENEW-ANSWER empfangen	RENEW-ANSWER senden
wart_ren_ans_nz	empf_ren_gz

23.1.1

Senden des ILL-Dienstes RENEW-ANSWER durch den Responder

Lokaler Zustand
empf_ren_gz

Der Responder hat einen Antrag auf Leihfristverlängerung zu einer bestimmten ILL-Transaktion vom Requester bekommen. Diesen hat er bislang noch nicht beantwortet.

23.1.2

Empfangen des ILL-Dienstes RENEW-ANSWER durch den Requester

Lokaler Zustand
wart_ren_ans_nz

Der Requester hat einen Antrag auf Leihfristverlängerung zu einer bestimmten ILL-Transaktion an den Responder gesendet und wartet nun auf dessen Antwort.

23.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes RENEW-ANSWER

Sowohl beim Senden als auch beim Empfangen des ILL-Dienstes RENEW-ANSWER findet kein Zustandswechsel statt. Es werden lediglich sog. Parallelzustände in der Fernleihdatenbank generiert, die zur Kontrolle und Steuerung der Abhandlung der Antwort auf den Leihfristverlängerungs-Antrag benutzt werden.

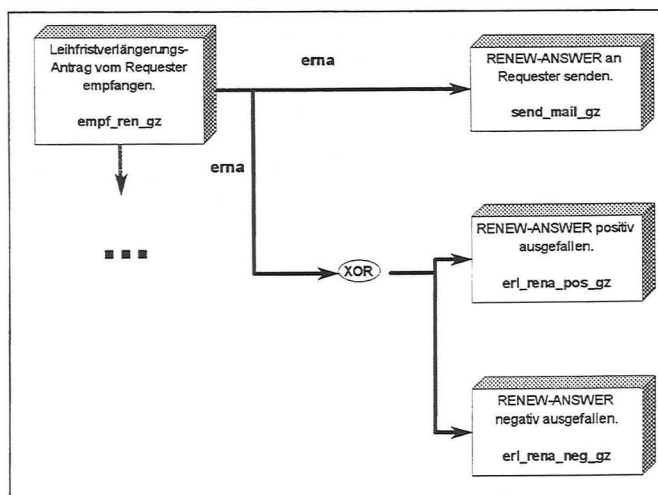
23.2.1 Zustandsübergänge beim Senden des ILL-Dienstes RENEW-ANSWER durch den Responder

Durch den Aufruf des Kommandos *erna* wird ein neuer Zustandssatz 'send_mail_gz' in der Fernleihdatenbank generiert, der in seiner Sekundärkennung den ILL-Dienst RENEW-ANSWER identifiziert. In diesem Satz ist die generelle Zustimmung oder Ablehnung des Leihfristverlängerungs-Antrages enthalten - ebenso eventuell das gewährte neue Leihfristdatum und der Hinweis, ob weitere Anträge zulässig sind. Bei Angabe eines gewährten neuen Leihfristdatums wird zusätzlich geprüft, ob das vom Responder angegebene neue Datum auch wirklich größer als das ursprüngliche Leihfristdatum ist. Ist dies der Fall, wird das neue Leihfristdatum in der Fernleihdatenbank des Responders aktualisiert.

Der generierte Satz wird dann vom Sendeserver verschickt und mit Enddatum und Endezeit versehen. Zusätzlich wird ein weiterer Parallelzustand generiert, der signalisiert, wie die Antwort auf den Leihfristverlängerungs-Antrag ausgefallen ist. Ist diese positiv ausgefallen, wird der Parallelzustand 'erl_rena_pos_gz' generiert, ansonsten der Zustand 'erl_rena_neg_gz'. Bei der Generierung dieser Zustände wird der Zustand, der den Empfang des Leihfristverlängerungs-Antrages signalisiert hat, abgeschlossen, d.h. er wird mit Enddatum und Endezeit versehen. Der ursprüngliche Zustand der ILL-Transaktion ('wart_return_gz') bleibt aber bestehen. Ebenso wird der weitere Verlauf der ILL-Transaktion durch die Generierung dieser Parallelzustände nicht beeinflusst.

In der folgenden Abbildung wird dieser Vorgang veranschaulicht:

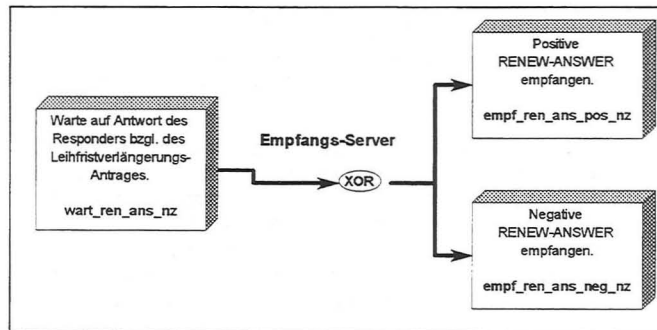
Abb. 23.2
Zustandsübergänge:
Senden
RENEW-ANSWER



23.2.2 Zustandsübergänge beim Empfangen des ILL-Dienstes RENEW-ANSWER beim Requester

Durch den Empfang des ILL-Dienstes RENEW-ANSWER bleibt der ursprüngliche Zustand der ILL-Transaktion ('benu_bereit_nz' oder 'benu_ausl_nz') auf der Seite des Requesters ebenfalls bestehen. Der Dienst kann nur empfangen werden, wenn zuvor ein Antrag auf Leihfristverlängerung zur selben ILL-Transaktion gesendet wurde, auf den bislang noch keine Antwort empfangen worden ist ('wart_ren_ans_nz'). Es wird auch hier ein Parallelzustand generiert, der den Empfang des Dienstes anzeigt. Dieser Zustand heißt je nachdem, ob die Antwort positiv oder negativ ausgefallen ist, entweder 'empf_ren_ans_pos_nz' oder 'empf_ren_ans_neg_nz'. Dieser Zustand bleibt solange offen, d.h. ohne Enddatum und Endezeit, bis eine weitere Antwort auf einen neuen Leihfristverlängerungs-Antrag zur selben ILL-Transaktion empfangen wurde. Ist die Antwort des Responders positiv ausgefallen, wird das gewährte neue Leihfristdatum in der Fernleihdatenbank des Requesters aktualisiert, falls es größer als das ursprüngliche Leihfristdatum ist.

Abb. 23.3
Zustandsübergänge:
Empfangen
RENEW-ANSWER



23.3 Die globalen ILL-Transaktionszustände

Zusätzlich zu den lokalen Zuständen existieren auch globale ILL-Transaktionszustände⁹⁶. Sie dienen zur Prüfung, ob ein Sender einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand senden darf und ob analog ein Empfänger einen ILL-Dienst in seinem aktuellen globalen ILL-Transaktionszustand empfangen darf.

Die folgende Tabelle zeigt, in welchen globalen Zuständen der ILL-Dienst RENEW-ANSWER gesendet bzw. empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

⁹⁶ Siehe Abschnitt 3.4.4.

Abb. 23.4
Globale ILL-
Transaktions-
zustände

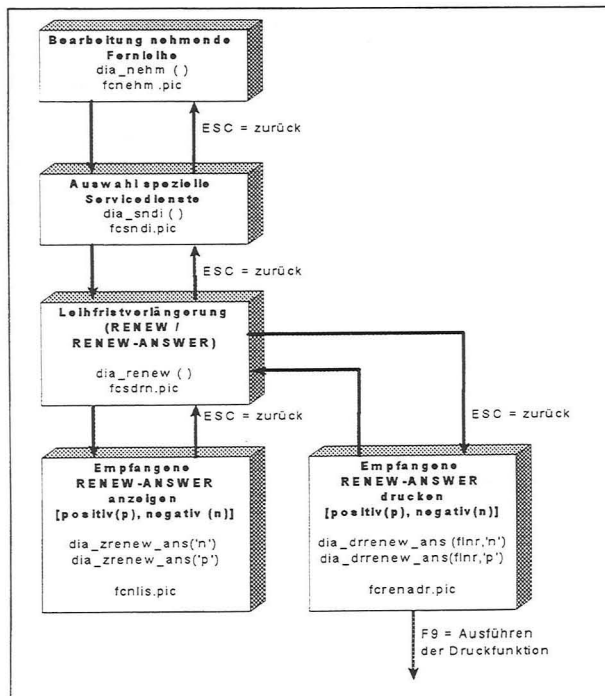
Requester (nehmende Bibliothek)	Responder (gebende Bibliothek)
RENEW-ANSWER empfangen	RENEW-ANSWER senden
	SHIPPED (*)
RENEW / PENDING	RENEW / PENDING
RENEW / OVERDUE	RENEW / OVERDUE
OVERDUE (*)	OVERDUE (**)
RECALL	RECALL
LOST	LOST
<u>FOLGEZUSTÄNDE:</u> RENEW / OVERDUE für (*)	<u>FOLGEZUSTÄNDE:</u> RENEW / PENDING für (*) RENEW / OVERDUE für (**)

23.4 Menüsteuerung beim Fernleih-Client

Die ILL-Dienste RENEW und RENEW-ANSWER sind auf Requester- und Responderseite in einem Menüpunkt zusammengefaßt. In der Menüsteuerung werden aber immer nur diejenigen (Unter-)Menüpunkte aufgeführt, die zu dem beschriebenen ILL-Dienst gehören.

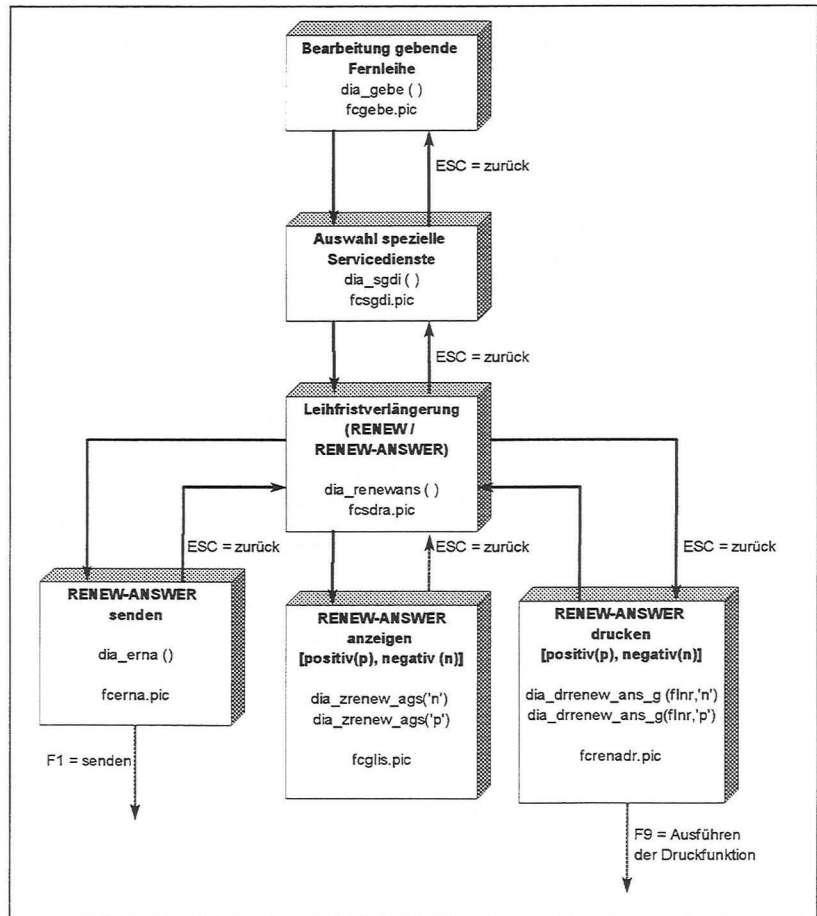
23.4.1 Nehmende Fernleihe (Requester)

Abb. 23.5
Menüsteuerung
RENEW-ANSWER
(Requester)



23.4.2 Gebende Fernleihe (Responder)

Abb. 23.6
Menüsteuerung
RENEW-ANSWER
(Responder)



24

LOST

Verwendung des
ILL-Dienstes LOST

Dieser Dienst wird benutzt, wenn ein ausgeliehenes Objekt entweder beim Requester (oder einem seiner Benutzer) oder auf dem Transportweg verloren gegangen ist. Dieser Dienst wird nur verwendet, wenn der Verlust des Objektes nachgewiesen ist. Wenn nur vermutet wird, daß das Objekt verloren gegangen ist, dann sollte der Dienst MESSAGE verwendet werden, um die andere Partei zu informieren. Ebenso wird der Dienst LOST auch nicht verwendet, wenn das Objekt beim Bearbeiten der eingegangenen Anfrage in der gebenden Bibliothek nicht auffindbar ist. In diesem Fall sollte der Dienst ILL-ANTWORT mit der Antwort UNFILLED (Das gewünschte Objekt ist nicht verfügbar) benutzt werden.

Beim Erfassen des ILL-Dienstes LOST kann ein Bemerkungstext von 160 Zeichen eingegeben werden. Diese Bemerkung kann zur genauen Beschreibung des Verlustes dienen, z.B. „Das Objekt ging beim Transport verloren“. Der Empfänger eines LOST kann diesen Text lesen und ausdrucken.

Wie der Ersatz des Objektes gehandhabt wird, steht den Partnerbibliotheken frei und geschieht getrennt vom Fernleihprogramm.

24.1

Senden und Empfangen des ILL-Dienstes LOST

Der ILL-Dienst LOST kann sowohl von der nehmenden als auch von der gebenden Bibliothek gesendet und empfangen werden. Das Senden und Empfangen eines LOST ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt in welchen lokalen Zuständen der ILL-Dienst LOST von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann:

Abb. 24.1
Lokale Zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
LOST senden	LOST empfangen	LOST senden	LOST empfangen
wart_illreq_nz	beendet_nz	wart_return_gz	wart_return_gz
benu_bereit_nz			
benu_ausl_nz			
ruecksend_nz			

24.1.1

Senden des ILL-Dienstes LOST durch den Requester

Der Requester kann den Dienst LOST senden, wenn sich der Vorgang in den folgenden vier lokalen Zuständen befindet:

Lokaler Zustand
wart_illreq_nz

Der Requester hat mit dem ILL-Dienst ILL-REQUEST beim Responder eine Fernleihanfrage gestellt. Er wartet nun auf Eingang des Objektes, ohne daß er

vom Responder durch einen SHIPPED mitgeteilt bekommen hat, daß das gewünschte Objekt versendet wurde. Da das Versenden des Dienstes SHIPPED durch die gebende Bibliothek nicht zwingend ist, kann die nehmende Bibliothek in diesem Fall den Verlust erst durch eines der folgenden Ereignisse bemerken:

- Der Responder teilt durch einen OVERDUE mit, daß das Objekt überfällig ist oder er fordert das Objekt durch einen RECALL vorzeitig zurück, ohne daß der Requester es jemals erhalten hat.
- Die Post informiert den Requester, daß das Objekt auf dem Transportweg zu ihm verloren ging.

Der andere Fall wäre, daß der Requester mit dem ILL-Dienst REQUEST beim Responder eine Fernleihanfrage gestellt hat und daraufhin durch den Dienst SHIPPED mitgeteilt bekommen hat, daß das gewünschte Objekt gesendet wurde. In diesem Fall kann eine Verlustanzeige gesendet werden, wenn folgende Ereignisse eingetreten sind:

- Nachdem das Objekt nach langer Wartezeit beim Requester nicht eingetroffen ist und ein Verlust des Objektes auf dem Transportweg zum Requester festgestellt wurde.
- Die Post informiert den Requester, daß das Objekt auf dem Transportweg zu ihm verloren ging.

Lokaler Zustand
benu_bereit_nz

Die nehmende Bibliothek hat das Objekt von der gebenden Bibliothek erhalten und die Eingangskontrolle durchgeführt. Der Requester informiert nun den Benutzer, daß sein gewünschtes Objekt eingetroffen ist und zur Abholung bereitsteht. Der Requester befindet sich somit im Zustand 'benu_bereit_nz'. Geht das Objekt nach Durchführung der Eingangskontrolle und vor der Abholung durch den Benutzers in der nehmenden Bibliothek verloren, so kann in diesem Zustand eine Verlustanzeige an die gebende Bibliothek gesendet werden.

Lokaler Zustand
benu_ausl_nz

Die nehmende Bibliothek hat das Objekt an den Benutzer ausgeliehen und wartet nun auf die Rückgabe durch den Benutzer. Sie befindet sich somit im Zustand 'benu_ausl_nz' und kann eine Verlustanzeige an die gebende Bibliothek senden, wenn der Benutzer der nehmenden Bibliothek mitteilt, daß er das Objekt verloren hat.

Lokaler Zustand
ruecksend_nz

Der Requester hat das Objekt vom Benutzer zurückerhalten und bearbeitet nun die Rücksendung an den Responder vor. Die nehmende Bibliothek befindet sich nun im Zustand 'ruecksend_nz'. Geht das Objekt während dieses Vorgangs in der nehmenden Bibliothek verloren, so sendet sie einen LOST an die gebende Bibliothek.

24.1.2

Empfangen des ILL-Dienstes LOST beim Responder

Lokaler Zustand
wart_return_gz

Der Responder kann einen LOST empfangen, wenn sich die Transaktion im lokalen Zustand 'wart_return_gz' befindet. Der Responder hat das gewünschte Objekt an die nehmende Bibliothek abgeschickt und wartet nun auf die Rückgabe des Objektes. Während dieser Zeitspanne kann der Responder eine Verlustanzeige empfangen, wenn das Objekt entweder auf dem Transportweg vom Responder zum Requester oder bei der nehmenden Bibliothek verloren geht und sie dies durch Senden eines LOST dem Responder mitteilt.

24.1.3 Senden des ILL-Dienstes LOST durch den Responder

Lokaler Zustand
wart_return_gz

Der Responder kann einen LOST senden, wenn sich der Vorgang im lokalen Zustand 'wart_return_gz' befindet. Der Responder hat das Objekt an den Requester verschickt und wartet nun auf die Rückgabe des Objekts. In diesem Zustand kann die gebende Bibliothek eine Verlustanzeige senden, wenn die nehmende Bibliothek durch den Dienst RETURNED anzeigt, daß sie das Objekt zurückgesendet hat, das Objekt aber auf dem Transportweg verloren geht und niemals beim Responder ankommt.

24.1.4 Empfangen des ILL-Dienstes LOST beim Requester

Lokaler Zustand
beendet_nz

Der Requester kann einen LOST empfangen, wenn sich die Transaktion im lokalen Zustand 'beendet_nz' befindet. Der Requester hat das Objekt an den Responder zurückgeschickt und dies durch den Dienst RETURNED dem Responder mitgeteilt. Mit dem Absenden des Objekts an den Responder ist die Transaktion auf der nehmenden Seite abgeschlossen und der Zustand 'beendet_nz' ist erreicht. In diesem Zustand kann die nehmende Bibliothek eine Verlustanzeige empfangen, wenn das Objekt auf dem Rückweg zur gebenden Bibliothek verloren ging und der Responder diesen Verlust durch einen LOST dem Requester mitteilt.

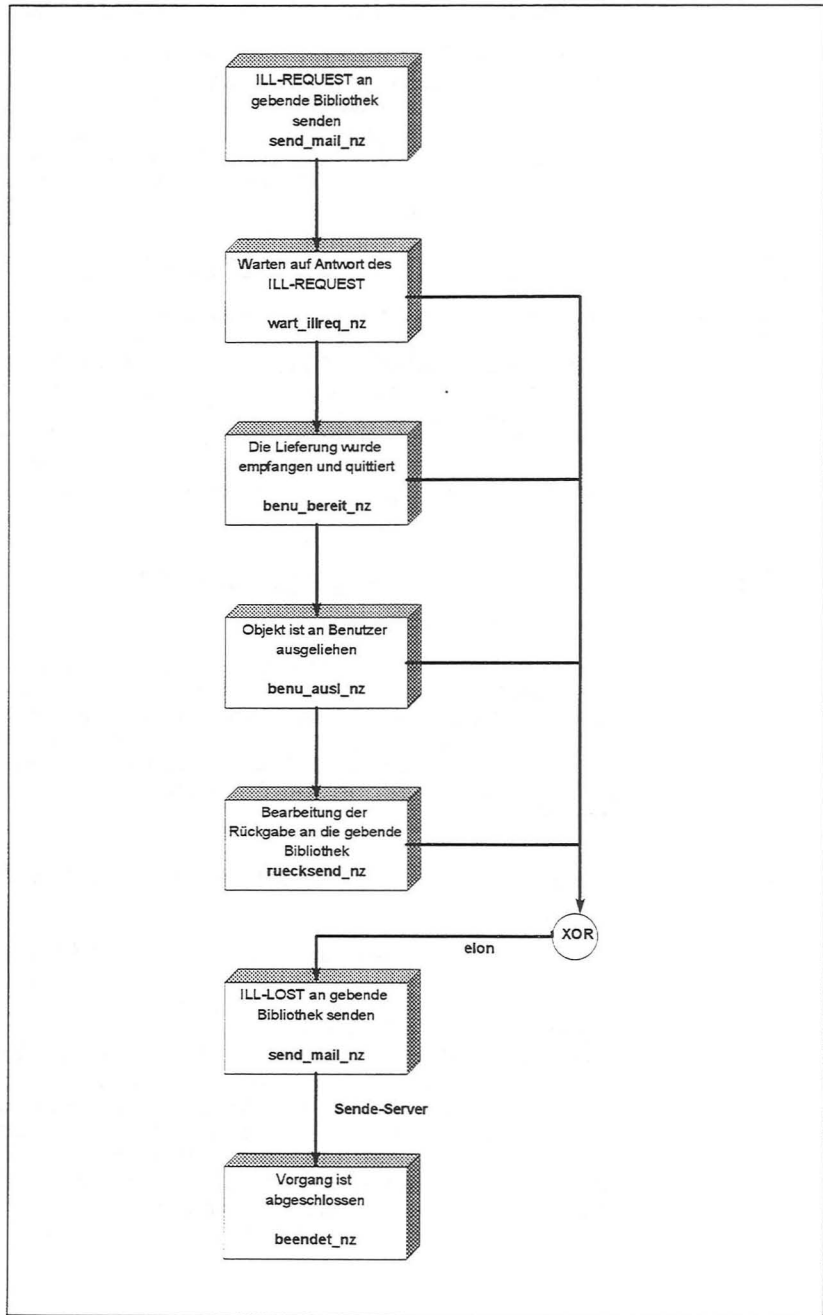
24.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes LOST

Sowohl beim Senden als auch beim Empfangen des Dienstes LOST findet ein Zustandswechsel statt. Dabei wird der aktuelle Zustand beendet, d.h. Systemdatum und -uhrzeit werden als Endedatum und Endezeit dem Zustand zugeordnet und in die Datenbank eingetragen. Mit dem Abschluß des aktuellen Zustandes wird sofort der nachfolgende Zustand generiert, bei dem Anfangsdatum und Anfangszeit gesetzt werden, Endedatum und -zeit aber bis zu seinem Abschluß offen bleiben.

24.2.1 Zustandsübergänge beim Senden des ILL-Dienstes LOST durch den Requester

Die nehmende Bibliothek kann den ILL-Dienst LOST senden, wenn sich die Transaktion im Zustand 'wart_illreq_nz', 'benu_bereit_nz', 'benu_ausl_nz' oder 'ruecksend_nz' befindet. Wünscht der Benutzer am Fernleih-Client das Versenden eines LOST, so sendet er das Kommando *elon* zum Fernleih-Server. Dieser ruft das Programm *feelon.c* auf, welches nach erfolgreicher Prüfung des lokalen Zustandes diesen abschließt und einen neuen Zustandssatz 'send_mail_nz' in der Fernleihdatenbank generiert. Somit wird der bisherige lokale Zustandssatz mit Ende-Zeit und Ende-Datum versehen und es wird der Folgezustand 'send_mail_nz' mit Sekundärkennung 'send_lost' erzeugt. Sobald der Sendeserver diesen sendefähigen Dienst verschickt hat, beendet er den Zustand 'send_mail_nz' und generiert den neuen Zustand 'beendet_nz', der die gesamte Transaktion auf der nehmenden Seite abschließt.

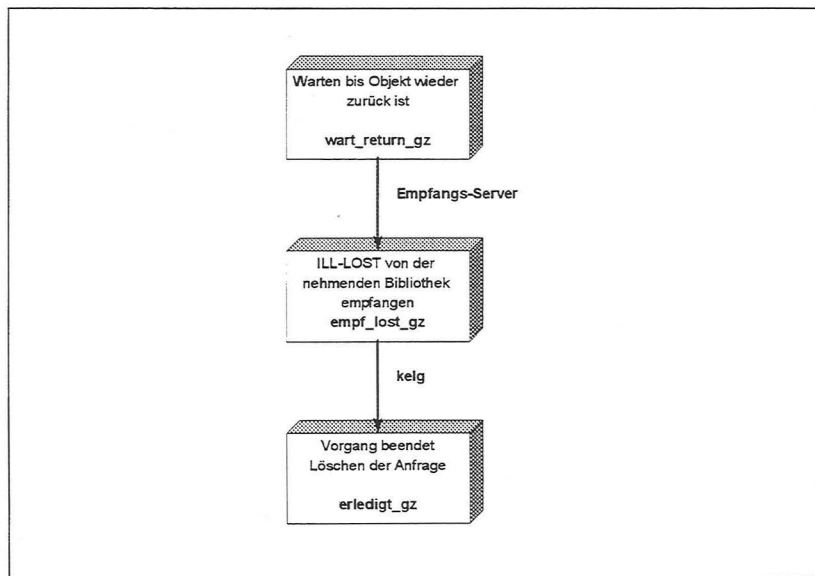
Abb. 24.2
Zustandsübergänge
beim Senden des
ILL-Dienstes LOST
(Requester)



24.2.2 Zustandsübergänge beim Empfangen des Dienstes LOST beim Responder

Der Responder kann den Dienst LOST nur empfangen, wenn sich der Vorgang augenblicklich im Zustand 'wart_return_gz' befindet. Der Empfangsserver überprüft, ob beim Eintreffen eines LOST dieser Zustand vorliegt, beendet ihn gegebenenfalls und generiert den Folgezustand 'empf_lost_gz'. Nachdem der Fernleih-Client den empfangenen LOST bearbeitet hat wird durch das Programm *fekekg.c* der Zustand 'empf_lost_gz' beendet und der neue Zustand 'erledigt_gz' erzeugt. Somit ist die Transaktion auf der gebenden Seite abgeschlossen.

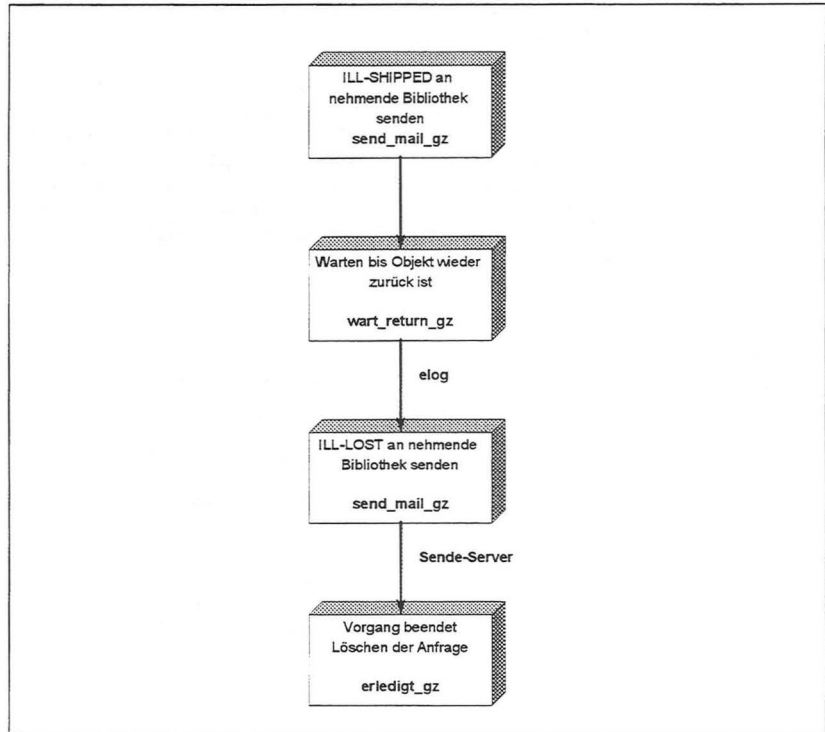
Abb. 24.3
Zustandsübergänge
beim Empfangen des
ILL-Dienstes LOST
(Responder)



24.2.3 Zustandsübergänge beim Senden des Dienstes LOST durch den Responder

Die gebende Bibliothek kann den Dienst LOST nur senden, wenn sich die Transaktion im aktuellen Zustand 'wart_return_gz' befindet. Dabei wird dieser Zustand vom Fernleih-Server durch das Programm *feelog.c* beendet und der nachfolgende Zustand 'send_mail_gz' generiert, der einen sendefähigen ILL-LOST erzeugt. Der Sendeserver versendet den Dienst, schließt den Zustand 'send_mail_gz' ab und generiert den neuen Zustand 'erledigt_gz'. Somit ist die gesamte Transaktion auf der gebenden Seite abgeschlossen.

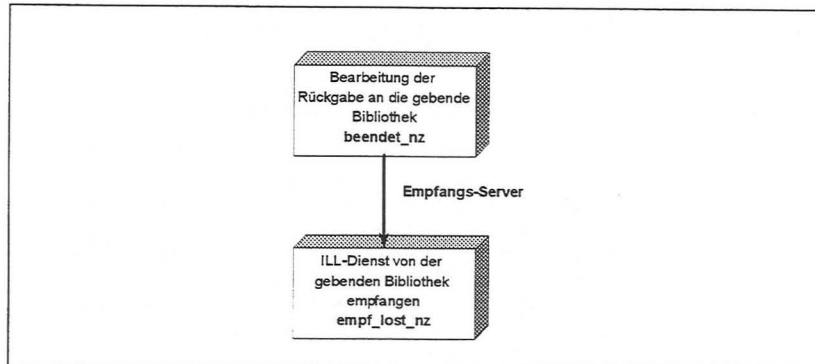
Abb. 24.4
Zustandsübergänge
beim Senden des
ILL-Dienstes LOST
(Responder)



24.2.4 Zustandsübergänge beim Empfangen des Dienstes LOST beim Requester

Der Requester kann einen LOST nur empfangen, wenn sich der Vorgang im aktuellen Zustand 'beendet_nz' befindet. Der Empfangsserver der nehmenden Bibliothek generiert beim Erhalt des LOST den neuen Zustand 'empf_lost_nz'. In diesem Zustand wird solange verweilt, bis der Fernleih-Client die Verlustanzeige bearbeitet hat. Danach wird der Zustand 'empf_lost_nz' durch das Programm *fekeln.c* beendet. Da bei dieser Konstellation der Endzustand 'beendet_nz' bereits vor Eintreffen des LOST erreicht wurde, wird hier kein neuer Zustand generiert, sondern lediglich 'empf_lost_nz' beendet.

Abb. 24.5
Zustandsübergänge
beim Empfangen des
ILL-Dienstes LOST
(Requester)



24.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind also für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Dabei spielt im lokalen Prozeß der Zustand der Partnerbibliothek keine Rolle.

Erst beim Senden und Empfangen von Diensten werden die Zustände der Partnerbibliothek wichtig. Daher existieren zusätzlich zu den lokalen Zuständen auch globale ILL-Transaktionszustände⁹⁷. Sie dienen zur Prüfung, ob ein Sender einen Dienst in seinem aktuellen globalen Zustand senden darf und ob ein Empfänger einen Dienst in seinem aktuellen globalen Zustand empfangen darf.

Die folgende Tabelle zeigt in welchen globalen Zuständen ein LOST von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 24.6
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
LOST senden	LOST empfangen	LOST senden	LOST empfangen
PENDING			
CANCEL-PENDING			
SHIPPED		SHIPPED	SHIPPED
RECEIVED			
RENEW/PENDING			RENEW/PENDING
RENEW/OVERDUE			RENEW/OVERDUE
OVERDUE			OVERDUE
NOT-RECEIVED/ OVERDUE			
RECALL			RECALL
	RETURNED		
Folgezustand: LOST	Folgezustand: LOST	Folgezustand: LOST	Folgezustand: LOST

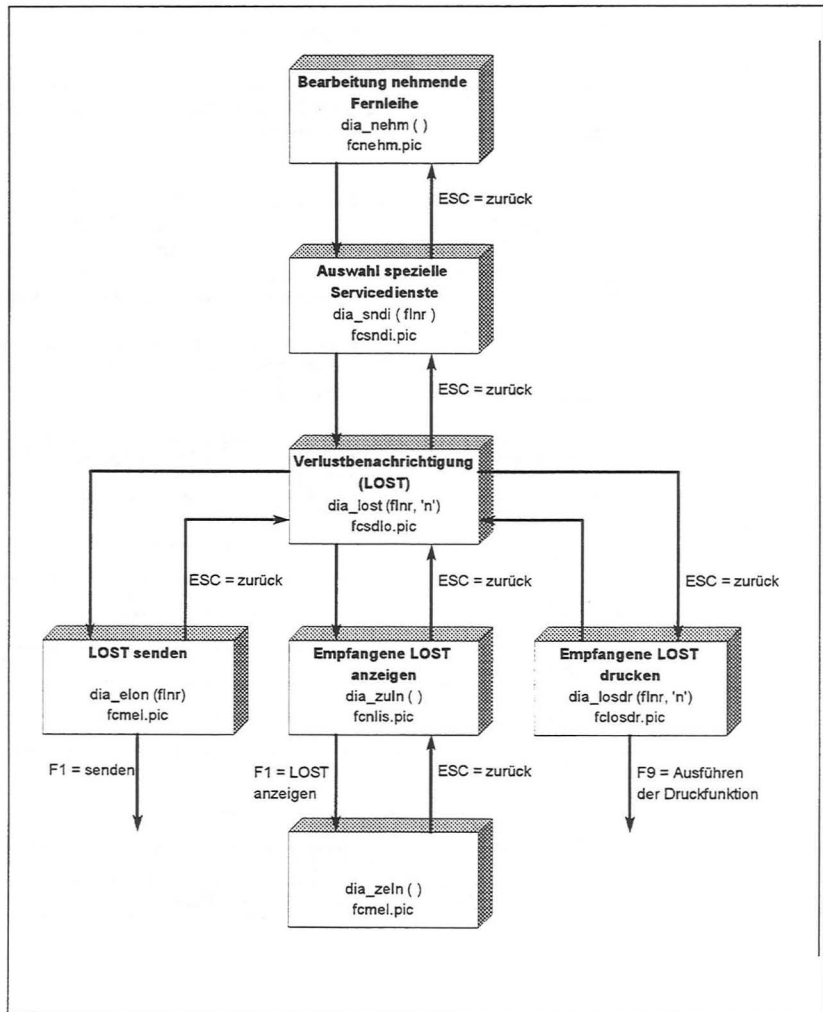
⁹⁷ Siehe Abschnitt 3.4.4.

24.4 Menüsteuerung beim Fernleih-Client

24.4.1 Nehmende Fernleihe (Requester)

In der nehmenden Bearbeitung kann man zum Menü des ILL-Dienstes LOST gelangen, indem man im Hauptmenü den Punkt „Bearbeitung nehmende Fernleihe“ und im darauffolgenden Untermenü den Punkt „Auswahl spezielle Servicedienste“ wählt. Unter allen Servicediensten, die dann zur Auswahl stehen, findet man auch den Dienst Verlustbenachrichtigung (LOST). Wird der Menüpunkt LOST gewählt, so wird die Funktion 'dia_lost' aufgerufen, die das Bild 'fcsdlo.pic' öffnet. Das Menü des ILL-Dienstes LOST ist wiederum unterteilt in die Untermenüs „LOST senden“, „Empfangene LOST anzeigen“ und „Empfangene LOST drucken“.

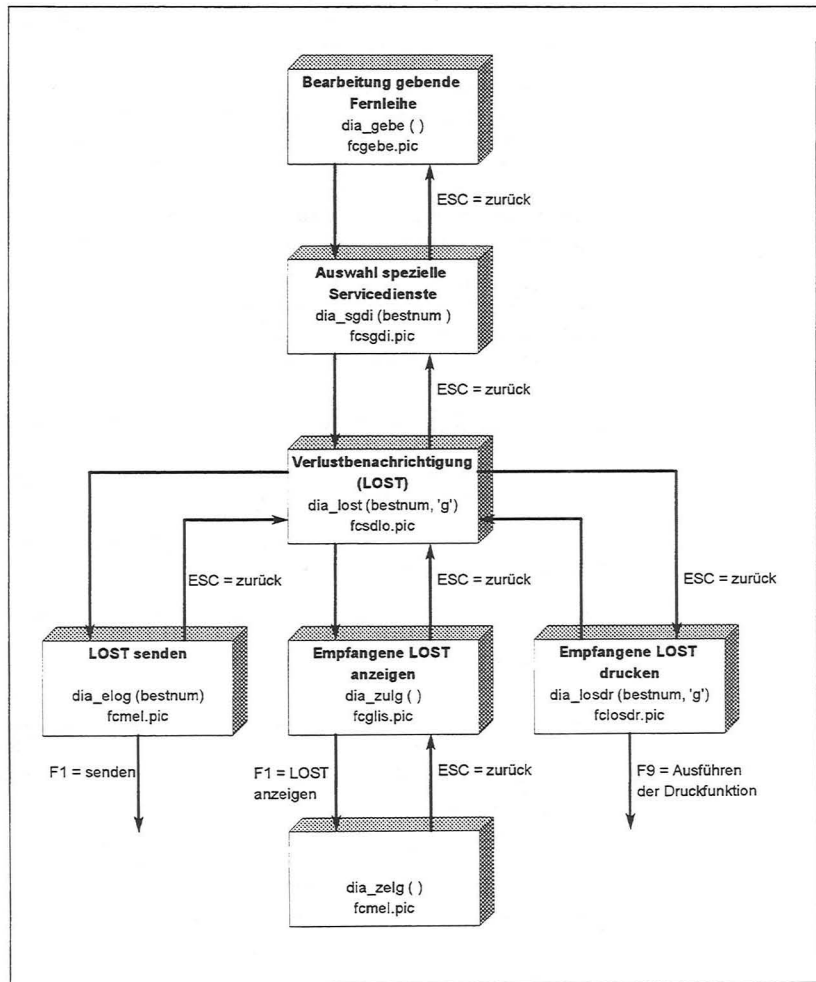
Abb. 24.7
Menü LOST
(Requester)



24.4.2 Gebende Fernleihe (Responder)

In der gebenden Bearbeitung gelang man analog zur nehmenden Bearbeitung zum ILL-Dienst LOST. Allerdings muß hier im Hauptmenü die Verzweigung „Bearbeitung gebende Fernleihe“ gewählt werden. Die Dialogfunktionen der gebenden Bearbeitung unterscheiden sich zu denen der nehmenden Bearbeitung lediglich dadurch, daß sie nicht mit der Fernleihnummer, sondern mit der Bestellnummer arbeiten und daß sie Zugriffe auf die Datenbanktabellen der gebenden Fernleihe realisieren.

Abb. 24.8
Menü LOST
(Responder)



25 DAMAGED

Verwendung des ILL-Dienstes DAMAGED

Dieser Dienst wird benutzt, wenn ein ausgeliehenes Objekt entweder beim Requester (oder einem seiner Benutzer) oder auf dem Transportweg beschädigt worden ist. Der Teilnehmer der ILL-Transaktion, der die Beschädigung entdeckt, sollte den anderen Teilnehmer sofort informieren. Somit kann bei Ersatzansprüchen, die Schuldfrage leichter nachvollzogen werden. Ob und wie der Ersatz des Objektes gehandhabt wird, steht den Partnerbibliotheken frei und wird getrennt vom Fernleihprogramm geklärt.

Durch die Beschädigung eines Objektes ist die ILL-Transaktion nicht beendet, da in den meisten Fällen das Objekt noch seinen Zweck erfüllt, z.B. wenn sich nur eine Seite gelöst hat. Der Vorgang wird in diesem Fall fortgeführt. Wenn das Objekt so beschädigt ist, daß es nicht mehr verwendet werden kann, wird die ILL-Transaktion ebenfalls nicht sofort nach Entdeckung der Beschädigung beendet. Der Vorgang wird auch in diesem Fall auf die übliche Art und Weise abgeschlossen, indem die gebende Bibliothek das Objekt zurückbucht.

Beim Erfassen des ILL-Dienstes DAMAGED kann ein Bemerkungstext von 160 Zeichen eingegeben werden. Diese Bemerkung kann zur genauen Beschreibung der Beschädigung dienen, z.B. „Es fehlen die Seiten 365-378“. Der Empfänger eines DAMAGED kann diesen Text lesen und ausdrucken.

25.1 Senden und Empfangen des ILL-Dienstes DAMAGED

Der ILL-Dienst DAMAGED kann sowohl von der nehmenden als auch von der gebenden Bibliothek gesendet und empfangen werden. Das Senden und Empfangen eines DAMAGED ist jedoch nur in bestimmten Fällen möglich und sinnvoll. Diese Zeitpunkte können anhand der lokal definierten Zustände einer ILL-Transaktion eindeutig bestimmt werden.

Die folgende Tabelle zeigt in welchen lokalen Zuständen der ILL-Dienst DAMAGED von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann:

Abb. 25.1 Lokale Zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
DAMAGED senden	DAMAGED empfangen	DAMAGED senden	DAMAGED empfangen
wart_illreq_nz	beendet_nz	wart_return_gz	wart_return_gz
benu_bereit_nz			
benu_ausl_nz			
ruecksend_nz			

Der ILL-Dienst DAMAGED kann in den erlaubten lokalen Zuständen beliebig oft gesendet und empfangen werden.

25.1.1

Senden des ILL-Dienstes DAMAGED durch den Requester

Der Requester kann den Dienst DAMAGED senden, wenn sich der Vorgang in den folgenden vier lokalen Zuständen befindet:

Lokaler Zustand
wart_illreq_nz

Der Requester hat mit dem Dienst ILL-REQUEST beim Responder eine Fernleihanfrage gestellt. Er befindet sich nun im Zustand 'wart_illreq_nz' und wartet auf den Eingang des Objektes. Sobald der Requester das Objekt von der gebenden Bibliothek erhalten hat, führt er eine Eingangskontrolle durch. Stellt er dabei eine Beschädigung des Objektes fest, so kann er den Responder durch den ILL-Dienst DAMAGED davon unterrichten. In diesem Fall gibt es zwei Möglichkeiten für den Zeitraum der Beschädigung:

- Die Beschädigung erfolgte auf dem Transportweg von der gebenden zur nehmenden Bibliothek.
- Die Beschädigung erfolgte in der Zeitspanne zwischen dem Eintreffen des Objektes beim Requester und der Durchführung der Eingangskontrolle

Lokaler Zustand
benu_bereit_nz

Die nehmende Bibliothek hat das Objekt von der gebenden Bibliothek erhalten und die Eingangskontrolle durchgeführt. Der Requester informiert nun den Benutzer, daß sein gewünschtes Objekt eingetroffen ist und zur Abholung bereitsteht. Der Requester befindet sich nun im lokalen Zustand 'benu_bereit_nz'. Erleidet das Objekt nach Durchführung der Eingangskontrolle und vor der Abholung durch den Benutzer in der nehmenden Bibliothek Schaden, so kann in diesem Zustand eine Beschädigungsanzeige an die gebende Bibliothek gesendet werden.

Lokaler Zustand
benu_ausl_nz

Die nehmende Bibliothek hat das Objekt an den Benutzer ausgeliehen und wartet nun auf die Rückgabe durch den Benutzer. Sie befindet sich somit im Zustand 'benu_ausl_nz' und kann eine Beschädigungsanzeige an die gebende Bibliothek senden, wenn der Benutzer der nehmenden Bibliothek mitteilt, daß das Objekt beschädigt ist.

Lokaler Zustand
ruecksend_nz

Der Requester hat das Objekt vom Benutzer zurückerhalten und bearbeitet nun die Rücksendung an den Responder vor. Die nehmende Bibliothek befindet sich nun im Zustand 'ruecksend_nz'. Wird das Objekt während dieses Vorgangs beschädigt, so kann sie einen DAMAGED an die gebende Bibliothek senden.

25.1.2

Empfangen des ILL-Dienstes DAMAGED beim Responder

Lokaler Zustand
wart_return_gz

Der Responder kann einen DAMAGED empfangen, wenn sich die Transaktion im lokalen Zustand 'wart_return_gz' befindet. Der Responder hat das gewünschte Objekt an die nehmende Bibliothek abgeschickt und wartet nun auf Rückgabe des Objektes. Während dieser Zeitspanne kann der Responder eine Beschädigungsanzeige empfangen, wenn das Objekt entweder auf dem Transportweg vom Responder zum Requester oder bei der nehmenden Bibliothek beschädigt wurde und der Requester dies durch Senden eines DAMAGED dem Responder mitteilt.

25.1.3

Lokaler Zustand
wart_return_gz

Senden des ILL-Dienstes DAMAGED durch den Responder

Der Responder kann einen DAMAGED senden, wenn sich der Vorgang im lokalen Zustand 'wart_return_gz' befindet. Der Responder hat das Objekt an den Requester verschickt und wartet nun auf die Rückgabe des Objekts. Nachdem die gebende Bibliothek das Objekt zurückerhalten hat, führt sie eine Eingangskontrolle durch. Stellt sie dabei eine Beschädigung des Objektes fest, so kann sie den Requester durch den ILL-Dienst DAMAGED davon unterrichten. In diesem Fall gibt es zwei Möglichkeiten für den Zeitraum der Beschädigung:

- Die Beschädigung erfolgte auf dem Transportweg von der nehmenden zur gebenden Bibliothek.
- Die Beschädigung erfolgte in der Zeitspanne zwischen dem Eintreffen des Objektes beim Responder und der Durchführung der Eingangskontrolle.

25.1.4

Lokaler Zustand
beendet_nz

Empfangen des ILL-Dienstes DAMAGED beim Requester

Der Requester kann einen DAMAGED empfangen, wenn sich die Transaktion im lokalen Zustand 'beendet_nz' befindet. Der Requester hat das Objekt an den Responder zurückgeschickt und dies durch den Dienst RETURNED dem Responder mitgeteilt. Mit dem Absenden des Objekts an den Responder ist die Transaktion auf der nehmenden Seite abgeschlossen und der Zustand 'beendet_nz' ist erreicht. In diesem Zustand kann die nehmende Bibliothek eine Beschädigungsanzeige empfangen, wenn das Objekt auf dem Rückweg zur gebenden Bibliothek beschädigt wurde und der Responder dies durch einen DAMAGED dem Requester mitteilt.

25.2

Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes DAMAGED

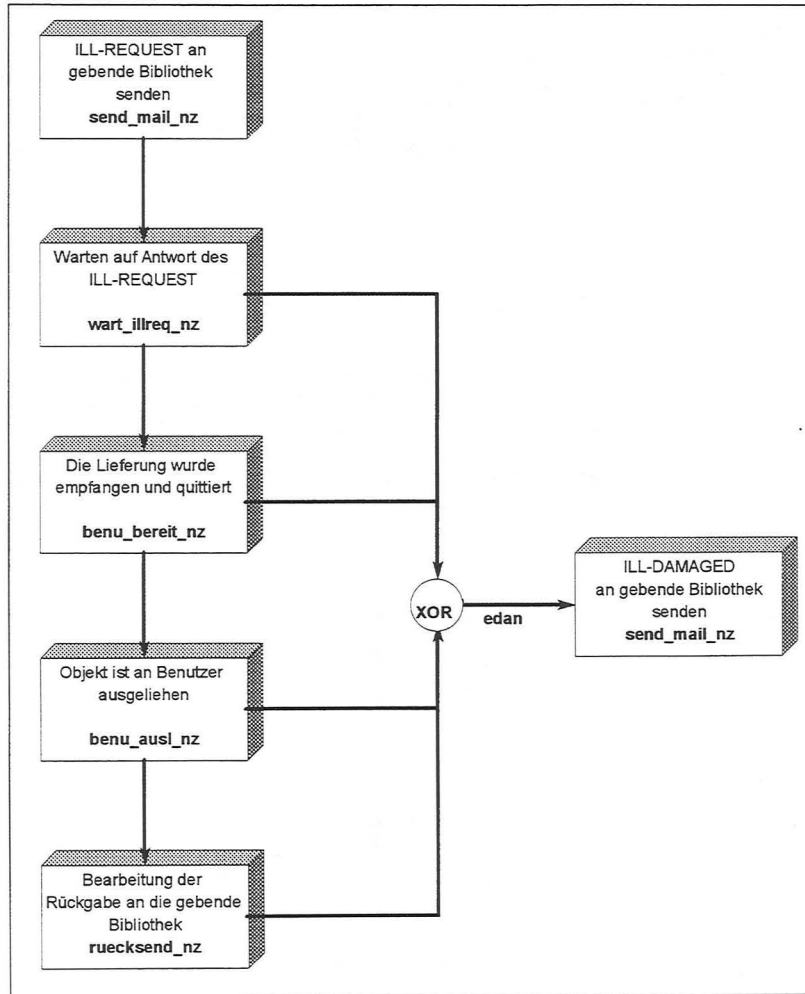
Sowohl beim Senden als auch beim Empfangen des Dienstes DAMAGED findet kein Zustandswechsel statt. Es werden lediglich sog. Parallelzustände in der Fernleihdatenbank generiert, die das Versenden bzw. den Empfang des ILL-Dienstes DAMAGED quittieren.

25.2.1

Zustandsübergänge beim Senden des ILL-Dienstes DAMAGED durch den Requester

Die nehmende Bibliothek kann den ILL-Dienst DAMAGED nur senden, wenn sich die Transaktion in den Zuständen 'wart_illreq_nz', 'benu_bereit_nz', 'benu_ausl_nz' oder 'ruecksend_nz' befindet. Wünscht der Benutzer am Fernleih-Client das Versenden eines DAMAGED, so sendet er das Kommando *edan* zum Fernleih-Server. Dieser ruft das Programm *feedan.c* auf, welches nach erfolgreicher Prüfung des lokalen Zustandes einen neuen Zustandssatz 'send_mail_nz' in der Fernleihdatenbank generiert. Somit bleibt der bisherige lokale Zustand bestehen und es wird parallel der Zustand 'send_mail_nz' mit Senkundärkennung 'send_damaged' erzeugt. Sobald der Sendeserver diesen sendefähigen Dienst verschickt hat, beendet er den generierten Satz, indem er ihn mit Endedatum und Endezeit versieht.

Abb. 25.2
Zustandsübergänge
beim Senden des
Dienstes DAMAGED
(Requester)

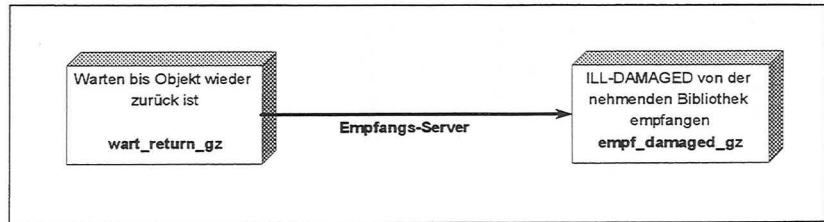


25.2.2 Zustandsübergänge beim Empfangen des ILL-Dienstes DAMAGED beim Responder

Der Responder kann den Dienst DAMAGED nur empfangen, wenn sich der Vorgang augenblicklich im Zustand 'wart_return_gz' befindet. Der Empfangs-server überprüft, ob beim Eintreffen eines DAMAGED dieser Zustand vorliegt und generiert den Zustand 'empf_damaged_gz'. Nachdem der Benutzer am Fernleih-Client den empfangenen DAMAGED bearbeitet hat, wird durch das Programm *fekedg.c* der Zustand 'empf_damaged_gz' beendet.

DAMAGED

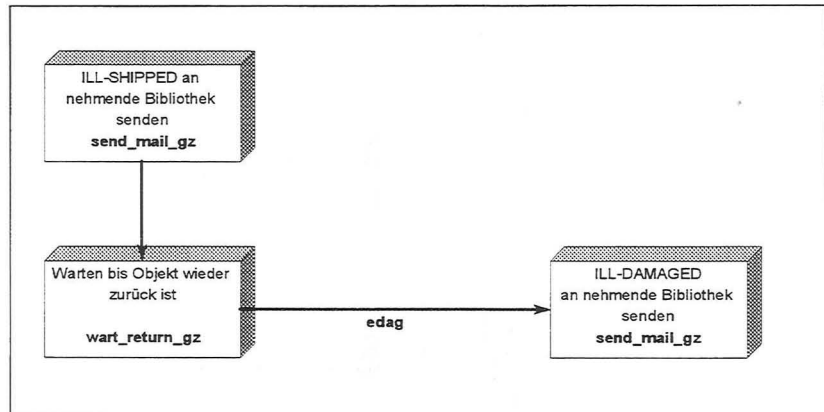
Abb. 25.3
Zustandsübergänge
beim Empfangen des
Dienstes DAMAGED
(Responder)



25.2.3 Zustandsübergänge beim Senden des ILL-Dienstes DAMAGED durch den Responder

Die gebende Bibliothek kann den Dienst DAMAGED nur senden, wenn sich die Transaktion im aktuellen Zustand 'wart_return_gz' befindet. Durch das Programm *feedag.c* wird der Zustand 'send_mail_gz' mit Sekundärkennung 'send damaged' generiert, der einen sendefähigen ILL-DAMAGED erzeugt. Der Sendeserver versendet den Dienst und schließt den Zustandssatz 'send_mail_gz' danach ab.

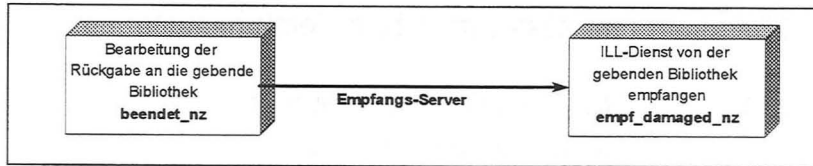
Abb. 25.4
Zustandsübergänge
beim Senden des
Dienstes DAMAGED
(Responder)



25.2.4 Zustandsübergänge beim Empfangen des ILL-Dienstes DAMAGED beim Requester

Der Requester kann einen DAMAGED nur empfangen, wenn sich der Vorgang im aktuellen Zustand 'beendet_nz' befindet. Der Empfangsserver der nehmenden Bibliothek generiert beim Erhalt des DAMAGED den neuen Zustand 'empf_damaged_nz'. Sobald der Benutzer die Beschädigungsanzeige bearbeitet hat, wird der Zustand 'empf_damaged_nz' durch das Programm *fekedn.c* beendet.

Abb. 25.5
Zustandsübergänge
beim Empfangen des
Dienstes DAMAGED
(Requester)



25.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind also für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Dabei spielt im lokalen Prozeß der Zustand der Partnerbibliothek keine Rolle.

Erst beim Senden und Empfangen von Diensten werden die Zustände der Partnerbibliothek wichtig. Daher existieren zusätzlich zu den lokalen Zuständen auch globale ILL-Transaktionszustände⁹⁸. Sie dienen zur Prüfung, ob ein Sender einen Dienst in seinem aktuellen globalen Zustand senden darf und ob ein Empfänger einen Dienst in seinem aktuellen globalen Zustand empfangen darf.

Die folgende Tabelle zeigt in welchen globalen Zuständen ein DAMAGED von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann und welches in allen Fällen der resultierende Folgezustand ist:

Abb. 25.6
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
DAMAGED senden	DAMAGED empfangen	DAMAGED senden	DAMAGED empfangen
PENDING			
CANCEL-PENDING			
SHIPPED		SHIPPED	SHIPPED
RECEIVED			
RENEW/PENDING			RENEW/PENDING
RENEW/OVERDUE			RENEW/OVERDUE
OVERDUE			OVERDUE
NOT-RECEIVED/ OVERDUE			
RECALL			RECALL
	RETURNED		
Folgezustand: DAMAGED	Folgezustand: DAMAGED	Folgezustand: DAMAGED	Folgezustand: DAMAGED

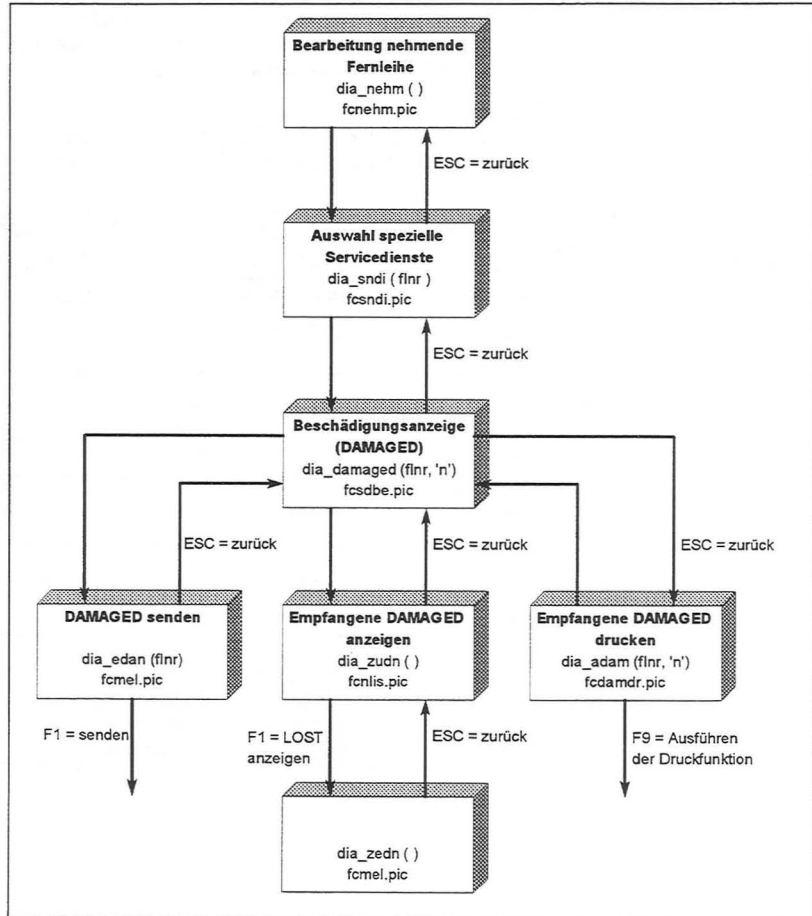
⁹⁸ Siehe Abschnitt 3.4.4.

25.4 Menüsteuerung beim Fernleih-Client

25.4.1 Nehmende Fernleihe (Requester)

In der nehmenden Bearbeitung kann man zum Menü des ILL-Dienstes DAMAGED gelangen, indem man im Hauptmenü den Punkt „Bearbeitung nehmende Fernleihe“ und im darauffolgenden Untermenü den Punkt „Auswahl spezielle Servicedienste“ wählt. Unter allen Servicediensten, die dann zur Auswahl stehen, findet man auch den Dienst Beschädigungsanzeige (DAMAGED). Wird der Menüpunkt DAMAGED gewählt, so wird die Funktion 'dia damaged' aufgerufen, die das Bild 'fcsdbe.pic' öffnet. Das Menü des ILL-Dienstes DAMAGED ist wiederum unterteilt in die Untermenüs „DAMAGED senden“, „Empfangene DAMAGED anzeigen“ und „Empfangene DAMAGED drucken“.

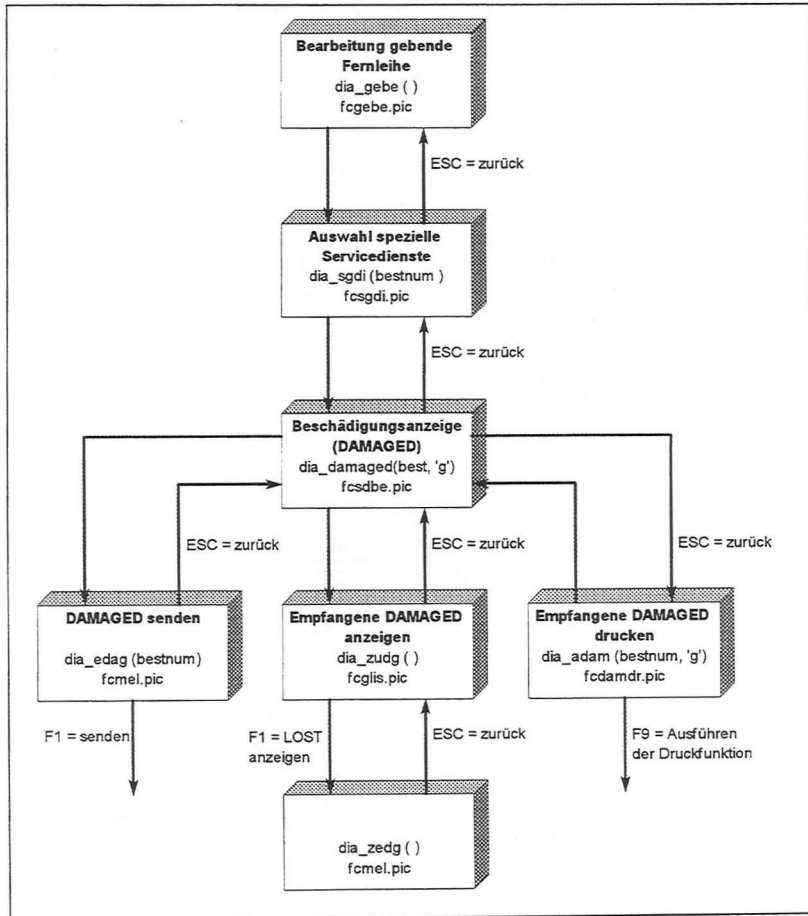
Abb. 25.7
Menü DAMAGED
(Requester)



25.4.2 Gebende Fernleihe (Responder)

In der gebenden Bearbeitung gelang man analog zur nehmenden Bearbeitung zum ILL-Dienst DAMAGED. Allerdings muß hier im Hauptmenü die Verzweigung „Bearbeitung gebende Fernleihe“ gewählt werden. Die Dialogfunktionen der gebenden Bearbeitung unterscheiden sich zu denen der nehmenden Bearbeitung lediglich dadurch, daß sie nicht mit der Fernleihnummer, sondern mit der Bestellnummer arbeiten und daß sie Zugriffe auf die Datenbanktabellen der gebenden Fernleihe realisieren.

Abb. 25.8
Menü DAMAGED
(Responder)



26 MESSAGE

Verwendung des ILL-Dienstes MESSAGE

Dieser Dienst erlaubt dem ILL-Benutzer jederzeit formlose Textnachrichten zu einer existierenden ILL-Transaktion zu versenden. Mit dem Dienst MESSAGE können Informationen ausgetauscht werden, die normalerweise nicht durch andere Dienste übermittelt werden können. Allerdings sollte der Dienst MESSAGE nicht als Mail-System mißbraucht werden.

Beim Erfassen des ILL-Dienstes MESSAGE kann ein Text von 160 Zeichen eingegeben werden. Der Empfänger einer formlosen Nachricht kann deren Text lesen und ausdrucken.

26.1 Senden und Empfangen des ILL-Dienstes MESSAGE

Der ILL-Dienst MESSAGE kann sowohl von der nehmenden als auch von der gebenden Bibliothek gesendet und empfangen werden. Das Senden und Empfangen einer formlosen Nachricht zu einer ILL-Transaktion ist jederzeit möglich. Eine ILL-Transaktion existiert dann, wenn die nehmende Bibliothek den Dienst ILL-REQUEST abgeschickt hat und die gebende Bibliothek diesen erhalten hat.

Die folgende Tabelle zeigt in welchen lokalen Zuständen der ILL-Dienst MESSAGE von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann:

Abb. 26.1 Lokale Zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
MESSAGE senden	MESSAGE empfangen	MESSAGE senden	MESSAGE empfangen
wart_illreq_nz	wart_illreq_nz	anfangs_gz	anfangs_gz
alle logisch folgenden Zustände nach dem Zustand wart_illreq_nz	alle logisch folgenden Zustände nach dem Zustand wart_illreq_nz	alle logisch folgenden Zustände nach dem Zustand anfangs_gz	alle logisch folgenden Zustände nach dem Zustand anfangs_gz

Der ILL-Dienst MESSAGE kann in den erlaubten lokalen Zuständen beliebig oft gesendet und empfangen werden.

26.1.1 Senden und Empfangen des ILL-Dienstes MESSAGE durch den Requester

Sobald der Requester den Dienst ILL-REQUEST an den Responder abgeschickt hat befindet er sich im lokalen Zustand 'wart_illreq_nz' und kann somit eine formlose Nachricht an die gebende Bibliothek senden oder von ihr empfangen. In allen Zuständen, die nach dem Zustand 'wart_illreq_nz' bis zum Abschluß der ILL-Transaktion in der nehmenden Bearbeitung noch folgen, ist das Senden und Empfangen des ILL-Dienstes MESSAGE erlaubt.

26.1.2 Senden und Empfangen des ILL-Dienstes MESSAGE durch den Responder

Sobald der Responder den Dienst ILL-REQUEST des Requesters erhalten hat, befindet er sich im lokalen Zustand 'anfangs_gz' und kann somit eine formlose Nachricht an die gebende Bibliothek senden oder von ihr empfangen. In allen Zuständen, die nach dem Zustand 'anfangs_gz' bis zum Abschluß der ILL-Transaktion in der gebenden Bearbeitung noch folgen, ist das Senden und Empfangen des ILL-Dienstes MESSAGE erlaubt.

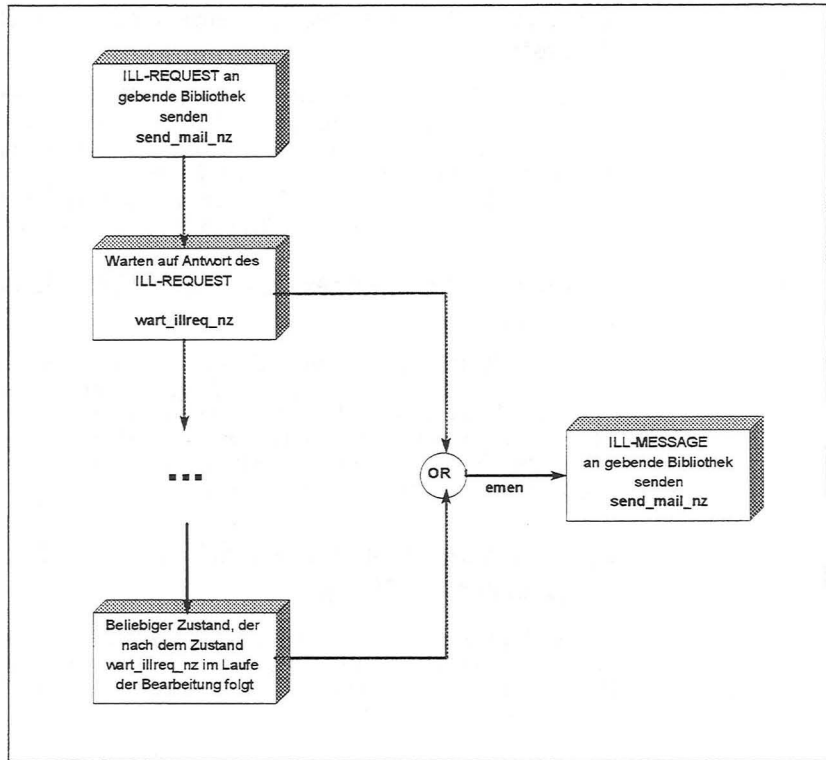
26.2 Lokale Zustandsübergänge beim Senden und Empfangen des ILL-Dienstes MESSAGE

Sowohl beim Senden als auch beim Empfangen des Dienstes MESSAGE findet kein Zustandswechsel statt. Es werden lediglich sog. Parallelzustände in der Fernleihdatenbank generiert, die das Versenden bzw. den Empfang des ILL-Dienstes MESSAGE quittieren.

26.2.1 Zustandsübergänge beim Senden des ILL-Dienstes MESSAGE durch den Requester

Die nehmende Bibliothek kann den ILL-Dienst MESSAGE senden, sobald eine ILL-Transaktion existiert. Wünscht der Benutzer am Fernleih-Client das Versenden des Dienstes MESSAGE, so sendet er das Kommando *emen* zum Fernleih-Server. Dieser ruft das Programm *feemen.c* auf, welches nach erfolgreicher Prüfung des lokalen Zustandes einen neuen Zustandssatz 'send_mail_nz' in der Fernleihdatenbank generiert. Somit bleibt der bisherige lokale Zustand bestehen und es wird parallel der Zustand 'send_mail_nz' mit Sekundärkennung 'send_message' erzeugt. Sobald der Sendeserver diesen sendefähigen Dienst verschickt hat, beendet er den generierten Satz, indem er ihn mit Enddatum und Endezeit versieht.

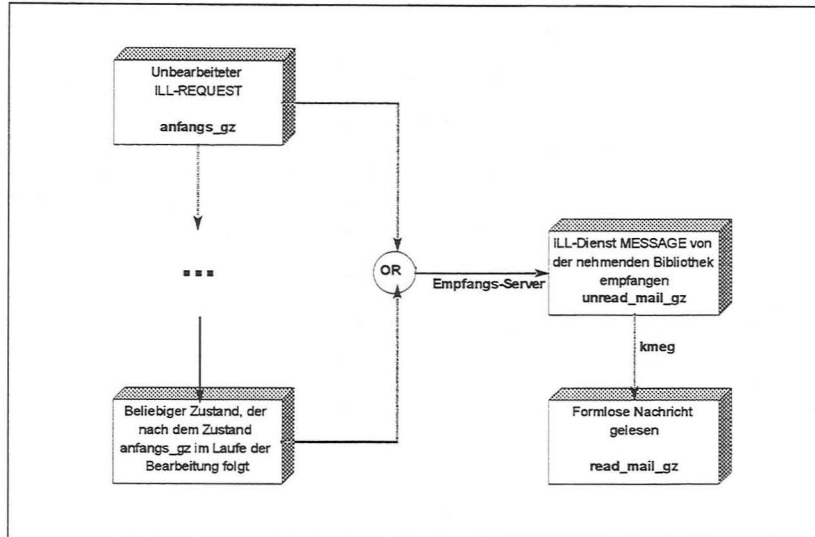
Abb. 26.2
Zustandsübergänge
beim Senden des
Dienstes MESSAGE
(Requester)



26.2.2 Zustandsübergänge beim Empfangen des ILL-Dienstes MESSAGE beim Responder

Der Responder kann den Dienst MESSAGE nur empfangen, wenn eine ILL-Transaktion existiert. Der Empfangsserver überprüft beim Eintreffen einer formlosen Nachricht, ob ein entsprechender Vorgang vorhanden ist und generiert den Zustand 'unread_mail_gz'. Nachdem der Fernleih-Client die empfangene Nachricht bearbeitet hat, wird durch das Programm *fekmeg.c* der Zustand des entsprechenden Datensatzes in der Fernleihdatenbank von 'unread_mail_gz' auf 'read_mail_gz' gesetzt.

Abb. 26.3
Zustandsübergänge
beim Empfangen des
Dienstes MESSAGE
(Responder)

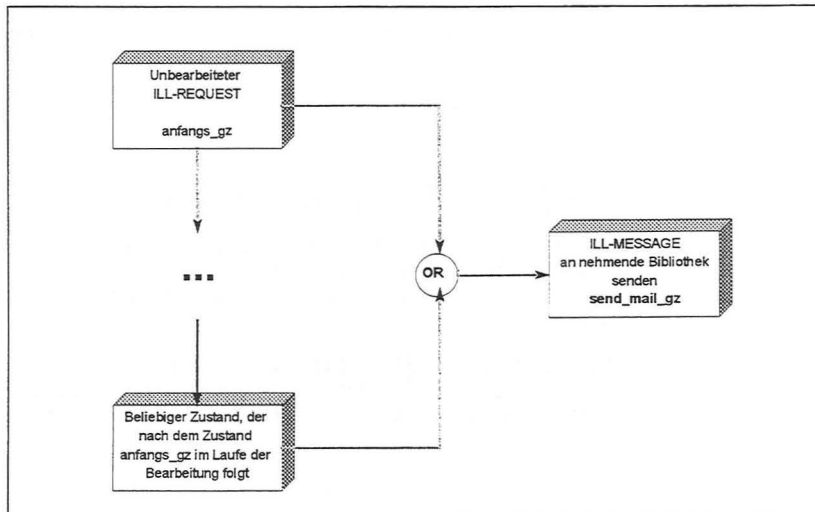


26.2.3

Zustandsübergänge beim Senden des ILL-Dienstes MESSAGE durch den Responder

Die gebende Bibliothek kann den Dienst MESSAGE senden, sobald sie einen ILL-REQUEST von der nehmenden Bibliothek erhalten hat. Durch das Programm *feemeg.c* wird der Zustand 'send_mail_gz' mit Sekundärerkennung 'send_message' generiert, der einen sendefähigen ILL-MESSAGE erzeugt. Der Sendeserver versendet den Dienst und schließt den Zustand 'send_mail_gz' danach ab.

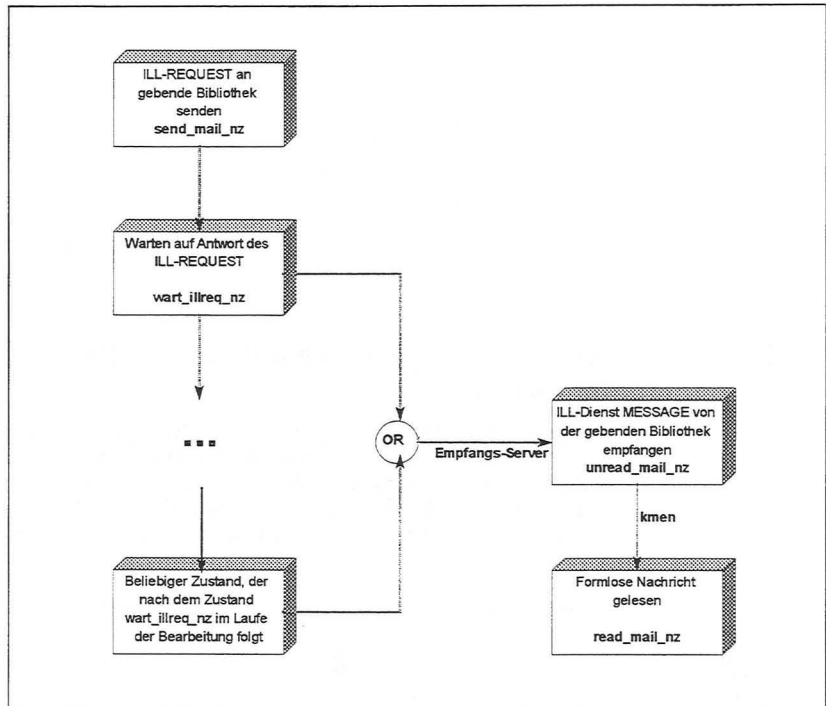
Abb. 26.4
Zustandsübergänge
beim Senden des
Dienstes MESSAGE
(Responder)



26.2.4 Zustandsübergänge beim Empfangen des ILL-Dienstes MESSAGE beim Requester

Der Requester kann den Dienst MESSAGE nur empfangen, wenn eine ILL-Transaktion existiert. Der Empfangsserver überprüft beim Eintreffen einer formlosen Nachricht, ob ein entsprechender Vorgang vorhanden ist und generiert den Zustand 'unread_mail_nz'. Nachdem der Benutzer am Fernleih-Client die empfangene Nachricht bearbeitet hat, wird durch das Programm *fekmen.c* der Zustand des entsprechenden Datensatzes in der Fernleihdatenbank von 'unread_mail_gz' auf 'read_mail_nz' gesetzt.

Abb. 26.5
Zustandsübergänge
beim Empfangen des
Dienstes MESSAGE
(Requester)



26.3 Die globalen ILL-Transaktionszustände

Die lokalen Zustände eines Endsystems sind also für die Kontrolle des Verhaltens und für die Prüfung und Durchführung von Zustandsübergängen von Bedeutung. Dabei spielt im lokalen Prozeß der Zustand der Partnerbibliothek keine Rolle.

Erst beim Senden und Empfangen von Diensten werden die Zustände der Partnerbibliothek wichtig. Daher existieren zusätzlich zu den lokalen Zuständen auch globale ILL-Transaktionszustände⁹⁹. Sie dienen zur Prüfung, ob ein Sen-

⁹⁹ Siehe Abschnitt 3.4.4.

der einen Dienst in seinem aktuellen globalen Zustand senden darf und ob ein Empfänger einen Dienst in seinem aktuellen globalen Zustand empfangen darf.

Die folgende Tabelle zeigt in welchen globalen Zuständen ein MESSAGE von der gebenden/nehmenden Bibliothek gesendet/empfangen werden kann:

Abb. 26.6
Globale ILL-
Transaktions-
zustände

Requester (nehmende Bibliothek)		Responder (gebende Bibliothek)	
MESSAGE senden	MESSAGE empfangen	MESSAGE senden	MESSAGE empfangen
		IN-PROCESS	IN-PROCESS
		FORWARD	FORWARD
PENDING	PENDING		
NOT-SUPPLIED	NOT-SUPPLIED	NOT-SUPPLIED	NOT-SUPPLIED
CONDITIONAL	CONDITIONAL	CONDITIONAL	CONDITIONAL
CANCEL-PENDING	CANCEL-PENDING	CANCEL-PENDING	CANCEL-PENDING
CANCELLED	CANCELLED	CANCELLED	CANCELLED
SHIPPED	SHIPPED	SHIPPED	SHIPPED
RECEIVED	RECEIVED		
RENEW/PENDING	RENEW/PENDING	RENEW/PENDING	RENEW/PENDING
RENEW/OVERDUE	RENEW/OVERDUE	RENEW/OVERDUE	RENEW/OVERDUE
OVERDUE	OVERDUE	OVERDUE	OVERDUE
NOT-RECEIVED/ OVERDUE	NOT-RECEIVED/ OVERDUE		
RECALL	RECALL	RECALL	RECALL
RETURNED	RETURNED		
		CHECKED-IN	CHECKED-IN
LOST	LOST	LOST	LOST

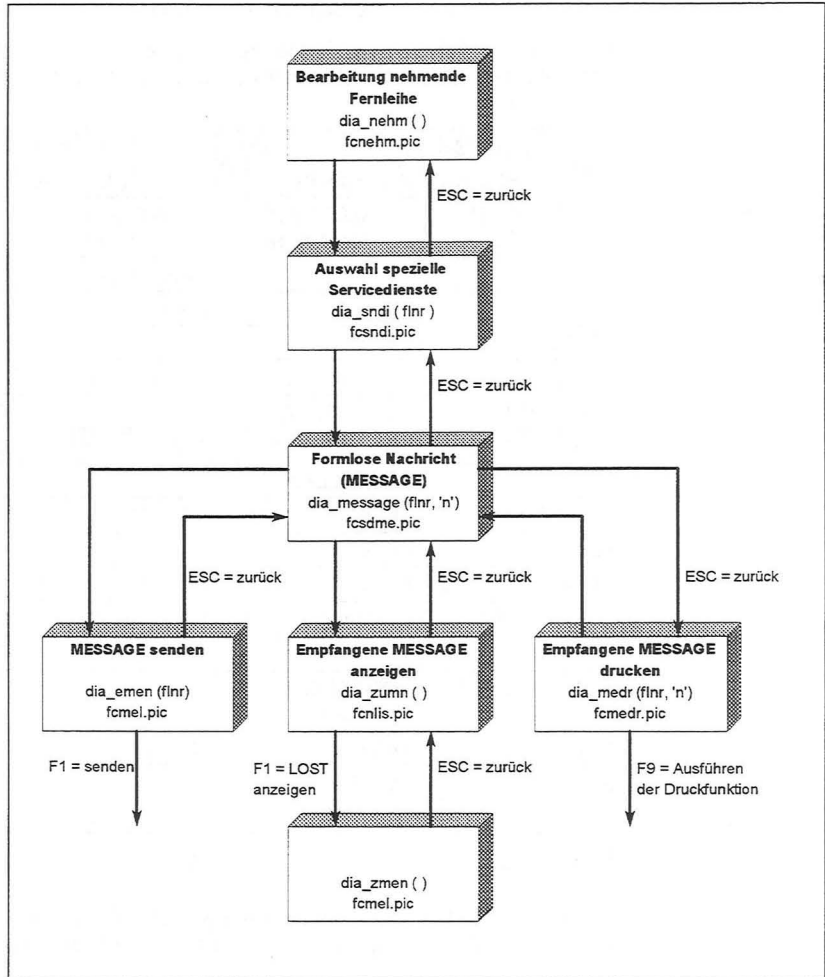
Nach dem Senden oder Empfangen des ILL-Dienstes MESSAGE bleibt der aktuelle globale Zustand bestehen. Es findet somit kein Zustandswechsel statt.

26.4 Menüsteuerung beim Fernleih-Client

26.4.1 Nehmende Fernleihe (Requester)

In der nehmenden Bearbeitung kann man zum Menü des Dienstes MESSAGE gelangen, indem man im Hauptmenü den Punkt „Bearbeitung nehmende Fernleihe“ und im darauffolgenden Untermenü den Punkt „Auswahl spezielle Servicedienste“ wählt. Unter allen Servicediensten, die dann zur Auswahl stehen, findet man auch den Dienst Formlose Nachricht (MESSAGE). Wird der Menüpunkt MESSAGE gewählt, so wird die Funktion 'dia_message' aufgerufen, die das Bild 'fcsdme.pic' öffnet. Das Menü des ILL-Dienstes MESSAGE ist wiederum unterteilt in die Untermenüs „MESSAGE senden“, „Empfangene MESSAGE anzeigen“ und „Empfangene MESSAGE drucken“.

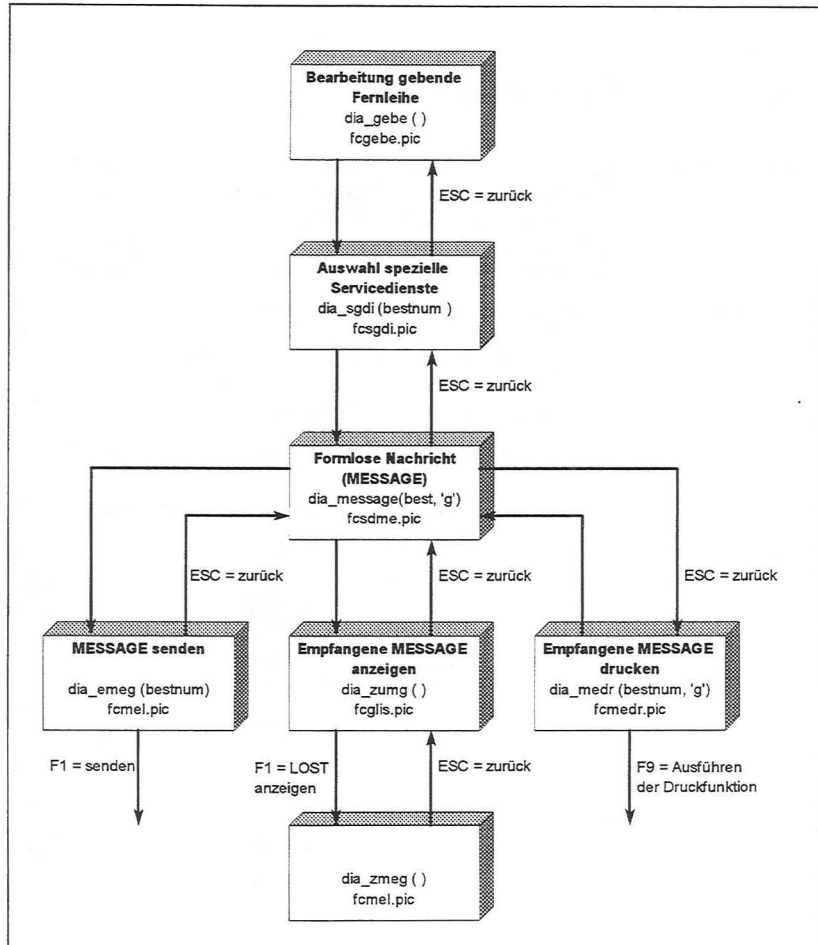
Abb. 26.7
Menü MESSAGE
(Requester)



26.4.2 Gebende Fernleihe (Responder)

In der gebenden Bearbeitung gelang man analog zur nehmenden Bearbeitung zum ILL-Dienst MESSAGE. Allerdings muß hier im Hauptmenü die Verzweigung „Bearbeitung gebende Fernleihe“ gewählt werden. Die Dialogfunktionen der gebenden Bearbeitung unterscheiden sich zu denen der nehmenden Bearbeitung lediglich dadurch, daß sie nicht mit der Fernleihnummer, sondern mit der Bestellnummer arbeiten und daß sie Zugriffe auf die gebenden Tabellen der Datenbank realisieren.

Abb. 26.8
Menü MESSAGE
(Responder)



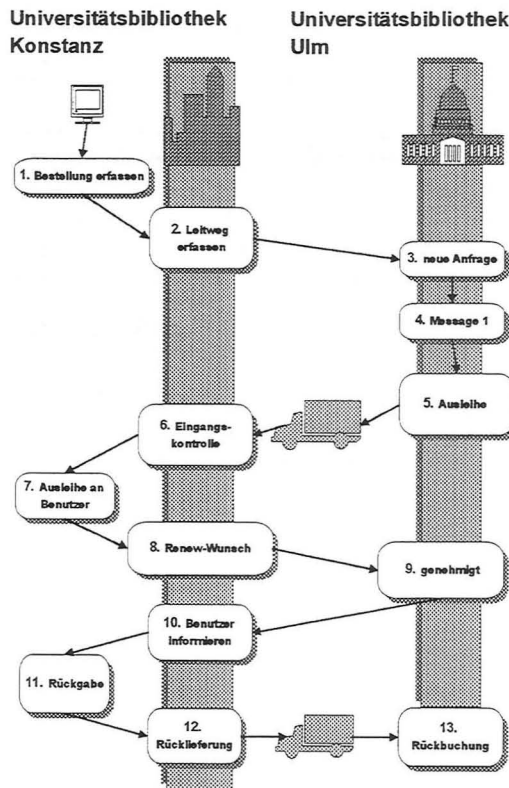
27 Benutzerakzeptanz

Anforderungsprofil Bereits vor dem Beginn der Implementierung der neun ILL-Dienste fand ein Gespräch mit den künftigen Benutzern des Fernleihsystems statt. Dabei wurden die gewünschten **Anforderungen**, die die Benutzer an das System stellen, besprochen. Die Verwirklichung der Anforderungen wurde anhand der Möglichkeiten des ILL-Protokolls und des bereits bestehenden Fernleihsystems geprüft.

Implementierung Nachdem die Anforderungen festgelegt waren, wurde mit der **Implementierung** begonnen. Nach Fertigstellung der ersten fünf Dienste wurden diese den künftigen Benutzern vorgestellt. Anhand einer schriftlich festgelegten Abfolge wurden die Benutzer aufgefordert, die einzelnen Bearbeitungsschritte selbstständig im Fernleihsystem durchzuführen.

Ein Beispiel für einen solchen Ablauf ist in der folgenden Abbildung dargestellt:

Abb. 27.1
Ablauf einer
Fernleihtransaktion



Benutzerakzeptanz	<p>Dieser erste Test zur Benutzerakzeptanz zeigte, daß die Benutzer sich schnell im Menü orientieren und die Bearbeitungsschritte durchführen konnten. Folgende Änderungswünsche und Verbesserungsvorschläge wurden anschließend entgegengenommen:</p> <ul style="list-style-type: none">• Beschriftung der ILL-Dienste durch ihre englische Bezeichnung anstelle der deutschen Übersetzung. <p>Diesem Benutzerwunsch wurde Folge geleistet.</p> <ul style="list-style-type: none">• Die Sicherheitsabfrage „Wollen Sie den ILL-Dienst zur Transaktion mit der Nummer xxxx.xxxx.xx wirklich versenden (j/n)“ wurde von den Benutzern als störend empfunden. Die Benutzer begründeten dies durch den erhöhten Arbeitsaufwand, wenn beim Versenden von größeren Mengen von ILL-Diensten die Kontrollabfrage jedesmal zu beantworten ist. <p>Dieser Änderungswunsch wurde nicht durchgeführt, da er zwar die Bearbeitung beschleunigen würde, aber die Fehleranfälligkeit erhöhen würde. In diesem Fall wurde die Sicherheit der Schnelligkeit vorgezogen.</p> <ul style="list-style-type: none">• Eine graphische Benutzeroberfläche mit Maussteuerung wurde für die Zukunft gefordert.
Fertigstellung und Testphase	<p>Nach diesem ersten Benutzertest wurden die restlichen vier Dienste realisiert. Nach der Fertigstellung aller neun ILL-Dienste und nach Abschluß der Testphase wurde der Fernleih-Client auf einem Rechner in der Fernleihabteilung installiert. Die Benutzer wurden aufgefordert das Fernleihsystem über einen längeren Zeitraum mit Testdaten zu testen und auftretende Fehler oder Änderungsmaßnahmen zu notieren. In dieser einmonatigen Testphase wurden einige kleine Fehler, die der Benutzer entdeckte, korrigiert. Größere Änderungen waren jedoch nicht durchzuführen.</p>
Präsentation	<p>Während der Testphase in der Fernleihabteilung wurden zwei Präsentation des Systems durchgeführt. Einmal wurde das Fernleihsystem der Informatikabteilung der Bibliothek Konstanz vorgestellt und ein anderes Mal 21 Interessenten aus Bibliotheken in der Schweiz.</p>

28

Schlußbetrachtung

- Einarbeitung** In den ersten drei Wochen der Diplomarbeit arbeiteten wir uns in die bestehenden Programme des Konstanzer Fernleihsystems und in das ILL-Protokoll ein. Aufgrund der Möglichkeit, sich im Team auszutauschen, hatten wir schnell einen Überblick über die Funktionsweise des Fernleihsystems und seiner Programme gewonnen. Auch die Betreuung von Seiten der Bibliothek durch Herrn Brink war optimal, so daß bei auftretenden Unklarheiten bezüglich der bestehenden Programme sofort Rücksprache gehalten werden konnte. Infolge dieser Voraussetzungen konnten wir die Einarbeitung forcieren und bereits nach 18 Arbeitstagen mit der Programmierung des ersten ILL-Dienstes beginnen.
- Implementierung des ersten ILL-Dienstes** Wir beschlossen, diesen ersten ILL-Dienst gemeinsam zu programmieren, da wir so zusammen in die Thematik einsteigen und bei auftretenden Problemen die Lösung im Team erarbeiten konnten. Hätten wir bereits zu Beginn jeder einen getrennten Dienst bearbeitet, so hätten Fragen von Seiten einer Person, die deren speziellen Dienst betreffen, von den anderen nicht beantwortet werden können, da sie die Definition und Funktionsweise dieses Dienstes nicht gekannt hätten. Durch die gemeinsame Implementierung des ersten ILL-Dienstes konnte auch eine optimale Vorgehensweise und Reihenfolge für die verschiedenen Bearbeitungsschritte gefunden werden, die bei der Programmierung der weiteren Dienste sehr hilfreich war und stets eingehalten wurde.
- Aufteilung der ILL-Dienste** Von den verbleibenden acht ILL-Diensten wurden je zwei Dienste auf die drei Teammitglieder verteilt. Die restlichen zwei Dienste wurden noch keiner Person zugeordnet. Da der Arbeitsaufwand für die ILL-Dienste unterschiedlich groß ist, aber vorher nicht genau abschätzbar war, sollte die Person, die ihre beiden Dienste zuerst abgeschlossen hat, mit der Implementierung der restlichen zwei Dienste beginnen.
- Zeiteinteilung** Um den Abschluß der Diplomarbeit nach sechs Monaten gewährleisten zu können, stellten wir mit der Software „Microsoft Project“ einen Zeitplan für die Durchführung der einzelnen Aufgaben auf. Wir planten für die Implementierung eines ILL-Dienstes durch eine Person drei Wochen ein. Für die restlichen zwei Dienste kalkulierten wir nur noch zwei Wochen ein, begründet durch die routiniertere Vorgehensweise, die sich in der Zwischenzeit eingestellt haben mußte. Für die Dokumentation veranschlagten wir die restlichen sieben Wochen.

Die folgende Abbildung zeigt den erstellten Projektplan mit Microsoft Project:

Abb. 28.1
Projektplan

Nr.	Name	Dauer	Berechneter Anfang	Berechnetes Ende	Vorgänger	Ressourcennamen
1	Einarbeitung	15d	4.10.95	24.10.95		Team
2	MESSAGE	17d	25.10.95	17.11.95	1	Team
3	OVERDUE	15d	20.11.95	8.12.95	2	Cornelia Basler
4	RECALL	15d	11.12.95	12.1.96	3	Cornelia Basler
5	RENEW	15d	20.11.95	8.12.95	2	Axel Glatz
6	RENEWANSWER	15d	11.12.95	12.1.96	5	Axel Glatz
7	LOST	15d	20.11.95	8.12.95	2	Claudia Zen
8	DAMAGED	15d	11.12.95	12.1.96	7	Claudia Zen
9	CANCEL	10d	15.1.96	26.1.96	4,6,8	Team
10	CANCEL-REPLY	10d	29.1.96	9.2.96	9	Team
11	Dokumentation	35d	12.2.96	29.3.96	10	Team
12						
13						
14						

Projekt	Nr.	Vorgangname	Einh.	Arbet	Verzögerung	Berechneter Anfang	Berechnetes Ende
DIPLOM	11	Dokumentation	1	245h	0h	12.2.96	29.3.96
DIPLOM	2	MESSAGE	1	115h	0h	25.10.95	17.11.95
DIPLOM	1	Einarbeitung	1	105h	0h	4.10.95	24.10.95
DIPLOM	9	CANCEL	1	10d	0d	15.1.96	26.1.96
DIPLOM	10	CANCEL-REPLY	1	10d	0d	29.1.96	9.2.96

Zwischenzeitlich verbrachten wir einen Tag in der Fernleih-Abteilung der Bibliothek Konstanz, um uns mit den Verwaltungsarbeiten der Fernleihe und den einzelnen Bearbeitungsschritten vertraut zu machen. Aufgrund dieser Informationen konnten wir die Arbeitsschritte der Fernleihe benutzerorientiert implementieren.

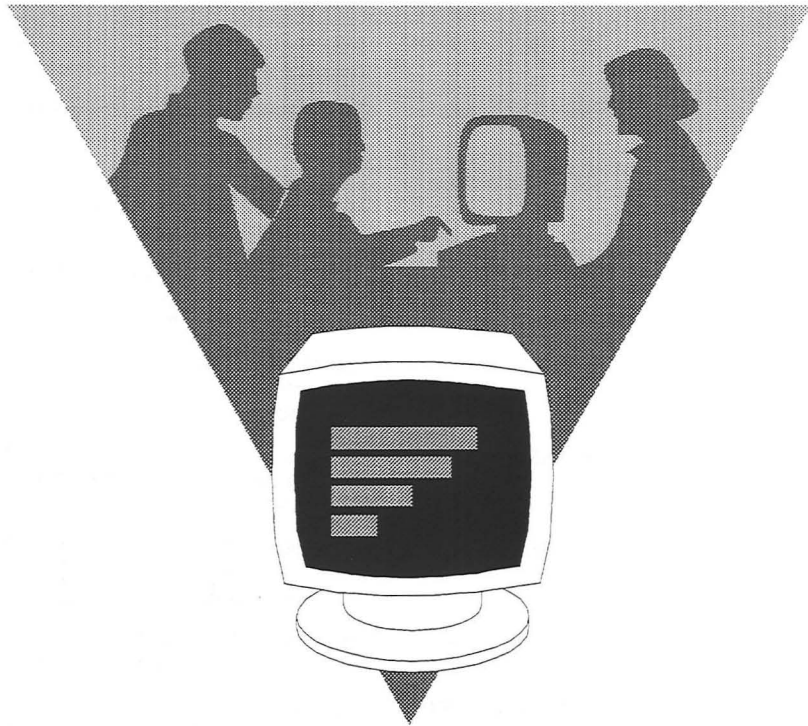
Rückblickend auf das Arbeiten im Team ist zu sagen, daß die Einarbeitung in das bereits existierende komplexe Fernleihsystem und das ILL-Protokoll nur durch die gemeinsame Teamarbeit so schnell vollzogen werden konnte. Aufgrund ständigen Informationsaustausches zwischen den Teammitgliedern konnten einerseits die ILL-Dienste konsistent implementiert werden und andererseits ihre Wechselwirkung mit anderen Diensten berücksichtigt werden.

Nach Abschluß unserer Diplomarbeit hatte sich der Quelltext des Konstanzer Fernleihsystems verdoppelt:

- | | |
|--------|---|
| Server | <ul style="list-style-type: none"> • 160 Quelldateien • 80.000 Lines of C-Code • 4 Server-Prozesse |
| Client | <ul style="list-style-type: none"> • 10 Quelldateien • 10.000 Lines of C-Code • 60 Dialogbilder |

Schlußbetrachtung

Anhang



Normative Verweise

ISO 2108:1978	Documentation - International Standard Book Number (ISBN).
ISO 2709:1981	Documentation - Format for Bibliographic Information Interchange on Magnetic Tape.
ISO 3297:1986	Documentation - International Standard Serial Number (ISSN).
ISO 4217:1981	Codes for the Representation of Currencies and Funds.
ISO 7498:1988	Information Processing Systems - Open Systems Interconnection (OSI) - Basic Reference Model.
ISO 8459-1:1988	Documentation - Bibliographic Data Element Directory - Part 1: Interloan Application.
ISO TR8509:1987	Information Processing Systems - Open Systems Interconnection - Service Conventions.
ISO 8601:1988	Data Elements and Interchange Formats Information Interchange - Representation of Dates and Times.
ISO 8649:1988	Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element (ACSE).
ISO 8650:1990	Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element (ACSE).
ISO 8822:1988	Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Service Definition.
ISO 8824:1990	Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).
ISO 8825:1990	Information Processing Systems - Open Systems Interconnection - Specification of Encoding Rules for Abstract Syntax Notation One (ASN.1).
ISO 9545:1989	Information Processing Systems - Open Systems Interconnection - Application Layer Structure.
ISO 9646-2:1990	Information Processing Systems - Open Systems Interconnection - OSI Conformance Testing Methodologie and Framework. Part 2: Abstract Test Suite Specification.

ISO 9735:1988	Electronic Data Interchange for Administration, Commerce and Trade (EDIFACT)- Application Level Syntax Rules.
ISO 9834-1:	Information Processing Systems - Open Systems Interconnection - Procedures for Specific OSI Registration Authorities - Part 1: General Procedures.
ISO/IEC TR 10000-1	Information Technology - Framework and Taxonomy of International Standardized Profiles - Part 1: Framework.
ISO/IEC TR 10000-2	Information Technology - Framework and Taxonomy of International Standardized Profiles - Part 2: Taxonomy of Profiles.
ISO 10021-4:1990-	Information Processing Systems - Text Communication - Message Oriented Text Interchange System (MOTIS) - Part 4: Message Transfer System: Abstract Service Definition and Procedures.
ISO 10021-7:1990-	Information Processing Systems - Text Communication - Message Oriented Text Interchange System (MOTIS) - Part 7: Interpersonal Messaging System.
ISO 10021-12-	Information Technology - Text Communication-Message Oriented Text Interchange System (MOTIS) - Part 12: PICS Proforma for Message Transfer Protocol.
ISO 10021-15-	Information Technology - Text Communication-Message Oriented Text Interchange System (MOTIS) Part 15: PICS Proforma for Interpersonal Messaging.
ISO DIS 10026-	Information Processing Systems - Open Systems Interconnection - Distributed Transaction Processing. Parts 1 to 3.
ISO 10160:1991-	Information and Documentation - Open Systems Interconnection - Interlibrary Loan Application Service Specification.
ISO 10161:1991-	Information and Documentation - Open Systems Interconnection - Interlibrary Loan Application Protocol Specification.
ISO 10163:1991-	Information and Documentation - Open Systems Interconnection - Search and Retrieve Application Specification.
IFOBS N91	Information and Documentation - Open Systems Interconnection - International Standardized Profile - Application Profile for Interlibrary Loan with Interpersonal Messaging (ILL-IPM).

Literaturverzeichnis

[Babatz, Bogen und Pankoke-Babatz, 1990] Robert Babatz, Manfred Bogen und Uta Pankoke-Babatz. *Elektronische Kommunikation - X.400 MHS*. Verlag Vieweg, Braunschweig, 1990.

[Bibliotheksdienst 25, 1991] *Bibliotheksdienst 25*. Deutsches Bibliotheksinstitut, Berlin, 1991.

[Bork, 1989] Heinz Bork. *OSI für Bibliotheken - Normen, Projekte und Vorschläge zur Anwendung elektronischer Kommunikation*. Deutsches Bibliotheksinstitut, Berlin 1989.

[Busch, 1993] Michael Busch. *Kommunikative Eleganz - Sockets unter BSD-Unix*. iX Multiuser-Multitasking-Magazin, S. 156-158, Ausgabe 2/1993. Verlag Heinz Heise GmbH & Co KG, Hannover, 1993.

[Die deutsche Bibliothek, 1992] OSI-Anwendungen für Bibliotheken und Fachinformationen. *Band 1, Anforderungen und Lösungskonzepte*. Die deutsche Bibliothek, Frankfurt a.M., 1992.

[Die deutsche Bibliothek, 1992] OSI-Anwendungen für Bibliotheken und Fachinformationen. *Band 2, Standortnachweise für die Fachinformation: Realisierungsvorschläge*. Die deutsche Bibliothek, Frankfurt a.M., 1992.

[Eckardt und Nowak, 1988] Dr. Klaus-Jürgen Eckardt, Dr. Robert Nowak. *Standard-Architekturen für Rechnerkommunikation*. R. Oldenbourg Verlag, München, Wien, 1988.

[Franken und Neubauer, 1994] Klaus Franken und Karl-Wilhelm Neubauer. *Elektronische Fernleihe und Dokumentlieferung*. Bibliothek aktuell, Sonderheft 11, Konstanz, 1994.

[Friede, 1995] Erich Friede. *Das Konstanzer Fernleihprojekt*. Bibliotheksdienst 29 H.3, S.536-538. Deutsches Bibliotheksinstitut, Berlin, 1995.

[Glaser, Hein, Vogl, 1990] Gerhard M. Glaser, Mathias Hein und Johannes Vogl. *TCP/IP: Protokolle, Projektplanung, Realisierung*. DATACOM-Buchverlag, Pulheim, 1990.

[Gorys, 1989] Lukas T. Gorys. *TCP/IP Arbeitsbuch: Kommunikationsprotokolle zur Datenübertragung in heterogenen Systemen*. Hüthig Buch Verlag, Heidelberg, 1989.

[ISO 10160, 1991] ISO 10160. *Information and Documentation - Open System Interconnection (OSI) - Interlibrary Loan Application Service Definition*. Beuth Verlag GmbH, Berlin, 1991.

[ISO 10161, 1991] ISO 10161. *Information and Documentation - Open System Interconnection (OSI) - Interlibrary Loan Application Protocol Specification*. Beuth Verlag GmbH, Berlin, 1991.

- [**Kauffels, 1992**] Franz-Joachim Kauffels. *Personal Computer und lokale Netzwerke: Grundlagen der Vernetzung, Aufbau und Wirkungsweise lokaler Netze LAN-Software von IBM, 3COM, Microsoft, Novell, Banyan unter DOS, OS/2 und Unix, LAN-Tests*. Markt&Technik Verlag AG, Haar bei München, 1992.
- [**Kerner, 1995**] Helmut Kerner. *Rechnernetze nach OSI*. Addison-Wesley, Bonn, München, Paris [u.a.], 1995.
- [**Kerninghan und Ritchie, 1990**] Brian W. Kerninghan und Dennis M. Ritchie. *Programmieren in C*. Carl Hanser Verlag, München, 1990.
- [**Kuon, Lehmler und Mühle, 1991**] Christa Kuon, Wilfried Lehmler und Helmut Mühle. *Analyse der gebenden und nehmenden Fernleihe an der Bibliothek der Universität Konstanz*. Deutsches Bibliotheksinstitut, Berlin, 1991.
- [**Nagelsmeier-Linke, 1992**] Marlene Nagelsmeier-Linke. *Beschleunigung der Fernleihe durch Nutzung moderner Informations- und Kommunikationstechnologien: Das Konstanzer Fernleihprojekt*. Zeitschrift für Bibliothekswesen und Bibliographie Sonderheft 55; 82. Deutscher Bibliothekartag in Bochum 1992; Bibliotheken in alten und neuen Hochschulen; S. 366-385. Vittorio Klostermann, Frankfurt a.M., 1992.
- [**Plattner, Lanz, Lubich, Müller, Walter, 1991**] Bernhard Plattner, C.Lanz, H.Lubich, M.Müller und T.Walter. *X.400 Message Handling: Standards, Interworking, Applications*. Addison-Wesley, New York, Amsterdam [u.a.], 1991.
- [**Stevens, 1992**] Richard W. Stevens. *UNIX-Netzwerkprogrammierung*. Hanser-Verlag, München, 1992.
- [**Tanenbaum, 1992**] Andrew S. Tanenbaum. *Computer-Netzwerke*. Wolfram's-Fachverlag, Attenkirchen, 1992.
- [**Zingler, 1992**] Kai Zingler. *Dokumentations des ASN.1-Compilers. Erweiterung/Modifikation des ASN.1-Compilers der Universität Erlangen*. ISOFT Kommunikationstechnologien GmbH, Berlin, 1990.