

Studienarbeit

Implementierung eines "Voice Mail Service Centre" am GSM Versuchsnetzwerk der TU Dresden

Vorgelegt von: Thomas Liske

Geboren am: 17. Dezember 1982 in: Dresden

Betreuer:	Dipl.-Ing. Samer Sulaiman
Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. Ralf Lehnert
Tag der Einreichung:	22. August 2005

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag an der Fakultät Elektrotechnik und Informationstechnik eingereichte Studienarbeit zum Thema

Implementierung eines "Voice Mail Service Centre" am GSM Versuchsnetzwerk der TU Dresden

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 22. August 2005

Thomas Liske

Inhaltsverzeichnis

1	Einleitung	3
2	GSM-Versuchsnetz	5
2.1	Vorhandene Hardware-Module	5
2.1.1	Base Station Subsystem	6
2.1.2	Mobile Switching Centre	6
2.2	Vorhandene Software-Module	7
2.2.1	MSC	7
2.2.2	Gateway	9
3	Voice Mail System im GSM-Netz	11
3.1	Übersicht	11
3.2	Einsatz	12
3.3	DTMF-Signalisierung	12
4	Implementierung	14
4.1	Konzept	14
4.2	VMS-Modul	14
4.2.1	Ereignisverarbeitung	14
4.2.2	Verbindungsverwaltung	15
4.2.3	Ablaufsteuerung	16
4.2.4	Datenhaltung	19
4.3	MSC-Software	21
4.3.1	HLR	21
4.3.2	MSC_TIMER	22
4.3.3	MSC	22

4.3.4	SMSC	24
4.4	Gateway-Software	24
4.4.1	SIP-Gateway	25
4.4.2	ISDN-Gateway	26
4.4.3	mbISDN	26
4.5	Signalisierungsabläufe	27
4.5.1	Ruf-Aufbau MS-zu-VMS	27
4.5.2	Ruf-Aufbau ISDN-zu-VMS	29
4.5.3	Ablaufsteuerung	31
4.5.4	Ruf-Abbau	32
5	Zusammenfassung	33
A	Abkürzungsverzeichnis	34
B	Abbildungsverzeichnis	36
C	Tabellenverzeichnis	37
D	Anhang	38
D.1	Lifero-Dateien	38
D.1.1	VMS	38
D.1.2	MSC	40
D.2	Tcl	46
D.2.1	Funktionsübersicht	46
D.2.2	Standardskripte	48
E	Literaturverzeichnis	55

1 Einleitung

Das GSM¹ ist die Grundlage der im großen Maße erfolgreicher Mobilfunknetze. Mit der fortwährenden Entwicklung und schrittweisen Einführung des UMTS² wird in Europa bereits die nächste Mobilfunkgeneration eingeleitet.

Dem Lehrstuhl für Telekommunikation an der Technischen Universität Dresden steht bereits seit 1998 ein komplettes BSS³ der Firma Lucent Technologies zur Verfügung. Der vorhandene BSC⁴ und die daran angeschlossenen BTS⁵ werden für Lehr- und Demonstrationzwecken eingesetzt.

Da für den Betrieb eines GSM-Netzes grundlegende Komponenten wie ein MSC⁶ fehlten, wurden diese durch zahlreiche Diplom- und Studienarbeiten ([Bas99], [Wei00], [Lem00], [Gue01], [Woh01] und [Eic02]) auf einem handelsüblichen Rechner implementiert. Dadurch ist es unter anderem möglich, nicht nur im GSM-Netz selbst, sondern auch in das universitätsinterne ISDN⁷-Netz zu telefonieren.

Das GSM-Netz wird teilweise von den Mitarbeitern des Lehrstuhls verwendet, um innerhalb der Reichweite des Netzes unter einer Telefonnummer erreichbar zu bleiben. Dafür kann von den hausinternen Endgeräten eine Rufweiterleitung in das GSM-Netz geschaltet werden.

Das GSM-Netz hatte bisher kein VMS⁸, welches eine aus kommerziellen Netzen bekannte Mailbox-Funktionalität bieten konnte. Im universitätsinternen ISDN-Netz ist eine solche Funktion für die einzelnen Hausapparate nicht vorgesehen. Daraus entstand die Motivation für diese Arbeit. Jeder Mitarbeiter sollte bei Bedarf eine eigene Mailbox mit den

¹Global System for Mobile communications

²Universal Mobile Telecommunication System

³Base Station Subsystem

⁴Base Station Controller

⁵Base Transceiver Station

⁶Mobile Switching Centre

⁷Integrated Services Digital Network

⁸Voice Mail System

heute üblichen Funktionen zur Verfügung zu haben.

In dieser Arbeit wird eine Implementierung eines VMS für das GSM-Versuchnetz des Lehrstuhles präsentiert. Dabei wurde vor allem auf eine möglichst breite Einsatzfähigkeit des Lösungskonzeptes geachtet.

Im Folgenden wird zunächst kurz die Struktur des vorhandenen GSM-Netzes dargestellt. Im Anschluß daran wird näher auf das Konzept und die Implementierung des VMS eingegangen. Im letzten Kapitel werden die erreichten Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf weitere Verwendungsmöglichkeiten der vorliegenden Implementierung gegeben.

2 GSM-Versuchsnetz

2.1 Vorhandene Hardware-Module

Abbildung 2.1: GSM-Versuchsnetz der TU-Dresden

Die hier vorliegende Arbeit wurde auf Grundlage des GSM-Versuchsnetzwerkes der TU-Dresden (Abb. 2.1) angefertigt. Die Anlage wurde von der Firma Lucent Technologies zur Verfügung gestellt und besteht aus einem BSS. Im Rahmen der Diplomarbeiten von M. Lemke [Lem00] und M. Weidle [Wei00] wurde ein MSC auf einem handelsüblichen PC implementiert. Dies ermöglicht Telefonverbindungen zwischen zwei MS¹ innerhalb des Versuchsnetzes. Später wurde durch die Diplomarbeit von R. Wohlfahrt [Woh01] ein Gateway zur Anbindung an das ISDN-Netz der TU-Dresden entwickelt. Die darauf aufsetzende Diplomarbeit von U. Eichler [Eic02] ergänzte das Gateway um eine Schnittstelle

¹Mobile Station

zum SIP²-Standard für VoIP³.

2.1.1 Base Station Subsystem

Das BSS besteht aus einem BSC mit zwei angeschlossenen BTS. An jede BTS sind jeweils zwei *Radio Terminals* (RT) angeschlossen. Der TU-Dresden stehen zwei Trägerfrequenzen im 900 Mhz-Band zur Verwendung innerhalb des Institutes zur Verfügung.

Zusätzlich ist ein OMC⁴ vorhanden. Damit können Parameter des BSS wie Sendefrequenzen und -leistungen eingestellt und der Systemzustand der BSS ermittelt werden.

2.1.2 Mobile Switching Centre

Das MSC wurde durch M. Lemke und M. Weidle auf einem Standard-PC implementiert. Dafür ist dieser mit einer PCM30-fähigen SS7-Schnittstellenkarte ausgerüstet und darauf die Funktionalität eines MSC in Software realisiert.

SS7-Schnittstellenkarte

Für die Kommunikation mit dem BSS enthält der MSC Rechner eine *Septel SS7 Line Card* der Firma DataKinetics. Diese bietet einen E1-konformen PCM30-Anschluß und beherrscht die Verarbeitung von Signalisierungsinformationen nach dem SS7⁵-Standard. Zur Kommunikation mit anderen Schnittstellenkarten ist ein MVIP⁶-Anschluß vorhanden. Die Karte enthält eine Schaltmatrix, mit der die Datenkanäle beliebig zwischen der E1-Schnittstelle und dem MVIP-Bus geschaltet werden können.

ISDN-Schnittstellenkarte

Für das im Rahmen der Diplomarbeit von R. Wohlfahrt [Woh01] entwickelte ISDN-Gateway enthält der PC außerdem eine ISDN-Karte vom Typ *Netaccess BRI-PCI8 DV/ST* der Firma Brooktrout Technology. Diese bietet insgesamt vier S_0 -Schnittstellen

²Session Initiation Protocol

³Voice over IP

⁴Operation and Maintenance Centre

⁵Signalling System No. 7

⁶Multi Vendor Integration Protocol

und einen Anschluß für den MVIP-Bus. Für die Verarbeitung der S_0 -Signalisierung ist ein HDLC-Controller enthalten. Auch hier ist eine Schaltmatrix vorhanden, die das Schalten der Datenkanäle zwischen verschiedenen Schittstellen erlaubt. Außerdem ist es möglich, Datenkanäle softwareseitig auszulesen und zu schreiben.

2.2 Vorhandene Software-Module

Die Softwareimplementation des MSC ist in zwei seperate Softwarepakete unterteilt. Auf der einen Seite steht das MSC. Dort werden alle Verwaltungsaufgaben und Funktionen für Mobil-zu-Mobil Rufe realisiert. Die zweite Komponente ist das Gateway, welches Schnittstellen in andere Netze bietet. Zur Zeit sind Schnittstellen ins ISDN-Netz und zu SIP-Adressen realisiert.

2.2.1 MSC

Das MSC hat in einem GSM-Netz die Aufgabe der Vermittlungsstelle. Dafür müssen entsprechende Signalisierungen, die für den Rufauf- und -abbau nötig sind, mit dem BSC abgewickelt werden. Die in der Programmiersprache C implementierte Software gliedert sich dabei in mehrere Module (Abb. 2.2), die als eigene Prozesse laufen. Die Kommunikation zwischen diesen Prozessen ist mittels *Message Queues* [Wei00, Seite 65] realisiert. Die Module wurden mittels des Entwicklungswerkzeuges *Libero* in Form einer FSM⁷ [Wei00, Seite 61] erstellt.

Libero bietet dabei ein Rahmenwerk, welches anhand einer formalen Spezifikation von Zuständen und Zustandsübergängen ein Grundgerüst des Quelltextes erstellt. Dieses muß dann durch den Entwickler um die eigentliche Funktionalität ergänzt wird.

⁷Finite State Machine

Abbildung 2.2: Software-Stapel MSC

- Der CONNH⁸ bildet die Schnittstelle zwischen SS7-Schnittstellenkarte und den verschiedenen Software-Modulen.
- Die Protokoll-Module der GSM-Signalisierung verarbeiten Signalisierungsinformationen für ihre jeweilige Protokollschicht und leiten diese an die höheren oder niedrigeren Module im Protokoll-Stapel weiter. Jede Protokollschicht ist dabei in einem separaten Prozess implementiert. Für den verbindungspezifischen Teil wird hier außerdem für jede Verbindung, zwischen Mobilstation und MSC, eine Kopie der Prozesse erzeugt.
- Das MSC-Modul übernimmt die eigentliche Vermittlung der Nachrichten zwischen den Protokoll-Stapeln und verwaltet die Verbindungsinformationen für die Teilnehmer im GSM-Netz.

⁸Connection Handler

- Das HLR-Modul nimmt Funktionen des HLR⁹ und VLR¹⁰ des GSM-Netzes wahr. Da das GSM-Versuchsnetz nur zwei Zellen umfasst, wird kein separates VLR verwendet.
- Das SMS-Modul realisiert die Funktion des SMS¹¹ im Netz. Das SMSC¹²-Modul ist für das Zwischenspeichern von SMS-Nachrichten, die nicht sofort zugestellt werden können, verantwortlich.
- Das LOGMAN-Modul wird von allen Modulen des MSC verwendet, um Log-Meldungen zentral zu verarbeiten.
- Das MSC_TIMER-Modul wird vom MSC für das Feststellen von Zeitüberschreitungen beim Suchen von Teilnehmern im Netz verwendet.

2.2.2 Gateway

Das Gateway ist in der Programmiersprache *C++* implementiert und läuft unabhängig vom MSC. Der Nachrichtenaustausch zwischen externen Entitäten wird über sogenannte *Message Broker* abstrahiert. Dabei existieren *Message Broker* für das Anbinden des MSCs, die ISDN-Hardware oder des SIP-Software-Stapels. Die einzelnen Teile des Gateways kommunizieren über zwei *Message Queues* miteinander.

ISDN

Das ISDN-Gateway, welches im Rahmen der Diplomarbeit von R. Wohlfahrt [Woh01] entstanden ist, erweitert das GSM-Versuchsnetz um eine Anbindung an das ISDN-Netz der TU-Dresden. Für diese Anbindung müssen vom ISDN-Gateway die Signalisierungsnachrichten zwischen GSM- und ISDN-Netz übersetzt werden. Über den gemeinsamen MVIP-Bus können die Sprachkanäle zwischen GSM und ISDN zusammengeschaltet werden. Da beide Netze eine gleiche Kodierung der Sprachdaten verwenden, ist dies ohne Probleme möglich.

Das MSC wurde im Rahmen dieser Arbeit um eine Schnittstelle zum Registrieren für externe Gateways erweitert. Dabei ist es möglich, dass externe Gateways einen Vorwahl-

⁹Home Location Register

¹⁰Visitor Location Register

¹¹Short Message Service

¹²Short Message Service Centre

bereich für sich registrieren. Alle Rufe die mit dieser Vorwahl beginnen werden dann direkt an das Gateway weitergeleitet. Diese Funktion wurde auch in dieser Arbeit verwendet, um das Abhören der Mailbox zu realisieren.

SIP

Die Implementierung eines SIP-Gateways geschah im Rahmen der Diplomarbeit von U. Eichler [Eic02]. Die Funktionsweise ist vergleichbar mit der des ISDN-Gateways, nur das hier nicht eine Schnittstelle in das ISDN-Netz sondern zu SIP-Adressen realisiert wurde. Dabei wird der *Vovida SIP-Stapel* verwendet, der das SIP kapselt.

3 Voice Mail System im GSM-Netz

3.1 Übersicht

Das VMS ist ein grundlegender Service in heutigen GSM-Netzen. Für den Teilnehmer bietet es die Funktionen eines Anrufbeantworters für das GSM-Netz an. Es ist im GSM-Netz als eigenständige Netz-Komponente (Abb. 3.1) vorgesehen. Das MSC übernimmt dabei die Vermittlung zwischen den Teilnehmern und dem VMS.

Abbildung 3.1: wesentliche Komponenten eines GSM-Netzes

3.2 Einsatz

Für die Mailbox Funktion kann das VMS aus unterschiedlichen Gründen zum Einsatz kommen.

Wenn ein Teilnehmer gerufen wird, wird er zuerst im GSM-Netz gesucht (sogen. *Pagen* des Teilnehmers). Für diesen Vorgang gibt es ein festgelegtes Zeitlimit nachdem der Teilnehmer als nicht (mehr) eingebucht gilt und der Ruf auf das VMS geschaltet wird. Der Vorgang des Pagen ist nötig, da sich die MS nicht unbedingt beim Verlassen des Netzes abgemeldet haben muß. Mögliche Gründe hierfür wären ein leerer Akku oder ein zu schnelles Verlassen der Netzabdeckung.

Eine weitere Möglichkeit ist, das ein gerufener Teilnehmer nicht auf einen Ruf reagiert. Nach einem gewissen Zeitlimit kann auch hier der Ruf auf das VMS umgeleitet werden, da davon ausgegangen wird, das der Teilnehmer den Ruf aktuell nicht entgegen nehmen kann oder möchte.

In beiden Fällen ist es üblich, das ein entsprechender Ansagetext den Anrufenden über die Umleitung zum VMS informiert und ihm die Möglichkeit gibt, eine Sprachnachricht zu hinterlassen. Dabei informiert das VMS üblicherweise mittels einer SMS-Nachricht den Besitzer einer Mailbox, wenn neue Sprachnachrichten verfügbar sind.

Damit der Teilnehmer seine Mailbox abfragen kann gibt, besitzt das VMS für diese Funktion eine dedizierte Rufnummer. Dort sollte die Möglichkeit bestehen, aufgenommene Nachrichten (eventuell mehrfach) abzuhören und bei Bedarf auch zu löschen. Die Steuerung der Mailbox geschieht dabei üblicherweise durch DTMF¹-Signalisierung, die auf dem Endgerät durch Drücken einer der Zahlen-, der Stern- oder der Rautetaste ausgelöst werden.

3.3 DTMF-Signalisierung

Für ein VMS wird eine Interaktionsmöglichkeit zwischen Teilnehmer und VMS benötigt. Wenn ein Teilnehmer seine Mailbox abhört, sollte er die Möglichkeit haben, ausgewählte Nachrichten zu löschen oder zu wiederholen. Neben dem Einsatz für das Tonwahlverfahren im PSTN² kommen DTMF-Töne für die Interaktion zwischen Benutzern und

¹Dual Tone Multi Frequency

²Public Switched Telephone Network

Computersystemen wie einem VMS zum Einsatz. Diese ermöglichen es, das Drücken der Tasten 0-9, A-D, * und # durch Überlagerung zweier Sinusschwingungen auf dem Sprachkanal zu übertragen.

Während im PSTN die DTMF-Töne direkt im Sprachkanal übertragen werden, wird im GSM die Übertragung mittels extra Signalisierungsnachrichten [ETS96a] realisiert. Die beim Tastendruck ausgelösten Nachrichten sind in der Abbildung 3.2 dargestellt.

Abbildung 3.2: DTMF-Signalisierungsnachrichten im GSM-Netz

Die eigentliche Generierung der Töne muß auf Seite des MSC erfolgen. Somit kann es zu keiner Beeinträchtigung des DTMF-Tons durch die verlustbehaftete Komprimierung auf der Luftschnittstelle im GSM-Netz kommen.

4 Implementierung

4.1 Konzept

Es wurde entschieden, die nötigen Funktionen zur Realisierung möglichst vollständig in ein neues Software-Modul im MSC zu implementieren und somit dem generellen Aufbau von GSM-Netzen (Abb. 3.1) zu entsprechen. Dadurch beschränken sich die Änderungen im MSC auf das Umleiten der Anrufe zum VMS. Das Auslesen und Wiedergeben von Sprachdaten ist bereits für das SIP-Gateway implementiert, so dass eine Verwendung dieser Funktionen mit geringen Änderungen am Gateway als sinnvoll erschien.

Im MSC selbst müssen entsprechende Schnittstellen geschaffen werden, um eingehende Rufe auf die Mailbox umzuleiten. Für das Abfragen von Mailboxen ist außerdem eine entsprechende dedizierte Rufnummer vorzusehen und das Verarbeiten von DTMF-Signalisierungen zu implementieren.

4.2 VMS-Modul

4.2.1 Ereignisverarbeitung

Das VMS-Modul wurde analog zu den bereits existierenden Modulen implementiert. Es arbeitet ereignisbasiert, wobei Ereignisse durch Nachrichten von anderen Modulen ausgelöst werden. Es wurde wie in den bereits existierenden Modulen auf die Software *Libero* zurückgegriffen, um das Rahmenwerk zur Ereignisverarbeitung zu erstellen.

Zur Integration in das bestehende MSC und Gateway Konzept mußten eine Reihe von neuen Nachrichten eingeführt werden (siehe Tabelle 4.1).

Nachricht	Bedeutung
MSC_VMS_IND	MSC übergibt dem VMS einen Ruf
VMS_MSC_ACK	VMS akzeptiert einen Ruf
VMS_MSC_BUSY	VMS hat keine freien Ressourcen um den Ruf anzunehmen
MSC_VMS_RELEASE_IND	MSC zeigt Ende eines Rufes dem VMS an
VMS_GW_CONNECTION_REQ	VMS fordert neue Verbindung beim Gateway an
GW_VMS_CONNECTION_ACK	Gateway akzeptiert die Verbindung
GW_VMS_CONNECTION_REJ	Gateway weist die Verbindung ab
VMS_GW_PLAYBACK_REQ	VMS fordert das Abspielen von Sprachdaten vom Gateway
GW_VMS_PLAYBACK_FIN	Gateway hat die Sprachdaten erfolgreich abgespielt
GW_VMS_PLAYBACK_ERR	im Gateway ist beim Abspielen ein Fehler aufgetreten
VMS_GW_RECORD_REQ	VMS fordert das Aufnehmen von Sprachdaten vom Gateway
GW_VMS_RECORD_FIN	Gateway hat die Sprachdaten erfolgreich aufgenommen
GW_VMS_RECORD_ERR	im Gateway ist beim Aufnehmen ein Fehler aufgetreten
VMS_GW_RELEASE	VMS informiert Gateway über das Ende einer Verbindung
GW_VMS_CONNECTION_EST	Gateway zeigt dem VMS abgeschlossenen Verbindungsaufbau an
MSC_GW_LAPDID_REQ	MSC erfragt die LAPD-ID eines Rufes beim Gateway
GW_MSC_LAPDID_IND	Gateway teilt LAPD-ID eines Rufes dem MSC mit
DTMF_START	Start einer DTMF-Signalisierung aus dem GSM-Netz
DTMF_START_ACK	Empfangsbestätigung für DTMF_START
DTMF_STOP	Ende einer DTMF-Signalisierung aus dem GSM-Netz
DTMF_STOP_ACK	Empfangsbestätigung für DTMF_STOP

Tabelle 4.1: neue Nachrichten Primitiven

Eine Übersicht über die eingehenden und ausgehenden Nachrichten am VMS-Modul und der Aufruf der damit verbundenen Verarbeitungsroutinen ist im Anhang D.1.1 durch die *Libero*-Datei gegeben.

4.2.2 Verbindungsverwaltung

Im VMS-Modul gibt es eine zentrale Verwaltung der aktiven Verbindungen im VMS. Für jede Verbindung wird eine Datenstruktur vom Typ `vms_conn` angelegt, um verbindungs-spezifische Daten zu halten (siehe Tabelle 4.2).

Variable	Bedeutung
<code>num_caller</code>	MSISDN des Anrufers
<code>num_mowner</code>	MSISDN des Mailbox Besitzers
<code>connid</code>	Verbindungskennung auf MSC Seite
<code>connid_sip</code>	Verbindungskennung auf VMS Seite
<code>cause</code>	Grund, warum das VMS aktiviert wurde
<code>switching_type</code>	Verbindungstyp des Rufes (MVIP/ISDN)
<code>spool</code>	Verwaltung von Anrufer-Sprachnachrichten
<code>spool_ptr</code>	Zeiger auf aktuelle Anrufer-Sprachnachricht
<code>sms_stack</code>	am Verbindungsende zu schickende SMS-Nachrichten
<code>tcl_running</code>	zeigt an ob die Ablaufsteuerung bereits gestartet wurde
<code>session_removed</code>	zeigt an ob die Sitzung bereits beendet wurde
<code>dtmf_codes</code>	Puffer der letzten DTMF-Signalisierungen
<code>tcl_thread</code>	Thread der Ablaufsteuerung
<code>gw_mutex</code>	MutEx zum Warten auf Gateway-Nachrichten
<code>dtmf_mutex</code>	MutEx zum Warten auf DTMF-Signalisierungen
<code>tcl_fn</code>	Dateiname des Ablaufsteuerungskriptes

Tabelle 4.2: Felder der Datenstruktur `vms_conn`

4.2.3 Ablaufsteuerung

Damit das VMS möglichst flexibel eingesetzt werden kann, ist es nötig, dass eine Ablaufsteuerung der Sitzungen außerhalb des VMS-Modules erfolgt. Dadurch kann ohne Änderung am VMS-Modul der Ablauf einer Mailbox-Verbindung angepasst werden. Für jeden Benutzer kann individuell ein Ablaufsteuerungsskript abgelegt werden. Da es im Allgemeinen nicht nötig ist, für jeden Benutzer individuelle Sitzungsskripte vorzuhalten, wird, falls solch ein Skript für einen Benutzer nicht existiert, auf ein Standardskript zurückgegriffen. Außerdem ist es möglich, für jeden Grund, aus dem eine VMS-Sitzung eingeleitet werden kann (Alerting-, Paging-Timeout oder Mailbox-Anruf), unterschiedliche Skripte zu hinterlegen.

Das Entwickeln einer eigenen Ablaufsteuerung ist für die gegebene Aufgabe nicht nötig, da bereits zahlreiche Implementierungen von Skriptsprachen existieren, die in eigene Programme eingebunden werden können.

Wichtig für die vorliegende Problemstellung war ein einfaches Einbinden des Interpreters in das VMS-Modul, die Unterstützung von Multithreading und eine einfache Schnittstelle zum Informationsaustausch zwischen interpretiertem Skript und VMS-Modul. Die Wahl fiel hier auf die Tcl¹, da sie alle genannten Forderungen erfüllt.

Tcl-Übersicht

Tcl ist ein sehr einfacher und unter einer Open-Source Lizenz vertriebener Interpreter. Tcl bietet einen grundlegenden Sprachumfang mit Variablen, Funktionen und Kontrollstrukturen. Ein besonderes Merkmal ist seine Erweiterbarkeit.

Tcl wurde ursprünglich als wiederverwendbare Kommando-Sprache im *Computer Aided Design* Umfeld erschaffen. Der Interpreter ist in der Programmiersprache *C* geschrieben, was ein einfaches Einbinden in andere Anwendungen ermöglicht.

Verwendung des Tcl-Interpreters

Beim Einbinden des Tcl-Interpreters in das VMS-Modul werden einige selbstdefinierte Variablen aus dem VMS-Modul in das Tcl-Skript exportiert (siehe Tabelle 4.3).

Variable	Bedeutung
<code>vms_msisdn_caller</code>	MSISDN des Anrufers
<code>vms_msisdn_mowner</code>	MSISDN des Mailbox-Besitzers
<code>vms_msisdn_gw</code>	MSISDN zum Abhören der Mailbox

Tabelle 4.3: eingeführte Tcl-Variablen

Zur eigentlichen Ablaufsteuerung wurde eine Reihe von Funktionen definiert, um auf die Informationen aus der Datenstruktur `vms_conn`, die für jede Verbindung angelegt wird, zuzugreifen (siehe Tabelle 4.4).

¹Tool Command Language

Funktion	Bedeutung
<code>vms_announcement_play</code>	spielt eine Ansage ab
<code>vms_debug</code>	schreibt einen Log-Eintrag
<code>vms_dtmf_count</code>	ermittelt die Anzahl der DTMF-Signalisierungen
<code>vms_dtmf_clear</code>	löscht den DTMF-Signalisierungspuffer
<code>vms_dtmf_next</code>	holt nächste DTMF-Signalisierung
<code>vms_dtmf_wait</code>	wartet auf nächste DTMF-Signalisierung
<code>vms_hangup</code>	beendet eine Verbindung
<code>vms_sms</code>	sendet eine SMS
<code>vms_spool_count</code>	ermittelt die Anzahl der Nachrichten in der Mailbox
<code>vms_spool_delete</code>	löscht die aktuelle Nachricht
<code>vms_spool_get_caller</code>	gibt Nummer des Anrufers zurück
<code>vms_spool_get_time</code>	gibt Zeit der Aufnahme der Nachricht zurück
<code>vms_spool_isentry</code>	Test ob Nachricht vorhanden
<code>vms_spool_previous</code>	bewegt Zeiger auf aktuelle Nachricht zur Vorherigen
<code>vms_spool_next</code>	bewegt Zeiger auf aktuelle Nachricht zur Nächsten
<code>vms_spool_play</code>	Spielt aktuelle Nachricht ab
<code>vms_spool_record</code>	nimmt neue Nachricht auf
<code>vms_spool_reset</code>	setzt Zeiger auf aktuelle Nachricht an den Anfang

Tabelle 4.4: eingeführte Tcl-Funktionen

Häufig wiederkehrende Aufgabenstellungen wie das Vorlesen einer Zahl oder Rufnummer lassen sich in einer Tcl-Datei zusammenfassen. Die verschiedenen Skripte, die diese Funktionalität benötigen, können diese Datei dann einbinden.

Vorhandene Tcl-Skripte

Im Rahmen dieser Arbeit wurden mehrere Tcl-Skripte (siehe Anhang D.2.2) erstellt.

Wenn der gerufene Teilnehmer nicht erreichbar ist, wird eine entsprechende Sprachnachricht abgespielt die den Rufenden darüber informiert. Im Anschluß daran wird eine Sprachnachricht aufgenommen und der gerufene Teilnehmer mittels einer SMS-Nachricht darüber informiert.

Das Skript zum Abhören der Mailbox verwendet fertige Sprachdateien, um eine Ansage zusammzusetzen. So ist es u.a. möglich, die Anzahl der vorhandenen Nachrichten vorzu-

lesen. Die Sprachdateien stammen von der *Datenverarbeitung Stadt Pforzheim*², die diese im Rahmen ihres Telefonesystems einsetzen und kostenlos unter einer Open-Source Lizenz zur Verfügung stellen. In der Mailbox kann der Benutzer seine Nachrichten abhören und ausgewählte Nachrichten löschen oder wiederholen lassen.

Dem Benutzer wird unter anderem auch die Anzahl der vorhandenen Sprachnachrichten vorgelesen. Die Funktion für das Vorlesen einer Zahl wurde dabei in die Datei `global.tcl` ausgelagert und kann so auch von anderen Skripten verwendet werden.

4.2.4 Datenhaltung

Die Sitzungsskripte und Sprachdaten werden in einer Verzeichnisstruktur im Dateisystem unter `/var/spool/vms` abgelegt (Abb. 4.1).

```
'-- /var/spool/vms/
  |-- default
  | |-- scripts
  | | |-- global.tcl
  | | |-- VMS_ETC_ALERTING.tcl
  | | |-- VMS_ETC_PAGING.tcl
  | | '-- VMS_ETC_MAILBOX.tcl
  |-- announcements
  |   |-- welcome.al
  |   '-- beep.al
  |-- 333
  | '-- spool
  |   |-- content.xml
  |   |-- rechtulXy
  |   '-- reckasiwS
  '-- 6789
     '-- spool
        '-- content.xml
```

Abbildung 4.1: Verzeichnisstruktur zur Datenhaltung

²<http://www.stadt-pforzheim.de/asterisk/>

Für jede Mailbox existiert in diesem Verzeichnis ein weiteres Unterverzeichnis in dem die spezifischen Sitzungsskripte und Sprachdaten hinterlegt sind. Da es im Allgemeinen nicht nötig ist, für jede Mailbox eigene Sitzungsskripte und Ansagetexte zu erzeugen, gibt es außerdem das Verzeichnis `default`, wo die Vorgabeskripte und Ansagetexte hinterlegt werden.

Innerhalb dieser Verzeichnisse existieren wiederum mehrere Unterverzeichnisse. Das Verzeichnis

- `announcements` enthält die Ansagetexte,
- `scripts` enthält die Sitzungsskripte und
- `pool` enthält die aufgezeichneten Sprachnachrichten

für die jeweilige Mailbox.

announcements Verzeichnis

In diesem Verzeichnis liegen die Ansage-Texte. Der Inhalt dieser Dateien ist bereits fertig kodiert und kann zum Abspielen auf die Datenkanäle direkt ausgegeben werden.

scripts Verzeichnis

Das `scripts` Verzeichnis enthält die Sitzungsskripte für die Mailbox. In Abhängigkeit aus welchem Grund die VMS zum Einsatz kommt, wird das entsprechende Skript verwendet (siehe Tabelle 4.5).

Dateiname	Verwendung
<code>VMS_ETC_ALERTING.tcl</code>	Teilnehmer nimmt Ruf nicht an
<code>VMS_ETC_PAGING.tcl</code>	Teilnehmer nicht eingebucht
<code>VMS_ETC_MAILBOX.tcl</code>	Teilnehmer fragt Mailbox ab

Tabelle 4.5: verwendete Namen der Sitzungsskripte

spool Verzeichnis

Im spool Verzeichnis werden die aufgenommenen Sprachdaten abgelegt. Der Dateiname der Sprachdatei wird dabei durch Funktionen des Betriebssystems generiert. Dabei ist garantiert, dass der Dateiname im Verzeichnis eindeutig ist.

Die Information, welche Aufnahmen verfügbar sind, werden in einer XML-Datei mit dem Namen `content.xml` im gleichen Verzeichnis abgelegt. Neben dem Dateinamen für die Aufnahme werden zusätzliche Informationen wie Zeitpunkt des Anrufes und Nummer des Anrufers gespeichert.

Folgendes Beispiel gibt einen möglichen Inhalt dieser Datei mit zwei Sprachnachrichten an:

```
<spool>
  <call time="1120258163">
    <file name="/var/spool/vms/333/spool/rechtulXy"/>
    <caller number="38611"/>
  </call>
  <call time="1120258209">
    <file name="/var/spool/vms/333/spool/reckasiwS"/>
    <caller number="38611"/>
  </call>
</spool>
```

4.3 MSC-Software

Für die VMS Funktionalität wurden mehrere Schnittstellen in der bestehenden Software-Architektur eingerichtet. Für die Weiterleitung eingehender Anrufe mußten entsprechende Anpassungen im MSC vorgenommen werden.

4.3.1 HLR

Da nicht jeder Teilnehmer des GSM-Versuchsnetzes unbedingt eine Mailbox besitzen soll, wurden die Einträge im HLR um ein weiteres Feld erweitert. Dieses gibt an, ob für

den entsprechende Teilnehmer die Mailbox-Funktion verfügbar ist oder nicht. Einträge im HLR sehen dann wie in Tabelle 4.6 aus.

MSISDN	IMSI	Eingebucht	Roaming	SMS	VMS
333	26288222222207	0	1	1	1
6789	262892000000155	0	1	1	1

Tabelle 4.6: Beispiel für Eintrag im HLR

Für das neue Feld wurde der IEI³ `HLR_IEI_VMS_ALLOWED` eingeführt, um die Information in den `HLR_DATA`-Nachrichten, die vom HLR an das MSC gesendet werden, mit zu übertragen.

4.3.2 MSC_TIMER

In der bisherigen Implementation wurde das `MSC_TIMER`-Modul für das Paging-Timeout verwendet, wenn eine Mobilstation im GSM-Netz gesucht wird. Die Dauer des Klingelns beim Angerufenen war nicht begrenzt - für das VMS ist dies aber nötig. Deshalb wurde das `MSC_TIMER`-Modul so erweitert, das es möglich ist, dem Timer einen Typ zu übergeben, je nachdem ob dieser für das Paging- (`TIMER_TYPE_PAGING`) oder das Alerting-Timeout (`TIMER_TYPE_ALERTING`) verwendet werden soll. Um die Typinformation in den Nachrichten zwischen MSC und `MSC_TIMER` zu übertragen, wurde die neue IEI `TIMER_IEI_TYPE` eingeführt.

4.3.3 MSC

Das MSC hält für jede Verbindung eine Struktur vom Typ `conn_rec` bereit (siehe [Wei00]), in der für den Teilnehmer und die Verbindung relevante Daten gespeichert sind. Dazu gehören u.a. die MSISDN des Rufenden und des Gerufenen oder der verwendete Zeitschlitz. Diese Struktur wurde für das VMS um zwei Einträge erweitert.

- `vms_allowed` gibt an, ob die VMS für diese Verbindung aktiv werden darf. Gesetzt wird dies durch die Auswertung der `HLR_DATA_IND`-Nachricht, die vom HLR an das MSC geschickt wird.

³Information Element Identifier

- `vms_cause` gibt den Grund an, warum eine Verbindung mit dem VMS aufgebaut wurde. Dieses Datum bekommt eine der folgenden Konstanten zugewiesen:
 - `VMS_ETC_UNKOWN` wird als Initialisierung verwendet, wenn noch keine VMS für eine Verbindung aktiv ist.
 - `VMS_ETC_PAGING` zeigt an, dass der Teilnehmer nicht im GSM-Netz eingebucht ist.
 - `VMS_ETC_ALERTING` wird verwendet, wenn der Teilnehmer einen Ruf nicht annimmt.
 - `VMS_ETC_MAILBOX` signalisiert, dass ein Teilnehmer seine Mailbox abruft.

Einige Funktionen im existierenden MSC-Modul mußten für die neuen Abläufe durch das VMS angepasst werden.

- `get_external_event` setzt die Meldungen aus der *Message Queue* in Ereignisse um. Für die neu eingeführten Nachrichten (siehe Tabelle 4.1) mußte die Funktion entsprechend erweitert werden.
- `check_timeout` wird aufgerufen, wenn eine Zeitüberschreitung auftritt. Dies kann passieren, während ein Teilnehmer gepaged oder gerufen wird. In der Funktion muß entschieden werden, ob die VMS gestartet werden soll.
- `get_msisdn_status` wertet die HLR_DATA_IND-Antworten vom HLR aus. Hier muß der Wert für `vms_allowed` in der `conn_rec` Struktur für die Verbindung gesetzt werden.
- `send_setup_info` schaltet die Sprachkanäle zwischen dem PCM30- und dem MVIP-Anschluß auf der SS7-Karte. Für Rufe, die aus dem ISDN-Netz an die VMS weitergeleitet werden ist dies nicht erforderlich.
- In `send_isup_data_req` wird überprüft, ob der Teilnehmer einen Ruf annimmt. Wenn dem so ist, wird die VMS für diese Verbindung deaktiviert, damit spätere Nachrichten vom MSC_TIMER-Modul, die eine Zeitüberschreitung signalisieren, keine VMS-Sitzung für die Verbindung auslösen können.

4.3.4 SMSC

Bisher war es möglich, über das SMSC neue SMS-Nachrichten zu erzeugen. Dies wurde bereits für ein Gateway von E-Mail zu SMS verwendet. Diese Implementierung war jedoch fehlerhaft, Nachrichten für bereits eingebuchte Teilnehmer wurden nicht umgehend ausgeliefert. Dies geschah erst nach einem erneuten Einbuchen des Teilnehmers. Im Rahmen dieser Arbeit wurde diese Unstimmigkeit behoben, da die Funktion zum Senden von Benachrichtigungen der Mailbox verwendet wird. Dabei ist es wichtig, dass der Mailbox-Besitzer die Benachrichtigung über neue Nachrichten zeitnah zugestellt bekommt.

4.4 Gateway-Software

Die Abbildung 4.2 zeigt die Architektur der Gateway-Software nach dem Hinzufügen der Komponenten für das VMS. Wie in [Woh01] und [Eic02] beschrieben, verwendet das Gateway sogenannte *Message Broker*, um mit der Hardware und dem MSC zu kommunizieren.

Abbildung 4.2: Änderung der Architektur von MSC, SIP- und ISDN-Gateway

Die Implementation der mbGSMi- und mbGSMs-Module erlaubt nur eine direkte Kommunikation zwischen MSC-Modul und ISDN- bzw. SIP-Gateway. Damit das VMS-Modul im MSC Meldungen direkt vom SIP-Gateway empfangen kann, war deshalb die Implementation eines neuen mbVMS-Modules notwendig. Dieses leitet Nachrichten, die vom SIP-Gateway über die *Message Queue* des Gateways empfangen werden, über die *Message Queue* des MSC an das VMS-Modul weiter.

4.4.1 SIP-Gateway

Da das SIP-Gateway bereits Funktionen zum Auslesen und Wiedergeben von Sprachdaten enthält, wurde es entsprechend erweitert, um Sprachnachrichten für das VMS aufzunehmen und wiederzugeben. Wie in [Eic02] beschrieben, enthält das SIP-Gateway eine Klasse `Call`, die alle Signalisierungsnachrichten für eine Verbindung abhandelt. Der Zustandsautomat, der durch die Klasse `Call` realisiert wird, wurde durch Einführung neuer Zustände (Abb. 4.3) um das Abspielen und Wiedergeben von einzelnen Sprachnachrichten für das VMS erweitert.

Abbildung 4.3: geänderter Zustandsautomat der Klasse `Call`

Nach dem Eintreffen einer vom VMS-Modul gesendeten `VMS_GW_CONNECTION_REQ`-Nachricht, wird eine neue Instanz der Klasse `Call` vom SIP-Gateway angelegt. Wenn genügend freie Ressourcen vorhanden sind, um die neue Anfrage zu bearbeiten, wird eine `GW_VMS_CONNECTION_ACK`-Nachricht als Antwort an das VMS-Modul gesendet, und

der Zustandsautomat geht in den Zustand `setup_gsm2vms` über. Falls nicht genügend freie Ressourcen vorhanden sind, wird die Anfrage mit einer `GW_VMS_CONNECTION_REJ`-Nachricht an das VMS-Modul abgelehnt. Die begrenzte Ressource ist hier die Anzahl der freien S_0 -Kanäle auf der ISDN-Karte.

Mit dem Erhalt der `MSC_INFO`-Nachricht kann auf den Sprachkanal zugegriffen werden, und die VMS geht damit in den Zustand `callActiveVMS` über. In diesem Zustand wird auf `VMS_GW_PLAYBACK_REQ` und `VMS_GW_RECORD_REQ`-Nachrichten gewartet, die das Abspielen bzw. Aufnehmen von Sprachnachrichten einleiten.

Sobald eine `ISUP_RELEASE`-Nachricht eintrifft, ist die Verbindung beendet und die Instanz kann wieder freigegeben werden. Das Beenden einer VMS-Sitzung wird nie vom SIP-Gateway selbst, sondern immer vom Anrufer oder VMS-Modul ausgelöst.

4.4.2 ISDN-Gateway

In der ursprünglichen Implementation des SIP-Gateways war es nicht, möglich vom ISDN-Netz eine Weiterleitung auf eine SIP-Adresse zu verwenden. Für die Implementation des VMS ist es aber grundlegend, dass auch dessen Funktionalität sowohl aus GSM- als auch aus dem ISDN-Netz zur Verfügung steht. Damit das VMS-Modul im MSC die Aufnahme und das Abspielen von Sprachdaten für eine ISDN-Verbindung realisieren kann, benötigt es die Nummer des B-Kanals des jeweiligen S_0 -Buses, auf den der Ruf geschaltet ist.

Um den B-Kanal zu ermitteln, sendet das MSC an die Instanz der Klasse `Call` für die Verbindung, während diese sich im Zustand `wait_for_gsmConnection` befindet eine `MSC_GW_LAPDID_REQ`-Nachricht. Diese wird mit einer `GW_MSC_LAPDID_IND`-Nachricht beantwortet und enthält die Nummer des B-Kanal des Rufes. Diese Nummer wird dann vom MSC-Modul an das VMS-Modul weitergeleitet. Wenn die Verbindung mit der VMS auf der Seite des Gateways später aufgebaut wird, kann die Nummer des B-Kanales mit übertragen werden.

4.4.3 mbISDN

Der eigentliche Signalisierungs- und Datenaustausch zwischen ISDN-Karte und Gateway findet im mbISDN-Modul des Gateways statt. Hier wurden bereits die Sprachdaten zwi-

schen einem SIP-Teilnehmer und einem GSM-Teilnehmer ausgetauscht. Für das VMS wurde die Funktionalität um das Aufnehmen und Abspielen von Sprachdaten auf Festplatte erweitert.

4.5 Signalisierungsabläufe

4.5.1 Ruf-Aufbau MS-zu-VMS

In Abbildung 4.4 ist der Signalisierungsablauf zum Aufbau einer Verbindung einer MS zu einer MS dargestellt, wenn die gerufene MS nicht eingebucht ist.

Abbildung 4.4: Signalisierung MS-zu-VMS

Bis zum Eintreffen der `MSC_TIMEOUT`-Nachricht vom `MSC_TIMER` hat sich nichts am Signalisierungsablauf geändert. Jetzt wird jedoch die neue `MSC_VMS_IND`-Nachricht vom `MSC` an das `VMS` gesendet, um eine neue `VMS`-Sitzung einzuleiten, nachdem der geru-

fene Teilnehmer als nicht eingebucht festgestellt wurde.

Als Erstes reserviert das VMS entsprechende Ressourcen beim SIP-Gateway (SIP_gw). Wenn dies gelingt, erhält das MSC vom VMS eine VMS_MSC_ACK-Nachricht. Darauf wird der Ruf auf den MVIP-Bus geschaltet und das SIP-Gateway über den verwendeten Kanal auf dem MVIP-Bus über eine MSC_INFO-Nachricht informiert.

Darauf hin sendet das SIP-Gateway für die VMS-Sitzung ein ISUP_ADDRESS_COMPLETE und ISUP_ANSWER Meldung in Form von ISUP_ADDRESS_IND-Nachrichten an das MSC. Damit wird der Ruf in Richtung GSM aufgebaut.

Auf Seiten des ISDN-Karte wird der Sprachkanal aktiviert und geschaltet. Daraufhin wird dem VMS vom SIP-Gateway mitgeteilt (GW_VMS_CONNECTION_EST-Nachricht), dass die Sitzung aufgebaut ist, und das VMS kann die Ablaufsteuerung starten.

4.5.2 Ruf-Aufbau ISDN-zu-VMS

In Abbildung 4.5 ist der Signalisierungsablauf zum Aufbau einer Verbindung aus dem ISDN-Netz zu einem GSM-Teilnehmer dargestellt, wenn der gerufene Teilnehmer nicht eingebucht ist.

Analog zum vorhergehenden Fall werden als erstes die Ressourcen im SIP-Gateway reserviert. Wenn dies mit dem Eintreffen der VMS_MSC_ACK-Nachricht gelungen ist, wird die Nummer des verwendeten B-Kanals des S_0 -Buses, auf dem der Anruf eingegangen ist, vom ISDN-Gateway durch das MSC mit einer MSC_GW_LAPDID_REQ-Nachricht erfragt. Dies wird durch eine MSC_GW_LAPDID_IND-Nachricht beantwortet. Die Nummer des B-Kanals wird dem SIP-Gateway durch eine MSC_INFO-Nachricht übermittelt.

Darauf hin sendet das SIP-Gateway auch hier für die VMS-Sitzung eine ISUP_ADDRESS_COMPLETE und ISUP_ANSWER Meldung in Form von ISUP_ADDRESS_IND-Nachrichten an das MSC. Diese werden an das ISDN-Gateway weitergeleitet, welches das in entsprechende Signalingmeldungen ins ISDN-Netz sendet. Damit ist der Ruf aufgebaut, und mit Aktivierung des Sprachkanales signalisiert das SIP-Gateway über eine GW_VMS_CONNECTION_EST-Nachricht, dass die VMS-Sitzung bereit ist, und das VMS startet die Ablaufsteuerung.

Abbildung 4.5: Signalisierung ISDN-zu-VMS

4.5.3 Ablaufsteuerung

Während eine Sitzung mit dem VMS aufgebaut ist, gibt es nur zwei Arten von Signalisierungsinformationen, die durch das VMS bearbeitet werden müssen. Zum Einen sind das Meldungen zum Abspielen und Aufnehmen von Sprachnachrichten, zum Anderen sind das eventuelle DTMF-Signalisierungen aus dem GSM-Netz.

Abspielen und Aufnahme von Sprachnachrichten

Nach Ruf-Aufbau zum VMS wird die Ablaufsteuerung ausgelöst. Durch die entsprechenden Funktionen im Tcl-Ablaufskript wird dann das Abspielen und Aufnehmen von Sprachnachrichten ausgelöst. Hier soll nur das Abspielen in Abbildung 4.6 dargestellt werden.

Abbildung 4.6: Signalisierung zum Abspielen einer Sprachnachricht

Die Signalisierung zum Aufnehmen von Sprachnachrichten läuft analog und wird durch eine entsprechende `VMS_GW_RECORD_REQ`-Nachricht eingeleitet.

DTMF-Signalisierungen

Die Signalisierung von DTMF-Tönen wird durch das MSC transparent an das VMS weitergeleitet (siehe Abbildung 4.7). Im VMS wird beim Eintreffen einer `DTMF_START`-Nachricht die Tasten-Information in dem DTMF-Puffer der Verbindung abgelegt (siehe Tabelle 4.2) und durch eine `DTMF_START_ACK`-Nachricht bestätigt. Die Verwendung dieser Information hängt dann vom aktuellen Sitzungsskript ab und soll hier nicht weiter beleuchtet werden.

Abbildung 4.7: Signalisierung von DTMF-Tönen

Das Ende des Tastendruckes, der durch `DTMF_STOP`-Nachrichten angezeigt wird, wird im VMS nur bestätigt, aber nicht weiter beachtet, da davon ausgegangen wird, dass nur der einzelne Tastendruck selbst für das Sitzungsskript von Belang ist.

4.5.4 Ruf-Abbau

Der Abbau eines Rufes vom GSM- oder ISDN-Netz zum VMS erfolgt durch die bereits vorhandene Verarbeitung der Signalisierungsmeldungen im SIP-Gateway. Da alle Signalisierungsinformationen über das MSC ausgetauscht werden, wurde dies bei der Verarbeitung von `MSC_RELEASE_IND`-Nachrichten entsprechend angepasst, um bei VMS-Verbindungen an das VMS eine `MSC_VMS_RELEASE_IND`-Nachricht zu übermitteln. Dies ist nötig, da die Signalisierung zwar zwischen SIP-Gateway und GSM- oder ISDN-Netz bereits implementiert war, das VMS jedoch so nicht über das Ende einer Sitzung informiert wurde und somit keine Ressourcen freigegeben werden konnten.

5 Zusammenfassung

Durch die vorliegende Arbeit wurde entsprechend dem generellen Aufbau von GSM-Netzen ein VMS als separates Software-Modul implementiert. Die nötigen Änderungen an den bereits bestehenden Komponenten wurde dabei minimal gehalten und beschränkt sich auf die nötigen Funktionen, um Rufe auf das VMS umzuleiten.

Durch das Auslagern der Geschäftslogik zur Ablaufsteuerung von VMS-Sitzungen in externe Skriptdateien lässt sich das Verhalten des VMS zur Laufzeit einfach anpassen. Durch die Verwendung von DTMF-Signalisierung lassen sich außerdem bekannte Mailbox-Funktionen aus kommerziellen Netzen nachbilden. Diese wurde beispielhaft durch entsprechende Sitzungsskripte veranschaulicht.

Diese Arbeit kann als Grundlage für weitere mögliche Funktionen gesehen werden. In der vorliegenden Implementierung kann die Mailbox eines Benutzers nur durch direkte Änderungen im HLR aktiviert oder deaktiviert werden. Eine individuelle Konfigurationsmöglichkeit über die Funktionen der Mobilstationen oder das VMS wäre hier vorstellbar.

Die vorhandenen Sitzungsskripte liefern nur grundlegende Funktionen, um Nachrichten aufzunehmen und später abzuhören und zu löschen. Hier wären Erweiterungen wie individuelle Ansagetexte für jeden Teilnehmer oder eine Benachrichtigung über neue Sprachnachrichten per E-Mail denkbar. Beim Abhören der Mailbox könnten mit den vorhanden Sprachdateien Funktionen wie das Vorlesen der Nummer des Anrufers oder das Datum der Aufnahme realisieren.

A Abkürzungsverzeichnis

AuC	Authentication Centre
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CONNH	Connection Handler
DTMF	Dual Tone Multi Frequency
EIR	Equipment Identity Register
FSM	Finite State Machine
GMSC	Gateway Mobile Switching Centre
GSM	Global System for Mobile communications
HLR	Home Location Register
IEI	Information Element Identifier
IP	Internet Protocol
ISC	International Switching Centre
ISDN	Integrated Services Digital Network
MMI	Man Machine Interface
MS	Mobile Station
MSC	Mobile Switching Centre
MSISDN	Mobile Subscriber Integrated Services Digital Networknumber
MVIP	Multi Vendor Integration Protocol
OMC	Operation and Maintenance Centre
PDN	Public Data Networks
PLMN	Public Lands Mobile Network
PSTN	Public Switched Telephone Network
SIP	Session Initiation Protocol
SMS	Short Message Service
SMSC	Short Message Service Centre

SS7	Signalling System No. 7
Tel	Tool Command Language
UMTS	Universal Mobile Telecommunication System
VLR	Visitor Location Register
VMS	Voice Mail System
VoIP	Voice over IP

B Abbildungsverzeichnis

2.1	GSM-Versuchsnetz der TU-Dresden	5
2.2	Software-Stapel MSC	8
3.1	wesentliche Komponenten eines GSM-Netzes	11
3.2	DTMF-Signalisierungsnachrichten im GSM-Netz	13
4.1	Verzeichnisstruktur zur Datenhaltung	19
4.2	Änderung der Architektur von MSC, SIP- und ISDN-Gateway	24
4.3	geänderter Zustandsautomat der Klasse Call	25
4.4	Signalisierung MS-zu-VMS	28
4.5	Signalisierung ISDN-zu-VMS	30
4.6	Signalisierung zum Abspielen einer Sprachnachricht	31
4.7	Signalisierung von DTMF-Tönen	32

C Tabellenverzeichnis

4.1	neue Nachrichten Primitiven	15
4.2	Felder der Datenstruktur <code>vms_conn</code>	16
4.3	eingeführte Tcl-Variablen	17
4.4	eingeführte Tcl-Funktionen	18
4.5	verwendete Namen der Sitzungsskripte	20
4.6	Beispiel für Eintrag im HLR	22

D Anhang

D.1 Libero-Dateien

D.1.1 VMS

```
!=====
!  
!-----
! Project: Linux MSC Project
! Authors: Thomas Liske
!          (C) 2005
!-----
! Module : vms
! Description: Voice Message System
!  
!=====
```

```
-path="$PATH:../include"  
-schema=msc_schema.lr
```

After-Init:

```
(--) Ok -> Idle  
      +
```

Idle:

```
(--) Msc-Vms-Ind  
      + connection-ind
```

```
(--) Msc-Vms-Release-Ind
    + connection-release

(--) Gw-Vms-Connection-Ack
    + connection-acknowledge

(--) Gw-Vms-Connection-Rej
    + connection-reject

(--) Gw-Vms-Connection-Est
    + proceed-connection

(--) Gw-Vms-Playback-Fin
    + proceed-connection

(--) Gw-Vms-Playback-Err
    + proceed-connection

(--) Gw-Vms-Record-Fin
    + proceed-connection

(--) Gw-Vms-Record-Err
    + proceed-connection

(--) Gw-Call-Req
    + mailbox-req

(--) MSC-Info
    + mailbox-fwd-msg2gw

(--) ISUP-DATA-REQ
```

```
+ mailbox-fwd-msg2gw
+ connection-release
```

```
(--) DTMF-START
+ send-dtmf-ack
```

```
(--) DTMF-STOP
+ send-dtmf-ack
```

```
(--) Error
+
```

D.1.2 MSC

```
!=====
! $Header: /mnt/msc/cvsroot/msc/msc/msc.l,v 1.17 2000/06/13 16:15:26 matt Exp $
!-----
! Project: Linux MSC Project
! Authors: Matthias Lemke & Matthias Weidle
!         (C) 1999, 2000
! Authors: Thomas Liske
!         (C) 2005
!-----
! Module :
! Description:
!
! $Date: 2000/06/13 16:15:26 $
!=====
```

```
-path="$PATH:../include"
-schema=msc_schema.lr
```

After-Init:

(--) Ok

-> Id

+

Idle:

(--) Msc-Initial-Address-Ind

->

+ Write-Data-Row

+ Get-Number

(--) Msc-SMS-Initial-Address-Ind

->

+ Write-SMS-Data-Row

+ Get-SMS-Number

(--) Msc-SMSC-Initial-Address-Ind

->

+ Get-SMS-Number

(--) Msc-Paging-Resp

->

+ Write-Data-Row

+ Check-Kind-of-Service

! UE

(--) Msc-Timeout

->

+ Check-Timeout

(--) Msc-Paging-Timed-Out

->

+ Send-Isup-Release

+ Del-Data-Rows

! end UE

(--) Msc-Call-Paging-Resp

-> CheckCall

+ Check-Call

(--) Msc-SMS-Paging-Resp

->

+ Send-SMS-PID-Request

```

(-- ) CP-DATA                                     ->
      + Store-CP-Data

(-- ) Received-CP-DATA                             ->
      + Send-received-CP-Data

(-- ) Msc-Release-CPDATA-Ind                       ->
      + SMSC-Send-CPDATA
      + Get-Cause-Value
      + Send-Isup-Release-Other
      + Del-Data-Rows

(-- ) Msc-Release-Ind                               ->
      + Get-Cause-Value
      + Send-Isup-Release-Other
! by ThoLI/VMS
! tell VMS that call was released
      + Send-Msc-VMS-Release
!
      + Del-Data-Rows

(-- ) Msc-Set-Pcm                                   ->
      + Mark-Pcm-Channel

(-- ) SMSC-Send-CPDATA                             ->
      + SMSC-Send-CPDATA

(-- ) SMSC-Send-MEM-Available                       ->
      + SMSC-Send-MEM-Available

(-- ) SMSC-CP-DATA                                 ->
      + Store-SMSC-CP-DATA
      + Send-received-CP-Data

(-- ) SMSC-NO-CP-DATA                              ->

```

```

        + Send-SMS-No-Data-Ind

(-- ) SMSC-SEND-More-SMS          ->
        + Send-More-SMS

(-- ) Isup-Data-Ind                ->
        + Send-Isup-Data-Req

(-- ) Hlr-Data-Ind                 -> Number-Dis
        + Get-MSISDN-Status

(-- ) Hlr-SMS-Data-Ind             -> Number-Discrim
        + Get-SMS-MSISDN-Status

(-- ) Handover-Info                -> Idle
        + Get-Timeslot

(-- ) Bssmap-Ho-Ind                -> Idle
        + Check-Ressources

(-- ) Msc-Register-Gateway         ->
        + Register-Gateway

(-- ) Msc-Unregister-Gateway       ->
        + Unregister-Gateway

(-- ) Gw-Call-Ack                  -> CheckCall
        + Write-Data-Row
        + Check-Gw-Call

(-- ) Gw-Call-Reject               ->
        + Send-Isup-Release
        + Del-Data-Rows

```

```

! by 2n & ThoLI/VMS
! tell VMS to take the call

```

(--) Start-VMS -> Idle
+ Send-Msc-VMS-Start

(--) VMS-Msc-Ack
+ vms-add-connids
+ request-gw-lapdid

(--) GW-Msc-Lapdid-Ind
+ get-gw-lapdid
+ send-setup-info

(--) VMS-Msc-Busy
+ VMS-is-Busy

(--) DTMF-START
+ Forward-DTMF-Ind

(--) DTMF-START-ACK
+ Forward-DTMF-Ack

(--) DTMF-STOP
+ Forward-DTMF-Ind

(--) DTMF-STOP-ACK
+ Forward-DTMF-Ack

(--) Error -> Id
+

(--) Ok -> Id
+

Number-Discrim:

(--) Number-Local -> Idle
+ Send-Bssmap-Paging

```

        + Send-Paging-Timer-Set
        + Send-Pcm-Info
    (-- ) SMS-Number-Local                -> Idle
        + Send-SMS-Bssmap-Paging
        + Send-Paging-Timer-Set
    (-- ) Send-CP-DATA-in-Call            -> Idle
        + Send-received-CP-Data-1
    (-- ) Number-Not-Local                ->
        + Check-Routing
    (-- ) SMS-Number-Not-Local            -> Idle
        + Number-unknown
    (-- ) Number-not-active                -> Idle
        + Send-Isup-Release
        + Del-Data-Rows
    (-- ) SMS-Number-not-active            -> Idle
        + SMSC-Send-CPDATA
!        + Send-SMS-Disconnect
    (-- ) Number-Not-Found                -> Idle
        + Send-Isup-Release
        + Del-Data-Rows
    (-- ) Number-busy                      -> Idle
        + Send-Isup-Release
        + Del-Data-Rows
    (-- ) PCM-fail                          -> Idle
        + Send-Isup-Release
        + Del-Data-Rows
    (-- ) Route-To-Gateway                 -> Idle
        + Send-Gateway-Setup
        + Send-Pcm-Info
! by 2n & ThoLI/VMS
! tell VMS to take the call
    (-- ) Start-VMS                        -> Idle
        + Send-Msc-VMS-Start
    (-- ) Error                             -> Idle
        +

```

```

CheckCall:
    (--) PCM-fail                                -> Idle
        + Send-Isup-Release
        + Send-Isup-Release-Other
        + Del-Data-Rows
    (--) Call-Ok                                  -> Idle
        + Send-Setup-Info
    (--) Call-Mailbox                             -> Idle
        + Start-Mailbox
    (--) Call-Fail                                -> Idle
        + Send-Isup-Release
        + Del-Data-Rows
    (--) Error                                    -> Idle
        +

```

```
! End $Source: /mnt/msc/cvsroot/msc/msc/msc.l,v $
```

D.2 Tcl

D.2.1 Funktionsübersicht

- `vms_announcement_play <Dateiname>`
Spielt die Ansage die als Parameter `Dateiname` übergeben wird ab. Der `Dateiname` wird dabei relativ zum `announcements`-Verzeichnis interpretiert.
- `vms_debug <Parameter>*`
Erzeugt einen Eintrag in der Log-Datei. Die übergebenen Parameter werden dabei mit ausgegeben.
- `vms_dtmf_count`
Gibt die Anzahl der DTMF-Signalisierungen im Puffer zurück.
- `vms_dtmf_clear`
Leert den DTMF-Signalisierungspuffer.

- `vms_dtmf_next`
Gibt die nächste DTMF-Signalisierung aus dem Puffer zurück. Wenn keine Signalisierung vorhanden ist, wird eine leere Zeichenkette zurückgegeben.
- `vms_dtmf_wait`
Falls der DTMF Puffer nicht leer ist kehrt die Funktion sofort zurück. Ansonsten wird gewartet bis eine DTMF-Signalisierung verfügbar ist.
- `vms_hangup`
Beendet eine VMS-Sitzung.
- `vms_sms [<MSISDN>] <Text>`
Sendet eine SMS mit dem Inhalt des Parameters `Text` an die Nummer die im Parameter `MSISDN` übergeben wurde. Falls der Parameter `MSISDN` weggelassen wird, wird die Nachricht an den Besitzer der Mailbox geschickt.
- `vms_spool_count`
Gibt die Anzahl der verfügbaren Sprachnachrichten in der Mailbox zurück.
- `vms_spool_delete`
Löscht die aktuelle Sprachnachricht. Die nächste vorhandene Sprachnachricht wird die aktuelle Sprachnachricht.
- `vms_spool_get_caller`
Liefert die Rufnummer des Anrufers für die aktuelle Sprachnachricht zurück. Falls keine Nummer übermittelt wurde, wird eine leere Zeichenkette zurückgegeben.
- `vms_spool_get_time`
Gibt die Zeit der Aufnahme der aktuellen Sprachnachricht zurück. Die Zeit wird in Sekunden seit dem 01. Januar 1970 um 00:00:00 UTC zurück.
- `vms_spool_isentry`
Gibt 1 zurück, wenn eine aktuelle Sprachnachricht verfügbar ist.
- `vms_spool_previous`
Bewegt den Zeiger auf die aktuelle Sprachnachricht zur Vorherigen.
- `vms_spool_next`
Bewegt den Zeiger auf die aktuelle Sprachnachricht zur Nächsten.

- `vms_spool_play`
Spielt aktuelle Sprachnachricht ab.
- `vms_spool_record`
Nimmt neue Sprachnachricht auf.
- `vms_spool_reset`
Setzt den Zeiger auf die aktuelle Nachricht auf den erste verfügbare Sprachnachricht.

D.2.2 Standardskripte

`global.tcl`

```
# Liest den Parameter als Zahl vor
proc play_number {number} {
    if {$number==0} {
        set number 0
    } else {
        set number [string trimleft $number 0]
    }

    if {$number >= 1000000} {
        play_number_millions $number
    } else {
        if {$number >= 1000} {
            play_number_thousands $number
        } else {
            if {$number >= 100} {
                play_number_3 $number
            } else {
                play_number_2 $number
            }
        }
    }
}

# Hilfsfunktion um Zahlen 0-99 vorzulesen
```

```

proc play_number_2 {number} {
    if { $number < 21 } {
        vms_announcement_play de/digits/$number.al
    } else {
        if { [string index $number end] != 0 } {
            vms_announcement_play de/digits/[string index $number end]-and.al
        }
        vms_announcement_play de/digits/[string index $number 0]0.al
    }
}

```

Hilfsfunktion um Zahlen 100-999 vorzulesen

```

proc play_number_3 {number} {
    if {[string index $number 0]==1} {
        vms_announcement_play de/digits/ein.al
    } else {
        vms_announcement_play de/digits/[string index $number 0].al
    }
    vms_announcement_play de/digits/hundred.al
    if {[string index $number 1]==0} {
        if {[string index $number 2]>0} {
            play_number_2 [string index $number 2]
        }
    } else {
        play_number_2 [string range $number 1 2]
    }
}

```

Hilfsfunktion um Zahlen 1000-999999 vorzulesen

```

proc play_number_thousands {number} {
    if {[string length $number]==6} {
        play_number [string range $number 0 2]
        set rest [string range $number 3 5]
    }
    if {[string length $number]==5} {

```

```

        play_number [string range $number 0 1]
        set rest [string range $number 2 4]
    }
    if {[string length $number]==4} {
        if {[string index $number 0]==1} {
            vms_announcement_play de/digits/ein.al
        } else {
            play_number [string index $number 0]
        }
        set rest [string range $number 1 3]
    }

    vms_announcement_play de/digits/thousand.al
    if {[expr $rest]>0} {
        play_number $rest
    }
}

# Hilfsfunktion um Zahlen 1000000-999999999 vorzulesen
proc play_number_millions {number} {
    if {[string length $number]==9} {
        play_number [string range $number 0 2]
        set rest [string range $number 3 8]
    }
    if {[string length $number]==8} {
        play_number [string range $number 0 1]
        set rest [string range $number 2 7]
    }
    if {[string length $number]==7} {
        if {[string index $number 0]==1} {
            vms_announcement_play de/digits/eine.al
        } else {
            play_number [string index $number 0]
        }
        set rest [string range $number 1 6]
    }
}

```

```

}

if {[string length $number]==7 && [string index $number 0]==1} {
    vms_announcement_play de/digits/million.al
} else {
    vms_announcement_play de/digits/millions.al
}
if {[expr $rest]>0} {
    play_number $rest
}
}

```

Skript für Alerting- und Paging-Timeout

```

vms_announcement_play welcome
vms_announcement_play beep
vms_sms "Neuer Nachricht von $vms_msisdn_caller auf der Mailbox verfügbar.\
Rufen Sie $vms_msisdn_gw an um die Nachricht abzuhören!"
vms_spool_record
vms_hangup

```

Skript für Mailbox-Abfrage

```

source "/var/spool/vms/default/scripts/global.tcl"

# Anzahl Sprachnachrichten ermitteln
set COUNT [vms_spool_count]

# Ansage: "Sie haben"
vms_announcement_play de/vm-youhave.al

if { $COUNT == 0 } {
    # Ansage: "Keine weiteren Nachrichten"
    vms_announcement_play de/vm-no.al
} else {

```

```

if { $COUNT == 1 } {
    # Ansage: "eine"
    vms_announcement_play de/digits/eine.al
} else {
    # Zahl $COUNT ansagen
    play_number $COUNT
}
}

if { $COUNT != 1 } {
    # Ansage: "Nachrichten"
    vms_announcement_play de/vm-messages.al
} else {
    # Ansage: "Nachricht"
    vms_announcement_play de/vm-message.al
}

if { $COUNT > 0 } {
    vms_spool_reset
    set i 1
    set instr 0
    while {[vms_spool_isentry]} {
        # Ansage: "Nachricht"
        vms_announcement_play de/vm-message.al

        # Zahl $i ansagen
        play_number $i

        # Sprachnachricht abspielen
        vms_spool_play

        # DTMF-Puffer leeren
        vms_dtmf_clear
        set eol 0
        while {$eol == 0} {

```

```

set eol 1

if {$instr} {
    # Ansage: Tastenbelegung
    vms_announcement_play de/vm-msginstruct.al
    set instr 0
} else {
    # Ansage: Tastenbelegung (kurz)
    vms_announcement_play de/vm-helpexit.al
}

# auf DTMF-Ton warten
vms_dtmf_wait
set code [vms_dtmf_next]
# DTMF-Ton auswerten
switch $code {
    "0" {
        # Nachricht loeschen
        vms_spool_delete
        # Ansage: "Nachricht geloescht"
        vms_announcement_play de/vm-deleted.al
        set i [ expr $i + 1 ]
    }
    "4" {
        # vorherige Nachricht abspielen
        vms_spool_previous
        set i [ expr $i - 1 ]
    }
    "5" {
        # Nachricht wiederholen
    }
    "6" {
        # nachste Nachricht
        vms_spool_next
        set i [ expr $i + 1 ]
    }
}

```

```
    }
    "#" {
        # Ansage: "Auf Wiedersehen"
        vms_announcement_play de/vm-goodbye.al
        # Mailbox verlassen
        vms_hangup
        exit
    }
    default {
        set instr 1
        set eol 0
    }
}
}
```

```
# Ansage: "Keine weiteren Nachrichten"
vms_announcement_play de/vm-nomore.al
}
```

```
# Ansage: "Auf Wiedersehen"
vms_announcement_play de/vm-goodbye.al
vms_hangup
```

E Literaturverzeichnis

- [Bas99] BASTEK, Markus: *Nutzungsmöglichkeiten eines GSM-Base-Station-Subsystems zur Demonstration von GSM-spezifischen Abläufen*, TU-Dresden, Diplomarbeit, März 1999
- [Eic02] EICHLER, Uwe: *VoIP-Gateway für das GSM Versuchsnetzwerk der TU Dresden*, TU-Dresden, Diplomarbeit, Juni 2002
- [ETS96a] ETSI, TC-SMG: Digital cellular telecommunication system; Support of Dual Tone Multi-Frequency signalling (DTMF) via the GSM system (GSM 03.14) / European Telecommunications Standards Institute. 1996 (Reference TS/SMG-030314Q). – Technical Specification
- [ETS96b] ETSI, TC-SMG: Digital cellular telecommunications system (phase 2+); Abbreviations and acronyms (GSM 01.04 version 5.0.1) / European Telecommunications Standards Institute. 1996 (Reference DTR/SMG-010104Q). – Technical Report
- [Gue01] GUERRA, Fernando G.: *Implementation von Short Message Service an GSM Versuchsnetzwerk der TU Dresden*, TU-Dresden, Diplomarbeit, April 2001
- [Lem00] LEMKE, Matthias: *Implementierung grundlegender "Mobile Switching Center-Funktionen für das GSM-Versuchsnetzwerk der TU-Dresden*, TU-Dresden, Diplomarbeit, Juni 2000
- [Wei00] WEIDLE, Matthias: *Implementierung grundlegender "Mobile Switching Center-Funktionen für das GSM-Versuchsnetzwerk der TU-Dresden*, TU-Dresden, Diplomarbeit, August 2000
- [Woh01] WOHLFAHRT, Robert: *ISDN-Anbindung des GSM-Versuchsnetzwerkes der TU-Dresden*, TU-Dresden, Diplomarbeit, September 2001