

Funkrufsysteme als redundante verteilte Anwendungen

Michael Heinz

Scall und Skyper, die Funkrufdienste der Deutschen Telekom Mobilnet GmbH sind als verteilte Anwendungen in Client-Server-Technik realisiert. Solche Radio-Paging-Systeme sind nach den Prinzipien des Open Distributed Processing (ODP) aufgebaut. Als Middleware wird ANSAware/CORBAware eingesetzt. Die Middleware-Mechanismen werden durch eine Client Application Interface Library gekapselt und damit von der eigentlichen Applikation getrennt; diese Zwischenschicht ist auch für die Redundanzumschaltung zuständig. Die logische Strukturierung des Gesamtsystems und die klare Definition der Schnittstellen ermöglichen es, sehr effizient solche komplexen Applikationen zu realisieren.

1 Funkrufsysteme und ihre Aufgabe

Scall und Skyper, die Funkrufdienste der T-Mobil (Deutsche Telekom Mobilnet GmbH), ermöglichen die Einweg-Kommunikation zu kleinen tragbaren Empfängern und erlauben es jedermann, von jedem Telefon aus eine individuelle Nachricht an den Funkruf-Empfänger eines Teilnehmers von Scall bzw. Skyper zu senden.

Bei Scall sind nur Ziffern-Nachrichten möglich. Skyper bietet auch Text-Nachrichten und zusätzlich zu den individuellen Mitteilungen die Aussendung von Nachrichten von allgemeinem Interesse (News, Politik, Wetter, Lottozahlen, regionale Kinoprogramme, Börsenkurse etc.) an alle Empfänger oder an Benutzergruppen.

Ein Funkrufsystem besteht aus verschiedenen Zugangssystemen (Bild 1). Verarbeitungssysteme nehmen Aufträge von den Zugangssystemen entgegen, prüfen beispielsweise in der Teilnehmer-Datenbank, ob die angegebene Teilnehmernummer und die Adresse eines Funkrufempfängers existiert, und sorgen für die Funkruffaussendung durch Weitergabe an die Aussendesysteme. Obwohl die Zugangssysteme sich je nach Zugangsart gegenüber dem Benutzer sehr unterschiedlich verhalten können, sind deren Anforderungen an Verarbeitung, Datenbankzugriff und Aussendung weitgehend identisch.

2 Architektur eines Funkrufsystems

Ein Funkrufsystem lässt sich in drei getrennte logische Ebenen mit klaren Schnittstellen aufteilen: (1) Domäne der Zugangssysteme, (2) Verarbeitungs- und

Datenbank-Domäne, (3) Aussende-Domäne. Es bietet sich deshalb an, das Zusammenspiel der drei Ebenen mit Client-Server-Technik zu realisieren. Eine Ebene des Systems stellt der nächsten Ebene ihre Server-Objekte zur Verfügung.

Ein Zugangssystem ist Client der Zugangsdienste. Die Systeme der Verarbeitungsebene bieten für alle Systeme der Zugangsebene die spezifischen Zugangsdienste. Typische Zugangsdienste sind: "Prüfe Teilnehmernummer", "Funkruf aussenden an Teilnehmernummer". In ähnlicher Weise sind zwischen Verarbeitungs- und Aussende-Ebene die Aussendedienste definiert.

3 Systemstruktur von Scall und Skyper

Aus den Anforderungen an die Dienste Scall und Skyper ergibt sich die in Bild 2 gezeigte Systemstruktur. Die Scall-Zugangssysteme wenden sich als

Michael Heinz ist Projektleiter für das T-Mobil-Projekt "Skyper" bei der ATM Computer GmbH, Konstanz, im Systembereich für Kommunikationsanwendungen. Er hat Elektrotechnik und Nachrichtenverarbeitung an der Technischen Hochschule Darmstadt studiert und war anschliessend in der Entwicklung von Betriebssystemen und Kommunikations-Grundsoftware tätig. Nach Gründung der ATM Computer GmbH im Jahr 1980 war er als Hauptabteilungsleiter für Kommunikations-Software auch für Radio-Paging-Systeme zuständig. Die bekanntesten Paging-Systeme der ATM sind *Telepage Swiss* der Schweizer PTT, in Deutschland *Cityruf*, *Scall* und *Skyper*.
<h1@atm.computer.de>

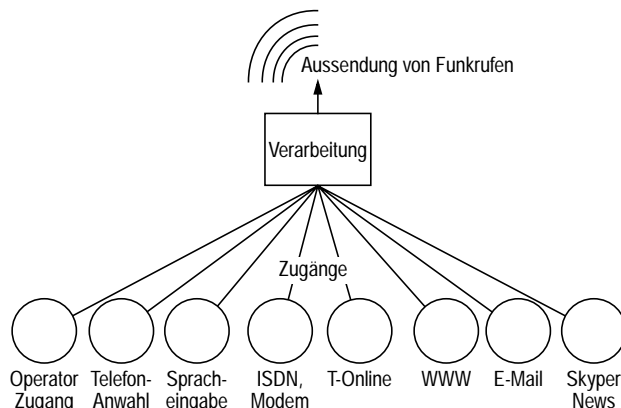


Bild 1: Funkrufsystem

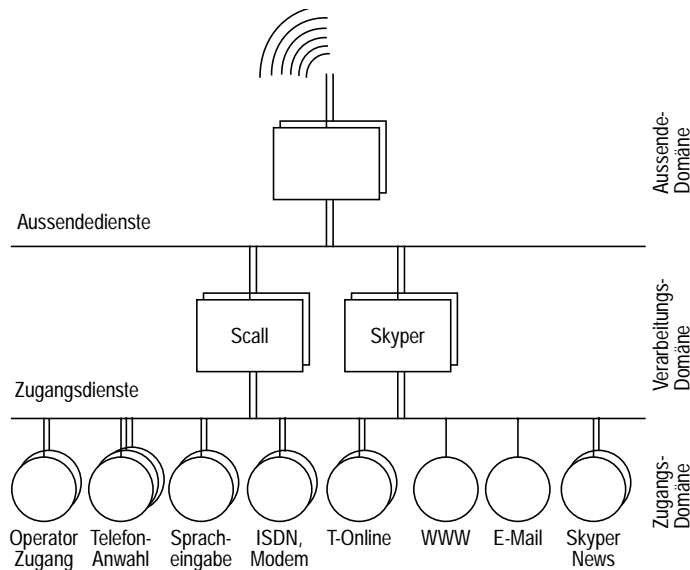


Bild 2: Systemstruktur von Scall und Skyper

Clients an das Scall-Verarbeitungssystem, das die Scall-Zugangsdienste (z.B. Scall-Teilnehmer-Prüfung, Scall-Teilnehmer-Rufaussendung) zur Verfügung stellt. In gleicher Weise implementiert das Skyper-Verarbeitungssystem die Skyper-Zugangsdienste (z.B. Skyper-Teilnehmer-Prüfung, Skyper-Teilnehmer-Rufaussendung, Skyper-Programmaussendung) für die Skyper-Zugangs-Clients.

Bild 2 zeigt die logische Struktur: Kreise und Rechtecke sind Clients und Server und die Verbindungslinien sind Dienstschnittstellen; die logische Verteilung wird durch die verwendete ODP-Middleware erreicht. Es wäre durch die Ortsunabhängigkeit der Middleware ohne weiteres möglich (und wird für Applikationstests sogar genutzt), die verteilte Applikation auch auf einem einzigen Rechner ablaufen zu lassen.

Aus Gründen der Ausfallsicherheit und der Lastverteilung wurde eine der logischen Struktur entsprechende Hardware-Struktur gewählt: die Kreise und Rechtecke sind zumeist separate Rechner und die Verbindungslinien sind Ethernet-LAN-Verbindungen.

4 Client Application Interface

Als Middleware zur Verteilung der Applikation wurde ANSAware eingesetzt. ANSAware ist in einem ESPRIT-Projekt entstanden und war eine der

ersten stabil verfügbaren Verteilplattformen, die zur Architektur des ODP-Standards (Open Distributed Processing) konform war [Zweiacker 95]. Es lagen im eigenen Haus gute Erfahrungen bei der Erstellung grosser verteilter Anwendungen mit ANSAware vor [Debski 96]. Die ursprüngliche ANSAware hatte eine eigene ANSA-IDL (Interface Definition Language), es wurde ein eigenes RPC-Protokoll REX (Remote Execution Protocol) eingesetzt, und die ANSAware hatte schon von Anfang an einen sogenannten Trader, einen Name-Service mit grossem Funktionsumfang (mit einer Sprache zur Formulierung spezieller Objekteigenschaften, sog. Properties), der dem jetzt definierten CORBA-Trader sehr ähnlich ist.

In einer Applikation der beschriebenen Art werden nur statische Aufrufschnittstellen benötigt. Die Schnittstellen der Zugangsdienste und der Aussendedienste sind in der ANSA-IDL (Interface Definition Language) definiert. Die ANSAware-Aufrufe werden jedoch nicht direkt in der Anwendung benutzt; die Dienste werden vielmehr in Library-Aufrufe des Client Application Interface (CAI) "verpackt". Der Anwendungsprogrammierer eines Zugangssystems ruft beispielsweise zur Prüfung einer Teilnehmernummer die Funktion "PAGER_VALID" der Zugangs-CAI-Library auf, die den gewünschten Zu-

gangsdienst "Prüfe Teilnehmernummer" für ihn transparent "irgendwo" erbringen lässt.

Das Einziehen dieser Zwischenschicht von "Library-Objekten" hat folgende Gründe:

- es ist eine Vereinfachung für die Anwendungsprogrammierung: der Programmierer eines Client befasst sich nur mit seinen Anforderungen und nicht mit der verwendeten Middleware, deren IDL-Syntax und deren Aufrufkonventionen. Das ist insbesondere wichtig für das Zugangs-CAI, da dort für Spezialanforderungen (Spracherkennung, Sonderprotokolle wie T-Online etc.) auch Fremdfirmen eingesetzt werden sollen.
- die Applikation ist entkoppelt von der Art der eingesetzten Middleware; dies eröffnet einen Migrationspfad zu CORBA, der weiter unten noch diskutiert wird,
- die Library-Zwischenschicht enthält Erweiterungen für die Redundanzumschaltung,
- trotz dieser wesentlichen Erweiterung an Funktionalität und Komfort entsteht durch das Einziehen der Library-Zwischenschicht praktisch kein zusätzlicher Overhead.

5 Redundante verteilte Anwendung

Die Anforderungen an ein Funkrufsystem bezüglich Ausfallsicherheit sind sehr hoch: es muss praktisch immer betriebsbereit sein und darf auch bei Störungen, z.B. bei Hot-Standby-Umschaltung, keine bereits angenommenen und vom Benutzer bezahlten Funkrufe verlieren. Alle Rechner sind deshalb mindestens doppelt vorhanden; die Platten werden durch RAID-Systeme gedoppelt, die Datenbanken werden auf die (Doppel-)Platten eines Slave-Systems gespiegelt, sind also vierfach vorhanden. Die Redundanzforderung wird auf zwei Arten gelöst:

- Hot-Standby-Betrieb eines Doppelsystems mit Master-Slave-Umschaltung; die Systeme überwachen sich gegenseitig und schalten notfalls automatisch um. Diese Variante wird immer dann eingesetzt, wenn zentrale Betriebsmittel benötigt werden, insbesondere für die Systeme der Verarbei-

tungs- und Datenbankebene. Es gibt also einen Scall-Master-Verarbeitungsrechner und einen zugehörigen Scall-Slave, einen Skyper-Master und einen Skyper-Slave (Bild 2).

- Parallele, unabhängig voneinander arbeitende Rechner, die für die Zugangssysteme auch mehr als zweifach vorhanden sein können. Beispielsweise wirkt sich bei mehreren Telefon/Sprach-Zugangssystemen, die parallel über eine einheitliche Telefonnummer erreicht werden, der Ausfall eines Systems nur in einer Reduktion der gleichzeitig möglichen "Gespräche" aus.

Wie erwähnt, ist die CAI-Library-Zwischenschicht auch für die Umschaltung der Client-Zugriffe auf einen neuen Master zuständig. Nach einem Ausfall eines Servers oder nach einem Master-Slave-Wechsel erkennt ein betroffener Client den Ausfall, beschafft sich über den Trader den benötigten Dienst (eine Interface-Referenz oder Objekt-Referenz des benötigten Dienstes mit gewissen Eigenschaften z.B. "System ist Master") neu und wiederholt den Auftrag. Dies geschieht innerhalb der Library und, abgesehen von einer zeitlichen Verzögerung, ohne dass die Applikation es merkt.

Die Ausführungssemantik eines RPC ist üblicherweise "at most once", d.h. bei positivem Ergebnis wurde der Dienst genau einmal ausgeführt, bei negativem Ergebnis wurde der Dienst entweder nicht oder einmal ausgeführt (z.B. bei Störung des Quittierungskanals und Zeitüberwachung der Quittung). Die Auftragswiederholung durch die Client-Library bei vermutetem Server-Ausfall könnte in sehr seltenen Extremfällen die Ausführungssemantik zu "at most twice" verändern: bei Verzögerung der Server-Reaktion durch Überlast auf dem LAN oder auf dem Server könnte die OK-Quittung des Servers erst nach Ablauf der Überwachungszeit eintreffen und der Auftrag würde durch die Wiederholung ein zweites Mal (an den gleichen Server) gestellt.

Zur Vermeidung dieser doppelten Ausführung müssten eigentlich Transaktionsdienste eingeführt werden. Die Dienste in Funkrufsystemen sind weit-

gehend unempfindlich gegenüber der seltenen doppelten Ausführung: schlimmstenfalls wird ein Funkruf zweimal ausgesendet, doppeltes Lesen in der Teilnehmer-Datenbank bei der Teilnehmernummern-Prüfung ist unschädlich; sogar der Grossteil der schreibenden Datenbankzugriffe ist von einer Verdopplung nicht schädlich betroffen (z.B. "Follow Me"-Dienst zur Veränderung der Aussenderegion für einen Teilnehmer). Für einige wenige Dienste kann die mehrfache Ausführbarkeit durch kleine Änderungen der Anwendung erreicht werden (z.B. der "Pin-Ändern"-Dienst erkennt bei der Wiederholung bereits die neu einzutragende PIN als alte PIN und lehnt den Auftrag nicht wegen falscher PIN ab). Die Belastung des Normalfalls durch einen Transaktionsdienst für praktisch nie vorkommende Fälle kann vermieden werden.

Die Redundanzumschaltung ist somit durch einen trivialen Algorithmus in der CAI-Library zu erreichen, bei dem die Eigenschaft des Traders, Dienst-Schnittstellen über Properties weiter zu qualifizieren, ausgenutzt wird.

6 CORBA

Die Entkopplung der Anwendung von der eingesetzten Middleware durch die CAI-Library-Zwischenschicht ermöglicht eine Migration zu CORBA. Die ANSAware wurde mittlerweile "corbaisiert" zur CORBAware, d.h. die ANSA-IDL wurde durch die CORBA-IDL ersetzt. Es werden die CORBA-Aufrufe zur Verfügung gestellt. Als RPC kann der ANSA-REX anstelle des CORBA-IIOP beibehalten werden.

Für die CORBA-Umstellung wird nur die derzeitige CAI-Library (intern ANSAware-basiert mit ANSA-IDL und ANSA-Aufrufen) durch eine funktionell gleiche CORBA-basierte CAI-Library mit CORBA-IDL und CORBA-Aufrufen ersetzt. Die Applikation ist nicht betroffen, da sie ja nur CAI-Library-Calls absetzt. Die Interoperabilität zu den vorhandenen Systemen wird durch Beibehalten des ANSA-REX (Remote Execution Protocol), als CORBA-ESIOP (Environment Specific Interobject Protocol) unter CORBAware erreicht. Weiterhin wird auch die Funktionalität des

Abkürzungen:

ANSA	Advanced Network Systems Architecture
CAI	Client Application Interface
CORBA	Common Object Request Broker Architecture
ESIOP	Environment Specific Inter ORB Protocol
ESPRIT	European Scientific Programme on Research in Information Technology
IDL	Interface Definition Language
IIOP	Internet Inter ORB Protocol
PIN	Persönliche Identifikationsnummer (Passwort)
ODP	Open Distributed Processing
ODPS	Open Distributed Paging System
REX	Remote Execution Protocol
RPC	Remote Procedure Call

ANSAware-Traders (der von Clients zur Ermittlung der Schnittstellenreferenzen der benötigten Dienste im Normalfall und im Umschaltfall benötigt wird) in der CORBA-Umgebung als CORBA Object Service beibehalten.

7 Bewertung

Das Gesamtsystem Scall und Skyper ist eine nach ODP-Prinzipien völlig verteilte Applikation: ODPS (Open Distributed Paging System). Das auf den gleichen Grundlagen basierende Redundanzkonzept ermöglicht die Implementierung ausfallsicherer Systeme ohne die Notwendigkeit spezieller Hardware oder Software. Die klare Architektur ermöglicht problemlos die Erweiterung um weitere Clients oder neue Server. Das können neue Zugangssysteme oder andere Aussendesysteme sein oder auch neue Dienste; beispielsweise wurde jetzt zwei Jahre nach Scall der neue Dienst Skyper hinzugefügt. Die klare Struktur ermöglicht es, komplexe Systeme sehr effizient zu implementieren.

Referenzen:

[Zweiacker 95]

Zweiacker, M.: Offene verteilte Systeme. Technische Mitteilungen PTT 3(1995), S. 143-155.

[Debski 96]

Debski, A., Janas, E.: The SysMan Monitoring Service and its Management Environment. Distributed Engineering 3 (1996), S. 136-147.