Diplomarbeit

Modellierung der verteilten Anwendung SAP R/3 für ein integriertes Management

Stefan Deml

INSTITUT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Diplomarbeit

Modellierung der verteilten Anwendung SAP R/3 für ein integriertes Management

Bearbeiter . . . . : Stefan Deml
Aufgabensteller . . . : Prof. Dr. Heinz-Gerd Hegering
Betreuer . . . . : Dr. Sebastian Abeck
Abgabedatum . . . . . : 15. Mai 1995

Ehrenwörtliche Erklärung

Ich versichere, daß ich diese Diplomarbeit selbständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Mai 1995

....
(Unterschrift des Kandidaten)

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einführung

1.1 Umfeld und Motivation der Arbeit

Bedingt durch drastisch gefallene Hardwarekosten für kleine und mittlere Systeme und die Verfügbarkeit von verteilten Anwendungen haben sich die DV-Versorgungsstrukturen in Unternehmen verändert. Wie in Abbildung 1.1 dargestellt, vollzieht sich ein Wandel von der Mainframe-Struktur hin zu einer „koordinierten Kooperation verteilter HW/SW-Komponenten“ [HNGU 95]. Um eine verteilte Ausführung zu ermöglichen, werden Anwendungen meist in die Funktionsbereiche Datenhaltung, Datenverarbeitung und Präsentation aufgeteilt. Die Kooperation zwischen diesen Moduln erfolgt oft nach dem Client-Server-Prinzip, wobei eine Komponente auch in mehreren Rollen auftreten kann. Dieses Prinzip ermöglicht eine flexible Verteilung der Komponenten auf Systeme und bildet so die Grundlage für die als „Rightsizing“ bezeichnete Anpassung der Systemkonfiguration an die Bedürfnisse des Unternehmens.

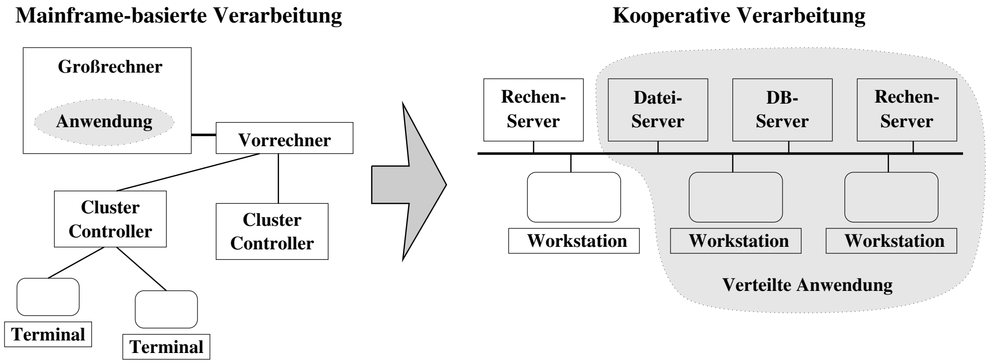

Abbildung 1.1: Neue DV-Versorgungsstrukturen (aus [HNGU 95])

Diesen Weg ist auch die Firma SAP mit ihrem System R/3 für betriebswirtschaftliche Anwendungen gegangen. Während der Vorgänger R/2 noch als Mainframe-

Anwendung konzipiert war, besteht ein R/3-System aus einem Datenbanksystem sowie Anwendungs- und Präsentationskomponenten, die beliebig auf heterogene Systeme verteilt werden können.

Den beschriebenen Vorteilen einer Client-Server-Struktur wie Skalierbarkeit und besseres Preis-/Leistungsverhältnis steht heute noch der Nachteil zum Teil ungelöster Managementprobleme solcher Strukturen gegenüber. Während sich bei Mainframe-Verarbeitung der Verteilungsaspekt auf das Netz beschränkt, müssen bei verteilter Verarbeitung neue Methoden entwickelt werden, um auch System- und Anwendungsressourcen zentral überwachen und steuern zu können.

Diese Anforderungen lassen sich, insbesondere bei großen Unternehmensnetzen, nur rechnergestützt erfüllen. Dabei wird angestrebt, die verschiedenen Managementaufgaben nicht mit isolierten Werkzeugen anzugehen, sondern vielmehr ein integriertes Management zu verwirklichen. Der Begriff „Integriertes Management“ beschreibt das Ziel, eine heterogene Netz-, System- und Anwendungs-Umgebung gemäß eines einheitlichen, abstrakten Modells dieser Wirklichkeit zu überwachen und zu steuern.

Ziel eines integrierten Anwendungsmanagement-Konzepts muß also sein, zunächst die Anforderungen an eine Managementlösung zu ermitteln, um davon ausgehend ein geeignetes Managementmodell der Anwendung entwickeln zu können. Um den Anforderungen gerecht werden zu können, wird ein solches Modell nicht nur die verteilte Anwendung selbst mit ihren Komponenten und Diensten repräsentieren, sondern auch Objekte aus den Bereichen des Netz- und Systemmanagements einbeziehen müssen.

Weitere Managementaspekte ergeben sich aus der Bedeutung der verteilten Anwendung für die Arbeitsabläufe im Unternehmen. Im Gegensatz zu Netz- und Systemkomponenten spiegeln Anwendungen stark die Anforderungen und Strukturen eines Unternehmens wider. Dies gilt insbesondere für SAP R/3. R/3 stellt in vielen Unternehmen die wichtigste, wenn nicht einzige Anwendung dar, die zentrale Unternehmensaufgaben unterstützt. Durch diese zentrale Rolle stellt die ständige Verfügbarkeit der Anwendung einen wichtigen Erfolgsfaktor dar. Deshalb muß das Anwendungsmanagement verstärkt die Unternehmensstruktur und Betriebsabläufe berücksichtigen.

1.2 Thema der Arbeit

In dieser Arbeit wird am Beispiel der Anwendung SAP R/3 ein Managementmodell für ein integriertes Management verteilter Anwendungen entwickelt, welches die genannten Managementaspekte berücksichtigt. Dabei soll im ersten Ansatz der Aspekt der Überwachung im Vordergrund stehen. Dieses Modell wird auf der Basis der bestehenden Managementkonzepte, die für das Netzmanagement entwickelt wurden, realisiert. Dabei zeigt sich, inwieweit diese Konzepte den erweiterten Anforderungen des Anwendungsmanagements genügen.

1.3 Zusammenfassung

Im folgenden Kapitel werden zunächst bestehende Managementkonzepte vorgestellt und ein für dieses Thema besonders relevanter Ansatz, das sog. Internet-Managementframework, bezüglich seiner aktuellen Erweiterungen für das System- und Anwendungsmanagement betrachtet. Dabei wird auch auf die konzeptionellen Schwächen dieses Ansatzes eingegangen.

Zur Entwicklung eines Managementkonzepts ist eine genaue Kenntnis der Architektur der verteilten Anwendung erforderlich. Daher stellt Kapitel 3 das System R/3 detailliert vor. Dabei wird hauptsächlich auf die managementrelevanten Architekturaspekte eingegangen. Die Möglichkeiten, die R/3 dem Benutzer bietet, werden nur knapp vorgestellt.

In Kapitel 4 werden zunächst allgemeine Anforderungen an die Überwachung verteilter Anwendungen formuliert, die dann an Hand der speziellen Managementaspekte des R/3-Systems verfeinert werden. Auf dieser Grundlage wird ein abstraktes Managementmodell für R/3 entwickelt, das insbesondere die Anforderungen an die erforderlichen Managementobjekte bestimmt.

Kapitel 5 beschreibt die bestehenden, isolierten R/3-Managementwerkzeuge sowie den Entwurf einer standardkonformen R/3-Managementschnittstelle. Dieser Entwurf der SAP AG wird bezüglich der Anforderungen aus dem vorangegangenen Kapitel bewertet und entsprechend weiterentwickelt.

Um nicht nur theoretisch erarbeitete Anforderungen an ein integriertes R/3-Management, sondern auch praktische Erfahrungen in die Konzeption des R/3-Anwendungsmanagements einfließen zu lassen, stellt Kapitel 6 die R/3-Managementszenarien bei der BASF AG vor. Da dieses Unternehmen eines der größten privaten Netze in Europa betreibt, konnten hier insbesondere neue Erkenntnisse über die spezifischen Probleme, die sich aus der Menge der zu überwachenden Systeme ergeben, gewonnen werden.

Die Konzeption einer R/3-Managementanwendung erfolgt in Kapitel 7. Dazu werden zunächst die bestehenden Konzepte einer kommerziellen Managementplattform bezüglich ihrer Erweiterungsmöglichkeiten für das Anwendungsmanagement analysiert. Auf der Basis dieser Erkenntnisse wird ein Implementierungskonzept für ein anforderungskonformes R/3-Managementmodul entworfen und sein Einsatz am Beispiel der R/3-Managementszenarien bei der BASF AG beschrieben.

Teile der in Kapitel 7 entworfenen Konzepte konnten bereits im Rahmen dieser Arbeit implementiert werden. Kapitel 8 beschreibt den entwickelten Prototypen und die erforderliche Testumgebung. Am Ende dieses Kapitels werden die bei der Implementierung gemachten Erfahrungen erläutert sowie erforderliche Weiterentwicklungen besprochen.

Kapitel 9 faßt die durch diese Arbeit gewonnenen Erkenntnisse zusammen und gibt einen Ausblick auf mögliche zukünftige Entwicklungen des integrierten R/3-Anwendungsmanagements.

Kapitel 2

Aktuelle Ansätze zum System- und Anwendungsmanagement

Im Bereich des integrierten Managements existieren derzeit zwei besonders wichtige konzeptionelle Lösungsansätze, sog. Management Frameworks. Ein solcher Ansatz ist das OSI Management Framework der ISO. Dieses versucht möglichst alle Managementaspekte zu berücksichtigen, was zu komplexen Anforderungen und hohem Implementierungsaufwand für diese Konzepte führt.

Dagegen verfolgt das Internet Management Framework einen pragmatischen Ansatz, bei dem Einfachheit und leichte Implementierbarkeit im Vordergrund stehen. Nicht zuletzt diese Eigenschaft hat zur starken Verbreitung dieses Konzepts im Bereich des Managements von Unternehmensnetzen beigetragen. Daher soll dieser Ansatz, der mittlerweile in seiner zweiten Fassung vorliegt, im folgenden Abschnitt näher betrachtet werden. Im Anschluß daran werden einige neuere Erweiterungen dieses Ansatzes hin zum System- und Anwendungsmanagement vorgestellt. Dabei wird auch immer wieder auf Schwächen dieses Ansatzes hingewiesen. Deshalb wird in Abschnitt 2.3.1 ein herstellerspezifischer Ansatz für das Systemmanagement vorgestellt und mit den zuvor beschriebenen Ansätzen verglichen.

Da in dieser Arbeit ein Managementkonzept für eine bereits bestehende Anwendung erarbeitet werden soll, werden Managementkonzepte, welche die Entwicklung verteilter Anwendungen unterstützen, hier nicht betrachtet.

2.1 Das Internet Management Framework

Das Internet Management Framework basiert auf dem in Abbildung 2.1 dargestellten Architekturmodell. Die als Managed Nodes bezeichneten Objekte des Managements (Netzkomponenten, Endgeräte) stellen Agenten bereit, die als Server

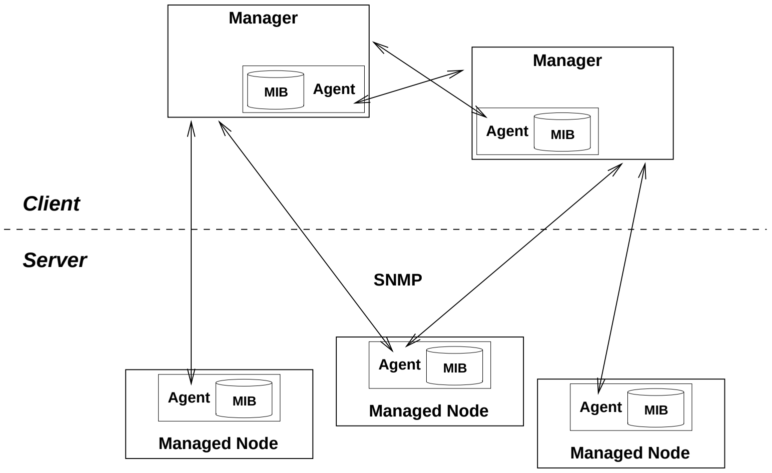

Abbildung 2.1: Das Internet Architekturmodell

für Managementinformation zu diesen Objekten dienen. Die Managementinformation wird in Form sog. Managed Objects bereitgestellt, die in einer Management Information Base (MIB) definiert werden [RFC 1155, RFC 1212]. Dabei wird nur die Struktur der Objekte definiert, um das Zugriffsverfahren normieren zu können. Die tatsächliche Implementierung der Datenstrukturen ist dem Agenten freigestellt. Managementanwendungen fragen diese Informationen über das SNMP-Protokoll [RFC 1157] ab, treten also in Client-Rolle auf. Zusätzlich können Agenten von sich aus Ereignismeldungen an die Manager schicken, deren Übertragung jedoch ungesichert abläuft.Solche Meldungen werden als Traps bezeichnet. Die dargestellte Kommunikationsbeziehung zwischen den Managementstationen ist eine der nachfolgend beschriebenen Erweiterungen dieses Framework.

Da die Elemente und Konzepte der Version 1 [RFC 1155, RFC 1157, RFC 1212, RFC 1213] dieses Modells in [HEAB 93] ausführlich beschrieben sind, werden hier nur einige Verbesserungen der Version 2 [RFC 1441] des Framework vorgestellt:

Sicherheitsaspekte Eine der wichtigsten Verbesserungen des neuen Framework betrifft die Sicherheit des Kommunikationsprotokolls SNMP. Die Nachfolgeversion SNMPv2 [RFC 1448] wurde um Verfahren zur Authetifizierung und Verschlüsselung der Protokolldateneinheiten [RFC 1446] versehen. Ferner wurde ein Administrationsmodell [RFC 1445] geschaffen, welches eine selektive Vergabe von Rechten auf Managementoperationen und Objekte erlaubt. Dieses Modell wird durch eine MIB [RFC 1447] unterstützt. Solche Maßnahmen sind eine unabdingbare Voraussetzung um System- und Anwendungsmanagementfunktionen über dieses Protokoll steuern zu können, da hier stark sicherheitsrelvante Bereiche berührt werden.

Verteiltes Management Das Management sehr großer Netze erfordert die Abkehr von einer zentralen Managementstation hin zu verteilten Managern, die jeweils einen Teilbereich des Netzes kontrollieren. Diese Entwicklung wird durch das neue Internet Management Framework SNMPv2 auf zwei Arten unterstützt. Zum einen wurde das Modell dahin erweitert, daß nun auch Agenten in den einzelnen Managementstationen vorgesehen sind. Zum anderen wurde die Manager-to-Manager-MIB [RFC 1451] für solche Agenten definiert, um die Kommunikation der Manager untereinander zu unterstützen. Diese Kommunikation erfolgt ebenfalls über SNMPv2.

Ein wesentlicher Mangel des SNMP-Protokolls wurde jedoch auch in der Version 2 nicht wirklich behoben. Das SNMP-Protokoll sieht keine Operationen zum expliziten Erzeugen oder Löschen der Instanzen von Managed Objects vor. Daher können solche Operationen nur als Seiteneffekte von Schreiboperationen auf bestehende Instanzen realisiert werden, was zum Teil zu sehr komplexen Operationsfolgen führt, die benutzerfreundlich nur von speziell programmierten Managementanwendungen durchgeführt werden können. Da diese Verwendung des SNMP-Protokolls im Standard bisher nicht vorgesehen war, wurden solche Operationen auch stark unterschiedlich realisiert. Dies ist natürlich dem eigentlichen Ziel standardisierter Managementinformation, nämlich der Entwicklung einheitlicher Managementanwendungen, nicht zuträglich. Für SNMPv2 ist in [RFC 1443] eine Konvention zur Implementierung solcher Mechanismen festgelegt worden, was aber auch nur als Übergangslösung angesehen werden kann. Die sich aus diesem Mangel ergebenden Probleme des Einsatzes von SNMP für das Systemmanagement werden in [KRUP 93] ausführlich diskutiert.

2.2 Ansätze zum System- und Anwendungsmanagement im Internet Management

Dieser Abschnitt beschreibt aktuelle Ansätzte zum System- und Anwendungsmanagement, die entsprechend dem Internet Management Framework realisiert sind. Die Beschreibungen basieren auf den in der jeweiligen MIB dokumentierten Objektdefinitionen. Einige der vorgestellten Lösungen sind als internationaler Standard vorgeschlagen (Host Resources MIB, Network Services Monitoring MIB, Relational Database Management System MIB) und im mgmt-Teilbaum des Internet-Registrierungsbaums (siehe Abbildung 2.2) registriert, während die Übrigen (Oracle MIB, SAP R/3 MIB, WinWatch MIB) herstellerspezifische Lösungen darstellen, die einzelne Produkte unterstützen und im private-Teilbaum registriert sind. Eine herstellerspezifische MIB wird deshalb oft auch als Private-MIB bezeichnet.

Die Auswahl der vorgestellten MIBs erfolgte auch natürlich entsprechend ihrer Bedeutung für das R/3-Management. Das von der Oracle-MIB unterstützte Da-

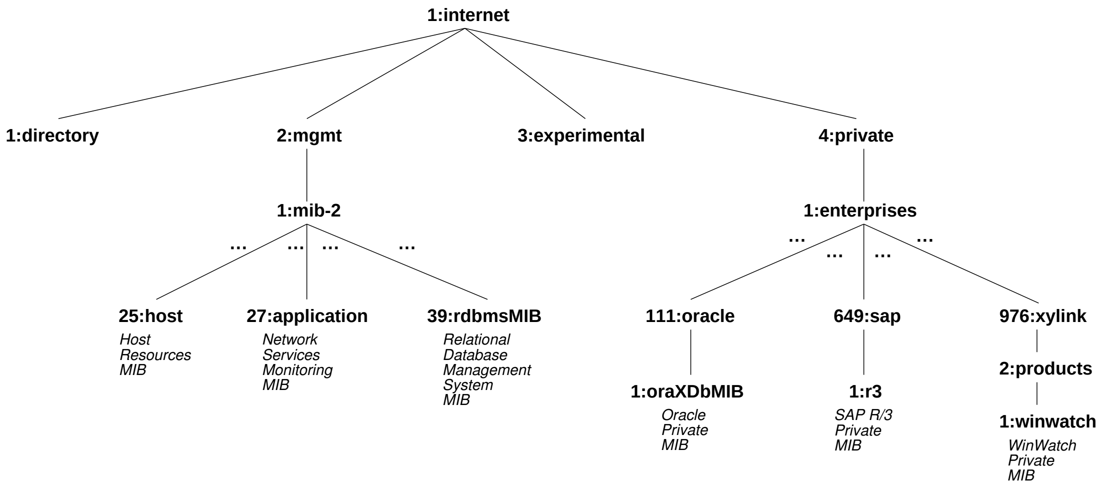

Abbildung 2.2: Der Internet Registrierungsbaum

tenbanksystem der Firma Oracle bildet in zahlreichen Installationen die Basis des R/3-Systems. Als R/3-Clients kommen häufig Personal-Computer mit Windows-Betriebssystem zum Einsatz, zu deren Management die WinWatch-MIB entworfen wurde.

2.2.1 Host Resources MIB

Die Host Resources MIB beschreibt Managed Objects für das Systemmanagement, die für möglichst viele verschiedene Arten von Endsystemen von Nutzen sein sollen. Die Bandbreite der unterstützten Systeme reicht also von PCs über Workstations bis hin zu Großrechnern. Es wurde ferner darauf geachtet, die Objektdefinitionen möglichst unabhängig von speziellen Betriebssystem-Eigenschaften der Systeme zu halten.

Die MIB gliedert sich in 6 Gruppen, wobei die Gruppen System, Storage und Device als „mandatory“ gekennzeichnet sind, also von jeder standardkonformen Implementierung bereitgestellt werden müssen. Dagegen sind die Gruppen Running Software, Software Performance und Installed Software „optional“, ihre Realisierung also zur Erhaltung der Standardkonformität nicht erforderlich.

Abbildung 2.3 zeigt die einzelnen Gruppen und ihre Objekte. Einige Objekte der Host Resources MIB enthalten Verweise auf andere Objekte der MIB, was in dieser Abbildung durch Pfeile dargestellt ist.

Da sich die Bedeutung der einzelnen Objekte i.a. schon aus ihrem Namen erschließt werden die Gruppen nur im Überblick beschrieben.

System Group Die System Group bietet einige Objekte zum Zustand des Systems (Datum, Anzahl Benutzer und Prozesse) sowie einen Verweis auf die

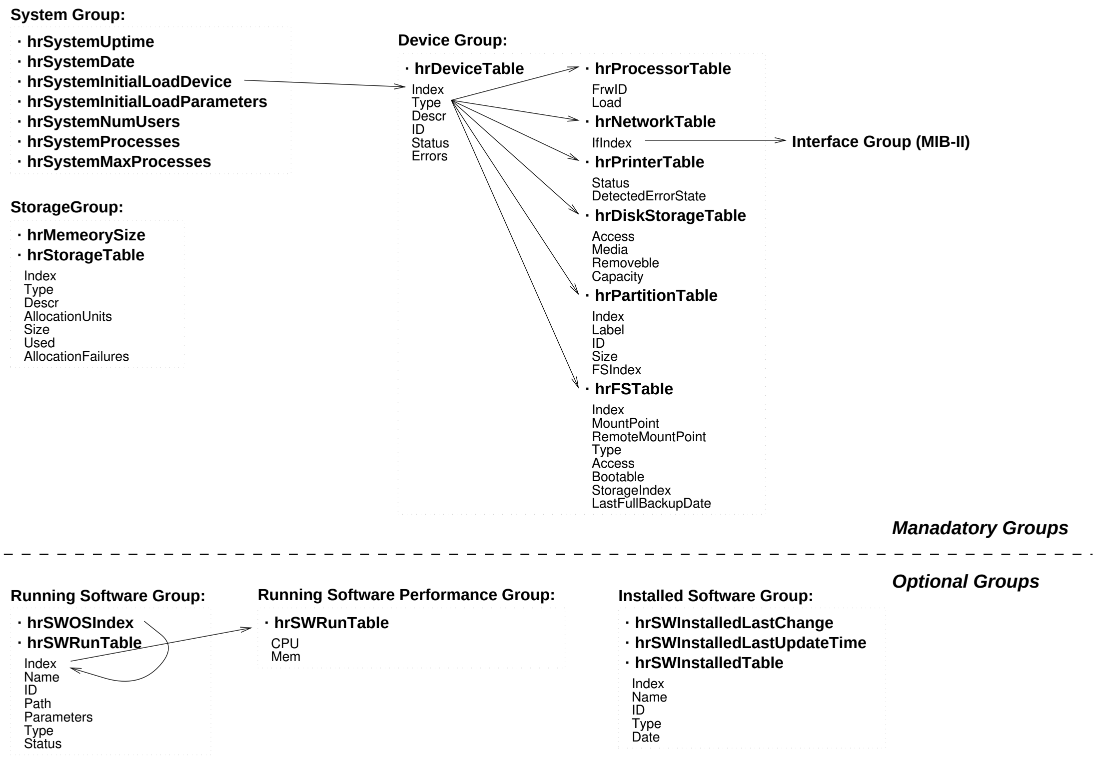

Abbildung 2.3: Struktur der Host Resources MIB

Komponente in der Device Table, von der beim Hochfahren des Systems die Betriebssystemkonfiguration geladen wird. Ein weiters Objekt enthält die dazu nötigen Parameter.

Storage Group Neben einem Objekt zur Größe des Hauptspeichers des Systems enthält die Storage Group eine Tabelle die alle logischen Speicherelemente des Systems wie z.B. RAM und Swap Space beschreibt.

Device Group Konfigurations- und Leistungsdaten der einzelnen Hardwarekomponenten des Systems sind in der Device Group beschrieben.

Die hrDeviceTable gibt zunächst einen Überblick über alle Komponenten, von denen einzelne in weiteren Tabellen detaillierter beschrieben sein können. Eine solche Tabelle ist die hrNetworkTable, die als einzige Spalte einen Verweis auf die Interface-Group der MIB-II [RFC 1213] enthält, in der die Netzanschlüsse eines Systems beschrieben sind.

Running Software Group Diese Gruppe besteht aus einer Tabelle, die alle z.Zt. laufenden Applikationen des Systems beschreibt und einem weiteren Objekt, welches auf den Betriebssystem-Eintrag innerhalb dieser Tabelle verweist.

Running Software Performance Group Als Erweiterung der Running Software Group enthält diese Gruppe zwei Leistungsparameter zu den Applikationen des Systems, nämlich CPU-Zeit und reservierter Speicherplatz.

Installed Software Group In dieser Gruppe sind alle auf dem System installierten Softwarepakete beschrieben.

Da alle Objekte der Host Resources MIB nur lesenden Zugriff erlauben, kommt diese MIB nur für Überwachungsaufgaben in Frage. Da diese MIB noch für das SNMPv1-Protokoll entworfen wurde, ist diese Einschränkung wegen der bekannten Sicherheitsmängel dieses Protokolls sicher sinnvoll. Der entscheidende Vorteil dieser MIB liegt darin, daß unterschiedlichste Systeme durch einheitliche Objekte beschrieben werden, was die Entwicklung integrierter Managementanwendungen erheblich erleichtert.

Ein denkbarer Einsatzbereich dieser MIB sind Managementsysteme für ein User-Help-Desk, das mit den bereitgestellten Informationen Fehlermeldungen der Benutzer genauer analysieren und eingrenzen kann.

2.2.2 WinWatch

Das Produkt WinWatch [WW USER] der Firma Metrix realisiert einen SNMP-Agenten, der eine in Anlehnung an die Host Resources MIB entworfene MIB für das Management von PCs mit Windows Betriebssystem bereitstellt. Diese WinWatch-MIB beschreibt in 5 Gruppen (System, Storage, Devices, Tasks und Installed Software), die nachfolgend genauer erläutert sind, zahlreiche Managementaspekte eines Windows PC. Die einzelnen Gruppen der MIB sind durch Dynamic Link Librarys (DLL) realisiert und müssen daher nicht immer vollzählig auf einen PC installiert sein. Eine weitere MIB-Gruppe (Extensions) gibt Auskunft darüber welche DLLs auf einem PC installiert und aktiviert sind.

System Group Die System Group enthält Konfigurationsinformationen zum Windows- und DOS-Betriebssystem wie z.B. Versionsnummern oder Farbeinstellungen der Benutzeroberfläche. Wie in der Host Resources MIB existiert ein Parameter mit einem Verweis auf das Boot-Device des PCs in der Devices Group. Ferner enthält die System Group Tabellen zu den wichtigsten Konfigurationsdateien eines Windows PC, nämlich CONFIG.SYS, AUTOEXEC.BAT, WIN.INI, SYSTEM.INI und dem General System Configuration File, einer Konfigurationsdatei mit frei wählbarem Namen. Diese Dateien werden durch Tabellen realisiert, die in jeder Zeile jeweils eine Zeile der Datei enthalten. Da diese Tabellen auch schreibenden Zugriff gestatten, können die Konfigurationsdateien zeilenweise von der Managementstation aus verändert werden.

Storage Group Die Storage Group befaßt sich mit den Speichermedien eines PCs wie RAM, Plattenspeicher, Floppy-Disk, CD-ROM, ... Neben Windows-spezifischen Konfigurations und Leistungsdaten zu diesen Bereichen bietet die Storage Group die Möglichkeit, Schwellwerte für verschiedene Objekte wie z.B freier Hauptspeicher oder Plattenplatz zu definieren, bei deren Verletzung entsprechende SNMP-Traps generiert werden. Im Gegensatz zur Disk Device Table der Devices Group erfaßt die Storage Group die logische Konfiguration.

Devices Group Die Devices Group beschreibt die Konfiguration der physikalischen Komponenten eines PCs und der direkt angeschlossenen Geräte. Dazu ist diese Gruppe in die Untergruppen Processor, Display, Printer, Keyboard, Mouse, Disk-Device, Serial Ports, Parallel Ports und Network aufgeteilt, die jeweils die entsprechende Komponentenkonfiguration und gegebenenfalls die Konfiguration zugehöriger Treibersoftware beschreiben.

Tasks Group Die Tasks Group informiert über die zur Zeit auf einem PC laufenden Applikationen einschließlich Gerätetreiber und Windows-Module. Verschiedene Untergruppen bieten Konfigurations- und Leistungsdaten zu einzelnen Applikationstypen. Zusätzlich realisiert diese Gruppe einen Mechanismus, der zur Realisierung einer unternehmensweiten Softwarelizenzkontrolle verwendet werden kann. Dazu werden in einer Tabelle diejenigen Applikationen definiert, vor deren Start zunächst eine Anfrage in Form eines Trap an die Managementstation geschickt wird. Die Managementstation kann dann aufgrund ihres globalen Überblicks entscheiden, ob eine weitere Applikation dieses Typs gestartet werden darf oder nicht. Die Entscheidung wird dem Agenten durch das entsprechende Setzen eines Attributs mitgeteilt. Auf dieses Ereignis wartet der Agent eine frei konfigurierbare Zeitspanne, danach wird, je nach Konfiguration des Agenten, der Start der Applikation freigegeben oder gesperrt.

Installed Software Group Die Installed Software Group gliedert sich in die Untergruppen Directory Listing, Checked Files, Installed Software und Program Manager.

Die Directory Listing Group gestattet es, den Inhalt eines Datei-Verzeichnisses zu ermitteln. Dazu wird der Pfadname des gewünschten Verzeichnisses in ein MIB-Attribut geschrieben, worauf der Agent den Inhalt ermittelt und das Ergebnis im Format des DIR-Kommandos in einer MIB-Tabelle ablegt.

Die Checked Files Group ermöglicht es Dateien oder Directorys zu spezifizieren, für die bestimmte Änderungen durch Traps gemeldet werden sollen. So läßt sich beispielsweise die Größe einer Datei oder die Anzahl der Dateien in einem Directory überwachen.

Die Installed Software Group bietet eine Liste der installierten Programme. Welche Dateien in diese Liste aufgenommen werden sollen, wird an Hand der Dateiendung spezifiziert. Die Dateien dieser Liste können bezüglich Veränderungen ihrer Gesamtzahl, Größe, Zeitstempel und Prüfsumme überwacht werden.

Die Program Manager Group enthält Daten zum aktuellen Zustand des Microsoft Program Manager, falls dieser als Shell verwendet wird.

Zu diesem Agenten existiert auch eine entsprechende Managementanwendung für die Plattform SPECTRUM (siehe 7.2). Dieses Managementmodul [WW SPEC] wurde unter Verwendung der in 7.4 beschriebenen Methoden realisiert.

Im Vergleich zur Host Resources MIB bietet die WinWatch MIB deutlich umfangreichere Funktionalität, die für die meisten Managementaufgaben im PC-Bereich ausreichen sollte. Einigen Anwendern dieses Produkts geht die Mächtigkeit von WinWatch sogar zu weit. Es ist ein Fall bekannt, in dem der Betriebsrat eines Unternehmens den Einsatz der Tasks Group verhindert hat, da dies zu unzulässiger Überwachung der Mitarbeiter führen kann. Neuere WinWatch-Versionen informieren jetzt den Benutzer über Zugriffe der Managementstation.

Durch Erstellen neuer DLLs läßt sich der WinWatch-Agent um neue Gruppen erweitern. Dies kann beispielsweise zur Realisierung von MIBs für einzelne Windows-Anwendungen genutzt werden, deren Funktion zentral überwacht werden soll.

2.2.3 Network Services Monitoring MIB

Die Network Services Monitoring MIB wurde als allgemeiner Beschreibungsrahmen für netznahe Applikationen entwickelt, wird jedoch zunehmend auch für allgemeine Applikationen verwendet. Daher wird diese MIB oft auch als Application-MIB bezeichnet. Diese MIB stellt Tabellen zur Beschreibung einzelner Applikationen (applTable) und ihrer Beziehungen (assocTable) zur Verfügung. Jede Zeile der über den applIndex indizierten applTable beschreibt genau eine Applikation des Systems. Der applIndex kann in weiteren MIBs verwandt werden, um zusätzliche Information zu einer Anwendung bereitzustellen. Dieses Verfahren wird z.B. bei der nachfolgend beschriebenen RDBMS-MIB angewandt.

Für eine detaillierte Beschreibung der Network Services Monitoring MIB wird auf [MAUR 95] verwiesen.

2.2.4 RDBMS-MIB

Die Relational Database Managemt System (RDBMS) MIB beschreibt Managed Objects für das Management allgemeiner, relationaler Datenbanksysteme [DATE 95], also ohne Berücksichtigung herstellerspezifischer Eigenschaften. Solche produktspezifischen Objekte werden in Private-MIBs des Herstellers beschrieben, auf die von der RDBMS-MIB verwiesen wird. Als Beispiel einer solchen MIB ist in 2.2.5 die MIB des Datenbankherstellers Oracle beschrieben.

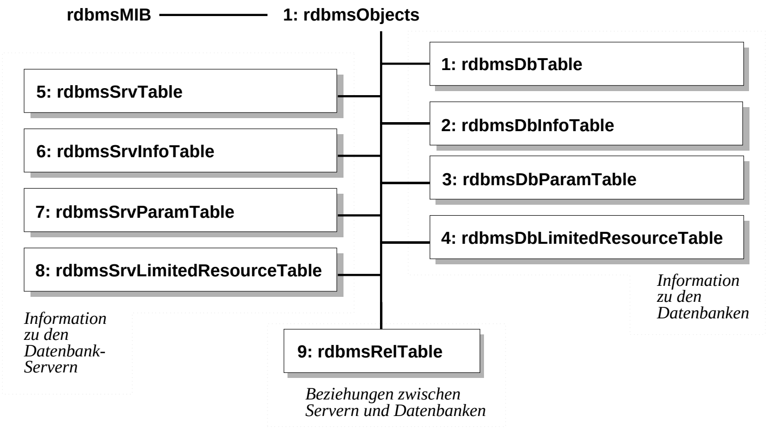

Abbildung 2.4: Struktur der RDBMS-MIB

Die RDBMS-MIB unterscheidet zwischen den logischen Einheiten eines Datenbanksystems (Datenbanken) und den operativen Einheiten (Server), welche den Zugriff auf eine oder mehrere Datenbanken ermöglichen. Die Beziehungen zwischen Datenbanken und Servern sind in einer eigenen Tabelle beschrieben.

rdbmsDbTable Diese Tabelle ist eine Liste der Datenbanken, die sich auf diesem Host befinden. Der Tabellenindex rdbmsDbIndex wird in einigen weiteren, nachfolgend beschriebenen Tabellen als erster Index zur Referenzierung einer Datenbank verwendet. Als Spalten dieser Tabelle finden sich weiterhin der Object-Identifier des Datenbankherstellers, d.h. der Pfad zur Wurzel des herstellerspezifischen Registrierungsbaums im private-Teilbaum des Internet-Registrierungsbaums. Dies ermöglicht den Zugriff auf Tabellen der herstellerspezifischen MIBs, die oft Erweiterungen der hier beschriebenen Tabellen darstellen. Weitere Spalten sind der Name des Datenbanksystem-Herstellers, der Name der Datenbank und Information über eine zuständige Kontaktperson.

rdbmsDbInfoTable Als Erweiterung der rdbmsDbTable ist diese Tabelle ebenfalls über den rdbmsDbIndex indiziert und bietet Information zur Produktbezeichnung und Version des Datenbanksystems, die Definition der in den weiteren Tabellen verwandten Maßeinheit für Speicherplatz, zu reserviertem und belegtem Plattenplatz sowie den Zeitpunkt der letzten Datenbanksicherung.

rdbmsDbParamTable Diese Tabelle enthält herstellerspezifische Konfigurationsparameter in beliebiger Anzahl. Dazu wird diese Tabelle neben dem rdbmsDbIndex über den Namen des Parameters (rdbmsDbParamName) und, falls es sich um einen aus mehreren Werten bestehenden Parameter handelt, zusätzlich über den rdbmsDbParamSubIndex indiziert. Andernfalls hat rdbmsDbParamSubIndex den Wert 1. Als eigentliche Spalten erscheinen ein Verweis auf die private MIB des Datenbankherstellers, die eine genauere Definition des Parameters enthalten kann, sowie der aktuelle Wert und eine textuelle Beschreibung der Bedeutung dieses Parameters.

rdbmsDbLimitedResourceTable Die Ressourcen der Datenbanken und ihre Beschränkungen werden in dieser Tabelle beschrieben. Dazu ist sie über den rdbmsDbIndex und den Namen der Ressource (rdbmsDbLimitedResourceName) indiziert. Zu jeder Ressource sind als Parameter der aktuelle Wert, Grenzwert, höchster je erreichter Wert, die Anzahl der Fälle, in denen versucht wurde den Grenzwert zu überschreiten, und eine Beschreibung des Parameters vorgesehen. Zusätzlich existiert wieder ein Verweis auf die Definition des Parameters in der privaten MIB.

Nachdem sich die bisher beschriebenen Tabellen mit den Datenbanken des Host befaßt haben, beschreiben die folgenden 3 Tabellen Eigenschaften der installierten Datenbankserver.

rdbmsSrvTable Diese Tabelle bietet allgemeine Information zu den Datenbankservern dieses Host. Da ein Datenbankserver eine Anwendung dieses Systems darstellt, wird als Tabellenindex der aus der Network Services Monitoring MIB bekannte applIndex verwendet. Es ist dabei nicht festgelegt ob ein Server aus einem oder mehreren Prozessen besteht. Als Tabellenspalten finden sich der OID des Herstellers, dessen Name, die Produktbezeichnung des Servers und Information zu einer zuständigen Kontaktperson.

rdbmsSrvInfoTable Diese ebenfalls über den applIndex indizierte Tabelle enthält im wesentlichen Leistungsdaten zu den Datenbankservern dieses Hosts. Im einzelnen sind dies Werte zur Performance der Dateizugriffe und des Caching, sowie zur Anzahl bestehender Verbindungen und bearbeiteter Transaktionen.

Neben diesen Werten ist in der RDBMS-MIB auch die Bedeutung der Spalten der applTable der Network Services Monitoring MIB für relationale Datenbanksysteme in Form von Kommentaren beschrieben. Die Spalten behalten im wesentlichen ihre ursprüngliche Bedeutung. Die Spalten zu Inbound

Associations beziehen sich auf aktive SQL-Verbindungen, die Spalten zu Outbound Associations werden nicht unterstützt.

rdbmsSrvParamTable Diese Tabelle beschreibt die Konfigurationsparameter der einzelnen Datenbankserver dieses Systems. Die Struktur der Tabellendefinition und die Bedeutung der einzelnen Spalten entspricht exakt der rdbmsDbParamTable.

rdbmsSrvLimitedResourceTable Die Ressourcen der Datenbankserver dieses Systems und ihre Beschränkungen werden in dieser Tabelle beschrieben. Die Struktur der Tabellendefinition und die Bedeutung der einzelnen Spalten entspricht exakt der rdbmsDbLimitedResourceTable.

rdbmsRelTable Die letzte Tabelle der RDBMS-MIB beschreibt schließlich die Zuordnung der Datenbanken zu Datenbankservern. Dazu ist sie über rdbmsDbIndex und applIndex indiziert. Weitere Tabellenspalten beschreiben den aktuellen Zugriffstatus eines Servers auf eine Datenbank und den Zeitpunkt, zu dem eine Datenbank durch einen Server aktiviert wurde.

Weiterhin beschreibt die RDBMS-MIB zwei Traps:

rdbmsStateChange Dieser Trap meldet Änderungen an der rdbmsRelTable.

rdbmsOutOfSpace Dieser Trap zeigt an, daß ein Datenbankserver keinen zusätzlichen Plattenplatz mehr belegen konnte.

Die Analyse der RDBMS-MIB zeigt, daß diese zwar zahlreiche Performancewerte enthält, die für die Überwachung von Datenbanksystemen notwendig sind, deren Überwachung aber nicht durch entsprechende Traps unterstützt wird. Hier müßten weitere Objekte in die MIB aufgenommen werden, welche die Definition von Schwellwerten erlauben, bei deren Verletzung Traps generiert werden. Andernfalls erfordert die Überwachung netzlastintensives Polling.

Für echte Datenbankadministration ist die RDBMS-MIB nicht entworfen worden und, wie sich im folgenden Abschnitt zeigt, trotz herstellerspezifischer Erweiterungen in Form von private MIBs nicht geeignet.

2.2.5 Oracle Private MIB

Die MIB des Datenbankherstellers Oracle stellt eine produktspezifische Erweiterung der RDBMS-MIB dar. Ein Agent, der die Oracle-MIB unterstützt, muß daher auch stets die RDBMS-MIB unterstützen. Die Oracle-MIB (oraXDbMIB) befindet sich derzeit noch in Weiterentwicklung, was durch das X für eXperimental im Namen unterstrichen wird.

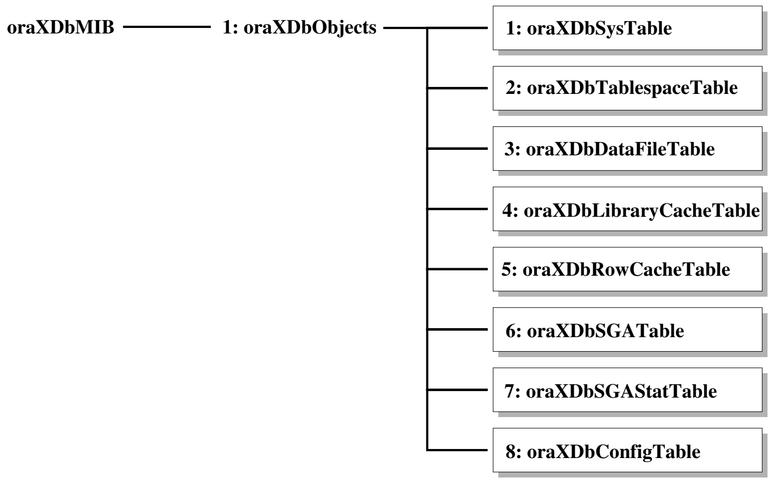

Abbildung 2.5: Struktur der Oracle Private MIB

Die oraXDbMIB beschreibt in 8 Tabellen Konfigurations- und Leistungsdaten von Oracle-Datenbanksystemen. Alle Tabellen verwenden rdbmsDbIndex aus der RDBMS-MIB als ersten Index, d.h. die Informationen beziehen sich stets auf Datenbanken, nicht auf Datenbankserver. Da die Interpretation der einzelnen Tabellenspalten detailierte Kentnisse der Architektur und Funktionsweise von Oracle-Datenbanken voraussetzt, wird die Bedeutung der Tabellen nur überblicksartig beschrieben:

oraXDbSysTable Die oraXDbSysTable enthält in jeder Zeile Performancedaten zu einer Oracle-Datenbank. Die einzelnen Spalten stellen eine Auswahl aus der Oracle-internen Performance-Tabelle dar. Es wurden diejenigen Werte ausgesucht, welche sich als besonders aussagekräftig für Tuningmaßnahmen erwiesen haben.

oraXDbTablespaceTable Die oraXDbTablespaceTable bietet eine Übersicht über die Tablespaces jeweils einer Datenbank. Unter einem Tablespace versteht man einen logischen Datenbankteil, der zur Speicherreservierung für relationale Tabellen verwendet wird. Ein Tablespace kann durch eine oder mehrere Dateien realisiert sein, welche in der oraXDbDataFileTable beschrieben sind. Die Indizierung der oraXDbTablespaceTable erfolgt zusätzlich über den oraXDbTablespaceIndex, der jedem Tablespace innerhalb einer Datenbank eine eindeutige Kennzahl zuordnet. Als Spalten finden sich Parameter wie Name, Größe, maximale Größe und Verfügbarkeit der Tablespaces.

oraXDbDataFileTable Die oraXDbDataFileTable bietet Konfigurations- und Performancedaten zu den Dateien aus denen ein Tablespace aufgebaut ist.

Die einzelnen Dateien werden durch den zweiten Index (oraXDbDataFileIndex) identifiziert. Zu jeder Datei werden Werte wie Name und Größe sowie Anzahl und Dauer der Zugriffe gespeichert.

oraXDbLibraryCacheTable Diese Tabelle bietet Performancedaten zum Library Cache der Oracle-Datenbanken dieses Systems. Als zweiter Index dient oraXDbLibraryCacheIndex, welcher jeweils einen Namensraum im Library Cache identifiziert.

oraXDbRowCacheTable Leistungsdaten zu den Row-Caches (Data-Dictionary-Caches) der Oracle-Datenbanken dieses Hosts werden von der oraXDbRowCacheTable bereitgestellt. Sie gibt damit Aufschluß über die Zugriffsaktivität auf das Data Dictionary. Als Index über die Data-Dictionary-Caches dient oraXDbRowCacheIndex.

oraXDbSGATable Die oraXDbSGATable liefert Übersichtsinformation zum System Global Area (SGA) jeder Datenbank dieses Host. Das SGA ist der globale Speicherbereich einer Datenbank. Die oraXDbSGATable erfaßt den statischen Teil dieses Bereichs, welcher aus einer Konfigurationsdatei entnommen wird und sich zur Laufzeit nicht ändert.

oraXDbSGAStatTable Die oraXDbSGAStatTable enthält Leistungsdaten zu den dynamischen Bereichen des System Global Area. Mit Hilfe der Parameter dieser Tabelle läßt sich beispielsweise prüfen, ob der reservierte Speicher für das SGA ausreicht.

oraXDbConfigTable Diese Tabelle enthält Oracle-spezifische Konfigurationsdaten zu jeder Oracle-Datenbank dieses Hosts. Die Werte für die 39 Spalten dieser Tabelle werden aus einer Konfigurationsdatei entnommen und sind zur Bewertung der Leistungsdaten anderer Tabellen erforderlich.

Da die Oracle-MIB im wesentlichen eine 1:1-Abbildung von Konfigurations- und Leistungsdaten aus internen Oracle-Tabellen in einen SNMP-Agenten vornimmt und keine Vorverarbeitung der Information stattfindet, muß auf der Managerseite umfangreiches Oracle-Expertenwissen nachgebildet werden, um die Daten für das Management nutzbar zu machen. Dies wird durch die spärliche Dokumentation der MIB nicht erleichtert. Leider enthält die MIB auch keine Traps, was sie für Überwachungsaufgaben weiter ungeeignet erscheinen läßt.

Managementmodule zur RDBMS- und Oracle-MIB für die Plattform SPECTRUM (siehe Abschnitt 7.2) werden z.Zt. vom Plattformhersteller Cabletron entwickelt.

2.3 Proprietäre Ansätze zum System- und Anwendungsmanagement

Wegen der erwähnten Beschränkungen des SNMP-Ansatzes und den nicht zuletzt daraus resultierenden Mängeln der beschriebenen, standardkonformen Ansätze werden insbesondere im Bereich des System- und Anwendungsmanagements heute noch verschiedene Werkzeuge eingesetzt, die nicht einem standardisierten Management-Framework entsprechen, sondern zur Kommunikation und Realisierung von Agenten eigene Konzepte verwirklichen und sich erst in der Managementplattform in die bestehende Umgebung einfügen. Als Beispiel eines solchen Produkts wird MaestroVision der Firma Calypso vorgestellt.

2.3.1 MaestroVision

Das Produkt MaestroVision besteht aus plattformspezifischen Agenten für verschiedene UNIX-Derivate und einer Managementanwendung für die Plattform SPECTRUM. Die Kommunikation zwischen Manager und Agent erfolgt über ein erweitertes RPC[1]-Protokoll. Auf den zu managenden Systemen ist in einer Konfigurationsdatei festgehalten, welche Managementstation mit welchen Rechten (Read, Write und/oder Execute) auf den mit ROOT-Rechten versehenen Agenten zugreifen darf. Zusätzlich erfolgt der Nachrichtenaustausch verschlüsselt. Im Vergleich zu SNMP ist diese Kommunikationsmethode deutlich sicherer und auch performanter.

Die Funktionalität des MaestroVision-Agenten gliedert sich in die 5 Module Processor, Users, File System, Memory und Application, die nachfolgend beschrieben sind. Mit Hilfe eines API[2] können weitere Module realisiert werden. Eine wichtige Eigenschaft des Agenten ist die Möglichkeit, zu allen definierten Managed Objects flexibel Schwellwerte definieren zu können, deren Einhaltung vom Agenten überwacht wird und bei deren Verletzung zum einen Nachrichten an die Managementstation geschickt werden können und zum anderen eine selbständige Reaktion des Agenten in Form der Ausführung eines Kommandos erfolgen kann. Eine mögliche Anwendung dieser Funktion ist beispielsweise das Löschen bestimmter Dateien bei knappem Plattenplatz. Die einzelnen Agenten-Module realisieren folgende Funktionen:

Processor Das Processor-Modul realisiert Objekte zur Performance der CPU wie Wait- und Idle-Time sowie Objekte zur den einzelnen Prozessen. Die Information zu den Prozessen entspricht der durch ein ps-Kommando erhält-

[1] Remote Procedure Call

[2] Application Programming Interface

lichen Information. Prozesse können angehalten und wieder gestartet oder auch beendet werden.

Users Durch das Users-Modul werden Objekte zur Benutzerverwaltung realisiert. Neben der Abfrage von Daten zu einzelnen Kennungen ist das Einrichten und Löschen von Kennungen möglich.

File System Die Objekte des File System Moduls beschreiben Größe und Füllungsgrad der lokalen und entfernten Dateisysteme. Ferner werden Funktionen zum Montieren und Demontieren von Dateisystemen angeboten.

Memory Durch das Memory-Modul werden verschiedene Objekte zur Größe und Auslastung des Haupt- und Hintergrundspeichers bereitgestellt.

Application Das Application-Modul realisiert erste Ansätze zum Anwendungsmanagement. Es bietet die Möglichkeit einzelne Anwendungen an Hand ihres Prozeßnamens auszuwählen und besonders zu überwachen. Neben der Überwachung von Prozeßattributen können auch Einträge in die SysLog-Datei, die von dieser Anwendung vorgenommen wurden, an die Managementstation gemeldet werden.

Die MaestoVision-Managementanwendung wurde mit Hilfe der SPECTRUM-Entwicklungswerkzeuge (siehe 7.3) realisiert und bietet neben zahlreichen Oberflächen zur Präsentation der Managed Objects und zur Initiierung von Managementoperationen eine weitere Viewhierarchie für das Systemmanagement neben den bestehenden SPECTRUM-Hierarchien (siehe 7.2.2). Diese Viewhierarchie ermöglicht nicht nur eine logische Strukturierung der zu managenden Systeme, sondern eröffnet auch einige neue Managementmöglichkeiten.

Die Systemmanagement-Hierachie besteht aus den Elementen „Policy Area“ und „System Manager“. System Manager dienen zur Gruppierung einzelner Workstations zu einer gemeinsam managebaren Einheit. Dadurch lassen sich beispielsweise Funktionen wie das Anlegen einer neuen Benutzerkennung gleichzeitig für mehrere Workstations durchführen. Policy Areas bieten die gleiche Funktionalität, können aber zusätzlich eine Hierarchie bilden, wohingegen System Manager sich immer in der untersten Ebene (abgesehen von den Workstations selbst) der Hierarchie befinden. Policy Areas und System Manager werden in der jeweils darüberliegenden Hierarchieebene durch Icons repräsentiert. Eine wichtige Funktion dieser Icons ist die Darstellung des aus den Zustandswerten der untergeordneten Objekte berechneten Gesamtzustands. Dadurch können auch große Mengen von Endgeräten einfach Überwacht und Störungen sicher erkannt werden.

2.4 Bewertung der Ansätze

Abschließend läßt sich feststellen, daß im Bereich des System- und Anwendungsmanagements Managementaufgaben, die über reine Überwachung hinausgehen, vor allem aus Sicherheitsgründen bisher proprietären Lösungen vorbehalten sind. Diese Situation wird sich jedoch sicher verändern, sobald echte SNMPv2-Lösungen, d.h. mit Implementierung der beschriebenen Sicherheitsmechanismen, stärkere Verbreitung finden. Da aber die Implementierung von SNMPv2-Agenten und Managementanwendungen im Vergleich zu SNMPv1 erheblich aufwendiger ist, werden solche Projekte nur zögernd realisiert.

Kapitel 3

Das SAP-System R/3 aus der Sicht des integrierten Managements

In diesem Kapitel soll zunächst des System R/3 kurz vorgestellt werden. Anschließend werden die Architektur des Systems und die Aufgaben der einzelnen Komponeneten erläutert. Abschließend folgt ein Beispiel einer typischen R/3-Systemkonfiguration, sowie einige Anmerkungen zur R/3-Architektur.

3.1 SAP R/3 – Ein integriertes System für betriebswirtschaftliche Anwendungen

Das System R/3 der SAP[1] AG bietet Anwendungsmodule zur Unterstützung verschiedener Unternehmensaufgaben, wie z.B.:

- Logistik
- Rechnungswesen
- Personalwesen

Ferner stehen Anwendungen zur Bürokommunikation und -automation zur Verfügung. Eine genauere Beschreibung der Anwendungsmodule findet sich z.B. in [CDI 94] und [BEGA 95].

Alle Anwendungen eines R/3-Systems gehen von der gleichen, abstrakten Sicht der Wirklichkeit, dem R/3-Informationsmodell, aus. Dieses Modell wird mit Hilfe

[1]Systeme, Anwendungen, Produkte in der Datenverarbeitung

einer zentralen, relationalen Datenbank realisiert. Diese gemeinsame Datenbasis ergibt, zusammen mit einer einheitlichen Benutzerschnittstelle, eine vollständige Integration der Anwendungen.

3.2 Architektur des R/3-Systems im Überblick

Im folgenden Abschnitt werden die wesentlichen Elemente eines R/3-Systems knapp vorgestellt. Einzelne Aspekte werden nur soweit vertieft, wie für die Entwicklung eines Managementkonzepts notwendig. Auf die aus Managementsicht besonders wichtige Architektur des R/3-Kernsystems wird im Detail in den Abschnitten 3.3 und 3.4 eingegangen.

Das System R/3 gliedert sich, wie in Abbildung 3.1 dargestellt, in die drei Ebenen R/3-Anwendungen, R/3-Kernsystem und Systemsoftware.

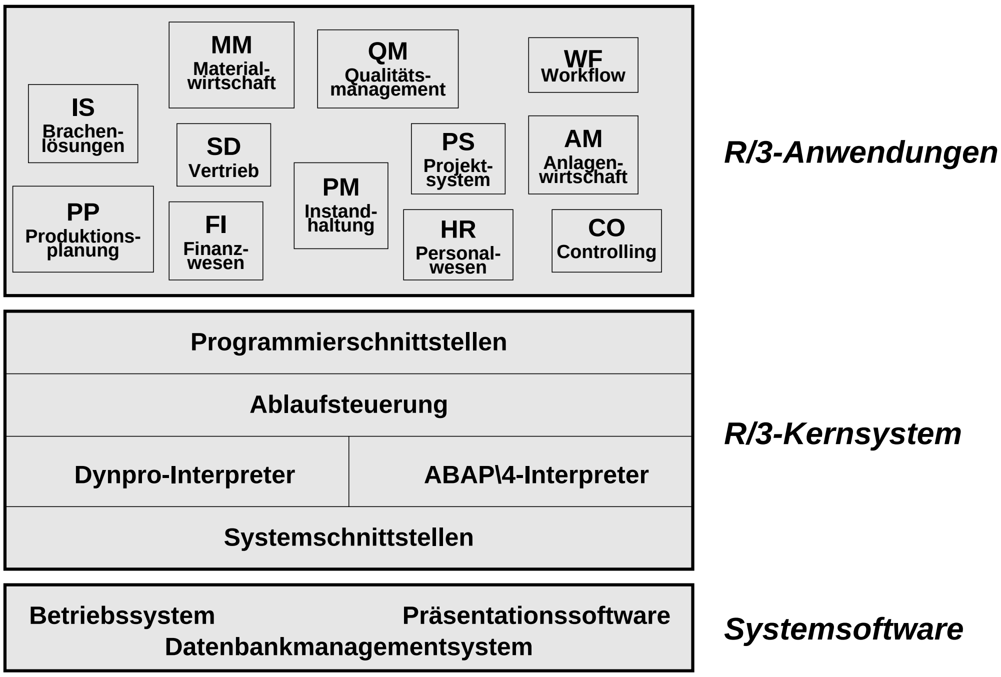

Abbildung 3.1: Ebenen der R/3-Architektur

3.2.1 R/3-Anwendungen

Als oberste Ebene eines R/3-Systems sind die R/3-Anwendungen anzusehen. Alle Anwendungslogik, die dem R/3-Benutzer zur Verfügung steht, ist in diesen Anwendungen enthalten. Diese Anwendungen sind in ABAP/4[2] codiert, einer von SAP entwickelten, interpretierten Programmiersprache speziell für betriebswirtschaftliche Anwendungen.

Die R/3-Anwendungen sind, wie die vom R/3-System verwalteten Daten, in einer zentralen Datenbank gespeichert. Sie werden nach Bedarf in das R/3-Kernsystem geladen und ausgeführt. Die R/3-Anwendungen sind also völlig unabhängig vom R/3-Kernsystem zu betrachten. Die sich daraus ergebenden Konsequenzen werden in Kapitel 4 erörtert.

3.2.2 R/3-Kernsystem

Alle zur Ausführung der R/3-Anwendungen benötigten Dienste sind in der mittleren Ebene, dem R/3-Kernsystem, realisiert. Im Detail werden die Konzepte des R/3-Kernsystems im Abschnitt 3.3 beschrieben. Die Realisierung wird in Abschnitt 3.4 erläutert.

Um die Portabilität des R/3-Systems zu sichern, ist das R/3-Kernsystem durch Präsentations-, Datenbank- und Betriebssystemschnittstelle von der Systemsoftware abgeschirmt. Aus Managementsicht besonders interessante Aspekte der Datenbank- und Betriebssystemschnittstelle werden in den Abschnitten 3.4.1.2 und 3.4.1.3 vorgestellt.

3.2.3 Systemsoftware

Die Systemsoftware, bestehend aus Betriebssystem, Datenbankmanagementsystem und Präsentationssoftware, bildet die Basis eines R/3-Systems. Sie versorgt das R/3-Kernsystem mit den benötigten Betriebsmitteln und sorgt für eine gewisse Abstraktion von der unterliegenden Hardware.

Das Datenbankmanagementsystem stellt dem R/3-System die zentrale Datenbank zur Verfügung. Datenbank und Datenbankmanagementsystem werden im folgenden auch gemeinsam als Datenbankserver bezeichnet. R/3 unterstützt die Verwendung relationaler Datenbanksysteme verschiedener Hersteller.

Auf Aspekte des Betriebssystems und der Präsentationssoftware wird, soweit nötig, bei der Beschreibung der Dienste und Server des R/3-Kernsystems eingegangen.

[2] Advanced Business Application Programming/4th Generation Language

3.3 Konzepte des R/3-Kernsystems

Die Architektur des Systems R/3 ist im wesentlichen durch zwei Merkmale gekennzeichnet:

Transaktionskonzept Im R/3-System wird ein 2-stufiges Transaktionskonzept verwendet. Aus diesem Transaktionskonzept und den Anforderungen der R/3-Anwendungen ergeben sich die Aufgaben des R/3-Kernsystems. In Abschnitt 3.3.1 wird dieses Konzept genauer vorgestellt.

Client-Server-Konzept Um die speziellen Anforderungen einzelner Unternehmen nach Skalierbarkeit und Ausbaufähigkeit des Systems erfüllen zu können, ist das System R/3 als Client/Server-Anwendung konzipiert. Diese Konzepte und ihre Elemente werden in Abschnitt 3.3.2 erläutert.

3.3.1 Transaktionskonzept des R/3-Systems

Das in Abbildung 3.2 dargestellte Transaktionskonzept des R/3-Systems basiert auf einer strikten Trennung der Transaktionsphasen Dialog und Verbuchung.

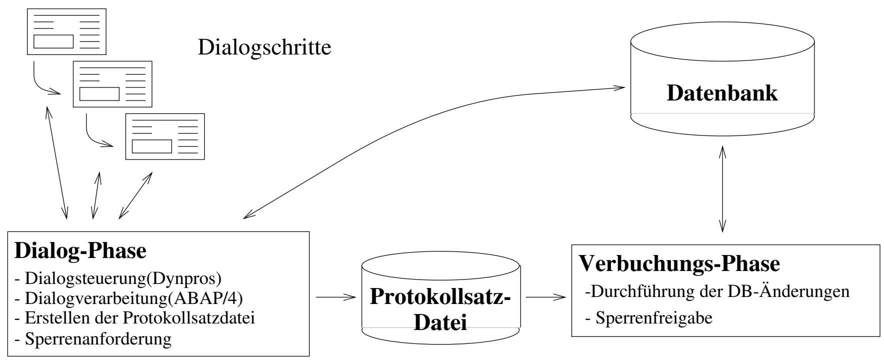

Abbildung 3.2: R/3-Transaktionskonzept (aus [R/3ARCHb])

Während des Benutzerdialogs werden Daten aus der Datenbank angefordert und die vom Benutzer veranlaßten Änderungen in einer sog. Protokollsatz-Datei dokumentiert. Erst nach Abschluß der Dialogphase wird diese Datei dem Verbuchungs-Dienst übergeben, der die nötig gewordenen Datenbankänderungen durchgeführt. Für den Benutzer bietet diese Asynchronität gleichmäßigere Antwortzeiten während des Dialogs, da eventuell aufwendigere Datenbankänderungen nicht abgewartet werden müssen.

Eine Folge dieser Trennung ist, daß Aufgaben wie die Verwaltung von Sperren auf Datenobjekte oder Recovery nach Systemfehlern nicht mehr dem Datenbankmanagementsystem überlassen werden können, sondern von R/3-System übernommen werden müssen. Die dazu erforderlichen Dienste und Server werden in Abschnitt 3.4 vorgestellt.

Neben diesen asynchronen Transaktionen können R/3-Anwendungen auch unmittelbare Datenbank-Transaktionen durchführen. Dies ist erforderlich, wenn Datenbankänderungen sofort allen R/3-Benutzern sichtbar werden sollen.

3.3.2 Client-Server-Konzept des R/3-Systems

Aus technischer Sicht lassen sich Anwendungen in die drei Bereiche Präsentation, Anwendungslogik und Datenhaltung trennen. Das dreistufige Client-Server-Konzept von R/3 spiegelt diese Aufteilung wider.

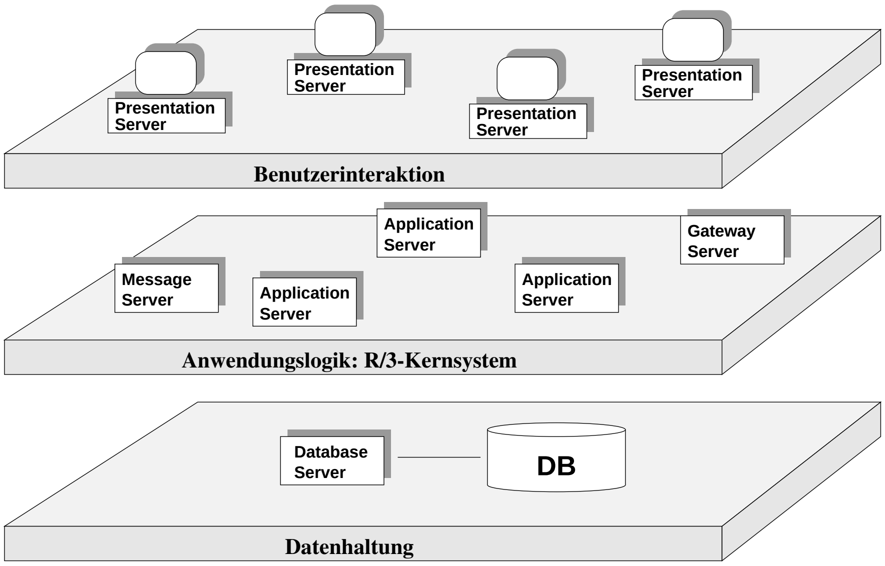

Abbildung 3.3: R/3 Client-Server Architektur

Das R/3-Transaktionskonzept gibt die Trennung von Anwendungslogik und Datenmanipulation vor. Die Schnittstelle Präsentation/Anwendungslogik wird durch die Client-Server-Beziehung zwischen Presentation-Server und Application-Server repräsentiert. Weitere Client-Server-Beziehungen finden sich innerhalb des R/3-Kernsystems. Die Komponenten dieser Ebene und ihre Beziehungen werden im folgenden Abschnitt erläutert.

3.4 Server und Dienste des R/3-Kernsystems

Dieser Abschnitt beschreibt die Komponenten des R/3-Kernsystems

- Application-Server

- Message-Server
- Gateway-Server

und die von ihnen angebotenen Dienste.

3.4.1 Application-Server

Die wichtigsten Komponenten eines R/3-Systems sind die Application-Server. Hier werden die in der zentralen Datenbank abgelegten R/3-Anwendungsmodule ausgeführt. Dazu werden folgende Dienste zur Verfügung gestellt:

Update-Service Der Update-Service (Verbuchungs-Dienst) ist für die Ausführung der in der Protokollsatz-Datei aufgelisteten Datenbankänderungen verantwortlich.

Enqueue-Service Der Enqueue-Service (Sperrenverwaltungs-Dienst) realisiert die Verwaltung von Sperren auf Datenobjekten.

Dialog-Service Der Dialog-Service (Dialog-Dienst) steuert den Benutzerdialog und ermöglicht eine Vorverarbeitung der Eingaben wie z.B. Wertebereichsüberprüfungen

Background-Service Die dialogfreie Datenübernahme in das R/3-System erfolgt über den Background-Service (Hintergrundverarbeitungs-Dienst).

Spool-Service Der Spool-Service (Druckdienst) erledigt alle Aufgaben im Zusammenhang mit der Druckerausgabe.

Ein Application-Server bietet mindestens Dialog-Service an, kann aber zusätzlich eine beliebige Teilmenge der genannten Dienste anbieten. Zu beachten ist, daß jeder Dienst mindestens einmal in einem R/3-System vorkommt. Der Enqueue-Service muß genau einmal vorhanden sein.

3.4.1.1 Architektur des Application-Servers

Technisch gesehen besteht ein Application Server aus mehreren nebenläufigen Prozessen, genauer aus einem Dispatcherprozeß und einem oder mehreren Workprozessen. Die genannten Dienste werden durch Workprozesse des entsprechenden Typs erbracht. Der Dispatcher nimmt die Dienst-Anforderungen der R/3-Anwendungen entgegen und legt sie zunächst in einer Warteschlange ab. Werden Workprozesse frei, ordnet ihnen der Dispatcher Aufträge aus dieser Queue zu.

Softwaretechnisch unterscheiden sich die Workprozesse nicht, jedoch ordnet der Dispatcher einem Workprozeß stets nur Aufträge eines Typs zu. So können in einem R/3-System auch Application Server existieren, die nur über Dialogprozesse verfügen, z.B. als dedizierte Workstation eines Benutzers. Die von einem Application-Server angebotenen Dienste lassen sich auch dynamisch zur Laufzeit verändern, wobei die Gesamtzahl seiner Workprozesse jedoch konstant bleibt.

Abbildung 3.4 zeigt die Architektur des Application-Servers sowie seine Schnittstellen. Zu den dargestellten Kommunikationsbeziehungen siehe auch 3.6.

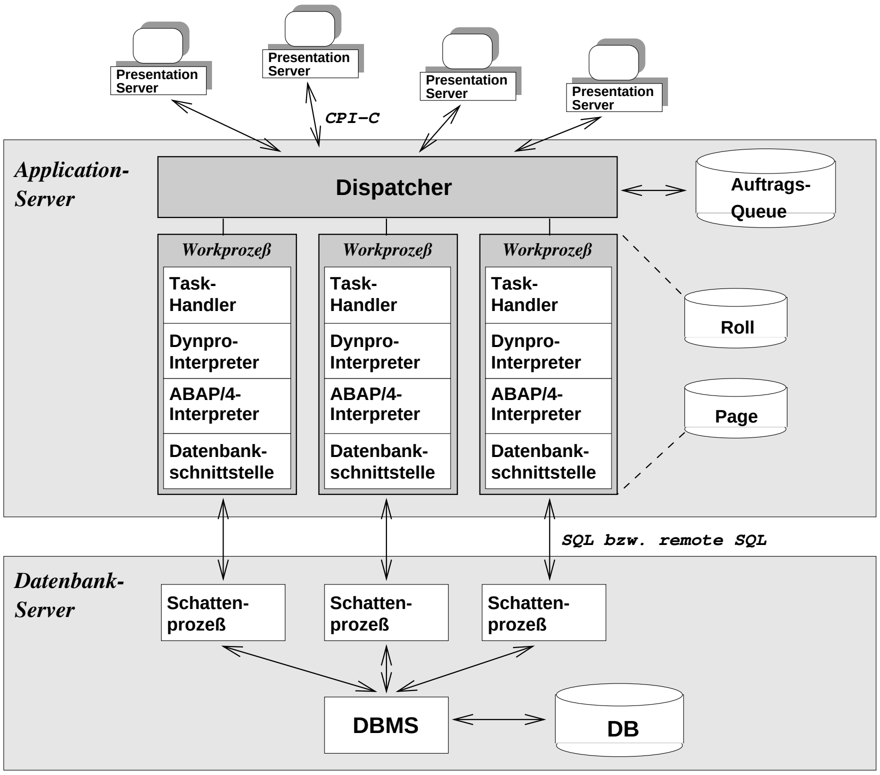

Abbildung 3.4: Architektur des Application-Servers (z.T. aus [R/3UNIX])

Workprozesse bestehen aus folgenden Komponenten:

Task-Handler Der Task-Handler übernimmt die Kommunikation mit dem Dispatcher und steuert die Abläufe innerhalb des Workprozesses.

Dynpro-Interpreter Zur Steuerung des Dialogs mit dem Benutzer werden sog. Dynpros[3] verwendet, für die dieser Interpeter existiert. Dynpros übernehmen die Erstellung der Bildschirmmasken und einfachere Überprüfungen der vom Benutzer eingegebenen Daten wie z.B. Wertebereichsprüfungen.

[3]Dynamische Programme

Diese Steuerprogramme werden als „dynamisch“ bezeichnet, da sie sich an Änderungen der Datenbeschreibung in der Datenbank anpassen können, was jedoch auch für ABAP/4-Programme gilt.

ABAB/4-Interpreter Der ABAB/4-Interpreter erledigt die Interpretation und Ausführung der R/3-Anwendungen.

Datenbankschnittstelle Die Datenbankschnittstelle realisiert die Kommunikation mit der zentralen Datenbank des R/3-Systems. Dazu baut jeder Workprozeß eine Verbindung zu einem sog. Schattenprozeß des Datenbanksystems auf. Weitere Eigenschaften der Datenbankschnittstelle siehe 3.4.1.2.

Die Bedeutung der dargestellten Roll- und Page-Dateien wird zusammen mit der Betriebssystemschnittstelle (siehe 3.4.1.3) erläutert.

3.4.1.2 Schnittstelle zum Datenbankserver

Die Datenbankschnittstelle bildet die von den R/3-Anwendungen verwendete Datensicht auf die relationalen Tabellen der Datenbank ab. Im allgemeinen sind diese Sichten identisch; aus Performancegründen oder wegen interner Beschränkungen des Datenbanksystems werden jedoch Unterschiede in Kauf genommen. Oft werden z.B. mehrere in R/3-Anwendungen verwendete Tabellen in sog. Pooltabellen zusammengefaßt gespeichert, da viele Datenbanksysteme Beschränkungen bezüglich der Tabellenanzahl aufweisen.

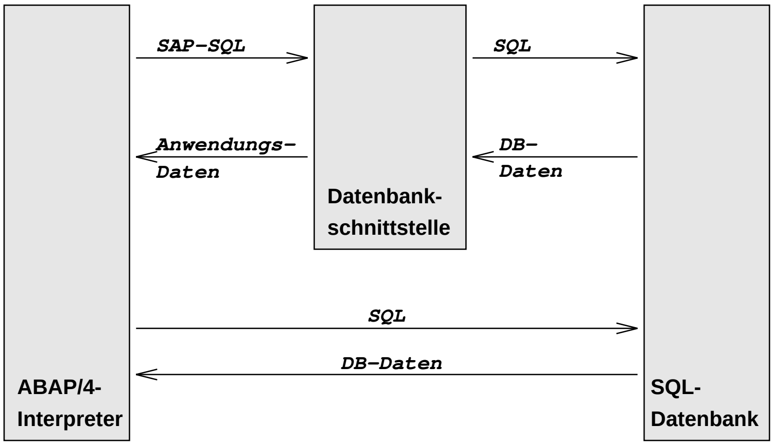

Abbildung 3.5: Die R/3-Datenbankschnittstelle (aus [R/3ARCHb])

Wie in Abbildung 3.5 dargestellt, erfolgt die Kommunikation zwischen Datenbankschnittstelle und Datenbanksystem über remote SQL. Um von speziellen Eigenschaften des verwendeten Datenbanksystems unabhängig zu sein, verwenden die R/3-Anwendungen jedoch i.a. einen im Sprachumfang eingeschränkten SQL-Dialekt, der als SAP-SQL bezeichnet wird. Die Umsetzung von SAP-SQL nach SQL ist ebenfalls Aufgabe der Datenbankschnittstelle. Die Datenbankschnittstelle beinhaltet auch einen Datenbank-Cache, um z.B. Anwendungsprogramme lokal halten zu können und so Datenbankzugriffe einzusparen.

3.4.1.3 Betriebssystemschnittstellen

Alle Komponenten eines R/3-Systems nutzen Dienste des Betriebssystems, jedoch sind die Betriebssystemschnittstellen des Application-Servers aus Managementsicht von besonderer Bedeutung.

Im Application-Server bildet ein R/3-System (im wesentlichen aus Performancegründen) elementare Betriebssystemfunktionen nach und entzieht sie so dem Systemmamangement. Im einzelnen sind dies folgende Funktionen:

Prozeß-Scheduling Mit der Dispatcher-Workprozeß-Architektur realisiert R/3 sein eigenes Scheduling. Die Kommunikation zwischen diesen Komponenten erfolgt über Shared-Memory-Bereiche.

Prozeßkontextwechsel und Speicherverwaltung Der Kontextwechsel zwischen den Aufgaben für einen Workprozeß erfolgt über den Roll-Bereich, d.h. der Kontext des Vorgängerauftrags wird dort gesichert und der des nächsten Auftrags geladen. Der Roll-Bereich kann vollständig im Hauptspeicher liegen, oder aber zum Teil in eine Datei ausgelagert sein. Von den Workprozessen gemeinsam genutzte Daten wie ABAP/4-Programme werden im Paging-Bereich gehalten. Die Puffer für die Roll- und Page- Bereiche im Arbeitsspeicher werden so groß gewählt, daß ein Paging des Betriebssystems möglichst unterbleibt.

3.4.2 Message-Server

Der nur einmal pro R/3-System vorhandene Message-Server ermöglicht den Application-Servern den Austausch von Nachrichten. Ein Anwendungsbeispiel ist die Suche nach R/3-Ressourcen im System. Will z.B. ein Application-Server eine Sperre auf ein Datenobjekt anfordern, so richtet er diese Anfrage zunächst an den Message-Server, der sie dann an den Application-Server mit Enqueue-Service weiterreicht. Dieser Dienst wird im folgenden auch als **Message-Service** bezeichnet.

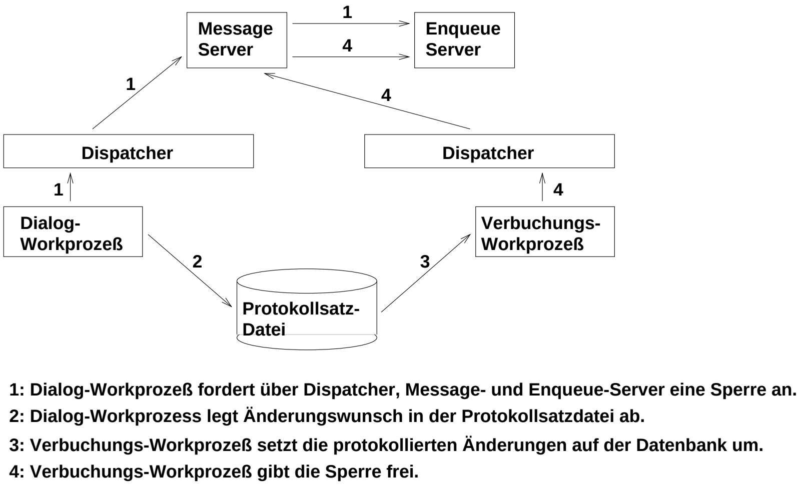

Abbildung 3.6: Realisierung des R/3-Sperrprotokolls (nach [R/3ADMIN])

3.4.3 Gateway-Server

Der Gateway-Server realisiert die Kommunikation zwischen R/3-Systemen und zu anderen Anwendungen. Dazu wird ein Protokoll auf Basis des CPI-C-Standards (siehe auch 3.6) verwendet.

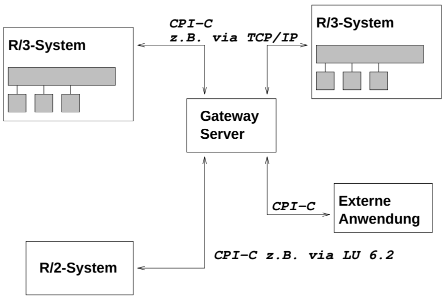

Abbildung 3.7: Aufgaben des Gateway-Servers (aus [R/3UNIX])

Im Gegensatz zum Message-Server, dessen Aufgabe die Vermittlung kurzer, systeminterner Nachrichten ist, ermöglicht der Gateway-Server den Transport größerer Datenmengen über Systemgrenzen hinweg. Eine Anwendung ist also z.B. die Übernahme von Fremddaten in ein R/3-System [NERB 95].

Die von diesem Server angebotenen Dienste werden im folgenden auch als **Gateway-Service** bezeichnet.

3.5 Presentation-Server

Der Presentation Server bildet das Frontend des R/3-Systems. Er realisiert die graphische Benutzerschnittstelle auf unterschiedlichen Systemplattformen und nutzt dazu systemspezifische Anwendungen wie z.B. ein Graphikpaket. Die Kommunikation zwischen Presentation-Server und Application-Server basiert ebenfalls auf CPI-C. Der Presentation-Server läuft i.a. nicht auf denselben Rechnern wie das R/3-Kernsystem, sondern entweder auf Endgeräten wie PCs und Workstations oder auf X-Terminal-Servern, an die wiederum X-Terminals als Endgeräte angeschlossen werden.

3.6 Kommunikation im R/3-System

R/3 verwendet zur Kommunikation zwischen den einzelnen Komponenten ein verbindungsorientiertes Protokoll auf Basis des CPI-C Standards[4]. CPI-C steht auf einigen Systemplattformen bereits zur Verfügung, andernfalls wurde es von SAP auf Basis der vorhandenen Transportsysteme implementiert. Abbildung 3.8 zeigt die unterstützten Protokollebenen.

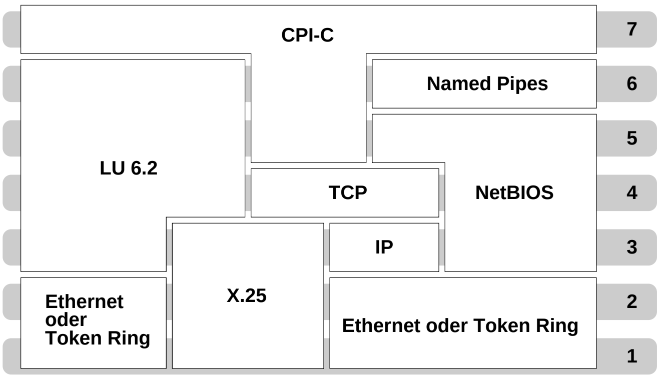

Abbildung 3.8: CPI-C Protokollebenen (aus [R/3ARCHa])

Die Kommunikation zwischen Dispatcher-Prozeß und seinen zugehörigen Workprozessen erfolgt über Shared-Memory-Segmente. Die Kommunikation zwischen

[4]Common Programming Interface for Communication

Workprozessen und Datenbank erfolgt über SQL bzw. über remote SQL, falls sich Application- und Datenbank-Server auf unterschiedlichen Rechnern befinden.

3.7 R/3-Instanzen

Die Server eines R/3-Systems auf einem Host werden zu sog. R/3-Instanzen zusammengefaßt, die stets gemeinsam gestartet und gestoppt werden. Aus Managementsicht kann man ein R/3-System also auch als Menge von R/3-Instanzen betrachten, welche die genannten Dienste anbieten. Dabei ist jedoch zu beachten, daß die Presentation-Server und der Datenbank-Server weder Bestandteil von R/3-Instanzen sind noch eigene R/3-Instanzen darstellen. Daraus ergeben sich einige Konsequenzen für das R/3-Management, auf die in Kapitel 5 eingegangen wird.

Zum besseren Verständnis der in den vorangegangenen Abschnitten beschriebenen R/3-Architektur und des Instanzen-Konzepts zeigt Abbildung 3.9 eine typische R/3-Konfiguration.

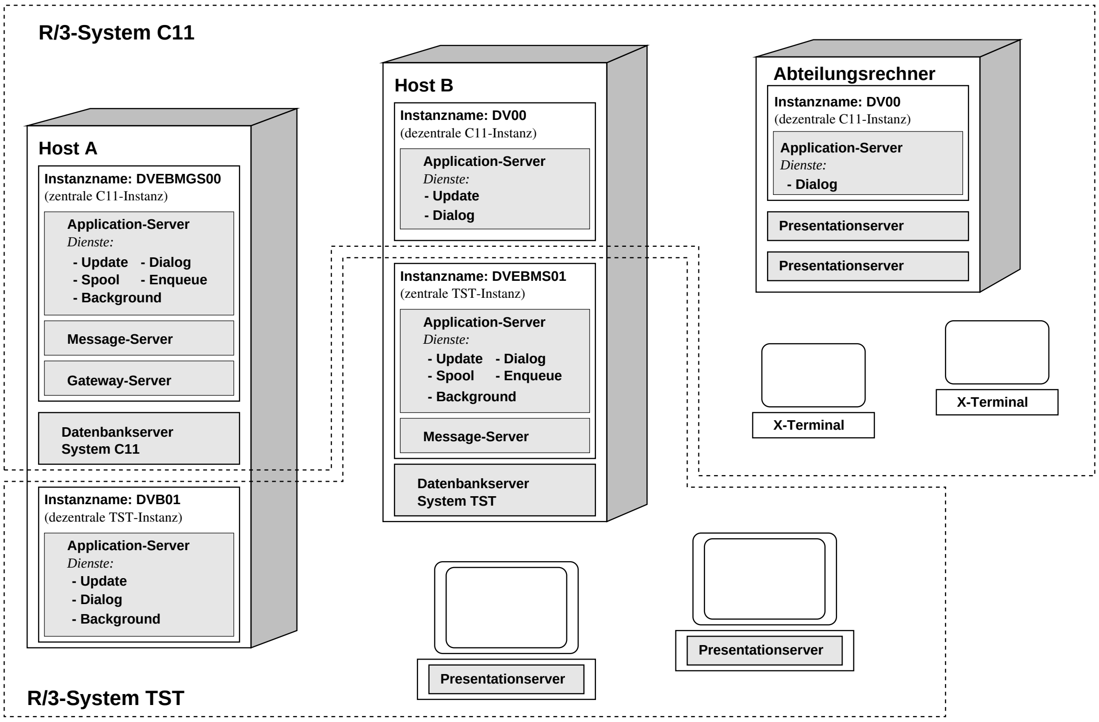

Abbildung 3.9: Typische R/3-Konfiguration

Die R/3-Systeme C11 (Produktiv-System) und TST (Testsystem) laufen verteilt auf Host A und Host B, wobei Host A den Zentralrechner von System C11 und Host B den Zentralrechner von System TST darstellt. Auf dem Zentralrechner befindet sich auch die Datenbank des jeweiligen Systems. Die Instanzen der Systeme bieten die dargestellten Dienste an. System TST besitzt keinen eigenen Gateway-Server, sondern benutzt das Gateway von System C11 mit.

Die Systeme werden nach Bedarf um Abteilungsrechner mit Dialog-Instanzen erweitert (hier beispielhaft für System C11 dargestellt). Um nachts den Background-Durchsatz zu erhöhen, können diese dezentralen Instanzen auf Background-Service umgeschaltet werden.

Die Abbildung zeigt auch verschiedene Möglichkeiten, Endgeräte an ein R/3-System anzuschließen. An das System TST sind Endgeräte mit eigenem Presentation-Server angeschlossen (PCs, Workstations, ...). Dagegen werden am System C11 X-Terminals als Endgeräte verwendet, deren Presentation-Server auf einem der dezentralen Rechner läuft.

3.8 Anmerkungen zur R/3-Architektur

3.8.1 Zentralistische Architektur

Bei der Betrachtung der R/3-Architektur fällt auf, daß die theoretisch denkbaren Möglichkeiten zur Aufgabenverteilung innerhalb des R/3-Systems nur in beschränktem Umfang realisiert sind. Beispiele hierfür sind:

- Der Datenbankserver als einzige zentrale Komponente eines R/3-Systems beschränkt die Größe und Leistung des gesamten Systems. Die Verwendung verteilter Datenbanksysteme ist bisher nicht vorgesehen.
- Eine automatische, lastabhängige Verteilung der Dienstanforderungen über die Application-Server findet nicht statt. Die Verteilung der Benutzeraufträge ist durch die Anmeldung des Benutzers an einen Application-Server vorgegeben. Hintergrund-Jobs und periodisch auszuführende System-Jobs werden vom Systemadministrator auf die Server verteilt. Dazu stehen ihm jedoch unterstützende Werkzeuge (siehe 5.1.1) zur Verfügung.
- Die Anzahl der Workprozesse pro Dienst-Typ wird nicht dynamisch entsprechend der Menge eintreffender Aufträge verändert. Vielmehr muß der Systemadministrator die Warteschlangen überwachen und das System gegebenenfalls neu konfigurieren.

3.8.2 Softwaretechnik

Die Architektur des Application-Servers, bestehend aus Dispatcher und Workprozessen, erscheint heute ungewöhnlich, da man dieses Konzept auch mit Hilfe sog. leichtgewichtiger Prozesse (Threads) realisieren könnte. Diese standen aber zur Zeit der Entwicklung von R/3 noch nicht mit der erforderlichen Stabilität zur Verfügung.

Kapitel 4

Anforderungen an ein integriertes R/3-Management

In diesem Kapitel werden zunächst allgemeine Anforderungen an die Überwachung verteilter Anwendungen formuliert, die dann an Hand der speziellen Managementaspekte des R/3-Systems verfeinert werden. Auf dieser Grundlage wird ein abstraktes Managementmodell für R/3 entwickelt, das insbesondere die Anforderungen an die erforderlichen Managementobjekte bestimmt.

4.1 Managementaspekte verteilter Anwendungen

Um die vielfältigen Managementaspekte einer verteilten Anwendung zu erfassen und zu strukturieren, eignen sich die Specific Management Functional Areas der ISO:

- Performance Management
  - Leistungsmessung und Überwachung.
  - Festlegung von Dienstgüteparametern (QoS, Quality of Service).
  - Aufbereitung und Visualisierung der Leistungsdaten.
- Fault Management
  - Alarmgenerierung und Verarbeitung.
  - Fehlerdiagnose und Behebung.
- Configuration Management

  - Überwachung und Steuerung der Systemkonfiguration an Hand der durch die Anwendungsarchitektur vorgegebenen Randbedingungen.
  - Veränderung der Systemkonfiguration an Hand der Leistungsdaten.

- Security Management
  - Benutzerspezifische Verwaltung und Überwachung von Zugriffsrechten.
  - Möglichkeiten zur selektiven Vergabe von Rechten auf Managementoperationen für unterschiedliche Aufgabenbereiche und Personengruppen.

- Accounting Management
  - Benutzerverwaltung
  - Erfassung und Zuordnung des Ressourcenverbrauchs.

Neben dieser funktionalen Gliederung lassen sich Managementaufgaben auch in aktive und passive Aufgaben unterteilen. Passive Managementaufgaben sind alle Überwachungsaufgaben (Monitoring), aktive Managementaufgaben erfordern Eingriffe in das Netz (Controlling).

Da sich diese Arbeit als erste im MNM-Team mit dem R/3-Anwendungsmanagement befaßt, soll im folgenden der Aspekt der Verfügbarkeit, also insbesondere die Überwachung eines R/3-Systems vertieft werden. Diese Einschränkung ist notwendig, um sich nicht in der Vielzahl der R/3-Managementaspekte zu verlieren.

Darüber hinaus ist diese Einschränkung für einen ersten Integrationsansatz sinnvoll, da eine vollständige Integration des R/3-Managements in eine Managementplattform mit immensem Aufwand verbunden ist. Es ist daher zunächst eine Integration des Management-Aspekts Überwachung anzustreben, weitergehende Managementaufgaben jedoch den bereits vorhandenen, hochspezialisierten R/3-Managementwerkzeugen zu überlassen. Diese können jedoch zur leichteren Bedienung in die Managementplattform oberflächenintegriert [ABEC 95] werden.

Aus technischer Sicht besteht eine verteilte Anwendung aus einzelnen Anwendungskomponenten, die durch Zusammenarbeit entsprechend einem Kooperationsmodell (meist Client-Server-Prinzip) die von der Gesamtanwendung zu erbringenden Dienste (Anwendungsdienste) realisieren.

Die einzelnen Komponenten können dabei sowohl in der Client-Rolle auftreten, in der sie Dienste anderer Komponenten nutzten, als auch in der Server-Rolle, indem sie Dieste für andere Komponenten erbringen. Diese anwendungsspezifischen Dienste, die zur Erbringung der Anwendungsdienste benötigt werden, bezeichnet man als Basisdienste.

Der folgende Abschnitt formuliert zunächst Managementanforderungen für die einzelnen Komponenten, Abschnitt 4.1.2 beschreibt Anforderungen zur Überwachnung des Gesamtsystems.

4.1.1 Managementaspekte der Anwendungskomponenten

In diesem Abschnitt werden Anforderungen an Managementobjekte formuliert, die zur Überwachnung einzelner Anwendungskomponeneten erforderlich sind. Diese Anforderungen dienen später zur Konzeption geeigneter Agenten. Die vorgestellten Konzepte sind so angelegt, daß die Komponente selbst ein Bild ihres Zustands ermittelt und zur Verfügung stellt. Diese Vorverarbeitung der Managementinformation bereits in der Anwendungskomponente soll die Umsetzung der im nächsten Abschnitt besprochenen Managementanforderungen an das Gesamtsystem vereinfachen.

Die Überwachung einzelner Anwendungskomponeneten gliedert sich in die Bereiche Systemressourcen, Basisdienste und Konfiguration.

Systemressourcen Um eine Bewertung des Zustandes einer Anwendungskomponente vornehmen zu können, müssen neben den Basisdiensten auch die benötigten Systemressourcen überwacht werden. Die Ermittlung von Konfigurations- und Leistungsdaten zu den relevanten Ressourcen kann sowohl durch die Anwendungskomponente selbst als auch durch Systemmanagementagenten erfolgen. Welche Ressourcen konkret wichtig sind und wie die Schwellwerte zu ihrer Überwachung gewählt werden müssen, ist vom Typ der jeweiligen Komponente abhängig.

Basisdienste Die Verfügbarkeit der Basisdienste stellt natürlich den wichtigsten zu überwachenden Aspekt einer Anwendungskomponente dar. Man könnte annehmen, daß die Überwachung der Systemressourcen bereits die Verfügbarkeit der Basisdienste sichert. Es zeigt sich jedoch, daß dies allenfalls für einfache Anwendungen zutrifft. Ein einfaches Gegenbeispiel stellen Datenbanksysteme dar:

Die Vorraussetzung für ein korrektes Arbeiten eines DBMS ist natürlich die Verfügbarkeit der benötigten Systemressourcen. Es gibt jedoch Zustände, die zu Leistungseinbrüchen im Datenbanksystem führen, aber nicht durch die Überwachung der Systemparameter erkannt werden können. Dazu gehören z.B. sog. Deadlock-Zustände, die immer dann auftreten, wenn Transaktionen wechselseitig auf die Freigabe von Datenobjekten warten. Bis zur Auflösung eines solchen Zustandes zeigt das DBMS zwar normalen Systemressourcen-Verbrauch, die Transaktions-Leistung sinkt aber dennoch.

Derartige Probleme ergeben sich zwangsläufig immer dann, wenn Anwendungen Systemfunktionen selbst übernehmen wie z.B. die Sperrenverwaltung auf Datenobjekte im obigen Beispiel. Relevante Managementinformation zu diesen Aspekten kann dann nur noch von der Anwendung selbst bereitgestellt werden.

Deshalb müssen zu jedem angebotenen Basisdienst-Typ relevante Leistungsdaten definiert und erhoben werden. Die Verfügbarkeit wird dann an Hand einer mindestens erforderlichen Dienstgüte (Quality of Service, QoS) ermittelt, die durch Schwellwerte für die einzelnen Leistungsparameter des Dienstes definiert ist. Die Konfiguration der Schwellwerte sollte von der Managementstation aus veränderbar sein. Je nach Bedeutung der Komponente können Meldungen an die Managementstation bereits bei einzelnen Schwellwertverletzungen erfolgen oder aber erst bei Ausfall bzw. ungenügender QoS des gesamten Dienstes.

Konfiguration Um einen Überblick über das Gesamtsystem gewinnen zu können, muß jede Komponente bei Start und Beendigung Meldungen an die Managementstation schicken. „Start"-Meldungen müssen insbesondere die von der Komponente angebotenen Basisdienste enthalten, um so die Gesamtkonfiguration des Systems feststellen zu können. Jede Änderung der Konfiguration muß ebenfalls gemeldet werden.

4.1.2 Managementaspekte des Gesamtsystems

In den Managementbereich des Gesamtsystems fallen alle Aufgaben, die einen globalen Überblick über die verteilte Anwendung erfordern. Die Realisierung der hier spezifizierten Anforderungen erfolgt i.a. in der zentralen Managementstation, oder aber auch dezentral innerhalb zentraler Anwendungskomponenten, die dann nur noch Meldungen an die Managementstation weiterleiten.

Um die Verfügbarkeit des Gesamtsystems bewerten zu können, ist die Überwachung der Systemkonfiguration und der Gesamtleistung erforderlich.

Systemkonfiguration An Hand der Meldungen der einzelnen Anwendungskomponenten muß eine Übersicht der insgesamt verfügbaren Basisdienste erstellt und bezüglich der architekturspezifischen Randbedingungen überprüft werden. Diese Konfiguration muß in regelmäßigen Zeitintervallen durch Polling überprüft werden, um Ausfälle von Anwendungskomponenten festzustellen. Die Konfigurationsdaten werden außerdem zur Erstellung von graphischen Darstellungen der Anwendungstopologie benötigt.

Gesamtleistung Durch Sammeln der Leistungsdaten der einzelnen Komponenten werden Parameter für die Gesamtleistung der Anwendung ermittelt. Die Überwachung erfolgt wieder durch Schwellwerte. Eine Neuberechnung sollte insbesondere immer dann erfolgen, wenn Anwendungskomponenten Performanceprobleme melden.

4.2 Spezielle Managementaspekte von SAP R/3

In den folgenden Abschnitten werden die zuvor allgemein vorgestellten Verfügbarkeitsmanagement-Aspekte für die Anwendung SAP R/3 konkretisiert. Dazu wird zunächst das Gesamtziel „Verfügbarkeitsüberwachung eines R/3-Systems“ genauer spezifiziert. Davon ausgehend werden Managementanforderungen für die R/3-Komponenten und das Gesamtsystem erstellt.

Die Verfügbarkeit eines R/3-Systems läßt sich folgendermaßen definieren:

> **Der R/3-Anwender kann die R/3-Anwendungsmodule mit ausreichender Dienstgüte nutzen.**

Aus der Architektur des R/3-Systems ergibt sich sofort, daß die Verfügbarkeit der R/3-Anwendungen genau dann gegeben ist, wenn die von ihnen benötigten R/3-Basisdienste mit ausreichender QoS zur Verfügung stehen. Für den Verfügbarkeitsaspekt ist also die Betrachtung der Basisdienste ausreichend.

4.2.1 Managementaspekte der R/3-Instanzen

Neben dem Datenbankserver sind die R/3-Instanzen die wichtigsten Komponenten eines R/3-Systems, da sie die R/3-Basisdienste für die R/3-Anwendungen bereitstellen. Die in Abschnitt 4.1.1 allgemein formulierten Anforderungen werden hier für R/3-Instanzen konkretisiert. Die im Unterpunkt „Konfiguration“ genannten Managementanforderungen gelten unverändert für R/3-Instanzen und werden hier nicht weiter vertieft.

4.2.1.1 Systemressourcen

Die folgende, auf Erfahrungswerten der SAP AG beruhende Liste, gibt eine Übersicht über die für eine R/3-Instanz zu überwachenden Systemressourcen. Soweit möglich sind auch sinnvolle Schwellwerte angegeben, bei deren Verletzung Meldungen an die Managementstation generiert werden müssen.

CPU Die Auslastung der Prozessoren läßt sich an Hand der Anzahl vor den CPUs wartender Prozesse (Load) und des prozentualen Anteils von Wartezyklen pro Zeiteinheit (Idle) ermitteln. Dabei sollte der Load-Wert stets kleiner als 3 sein. Liegt der Idle-Wert nahe 0%, ist der Prozessor überlastet. Wichtig ist ferner die Anzahl der Prozeßkontextwechsel pro Zeiteinheit, die nicht über 200/sec liegen sollte.

Speicherverwaltung Um Paging des Betriebssystems zu vermeiden, sollten stets mehr als 10MB freier Hauptspeicher zur Verfügung stehen.

Dateisystem Überwachungsparameter sind hier der belegte/verfügbare Plattenplatz, die Anzahl der belegten/verfügbaren Dateideskriptoren sowie die Anzahl der Zugriffe pro Zeiteinheit. Von besonderem Interesse ist dabei die Auslastung der SWAP-Bereiche und der Puffer des R/3-Systems. Einige dieser Puffer dürfen nie zu 100% gefüllt sein, da dies zu Fehlern im R/3-System führt. Traps sollten daher spätestens bei 95% Auslastung generiert werden.

Syslog Fehlerzustände des Systems und der Anwendungskomponenten werden in der Syslog-Datei protokolliert und entsprechend ihrer Bedeutung in 8 Klassen eingeteilt. Wichtige Einträge in Syslog-Dateien müssen erkannt und mittels Traps an die Managementstation weitergeleitet werden.

4.2.1.2 Basisdienste

Aus den in Abschnitt 4.1.1 genannten Gründen müssen für die nachfolgend aufgelisteten R/3-Basisdienste Leistungsdaten ermittelt werden, um ihre Verfügbarkeit bewerten zu können. Für die durch einzelne Prozesse realisierten Basisdienste Message und Gateway reicht die Überwachung der Existenz des entsprechenden Prozesses und der Systemressourcen aus. Für die übrigen Basisdienste ist im folgenden angegeben, welche Parameter ihre Verfügbarkeit definieren und wie die Schwellwerte zu wählen sind.

Dialog Aus Benutzersicht ist die Performance des Dialog-Dienstes, die natürlich auch von der Performance der anderen Basisdienste abhängig ist, am wichtigsten. Die Antwortzeiten sollten i.a. unter 1 bis 2 Sekunden liegen.

Background, Update Die aktuelle Dienstgüte dieser Dienste ergibt sich aus der Wartezeit auf einen freien Workprozess und der Ausführzeit. Diese ist stark von der Leistung des Datenbanksystems abhängig. Lesende DB-Zugriffe sollten im Mittel nicht länger als 10ms dauern, schreibende nicht länger als 25ms.

Spool Wenn sich Druckaufträge länger als 30min in der Warteschlange befinden, muß der Spool-Service als ausgefallen betrachtet werden.

Enqueue Für die Verfügbarkeit des Enqueue-Service ist es wichtig zu überwachen, daß weder die Sperrtabelle im Arbeitsspeicher noch die zugehögige Sperrendatei ihre maximal zulässige Größe erreichen. Bei einem Füllungsgrad von ca. 95% muß ein Trap generiert werden. Ferner sollte überwacht werden, ob einzelne Sperren sehr lange (> 30min) gehalten werden. Dies deutet auf Deadlockzustände hin, die eventuell vom Administrator aufgelöst werden müssen.

4.2.2 Managementaspekte des R/3-Kernsystems

Die Hauptaufgabe bei der Überwachung des gesamten, aus einzelnen R/3-Instanzen aufgebauten R/3-Kernsystems, liegt in der Überwachung der Korrektheit und Vollständigkeit der Systemkonfiguration bezüglich der insgesamt angeboteten Basisdienste. Aus der in Kapitel 3.3 beschriebenen R/3-Architektur ergeben sich folgende Anforderungen:

- Der Enqueue- und Message-Service müssen von genau einer R/3-Instanz angeboten werden.
- Jede R/3-Instanz sollte Dialog-Service anbieten.
- Alle übrigen R/3-Dienste können beliebig oft vorhanden sein, müssen aber von wenigstens einer R/3-Instanz angeboten werden.

Zur Überwachung dieser Anforderungen, müssen die Meldungen der einzelnen R/3-Komponenten ausgewertet und mit diesen Mindestanforderungen verglichen werden. Dabei wird nicht nur der Neustart bzw. Wegfall ganzer Komponenten berücksichtigt, sondern es werden auch Performance-Traps ausgewertet, um die tatsächliche Verfügbarkeit der Basisdienste einer Komponente zu ermitteln. Zur Erkennung von Totalausfällen einzelner R/3-Komponenten muß ferner regelmäßiges Polling durchgeführt werden.

Die Berechnung einer mittleren Leistung für Diensttypen, die von mehreren R/3-Komponenten angeboten werden, erscheint nicht sinnvoll, da ein Benutzer stets nur Dienste einer R/3-Instanz nutzt.

4.2.3 Managementaspekte des Datenbank-Servers

Obwohl der Datenbank-Server keine Komponente des R/3-Kernsystems darstellt, ist er für das R/3-System als Ganzes von zentraler Bedeutung, da er eines der wenigen nicht redundant vorhandenen Betriebsmittel eines R/3-Systems darstellt. Performance-Mängel oder gar ein Ausfall des zentralen Datenbank-Servers führen sofort zu Beeinträchtigungen im Benutzerbetrieb, da manche Arten von Datenbankänderungen so zeitkritisch sind, daß sie nicht nach dem asynchronen R/3-Transaktionskonzept durchgeführt werden können, sondern unmittelbar als Datenbanktransaktion ausgeführt werden müssen. Totalausfälle des Datenbank-Servers führen also nach kürzester Zeit zum Stillstand eines R/3-Systems, Performance-Mängel beschränken die Leistung des gesamten R/3-Systems.

Aus Managementsicht ist der Datenbank-Server also von herausragender Bedeutung, und es ergeben sich die folgenden Managementaspekte:

- Ein rein auf Fehlerbehebung ausgelegtes Management ist für den Datenbank-Server wegen der genannten Folgen nicht akzeptabel. Es ist vielmehr notwendig, Probleme des Datenbank-Servers frühzeitig zu erkennen und zu beheben.
- Um ein solches vorausschauendes Management verwirklichen zu können, müssen wenigstens folgende Ressourcen und Parameter des Datenbankservers überwacht und kritische Werte an die Managementstation gemeldet werden:
  - Performance
  - Freier Speicherplatz
  - Systembedingte Bechränkungen wie z.B. die maximale Tabellenzahl
- Daneben sollte der DB-Server selbständig Meldungen über besondere Systemzustände wie z.B. Deadlock-Zustände zur Verfügung stellen, da diese nach außen nur als Performance-Verschlechterung sichtbar werden.

Um die zur Erfüllung dieser Anforderungen notwendigen Informationen zu bekommen, stehen zwei Möglichkeiten zur Wahl:

- Die Managementstation erhebt die Daten direkt vom DB-Server, z.B. mittels eines Agenten und übernimmt die Auswertung dieser Daten.
- Die R/3-Komponenten, die den DB-Server als Ressource nutzen, überwachen selbst die kritischen Parameter und geben nur bei Problemen Meldungen an die Managementstation weiter.

Die zweite Alternative erscheint aus mehreren Gründen besonders günstig. Zum einen wird die Managementinformation dezentral vorverarbeitet, zum anderen erspart sie die Nachbildung umfangreichen Managementwissens in der Managementplattform, welches wegen der speziellen Eigenschaften der R/3-Datenbankschnittstelle notwendig ist, um eine aussagekräftige Bewertung der Informationen über den Datenbank-Server vornehmen zu können.

4.2.4 Managementaspekte des Presentation-Servers

Wegen der einfachen Architektur und der Tatsache, daß ein Presentation-Server ohnehin unter der ständigen Überwachung seines Benutzers steht, ergeben sich nur wenige Managementanforderungen.

Um Anfragen von R/3-Benutzern wegen Funktionsstörungen des R/3-Systems besser bearbeiten zu können, ist es erforderlich von der Managementstation aus Zugriff auf folgende Daten des Presentation-Server-Systems zu haben:

- Allgemeine Systemmanagement-Parameter
- Prozeßdaten zum Presentation-Server-Prozeß
- Informationen zur installierten Presentation-Server-Version

Mit Hilfe dieser Informationen läßt sich dann leicht feststellen ob es sich tatsächlich um ein Problem des R/3-Systems handelt oder ob das Problem auf Seite des Presentation-Server liegt.

4.3 Managementmodell

Abbildung 4.1 zeigt ein abstraktes, unter Managementgesichtspunkten entworfenes Modell eines R/3-Systems. Die einzelnen Ebenen enthalten Managed Objects, deren Attribute sich zum einen aus den beschriebenen Anforderungen und zum anderen aus den Beziehungen zu den benachbarten Schichten ableiten. Die Beziehungsattribute ermöglichen eine strukturierte Darstellung der Managed Objects in der Managementplattform.

Dieses Modell bildet auch die Grundlage zum Entwurf einer R/3-MIB und für die Entwicklung eines anforderungskonformen R/3-Managementmoduls.

4.3.1 Anwendungsdienste-Ebene

Die Objekte dieser Ebene repräsentieren Anwendungsdienste, die dem Benutzer zur Verfügung stehen. Attribute sind im wesentlichen die Verfügbarkeit bzw. Auslastung und die benötigten Basisdienste.

Zur Zeit ist die Anwendungsdienste-Ebene jedoch nur von konzeptioneller Bedeutung, da im R/3-Kernsystem keine Information darüber verwaltet wird, welche Anwendung welche Dienste nutzt. Es ist daher davon auszugehen, daß eine Anwendung i.a. auf alle R/3-Basisdienste angewiesen ist und so auf eine Modellierung der Relation „Anwendung → *benötigt* → Basisdienst“ verzichtet werden kann.

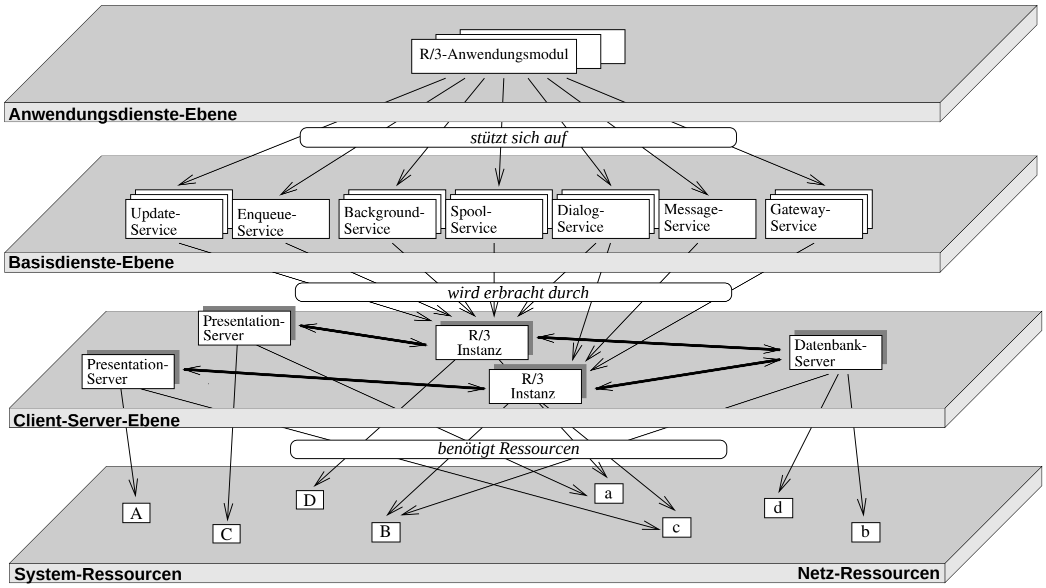

Abbildung 4.1: Managementmodell von SAP R/3

Statt dessen wird auf dieser Abstraktionsebene ein Objekt eingeführt, welches das R/3-System als Ganzes repräsentiert und dazu die in 4.2.2 definierten Überwachungsaufgaben durchführt. Der Zustand dieses Objekts wird nur aus dem Zustand der R/3-Basisdienste abgeleitet, da diese den Zustand der vom Datenbankserver angebotenen Dienste bereits ausreichend berücksichtigen.

Der Betriebszustand der Presentation-Server geht nicht in die Zustandsberechnung für das R/3-Systemobjekt ein, da Störungen einzelner Presentation-Server keine Auswirkung auf das Gesamtsystem haben.

4.3.2 Basisdienste-Ebene

Auf dieser Ebene finden sich Managed Objects, welche die von den R/3-Instanzen angebotenen Basisdienste repräsentieren. Die Bestimmung der aktuellen Verfügbarkeit erfolgt mit den in 4.2.1.2 genannten Methoden. Die dazu erforderliche Managementinformation wird von den Objekten der Client-Server-Ebene bereitgestellt.

4.3.3 Client-Server-Ebene

Die Managed Objects dieser Ebene repräsentieren die Komponenten eines R/3-Systems, also R/3-Instanzen, Datenbanksystem und Presentation-Server. Attribute dieser Objekte sind:

- Systemprozesse, welche die Komponente realisieren.
- Client-Server-Relationen zu anderen Komponenten dieser Ebene.
- Angebotene Dienste sowie Leistungsdaten zu diesen Diensten.
- Anforderungen an Ressourcen-Objekte aus der Netz- und Systemebene.

4.3.4 Netz- und System-Ebene

Diese Ebene enthält Managed Objects, die für die Managementanforderungen der oberen Schichten geeignete Abstraktionen der realen Netz- und Systemressourcen darstellen. Die Informationen über Systemressourcen erlauben den Objekten der Client-Server-Ebene detailliertere Auskünfte über ihren Zustand zu geben, also insbesondere auf Störungen hinzuweisen, die durch Mängel an den Systemressourcen verursacht werden. Störungsmeldungen von Netzobjekten müssen aus demselben Grund beachtet werden, außerdem kann es hier notwendig sein, Fehlermeldungen zu unterdrücken, die auf bereits bekannte Netzprobleme zurückzuführen sind.

Kapitel 5

Werkzeuge und Schnittstellen für das R/3-Management

In diesem Kapitel werden zunächst die proprietären R/3-Managementwerkzeuge in ihrem Funktionsumfang beschrieben. Anschließend werden die in Kapitel 2 vorgestellten Quellen standardisierter Managementinformation in den Kontext des R/3-Managements eingeordnet. Daraus ergibt sich die Notwendigkeit einer R/3-spezifischen MIB, deren Entwurf im weiteren Verlauf des Kapitels diskutiert wird.

5.1 R/3-Element-Management-Systeme

5.1.1 Computing Center Management System (CCMS)

Das Computing Center Management System (CCMS) bietet eine Vielzahl von Diensten zur Administration, Überwachung und zum Tuning eines R/3-Systems. Bereits die Installation des Systems wird durch das CCMS unterstützt. Das CCMS ist vollständig in die graphische Benutzeroberfläche des R/3-System integriert.

Die Managementfunktionen des CCMS beschränken sich nicht auf das R/3-System selbst, sondern es werden auch Konfigurations- und Leistungs-Daten aus den Bereichen Datenbank, Betriebssystem und Netz erfaßt. Zu den Netz- und Systemressourcen werden die Daten mit Hilfe plattformspezifischer Datenkollektoren erhoben. Informationen zum Datenbanksystem werden durch SQL-Anfragen an die Pseudo-Tabellen des Datenbankmanagementsystems gewonnen.

Die Erfassung von Managementinformation zu R/3-Ressourcen bereits durch das R/3-System selbst ist aus zwei Gründen erforderlich:

- Bei kleineren Unternehmen, die im wesentlichen nur R/3-Anwendungen nutzen, sollen Netz- und Systemüberwachung weitgehend vom R/3-System durchgeführt werden, um auf eine Managementplattform verzichten zu können.

- Aus den in Kapitel 4 genannten Gründen benötigt das R/3-System Informationen zu seinen Ressourcen, um aussagekräftige Information zum eigenen Zustand bereitstellen zu können. Diese Interpretation könnte in der Managementanwendung nur mit erheblichen Aufwand durchgeführt werden, da hierzu umfangreiches Wissen zur internen Struktur von R/3 modelliert werden müßte.

Neben der Darstellung des aktuellen Systemzustandes ermöglicht das CCMS auch Auswertungen über längere Zeiträume.

5.1.2 Early Watch Support

Der Early-Watch-Dienst stellt eine Erweiterung des CCMS um R/3-spezifisches Managementwissen dar. Durch dieses Wissen sollen Prognosen bezüglich der Entwicklung eines R/3-Systems ermöglicht werden, um drohende Engpässe rechtzeitig erkennen und so ein vorausschauendes Management verwirklichen zu können.

Die Durchführung einer solchen Analyse mit Hilfe der Early-Watch-Werkzeuge kann sowohl direkt vom jeweiligen R/3-Administrator vor Ort vorgenommen werden als auch über X.25 Verbindung von R/3-Experten der SAP. Bei letzterer Lösung besteht zusätzlich die Möglichkeit, die Daten mit denen anderer R/3-Anwender zu vergleichen.

5.2 Kommunikationsschnittstellen

Im vorangegangenen Kapitel wurden die relevanten Managementaspekte eines R/3-Systems identifiziert und Anforderungen an die benötigte Managementinformation formuliert. Abbildung 5.1 zeigt einige Möglichkeiten zur Gewinnung dieser Managementinformation über Kommunikationsschnittstellen zum R/3-System.

Für das Management der Presentation-Server eignen sich die in der Host Resources MIB (siehe 2.2.1) definierten Objekte. Für die beschriebenen Überwachungsaufgaben reicht diese MIB aus. Falls Windows-PCs als Presentation-Server eingesetzt werden, bietet sich auch die Nutzung der WinWatch-MIB (siehe 2.2.2) an. Durch den Einsatz dieses Agenten eröffnet sich auch die Möglichkeit durch Erstellen einer neuen DLL Presentation-Server-spezifische Managementinformation bereitzustellen, falls sich dies als notwendig erweisen sollte.

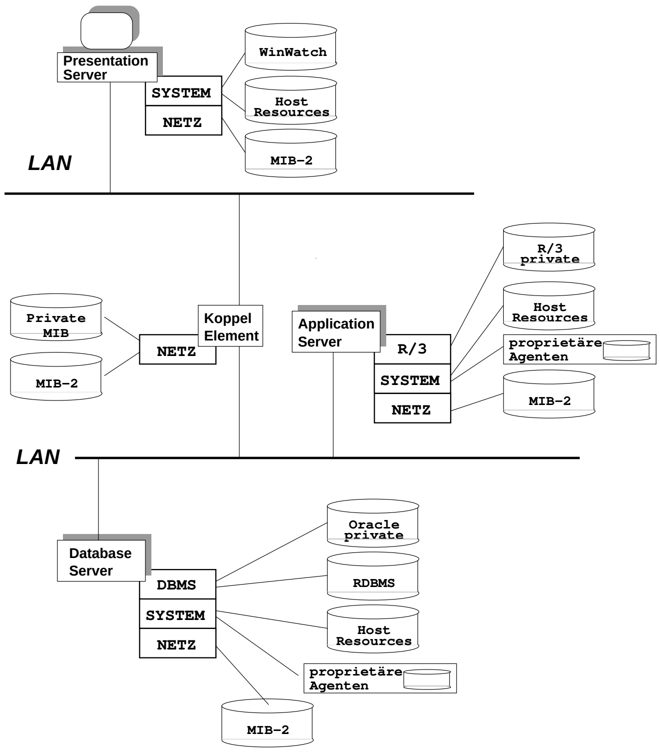

Abbildung 5.1: Kommunikationsschnittstellen im R/3-System

Die Systemmanagementanforderungen der Application-Server können nur zum Teil von der Host Resources MIB erbracht werden. Da die Überwachung von Systemressourcen aber eine der wichtigsten Aufgaben ist, muß ein Systemmanagement-Agent entsprechende Funktionalität aufweisen. Ein leider nicht standardkonformes Produkt, das diese Mächtigkeit bietet, ist z.B. der proprietäre Systemmanagement-Agent MaestroVision (siehe 2.3.1).

Für die Managementaspekte der R/3-Instanzen selbst ist eine Standardlösung wie die Network Services Monitoring MIB (siehe 2.2.3) keinesfalls ausreichend. Hier ist die Entwicklung einer spezifischen R/3-MIB erforderlich. Der Entwurf einer solchen MIB und Vorschläge zu ihrer Weiterentwicklung werden in den folgenden Abschnitten dargestellt.

Die zusätzlichen Managementaspekte des Datenbankservers können nur teilweise von der RDBMS-MIB (siehe 2.2.4) und herstellerspezifischen Erweiterungen wie der Oracle-MIB (siehe 2.2.5) abgedeckt werden. Die Definition dieser MIBs verlagert noch zu wenig Funktionalität in den Agenten und überläßt die Auswertung der gesammelten Daten fast gänzlich der Managementstation. Für eine sinnvolle Überwachung des Datenbankservers müssen diese MIBs insbesondere um aussagekräftige Trap-Definitionen erweitert werden.

Informationen zum Zustand der Kommunikationsverbindungen zwischen diesen

Komponenten liefert z.B. die Interfaces-Gruppe der MIB-2. Da die Standardinformationen der MIB-2 für das Management von Koppelelementen (Hubs, Bridges, …) meist nicht ausreicht, wird diese Information oft durch Private-MIBs des Komponentenherstellers ergänzt, die dann i.a. ein umfassendes Komponentenmanagement ermöglichen.

5.3 Die R/3-SNMP-Schnittstelle

Die SAP AG hat bereits vor einiger Zeit erkannt, daß die R/3-eigenen Managementwerkzeuge für das Management von großen R/3-Installationen nicht ausreichend sind, so daß eine Integration des R/3-Anwendungsmanagements in die Managementplattform des R/3-Anwenders notwendig ist. Um eine solche Integration in unterschiedliche Managementplattformen zu ermöglichen, wurde eine MIB entworfen, die der Managementplattform mit Hilfe von SNMP-Agenten zugänglich gemacht werden soll.

Nach dem derzeitigen Konzept sollen alle Systeme, die R/3-Komponenten beherbergen, mit SNMP-Agenten ausgestattet werden, die dann auch nur Daten zu den R/3-Komponenten dieses Systems liefern. Ein Austausch von Managementinformation über Systemgrenzen hinweg ist nicht vorgesehen. Daher kann eine R/3-MIB keine direkte Information zum Zustand des Gesamtsystems liefern.

Die SNMP-Schnittstelle soll im wesentlichen zur Meldung von Störungen des R/3-Systems an die Managementplattform dienen. Die eigentliche Fehleranalyse und Behebung sowie die Konfiguration des R/3-Systems soll dem CCMS vorbehalten bleiben. Daher wird auch nur lesender Zugriff auf diese Schnittstelle ermöglicht. Diese Einschränkungen sind vor allem wegen der bekannten Sicherheitsmängel des SNMP-Protokolls erforderlich.

Um einheitliche Managementanwendungen realisieren zu können, soll die R/3-SNMP-Schnittstelle architekturunabhängig gehalten werden, während das CCMS auf die Hardware-spezifischen R/3-Ressourcen eingeht.

5.3.1 Architektur der R/3-SNMP-Schnittstelle

Abbildung 5.2 zeigt die Architektur der R/3-SNMP-Schnittstelle. Um die in der MIB definierten Managementinformationen bereitstellen zu können, greift der SNMP-Agent über eine prozedurale Schnittstelle auf die von den R/3-Instanzen bereitgestellten Shared-Memory-Segmente zu.

Innerhalb einer R/3-Instanz ist der Application-Server für die Verwaltung dieses Speicherbereichs zuständig. Daraus folgt, daß zu R/3-Instanzen, die keinen Application-Server enthalten, keine Managementinformationen zur Verfügung stehen.

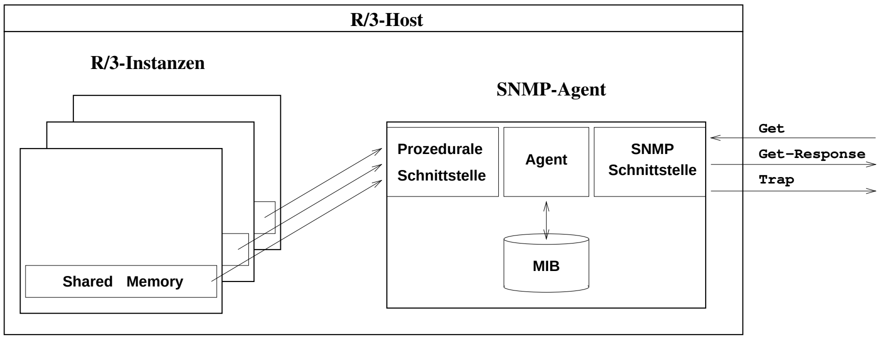

Abbildung 5.2: R/3-SNMP-Schnittstelle

Dies sollte jedoch kein Problem darstellen, da i.a. alle R/3-Instanzen Application-Server beinhalten. Da die Presentation-Server keine R/3-Instanzen darstellen, steht für sie keine Managementinformation in der R/3-MIB zur Verfügung.

SAP plant, die Zugriffsfunktionen auf den Shared-Memory-Bereich zur Verfügung zu stellen. Auf dieser Grundlage soll die Entwicklung von SNMP-Agenten für unterschiedliche Betriebssysteme durch die jeweiligen Systemanbieter erfolgen.

5.3.2 Entwurf einer R/3-MIB

In Zusammenarbeit mit verschiedenen Hardwareherstellern hat die SAP AG eine MIB für das System R/3 entworfen. In den 4 Gruppen r3system, r3config, r3alert und r3perf, die jeweils genau eine Tabelle enthalten, werden Konfigurations- und Leistungsdaten zu den R/3-Instanzen eines Host beschrieben. Alle Tabellen verwenden die r3systemNumber als ersten Index. Daher beziehen sich die Werte einer Tabellenzeile auf die durch diese Nummer identifizierte R/3-Instanz.

Weiterhin werden 28 Traps definiert, die über unterschiedlichste Störungen am R/3-System informieren. Dabei werden nicht nur R/3-interne Störungen berücksichtigt, sondern auch Probleme des Datenbanksystems gemeldet.

Der nachfolgend genau erläuterte Entwurf trägt die Versionsnummer 2.0 und ist in Abbildung 5.3 im Überblick dargestellt. Auf alle Objekte der MIB ist nur lesender Zugriff möglich.

5.3.2.1 Beschreibung der Tabellen

r3systemTable (Index: r3systemNumber)

Die r3systemTable enthält allgemeine Daten zum R/3-System und ermöglicht über den Namen des R/3-Systems (r3systemName) die Zuordnung der R/3-Instanzen zum zugehörigen R/3-System.

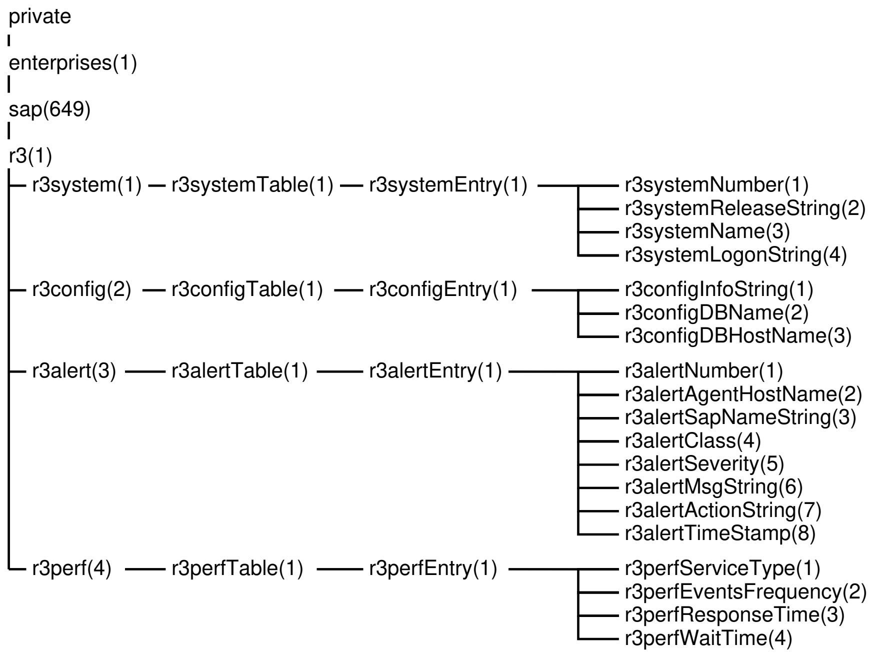

Abbildung 5.3: R/3-MIB Version 2.0

r3systemNumber Index über die R/3-Instanzen auf diesem Host.

r3systemReleaseString Version des R/3-Systems.

r3systemName Name des R/3-Systems, zu dem diese R/3-Instanz gehört.

r3systemLogonString UNIX-Kommando zum Start eines Presentation-Servers, der automatisch eine Verbindung zu dieser R/3-Instanz aufbaut.

r3configTable (Index: r3systemNumber)

Die r3configTable gibt Auskunft über die von den R/3-Instanzen angebotenen R/3-Basisdienste und über die zentrale Datenbank des zugehörigen Gesamtsystems.

r3configInfoString ASCII-String, der die von dieser R/3-Instanz angebotenen Dienste beschreibt. Jedem R/3-Dienst ist dazu ein Kennbuchstabe zugeordnet. Zusätzlich enthält der String die zweistellige Nummer der Instanz. Diese ist identisch mit der r3systemNumber

r3configDBName Name der zentralen Datenbank des R/3-Systems, zu dem diese Instanz gehört.

r3configDBHostName Name des Datenbankservers des R/3-Systems, zu dem diese Instanz gehört.

r3alertTable (Index: r3systemNumber, r3alertClass)

In der r3alertTable werden zu jeder R/3-Instanz auf diesem Host die Parameter des zuletzt verschickten Traps jedes Trap-Typs gespeichert. Für eine genaue Beschreibung des R/3-Trap-Konzepts siehe Abschnitt 5.3.2.2.

r3alertNumber Nummer des Traps.

r3alertAgentHostName Name des Hosts des Agenten, der den Trap absetzte.

r3alertSapNameString Name der R/3-Instanz. Dieser ist i.a. identisch mit dem r3configInfoString.

r3alertClass Typ des Traps, siehe Abschnitt 5.3.2.2.

r3alertSeverity Gewichtung des Traps, 3-stufige Einteilung in normal, warning und critical.

r3alertMsgString Textuelle Beschreibung der Trap-Ursache.

r3alertActionString Anweisung an einen Presentation-Server, die diesen veranlaßt einen R/3-View darzustellen, der sich zur weiteren Diagnose der Fehlerursache eignet. Diese Funktion wird vom Presentation-Server zur Zeit noch nicht unterstützt.

r3alertTimeStamp Zeitstempel.

r3perfTable (Index: r3systemNumber, r3perfServiceType)
Die r3perfTable enthält Leistungsdaten zu 4 verschiedenen R/3-Basisdiensten (Dialog, Update, Background, Spool) jeweils einer R/3-Instanz. Die Daten sind geglättete Durchschnittswerte der letzten 15 Minuten.

r3perfServiceType Art des Dienstes.

r3perfEventsFrequency Anzahl der Aufträge dieses Typs pro Minute.

r3perfResponseTime Bearbeitungszeit in Millisekunden.
Bearbeitungszeit=Wartezeit+Ausführzeit. Für den Dialog-Dienst ist dies die Antwortzeit, die der Benutzer bemerkt.

r3perfWaitTime Wartezeit auf freien Workprozess dieses Typs.

5.3.2.2 Beschreibung der Traps

Die R/3-MIB beschreibt 28 Traps, die fast alle[1] genau die Parameter enthalten, die auch die Spalten der zuvor beschriebenen r3alertTable bilden. In jedem Fall verfügt ein R/3-Trap über den r3alertSeverity-Parameter, der dem Trap die Bedeutung „Warnung“, „Kritisch“ oder aber „Normaler Zustand“ zuordnet. Traps im eigentlichen Sinn sind nur solche mit Severity „warning“ oder „critical“. Traps mit Severity „normal“ dienen nur dazu, den von vorangegangenen Traps gesetzten Zustand wieder zurückzusetzen. Der aktuelle Zustand jedes Trap-Typs kann

[1] Bei bestimmten Traps wären manche Parameter nicht sinnvoll.

zusätzlich aus der r3alertTable gepollt werden. Dieses Konzept, die Bedeutung eines Traps nicht wie sonst oft üblich statisch zu definieren, sondern in Form eines Parameters dynamisch festzulegen, bietet den Vorteil, daß Trap-Typen nicht mehrmals mit unterschiedlichem Gewicht definiert werden müssen. Ein weiterer, noch wichtigerer Vorteil ist, daß sog. Trap-directed Polling unterbleiben kann, da die Beendigung eines via Trap gemeldeten Ausnahmezustandes nicht durch Polling der Managementstation festgestellt werden muß, sondern durch einen Trap mit Severity „normal" vom Agenten gemeldet wird. Dies kann unter Umständen eine deutliche Netzentlastung bedeuten.

Im folgenden werden alle definierten Trap-Typen[2] kurz beschrieben. Falls dabei die Bedeutung von einzelnen Trap-Parametern erklärt wird, erscheinen deren Bezeichner *kursiv*.

- Veränderungen an der Konfiguration des R/3-Systems:

  trapStateChange Die von der R/3-Instanz *r3alertSapNameString* angebotenen R/3-Basisdienste haben sich verändert.

  trapInstanceStartUp Auf dem Rechner *r3alertAgentHostName* wurde die Instanz *r3alertSapNameString* gestartet.

  trapInstanceShutDown Die R/3-Instanz *r3alertSapNameString* wurde gestoppt.

- Hinweise auf die Systemlog-Datei:

  trapSyslogId In die Syslog-Datei wurde ein als kritisch konfigurierter Eintrag vorgenommen. Der *r3alertMsgString* enthält den Wortlaut des Eintrags.

  trapSyslogFreq Der konfigurierte Grenzwert für Einträge pro Stunde in die Syslog-Datei wurde überschritten.

- Probleme mit Systemressourcen:

  trapBufferSum Probleme mit der Größe oder Performance der R/3-Puffer.

  trapRollPage Die Größe der Roll- oder Page-Dateien liegt über einem definierten Grenzwert.

- Probleme mit der zentralen Datenbank:

  trapAbapSql Fehler beim Datenbank-Zugriff im ABAP/4-Interpreter.

[2]entspricht der r3alertClass in der r3alertTable

trapDbIndices Fehlende Indizes an einer relationalen Tabelle des Datenbanksystem. Dies führt zu stark erhöhten Zugriffszeiten auf die betroffene Tabelle.

trapDbFreeSpace Der freie Speicherplatz für das Datenbanksystem liegt unterhalb des definierten Grenzwertes.

trapDbArcStuck Das Datenbank-Archivierungssystem arbeitet nicht.

trapDbBackup Das letzte Datenbank-Backup liegt zu lange zurück.

- R/3-Interne Probleme:

trapEnqueue Problem des Enqueue-Service. Die Ursache dieses Problems sind vollständig gefüllte Sperr-Tabellen oder interne Fehler des Enqueue-Dienstes.

trapAbapUpd Update-Unterbrechung im ABAP/4-Interpreter.

trapAbapError Sonstiger Fehler im ABAP/4-Interpreter.

trapTrace Trace-Option aktiviert. Dies kann zu Performance-Problemen führen.

- Performance-Probleme:

trapPerfDia Die Performance des Dialog-Service der Instanz *r3alertSapNameString* liegt unterhalb des definierten Grenzwertes.

trapPerfUpd Die Performance des Update-Service der Instanz *r3alertSapNameString* liegt unterhalb des definierten Grenzwertes.

trapPerfBtc Die Performance des Background-Service der Instanz *r3alertSapNameString* liegt unterhalb des definierten Grenzwertes.

trapPerfSpo Die Performance des Spool-Service der Instanz *r3alertSapNameString* liegt unterhalb des definierten Grenzwertes.

- Von ABAP/4-Anwendungen generierte Traps:

trapSpool Spool-Problem.

trapArchive Archive-Problem.

trapGenPur3...8 In ABAP/4-Programmen frei verwendbare Traps; zur späteren Verwendung.

5.3.3 Bewertung des R/3-MIB-Entwurfs; Verbesserungsvorschläge

Auf den ersten Blick erscheint die R/3-MIB viel zu knapp für eine so komplexe Anwendung wie R/3. Tatsächlich enthält die MIB selbst im wesentlichen Konfigurationsdaten und nur wenig Leistungsdaten. Dieser scheinbare Mangel wird

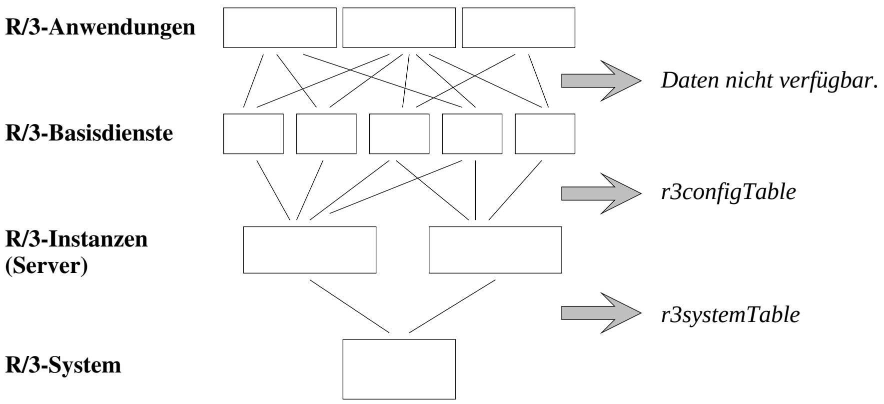

Abbildung 5.4: Systemkonfigurationsdaten der R/3-MIB

jedoch durch den umfangreichen Trap-Mechanismus ausgeglichen, der sehr differenzierte Zustandsinformation zum R/3-System liefert. Allgemein ist zu bemerken, daß die von der R/3-MIB zur Verfügung gestellte Information bereits stark verdichtet ist und keine irrelevante Detailinformation darstellt, wie sie sich sonst oft in MIBs findet. Abgesehen von einigen Verbesserungen im Detail wird deshalb auch nur ein Verbesserungsvorschlag gemacht, der nachfolgend beschrieben ist.

Betrachtet man die R/3-MIB an Hand des in Abschnitt 4.3 vorgestellten Modells, so erkennt man die in Abbildung 5.4 dargestellte Zuordnung von geforderten Systemkonfigurationsdaten zu MIB-Tabellen.

Wie bereits erwähnt, werden im R/3-Kernsystem keine Daten über die gerade laufenden R/3-Anwendungen verwaltet, so daß diese Information auch nicht über die MIB zur Verfügung gestellt werden kann. Die geforderte Zuordnung von R/3-Instanzen zu R/3-Systemen ist durch die r3systemTable gegeben. Informationen darüber, welche R/3-Instanzen welche Dienste anbieten, steht prizipiell über die r3configTable zur Verfügung, jedoch nur in codierter Form. Um diese Information in der Managementplattform besser verarbeiten zu können, wurde deshalb folgende Änderung angeregt:

Die im r3configInfoString enthaltene Information über die Dienste einer R/3-Instanz wird zusätzlich in der r3configServiceTable, einer neuen Tablelle der r3config-Group, abgelegt. Diese Tabelle besteht aus nur einer Spalte namens r3configServiceType, die als Aufzählungstyp über die möglichen R/3-Basisdienste definiert ist. Indiziert wird die Tabelle über die R/3-Instanzen (r3systemNumber) und den Dienst-Typ (r3configServiceType) selbst.

Der r3configInfoString soll dennoch als Name der R/3-Instanz erhalten bleiben, jedoch unter der Bezeichnung r3configInstanceName.

Alternativ könnte man auch auf diese zusätzliche Tabelle verzichten und statt dessen den r3configInfoString durch den r3configServiceType ersetzen. Dazu müßte

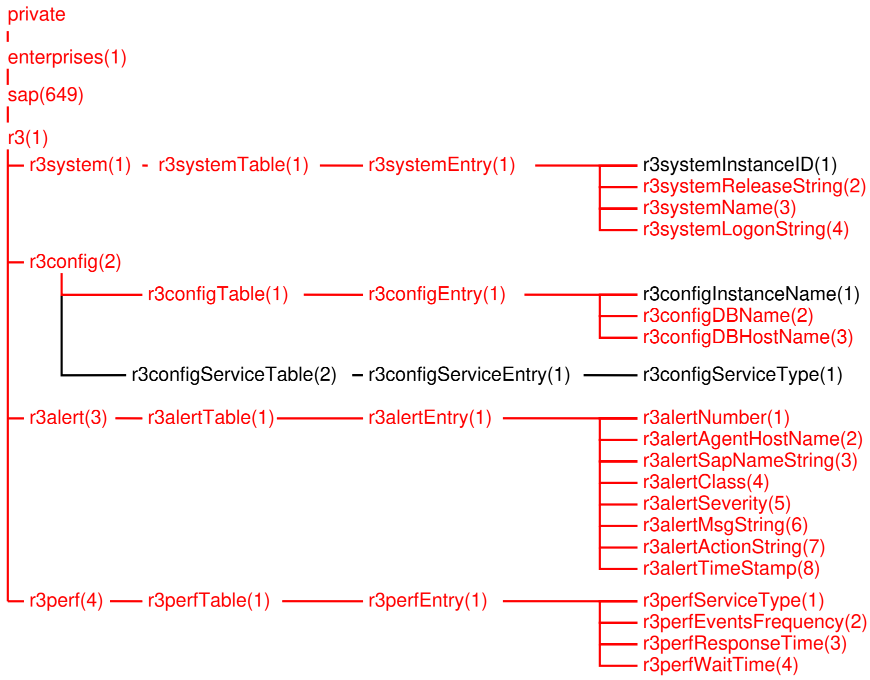

Abbildung 5.5: Änderungsvorschlag zur R/3-MIB

die r3configTable diesen Aufzählungstyp als zweiten Index erhalten. Dies würde aber zu großer Redundanz in der r3configTable führen, da dann der r3configDBName und der r3configDBHostName für eine Instanz bis zu 7 mal[3] gespeichert wäre. Beim Lesen der Tabelle mittels SNMP würde diese Information auch mehrfach übertragen.

Da sich die r3systemNumber auf R/3-Instanzen und nicht auf R/3-Systeme bezieht, wurde ferner angeregt, diese MIB-Variable in r3systemInstanceID umzubenennen.

Diese Vorschläge wurden in ASN.1-Notation [GORA 90, ISO 8824] formuliert und an SAP weitergeleitet. Sie wurden weitgehend übernommen und finden sich ab Version 2.4 der R/3-MIB, die im Anhang abgedruckt ist. Die neu entworfene r3configServiceTable ist dort ab Seite 106 enthalten.

5.4 Bewertung der Managementschnittstellen

Für eine Integration des R/3-Managements in eine Managementplattform ist in jedem Fall die SNMP-Kommunikationsschnittstelle zu bevorzugen, zum einen, weil sie über ein standardisiertes Protokoll angesprochen wird, das die meisten

[3]Ergibt sich aus der Anzahl verschiedener Basisdienste

Managementplattformen unterstützen, zum anderen, weil diese Schnittstelle gut dokumentiert und ihre Wartung durch SAP gesichert ist.

Dagegen besteht für weitere mögliche Managementschnittstellen wie Log-Dateien und Shared-Memory-Interfaces keine Stabilitätszusage seitens der SAP AG. Dies würde bei Versionswechseln immer wieder zu Problemen führen. Ferner ist zum Shared-Memory-Interface keine Dokumentation erhältlich. Auch aus Sicherheitserwägungen heraus ist ein direkter Zugriff auf das Shared-Memory-Interface nicht wünschenswert, da diese Schnittstelle Zugriff auf Managementfunktionen hat, die sonst nur ausgewählten Mitarbeitern über die R/3-Element-Management-Werkzeuge zugänglich sind.

Diese Schnittstellen scheiden folglich zur Akquisition von Managementinformation aus.

Kapitel 6

Szenarien des R/3-Anwendungsmanagements

In diesem Kapitel werden am Beispiel der BASF AG R/3-Einsatz- und Managementszenarien vorgestellt sowie Management-Aufgaben und Arbeitsabläufe erläutert. Ferner werden spezielle Anforderungen der BASF AG vorgestellt, die bei der Konzeption des R/3 Managementmoduls berücksichtigt werden müssen.

6.1 R/3-Einsatz bei der BASF AG

Im Rahmen des EURIS[1]-Projekts führt die BASF AG zur Zeit ein auf SAP R/3 basierendes Informationssystem für Marketing und Vertrieb ein. Dazu werden an den über ganz Europa verteilten Standorten der BASF jeweils ein oder mehrere R/3-Systeme installiert. Das Management der Anwendungen, Systeme und LANs erfolgt direkt am jeweiligen Standort. Dagegen wird das WAN der BASF, welches zur Kommunikation zwischen den Standorten dient, zentral von Ludwigshafen aus überwacht.

Am Standort Ludwigshafen befinden sich derzeit 7 R/3-Systeme im Aufbau, 3 bis 4 weitere sind geplant. Jedes dieser R/3-Systeme wird mit sog. UNIX-Rechenzentren bestehend aus ca. 7 UNIX-Servern realisiert. Als Server kommen Systeme verschiedener Hersteller (HP, SUN, IBM) und Leistungsklassen zum Einsatz. Wegen der besonders hohen Leistungsanforderungen an den Datenbankserver eines R/3-Systems wird hier z.Zt. der Einsatz von Multiprozessorsystemen der Firma Cray, die auf SUN-Sparc-Technologie basieren, getestet.

Diese Systeme werden durch Ethernet-Segmente zu Clustern zusammengeschlossen, die wiederum über Bridges mit dem FDDI-Backbone des Standortes verbun-

[1]Europäisches Informationssystem für Marketing und Vertrieb

den sind. Obwohl sich noch keines der R/3-Systeme im produktiven Betrieb befindet, ist abzusehen, daß die Ethernet-Technologie nicht ausreichend ist, um die zwischen dem Datenbankserver und den Application-Servern zu übertragenden Datenmengen zu bewältigen. Daher soll mittelfristig auch hier auf FDDI-Technik hochgerüstet werden.

Es ist geplant an jedem Application-Server ca. 150 Endgeräte als Clients zu betreiben. Damit ergeben sich ca. 1000 Benutzer pro R/3-System.

6.2 R/3-Management bei der BASF AG

6.2.1 R/3-Betriebskonzepte

Um den störungsfreien Betrieb der R/3-Systeme gewährleisten und dennoch Wartungs- und Entwicklungsarbeiten durchführen zu können, müssen neben den produktiven R/3-Systemen weitere R/3-Systeme zur Verfügung stehen. Im einzelnen sind dies:

R/3-Entwicklungssystem Zur Anpassung eines R/3-Systems an die speziellen Bedürfnisse eines Unternehmens sind i.a. Änderungen und Erweiterungen der R/3-Anwendungen erforderlich. Die BASF-AG führt diese Arbeiten größtenteils selbst durch. Dazu steht ein Entwicklungs- und Testsystem zur Verfügung, an dem die ABAP/4-Entwicklung durchgeführt wird. Diese Entwicklungsarbeit wird von SAP R/3 durch eine ABAP/4-Entwicklungsumgebung unterstützt, die auch von der SAP selbst zur Entwicklung der angebotenen Standardanwendungen verwendet wird.

R/3-Consolidierungssystem Das R/3-System erfährt zur Zeit noch recht häufig Releasewechsel, die nicht in Form von Patches eingespielt werden können, sondern eine Neuinstallation des Systems erforderlich machen. Um die Verfügbarkeit des Produktiv-Systems nicht zu gefährden, werden neue Releases zunächst auf einem Consolidierungssystem an die Umgebung des Unternehmens angepaßt und erst bei ausreichender Stabilität auf das Produktiv-System überspielt. Auch die von der BASF selbst entwickelten R/3-Erweiterungen werden hier einem Praxistest unterzogen.

6.2.2 Aufgabenbereiche und Managementwerkzeuge

Die im Zusammenhang mit dem R/3-Betrieb anfallenden Managementaufgaben werden bei der BASF AG von 3 verschiedenen Gruppen erbracht, deren Zuständigkeiten nachfolgend beschrieben sind. Dabei werden die jeweiligen Anforderungen an eine integrierte R/3-Managementanwendung erkannt.

6.2.2.1 Netzwerkbetrieb

Die Abteilung Netzwerkbetrieb ist für den Betrieb der LANs am Standort Ludwigshafen sowie für das BASF-WAN verantwortlich. Ferner ist diese Gruppe für das Systemmanagement verschiedener Server dieses Standorts zuständig. Dazu zählen z.B. Novell File-Server oder auch die Server der R/3-Systeme. Die Aufgaben dieser Gruppe beschränken sich also auf die Überwachung und Steuerung von Netzkomponenten und zentralen Rechnern. Dazu werden folgende Werkzeuge eingesetzt:

- SPECTRUM als Managementplattform
- MaestroVision für das Systemmanagement
- Spezielle, herstellerspezifische Werkzeuge für das WAN-Managemenet
- CommunicationManager als Netz-Dokumentationssystem
- Remedy ARS und CommunicationManager als Trouble-Ticket-Systeme

Ein zentrales Management der Endgeräte wie PCs und Workstations für einzelne Benutzer findet heute noch nicht statt. Der wesentliche Grund hierfür ist, daß wegen der Menge dieser Geräte (einige 10000) eine zentrale Überwachung durch Polling ausgeschlossen ist und Agenten, welche die Überwachung selbständig durchführen und nur im Fehlerfall Meldungen an das zentrale Management schicken, bisher nicht verfügbar sind. Hier eröffnen intelligente Agenten wie z.B. der WinWatch-Agent neue Möglichkeiten.

Aus dem Bereich des Anwendungsmanagements übernimmt diese Abteilung nur Überwachungsaufgaben. Die in Abschnitt 6.3 dargestellten Anforderungen der BASF an eine integrierte R/3-Managementlösung stammen daher im wesentlichen von dieser Gruppe. Bei Störungen an R/3-Systemen, die nicht auf Netz- oder Systemproblemen beruhen, wird zur weiteren Analyse und Behebung der SAP Basis Support informiert.

6.2.2.2 SAP Basis Support

Der SAP Basis Support ist für alle R/3-spezifischen Managementaufgaben verantwortlich. Dazu zählen insbesondere:

- Installation und Konfiguration der R/3-Systeme
- Anpassung der R/3-Anwendungen an die Anforderungen bei der BASF, wozu auch umfangreiche ABAP/4-Programmierung durchgeführt wird.

- Störungsbeseitigung und Tuning der R/3-Systeme

Bemerkenswert sind die beiden erstgenannten Aufgaben, da diese sonst auch oft von Experten der SAP durchgeführt werden.

Diese Gruppe verwendet hauptsächlich die in Abschnitt 5.1.1 vorgestellten Werkzeuge des CCMS. Der Early-Watch-Dienst wird nicht genutzt.

6.2.2.3 Support Center

Der Support Center realisiert einen User-Help-Desk für alle bei der BASF eingesetzten Anwendungen, der Störungsmeldungen der Benutzer entgegen nimmt und Fragen beantwortet. Auch diese Gruppe soll die integrierte R/3-Managementanwendung nutzen, um bei Störungen an R/3-Systemen besser Auskunft geben und detailliertere Fehlerreports an den SAP Basis Support oder die Netzbetriebsgruppe schicken zu können.

6.3 Anforderungen der BASF AG an eine R/3-Managementlösung

Da sich die für den produktiven Betrieb vorgesehenen R/3-Systeme bei der BASF erst im Aufbau bzw. Teststadium befinden, ist es schwierig schon heute top-down Anforderungen an eine R/3-Managementlösung zu formulieren. Aus den in den vorangegangenen Abschnitten beschriebenen Gegebenheiten bei der BASF und den Erfahrungen mit den Testsystemen ergeben sich jedoch folgende Mindestanforderungen:

- Integration der R/3-Managementlösung in die bestehende SPECTRUM-Managementumgebung. Besonders wünschenswert ist eine Integration in die Systemmanagement-Konzepte von MaestroVision.

- Durch die im vorangegangenen Kapitel vorgestellte R/3-MIB wurde eine gute Grundlage geschaffen diese Integration verwirklichen zu können. Die R/3-MIB erfüllt weitgehend die Anforderungen der Netzbetriebsgruppe und des Support Centers nach Überwachungsinformation und selbständig generierten Fehlermeldungen. Von der R/3-Managementanwendung wird eine geeignete Auswertung und Filterung der Fehlerinformation sowie eine Darstellung der MIB-Informationen erwartet. Wegen der komplexen Semantik der R/3-MIB ist hier ein MIB-Browser ungeeignet. Statt dessen ist eine inhaltliche Gliederung und graphische Aufbereitung der Information erforderlich.

- Da die R/3-MIB im wesentlichen für Überwachungsaufgaben ausgelegt ist, sind für eventuell notwendige Managementoperationen auf den R/3-Systemen weiterhin die proprietären R/3-Managementwerkzeuge notwendig. Für einen leichten Zugriff auf diese Werkzeuge soll es möglich sein, aus der neu zu entwerfenden R/3-Managementoberfläche R/3-Instanzen auszuwählen und eine Verbindung dorthin aufzubauen.

Kapitel 7

Konzeption des R/3-Managementmoduls

In diesem Kapitel werden zunächst die heute gegebenen Implementierungsmöglichkeiten betrachtet und davon ausgehend ein Konzept für ein R/3-Managementmodul entworfen. Dazu werden die Modellierungskonzepte, Architektur und Entwicklungswerkzeuge der Managementplattform SPECTRUM vorgestellt und bezüglich ihrer Tauglichkeit für das Anwendungsmanagement analysiert. Diese Plattform wurde gewählt, da sie über eine mächtige Modellierungstechnik verfügt und ihre Erweiterung durch zahlreiche Werkzeuge unterstützt wird.

Auf der Grundlage der erkannten Möglichkeiten wird ein R/3-Managementmodul entworfen und sein Einsatz am Beispiel der in Kapitel 6 vorgestellten Szenarien beschrieben.

7.1 Die Inductive Modeling Technology

Die Inductive Modeling Technology (IMT) bildet die konzeptionelle Grundlage der Managementplattform SPECTRUM. Da dieses Modellierungsverfahren bereits in zahlreichen Publikationen des MNM-Teams erläutert und diskutiert worden ist, werden hier nur die wichtigsten Begriffe knapp vorgestellt. Für eine umfassende Einführung wird auf [KELL 93] und [SPEC CON] verwiesen.

Die IMT ist eine Methode zur strukturierten Beschreibung von deklarativem und prozeduralem Wissen über die zu managenden Objekte sowie ihrer Beziehungen untereinander. Objekte werden durch sog. Modelltypen (engl. Model Types) beschrieben, die den Klassen des objektorientierten Ansatzes entsprechen. Modelltypen bestehen zum einen aus Attributen, welche die Beschaffenheit und den Zustand des Modelltypen beschreiben, und zum anderen aus Inference-Handlern (Methoden der Klasse), welche das Verhalten des Modelltypen auf Veränderungen seiner Umwelt, insbesondere seiner Attribute, bestimmen. Der objektorientierte Ansatz gestattet die Mehrfach-Vererbung von Attributen und Inference-Handlern innerhalb einer Modelltypen-Hierarchie. Durch systematische Verfeinerung und Zusammenfassung können so Modelltypen für spezielle Management-Objekte erstellt werden. Die Beziehungen zwischen Modelltypen werden durch Relationen (engl. Relations) beschrieben, die durch Regeln (engl. Rules) eingeschränkt und so der realen Welt angepaßt werden. Für Basisklassen definierte Regeln gelten auch für davon abgeleitete Modelltypen, d.h. Relationen und Regeln werden ebenfalls vererbt.

Die Instanzen eines Modelltyps werden als Modelle (engl. Model) bezeichnet, die Beziehungen zwischen Modellen heißen Assoziationen (engl. Associations).

7.2 Die Architektur der Managementplattform SPECTRUM

Abbildung 7.1 zeigt die wesentlichen Architekturelemente der Managementplattform SPECTRUM sowie einige wichtige Entwicklungswerkzeuge und Programmierschnittstellen, die in Abschnitt 7.3 näher beschrieben sind.

7.2.1 SpectroSERVER

Die zentrale SPECTRUM-Komponente ist der SpectroSERVER bestehend aus Virtual Network Maschine (VNM), Datenbank und Device Communication Manager (DCM).

In der Datenbank sind die Definitionen aller Modelltypen sowie das durch deren Instanzen beschriebene Netzmodell gespeichert. Dieses Modell wird zur Laufzeit in die VNM geladen.

Die SPECTRUM-Clients, wie z.B. SpectroGRAPH greifen über das SpectroSERVER API auf das Netzmodell zu, um einzelne Attribute der Objekte abzufragen oder aber um das Modell zu verändern. Anfragen an Attribute, die nicht statisch definiert sind oder von der VNM berechnet werden, reicht die VNM an den DCM weiter. Der DCM realisiert die Kommunikation mit den Agenten der einzelnen Netzelemente über verschiedene Protokolle. SNMP und ICMP werden standardmäßig unterstützt, weitere Protokolle können über das External Protocol Interface (EPI) durch Programmieren eines Proxy-Agenten (siehe 7.3.2: EPI) angebunden werden. Zu den Aufgaben des DCM gehört auch die Verarbeitung der ankommenden Mitteilungen, wie z.B SNMP-Traps und ihre Weiterleitung an die entsprechenden Modelle.

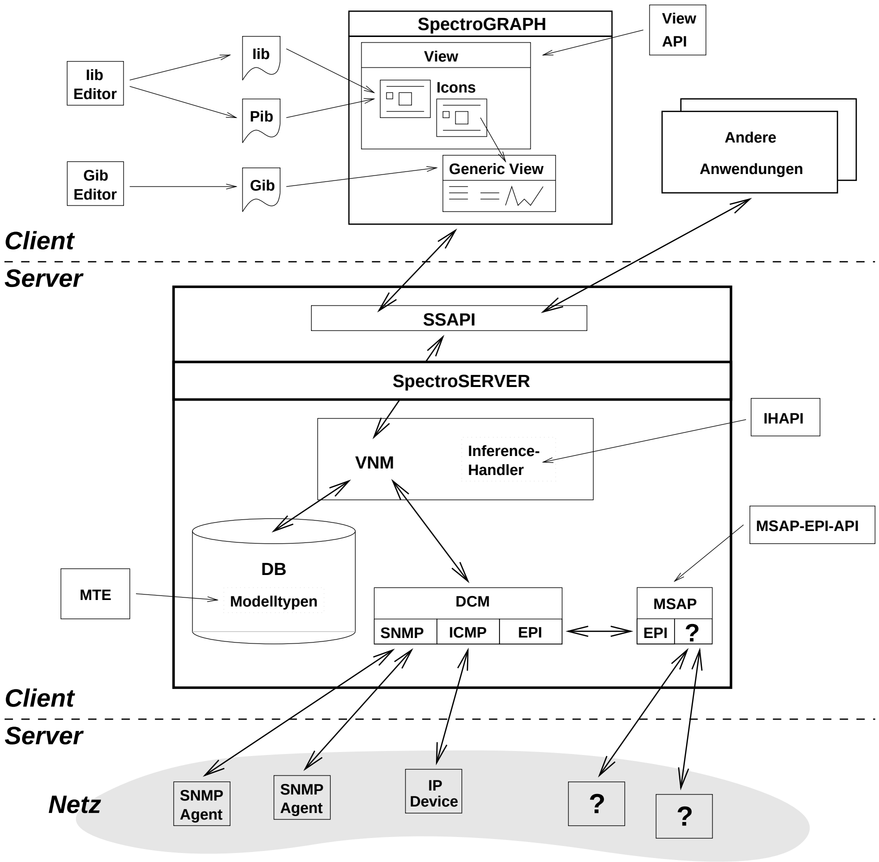

Abbildung 7.1: SPECTRUM : Architektur und Entwicklungswerkzeuge

Da die Auswertung der Traps des R/3-Agenten für des R/3-Management von großer Bedeutung ist, wird das SPECTRUM-Konzept zur Verarbeitung von Traps kurz vorgestellt. An Hand der IP-Adresse wird bestimmt auf welches Modell sich ein Trap bezieht. Mit Hilfe der Alert-Map-Datei zu diesem Modelltypen wird entschieden ob der Trap weiterverarbeitet werden soll. Wenn ja, wird ein sog. Event erzeugt, der zur weiteren Verarbeitung des Ereignisses innerhalb der VNM dient. Ein Event verfügt über eine eindeutige Nummer und enthält alle Parameter des Traps. Die Verarbeitung eines Events kann nun entweder durch Inference-Handler (siehe 7.3.2) oder aber über die Event-Disposition-Datei erfolgen. Diese Datei ermöglicht die Abbildung von Events auf Alarme, welche den Zustand eines Modells beeinflussen und ggf. dem Benutzer gemeldet werden.

7.2.2 SpectroGRAPH

Zur Visualisierung des vom SpectroSERVER bereitgestellten Netzmodells realisiert der SpectroGRAPH nach unterschiedlichen Anforderungen gegliederte Sichten auf die einzelnen Objekte.

Es existieren Sichten, die ein Unternehmensnetz nach topologischen, geographischen und organisatorischen Gesichtspunkten gliedern, ferner eine Sicht für das Systemmanagement, die bereits in Abschnitt 2.3.1 (MaestroVision) vorgestellt wurde.

Diese Sichten sind alle nach dem gleichen Prinzip aufgebaut. Für jede Sicht werden Strukturierungselemente in Form von Modelltypen erstellt, die jeweils eine bestimmte Klasse von Modellen enthalten können. So werden z.B. in der topologischen Sicht die an ein Ethernet-LAN angeschlossenen Geräte (z.B. PCs, Hub, Workstations) in einem Modell des Typs LAN_802_3 zusammengefaßt und stehen so einer übergeordneten Sicht als abstrakte Einheit zur Verfügung. Die Beziehung zwischen Strukturierungsmodelltypen und untergeordneten Modellen wird durch Instanzen einer Relation (Assoziationen) ausgedrückt. Die für diese Relation definierten Regeln bestimmen, welche Modelltypen ein Strukturierungsmodell enthalten darf. Durch weitere Modelltypen, die auch Strukturierungsmodelle enthalten dürfen, kann ein beliebiges Abstraktionsniveau erreicht werden. In der topologischen Sicht werden z.B. die verschiedenen LAN-Modelle zusammen mit Routern und WAN-Strecken zu einem WAN-Modell abstrahiert. Diese hierarchische Gliederung wird im SpectroGRAPH durch entsprechende View-Hierarchien dargestellt, in denen die Modelle jeweils einer Abstraktionsebene durch Icons dargestellt sind. Durch Mausklick auf dieses Icon wird ein View geöffnet, der die durch Assoziationen untergeordneten Modelle darstellt.

Diese Assoziationen erlauben auch jedem abstrakten Modell festzustellen, welche Modelle ihm zugeordnet sind. Dies wird in fast allen Hierarchien genutzt, um aus dem Zustand der untergeordnenten Modelle einen Gesamtzustand für das abstrakte Modell abzuleiten. Dieser durch Inference-Handler der jeweiligen Modelle realisierte Mechanismus wird als „Condition-Rollup“ bezeichnet und ist in [SPEC ADG] und [SPEC KBG] detailliert beschrieben.

Folgende View-Hierarchien sind bereits implementiert:

Topology Die Topology-Hierarchie ermöglicht die logische Gliederung eines Netzes in Subnetze bis hinunter zu einzelnen Geräten. Zur Strukturierung exisitieren Modelltypen wie LAN, WAN und Network. Die Beziehungen zwischen Modellen werden durch die „collects“-Relation beschrieben.

Location Die Location-Hierarchie gliedert das Unternehmensnetz nach geographischen Geschichtspunkten. Ausgehend von einer Landkarte wird das Netz über Standorte, Gebäude und Stockwerke bis hin zu Räumen und Schränken gegliedert, die einzelne Netzkomponenten enthalten. Strukturierung und Condition-Rollup-Funktion dieser Views basieren auf der „contains“-Relation.

Org-Chart Die Org-Chart-Hierarchie schließlich gliedert ein Unternehmen in organisatorische Einheiten, wie z.B. Unternehmensgruppen, Unternehmensbereiche, Abteilungen und Arbeitsgruppen. Diese Hierarchie wird durch die „organizes“-Relation aufgebaut. Um Besitzverhältnisse zu repräsentieren, kann diese Hierarchie ferner Modelle des Typs Org_Owns beinhalten. Solche Modelle ermöglichen die Zuordnung von Modellen aus den Location und Topology-Hierarchien (z.B. Geräte oder Gebäude) über die „owns“-Relation. Die Objekte dieser Hierachie verfügen noch nicht über die Condition-Rollup Funktionalität.

Modelle werden in den genannten Views durch verschiedene Icons dargestellt, deren Aussehen und Funktion in Textdateien (Icon Information Block, Iib) beschrieben ist. Es ist möglich, Modelle desselben Modelltypen in verschiedenen Views unterschiedlich darzustellen. Welches Icon jeweils verwendet werden soll, ist in der Pib-Datei (Perspective Information Block) des Views definiert.

Zur Darstellung der Attribute eines Modelltypen können zusätzlich sog. Generic Views definiert werden, die über einen Menüeintrag des Icons erreicht werden. Die Beschreibung solcher Views erfolgt über Gib-Dateien (Generic Information Block).

Daneben existieren einige interne Views des SpectroGRAPH, die ebenfalls über das Iconmenü angesprochen werden. Besonders wichtige Views dieser Kategorie sind:

Events View Dieser View enthält eine Liste aller Events des Modells.

Alarm View In diesem View werden die aktuellen Alarme des Modells zusammen mit einer Fehlerbeschreibung und Ratschlägen zu ihrer Behebung dargestellt.

Application View Die von einem Gerät unterstützten MIB-Gruppen sind in diesem View dargestellt. Dieser View wird in Abschnitt 7.4 ausführlich beschrieben.

MIB-Walk View Der MIB-Walk View realisiert einen einfachen Browser zum Anzeigen der Attribute eines Modelltypen und der für die Instanz aktuell gültigen Attributwerte.

7.3 SPECTRUM-Entwicklungswerkzeuge

7.3.1 Level 1 Entwicklungswerkzeuge

Mit „Level 1“ werden alle SPECTRUM-Entwicklungswerkzeuge bezeichnet, die es erlauben die Funktionalität der Managementplattform SPECTRUM zu erweitern, ohne neuen Programmcode erstellen zu müssen. Die Werkzeuge des Level 1 Toolkits und ihre Möglichkeiten sind nachfolgend beschrieben:

MTE Das Erstellen neuer Modelltypen und das Einfügen in die bestehende Modelltypen-Hierarchie erfolgt mit Hilfe eines Werkzeugs mit graphischer Benutzeroberfläche, dem Model Type Editor (MTE) [SPEC MTE]. Relationen und Regeln werden ebenfalls mit diesem Editor erstellt. Attribute können mit Hilfe des MTE entweder von Hand hinzugefügt werden oder aber durch Übersetzen einer RFC 1155-konformen MIB [RFC 1155] mit Hilfe des integrierten MIB-Compilers erstellt werden.

GIB-Editor Die Präsentation der Attribute eines Modelltypen kann entweder über den MIB-Walk View erfolgen, oder aber über sog. Generic Views. Generic Views sind insbesondere immer dann erforderlich, wenn entweder Tabellen dargestellt oder schreibender Zugriff ermöglicht werden soll.

Der Aufbau eines Views dieses Typs wird durch Textdateien, sog. Generic Information Blocks (GIB), beschrieben. Diese können zum einen mittels gewöhnlicher Texteditoren, zum anderen mit Hilfe des in die Oberfläche des SpectroGRAPH integrierten GIB-Editors [SPEC GIB] erstellt werden. Dieser gestattet es, mit Hilfe eines MIB-Browsers Attribute eines Modelltypen auszuwählen, und bietet anschließend eine Liste der möglichen Darstellungsformen für den Datentyp dieses Attributs an. Die gewählte Darstellungsform kann dann i.a. noch weiter parametrisiert werden, z.B. um die Zugriffsart (lesend / lesend und schreibend) auf das Attribut festzulegen oder um Aufzählungstypen mit entsprechendem Text zu versehen. Die Auswahl an Darstellungsformen ist äußerst vielfältig (von einfachen Feldern über Graphen bis hin zu Tortendiagrammen und Graphen, die mehrere Attribute zusammen darstellen können) und lassen kaum Wünsche offen. Durch zusätzliche Text und Graphikelemente, deren Erstellung der GIB-Editor ebenfalls unterstützt, kann die Präsentation weiter verbessert werden.

Dagegen wird die Darstellung von MIB-Tabellen bisher nur schlecht unterstützt. Die Beschreibungsdateien für Tabellen müssen von Hand erstellt werden, ferner sind die Präsentationsmöglichkeiten stark eingeschränkt. Durch Auswahl einzelner Tabellenzeilen ergibt sich jedoch die Möglichkeit, die Instanzen der Attribute dieser Zeile in einem neuen View mit den dann wieder verfügbaren, umfangreichen Darstellungsformen zu präsentieren. Die Anwendung dieses Verfahrens wird zusammen mit den neuen Views des R/3-Management-Prototypen beschrieben.

Weiterhin bietet der GIB-Editor die Möglichkeit Buttons zu erstellen, die weitere Views öffnen oder externe Programme starten. Damit läßt sich beispielsweise eine View-Folge aufbauen, um die MIB eines Modelltypen logisch zu strukturieren.

IIB-Editor Dieser Editor ist wie der MTE als eigenständiges Werkzeug mit graphischer Oberfläche realisiert und unterstützt die Erstellung der bereits erwähnten Beschreibungsdateien für Icons. Dazu können mit diesem Editor bereits bestehende Icons interaktiv verändert oder aus den vorhandenen Grundelementen neu erstellt werden. Im wesentlichen werden für jedes Icon verschiedene Zonen definiert, die auf Doppelklick mit der linken Maustaste mit dem Öffnen bestimmter Views reagieren. Diese und weitere Funktionen werden auch über die Definition eines Icon-Subviews-Menü zugänglich gemacht, das durch die mittlere Maustaste geöffnet wird.

Weiterhin wird festgelegt, wie der aktuelle Wert des Zustandsattributs des Modells auf die Farbe des Icons abgebildet werden soll.

Extensions Integration Toolkit Dieses aus einigen Shellskripten bestehende Werkzeug faßt alle zur Integration einer Komponente vorgenommenen Spectrum-Erweiterungen (Level 1 und 2) zu einer Datei zusammen, die sich zur Installation in anderen SPECTRUM-Umgebungen eignet. Ein solches Installationspaket wird als Managementmodul bezeichnet.

7.3.2 Level 2 Entwicklungswerkzeuge

Dieser Abschnitt gibt einen kurzen Überblick über die Programmierschnittstellen, die es erlauben SPECTRUM in verschiedener Hinsicht zu erweitern. Jede Level 2 Programmierung erfolgt in C++ [STRO 91].

Inference Handler API Das Erstellen neuer Infernce-Handler zu einem Modelltypen erfordert die Verwendung des Inference-Handler Application Programming Interface (IHAPI). Ein Inference-Handler wird genau einmal pro Modelltyp instanziiert und kann dann von allen Modellen dieses Typs benutzt werden. Der Aufruf eines Inference-Handlers erfolgt immer als Reaktion auf eine Veränderung in der VNM. Folgende Ereignisse können zum triggern von Inference-Handlern verwendet werden:

- Erzeugung und Löschung von Modellen
- Veränderungen an Attributwerten
- Veränderungen an Relationen, d.h. Entstehen oder Verschwinden von Assoziationen
- Events

Als Reaktion auf solche Ereignisse kann ein Inference-Handler zum Beispiel:

- Attributwerte verändern
- Assoziationen erzeugen oder löschen

- Modelle erzeugen oder löschen
- Alarme generieren

Eine umfassende Einführung in die Inference-Handler-Programmierung geben [KELL 93] und [SPEC INF].

SpectroSERVER API Das SpectroSERVER API unterstützt die Entwicklung neuer Anwendungen, die mit dem SpectroSERVER kommunizieren sollen. Es wird sowohl synchrone als auch asynchrone Kommunikation unterstützt [SPEC APIa, SPEC APIb]. Die Protokolldateneinheiten dieses Protokolls werden als VnmParmBlocks bezeichnet, deren Erzeugung durch das VnmParmBlock API [SPEC PAR] unterstützt wird.

View API Mit Hilfe des View API können zusammen mit dem SpectroSERVER API neue, hierarchische Sichten im SpectroGRAPH realisiert werden [SPEC VIE].

External Protocol Interface Das EPI erlaubt die Erweiterung des DCM um zusätzliche Protokolle. Dazu wird mit Hilfe des MSAP-EPI API[1] ein Proxy-Agent erstellt, der das neu zu integrierende Protokoll in das EPI-Format umsetzt [SPEC EPI].

Allgemeine Methoden, die zur Programmierung der beschriebenen Schnittstellen verwendet werden können, stellen die Global Classes [SPEC GCR] bereit.

7.4 Generische Managementkonzepte in SPECTRUM

Die Managementplattform SPECTRUM wird seit einigen Jahren kontinuierlich um Methoden erweitert, welche die Integration von neuen Netzkomponenten erheblich erleichtern bzw. eine umfassendere Integration ohne Programmieraufwand ermöglichen. Die wesentlichen Neuentwicklungen in dieser Richtung sind Methoden zur Integration allgemeiner SNMP-fähiger Geräte[2] und spezielle Methoden zur Integration modular aufgebauter Sternkoppler (Hubs). Da zur Integration des SAP R/3 SNMP-Agenten die allgemeine Methode in Frage kommt, soll diese im folgenden vorgestellt werden.

Wesentliches Konzept dieser als „Generic SNMP Device" bezeichneten Methode ist die Aufteilung der Internet-MIB in einzelne Teilbäume, sog. Applikationen. Diese Applikationen werden als Modelltypen realisiert. Die Zuordnung von

[1] Management Station Access Provider - External Protocol Interface - Application Programming Interface

[2] d.h. Netzkomponenten und Endgeräte, die über SNMP-Agenten verfügen.

Instanzen dieser Applikations-Modelltypen zu Gerätemodellen erfolgt erst zur Laufzeit. Dazu wird von der Gerätemodell-Instanz ein ausgezeichnetes Attribut (das sog. Default-Attribut) gepollt und an Hand der Verfügbarkeit dieses Attributs entschieden, ob das Gerät diesen MIB-Teilbaum unterstützt. Falls das Default-Attribut vorhanden ist, wird eine Instanz der jeweiligen Applikation erzeugt und mit Hilfe einer „manages“-Assoziation der Gerätemodell-Instanz zugeordnet. Um einen MIB-Teilbaum weiter untergliedern zu können, wird zwischen Major- und Minor-Applications unterschieden. Minor Applications gliedern einen MIB-Teilbaum in einzelne Gruppen und werden über die „provides“-Relation mit der zugehörigen Major-Application verbunden. Die Existenz einzelner Gruppen wird wieder mittels Default-Attributen überprüft. Minor Applications dienen also zur Unterstützung von MIBs mit optionalen Gruppen, die nicht von jeder Agentenimplementierung unterstützt werden. Die Darstellung der Applikationsmodelle und ihrer Beziehungen erfolgt über den Application-View (siehe Abbildung 7.2).

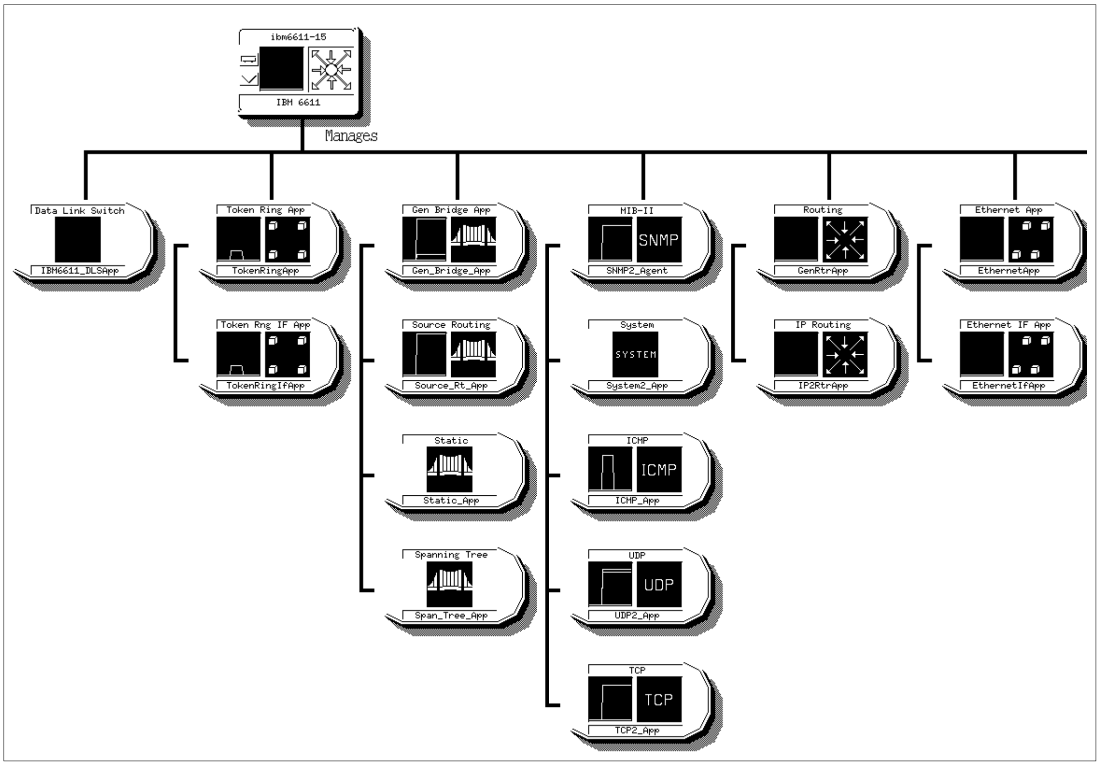

Abbildung 7.2: Application View eines Router-Modells

In diesem View repräsentiert das oberste Icon das Gerätemodell; die Icons der ersten Zeile repräsentieren Major-Applications, die darunterliegenden Icons Minor-Applications. In Abbildung 7.2 erkennt man beispielsweise in der vierten Spalte die von diesem Router (siehe auch [DEMA 94]) unterstützten Teilbäume der MIB-II, nämlich system, icmp, tcp und udp [RFC 1213]. Dabei wurde der gesamte MIB-II-Teilbaum als Major-Application modelliert, die einzelnen Gruppen als Minor-Applications.

Für die meisten standardisierten Internet-MIBs hat Cabletron bereits Applikationen erstellt. Applikationen für herstellerspezifische MIBs können mit Hilfe der Modelltypen des „Generic SNMP Device“ (siehe Abbildung 7.3) ohne Programmieraufwand erstellt werden.

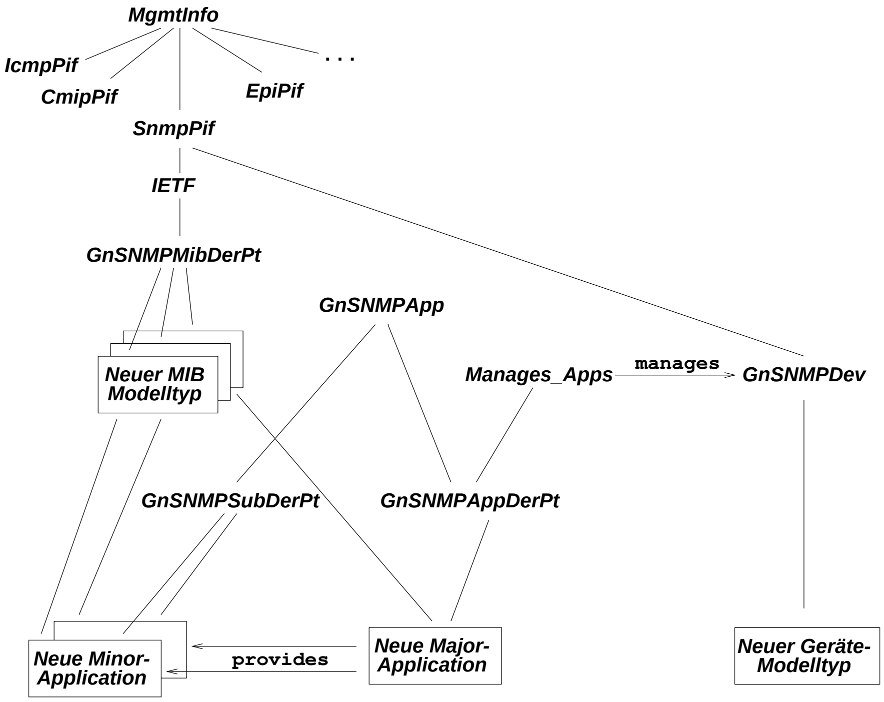

Abbildung 7.3: Erstellen neuer SNMP-Applikationen

Dazu werden zunächst vom GnSNMPMibDerPt ein oder, bei der Verwendung von Minor Applications, auch mehrere Modelltypen abgeleitet, in die mit Hilfe des MIB-Compilers des MTE die neu aufzunehmende MIB eingelesen wird. Der Modelltyp GnSNMPMibDerPt stellt die Funktionalität zum Zugriff auf die Managementinformation bereit. Dazu ist er als Unterklasse des SnmpPif-Modelltypen (SNMP Protocol Interface) realisiert. Ähnliche Ableitungspunkte existieren auch für andere Standard-Protokolle wie z.B. ICMP, sowie für proprietäre Protokolle.

Die Management-Funktionalität zum Erkennen von Applikationen und zur Erzeugung der benötigten Assoziationen wird vom GnSNMPAppDerPt ererbt, für den über den Basis-Modelltyp Manages-Apps die „manages“-Relation zum Gerätemodell-Ableitungspunkt GnSNMPDev definiert ist. Sollen zusätzlich Minor Applications verwendet werden, so werden diese vom GnSNMPSubDerPt abgeleitet und die „provides“-Relation um entsprechende Regeln erweitert, die diese Minor Applications ihrer Major Application zuordnen. Allen neuen Applikations-

Modelltypen muß ihr jeweiliger MIB-Anteil als Basismodelltyp hinzugefügt werden. Ferner müssen für alle Applikationen Default-Attribute festgelegt werden.

Diese Modellierung reicht bereits aus, um die neue Applikation im Application-View darzustellen und mit Hilfe eines MIB-Browsers auf die Daten des Agenten lesend zugreifen zu können. Um die Information übersichtlich strukturiert präsentieren zu können und auch schreibenden Zugriff zu ermöglichen, kann der Entwickler mit Hilfe des GIB-Editors eigene Views entwerfen und in das Applikations-Icon integrieren.

Um das Gerät, welches die neuen Applikationen bereitstellt, nicht nur als „Generic SNMP Device“ darstellen zu können, sondern z.B ein Workstation-Icon zu verwenden, kann vom Modelltyp GnSNMPDev ein neuer Modelltyp abgeleitet werden, dessen Instanzen dann durch ein frei definierbares Icon dargestellt werden.

7.5 Analyse der Vor- und Nachteile des Einsatzes von MaestroVision

Die in Kapitel 1, Abschnitt 2.3.1 vorgestellte SPECTRUM-Erweiterung für das Systemmanagement *MaestroVision* wird bei zahlreichen SPECTRUM-Anwendern, unter anderen bei der BASF, eingesetzt. Da MaestroVision die für das R/3-Management geforderten Systemmanagementfunktionen gut abdeckt, erscheint es sinnvoll zu versuchen, die durch das R/3-Managementmodul eröffneten Möglichkeiten zum R/3-Anwendungsmanagement in die Funktionalität von MaestroVision zu integrieren.

Wie in [KELL 93] und [MV TECH] beschrieben, realisiert MaestroVision zunächst Modelltypen für einzelne Workstation-Komponenten wie z.B. Prozessor, Filesystem, Hauptspeicher, ..., die dann zu Modelltypen für Rechner bestimmter Hersteller zusammengefaßt werden.

Eine einfache und elegante Möglichkeit, diese Integration zu realisieren, besteht daher in der Erstellung eines neuen Modelltypen mit dem GnSNMPDev-Modelltyp und einem MaestroVision-Workstation-Modelltypen als Basisklassen. Dieser neue Modelltyp verfügt über alle Attribute und Inference-Handler der Basisklassen, also insbesondere über die Funktionalität zur Erzeugung des Application Views und der Systemmanagement-Views von MaestroVision. Zu diesem Modelltyp läßt sich mit Hilfe des IIB-Editors ein Icon realisieren, das Zugriff auf all diese Views erlaubt.

Der Einsatz von MaestroVision als Implementierungsgrundlage für die Systemmanagementaspekte bringt neben den beschriebenen Vorteilen auch einige Probleme mit sich, die darin begründet sind, daß der MaestroVision-Client nur als Managementanwendung für den MaestroVision-Agenten entworfen wurde, jedoch nicht als Schnittstelle für Weiterentwicklungen ausgelegt ist. Zum Beispiel wird die Erweiterung von MaestroVision-Modellen dadurch erschwert, daß zu den Attributen und Inference-Handlern der MaestroVision-Modelltypen keine Dokumentation zur Verfügung steht und viele Modelltypen auch im Modeltype-Editor nicht sichtbar sind. Einige weitere Probleme wurden erst bei der Implementierung des Prototypen offenbar und sind in Abschnitt 8.3.1 beschrieben.

7.6 Konzeption eines SAP R/3 Managementmoduls

Auf der Grundlage der in Kapitel 4 erarbeiteten Anforderungen an eine R/3-Managementanwendung wurde unter Berücksichtigung der in Abschnitt 6.3 vorgestellten Betreiberanforderungen ein Managementmodul konzipiert, daß die Managementplattform SPECTRUM um die geforderte Funktionalität erweitert.

Die Implementierung dieses Moduls erfordert die Verwendung der meisten der zuvor vorgestellten SPECTRUM-Entwicklungswerkzeuge. Abschnitt 7.6.1 erläutert zunächst, wie die Modellierung mit SPECTRUM-Methoden erfolgen soll, in Abschnitt 7.6.3 werden Konzepte zur geeigneten Präsentation der modellierten Objekte vorgestellt und ihre Realisierung beschrieben.

7.6.1 Modellierung

Zunächst müssen alle relevanten realen und logischen Objekte eines R/3-Systems in SPECTRUM modelliert werden. Dazu werden folgende Modelltypen erstellt und gemäß Abbildung 7.4 in die bestehende Modelltypenhierarchie eingefügt.

WS_XXX_GnApp Diese Modelltypen repräsentieren die Systeme eines R/3-Systems. Sie werden jeweils von einem Workstation-Modell und dem Modelltyp GnSNMPDev (Generic SNMP Device) abgeleitet, wobei die drei X als Platzhalter für den jeweiligen Rechnertyp stehen. Für den Prototypen wurde lediglich der Modelltyp WS_SUN realisiert. Alle übrigen Workstation-Modelltypen können nach demselben Verfahren erweitert werden. Die naheliegende Alternative, den Modelltyp GnSNMPDev einfach als Basisklasse der bestehenden Workstation-Modelltypen hinzuzufügen ist leider nicht durchführbar, da es der MTE nicht zuläßt, Veränderungen an Modelltypen vorzunehmen, die von anderen Entwicklern erstellt wurden.

Diese Modelltypen erben also die gesamte Funktionalität des GnSNMPDev und des jeweiligen MaestroVision-Workstation-Modells. Damit stehen insbesondere der Application-View mit den Funktionen zur Generierung von

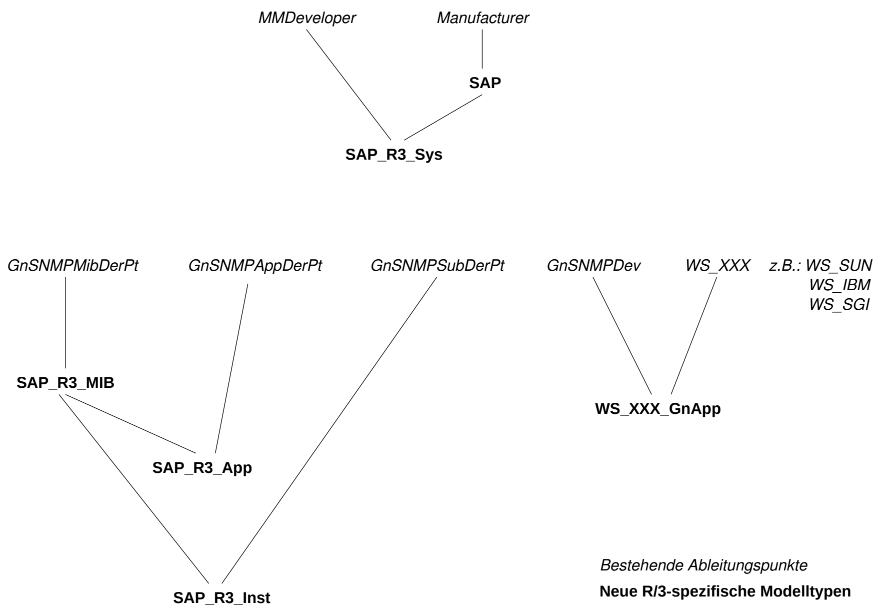

Abbildung 7.4: Modelltypenhierarchie mit R/3-Modelltypen

Applikationsmodellen und die Systemmanagement-Sichten von MaestroVision in einem Modell zur Verfügung.

SAP_R3_MIB In diesen Modelltyp wird die R/3-MIB mittels des MIB-Compilers des MTE eingelesen. Dieser Modelltyp dient nur als Ableitungspunkt für die R/3-Applikationsmodelle (SAP_R3_App, SAP_R3_Inst) und stellt diesen die Attribute der R/3-MIB zur Verfügung. Daher ist dieser Modelltyp nicht instanziierbar.

SAP_R3_App Dieser Modelltyp bildet eine neue Major Application und erhält neben dem GnSNMPAppDerPt den Modelltyp SAP_R3_MIB als Basisklasse. Instanzen dieses Modells repräsentieren R/3-Agenten. Durch die Inference Handler des GnSNMPDev wird dieses Modell automatisch angelegt, sobald auf einem als WS_XXX_GnApp modellierten Rechner ein R/3-Agent gestartet wird. Die Instanz dieses Modell wird dabei über eine „manages“-Assoziation mit dem Workstationmodell verbunden.

Für diesen Modelltyp wird ein Inference-Handler erstellt, der für jede in der r3systemTable verzeichnete R/3-Instanz ein Modell des Typs SAP_R3_Inst anlegt und über eine „provides“-Assoziation mit dem SAP_R3_App-Modell verbindet. Dieser Inference-Handler wird durch die Entstehung des SAP_R3_App-Modells und durch Traps der Typen r3InstanceStartUp und r3InstanceShutDown getriggert Bei r3InstanceShutDown-Traps wird das entsprechende SAP_R3_Inst-Modell wieder gelöscht.

SAP_R3_Inst Der Modelltyp SAP_R3_Inst repräsentiert R/3-Instanzen. Modelle dieses Typs werden vom Modell des R/3-Agenten (SAP_R3_App) angelegt. Dieser Modelltyp dient dem Zugriff auf die in der R/3-MIB zu einer Instanz abgelegten Informationen und ist daher von dem Modelltyp für die R/3-MIB, SAP_R3_MIB, abgeleitet. Um Modelle dieses Typs im Application-View darstellen zu können, ist der Modelltyp SAP_R3_Inst als Minor-Application realisiert, also zusätzlich von GnSNMPSubDerPt abgeleitet.

Diesem Modelltyp wird ein Inference-Handler zugeordnet, der bei der Entstehung eines SAP_R3_Inst-Modells an Hand des r3systemName prüft, ob schon ein Modell für ein R/3-System (SAP_R3_Sys) dieses Namens existiert. Wenn dies der Fall ist, wird das Modell über die „consists_of"-Relation mit diesem R/3-Systemmodell verbunden. Die „consists_of"-Relation existiert in der SPECTRUM-Datenbank bisher nicht und muß mit dem MTE neu angelegt werden.

Für diesen Modelltyp wird außerdem das in Abschnitt 7.6.2 beschriebene Verfahren zur Auswertung der R/3-Traps realisiert. Dadurch wird der Zustandswert für diese R/3-Instanz bestimmt.

SAP_R3_Sys Die Modellierung eines R/3-Systems als logische Einheit wird mit diesem Modelltypen verwirklicht. Über die „consists_of"-Relation sind Modelle dieses Typs mit den Modellen der R/3-Instanzen verbunden, aus denen das modellierte R/3-System besteht. Da die R/3-Instanzen in Form von Traps ausreichende Überwachungsmöglichkeiten für den Datenbankserver bieten, wird dieser hier nicht explizit modelliert.

In der SPECTRUM-Modelltypenhierarchie finden sich bisher keine geeigneten Ableitungspunkte für Modelle von verteilten Anwendungen. Daher wird dieser Modelltyp nur von den Modelltypen MMDeveloper und Manufacturer abgeleitet. Von MMDeveloper werden Attribute ererbt, in die Informationen über den Hersteller des Managementmoduls und die Modulversion abgelegt werden. Die Ableitung von Manufacturer erfolgt nicht direkt, sondern über den neuen Modelltyp für Produkte der Firma SAP.

Die Aufgaben des Modelltypen SAP_R3_Sys sind die Berechnung eines Gesamtzustandes für das R/3-System aus dem Zustand der über die „consists_of"-Relation untergeordneten R/3-Instanzmodelle, sowie die Überwachung der Gesamtkonfiguration des R/3-Systems. Dazu werden diesem Modelltypen zwei Inference-Handler zugeordnet.

Der eine wird durch Veränderungen an den Zustandsattributen der untergeordneten R/3-Instanzen und der über „provides"- und „manages"-Assoziationen zugeordneten Workstationmodelle getriggert und bildet diese Werte analog zu dem in [SPEC ADG] beschriebenen „Condition-Rollup"-Verfahren auf einen Gesamtzustand für das SAP_R3_Sys-Modell ab.

Der andere Inference Handler dieses Modelltypen wertet die r3configServiceTable aller über die „consists_of"-Relation zugeordneten Modelle aus und ermittelt daraus, ob die aktuelle Konfiguration noch den Mindestanforderungen (siehe 4.2.2) genügt. Sollte dies nicht der Fall sein, wird ein Alarm für dieses R/3-System erzeugt. Dieser Inference-Handler wird durch folgende Ereignisse getriggert:

- Erzeugung eines SAP_R3_Sys-Modells.
- r3StateChange-Traps, die Veränderungen an der Konfiguration von Instanzen anzeigen.
- Veränderungen an den consists_of-Assoziationen, ausgelöst durch Hinzukommen oder Verschwinden von R/3-Instanzmodellen (SAP_R3_Inst).

Um diese Modellierung besser verstehen zu können, ist in Abbildung 7.5 dargestellt, wie das in Kapitel 3, Abschnitt 3.9 vorgestellte Beispielsystem mit den beschriebenen Modelltypen und Relationen in SPECTRUM modelliert wird.

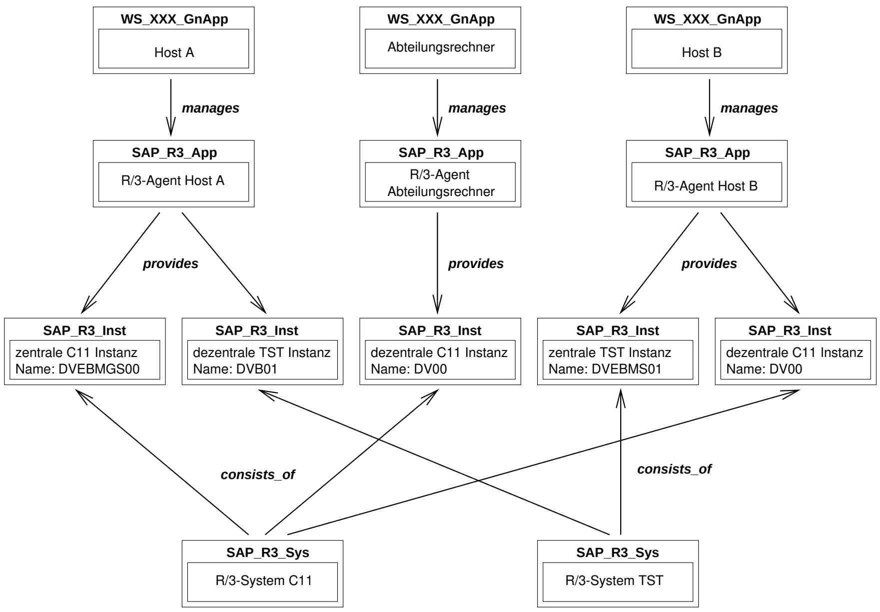

Abbildung 7.5: Modellierung eines R/3-Systems mit SPECTRUM

Die Inference-Handler der neuen R/3-Modelle wurden so entworfen, daß diese Modellierung ausgehend von der Benutzer- oder Autodiscovery-gesteuerten Erstellung der Workstation-Modelle automatisch generiert wird. Veränderungen an der Konfiguration des R/3-Systems werden ebenfalls automatisch erkannt und die Modellierung entsprechend angepaßt.

7.6.2 Implementierung des R/3-Trapmechanismus

SAP realisiert mit der aktuellen R/3-MIB einen sehr differenzierten Trap-Mechanismus. Wie in 5.3.2.2 bereits beschrieben, verfügen alle R/3-Traps über einen Parameter „Severity", der die Werte „normal", „warning" und „critical" annehmen kann. Das Zurücksetzen eines Trap mit Severity „warning" oder „critical" geschieht durch einen neuen Trap mit Severity „normal". Aus diesem Konzept ergibt sich zwingend, daß neben dem Typ des Traps insbesondere der Severity-Parameter bei der Bestimmung der SPECTRUM-internen Alarm-Severity berücksichtigt werden muß. Ein solches Konzept wird vom SPECTRUM-Alarmmanagement bisher nicht unterstützt, d.h. es läßt sich nicht mit Level I Tools realisieren. Das SPECTRUM-Konzept sieht nur eine statische Zuordnung von Trap-Typen zu Alarm-Severities vor.

Da die Verarbeitung von R/3-Traps aber von zentraler Bedeutung für das R/3-Management ist, wird folgende Lösung implementiert:

- Alle R/3-Traps werden zunächst nur auf Events abgebildet. Dies geschieht mit Hilfe einer Alert-Map. Da es nicht möglich ist, hier die Trap-Parameter auszuwerten, können noch keine Alarme erzeugt werden.

- Für jeden Trap-Typ wird ein Inference-Handler erstellt, der von Events dieses Typs getriggert wird und an Hand des Severity-Parameters das Zustandsattribut des R/3-Instanzmodells verändert. Der Zugriff auf dieses Attribut erfolgt über eine Asserter-Funktion, die gewährleistet, daß das Zustandsattribut den schlechtesten der aktuell von diesen Inference-Handlern gelieferten Werte erhält. Das IHAPI bietet bereits eine Inference-Handler-Klasse an, welche diese Asserter-Funktion realisiert [SPEC INF]. Daher müssen die neu zu erstellenden Inference-Handler nur von dieser Klasse abgeleitet werden, um die Funktion nutzen zu können.

  Die Asserter-Funktion sorgt gleichzeitig für die Generierung von Alarmen zu allen von den Inference-Handlern gemeldeten Events. Diese Alarme werden wieder gelöscht, sobald der entsprechende Inference-Handler wieder normalen Zustand meldet. Zu diesen Alarmen können Verweise auf Dateien angegeben werden, die die Ursache des Alarms näher erläutern. Diese Texte werden im Alarm-View dargestellt.

7.6.3 Visualisierung

Um die im vorangegangenen Abschnitt entworfenen Modelle dem Benutzer adäquat präsentieren zu können, wurden die in Abbildung 7.6 dargestellten Sichten entworfen. Diese Abbildung zeigt, wie das R/3-Beispielsystem aus Abschnitt 3.9 im SpectroGRAPH dargestellt wird.

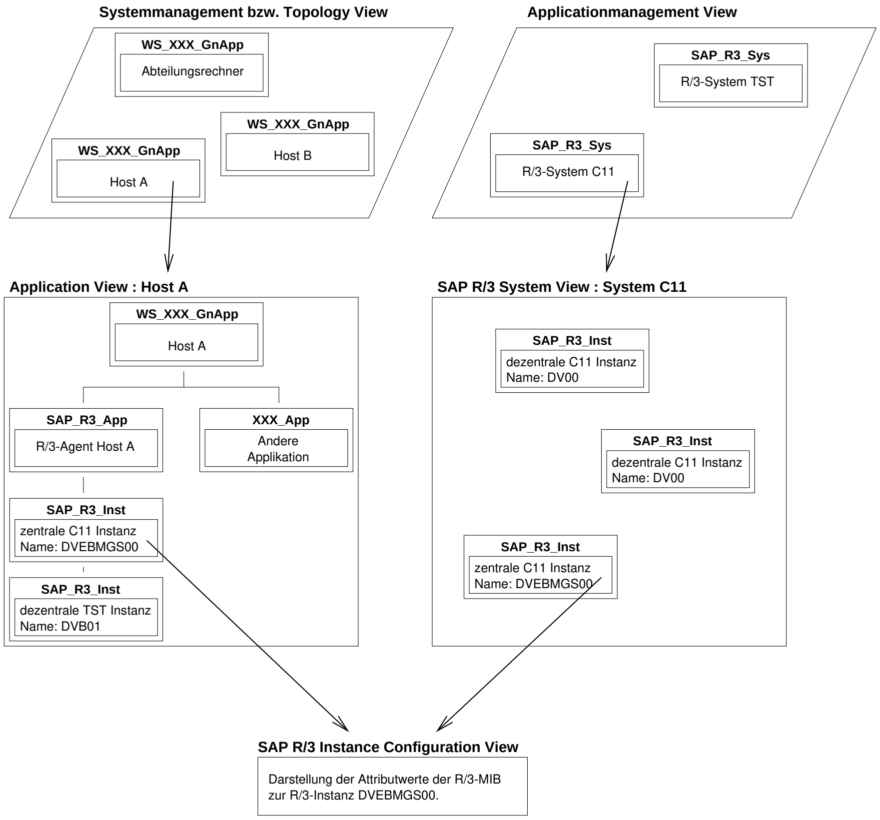

Abbildung 7.6: Darstellung eines R/3-Systems im SpectroGRAPH

Die Workstation-Modelle werden in der untersten Ebene der bereits bestehenden Topology- bzw. Systemmanagement-Sichten als Icons dargestellt. Die Farbe dieser Icons repräsentiert den aktuellen Zustand der Workstation unter Systemmanagement-Gesichtspunkten. Von diesen Icons aus lassen sich sowohl die MaestroVision-Sichten für das Systemmanagement als auch der Application-View erreichen. Falls auf der gewählten Workstation ein R/3-Agent läuft, erscheint im Application-View der R/3-Agent als Major Application zusammen mit Minor Applications für jede R/3-Instanz auf diesem System. Weitere auf diesem System laufende Anwendungen, wie z.B. das Datenbanksystem eines R/3-Systems, werden hier ebenfalls als Major- bzw. Minor-Applications dargestellt.

Mit Hilfe der View API wird eine neue Sicht für das Anwendungsmanagement erstellt. In dieser Sicht werden die Modelle der R/3-Systeme als Icons dargestellt, deren Farbe den Gesamtzustand des Systems repräsentiert. Von diesen Icons aus kann ein View geöffnet werden, der alle R/3-Instanzen dieses R/3-Systems darstellt. Dies ist durch Verfolgen der „consists_of“-Assoziationen des R/3-Systemmodells (SAP_R3_Sys) zu den Instanzmodellen (SAP_R3_Inst) möglich. Es handelt sich hier also um Darstellungen der gleichen Modelle, die auch im Application-View enthalten sind. Daher werden auch die aus dem Application-

View bekannten Icons verwendet.

Um die Attributwerte der R/3-MIB zu einer R/3-Instanz betrachten zu können, kann über die Icons der R/3-Instanzen auf eine Gib-View-Folge zugegriffen werden, deren erster View der SAP R/3 Instance Configuration View ist. Die gesamte Folge ist in Abbildung 7.7 dargestellt.

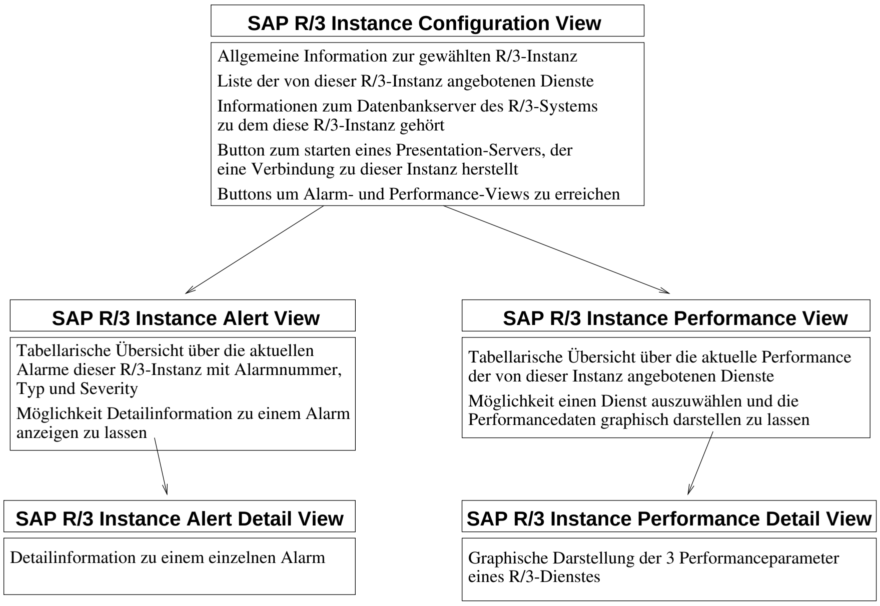

Abbildung 7.7: Präsentation der R/3-MIB

Diese Views wurden im Rahmen einer prototypischen Implementierung realisiert und sind im folgenden Kapitel, Abschnitt 8.2.3, detailliert beschrieben.

7.6.4 Integration von Organisations- und Anwendungssicht

Um die verschiedenen in einem Unternehmen eingesetzten Anwendungen in ihren Unternehmenskontext einordnen zu können, wird der bereits bestehende Org-Chart-View um einen Modelltyp erweitert, der die Zuordnung von Anwendungen zu Unternehmensbereichen ermöglicht. Dieser neue Modelltyp Org_Uses wird in Anlehnung an den bereits existierenden Modelltyp Org_Owns realisiert, der die Modellierung von Eigentumsbeziehungen zu Modellen des Topology-View (Geräte) und des Location-View (Zimmer, Gebäude, Standorte, ...) ermöglicht.

Dazu wird eine neue Relation „uses“ eingeführt, die beschreibt, welche Organisationseinheit eine Anwendung nutzt. Abbildung 7.8 zeigt, wie die Integration von Organisations- und Anwendungssicht mit Modellen diese Typs erfolgt.

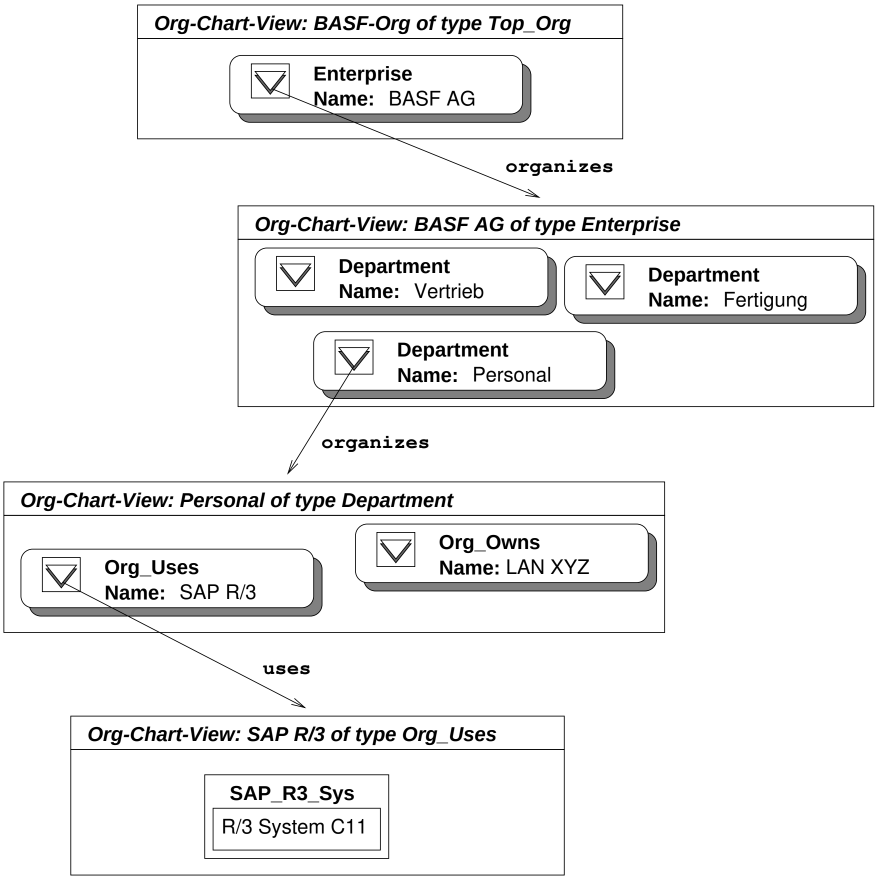

Abbildung 7.8: Integration von Organisations- und Anwendungssicht

Die oberste Ebene des Org-Chart-View ist ein Modell des Typs Top_Org, das automatisch bereitgestellt wird. Mit Hilfe von Modellen der Typen Enterprise, Division, Department, . . . wird die Unternehmensstruktur modelliert. Org_Uses- und Org_Owns-Modelle müssen dabei nicht notwendigerweise auf der untersten Hierarchieebene erstellt werden. Wenn Anwendungen z.B. von ganzen Unternehmensbereichen genutzt werden, können sie auch auf dieser Ebene durch Org_Uses-Modelle repräsentiert werden.

Zu dem Org_Uses-Modelltyp wird ein Inference-Handler realisiert, der nach dem bekannten Condition-Rollup-Verfahren den Zustand des Org_Uses-Modells aus dem Zustand der über die „uses"-Relation zugeordneten Anwendungsmodelle ableitet.

Auf der Grundlage der „organizes"-Relation, die den Org-Chart-View strukturiert, können dann auch die Modelltypen dieser Hierarchie mit einer Condition-Rollup-Funktion versehen werden. Damit wäre es z.B. möglich Verfügbarkeitsstatistiken für die verschiedenen Anwendungen der Unternehmensbereiche zu erstellen und über diese Icons zugänglich zu machen.

7.7 Einsatzszenarien des R/3-Managementmoduls bei der BASF AG

Wie in Kapitel 6 beschrieben soll das R/3-Managementmodul bei der BASF AG hauptsächlich vom Support Center und der Netzbetriebsgruppe genutzt werden. Dieser Abschnitt stellt am Beispiel von typischen Aufgaben dieser Gruppen den Einsatz der R/3-Managementanwendung vor.

7.7.1 Support Center

Eine typische Situation im Support Center ist der Anruf eines Benutzers, der mitteilt, daß seine R/3-Anwendung „steht". Durch die Einbindung der R/3-Systeme in die Organisationssicht kann der Mitarbeiter des Support Centers schnell herausfinden, welches R/3-System der Anrufer verwendet und ob für dieses System Störungsmeldungen vorliegen. Ist das System tatsächlich gestört, kann durch Betrachtung des Zustands der einzelnen Instanzen dieses Systems im SAP R/3 System View festgestellt werden, ob es sich um eine zentrale Störung handelt, von der alle Instanzen betroffen sind, oder ob nur die Instanz dieses Benutzers betroffen ist. Im letzteren Fall kann dem Benutzer vorübergehend ein anderer Application-Server zugewiesen werden. In jedem Fall wird ein präziser Fehlerreport an die Netzbetriebsgruppe geschickt.

Dieses Beispiel zeigt die Notwendigkeit, die Zustandsinformation über die Netz-, System- und Anwendungsressourcen eines R/3-Systems zu einem Gesamtzustand zu verdichten.

Kann keine Störung am R/3-System festgestellt werden, ist entweder die Netzverbindung zum Endsystem unterbrochen oder es liegt eine Störung des Endsystems bzw. des dort laufenden Presentation-Servers vor. Netzprobleme können durch Betrachtung der Netztopologiesichten abgeklärt werden. Falls danach eine Störung des Benutzerssystems feststeht, die auch durch Neustart des Presentation-Servers oder des Benutzerrechners nicht vom Anwender selbst behoben werden kann, wird die lokale Wartungsgruppe informiert. An dieser Stelle könnte ein Produkt wie WinWatch sinnvoll eingesetzt werden, um auch solche Störungen näher analysieren zu können.

Zusammenfassend läßt sich feststellen, daß der wichtigste Nutzen der integrierten Managementwerkzeuge für diese Gruppe ein deutlich verbesserter Informationszugriff und die dadurch ermöglichte effizientere Betreuung der Benutzer ist. Durch die beschriebene Eingrenzung der gemeldeten Fehler werden auch die nachgeschalteten Expertengruppen entlastet.

7.7.2 Netzbetriebsgruppe

Die Netzbetriebsgruppe wird parallel den Zustand abstrakter Objekte der Netz-, System- und Anwendungssichten überwachen und dort erkannte Fehler zu ihrem Ursprung verfolgen. Dies wird durch die jeweiligen hierarchischen Sichten unterstützt. Störungen an den R/3-Systemen können durch Veränderung der Farbe der SAP_R3_Sys-Icons im Applicationmanagement View auf einen Blick erkannt werden. Mit Hilfe des SAP R/3 System Views zu diesem Systemmodell können ebenso schnell die betroffenen R/3-Instanzen ermittelt werden. Melden alle Instanzen dieses Systems Fehler, liegt mit großer Wahrscheinlichkeit eine Störung des Datenbanksystems vor. Ist dagegen nur eine Instanz gestört, kann mit Hilfe des Alarm-Views oder des SAP R/3 Instance Alert View zu diesem Instanzmodell die Fehlerursache ermittelt werden. Ist die Fehlerursache mit Hilfe der Alarmbeschreibung und der weiteren Attribute der R/3-MIB nicht eindeutig zu ermitteln, kann vom SAP R/3 Instance Configuration View aus auch auf die R/3-spezifischen Managementwerkzeuge zugegriffen werden. Handelt es sich um einen durch mangelhafte Netz- oder Systemressourcen verursachten Fehler, kann dieser möglicherweise durch Rekonfiguration dieser Objekte behoben werden. Liegt dagegen eine R/3-interne Störung vor, so wird eine detaillierte Fehlermeldung an den SAP Basis Support gerichtet. Im allgemeinen wird die Netzwerkbetriebsgruppe mit Hilfe der Netz- und Systemmodelle Engpässe oder Störungen der R/3-Ressourcen jedoch bemerken, bevor sie Auswirkungen auf R/3-Komponenten haben.

Kapitel 8

Prototyp des R/3-Managementmoduls

Um die in den vorangegangenen Abschnitten vorgestellten Konzepte zur Realisierung eines R/3-Managementmoduls auf ihre Durchführbarkeit und Adäquatheit prüfen zu können, wurde eine prototypische Implementierung durchgeführt. Die folgenden Abschnitte beschreiben, welche Konzepte aus Kapitel 7 realisiert wurden, sowie die dazu erforderliche Testumgebung.

8.1 Testumgebung

Die in Kapitel 5 beschriebenen Erweiterungsvorschläge zur R/3-MIB wurden von der SAP aufgegriffen und fast vollständig in die Version 2.4 der R/3-MIB aufgenommen, welche Ende März 1995 veröffentlicht wurde. Da eine endgültige Version innerhalb der Laufzeit dieser Diplomarbeit nicht zu erwarten war, wurde diese Version als Implementierungsgrundlage gewählt. Wegen der noch nicht abgeschlossenen MIB-Spezifikation stand zu diesem Zeitpunkt auch noch kein entsprechender SNMP-Agent zur Verfügung. Um dennoch mit der Implementierung beginnen und diese auch testen zu können, wurden folgende Lösungsmöglichkeiten des Problems geprüft:

- Erstellen eines Testagenten auf Basis eines public-domain SNMP-Entwicklungspakets. [SNMP CMU] [SNMP MIT] Ein solcher Testagent könnte statisch definierte Werte auf SNMP-Anfragen liefern.
- Verwendung des SNMP-Agenten eines vorhandenen Hubs, dessen MIB vergleichbar strukturiert ist und die erforderlichen Datentypen bietet. Um mit dem R/3-Managementmodul auf diese MIB zugreifen zu können, müßten die Object Identifier der Attribute des R/3-Modelltypen mit Hilfe des MTE

entsprechend angepaßt und nach Abschluß der Tests wieder zurückgesetzt werden.

Da der Mehraufwand für die erstgenannte Lösung in keinem Verhältnis zu den zu erzielenden Vorteilen steht, wurde die zweite Variante gewählt.

Zur Generierung der in der R/3-MIB vorgesehenen Traps wurde das Programm **snmptrap** aus dem SNMP Development Kit des Massachusetts Institute of Technology [SNMP MIT] verwendet, welches es gestattet, Traps mit frei definierbaren Parametern zu erzeugen.

8.2 Beschreibung des Prototypen

Im Rahmen dieser Arbeit wurden von dem in Abschnitt 7.6 vorgestellten Lösungskonzept diejenigen Anteile implementiert, die sich mit Level 1 Werkzeugen realisieren lassen. Diese Einschränkung war zum einen aus Zeitgründen notwendig, zum anderen ist es wenig sinnvoll, eine aufwendige Level 2 Integration auf der Basis einer nur vorläufig definierten MIB durchzuführen.

Die realisierte Level 1 Integration ermöglicht den Zugriff auf die MIB-Attribute der R/3-Instanzen jeweils eines Host. Die geplante gemeinsame Darstellung aller Instanzen eines R/3-Systems in einem View konnte damit nicht verwirklicht werden.

8.2.1 Modelltypen

Die Level 1 Werkzeuge reichen bereits aus, um die Modelltypen für die R/3-MIB (SAP_R3_MIB), den R/3-Agenten (SAP_R3_App) und den R/3-Host (WS_XXX_GnApp) wie geplant zu realisieren. Als konkretes Beispiel für einen R/3-Host wurde der Modelltyp WS_SUN_GnApp implementiert, der also den Modelltypen für Workstations der Firma SUN (WS_SUN) und den GnSNMPDev-Modelltyp als Basisklassen erhielt.

Die ererbten Inference-Handler dieser Modelltypen reichen aus, um den R/3-Agenten eines Workstation-Modells zu erkennen und als SAP_R3_App-Modell zu repräsentieren. Die Darstellung dieses Agenten-Modells erfolgt dann über den Application-View des Workstation-Modells.

8.2.2 Icons

Für den Modelltyp WS_SUN_GnApp wurde auf der Grundlage des Icons für den Modelltypen WS_SUN ein erweitertes Icon erstellt, von dem aus sich neben den MaestroVision-Views für das Systemmanagement nun auch Application-und Mib-Walk-View erreichen lassen. Dieses Icon wird in allen Views für Modelle des Typs WS_SUN_GnApp verwendet.

Das Standard-Icon für Applikationen konnte unverändert für die neue R/3-Applikation SAP_R3_App übernommen werden. Dieses Icon ermöglicht den Zugriff auf den nachfolgend beschriebenen SAP R/3 Host View.

Abbildung 8.1 zeigt die Funktion dieser neuen Icons im Application View. Die beiden Application-Icons auf der linken Seite repräsentieren die MIB-II-Teilbäume, die der Testagent unterstützt.

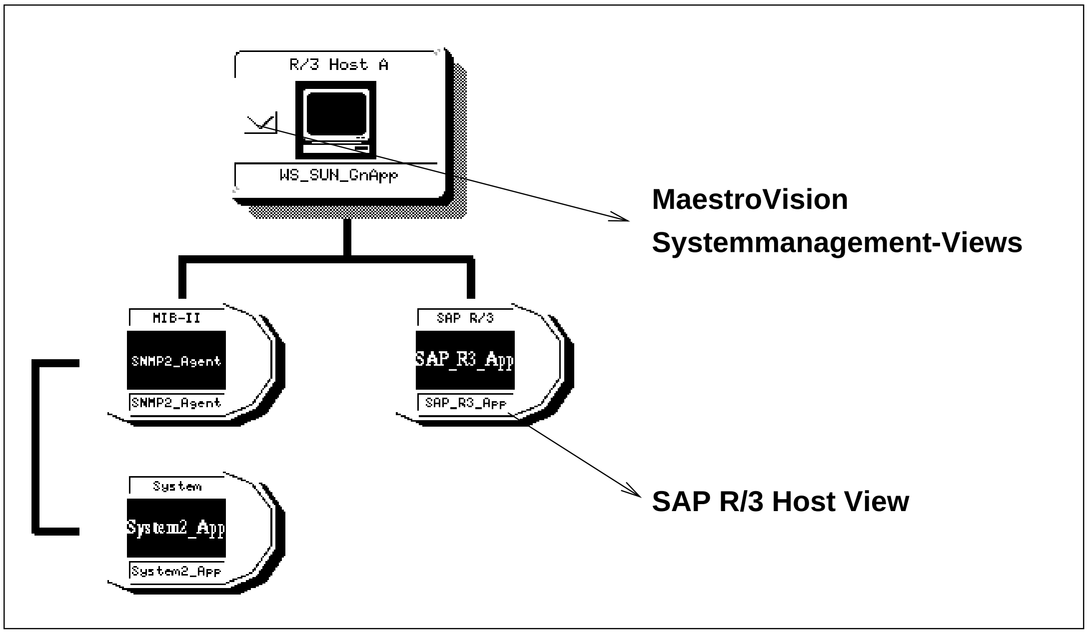

Abbildung 8.1: Application View mit R/3-Modellen

Die aktuelle Farbe der Icons zeigt den Zustand des repräsentierten Modells an. Die einzelnen Farben haben folgende Bedeutung:

Grün Normaler Betrieb

Gelb Leichte Störung, Betrieb eingeschränkt.

Orange Schwerer Fehler, Ausfall möglich.

Grau Der aktuelle Zustand kann wegen des bereits bekannten Ausfalls einer Netzkomponente z.Zt. nicht festgestellt werden.

Diese Zustände können sowohl bei Gerätemodellen als auch bei Anwendungsmodellen auftreten. Dagegen sind die folgenden Zustände nur für Gerätemodelle definiert:

Rot Verbindung zur Managementstation ausgefallen.

Blau Es konnte seit der Modellierung des Geräts noch keine Verbindung aufgebaut werden.

8.2.3 Views

Um die Werte der R/3-MIB im SpectroGRAPH angemessen darstellen zu können wurden die in Abbildung 7.7 vorgeschlagenen GIB-Views implementiert. Da diese View-Folge für einzelne R/3-Instanzen entworfen wurde, Modelle des Typs SAP_R3_App aber alle R/3-Instanzen eines Host repräsentieren, wurde zusätzlich der SAP R/3 Host View erstellt, der die Auswahl einer R/3-Instanz ermöglicht, also die Modelle des Typs SAP_R3_Inst ersetzt.

8.2.3.1 SAP R/3 Host View

Aus dem Application-View erreicht man über das untere Feld des R/3-Applikations-Icons oder den Icon-Menü-Eintrag „Configuration“ den SAP R/3 Host View.

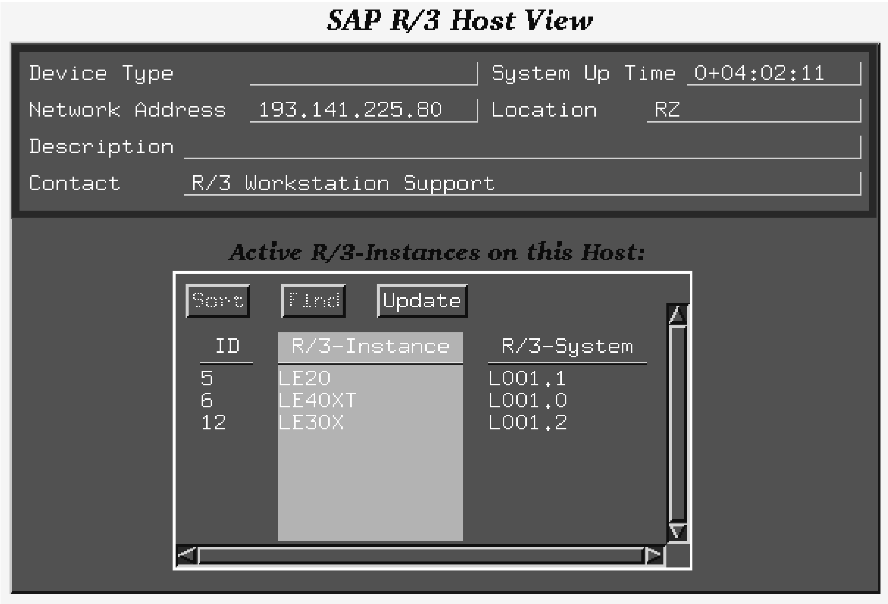

Abbildung 8.2: SAP R/3 Host View

Der Kopf dieses Views zeigt allgemeine Information zum Host-System wie Rechnertyp, IP-Adresse und Standort des Geräts. Die aktuelle Farbe der Umrandung um diese Daten zeigt den Zugriffsstatus auf den R/3-Agenten an. Dabei bedeutet Grün eine intakte Verbindung, Rot einen Verbindungsausfall und Grau einen unterdrückten Ausfall, z.B. weil bereits die Verbindung zum Gerätemodell unterbrochen ist. Diese Information ist zur Bewertung der Aktualität der dargestellten Informationen wichtig.

Die darunter dargestellte Tabelle stellt einen Ausschnitt aus der r3systemTable und der r3configTable dar, nämlich die Spalten r3systemNumber, r3configInstanceName und r3systemName. Es werden also die Namen der z.Zt. auf diesem Rechner laufenden R/3-Instanzen mit Angabe des zugehörigen R/3-Systems dargestellt. Um Information zu einer R/3-Instanz abzurufen, wird durch Doppelklick auf einen der Instanznamen der SAP R/3 Instance Configuration View geöffnet.

8.2.3.2 SAP R/3 Instance Configuration View

Durch das Selektieren einer Tabellenzeile im SAP R/3 Host View wurde implizit die sog. Instance ID dieses Views auf die Nummer der Zeile gesetzt. Da diese Instance ID an die original Object-Identifier der Attribute dieses Views angehängt wird, werden für diesen View stets die Attributwerte der zuvor gewählten R/3-Instanz gepollt und dargestellt.

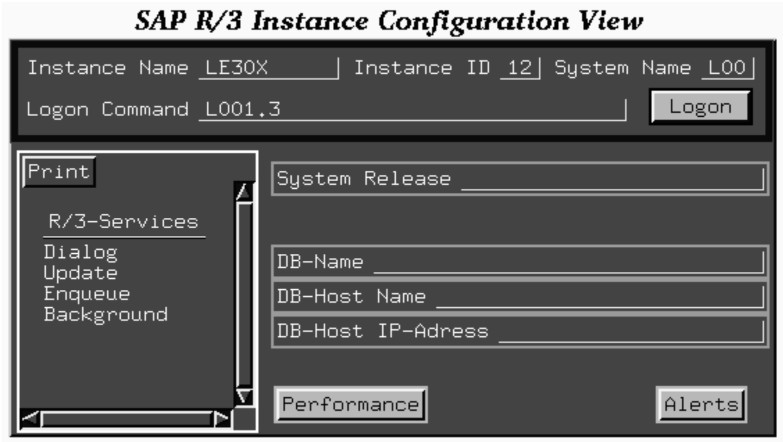

Abbildung 8.3: SAP R/3 Instance Configuration View

Der Kopf dieses Views zeigt die schon aus dem SAP R/3 Host View bekannten Parameter der Instanz und zusätzlich den r3systemLogonString, der bekanntlich ein UNIX-Kommando zum Starten eines Presentation-Servers enthält, der automatisch eine Verbindung zu dieser R/3-Instanz aufbaut. Durch Mausklick auf den mit „Logon" beschrifteten Button sollte dieses Kommando direkt ausgeführt werden. Diese Funktion ließ sich leider nicht realisieren, da es nicht möglich ist, View-Parameter an das von diesem Button aufgerufene Shellscript zu übergeben. Mittels Cut-and-Paste kann das Kommando aber problemlos in eine Shell kopiert und dort gestartet werden. Damit ist also insbesondere ein einfacher Zugriff auf die R/3-Managementwerkzeuge des CCMS gegeben. Die farbige Umrandung des View-Kopfes stellt wieder den Verbindungsstatus dar.

Die Tabelle unten links stellt die von dieser Instanz angebotenen Dienste dar. Dazu wird der in der r3configServiceTable enthaltene Aufzählungstyp (r3configServiceType) auf die Namen der entsprechenden R/3-Basisdienste abgebildet. Neben dieser Tabelle findet sich Information zur Version des R/3-Systems (r3-systemReleaseString) sowie Name (r3configDBName), Host-Name (r3configDB-HostName) und IP-Adresse (r3configDBHostIpAdr) des Datenbankservers. Dabei können DB-Host-Name und DB-Host IP-Adress auch ein Gateway zum Datenbankserver beschreiben.

Mit den Buttons „Performance“ und „Alerts“ erreicht man den SAP R/3 Instance Performance View bzw. den SAP R/3 Instance Alert View.

8.2.3.3 SAP R/3 Instance Performance View

Der Performance View bietet eine Übersicht über die aktuellen Leistungsparameter der R/3-Dienste dieser Instanz. Es handelt sich also um die r3perfTable mit den Spalten r3perfServiceType, r3perfEventsFrequency, r3perfResponeTime und r3perfWaitTime. Durch Selektieren eines R/3-Diestes wird der SAP R/3 Instance Performance Detail View geöffnet.

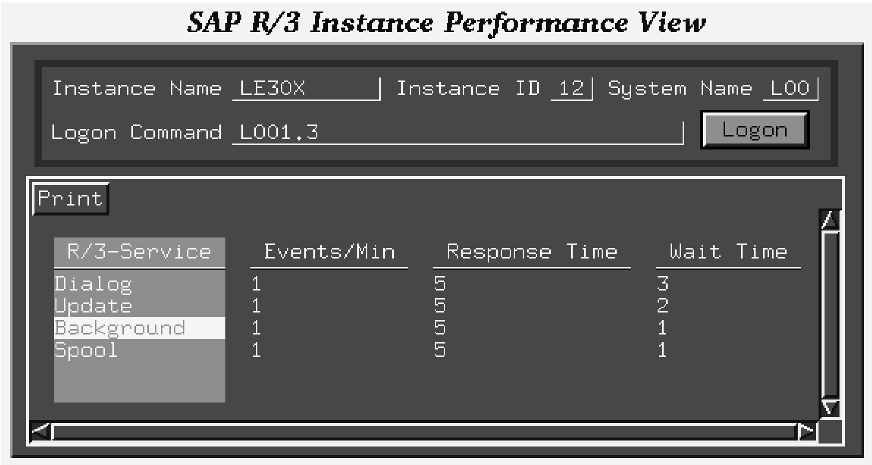

Abbildung 8.4: SAP R/3 Instance Performance View

Der Kopf dieses Views ist identisch mit dem Kopf des SAP R/3 Instance Configuration View.

8.2.3.4 SAP R/3 Instance Performance Detail View

Durch das Auswählen eines R/3-Dienstes im SAP R/3 Instance Performance View wurde die Instance ID dieses Views zusätzlich um den Index dieses Dienstes erweitert. Dadurch kann in diesem View auf die Performance-Werte des gewählten Dienst-Typs zugegriffen werden. Dies wurde genutzt um die drei zur Verfügung stehenden Leistungsparameter in einem sog. Multi Attribute Graph darzustellen, der es erlaubt den zeitlichen Verlauf dieser Werte zu verfolgen.

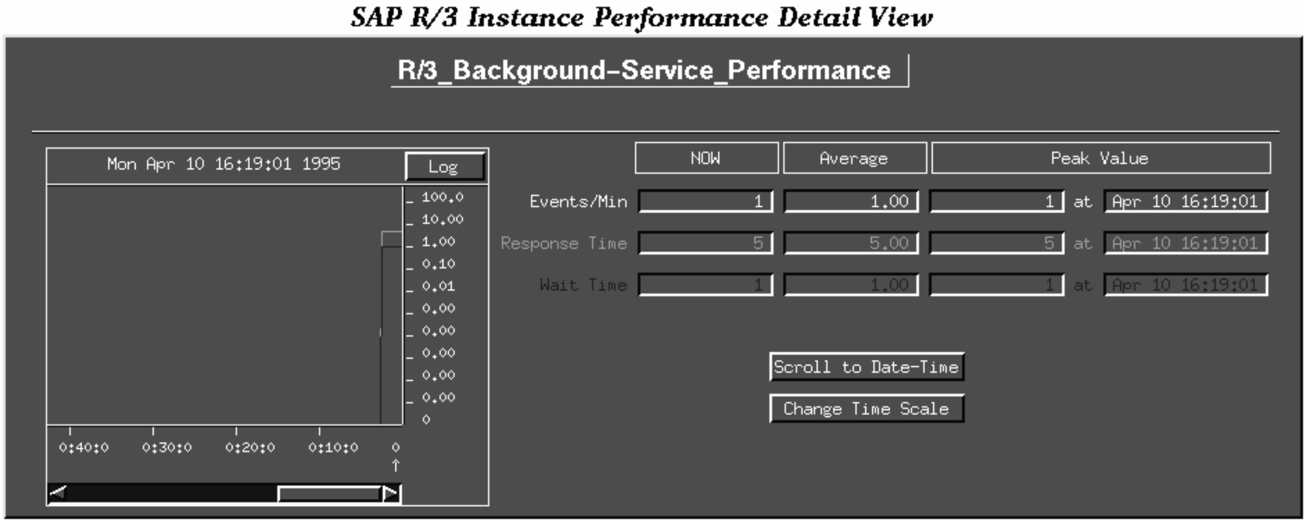

Abbildung 8.5: SAP R/3 Instance Performance Detail View

Der Parameter r3perfEventsFrequency wird in gelber, r3perfResponseTime in grüner und r3perfWaitTime in roter Farbe als Graph dargestellt. Auf der rechten Seite werden der aktuelle, durchschnittliche und höchste je erreichte Wert für die einzelnen Parameter als Zahlen angezeigt. Für den Maximalwert wird zusätzlich der Zeitpunkt dieser Messung vermerkt.

Die vertikale Skala des Graphen läßt sich von logarithmischer auf lineare Einteilung umschalten. Mittels des Scrollbar oder durch Angabe eines bestimmten Zeitpunkts über den Scroll to Date-Time Button kann der Graph horizontal verschoben werden. Der Maßstab für die horizontale Zeitachse kann über den Change Time Scale Button verändert werden.

Ein Polling der dargestellten Attributwerte erfolgt nur solange der View geöffnet ist. Für eine längerfristige Beobachtung des Lastverlaufs muß der View entsprechend lange geöffnet bleiben.

8.2.3.5 SAP R/3 Instance Alert View

Der Alert View zeigt neben dem bekannten Kopf eine Auswahl von Spalten der r3AlertTable zu dieser Instanz. Im einzelnen sind dies die Nummer (r3alertNumber), der Zeitpunkt (r3alertTimeStamp), der Typ (r3alertClass) und die Gewichtung (r3alertSeverity) des Alarms.

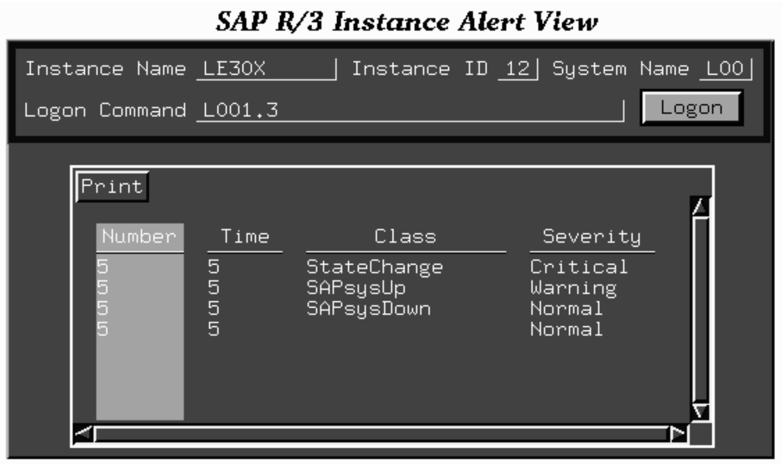

Abbildung 8.6: SAP R/3 Instance Alert View

Die übrigen Spalten der r3alertTable enthalten z.T. längere Text-Strings und können daher nicht in dieser Tabellenform dargestellt werde. Um die vollständige Information zu einem einzelnen Alarm abzurufen, kann durch Auswahl eines Alarms dieser Tabelle der SAP R/3 Instance Alert Detail View geöffnet werden.

8.2.3.6 SAP R/3 Instance Alert Detail View

Dieser View zeigt nun alle Parameter des im SAP R/3 Instance Alert View ausgewählten Alarms. Dies ist wieder durch Erweiterung der Object-Identifier um die Instance ID möglich.

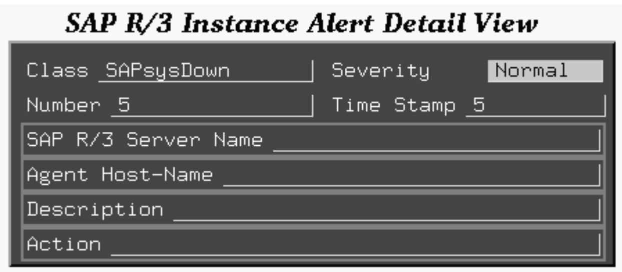

Abbildung 8.7: SAP R/3 Instance Alert Detail View

Neben den bereits im SAP R/3 Instance Alert View dargestellten Parametern werden zusätzlich folgende Werte angezeigt:

SAP R/3 Server Name : r3alertSapNameString, der Name der von diesem Alarm betroffen R/3-Instanz.

Agent Host Name : r3alertAgentHostName, der Name des Rechners, auf dem der Agent läuft, der diesen Alarm generiert hat.

Description : r3alertMsgString, eine Beschreibung der Ursache des Alarms

Action : r3alertActionString, eine Anweisung zum Navigieren im Presentation-Server. Damit soll eine weitere Analyse dieses Fehlers unterstützt werden.

Die Darstellung der Severity des Alarms erfolgt in diesem View über ein sog. Color-Enumeration Feld, das es gestattet, Integer-Werte nicht nur auf einen Text, sondern auch auf eine Farbe abzubilden. Hier wird der Wert für critical auf Rot, warning auf Gelb und normal auf Grün abgebildet. Dies entspricht genau der Darstellung von Alarmen im CCMS.

8.3 Ergebnisse der prototypischen Implementierung

8.3.1 Probleme mit MaestroVision-Modelltypen

Bei Tests der im vorangegangenen Kapitel beschriebenen Implementierung zeigte sich, daß der Condition-Rollup von den Modellen der Komponenten einer Workstation (Processor, File-System, ...) zu dem übergeordneten Workstationmodell für das abgeleitete Workstation-Modell WS_SUN_GnApp nicht mehr funktioniert.

Um auszuschließen, daß dieses Problems durch konkurrierenden Zugriff von Inference-Handlern der Basisklassen auf Zustandsattribute verursacht wird, wurde der Basismodelltyp GnSNMPDev wieder entfernt. Dennoch zeigte der nun nur noch von WS_SUN abgeleitete Modelltyp WS_SUN_GnApp keine Reaktion auf Zustandsveränderungen seiner Komponenten.

Die Ursache des Problems konnte leider auch durch eine Anfrage an die Firma Calypso, die den Modelltypen WS_SUN entwickelt hat, nicht geklärt werden.

8.3.2 Erforderliche Weiterentwicklungen

Neben der Behebung des zuvor beschriebenen Problems mit den MaestroVision-Modelltypen ist die Implementierung des R/3-Trapmechanismus dringend erforderlich, um das R/3-Managementmodul sinnvoll einsetzen zu können. Der

SPECTRUM-Hersteller Cabletron arbeitet bereits an einer flexibleren Lösung für die Event-Alarm-Abbildung, die es erlaubt, Event-Parameter bei der Abbildung zu berücksichtigen. Bis diese Lösung zur Verfügung steht, bietet sich als Interims-Lösung eine Überwachung der r3alertTable mit Hilfe des Werkzeugs SpectroWATCH [SPEC WAT] an. Dieses gestattet die Überwachung einzelner Attribute eines Modells an Hand frei konfigurierbarer Grenzwerte, bei deren Verletzung Alarme erzeugt werden können. Da SpectroWATCH eine reine Managementanwendung ist, erfordert diese Form der Überwachung netzlastintensives Polling und kann deshalb nur in begrenztem Umfang angewandt werden.

Sobald praktische Erfahrungen mit dem Prototypen vorliegen, kann entschieden werden, welche der in Kapitel 7 vorgestellten Konzepte noch zusätzlich realisiert werden sollen.

Kapitel 9

Ausblick

Von den vielfältigen Managementaspekten eines R/3-Systems wurde in dieser Arbeit hauptsächlich die Integration des Aspekts „Verfügbarkeitsüberwachung" betrachtet. Der folgende Abschnitt zeigt, welche Erfolge dabei erzielt wurden. Im Anschluß daran wird diskutiert, inwieweit sich die bestehenden Managementkonzepte als geeignet erwiesen haben und wo Probleme aufgetreten sind. Abschließend erfolgt ein Ausblick auf weitere Entwicklungsmöglichkeiten im integrierten Anwendungsmanagement und welche Rahmenbedingungen hierzu geschaffen werden müssen.

9.1 Analyse der Ergebnisse

Auf der Grundlage der an Hand der erkannten Anforderungen verbesserten R/3-MIB wurde ein Konzept zur Modellierung aller managementrelevanten Komponenten eines R/3-Systems entwickelt. Die erforderlichen Objekte aus dem Systemmanagementbereich standen bereits zur Verfügung und konnten in die Modellierung integriert werden, wobei jedoch die in 8.3.1 beschriebenen Probleme auftraten. Es wurden neue Modelle für die verteilte Anwendung als Ganzes und für ihre einzelnen Komponenten entworfen. Als Grundfunktionalität dieser Modelle wurde eine Zustandseskalation von den Modellen der System- und Anwendungskomponenten zum Modell des Gesamtsystems entworfen. Dieses Konzept beschreibt auch, wie diese Modelle dem Benutzer präsentiert werden sollen. Dazu wurden die bekannten geographischen und netztopologischen Sichten um eine Sicht für verteilte Anwendungen erweitert.

Diese Konzepte konnten aus zwei Gründen nur zum Teil implementiert werden:

- Bis zum Abschluß dieser Arbeit stand kein Agent zur R/3-MIB zur Verfügung.

- Die Modellierung von so abstrakten Objekten wie Anwendungskomponenten oder gar verteilten Anwendungen als Ganzes wird von heute verfügbaren Managementplattformen noch schlecht unterstützt. Aufgrund ihrer ursprünglichen Entwicklung als Netzmanagementanwendungen sind sie noch zu stark auf die Modellierung von Hardwarekomponenten fixiert.

Tatsächlich implementiert wurde eine Managementanwendung zur interpretierten Darstellung der R/3-MIB, die sich nahtlos in die bestehenden Systemmanagementkonzepte der Plattform einfügt. Damit ist eine Überwachung und grobe Fehleranalyse der R/3-Anwendungskomponenten eines Hosts möglich.

Als Anregung zur Weiterentwicklung der Managementplattform wurde ein Konzept zur Integration der neuen Anwendungssichten in die bestehende Unternehmensstruktursicht entworfen, um so die Bedeutung der Anwendung für einzelne Unternehmensaufgaben beschreiben zu können.

9.2 Bewertung der verwendeten Methoden

Die erarbeiteten Konzepte zeigen, daß es möglich ist, mit den ursprünglich für das Management von Netzkomponenten entworfenen Methoden eine integrierte Überwachung verteilter Anwendungen zu realisieren.

Dabei gelangen diese Methoden jedoch an ihre konzeptionellen Grenzen. Das einfache Internet-Informationsmodell ist dafür verantwortlich, daß die in der R/3-MIB eigentlich beschriebenen Objekte dort nur mit Mühe zu erkennen sind. Daher befaßt sich ein erheblicher Teil des Implementierungskonzepts mit der sinnvollen Abbildung der von der R/3-MIB angebotenen Managementinformation auf die logischen Objekte eines R/3-Systems. Da die Definition solch komplexer und abstrakter Managed Objects, wie sie für das R/3-Management erforderlich sind, in diesem Informationsmodell nicht vorgesehen ist, wird ihre Abbildung auf geeignete Repräsentationen von den Managementplattformen bisher kaum unterstützt. Die hier erarbeiteten Konzepte können als Grundlage für die Entwicklung generischer Modellierungskonzepte für verteilte Anwendungen dienen, wie sie schon heute für Netzkomponenten vorhanden sind (siehe 7.4).

Die beschriebenen Beschränkungen des SNMP-Protokolls behindern die Erweiterung der Konzepte von reiner Überwachung hin zu Steuerung.

9.3 Entwicklungsmöglichkeiten

Die in dieser Arbeit betrachteten Überwachungsaspekte sind nur ein Ausschnitt aus den vielfältigen Tätigkeiten, die erforderlich sind, um die Verfügbarkeit eines

R/3-Systems für die Benutzer zu sichern. Im folgenden wird diskutiert, welche Teilaufgaben in Zukunft noch dem integrierten Management überantwortet werden können bzw. sollten.

Überwachung und Fehlerbehebung Die entwickelten Konzepte zur Überwachung der R/3-Systeme verlangen noch immer die ständige oder zumindest regelmäßige Aufmerksamkeit des Managementpersonals. Um diese Personen zu entlasten muß die nächste Weiterentwicklung in diesem Bereich die Anbindung an ein Trouble-Ticket-System zur automatischen Weiterleitung von spezifischen R/3-Fehlern an entsprechende Fachleute sein. Sobald mehr Erfahrung mit dem R/3-Anwendungsmanagement besteht, kann geprüft werden, ob bestimmte Fehlersituationen immer gleiche Reaktionen erfordern, die dann automatisiert werden sollten. Dieses Vorgehen hat sich im Bereich des Systemmanagements bereits bewährt. Dazu muß natürlich schreibender Zugriff auf ausgewählte Anwendungsparameter gewährt werden.

Tuning und Administration Im Bereich der Leistungserhaltung bzw. -steigerung eines R/3-Systems gibt es noch zahlreiche Automatisierungsmöglichkeiten. Durch die Modellierung eines R/3-Systems als abstrakte Einheit hat die Managementstation einen Überblick über alle R/3-Komponenten dieses Systems. Dies kann genutzt werden, um eine manuelle oder besser automatische Lastverteilung zwischen den R/3-Instanzen vorzunehmen.

Dieser globale Überblick ermöglicht noch verschiedene andere Erweiterungen, wie z.B. die Einführung einer für das ganze R/3-System gültigen Benutzerdatenbank, die dann auch in die Benutzerverwaltung der Host-Systeme integriert werden kann.

Alle hier beschriebenen Erweiterungsmöglichkeiten erfordern erheblich umfangreichere MIBs und Managed Objects mit mächtigerem Funktionsumfang. Deshalb und aufgrund der Ergebnisse bei der Integration des Überwachungsaspekts darf bezweifelt werden, daß alle diese Managementszenarien auf der Grundlage des Internet-Management-Framework implementiert werden können. Hier fehlt insbesondere ein mächtiges Informationsmodell, das es erlaubt, die Managementinformation besser zu strukturieren und mehr Funktionalität in den Agenten zu verlagern. Ferner müssen die beschriebenen Mängel des Kommunikationsmodells behoben werden.

Generell stellt sich die Frage, inwieweit eine nachträgliche Integration von Anwendungen in bestehende Managementkonzepte erfolgreich sein kann und ob nicht die Berücksichtigung von Managementanforderungen ein integraler Bestandteil der Anwendungsentwicklung sein sollte. Hier bietet z.B. ODP[1] als Rahmenwerk zur

[1] Open Distributed Processing

Entwicklung verteilter Anwendungen gute Ansätze [ISO 10746]. Eine Einführung zu ODP findet sich in [HEAB 93] und [SZ 92].

Anhang A

Die SAP R/3 MIB Version 2.4

```
-- #==============================================================================
-- #
-- #  SAP R/3 Network Management MIB Definition
-- #
-- #  Created:     November 28 1994
-- #  Modified:    March    02 1995
-- #
-- #  Version:     2.4
-- #  SAP-Release: >= 3.0
-- #
-- #  Proposed by
-- #
-- #  Stefan Deml, TU-Muenchen
-- #  Wilfried Cleres, SAP-SNI-CC, Walldorf
-- #  Ulrich Haebel, SNI SU MR, Muenchen
-- #  Nicolai Jordt, SAP-Basis, Walldorf
-- #  Matthias Huber, IBM-ENC, Heidelberg
-- #  Christos Stieglitz, SUN/gnosys,
-- #  Dimitris Dovas, SAP-HP-CC, Walldorf
-- #  Thomas Strohmaier, SAP-DEC-CC, Walldorf
-- #  Guenter Zachmann, SAP-Basis, Walldorf
-- #
-- #
-- #  SAP AG
-- #  Neurottstrasse 16
-- #  69190 Walldorf Germany
-- #
-- #==============================================================================

SAP-MIB DEFINITIONS ::= BEGIN

IMPORTS
        OBJECT-TYPE
                FROM RFC1212
        DisplayString
```

```
                FROM RFC1213-MIB
        enterprises,
        Counter, TimeTicks, IpAddress
                FROM RFC1155-SMI;

-- ===========================================================================
-- The SAP enterprise MIB tree. SAP officially acquired the enterprise
-- MIB number 649. The MIB tree is organized into four groups; system,
-- config, alert and performance.
-- ===========================================================================

-- sap         OBJECT IDENTIFIER ::= { enterprises 649 }

-- r3          OBJECT IDENTIFIER ::= { sap 1 }

r3system    OBJECT IDENTIFIER ::= { r3 1 }
r3config    OBJECT IDENTIFIER ::= { r3 2 }
r3alert     OBJECT IDENTIFIER ::= { r3 3 }
r3perf      OBJECT IDENTIFIER ::= { r3 4 }
```

A.1 System Group

```
-- ===========================================================================
-- Th R/3 System Group
-- ===========================================================================

r3systemTable OBJECT-TYPE
                SYNTAX SEQUENCE OF R3systemEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "The (conceptual) table of active SAP R/3-Instances.
                         If an Instance is configured but not
                         started, it will not be in the list."
                ::= { r3system 1 }


r3systemEntry OBJECT-TYPE
                SYNTAX R3systemEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A (conceptual) entry for one active SAP R/3-Instance.
                         As an example, an instance of the
                         r3systemName object might be named
                         r3systemName.3"
                INDEX { r3systemNumber }
                ::= { r3systemTable 1 }
```

```
R3systemEntry ::= SEQUENCE
                        {
                         r3systemNumber        INTEGER,
                         r3systemReleaseString DisplayString,
                         r3systemName          DisplayString,
                         r3systemLogonString   DisplayString
                        }


r3systemNumber OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The r3systemNumber is an index on the
                         R/3-Instances on this Host.
                         Two-digit decimal number, Range { 00 .. 99 }. Must
                         be unique for all R3 instances on the same host.
                         Used to identify shared memories, semaphores,
                         services, ... of an R3 instance. Typically, all
                         instances of an R3 system use the same system
                         number when they are distributed over different
                         hosts. R3 instances on the same host must use
                         different system numbers in any case.
                         Management requests to currently inactive SAP R/3
                         instances will fail."
                ::= { r3systemEntry 1 }


r3systemReleaseString OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..20))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The release string of an SAP R/3 instance"

                ::= { r3systemEntry 2 }


r3systemName    OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..20))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The Name of the R3 system where this R3 instance
                         belongs to. Typically, the name of an R3 system
                         is a 3-character name in capital letters and is
                         identical to the database name of that R3 system."
```

```
                ::= { r3systemEntry 3 }

r3systemLogonString     OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..256))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The logon string for the SAP R/3 application server.
                         Typically, an application server is the active part
                         of an R3 instance which provides logon and execution
                         of R3 functionality written in the ABAP/4
                         programming language. On a UNIX system,
                         this string can be directly executed."
                ::= { r3systemEntry 4 }
```

A.2 Config Group

```
-- #============================================================================
-- #
-- # The R/3 Config Group
-- #
-- #============================================================================
```

A.2.1 r3configTable

```
r3configTable OBJECT-TYPE
                SYNTAX SEQUENCE OF R3configEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A list of R/3 Configuration entries"

                ::= { r3config 1 }


r3configEntry OBJECT-TYPE
                SYNTAX R3configEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A (conceptual) entry for one configuration
                        information data set. The r3systemNumber in the
                        index represents the entry in the r3systemTable
```

```
                        that corresponds to the r3configEntry."

                INDEX { r3systemNumber }

                ::= { r3configTable 1 }


R3configEntry ::= SEQUENCE
                        {
                         r3configSystemNumber INTEGER,
                         r3configInstanceName DisplayString,
                         r3configDBName       DisplayString,
                         r3configDBHostName   DisplayString,
                         r3configDBHostIpAdr  IpAddress
                        }

r3configSystemNumber OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The r3configSystemNumber corresponds to the
                         first index r3systemNumber."
                ::= { r3configEntry 1 }


r3configInstanceName OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..64))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "This MIB variable contains the name of the
                         R/3-Instance"

                ::= { r3configEntry 2 }

r3configDBName  OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..64))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "This MIB variable contains the
                         SAP R/3 database name"

                ::= { r3configEntry 3 }


r3configDBHostName OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..64))
                ACCESS  read-only
                STATUS  mandatory
```

```
                DESCRIPTION
                        "This MIB variable contains the name of the host
                         where the SAP R/3 database is located. May also be
                         the hostname for connecting the R3 database."

                ::= { r3configEntry 4 }

r3configDBHostIpAdr     OBJECT-TYPE
                SYNTAX  IpAddress
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "This MIB variable contains the IP address of the host
                         where the SAP R/3 database is located. May also be
                         the IP address for connecting the R3 database."

                ::= { r3configEntry 5 }

-- ===========================================================================
```

A.2.2 r3configServiceTable

```
r3configServiceTable OBJECT-TYPE
                SYNTAX SEQUENCE OF R3configServiceEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A list of R/3-Service-Types for each R/3-Instance"

                ::= { r3config 2 }


r3configServiceEntry OBJECT-TYPE
                SYNTAX R3configServiceEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A (conceptual) entry for one service
                        information data set. The r3systemNumber in the
                        index represents the entry in the r3systemTable
                        that corresponds to the r3configEntry."

                INDEX { r3systemNumber, r3configServiceType }

                ::= { r3configServiceTable 1 }


R3configServiceEntry ::= SEQUENCE
                        {
```

```
                         r3configServiceSystemNumber INTEGER,
                         r3configServiceType         INTEGER
                        }

r3configServiceSystemNumber OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The r3configServiceSystemNumber corresponds
                         to the first index r3systemNumber."
                ::= { r3configServiceEntry 1 }


r3configServiceType     OBJECT-TYPE
                SYNTAX  INTEGER {
                                dialogService          (1),
                                updateService          (2),
                                enqueueService         (3),
                                backgroundService      (4),
                                messageService         (5),
                                gatewayService         (6),
                                spoolService           (7)
                                }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "This MIB variable contains one Service offered
                         by the corresponding R/3-Instance"
                ::= { r3configServiceEntry 2 }
```

A.3 Alert Group

```
-- ==============================================================================
--
-- The R/3 Alert Group
--
-- ==============================================================================


r3alertTable OBJECT-TYPE
                SYNTAX SEQUENCE OF R3alertEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A list of R/3 Alert entries "

                ::= { r3alert 1 }
```

```
r3alertEntry OBJECT-TYPE
                SYNTAX R3alertEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "The remaining variables are needed for
                        the trap facility. When a SAP system
                        reports an alert condition, the agent
                        will send a SNMP trap to the registered
                        management station.
                        The Trap PDU contains the MIB variables defined below.
                        The r3systemNumber in the index represents the entry
                        in the r3systemTable that corresponds to the
                        r3alertEntry"

                INDEX { r3systemNumber, r3alertClass }

                ::= { r3alertTable 1 }

R3alertEntry ::= SEQUENCE
                        {
                         r3alertSystemNumber   INTEGER,
                         r3alertClass          INTEGER,
                         r3alertNumber         INTEGER,
                         r3alertAgentHostName  DisplayString,
                         r3alertSapNameString  DisplayString,
                         r3alertSeverity       INTEGER,
                         r3alertMsgString      DisplayString,
                         r3alertActionString   DisplayString,
                         r3alertTimeStamp      COUNTER
                        }

r3alertSystemNumber OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The r3alertSystemNumber corresponds
                         to the first index r3systemNumber."
                ::= { r3alertEntry 1 }


r3alertClass    OBJECT-TYPE
                SYNTAX  INTEGER {
                                  classStateChange  (1),
                                  classSAPsysUp     (2),
                                  classSAPsysDown   (3),
                                  classSlogId       (10),
                                  classSlogFreq     (11),
                                  classBuf          (12),
```

```
                                  classEnqueue     (13),
                                  classRollpag     (14),
                                  classTrace       (15),
                                  classPerfDia     (20),
                                  classPerfUpd     (21),
                                  classPerfBtc     (22),
                                  classPerfSpo     (23),
                                  classAbapUpd     (30),
                                  classAbapErr     (31),
                                  classAbapSql     (32),
                                  classDbIndcs     (41),
                                  classDbFreSp     (42),
                                  classDbArcSt     (43),
                                  classDbBckup     (44),
                                  classSpo         (51),
                                  classArch        (52),
                                  classGenP3       (53),
                                  classGenP4       (54),
                                  classGenP5       (55),
                                  classGenP6       (56),
                                  classGenP7       (57),
                                  classGenP8       (58)
                }

                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
        "Unique identifier for R3 alerts

         Class name         -  Comment
         -----------------------------------------------------------------


                               indicating state change if an instance
         classStateChange      services of an instance have changed
         classSAPsysUp         R3 instance coming up
         classSAPsysDown       R3 instance shut down

         classSlogId           critical systemlog occurred
         classSlogFreq         frequent writes into systemlog

         classBuf              local buffers alert, directory or storage
                               shortage, bad efficiency

         classEnqueue          lock facility alert, directory or storage
                               shortage, errors occurred

         classRollpag          rollfile, paging file shortage

         classTrace            trace on, may cause reduced performance

                               performance alert, exceeding wait time or
```

```
                              response time, exceeding processing time of
                              ABAP-programs
        classPerfDia          in dialog task
        classPerfUpd          in update task
        classPerfBtc          in background task
        classPerfSpo          in spool task

                              alerts in ABAP processor
        classAbapUpd          error in update processing
        classAbapErr          other error
        classAbapSql          error during database access

                              database alerts
        classDbIndcs          missing indices
        classDbFreSp          free space shortage
        classDbArcSt          archiver stuck
        classDbBckup          backup failed

                              ABAP-programmable general purpose alerts
                              GENP0..1 already used for spool and archive

        classSpo              spooler subsystem alert, exxeding time or
                              spool database shortage

        classArch             archive subsystem alert, archive-link

                              ABAP-programmable general purpose alerts
        classGenP3            reserved for further use
        classGenP4            reserved for further use
        classGenP5            reserved for further use
        classGenP6            reserved for further use
        classGenP7            reserved for further use
        classGenP8            reserved for further use                 "

                ::= { r3alertEntry 2 }

r3alertNumber   OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "Represents the alert number"

                ::= { r3alertEntry 3 }

r3alertAgentHostName OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..100))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "Represents the agent host name"
```

```
                ::= { r3alertEntry 4}


r3alertSapNameString OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..120))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "Represents the SAP R/3 server name string."

                ::= { r3alertEntry 5 }

r3alertSeverity OBJECT-TYPE
  SYNTAX        INTEGER {
                 severityNormal   (1),
                 severityWarning  (2),
                 severityCritical (3)
                }
  ACCESS        read-only
  STATUS        mandatory
  DESCRIPTION

        "Identifies the severity of an R3 alert

         Severity           -  Comment
         -----------------------------------------------------------------
         severityNormal        normal operation state, no problems
         severityWarning       problems coming up
         severityCritical      severe problems
                               for some alerts the severity is derived
                               from configurable high water marks or
                               thresholds                               "

                ::= { r3alertEntry 6 }

r3alertMsgString        OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..256))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "Represents the alert message string.
                         Gives more details of R3 alerts"

                ::= { r3alertEntry 7 }

r3alertActionString     OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..256))
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
```

```
                        "Represents the alert action string"

                ::= { r3alertEntry 8 }


r3alertTimeStamp        OBJECT-TYPE
                SYNTAX  COUNTER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "Time (UTC) of Alert Occurrance
                         UTC = time in seconds since 1970, Greenwich"

                ::= { r3alertEntry 9 }
```

A.4 Trap-Definitionen

```
-- #==========================================================================
-- #
-- # SAP R/3 traps
-- # The following are the SAP R/3 Performance-Traps.
-- #
-- #==========================================================================

r3trapStateChange TRAP-TYPE
        ENTERPRISE r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Services of the instance have changed."
        ::= 1

r3trapInstanceStartUp   TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertTimeStamp}

        DESCRIPTION
```

```
                "Inform the Network Management System if a new
                 R/3 instance is coming up."
        ::= 2

r3trapInstanceShutDown  TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertTimeStamp}
        DESCRIPTION
                "Inform the Network Management System if a
                 R/3 instance is going down."
        ::= 3

r3trapSyslogId  TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "An entry in the systemlog was written, which was configured
                 critical in the systemlog-filter."
        ::= 10

r3trapSyslogFreq        TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "A lot of entrys in the systemlog. The reason can be, that the
                 defined rate of entries (modulo) or the entries per hour went
                 beyond the defined limit."
        ::= 11

r3trapBufferSum TRAP-TYPE
        ENTERPRISE      r3config
```

```
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "The bufferperformance (hit/access) goes beyond a limiting value
                 or the size of the buffer-directory or the buffer-memory had
                 passed a defined value."
        ::= 12

r3trapEnqueue   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Enqueue space shortage or error occurred"
        ::= 13

r3trapRollPage  TRAP-TYPE
        ENTERPRISE r3config
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "The size of rollfiles or pagefiles had passed a defined value."
        ::= 14

r3trapTrace     TRAP-TYPE
        ENTERPRISE r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
```

```
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "A trace-option is active and may effect your performance."
        ::= 15

r3trapPerfDia   TRAP-TYPE
        ENTERPRISE      r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
            "You will have performance-traps, when the waiting time, the
             response time or the operating time of an ABAP-Program has passed
             a defined value. Work-Process-Typ Dialog: Performance goes
             beneath a limit."
        ::= 20

r3trapPerfUpd   TRAP-TYPE
        ENTERPRISE      r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
            "You will have performance-traps, when the waiting time, the
             response time or the operating time of an ABAP-Program has passed
             a defined value. Work-Process-Typ Update-Process: Performance
             goes beneath a limit."
        ::= 21

r3trapPerfBtc   TRAP-TYPE
        ENTERPRISE      r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
```

```
                        r3alertClass,
                        r3alertSeverity,
                        r3alertMsgString,
                        r3alertActionString,
                        r3alertTimeStamp}
        DESCRIPTION
            "You will have performance-traps, when the waiting time, the
             response time or the operating time of an ABAP-Program has passed
             a defined value. Work-Process-Typ Background: Performance goes
             beneath a limit."
        ::= 22

r3trapPerfSpo   TRAP-TYPE
        ENTERPRISE      r3perf
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
            "You will have performance-traps, when the waiting time, the
             response time or the operating time of an ABAP-Program has passed
             a defined value. Work-Process-Typ Spool: Performance goes beneath
             a limit."
        ::= 23

r3trapAbapUpd   TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Break in updateprocess was found in the ABAP-Processor."
        ::= 30

r3trapAbapError TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
```

```
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "An other error was found in the ABAP-Processor.
                 ABAP-Shortdump was written."
        ::= 31

r3trapAbapSql   TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Error during a database-access was recognized in
                 the ABAP-Processor."
        ::= 32

r3trapDbIndices TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "There are missing indices on database tables.
                 Will result in worse access time"
        ::= 41

r3trapDbFreeSpace       TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
```

```
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "The free space area in the database goes beneath a
                 critical value."
        ::= 42

r3trapDbArcStuck TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Database-archiver doesnt work!"
        ::= 43

r3trapDbBackup  TRAP-TYPE
        ENTERPRISE      r3system
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Problems during the database-backup were recognized."
        ::= 44

r3trapSpool     TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 1, used for spool"
        ::= 51
```

```
r3trapArchive   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 1, used for Archive"
        ::= 52

r3trapGenPur3   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 3."
        ::= 53

r3trapGenPur4   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 4."
        ::= 54

r3trapGenPur5   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
```

```
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 5."
        ::= 55

r3trapGenPur6   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 6."
        ::= 56

r3trapGenPur7   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
        DESCRIPTION
                "Free programmable alert 7."
        ::= 57

r3trapGenPur8   TRAP-TYPE
        ENTERPRISE      r3alert
        VARIABLES       {r3systemNumber,
                         r3alertNumber,
                         r3alertAgentHostName,
                         r3alertSapNameString,
                         r3alertClass,
                         r3alertSeverity,
                         r3alertMsgString,
                         r3alertActionString,
                         r3alertTimeStamp}
```

```
        DESCRIPTION
                "Free programmable alert  8."
        ::= 58
```

A.5 Performance Group

```
-- #==========================================================================
-- #
-- # The R/3 Performance Group
-- #
-- #==========================================================================

r3perfTable OBJECT-TYPE
                SYNTAX SEQUENCE OF R3perfEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "A list of R/3 Performance entries"

                ::= { r3perf 1 }


r3perfEntry OBJECT-TYPE
                SYNTAX R3perfEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "An entry for R/3 performance data.
                         The r3systemNumber in the index represents the
                         entry in the r3systemTable that corresponds to
                         the r3perfEntry."

                INDEX { r3systemNumber, r3perfServiceType }

                        ::= { r3perfTable 1 }


R3perfEntry ::= SEQUENCE
                        {
                         r3perfSystemNumber     INTEGER,
                         r3perfServiceType      INTEGER,
                         r3perfEventsFrequency  INTEGER,
                         r3perfResponseTime     INTEGER,
                         r3perfWaitTime         INTEGER
                        }

r3perfSystemNumber OBJECT-TYPE
```

```
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The r3perfSystemNumber corresponds to the
                         first index r3systemNumber."
                ::= { r3perfEntry 1 }



r3perfServiceType       OBJECT-TYPE
                SYNTAX  INTEGER {
                                 taskTypeDialog     (1),
                                 taskTypeUpdate     (2),
                                 taskTypeBackground (3),
                                 taskTypeSpool      (4)
                                }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
        " Performance statistics of various task types concerning the
          average wait time (queue-time), average response time
          ( response to user), average processing event per minute.

        Service Type      - Comment
        -----------------------------------------------------------------
        taskTypeDialog      dialog and rfc, remote function call
        taskTypeUpdate      update task
        taskTypeBackground  background processing, time/event scheduled
        taskTypeSpool       spool processing
                                                                        "
                ::= { r3perfEntry 2 }


r3perfEventsFrequency OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
        "Number of Events/min.
         (in task type dialog, an event is equivalent to a dialog step) "

                ::= { r3perfEntry 3 }


r3perfResponseTime OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "response time in milliseconds
```

```
                         response time = wait + process "

                ::= { r3perfEntry 4 }

r3perfWaitTime  OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "wait time in milliseconds
                         wait time for free work process"

                ::= { r3perfEntry 5 }

END
```

Abkürzungsverzeichnis

ABAP/4 Avanced Business Application Programming / 4th Generation Language

API Application Programming Interface

ARS Action Request System

ASCII American Standard Code for Information Interchange

ASN.1 Abstract Syntax Notation One (1)

BASF Badische Anilin und Soda Fabriken

CCMS Computing Center Management System

CD Compact Disk

CMIP Common Management Information Protocol

CPI-C Common Programming Interface for Communication

CPU Central Processing Unit

DB Database

DBMS Database Management System

DCM Device Communication Manager

DLL Dynamic Link Library

DOS Disk Operating System

DV Daten Verarbeitung

Dynpro Dynamisches Programm

EPI External Protocol Interface

EURIS Europäisches Informationssystem für Marketing und Vertrieb

FDDI Fiber Distributed Data Interface

GIB Generic Information Block

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IHAPI Inference Handler Application Programming Interface

IIB Icon Information Block

IMT Inductive Modeling Technology

IP Internet Protocol

ISO International Standardization Organisation

LAN Local Area Network

LU 6.2 Logical Unit 6.2

MIB Management Information Base

MNM-Team Münchner Netzmanagement Team

MO Managed Object

MSAP Management Station Access Provider

MTE Model Type Editor

ODP Open Distributed Processing

OID Object Identifier

OSI Open Systems Interconnection

PC Personal Computer

PIB Perspective Information Block

PIF Protocol Interface

QoS Quality of Service

R/3 Realtime-System, Version 3

RAM Random Access Memory

RDBMS Relational Database Management System

RFC Request for Comments

ROM Read Only Memory

RPC Remote Procedure Call

SAP Systeme Anwendungen Produkte

SGA System Global Area

SMFA Specific Management Functional Area

SMI Structure of Management Information

SNMP Simple Network Management Protocol

SQL Structured Query Language

SSAPI SpectroSERVER Application Programming Interface

TCP Transmission Control Protocol

VNM Virtual Network Maschine

WAN Wide Area Network

Literaturverzeichnis

[ABEC 95] Sebastian Abeck, “Integrationstechniken im Netzmanagement”, In *Kommunikation in verteilten Systemen*, GI/ITG–Fachtagung, Februar 1995.

[BEGA 95] Rüdiger Buck-Emden und Jürgen Galimow, *Die Client/Server Technologie des SAP-Systems R/3*, Addison-Wesley, Zweite Auflage, 1995.

[CCMS1] SAP AG, Neurottstraße 16, D-69190 Walldorf, *Computing Center Management System: Benutzerhandbuch BC120*, Januar 1995.

[CCMS2] SAP AG, Neurottstraße 16, D-69190 Walldorf, *Computing Center Management System: Product Overview*, April 1993.

[CDI 94] Control Data Institute, *Praxistrainer SAP R/3*, Markt & Technik, 1994.

[DATE 95] Christopher J. Date, *An Introduction to Database Systems*, Addison-Wesley, Sechste Auflage, 1995.

[DEMA 94] Stefan Deml und Johann Maurer, “Integration des IBM 6611 Network Processors in SPECTRUM”, Fortgeschrittenenpraktikum, Technische Universität München, Institut für Informatik, Mai 1994.

[GORA 90] Walter Gora und Reinhard Speyerer, *ASN.1 Abstract Syntax Notation One*, Datacom Verlag, 1990.

[HEAB 93] Heinz-Gerd Hegering und Sebastian Abeck, *Integriertes Netz- und Systemmanagement*, Addison-Wesley, Bonn, 1993.

[HNGU 94] Heinz-Gerd Hegering, Bernhard Neumair und Markus Gutschmidt, “Cooperative Computing and Integrated System Management — A Critical Comparision of Architectural Approaches”, Bericht 9403, Ludwig-Maximilians-Universität München, Institut für Informatik, Februar 1994.

[HNGU 95] Heinz-Gerd Hegering, Bernhard Neumair und Markus Gutschmidt, "Cooperative Computing und integriertes Systemmanagement", In *Informatik-Spektrum, Springer-Verlag*, 1995.

[HNW 95] Heinz-Gerd Hegering, Bernhard Neumair und Rene Wies, "Integriertes Management verteilter Systeme — Ein Überblick über den State-of-the Art", Bericht 9503, Ludwig-Maximilians-Universität München, Institut für Informatik, Januar 1995.

[ISO 10040] International Standardization Organization, *Information Processing Systems — Open Systems Interconnection — Systems Management Overview*, 1992.

[ISO 10746] International Standardization Organization, *Information technology — Open Systems Interconnection — Data Management and Open Distributed Processing — Basic Reference Model of Open Distributed Processing, Part 2–3.*

[ISO 7498-4] International Standardization Organization, *Information Processing Systems — Open Systems Interconnection — Basic Reference Model Part 4: Management Framework*, November 1989.

[ISO 8824] ISO, *Information Processing — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1)*, 1987.

[KELL 93] Alexander Keller, "Entwurf und Implementierung von Managementszenarien zu bestehenden Kommunikationsanwendungen", Diplomarbeit, TU München - Institut für Informatik, Mai 1993.

[KRUP 93] B. Krupczak, "UNIX Systems Management via SNMP", In *Proc. 3rd International Symposium on Integrated Network Management, San Francisco*, April 1993.

[LOHR 92] Jürgen Lohrmann, "Management der NWP-Knoten auf SNMP-Basis", Technischer Bericht, ICS GmbH, August 1992.

[LOLL 93] Jerry N. Luftman, Paul R. Lewis und Scott H. Oldach, "Transforming the Enterprise: The Alignment of Business and Information Technology Strategies", In *IBM Systems Journal, Vol. 32, No. 1*, Januar 1993.

[MAUR 95] Hans Maurer, "Entwicklung eines auf standardisierten Managementkonzepten basierenden Anwendungsmanagements am Beispiel von X.400", Diplomarbeit, TU München - Institut für Informatik, Mai 1995.

[MV RN] Calypso Software Systems, Inc., 134 South Mast Road, Goffstown, NH 03045, *MaestroVision 2.0 beta 1 Release Notes*, Oktober 1993.

[MV TECH] Calypso Software Systems, Inc., 134 South Mast Road, Goffstown, NH 03045, *MaestroVision 2.0 Technical Summary*, August 1993.

[MV USER] Cabletron Systems, Inc., Rochester, NH 03867-5005, *MaestroVision User's Guide*, März 1995, Order Number: 9031224E2.

[NERB 95] Michael Nerb, "Übernahme von Fremddaten in das System SAP R/3", Fortgeschrittenenpraktikum, Technische Universität München, Institut für Informatik, Mai 1995.

[ORA MIB] Robert Rose, *Oracle SNMP Support Reference Guide*, Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065, August 1994, Part No. A20611-1.

[R/3ADMIN] SAP AG, Neurottstraße 16, D-69190 Walldorf, *R/3 Administration: Handbuch BC020*, März 1993.

[R/3ARCHa] SAP AG, Neurottstraße 16, D-69190 Walldorf, *R/3 Systemarchitektur: Handbuch BC010*, März 1993.

[R/3ARCHb] SAP AG, Neurottstraße 16, D-69190 Walldorf, *Architektur System R/3*, März 1993.

[R/3INTEG] Dr. Uwe Hommel und Helmut Fieres, "Integration of SAP System Management into System Management Platforms", Technischer Bericht, SAP AG, Neurottstraße 16, D-69190 Walldorf, August 1994.

[R/3ORA] SAP AG, Neurottstraße 16, D-69190 Walldorf, *R/3 unter ORACLE: Handbuch BC100*, März 1993.

[R/3SYS] SAP AG, Neurottstraße 16, D-69190 Walldorf, *R/3 Systemverwaltung: Handbuch BC*, August 1993.

[R/3UNIX] SAP AG, Neurottstraße 16, D-69190 Walldorf, *R/3 in der UNIX-Umgebung: Handbuch BC090*, März 1993.

[RFC 1155] M.T. Rose und K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets", RFC 1155, Internet Activities Board, Mai 1990.

[RFC 1157] J.D. Case, M. Fedor, M.L. Schoffstall und C. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, Internet Activities Board, Mai 1990.

[RFC 1212] M.T. Rose und K. McCloghrie, "Concise MIB Definitions", RFC 1212, Internet Activities Board, März 1991.

[RFC 1213] K. McCloghrie und eds. Rose, M.T., "Management Information Base for network management of TCP/IP-based internets: MIB-II", RFC 1213, Internet Activities Board, März 1991.

[RFC 1441] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Introduction to version 2 of the Internet-standard Network Management Framework", RFC 1441, Internet Activities Board, April 1993.

[RFC 1442] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, Internet Activities Board, April 1993.

[RFC 1443] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1443, Internet Activities Board, April 1993.

[RFC 1444] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1444, Internet Activities Board, April 1993.

[RFC 1445] James M. Galvin und Keith McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Internet Activities Board, April 1993.

[RFC 1446] James M. Galvin und Keith McCloghrie, "Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1446, Internet Activities Board, April 1993.

[RFC 1447] Keith McCloghrie und James M. Galvin, "Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1447, Internet Activities Board, April 1993.

[RFC 1448] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, Internet Activities Board, April 1993.

[RFC 1449] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Transport Mappings for version 2 of the Simple

Network Management Protocol (SNMPv2)", RFC 1449, Internet Activities Board, April 1993.

[RFC 1450] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1450, Internet Activities Board, April 1993.

[RFC 1451] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Manager-to-Manager Management Information Base", RFC 1451, Internet Activities Board, April 1993.

[RFC 1452] Jeffrey D. Case, Keith McCloghrie, Marshall T. Rose und Steven Waldbusser, "Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework", RFC 1452, Internet Activities Board, April 1993.

[RFC 1514] Pete Grillo und Steven Waldbusser, "Host Resources MIB", RFC 1514, Internet Activities Board, September 1993.

[RFC 1565] Steve Kille und Ned Freed, "Network Services Monitoring MIB", RFC 1565, Internet Activities Board, Januar 1994.

[RFC 1697] David Brower, Bob Purvy, Anthony Daniel, Marc Sinykin und Jay Smith, "Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2", RFC 1697, Internet Activities Board, August 1994.

[ROSE 91] Marshall T. Rose, *The Simple Book: An Introduction to Management of TCP/IP-based internets*, Prentice Hall, 1991.

[SNMP CMU] Steve Waldbusser, "CMU SNMP Distribution, Release 1.0", Technischer Bericht, Carnegie Mellon University: Network Development, September 1989.

[SNMP MIT] James R. Davin, "The SNMP Development Kit", Relase Notes, Massacusetts Institute of Technology: Laboratory for Computer Science, Januar 1989.

[SPEC ADG] Cabletron Systems, Inc., Rochester, NH 03867-5005, *System Administrator's Guide*, August 1994, Order Number: 9030264-01 E14.

[SPEC APIa] Cabletron Systems, Inc., Rochester, NH 03867-5005, *SpectroSERVER API Developer's Guide*, April 1994, Order Number: 9030624 E9.

[SPEC APIb] Cabletron Systems, Inc., Rochester, NH 03867-5005, *SpectroSERVER API Reference*, März 1994, Order Number: 9030486 E6.

[SPEC APP] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum How to Integrate Applications with SPECTRUM*, August 1994, Order Number: 9031112 E2.

[SPEC BAS] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Basic Extensions Guide*, März 1992, Order Number: 9030453 E1.

[SPEC COM] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Command Line Interface User's Guide*, August 1994, Order Number: 9030664 E5.

[SPEC CON] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Concepts Guide*, August 1994, Order Number: 9030647 E4.

[SPEC EPI] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum MSAP/EPI Developers Guide*, Juni 1993, Order Number: 9030487 E4.

[SPEC EXT] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Extensions Integration Toolkit Developer's Guide*, Juli 1994, Order Number: 9030623 E8.

[SPEC GCR] Cabletron Systems, Inc., Rochester, NH 03867-0505, *Spectrum Global Classes Reference*, März 1994, Order Number: 9030665 E6.

[SPEC GEN] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Management Module Guide for Generic SNMP*, September 1994, Order Number: 9030856 E3.

[SPEC GIB] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Gib Editor Guide*, August 1994, Order Number: 9030660 E7.

[SPEC IIB] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Icon Information Block (IIB) Editor Guide*, Juli 1994, Order Number: 9030724 E5.

[SPEC INF] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Inference Handler Developer's Guide*, Dezember 1992, Order Number: 9030489 E5.

[SPEC KBG] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Knowledge-Base Guide*, August 1994, Order Number: 9030663 E1.

[SPEC MTE] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Model Type Editor Guide*, August 1994, Order Number: 9030659 E7.

[SPEC PAR] Cabletron Systems, Inc., Rochester, NH 03867-0505, *Spectrum VnmParmBlock Reference Guide*, März 1994, Order Number: 9030622 E7.

[SPEC SOF] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum Software Developer's Toolkit Overview*, August 1994, Order Number: 9030621 E4.

[SPEC USE] Cabletron Systems, Inc., Rochester, NH 03867-5005, *System User's Guide*, August 1994, Order Number: 9030263-02 E13.

[SPEC VIE] Cabletron Systems, Inc., Rochester, NH 03867-5005, *Spectrum View Application Programming Interface Developer's Guide*, März 1994, Order Number: 9030491 E5.

[SPEC WAT] Cabletron Systems, Inc., Rochester, NH 03867-5005, *SpectroWATCH User's Guide*, Juli 1994, Order Number: 9030919 E4.

[STRO 91] Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, Zweite Auflage, 1991.

[SZ 92] Alexander Schill und Martina Zitterbart, "Application- and Network-Management in Open Distributed Processing Environment", In *IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, International Federation of Information Processing (IFIP), Oktober 1992.

[WILL 94] Keith Willets, "Service Management – The Drive for Reengineering", In *IEEE/IFIP Network Operations and Management Symposium, Orlando*, März 1994.

[WOER 92] Oliver Wörl, "Enterprise Management für das EURIS Projekt der BASF AG", Technischer Bericht, ICS GmbH, August 1994.

[WW SPEC] Metrix s.a., 37 Val St. André, L-1128 Luxembourg, *Spectrum Management Module Guide for WindowsPC, Version 1.0*, 1994.

[WW USER] Metrix s.a., 37 Val St. André, L-1128 Luxembourg, *WinWatch User Guide Version 1.0*, 1994.

[ZIMM 92] Martin Zimmermann, "Management of Distributed Applications", In *IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, IFIP, 1992.