

JavaTMmagazin

Internet & Enterprise Technology

Auf Magazin-CD:

→ S. 3

- Tools, Testversionen & Samples!
- Video: Eclipse-Night mit Erich Gamma von der JAX 2004



ORACLE[®]

Mit Extra-CD → S. 27
Oracle JDeveloper 10g

w-jax 2004

15.-18.11.04 • www.wjax.de

Java & Flash

Flash-Clients für J2EE-Anwendungen

J2EE Management

Maximale Kontrolle von Enterprise-Systemen

mit 2 CDs!

www.javamagazin.de

TomC@ – die Kolumne
Was ist neu im Tomcat 5.5?

Oracle ADF
Java Data Binding an Bord

Telefonie mit Java
Call Center und Java-Anwendungen

JavaServer Faces
Eigene Komponenten erstellen

**XML
extra** included

XML 1.1

... und die umstrittenen Neuerungen

D 45867



12

4 194586 706503

IT-Management: Theorie und Umsetzung am Beispiel des J2EE-Managements

von Roger Zacharias

Die Spitze des Eisbergs

Eine alltägliche Situation: Bei der Arbeit mit Windows bemerkt man, dass die CPU-Auslastung ohne ersichtlichen Grund 100 Prozent beträgt. Daraufhin öffnet man den Task-Manager, um den CPU-Verbrauch der Prozesse zu kontrollieren und gegebenenfalls einen dieser Prozesse zu beenden. Nun stelle man sich ein Produktivsystem vor, welches im 5-Knoten-Cluster läuft, einen vorgeschalteten Apache als Load Distributor nutzt, für die Persistenz ein OODBMS verwendet, auf Legacy-Daten über ein Tuxedo-Gateway zugreift und natürlich eine Verfügbarkeit von 24 x 7/99,9 Prozent haben soll.



Im obigen Fall sollte Folgendes gewährleistet sein: Der Zustand des Systems sollte von einem zentralen Punkt aus überwachbar sein, das System sollte über problematische Situationen automatisch informieren und die Quelle des Problems muss schnell und auch effizient auffindbar sein. Weiterhin sollte der Zugriff auf die Ressourcen geregelt und überwacht geschehen und es muss möglich sein, bestimmte Probleme zu beheben, ohne das System herunterfahren zu müssen. Diese Anforderungen werden durch IT-Management-Systeme abgedeckt, welche parallel zu den operativen Systemen betrieben werden und insbesondere in der heutigen Zeit für die Überwachung und Administration der komplexen Systemlandschaften eine immer wichtigere Rolle spielen. Dies reicht von einfachen

Webanalysesystemen, welche die durchschnittliche Verweildauer eines Nutzers auf einer B2C-Seite analysieren, bis hin zu komplexen Enterprise-Management-Systemen, die innerbetriebliche Prozesse und Organisationseinheiten analysieren und daraus Zielvorgaben für die IT-Infrastruktur ableiten.

Im ersten Teil dieses Artikels soll die grundlegende Theorie des IT-Managements (Klassifikation, FCAPS, Architekturen etc.) dargestellt werden und im zweiten eine konkrete Umsetzung dieser Theorie auf den Bereich des J2EE-Managements (JSR 77) erfolgen.

Definition

Unter dem Begriff IT-Management versteht man sowohl die Überwachung als auch

die Administration beliebiger Ressourcen mithilfe der IT. IT-Management-Systeme sind hierbei lediglich ein unterstützendes Hilfsmittel, da es sich beim Management von Ressourcen in erster Linie um eine organisatorische Aufgabe handelt, d.h., das IT-Management wird in erster Linie durch z.B. Administratoren mithilfe von speziellen Werkzeugen realisiert.

Klassifikation des IT-Managements nach Ressourcen

Man kann das IT-Management, wie in Abbildung 1 dargestellt, vertikal nach den zu überwachenden und administrierenden Ressourcen klassifizieren:

- Netzwerk-Management: Diese Form des Managements befasst sich mit der Ver-

waltung von Netzdiensten und -komponenten, d.h. der untersten Ebene eines Netzwerks. Dazu gehören Komponenten wie Router, Switches, Hubs, Repeater, aber auch die Leitungen, über die die Kommunikation stattfindet.

- **System-Management:** Im System-Management werden nicht wie im Bereich des Netzwerk-Managements Netzkomponenten, sondern die Endsysteme der Rechnernetze verwaltet. Hierzu gehören Komponenten wie PCs oder Automaten. Die Aufgabe besteht hier u.a. darin, die verteilt erbrachten Systemfunktionen so zu organisieren, dass vernetzte Einzelsysteme nach außen hin wie ein (virtuelles) Gesamtsystem erscheinen.
- **Informations-Management:** Diese Form des Managements umfasst alle Management-Aktivitäten in Bezug auf unternehmensweite Datenbestände, deren Entwurf, Pflege und konsistente Haltung. D.h., hier findet eine Überwachung der Datenbanken, Repositories, Backup-Systeme, Archive usw. statt.
- **Anwendungs-Management:** Auch Applikationen und IT-Dienste erfordern Management-Aufgaben, welche mithilfe eines Management-Systems gelöst werden müssen. Beispiele hierfür sind Steuerungssysteme zur automatischen Fertigung oder Systeme zur Überwachung und zur Administration von Automatenetzen.
- **Enterprise-Management:** Das Enterprise-Management hat zur Aufgabe, die Teilbereiche des Managements im Kontext des gesamten Unternehmens zu sehen. D.h., hier werden die Aufgaben der einzelnen Unternehmensbereiche zusammengefasst und daraus Zielvorgaben für die IT-Infrastruktur abgeleitet.

Abb. 1: Vertikale Klassifikation

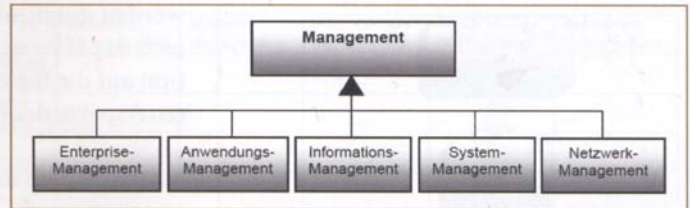
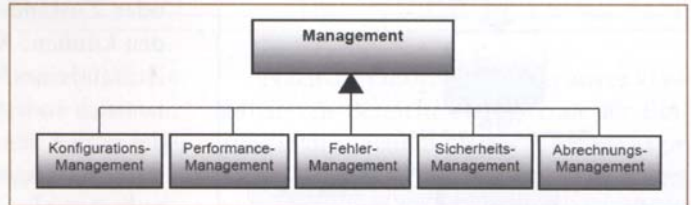


Abb. 2: Horizontale Klassifikation



gabe, einen Fehler (Abweichung von der Spezifikation) möglichst schnell zu erkennen und zu lokalisieren, um diesen mit geeigneten Mitteln zu beheben.

- **Konfigurations-Management:** Die Aufgaben des Konfigurations-Managements liegen in der Bestandsaufnahme und Inventarisierung der zu überwachenden Ressourcen. Hierzu muss zunächst eine Datenbank (MIB = Management Information Base) erstellt werden, in der jede Ressource mit den zugehörigen Parametern gespeichert ist. Eine weitere Aufgabe des Konfigurations-Managements liegt darin, die durch Performance- und Fehler-Management entdeckten Schwachstellen durch entsprechende Umkonfiguration der Ressourcen zu beseitigen. Ebenso ist es, je nach Dynamik der zu überwachenden Ressource, notwendig, die Parameter den Umständen entsprechend oft zu ändern.
- **Abrechnungs-Management:** Die Aufgabe des Abrechnungs-Managements besteht darin, die Kosten für in Anspruch

genommene Ressourcen zu ermitteln, sodass diese auf Benutzer oder Kostenstellen im Unternehmen gerecht verteilt werden können. Dazu muss jeder Zugriff auf das Netz bzw. die Ressourcen protokolliert und ausgewertet werden.

- **Performance-Management:** Bei dieser Form des Managements geht es um die Überwachung der Leistungsfähigkeit der Ressourcen. Performanceprobleme können entdeckt und durch das Konfigurations-Management behoben werden. Dadurch kann die Effizienz der Systeme erhöht werden, sodass mit vorhandenen Mitteln ein Maximum an Performance erzielt werden kann.
- **Sicherheits-Management:** Diese Form des Managements ist für den Schutz der Daten vor unberechtigten Zugriffen zuständig. Zum einen können durch unbeabsichtigte oder mutwillige Aktionen Modifikationen und Fehler im System hervorgerufen, zum anderen können durch Angriffe vertrauliche Informationen ausgespäht oder manipuliert werden. Sowohl

Anzeige

Das professionelle Präsentationsframework für J2EE™ Applikationen mit HTML-Frontends.
Optimiert für Apache Struts.

Mit Listen, Formularen, Bäumen, Menüs, TabSets zur schnellen Erstellung von komplexen und anspruchsvollen Benutzeroberflächen.

Common Controls
www.common-controls.com

Klassifikation des IT-Managements nach Funktionen

Senkrecht auf den klassifizierten Management-Formen kann eine horizontale Klassifikation nach Funktionen vorgenommen werden (Abb. 2):

- **Fehler-Management:** Mit zunehmender Größe und Komplexität eines Systems wird die Fehlererkennung immer schwieriger und gleichzeitig bedeutender. Daher hat das Fehler-Management die Auf-

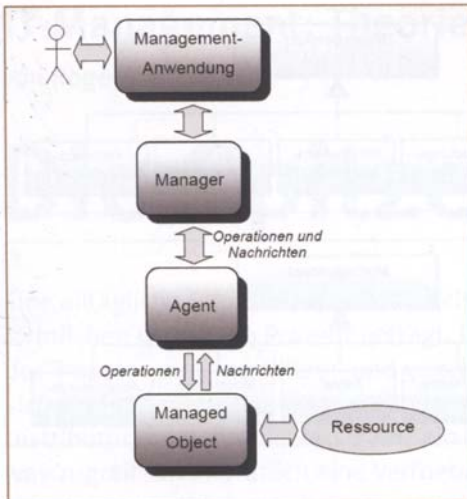


Abb. 3: Architektur eines IT-Management-Systems

Modifikationen und Fehler als auch Angriffe gilt es so schnell wie möglich zu erkennen und entsprechende Gegenmaßnahmen einzuleiten.

Diese Funktionen werden in der Literatur oft als FCAPS zusammengefasst, was für Fault-, Configuration-, Accounting-, Performance- und Security-Management steht.

Architektur

Trotz der Vielfalt der Klassifikationsmöglichkeiten und existierenden Produkte im IT-Management-Bereich basieren alle Lösungen auf einer grundlegenden Architektur. Diese gemeinsame Architektur definiert vier wesentliche Komponenten eines Management-Systems (Abb. 3): Management-Anwendung/Console, Manager und Agent, Managed Object. Die verwalteten logischen und physikalischen Ressourcen

werden durch so genannte Managed Objects repräsentiert, welche eine Abstraktion auf die für das Management relevanten Aspekte der Ressource darstellen:

- Attribute, in denen Management-Informationen wie zum Beispiel Parameter oder Zustandsgrößen gespeichert werden können. Attribute beschreiben den Zustand eines Managed Objects. Sie können sich ändern, wenn sich der Zustand des realen Pendant ändert oder gezielt durch Management-Operationen manipuliert wird.
- Operationen, welche auf dem Managed Object und ggf. auf der überwachten Ressource selbst ausgeführt werden können, wie zum Beispiel ein Geräte-Reboot. Operationen ermöglichen den Zugriff auf ein Managed Object, sodass die Menge der möglichen Operationen die Leistungsfähigkeit und Komplexität des Managed Objects beeinflusst. Typische Operationen sind zum Beispiel das Lesen und Schreiben von Parametern.
- Verhalten, mit dem das Managed Object auf die auszuführenden Operationen reagiert. Es bestimmt die Semantik sowie die Interaktionen mit dem realen Pendant und wird meist verbal definiert. Beim Verhalten wird zwischen internen und externen Stimuli unterschieden. Interne Stimuli sind Vorgänge, die direkt zwischen der Ressource und dem Managed Object ablaufen, externe Stimuli treten auf, wenn Management-Operationen auf das Managed Object wirken und dieses daraufhin durch definierte Verhaltensweisen reagiert.

- Nachrichten, mit denen ein Managed Object über den Eintritt eines Ereignisses informieren kann. Diese werden in der Regel durch Statusänderungen wie zum Beispiel Schwellwert-Überschreitungen ausgelöst.

Mit diesem Hintergrund kann man jede Ressource aus zwei Sichten betrachten (Abb. 4): Die erste Sicht ist die Service-sicht, d.h., welche Funktion hat die Ressource bzw. welchen Dienst erbringt sie. Die zweite Sicht ist die Management-Sicht, die Management-Informationen über die Ressource liefert und über die Management-Operationen auf der Ressource ausgeführt werden können. Nehmen wir als Beispielressource eine SAP-R/3-Debitoren-Datenbank. Als Bestandteil der Service-sicht können die typischen servicespezifischen CRUD-Operationen (*Create, Read, Update, Delete*) auf dieser Datenbank ausgeführt werden. Die Management-Sicht hingegen liefert uns zum Beispiel Informationen über TX/s oder die Anzahl der Tabellen. Weiterhin ermöglichen uns Management-Operationen die Konfiguration für optimale Datenbank-Performance anzupassen.

So existiert in der Regel für jede zu überwachende Ressource ein Managed Object, je nach Granularität kann die Anzahl der Managed Objects für ein zu überwachendes Gesamtsystem entsprechend hoch sein. Die Gesamtheit der Management-Informationen, die in und durch die Managed Objects bereitgestellt wird, wird als Management Information Base (MIB) bezeichnet.

Diese Management-Anwendung stellt das Frontend zum menschlichen Part des Managements, also einem Administrator oder Operator, dar. Es handelt sich hierbei meist um eine mehr oder weniger komplexe grafische Benutzeroberfläche, von der aus das gesamte System überwacht, gesteuert und manipuliert werden kann.

Der Manager nimmt Anfragen und Aufträge der Management-Anwendung entgegen und leitet diese an den Agenten weiter. Auf der anderen Seite erhält er vom Agenten sowohl Antworten auf diese Operationen als auch asynchrone Meldungen, welcher er an die Management-Anwendung weiterleitet. Der Agent verwaltet die Managed Objects und damit verbunden

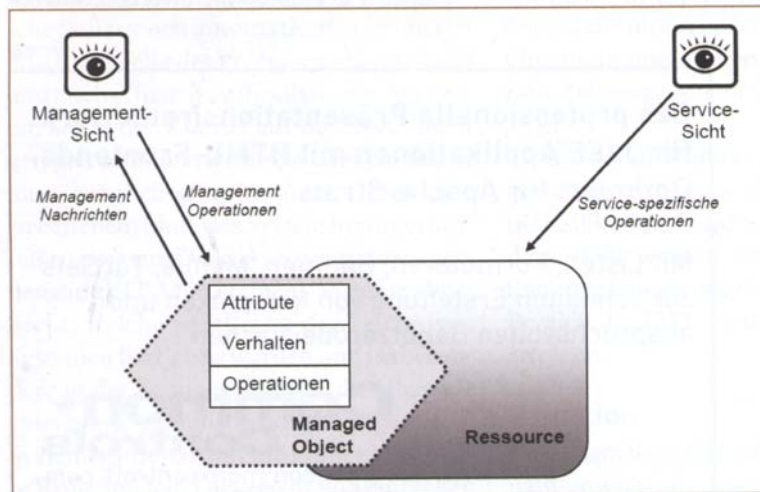


Abb. 4: Sichten auf Ressourcen

die MIB und dient damit als Repräsentant des zu überwachenden Systems und als Vermittler zwischen Manager und Managed Objects. Sowohl Anzahl als auch Relationen der beschriebenen Komponenten sowie deren Verteilung differieren in den verschiedenen konkreten IT-Management-Architekturen sowie IT-Management-Produkten.

Standards & Entwicklungen

Bei der Betrachtung des Themengebiets IT-Management in Bezug auf seine technologische Basis lassen sich drei grundsätzliche Entwicklungslinien unterscheiden:

- die Management-Infrastruktur um das Common Management Information Protocol (CMIP) als Bestandteil des ISO/OSI-Referenzmodells, welche sehr mächtig und theoretisch fundiert ist, aber aufgrund ihrer Komplexität wenig Verbreitung gefunden hat.
- die Management-Infrastruktur um den Standard der TCP/IP-Welt, das Simple Network Management Protocol (SNMP), welches ein weites Einsatzfeld vorweisen kann, aber gewisse konzeptionelle Schwachstellen birgt.
- neuere Ansätze, wie CORBA-basierte Lösungen, WBEM (Web-based Enterprise Management) und die Java Management Extensions (JMX), die flexible, dynamische, skalierbare, integrierte und webfähige Management-Systeme versprechen.

Im Folgenden soll die auf JMX basierende J2EE-Management-Architektur als Beispiel für eine IT-Management-Infrastruktur vorgestellt werden.

J2EE Management (JSR 77)

Natürlich ist ein Management auch oder insbesondere im J2EE-Bereich für geschäftskritische Anwendungen absolut notwendig. Dies geschah bisher allerdings nur mehr oder weniger ausgereift und mit proprietären Technologien und Werkzeugen der Serverhersteller. Jeder Hersteller verfolgte hier eine andere Strategie und die jeweiligen Werkzeuge unterschieden sich zusätzlich noch von Version zu Version.

Seit J2EE 1.4 existiert für das J2EE Management ein obligatorischer übergreifender Standard – die J2EE Management

Abb. 5: Einordnung des J2EE Management

	Network Management	System Management	Informations Management	Anwendungs Management	Enterprise Management
Konfigurations Management				X	
Performance Management				X	
Fehler Management				X	
Sicherheits Management					
Abrechnungs Management					

1.0 Specification, welche jeder J2EE 1.4-konforme Application Server unterstützen muss. Dieser wurde im Rahmen des JSR 77 seit August 2000 entwickelt.

Will man das J2EE Management in die im ersten Teil dargestellten Klassifikationsebenen einordnen, findet es sich im Anwendungs-Management mit den Funktionen Fehler-, Konfigurations- und Performance-Management wieder (Abb. 5).

Das heißt, der Fokus liegt hier klar auf dem Management der J2EE-Anwendungskomponenten (Servlets, EJBs etc.) und der von diesen benötigten Server-Systemressourcen (JVM, JDBC-Treiber, Connection-Factories etc.). Vermutlich werden die horizontalen Funktionsbereiche des Sicherheits- und Abrechnungs-Managements in späteren Versionen der Spezifikationen hinzukommen.

Nach der Einordnung in die zuvor klassifizierten Bereiche erfolgt nun der Einstieg in die Architektur des J2EE Management. Der JSR 77 spezifiziert ein Management-Information-Modell für die J2EE-Plattform, d.h., es handelt sich nicht direkt um eine Implementierung, sondern um ein Modell und ein Meta-Modell, die festlegen, wie die Management-Daten dem Client präsentiert werden, welche Managed Objects existieren (siehe MIB) und wie die Beziehungen und Kardinalitäten zwischen diesen aussehen. Dieses Modell basiert auf den Java Management Extensions (JMX) [3]. JMX bietet letztendlich die Infrastruktur, sagt aber nichts über die zu überwachenden Objekte und Dienstschnittstellen für externe Clients aus. Der JSR 77 wendet das JMX-Rahmenwerk an, spezifiziert die zu überwachenden Ressourcen im J2EE-

Anzeige

Ein Hinweis für alle J2ME Entwickler: Mit J2ME Polish optimieren und beschleunigen Sie Ihre Entwicklung!

- **Device-Optimierung**
Unterstützen Sie mit einem Projekt beliebig viele Geräte und reizen Sie deren Fähigkeiten aus
- **Design per CSS**
Designer können unabhängig vom Entwickler arbeiten
- **Ant basiert**
Alle IDEs werden unterstützt
- **Open Source**
GPL und kommerzielle Lizenz verfügbar
- **100% kompatibel**
Keine Änderungen am Source-Code notwendig, kaum Lernaufwand



Testen auch Sie J2ME Polish jetzt unverbindlich!
Auf der Heft-CD oder www.enough.de



SOFTWARE for professionals. www.enough.de

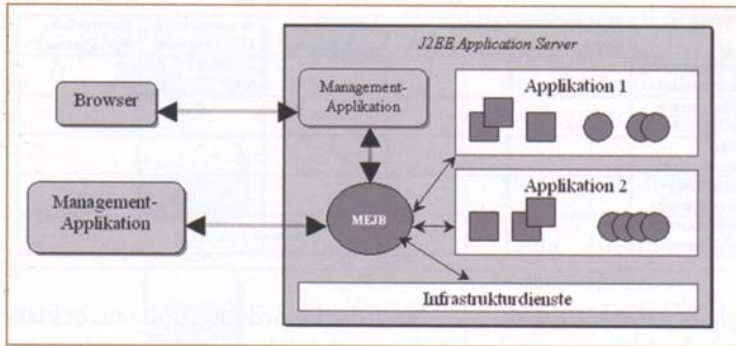


Abb. 6: Das J2EE Management EJB

Kontext und bietet zudem verschiedene Dienstschnittstellen, wie RMI/IIOP, SNMP und CIM-Protokolle.

Bildet man die in Abbildung 3 dargestellte allgemein gültige Architektur eines Management-Systems auf das J2EE-Management-System und hierbei auf das Java-Objektmodell ab, so ergibt sich folgendes Mapping (Abb. 6): Die Managed Objects, welche die eigentlichen Ressourcen repräsentieren – und weiter unten detaillierter betrachtet werden –, werden durch spezifische Java-Objekte (ähnlich JMX MBeans) abgebildet. Der Agent, welcher die Managed Objects einer Domäne verwaltet und damit in Bezug auf den Aspekt

des Managements als Repräsentant dieses Systems agiert, wird durch ein spezifisches und zentrales EJB realisiert. Diese zentrale Komponente wird als J2EE Management EJB (MEJB) bezeichnet und in Zukunft durch den Application-Server-Hersteller zur Verfügung gestellt, idealerweise direkt deploy und ihr Home Interface unter *ejb/mgmt/MEJB* registriert. Die Konzeption und Implementierung des Managers und der Management-Anwendung/-Konsole sind nicht spezifiziert und bleiben damit dem Serverhersteller oder einem Tool-Hersteller überlassen. Der Zugriff auf den Agent erfolgt also wie gewohnt über J2EE-Mechanismen (EJB Lookup). Ist der *RemoteInterface-Stub* erst einmal beim Client, kann dieser über das MEJB Managed Ob-

jects auffinden und auf diesen operieren (siehe nebenstehenden Kasten).

Das Konzept der Ereignisverarbeitung ähnelt dem JMX-Notification-Konzept beziehungsweise dem SNMP-Trap-Konzept. Managed Objects können Ereignisse versenden und Clients des Agenten können sich für Ereignisse registrieren, ohne dass ein Polling notwendig ist. Die Übermittlung des Ereignisses an den Client ist nicht spezifiziert, d.h., dieser Mechanismus ist proprietär und hängt von der serverseitigen Implementierung ab (z.B. RMI oder JMS).

Welche Managed Objects existieren nun, d.h., wie sieht die MIB für das J2EE Management aus? Das J2EE Management Information Model besteht aus folgenden Teilmodellen:

- Managed Object Model
- Event Provider Model
- State Management Model
- Performance Data Model

Hierbei spezifiziert das Managed Object Model die zu überwachenden und administrierenden Ressourcen. Das Event Provider Model spezifiziert Ressourcen, welche in der Lage sind, Ereignisse (z.B. über

Arbeit mit dem MEJB

Auffinden des MEJB Home und Erzeugen des MEJB:

```
Context context = new InitialContext();
Object objref = context.lookup("ejb/mgmt/MEJB");
ManagementHome home = (ManagementHome)
    PortableRemoteObject.narrow(objref,
        ManagementHome.class);
Management mejb = home.create();
```

Auffinden aller Managed Objects vom Typ *J2EEApplication*:

```
ObjectName searchPattern = new
    ObjectName("*:type=J2EEApplication,*");
Set managedObjects = mejb.queryNames(searchPattern);
System.out.println("Found " + managedObjects.size()
    + " matching Managed Objects");
```

Attribute eines Managed Object auslesen:

```
ObjectName app = managedObjects.iterator().next();
boolean isStateManageable = mejb.getAttribute
    (app, "StateManageable");
System.out.println("Object " + app + " is state
    manageable:" + isStateManageable);
```

Registrierung eines lokalen Event Listener:

```
System.out.println("addNotificationListener to: " + app);
ListenerRegistration reg = mejb.getListenerRegistry();
reg.addNotificationListener(app, listener, null,
    "MEJBTester");
```

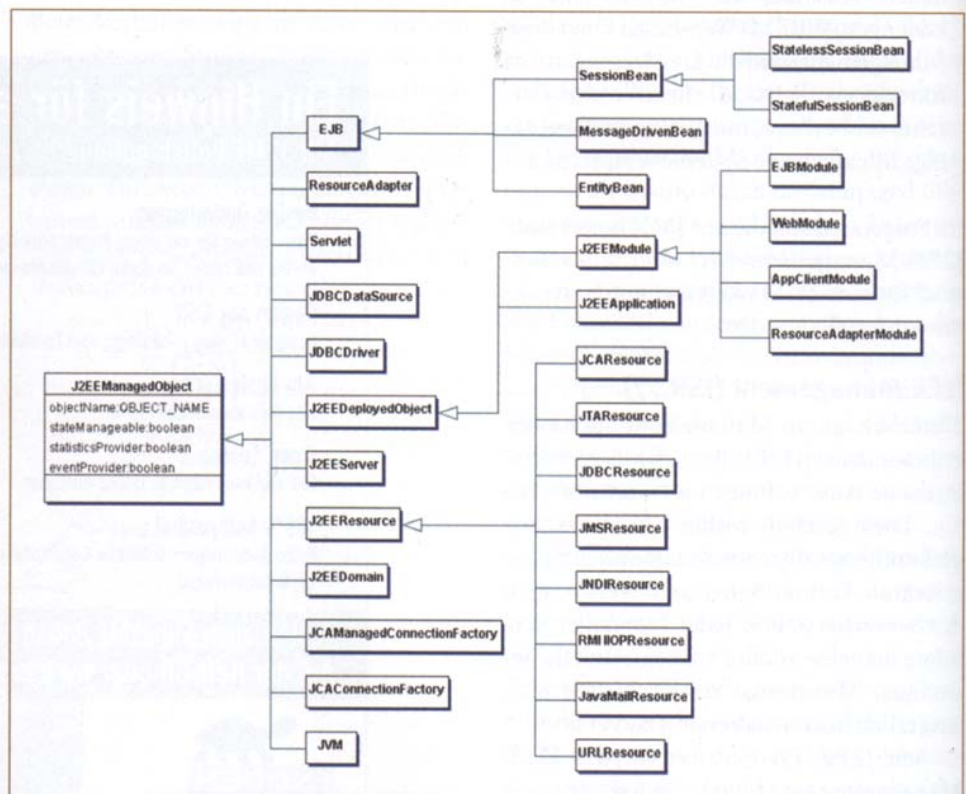


Abb. 7: J2EE Managed Objects (Quelle: [4])

Fehler, Start, Erzeugung etc.) zu versenden. Das State Management Model spezifiziert die Anforderungen an Ressourcen, die Statusmanipulationen ermöglichen. Hierzu zählen insbesondere das Starten und Stoppen von Ressourcen. Das Performance Data Model spezifiziert die Anforderungen an Ressourcen, die Performancedaten zur Verfügung stellen können. Hierzu zählen Statistiken bzgl. Zeiten, Zählern, Ranges, Thresholds etc.

Dieses Management Information Model ist auf andere Modelle wie CIM/WBEM oder SNMP abbildbar, um ein Mapping an der Dienstschnittstelle und die Anbindung an fremde Management-Systeme zu vereinfachen. Der Aufbau eines Managed Object stellt sich sehr einfach dar: Es hat die Attribute Domain-Name, Typ, Parent-Typ, eine Liste von Key/Value-Paaren und drei boolesche Werte, welche spezifizieren, ob dieses Managed Object das Event, State und/oder Performance Model implementiert.

Abbildung 7 stellt die spezifizierten Managed Objects dar. Hierzu zählen neben der J2EE-Domäne, den J2EE-Servern und der JVM als Repräsentanten der Plattform natürlich auch die Komponenten einer aufsetzenden Anwendung (Servlets, EJBs etc.). Ein J2EE Server Managed Object hat zum Beispiel folgende Attribute: Liste aller deployten Applikationen und Module, Liste aller diesem Server zugänglichen Ressourcen, Liste aller JVMs auf diesem Server sowie Informationen zum Hersteller und zur Serverversion. Abschließend sollen in einem Überblick (Abb. 8) die Beziehungen und Kardinalitäten der Managed Objects dargestellt werden.

Die Definition eines einheitlichen und übergreifenden Standards im Bereich des J2EE Management hat folgende Vorteile:

- Application-Server-Hersteller brauchen kein spezielles proprietäres Management-Werkzeug entwickeln, sondern können sich am JSR-77-Standard orientieren.
- Tool-Provider können ihre Management-Werkzeuge an beliebige Serverhersteller verkaufen oder getrennt vermarkten.
- Serveradministratoren brauchen sich nicht mehr in verschiedenste proprietäre Management-Modelle und -werkzeuge einarbeiten.

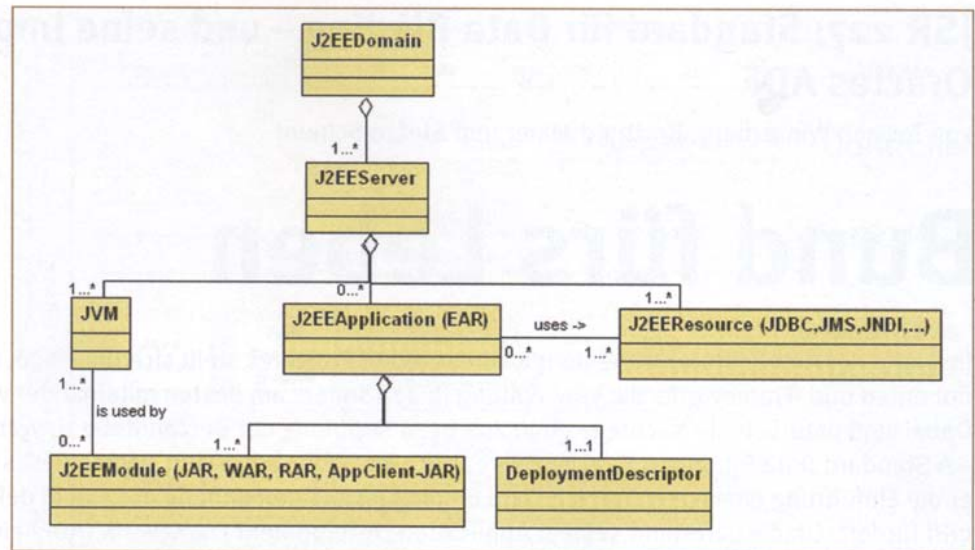


Abb. 8: Beziehungen und Kardinalitäten der MOs (Quelle: [4])

- Jeder standardkonforme Management-Client kann sich mit jedem standardkonformen Application Server verbinden.
- Eine einzelne Management-Plattform kann gleichzeitig mehrere J2EE-Server verschiedenster Hersteller managen.

Das J2EE Management Information Model kann und wird in Zukunft durch die Hersteller erweitert, indem diese zusätzliche Informationen wie eigene Managed Objects, Attribute, Ereignisse etc. hinzufügen. Der Client wird durch diese Erweiterungen nicht beeinträchtigt, er kann sie nutzen, muss dies aber nicht.

Fazit

Unter IT-Management versteht man die Überwachung und Administration beliebiger Ressourcen mithilfe der IT. Dieses Gebiet kann sowohl vertikal nach Ressourcen als auch horizontal nach Management-Funktionen klassifiziert werden, wobei allen IT-Management-Lösungen eine gemeinsame Architektur zugrunde liegt. Die Bedeutung des IT-Managements steigt proportional zur Bedeutung der IT für die Unternehmen selbst. Die sinnvolle Nutzung der neu entstehenden Systeme erfordert bei gleichzeitig wachsender Komplexität daher ein adäquates Management, da das sichere Funktionieren der Systeme und die schnelle Anpassbarkeit an geänderte Rahmenbedingungen zunehmend unternehmenskritische Bedeutung erlangen. Daher ist ein Management auch oder insbe-

sondere im J2EE-Bereich für geschäftskritische Anwendungen absolut notwendig. Die durchgeführte Standardisierung des J2EE-Management bringt Vorteile für alle „Betroffenen“ mit sich. Auch wenn diese Spezifikation momentan noch jung und unvollständig ist, ist zu erwarten, dass auch dieser Standard mit der J2EE-Plattform gemeinsam reift.

Es handelt sich beim Thema IT-Management um ein sehr umfassendes Thema, von welchem dieser Artikel lediglich die Spitze des Eisbergs in Form eines groben Überblicks darstellen kann. Es lohnt sich aber auf jeden Fall, sich tiefer in diese interessante Materie (siehe hierzu auch ITIL, Balanced Scorecard (BSC) etc.) einzuarbeiten.

Dipl.-Inf. (FH) Roger Zacharias ist Sun Certified Enterprise Architect und als Software Engineer bei Wincor Nixdorf beschäftigt. Die Schwerpunkte seiner Arbeit bilden Architektur, verteilte Systeme und insbesondere IT-Management-Systeme. Als JCP-Mitglied beteiligt er sich aktiv an der Weiterentwicklung der Java-/J2EE-Plattform.

Links & Literatur

- [1] Marshall T. Rose: The Simple Book, Prentice-Hall, 1991
- [2] William Stalling: SNMP, SNMPv2 and CMIP, Addison-Wesley, 1993
- [4] JSR 3 – JMX: www.jcp.org/en/jsr/detail?id=3
- [5] JSR 77 – J2EE Management: www.jcp.org/en/jsr/detail?id=77