Vorlesung

Middleware

Winter 2005/2006

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Inhalt

A Organisatorisches . . . A.2
A.1 Dozent . . . A.2
A.2 Übungsbetreuung . . . A.2
A.3 Inhalt . . . A.3
A.4 Vorlesung . . . A.4
A.5 Übungen . . . A.6
A.6 Prüfungen, Scheine . . . A.7

B Überblick . . . B.1
B.1 Objektorientierte Programmierung . . . B.1
B.2 Verteilte Systeme und Verteilte Objekte . . . B.2
B.3 Objektorientierte Komponentenmodelle . . . B.2
B.4 Design Patterns für nebenläufige und verteilte Anwendungen . . . B.2
B.5 Middleware und verteilte Anwendungen . . . B.3
B.6 Peer-to-Peer-Architekturen . . . B.4
B.7 WebServices . . . B.4
B.8 Grid-Architekturen . . . B.4
B.9 Verteilte Betriebssysteme . . . B.4

C Objektorientierte Programmierung . . . C.1
C.1 Überblick . . . C.1

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

C.2 Literatur . . . C.2
C.3 Motivation für das objektorientierte Paradigma . . . C.3
1 Ziele . . . C.3
C.4 Software-Design Methoden . . . C.4
1 Einordnung nach Booch (aus [Boo94]) . . . C.4
2 Klassen von Programmiersprachen . . . C.4
3 Top-down Structured Design
(Composite Design)C.5
4 Objektorientiertes Design . . . C.7
C.5 Objektorientierte Programmierung . . . C.10
1 Definition (Grady Booch) . . . C.10
2 Grundbegriffe . . . C.11
3 Objekte & Methoden . . . C.12
4 Klassen . . . C.13
5 Vererbung . . . C.14
6 Vererbung in C++ . . . C.19
7 Dynamisches Binden . . . C.20
8 Statisches Binden . . . C.22
C.6 Fundamentale Konzepte des objektorientierten Paradigmas . . . C.23
1 Abstraktion . . . C.24
2 Kapselung . . . C.25
3 Modularisierung . . . C.27
4 Hierarchie . . . C.28
5 Typisierung . . . C.29
6 Typhierarchie . . . C.30
7 Polymorphismus . . . C.31
8 Polymorphismus in C++ . . . C.32
9 Typen und C++: Abstrakte Klassen . . . C.35

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

10 Typen und Java: Interfaces . . . C.36
11 Generizität (Genericity) . . . C.38
12 Generizität und C++: Templates . . . C.39
13 Generizität und Java (ab Java 5) . . . C.40
14 Nebenläufigkeit (Concurrency) . . . C.41
15 Nebenläufigkeit und Java . . . C.42
16 Persistenz . . . C.45
C.7 Objektorientierte Software-Entwicklung . . . C.47
1 Objektorientiertes Software-Engineering . . . C.48
2 Warum objektorientiertes Software-Engineering? . . . C.49
3 Phasen der Software-Entwicklung . . . C.52
C.8 Objektorientierte Analyse . . . C.56
1 Der Prozess . . . C.57
2 OOA — Anforderungsanalyse . . . C.58
3 OOA — Beispiele einer Anforderungsanlyse . . . C.59
4 OOA — Use Cases . . . C.60
5 OOA — Objekte finden . . . C.62
6 OOA — Objekte organisieren . . . C.63
7 OOA — Beschreibe Interaktionen . . . C.71
8 OOA — Struktur verfeinern . . . C.74
9 OOA - OOD? . . . C.75
C.9 Objektorientiertes Design . . . C.76
1 Phasen . . . C.77
C.10 OOA / OOD - Zusammenfassung . . . C.78

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

C.11 Entwurfsmuster (Design Patterns) . . . C.79
1 Beispiel: Smalltalk's Model/View/Controller . . . C.80
2 Elemente eines Entwurfsmusters . . . C.84
3 Design Pattern Space . . . C.85

D Verteilte Systeme und Verteilte Objekte . . . D.1
D.1 Überblick . . . D.1
D.2 Verteilte Systeme . . . D.2
D.3 Kommunikationsmodelle . . . D.3
1 Protokollschichten nach dem ISO OSI Referenzmodell . . . D.3
2 Klassifikation . . . D.4
3 Synchrones Request-Reply Modell . . . D.5
4 Asynchrones Request-Reply Modell . . . D.6
5 Zuverlässigkeit . . . D.7
6 Remote Procedure Calls . . . D.10
7 Name-Server und Binden . . . D.15
D.4 OO Verteilte Anwendungen . . . D.16
D.5 OOP und Verteilung . . . D.17
1 Klassifikation von Interaktionsformen . . . D.17
2 explizite, orthogonale Interaktion . . . D.18
3 implizite, nicht-orthogonale Interaktion . . . D.19
4 uniforme / nicht-uniforme Interaktion . . . D.20
5 transparente / nicht-transparente Verteilung . . . D.21
6 Herausforderungen . . . D.22

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

D.6 Java RMI . . . D.23
1 Entfernte Objekte finden . . . D.24
2 Methodenaufruf . . . D.25
3 Parameterübergabe . . . D.26
4 Resumee . . . D.26
D.7 Middleware für verteiltes Programmieren . . . D.27

E Java & Komponentenmodelle, Jini . . . E.1
E.1 Überblick . . . E.1
E.2 Literatur . . . E.2
E.3 Komponentenmodelle . . . E.3
1 Software Komponenten . . . E.3
2 Softwarekomponenten (2) . . . E.4
3 Philosopie . . . E.5
4 Beispiel für ein Builder Tool . . . E.6
5 Beispiel für eine Anwendung . . . E.7
6 Komponentenmodell . . . E.8
E.4 Java & Softwarekomponenten . . . E.9
1 Java — Ziele . . . E.9
2 Java — wesentliche Eigenschaften für Komponenten . . . E.10
3 Java-Komponentenmodelle . . . E.10
E.5 JavaBeans . . . E.11
1 Definition . . . E.11
2 Architektur . . . E.12
3 Beispiele . . . E.13
4 Properties . . . E.14
5 Events . . . E.17

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

6 Adapter . . . E.19
7 Introspection . . . E.20
8 JavaBeans — Summary . . . E.21
E.6 Jini . . . E.22
1 Überblick . . . E.22
2 wesentliche Konzepte . . . E.23
3 Jini Services . . . E.24
4 Djinn . . . E.25
5 Discovery . . . E.26
6 Leasing . . . E.27
7 Ereignisse (Events) . . . E.28
8 Sicherheit . . . E.28
E.7 Realistische Anwendungen für Komponenten-Software? . . . E.29
E.8 Software-Komponenten für realistische Anwendungen? . . . E.30

F Design Patterns für nebenläufige und verteilte Objekte . . . F.1
F.1 Überblick . . . F.1
F.2 Literatur . . . F.1
F.3 Motivation . . . F.2
F.4 Standardprobleme . . . F.3
F.5 Design Patterns . . . F.4
1 Service Access and Configuration Patterns . . . F.4
2 Event Handling Patterns . . . F.7
3 Synchronization Patterns . . . F.8
4 Concurrency Patterns . . . F.9

G Middleware und verteilte Anwendungen: CORBA . . . G.1

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

1 Überblick . . . G.1
2 Literatur, URLs . . . G.2
G.1 Middleware für verteiltes Programmieren . . . G.3
G.2 CORBA-Überblick . . . G.4
1 Standard . . . G.4
2 Motivation . . . G.5
3 Entwurfsziele . . . G.6
4 OMA – Object Management Architecture . . . G.8
5 CORBA-Implementierungen . . . G.9
G.3 CORBA-Anwendungsobjekte . . . G.10
1 Verteilte CORBA-Objekte . . . G.10
2 Interface Definition Language (IDL) . . . G.12
3 Objekte Erzeugen und Binden . . . G.18
G.4 Object Request Broker – ORB . . . G.19
1 Architektur . . . G.20
2 Statische Stubs . . . G.21
3 Object Adaptor - Vorschau . . . G.22
4 Objekterzeugung . . . G.24
5 Objektnutzung . . . G.27
6 Initialisierung der Anwendung . . . G.32

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.5 Object-Adaptor . . . G.33
1 Portable-Object-Adaptor (POA): Ziele . . . G.34
2 Portable-Object-Adaptor (POA): Überblick . . . G.35
3 POA-Erzeugung . . . G.36
4 Erzeugung von CORBA-Referenzen . . . G.37
5 Deaktivierung von Objekten, POA und Server . . . G.38
6 POA-Architektur . . . G.39
7 Bearbeitung von ankommenden Aufrufen . . . G.40
8 POA Policies . . . G.43
G.6 Dynamic-Invocation-Interface (DII) . . . G.45
1 Überblick . . . G.45
2 Interface Repository . . . G.48
3 Aufrufarten . . . G.51
4 Dynamic Skeleton Interface (DSI) . . . G.53
G.7 Inter-ORB-Kommunikation . . . G.54
G.8 CORBA Services . . . G.57
1 Naming Service . . . G.58
2 Life Cycle Service . . . G.61
3 Object Transaction Service (OTS) . . . G.65
4 CORBA Services - Zusammenfassung . . . G.69

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H Ausführungsumgebungen für verteilte Dienste und Anwendungen: .NET, EJB, OSGi . . . H.1
H.1 Motivation . . . H.1
H.2 Einflüsse aus Web-Services und Multi-Tier-Architekturen . . . H.2
1 EJB-Architektur (Details später in Kapitel EJB) . . . H.2
H.3 Java Server Pages und Servlets . . . H.3
1 Java Server Pages . . . H.3
2 JSP Beispiel . . . H.4
3 JSP Ziele . . . H.4
4 Servlets . . . H.5
H.4 Einflüsse: Microsofts OLE, COM und DCOM . . . H.6
1 Literatur . . . H.6
2 OLE – Object Linking and Embedding . . . H.7
3 COM – Component Object Model . . . H.8
4 DCOM – Distributed COM . . . H.9
5 ActiveX . . . H.9
6 COM+ . . . H.9
7 COM Architecture:
IDL – Interface Definition LanguageH.10
H.5 .NET-Architektur . . . H.19
1 Motivation & Ziele (1) . . . H.19
2 Literatur . . . H.22
3 .NET-Framework — Überblick . . . H.23
4 Sprachunterstützung . . . H.24
5 Common Language Infrastructur — CLI . . . H.25
6 Common Language Runtime . . . H.26
7 Assemblies . . . H.27

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

8 Virtual Execution System — VES . . . H.28
9 Unmanaged / Unsafe Code . . . H.29
10 Application Domains . . . H.30
11 Programmierung verteilter Systeme . . . H.32
H.6 OSGi — Open Services Gateway Initiative . . . H.34
1 Überblick . . . H.34
2 Einsatzbeispiele . . . H.35
3 Architektur . . . H.36
4 Services . . . H.37

I P2P-Systeme . . . I.1
I.1 Überblick . . . I.1
I.2 Einführung . . . I.2
1 Ausgangssituation . . . I.2
2 Klassifikation . . . I.8
3 Literatur . . . I.9
I.3 Napster . . . I.10
1 Protokoll - Verbindungsaufbau . . . I.11
2 Protokoll - Dateisuche . . . I.12
3 Protokoll - Dateitransfer . . . I.13
4 Zusammenfassung . . . I.15
I.4 Gnutella . . . I.16
1 Grundlage . . . I.17
2 Verbindungsaufbau . . . I.18
3 Nachrichten - Erhaltung der Netzwerkstruktur . . . I.19
4 Dateitransfer . . . I.21
5 Routing - Erhaltung der Netzwerkstruktur: . . . I.22

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

I.5 Verteilte Hash-Tabellen I.26
1 Übersicht I.26
2 Prinzipielle Arbeitsweise I.27
3 Eigenschaften von verteilten Hash-Tabellen I.43
I.6 JXTA I.44
1 Überblick I.44
2 Überblick I.45
3 Überblick I.46
4 Überblick I.47
5 Literatur I.48
6 Grundkonzepte I.51
7 Protokolle I.59
8 Endpoint Routing Protocol (ERP) I.60
9 Peer Resolver Protocol (PRP) I.62
10 Peer Rendezvous Protocol (RVP) I.64
11 Peer Discovery Protocol (PDP) I.68
12 Peer Information Protocol (PIP) I.69
13 Pipe Binding Protocol (PIB) I.70
14 Protokollstack I.71
15 Bewertung des Protokollstack I.72
16 Rendezvous Netzwerk der Referenzimplementierung I.73
17 Evaluierung I.75
18 Zusammenfassung I.76

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

J Enterprise Java Beans J.1
J.1 Motivation J.1
J.2 Architektur J.3
1 EJB-Architektur J.7
2 Rollen von EJB J.9
3 Unterschied zu klassischer Middleware J.12
J.3 EJB-Grundlagen J.17
1 Bean-Klassen J.18
2 Interaktion mit Beans J.19
3 Erzeugung von Beans J.22
4 Verwaltung des Lebenszyklus J.24
5 Bean-Interaktion mit dem Container J.25
6 Beispiel J.26
7 Deployment J.30
8 Interaktion mit der Bean J.31
J.4 Einordnung J.32

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

K Webservices, Verteilte Anwendungen, Grid-Architekturen K.1
K.1 Überblick K.1
K.2 Problematik K.2
1 Standards für Anwendungsinteraktion K.2
2 Enterprise Application Integration — EAI K.3
3 Business-to-Business Integration — B2BI K.6
K.3 Web Services K.7
1 Web Service Techniken K.8
2 Politik K.10
K.4 Web Services — Grundlagen und Standards K.11
1 SGML - Standard für Auszeichnungssprachen K.11
2 XML - Extensible Markup Language K.12
3 XML-Resumee K.24
4 Simple Object Access Protocol — SOAP K.25
5 Web Services Description Language — WSDL K.28
6 Universal Description, Discovery and Integration — UDDI K.37
K.5 Web-Services in Java K.39
K.6 Vergleich mit objektbasierter Middleware K.41
1 Minimalanforderungen K.42
K.7 weitere Standards und Entwicklungen K.44

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

L Betriebssystemarchitekturen L.1
L.1 Entwicklung der Betriebssystemstrukturierung L.1
1 Literatur L.2
L.2 Monolithische Betriebssystemkerne L.3
1 Vorteile L.3
2 Nachteile L.4
3 Gesamtablauf L.5
L.3 Minimalkerne L.6
1 Vorteile L.7
2 Nachteile L.8
3 Gesamtablauf L.9
L.4 Objektbasierte, offene Betriebssysteme L.10
L.5 Hardware- vs. Softwareschutzkonzepte L.12
1 Hardware-Schutz L.12
2 Software-Schutzkonzepte L.13
L.6 Objektorientierte BS: JX L.14
1 Grundkonzepte L.14
2 Architektur L.15
3 Inter-Domain-Kommunikation — Portale L.19
4 Optimierungen: Fast Portals L.24
5 Optimierungen: Memory Objekte L.25
6 Gerätetreiber L.30
7 Betriebssystembaukasten: Spezialsystem L.32
8 Erweiterungsmöglichkeiten L.36

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.