

Block B (Software-Architektur): Middleware 18.11.2004 1

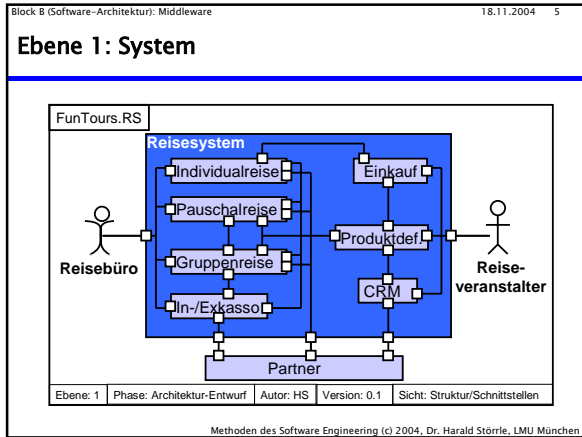
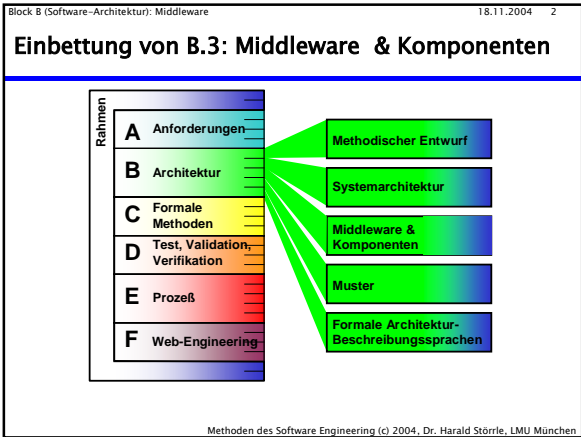
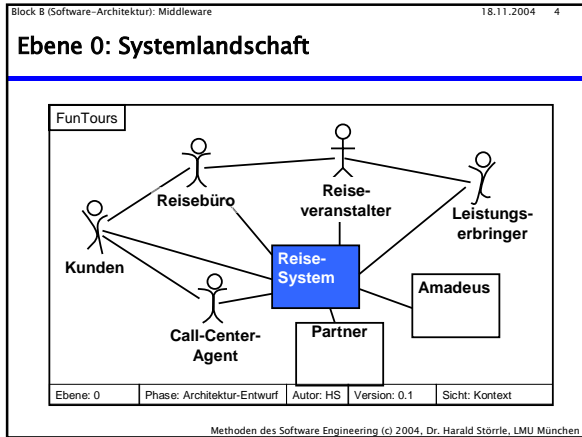
Vorlesung „Methoden des Software-Engineering“

Block B „Software Architektur“
Middleware

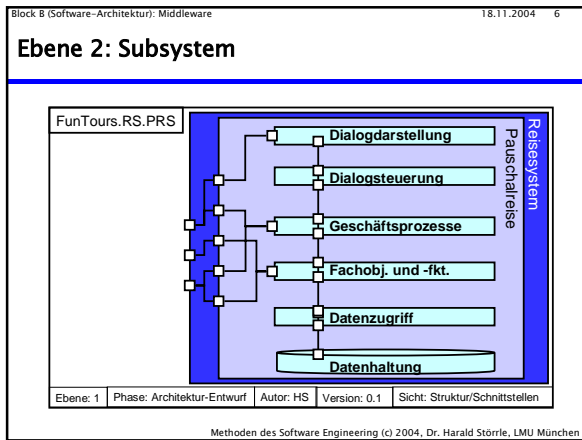
Dr. Harald Störrle

Einheit B.3, 18.11.2004

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München



- Block B (Software-Architektur): Middleware 18.11.2004 3
- ### Kernpunkte heute
- **Middleware ist die „Klebe“ zwischen Komponenten.**
 - Man spricht von „architectural glue“, d.h. die Infrastruktur für /zwischen Komponenten.
 - **Es gibt verschiedene Kategorien von Middleware, je nach Granularität und Art der Komponenten.**
 - Typische Beispiele sind Applikationsserver und Transaktionsmonitore, Corba, MQ-Series, RPC/RMI, und O2R-Mapping-Tools.
 - **Beispiele für verschiedene Arten von Middleware.**
- Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München



Block B (Software-Architektur): Middleware 18.11.2004 7

Ebene 3: Schicht

FunTours

Reisesystem

Pauschalreise

Fachobjekte und -funktionen

Partner

Ports

Reise

Daten-typen

Ebene: 1 | Phase: Architektur-Entwurf | Autor: HS | Version: 0.1 | Sicht: Struktur/Schnittstellen

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 10

Die Stecker-und-Steckdose-Metapher

- Die ursprüngliche Idee kommt in der Stecker-und-Steckdose-Metapher zum Ausdruck (McIlroy, 1968). Die Idee ist heute noch fesselnd, der Anspruch ist aber bis heute nicht wirklich eingelöst.

Komponenten

Steckplätze

Konnektor („Middleware“)

- Im Vordergrund standen zunächst die zu verbindenden Komponenten, das Verbinden und die Verbinder hielt man – in Anlehnung an das Vorbild Hardware – für trivial.
- Aber die Ports und Konnektoren können auch komplex sein. Dann ist die Middleware eher als eine Infrastruktur aufzufassen.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 8

Ebene 4: Modul

FunTours

Reisesystem

Pauschalreise

Fachobjekte und -funktionen

Reisetelle

Reise

Reise-produkte

Katalog

Ebene: 1 | Phase: Architektur-Entwurf | Autor: HS | Version: 0.1 | Sicht: Struktur/Schnittstellen

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 11

Komplexe Konnektoren als Abkürzung

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 9

Arten von Middleware je Ebene / Verteilungsraum

0 1 2 3 4 Ebene

Orte Rechner Prozesse Threads Verteilung über

MQ-Series Applikations-Server Laufzeitumgebung
Web-Services Corba Transaktions-Monitor O2R-Mapping
RMI / RPC

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 12

Middleware-Definition

- Konzeptuell steht Middleware zwischen einer Plattform (Hardware und Software) und der Applikation, daher der Name.
- Als Middleware bezeichnet man im Wesentlichen alle Infrastruktur-Komponenten, Produkte, Standards, Werkzeuge und Technologien, die die Herstellung und den Betrieb verteilter Anwendungen in heterogenen Umgebungen erleichtern.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 13

Beispiele für Middleware

Nachrichtenaustausch synchron	Nachrichtenaustausch asynchron	Daten- verteilung	Berechnungs- infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 16

DB/Datei als Puffer

...oder auch in eine Datei oder ein ähnliches Konstrukt
(z.B. in der Host-Welt).

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 14

Interprozesskommunikation mit Ports

Portnummer ist eine 16bit-Zahl
 Ports 0...1023 sind für spezielle Aufgaben
 reserviert und besonders geschützt.
 Beispiele
 - RMI: 1099
 - www: 80

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 17

Beispiele für Middleware

Nachrichtenaustausch synchron	Nachrichtenaustausch asynchron	Daten- verteilung	Berechnungs- infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 15

Beispiele für Middleware

Nachrichtenaustausch synchron	Nachrichtenaustausch asynchron	Daten- verteilung	Berechnungs- infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 18

Middleware in der Persistenzschicht O2R-Mapping

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 19

Middleware in der Persistenzschicht O2R-Mapping

Objektmodell
Konzepte und Beziehungen (im wesentlichen fachlich)

Relationenmodell
Relationen, Primär- und Fremdschlüssel, Attribute, Typen; Normalformen

physisches Datenmodell
Tablespaces, Indizes, DB-Schema, Optimierungen

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 22

O2R-Mapping Abbildung zwischen Objekt- und Relationenmodell

Reise
Produktnr.: String
Reiseantritt.: Date
Reiseende.: Date
Dauer.: int
AbOrt.: String

Pauschalreise
AnzErw.: int
AnzKinder.: int
Preis.: int
HotelKat.: String
Verpfl.: String

Gruppenreise
Gruppengröße.: int
Rabatt.: int
Teilnehmer.: PR

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 20

O2R-Mapping Abbildung zwischen Objekt- und Relationenmodell

(auch: „O2R-Mapping“)

- **Klasse** : **Tabelle**
- **Typ** : **Domäne**
- **Objekt** : **Zeile**
- **Attribut** : **Spalte**
- **Assoziation** : **Fremdschlüsselbeziehung**

- Vererbung wird als Delegation realisiert, d.h. über eine Assoziation.
- Dieses Schema wird aus Effizienzgründen aufgeweicht (entspricht De-Normalisierung in der klassischen Welt).

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 23

O2R-Mapping Embedded SQL – Daten laden

```

class DB_Reise {
    EXEC SQL BEGIN DECLARE SECTION;
    private r_id DB_ID;
    private abOrt String;
    private date reiseanfang;
    ...
    EXEC SQL END DECLARE SECTION;

    public load(r_ID String) {
        EXEC SQL select r_id, abOrt, ...
        into :r_id, :abOrt, ...
        from REISE
        where r_id = r_ID;
        if (SQLSTATE != '0000')
            // Error
    }

    class Reise extends DB_Reise {
        // zusätzliche Attribute
        // Methoden
    }
}

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 21

O2R-Mapping Klassen- vs. ER-Diagramme

Reise
Produktnr.: String
Reiseantritt.: Date
Reiseende.: Date
Dauer.: int
AbOrt.: String

Pauschalreise
AnzErw.: int
AnzKinder.: int
Preis.: int
HotelKat.: String
Verpfl.: String

Gruppenreise
Gruppengröße.: int
Rabatt.: int
Teilnehmer.: PR

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block 8 (Software-Architektur): Middleware 18.11.2004 24

O2R-Mapping Embedded SQL – Daten speichern

```

class DB_Reise {
    EXEC SQL BEGIN DECLARE SECTION;
    private r_id DB_ID;
    private abOrt String;
    private date reiseanfang;
    ...
    EXEC SQL END DECLARE SECTION;

    public store() {
        EXEC SQL insert
        into REISE (r_id, abOrt, ...)
        values (:r_id, :abOrt, ...)
        if (SQLSTATE != '0000')
            // Error
    }

    class Reise extends DB_Reise {
        // zusätzliche Attribute
        // Methoden
    }
}

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 25

O2R-Mapping Embedded SQL – Cursor

```

...
public bulkLoad(DB_ID r_id) {
    EXEC SQL DECLARE x Cursor FOR
    select pr_id, grr_id
    from GRR_TNR
    where grr_id = :r_id;
    EXEC SQL OPEN x;
    while (x.hasMoreElements) {
        EXEC SQL
        FETCH x into :grr_id, :pr_id
    }
    ...
}

```

Pauschalreise

AnzErw: int
AnzKinder: int
Preis: int
HotelKat: String
Verpfl: String

Gruppenreise

Gruppengröße: int
Rabatt: int
Teilnehmer: PR

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 28

Applikationsserver & Transaktionsmonitore

- Generell ist diese Art von System vor allem auf dem Host zu finden.
- Die beiden mit Abstand am weitesten verbreiteten Produkte sind
 - CICS (Customer Information and Control System) und
 - IMS (Integrated Management System).
- Auf Unix-Rechnern ist dieser Ansatz eher unüblich (->Tuxedo).
- Diese Lücke schließen seit einigen Jahren Applikationsserver wie IBM WebSphere und BEA WebLogic.
- Es gibt inzwischen auch konkurrenzfähige Open-Source-Produkte wie JBoss.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 26

Beispiele für Middleware

Nachrichtenaustausch		Daten-	Berechnungs-
synchron	asynchron	verteilung	infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
- Web-Server
- LDAP
- EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 29

CICS/IMS-Leistungskennzahlen

Anzahl der Sachbearbeitern	von einigen Hundert bis zu mehreren Tausend
Tägliches Transaktionsvolumen	Beispiele: • Inter Krankenversicherung: 300.000 • Volksfürsorge 2,5 Millionen
Antwortzeitverhalten	ca. 90 % der Transaktionen unter 1 Sek.
Datenvolumen	von einigen Hunderttausend bis zu mehreren Millionen aktive Verträge

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 27

Applikationsserver & Transaktionsmonitore

- Applikationsserver (App.-Server) und Transaktionsmonitore (Txn.-Mon.) verwalten Rechner-Ressourcen, z.B.:
 - Haupt- und Massenspeicher,
 - I/O- und Rechenleistung,
 - DB- und Kommunikationsverbindungen,
 - Transaktionskontrolle (ACID, Rollback/Recovery, usw.),
 um eine sehr hohe Zahl von Nutzern gleichzeitig zu bedienen.
- Sie bieten also ähnliche Funktionalität wie Betriebssysteme, aber viel leichtgewichtiger und mit zahlreichen Details Einschränkungen, die eine effizientere Verarbeitung erlauben.
- Zu den Einschränkungen zählt insbesondere ein das „Programmiermodell“, also die Richtlinien und Schnittstellen, an die man sich halten muß, um diese Infrastruktur zu nutzen.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 30

CICS-Code (Ausschnitt)

```

public class Reise {
    public int zustand = 0;

    public bool buchen(...) {
        if (zustand != 0) {throw...}
        zustand = 1;
        EXEC CICS START TRANSID('...') INTERVAL(100)
        END-EXEC
        . . .
        EXEC CICS INQUIRE TASK LIST LISTSIZE(WS-LISTSIZE)
        . . .
        EXEC CICS RETURN
    }
}

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 31

Beispiele für Middleware

Nachrichtenaustausch		Datenverteilung	Berechnungsinfrastruktur
synchron	asynchron		
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 34

CORBA – Grundlegender Aufbau

z.B. Naming, Transaction, Event

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 32

CORBA

- CORBA steht für Common Object Request Broker Architecture und ist als Industriestandard von der OMG seit den frühen 90ern entwickelt worden.
- Es gibt inzwischen sehr viele qualitativ hochwertige kommerzielle und frei verfügbare Implementationen von vielen verschiedenen Herstellern.
- Corba arbeitet plattformübergreifend und Programmiersprachen-unabhängig, und inzwischen gibt es auch sehr schnelle Implementierungen („RT-ORBs“).

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 35

CORBA – Grundlegender Aufbau

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 33

CORBA – Grundlegender Aufbau

z.B. Naming, Transaction, Event

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 36

CORBA-IDL

- Die Interface Description Language (IDL) beschreibt Signaturen von Komponenten ähnlich wie Definition-Module in Modula.
- Sie umfasst eine Menge von Basisdatentypen und Konstruktoren ähnlich wie in C++ oder Java.
 - bool, int, longint, ...
 - exception, array, sequence
- Zusätzlich gibt es Typ-Konstruktoren und -Modifikatoren ähnlich wie in Modula oder Ada.
 - enum
 - struct, union („record“ und „variant record“ in Modula 2)
 - in, out, inout in Parameterlisten
- Zeiger gibt es nur in Form von Interoperable Obj References (IOR).

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 37

IDL-Beispiel: Deklaration (1)

```

module com
{
  module jcom
  {
    module bank
    {
      module CorbaLink
      {
        exception InvalidUserOperation
        {
          string ErrorString;
        };

        // Exception Codes thrown when error occurs while processing
        enum FailReason
        {
          USER_AUTHENTICATION_ERROR, DATABASE_CONNECTION_ERROR,
          INVALID_PARAMS, NO_SUCH_METHOD, AMOUNT_TRANSACTION_ERROR
        };

        // Exception Codes thrown when error occurs while processing
        exception TransactionError
        {
          string ErrorString;
          FailReason resonCode;
        };
      };
    };
  };
};
  
```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 40

Kapselung von Middleware

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 38

IDL-Beispiel: Deklaration (2)

```

//All the Operations offered by the account interface AccountManager
{
  boolean update_balance(in string username, in double amount) raises (TransactionError);

  boolean current_balance(in string username, out double balance) raises (TransactionError);

  boolean delete_account(in string username) raises (TransactionError);
};

// Does the Authentication. User has to go through these methods to get
// access to the Account Object and perform operations on the account
interface UserAuthenticator
{
  AccountManager login_user(in string username, in string password) raises (InvalidUserOperation);
  AccountManager register_user(in string username, in string password) raises (InvalidUserOperation);
}; }; };
  
```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 41

Schichtenarchitektur Aufgaben der einzelnen Schichten

Präsentation	<ul style="list-style-type: none"> Benutzeroberfläche Dialogsteuerung 	<ul style="list-style-type: none"> Darstellung der Ergebnisse Benutzerführung innerhalb einer Maske Verarbeitung von Eingaben Steuerung der Abfolge von Masken erste Plausibilisierung
Anwendung	<ul style="list-style-type: none"> Geschäftsprozesse Domänenobjekte 	<ul style="list-style-type: none"> Koordinierung der NF'e eines GePros Plausibilisierung Bewertungen Berechnung abgeleiteter Attribute Domänenfunktionen implementieren NF'e Integritätsbedingungen
Daten	<ul style="list-style-type: none"> logischer Datenzugriff physischer Datenzugriff 	<ul style="list-style-type: none"> Zugriff auf Fachobjekte Caching Zugriff auf DB-Tabellen Caching Indexing

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 39

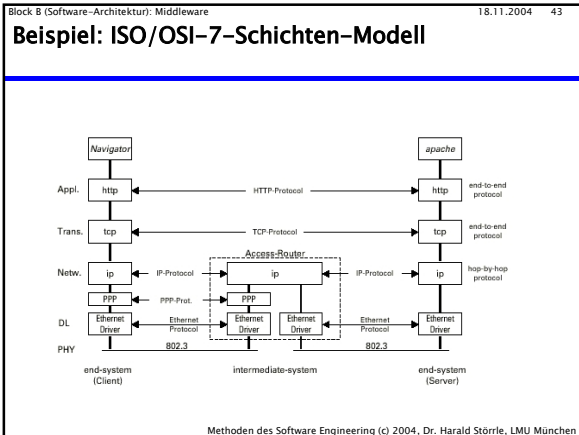
CORBA - Vorgehensweise

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 42

Schichtenarchitektur Übergreifende Dienste schlechte Darstellung

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München



Block B (Software-Architektur): Middleware 18.11.2004 46

MQSeries Beispielcode (1)

```

MQEnvironment.hostname = „mqjavatest“;
MQEnvironment.channel = „SYSTEM.DEF.SVRCONN“;
MQEnvironment.port = 1414;

MQQueueManager QM = new MQQueueManager („name“);
MQQueue queue = MQQueueManager.accessQueue(
    „System.DEFAULT.LOCAL.QUEUE“, MQC.MQOO_OUTPUT);
MQMessage msg = new MQMessage();

MQGetMessageOptions mqgmo = new MQGetMessageOptions();
mqgmo.options = MQC.MQGMO_NO_WAIT;

queue.getMessage(msg, mqgmo);

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 44

Beispiele für Middleware

Nachrichtenaustausch		Daten-	Berechnungs-
synchron	asynchron	verteilung	infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 47

MQSeries Beispielcode (2)

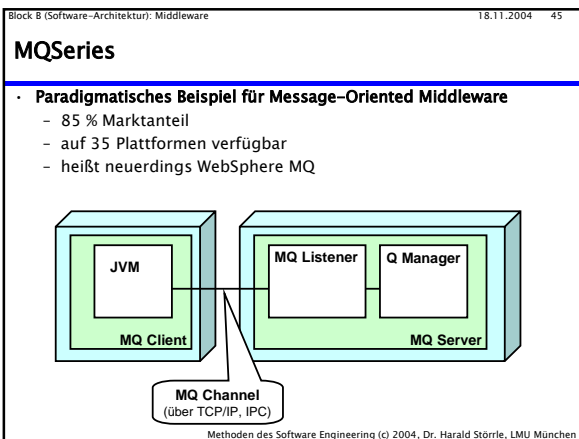
```

int qDepth = queue.getCurrentDepth();

try {
    queue.getMessage(msg, mqgmo);
}
catch (MQException mqe) {
    if (mqe.reasoncode == MQException.MQRC_NO_MSG_AVAILABLE) {
        // ok
    }
    else {
        // unexpected error
    }
}

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München



Block B (Software-Architektur): Middleware 18.11.2004 48

Beispiele für Middleware

Nachrichtenaustausch		Daten-	Berechnungs-
synchron	asynchron	verteilung	infrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services (SOAP)	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele
 - Web-Server
 - LDAP
 - EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 49

Tuple-Spaces in Linda

Der Tuple-Raum

- Linda ist ein abstraktes Konzept zur Beschreibung von Interaktionen.
- Linda selbst ist unabhängig von einer Programmiersprache.
- Es gibt Linda-Implementationen in allen möglichen Sprachen.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 52

Der Tuple-Raum

Slaves

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 50

Tupel und Felder

- Tupel sind eine Folge von getypten Feldern.
- Beispiele:
 - ("arraydata", dim1, 13, 2)
 - (var1, var2)
 - ("common block", datacom)
- Die Elemente in den Feldern sind entweder Werte oder formale Parameter oder auswertbare Ausdrücke (const, expr, var).
- Der Tupelraum ist ein assoziativer Speicher, d.h. ein Tupel wird über seinen Inhalt adressiert.
- Das Tupel ("arraydata", 4, 6, 8) paßt auf folgende beide Templates
 - ("arraydata", ?dim1, ?dim2, ?dim3);
 - ("arraydata", 4, ?dim2, ?dim3);

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 53

Beispiel in C-Linda

```

#include <stdio.h>
#include "LindaC.h"

const int SLAVES = 6;
union LiObject t;
int i;

void * slave (void *v)
{
    union LiObject t;
    int i = (int)v;

    printf("hello from slave %d\n", i);
    outLi("done", t);
    exitLi(0);
    return 0;
}

int main()
{
    for ( i = 0; i < SLAVES; ++i ) {
        int rstatus =
            evalLi(slave, (void *)i);
        if ( rstatus != 0 ) {
            fprintf(stderr,
                "eval: error %d\n", rstatus);
            exitLi(0);
        }
        for ( i = 0; i < SLAVES; ++i ) {
            inLi("done", &t);
            printf("done\n");
            exitLi(0);
            return 0;
        }
    }
}
    
```

Der Tuple-Space

6 Prozesse anlegen...

und ihre Nachrichten abfangen

Legende
 bl au Datentypen
 grün Kontrollstrukturen
 rot C-Linda

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 51

Linda Operationen

out

- erzeugt passives Tupel
- gibt Kontrolle sofort zurück

```
out('array data', dim1, dim2);
```

eval

- erzeugt aktive Tupel („Prozess“),
- gibt Kontrolle sofort zurück.
- alle Felder werden nebenläufig ausgewertet und im TS abgelegt

```
eval("test", i, f(i));
```

in

- wartet auf ein Tupel, das zum vorgegebenen Muster passt
- wenn ein Tupel gefunden wird, wird das Tupel entnommen

```
in("arraydata", ?dim1, ?dim2);
```

rd

- wartet auf ein Tupel, das zum vorgegebenen Muster passt
- wenn ein Tupel gefunden wird, wird das Tupel kopiert

```
rd("arraydata", ?dim1, ?dim2);
```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 54

Beispiele für Middleware

Nachrichtenaustausch synchron	Nachrichtenaustausch asynchron	Datenverteilung	Berechnungsinfrastruktur
OS-Ports	DB	O2R-mapping eSQL	CICS
Web-Services (SOAP)	MQSeries	Tuple-spaces in Linda	CORBA

Andere valide Beispiele

- Web-Server
- LDAP
- EJB, .NET, JavaBeans, COM

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 55

SOAP

Simple Object Access Protocol

- XML-basiertes Nachrichtenformat zur Übertragung von strukturierten und typisierten Daten
 - Dokumente (Messaging)
 - Entfernte Prozeduraufrufe (RPCs)
- mit allen gängigen Internetprotokollen übertragbar

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 58

SOAP in Java

- z. B. Package: `javax.xml.soap (Axis)`
(<http://ws.apache.org/axis>)

```

SOAPMessage msg = MessageFactory.newInstance().createMessage();
SOAPPart soapPart = msg.getSOAPPart();
jSOAPEnvelope envelope = soapPart.getEnvelope();
SOAPBody body = envelope.getBody();
body.addBodyElement(...);

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 56

Aufbau von SOAP-Nachrichten

```

Envelope
├── Header
│   └── Zusatz-information
└── Body
    └── Nachricht

```

- Envelope:** die Hülle der Nachricht
- Header:** der Briefkopf (optional)
- Body:** der Inhalt der Nachricht

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 59

Verwendung von SOAP als Konnektor in Java/A (I)

```

classDiagram
    class Bank
    class ATM
    class BankPort
    class ATMPort
    Bank --> BankPort
    ATM --> ATMPort
    BankPort -- ATMPort

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 57

SOAP: Beispielanfrage an google.com

```

<?xml version='1.0' encoding='UTF-8'?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <doGoogleSearch xmlns="urn:GoogleSearch">
      <key xsi:type="xsd:string">3289754870548097</key>
      <q xsi:type="xsd:string">Eine Anfrage</q>
      <start xsi:type="xsd:int">0</start>
      <maxResults xsi:type="xsd:int">10</maxResults>
    </doGoogleSearch>
  </env:Body>
</env:Envelope>

```

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 60

Verwendung von SOAP als Konnektor in Java/A (II)

```

composite component BankATM {
  configuration {
    component Bank bank = new Bank();
    component ATM atm = new ATM();
    connector SOAPConnector sn = new SOAPConnector( ...);
    sn.connect(bank.BankPort, atm.ATMPort);
  }
}

simple component ATM {
  port ATMPort {
    ...
    required {
      void enterPIN(int pin);
    }
  }
  ...
  ATMPort.enterPIN(pin);
  ...
}

```

BankATM: **connector SOAPConnector sn = new SOAPConnector(...);**

ATM: **void enterPIN(int pin);**

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 61

Themenwechsel

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 64

Ablauf einer Inspektion

Schritt	Wer	Aktivitäten, Umfang, Dauer
Planung	M, A (, I)	Vorbedingungen prüfen Termine & Gegenstand festlegen
Oberblick	M, A, I ₁ , ..., I _n	Autor stellt Gegenstände vor (ca. 1h) Gruppenabstimmung
Vorbereitung	M, I (, A)	Durchlesen (125NLoC/h · Person) Checklisten abarbeiten
Inspektion	M, A, I	L. trägt vor, I.'en unterbrechen, A. erläutert (90–125 NLoC/h, max.2h) Protokoll durch Moderator
Korrektur	A	gemäß Protokoll (nur diese Fehler!)
Abnahme	M, A	ggf. weitere Sitzung

} 3-5h

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 62

Ziel und Anwendungsbereich von Inspektionen

- **Ziel:**
 - Auffinden, Klassifizieren, Beheben von Fehlern im Prüfgegenstand
 - Validation
- **Anwendbar auf:**
 - im Prinzip auf alle Artefakte für die es ein Größenmaß gibt
 - (Programm: LoC, Entwurf: NoM, Textdokument: Seiten)
 - Kalibrierung/Bezugsgröße: Sprache, Stil- & Formatvorgaben
 - Inspektionsgruppe muß kompetent sein (doppelter Wortsinn)
 - mittlere und große Projekte, sehr frühe bis mittlere Phasen
- **Einfache, aber wirkungsvolle Methode**

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 65

Ablauf einer Inspektion (Größenordnungen)

- **Re-Inspektion notwendig bei (Faustregeln):**
 - Korrekturen verändern Prüfgegenstand um mehr als 5%.
 - Kann man z.B. in NLoC/dNLoC messen
 - NLOC = Non-Comment Lines of Code
 - Anz. Fehler > 2 * (φ frühere Inspektionen)
 - entspricht etwa 3–20 funktionale Fehler/1000 LoC
 - Killer-Fehler
- **Wichtig:**
- **Durch Inspektion werden die Grundlagen für quantitative Erfassung des Software-Prozesses gelegt!**
 - Zeitreihen von Fehlerdichten
 - Raten/Aufwände von Reparaturen

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Block B (Software-Architektur): Middleware 18.11.2004 63

Rollen in einer Inspektion

- **Es werden bestimmte Rollen mit bestimmten Aufgaben festgelegt.**
- **Moderator (M)**
 - Der Moderator bereitet die Inspektion vor, führt sie durch, und überwacht ihren Abschluß.
- **Inspekteur (I)**
 - Ein Inspekteur ist ein fachlich kompetenter Mitarbeiter, der bei der Inspektion mitarbeitet. Je nach Prüfgegenstand nehmen die Inspekteure reihum die Rolle des Lesers ein.
- **Autor (A)**
 - Der Autor stellt sein Produkt vor, und führt Nacharbeiten durch, die durch die Inspektion angefordert werden.

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

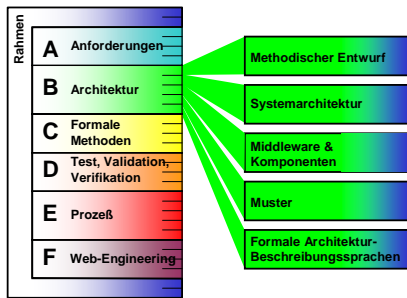
Block B (Software-Architektur): Middleware 18.11.2004 66

Inspektion Variante „Begehung“ (Walkthrough)

- **Alle Rollen werden aus dem Projekt besetzt:**
 - wechselseitige Begutachtung verhindert unkonstruktives Verhalten,
 - kann aber bei sehr kleinen Projekten schwierig zu besetzen sein.
- **Dafür gibt es die weniger formale Form der Begehung („Walk Through“).**
- **Hier sind Autor, Moderator und Leser ein und dieselbe Person. Vorbereitung entfällt (bis auf die Eingangsprüfung).**
- **Der Inspekteur stellt Fragen, gemeinsam wird das Protokoll angefertigt.**

Methoden des Software Engineering (c) 2004, Dr. Harald Störle, LMU München

Ausblick auf Block B Teil 4: Muster



ENDE