

Verteilte Web-basierte Systeme – SS 2006

Verteilte Web-basierte Systeme

Dr.-Ing. Martin Gaedke
Universität Karlsruhe (TH)

Verteilte Web-basierte Systeme – SS 2006

Part VII

Evolution

Part 7 – Overview

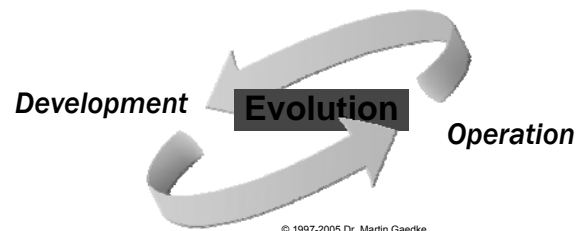
About Evolution...

1. Testing and Monitoring
 - Test Plans and Execution
 - Monitoring
 - Approaches, Models, Tools
2. Operations and Maintenance
 - Configuration and Evolution
 - UDDI
 - Promotion
 - Availability
 - Maintenance

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

3

Evolution... continuous progress



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

4

Verteilte Web-basierte Systeme – SS 2006

Chapter://1

Testing and Monitoring

Part VII ► Chapter://1 ► Testing and Monitoring

Introduction

- ☞ Testing is extremely difficult
 - Testing is a continuous process
 - Starts during Assessment Phase
 - Address Testing seriously! Prepare for Test Plans
- ☞ Early:
 - Requirements for testing
 - Criteria for non-functional requirements – How to test for “good”?
- ☞ Later:
 - Define Test Cases
 - Unit Tests etc.
- ☞ Final:
 - Release Test Criteria (e.g. ZBB, Customer Feedback, etc.)
 - Based on Criteria an Internal Release becomes a Release

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

5

Part VII ► Chapter://1 ► Testing and Monitoring

Testing Problems

- ☉ Things to look at – and define Test Plans for:
 - Spelling errors, broken links, buggy scripts
 - User and eCommerce night mares: overcharging accounts
 - Assumption of Correctness “site is correct because it looks and loads right”-Syndrome
 - **The scenario: Browser Types x Plugins x Script Engines x OS x Hardware x Network Connections**
- ☉ BTW, Job of Testing – a social problem...
 - Testing is a difficult job and in many cases thankless
 - Proper testing is often not understood or appreciated and often seen as a boring task

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

7

Part VII ► Chapter://1 ► Testing and Monitoring

Test Plans and Procedures

- ☉ Prepare Test Plans
 - Functionality Testing
 - Content Testing
 - User Testing
 - Security Testing
- ☉ If applicable try to use Test Labs
- ☉ Procedures for finding issues/problems
 - Report Problem Tracking System
 - **Track, Handle, Finalize:** Initiate Change Request
 - Be integral part of Configuration Management

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

8

Part VII ► Chapter://1 ► Testing and Monitoring

Functionality Testing

- ☉ Site functions properly and meets specification
- ☉ Main Testing: Units, integration (all units together), browser
- ☉ Final Testing: User's system configuration (e.g. speed of hard-drives, Java runtime with different processor speeds), delivery (network and server aspects)
- ☉ **Test Labs** may help in some cases, especially for final testing

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

9

Part VII ► Chapter://1 ► Testing and Monitoring

Content Testing

- ☉ Content of site is correctly implemented
- ☉ Consider proof-reading, especially spelling of names and companies
- ☉ Check for copyright inclusion and legal disclaimers
- ☉ Check images and other media type (includes consideration of user's system configuration)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

10

Part VII ► Chapter://1 ► Testing and Monitoring

User Testing

- ☉ Site meets user's needs and is usable
- ☉ If available testing rooms
- ☉ Low-cost testing with some people and questionnaires
- ☉ Online-testing with feedback option (“send us your comments and you may win...”)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

11

Part VII ► Chapter://1 ► Testing and Monitoring

Security Testing

- ☉ Should not be part of functionality testing
- ☉ Handle explicitly
- ☉ Include application-, server-, network-, physical site-security, and physical access by Staff, as well as many other issues

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

12

Part VII ► Chapter://1 ► Testing and Monitoring

Handling Results Of Testing

- ☉ Use a Problem Tracking System
 - ▀ **Track:** Problem (Id, problem description, discovered by, when, user's system configuration, severity,...)
 - ▀ **Handle:** ProblemId, HandledBy, Status, ...
 - ▀ **Finalize:** Initiate Change Request (CR)
- ☉ Should be part of a change request processing approach, cf. Requirements Engineering

13

Part VII ► Chapter://1 ► Testing and Monitoring

Bobby

http://bobby.watchfire.com/bobby/html/en/index.jsp

14

Part VII ► Chapter://1 ► Testing and Monitoring

Web Application Test Tools

Screenshot: Microsoft Application Center Test

15

Part VII ► Chapter://1 ► Testing and Monitoring

Evolution – Ongoing Process

16

Verteilte Web-basierte Systeme – SS 2006

Chapter://2

Operation and Maintenance

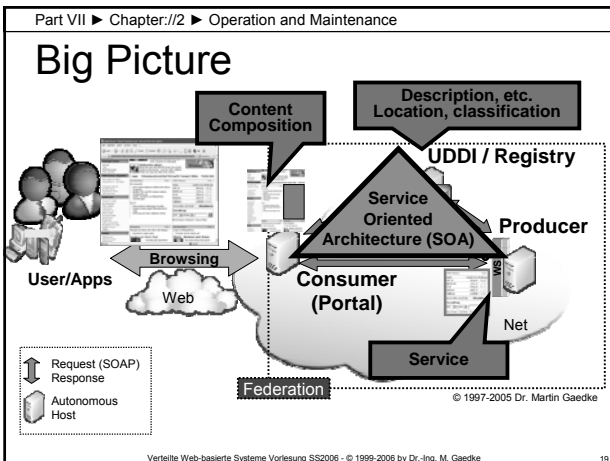
17

Part VII ► Chapter://2 ► Operation and Maintenance

Evolution: Plan For Change

- ☉ Evolution requires for continuous Operation and Maintenance, i.e.
 - ▀ Continuous testing and monitoring
 - ▀ Bugs or changed requirements → Change the application
 - ▀ Accepted changes (cf. Change Management and CCB) are handled
- ☉ Configuration as an approach to evolution
 - ▀ Add: Wire existing components by adding to configuration
 - ▀ Remove: Delete wiring context in component configuration

18



Verteilte Web-basierte Systeme – SS 2006

Section://1

UDDI

- Part VII ► Chapter://2 ► Operation and Maintenance: UDDI
- ### Discovering Web Services
- ⊛ **Universal Description, Discovery, and Integration (UDDI)** – Specifies what the API for a **Web-based Registry** looks like.
 - All about the “Yellow, White & Green Pages”
 - Defines how to run and operate Registry Sites on the Web
 - Defines how to pay for its Operation – encourages basic lookup services for free
 - ⊛ Further Information at <http://uddi.org>
- Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

- Part VII ► Chapter://2 ► Operation and Maintenance: UDDI
- ### Registry Operation
- ⊛ Peer nodes (websites)
 - ⊛ Companies register with any node
 - ⊛ Registrations replicated on a daily basis
 - ⊛ Complete set of “registered” records available at all nodes
 - ⊛ Common set of SOAP APIs supported by all nodes
 - ⊛ Compliance enforced by business contract
-
- Attention! Discontinuation Notice (for public uddi)**
- Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

- Part VII ► Chapter://2 ► Operation and Maintenance: UDDI
- ### Why a DNS-like Model?
- ⊛ Enforces cross-platform compatibility across competitor platforms
 - ⊛ Demonstration of trust and openness
 - ⊛ Avoids tacit endorsement of any one vendor’s platform
 - ⊛ May migrate to a third party
- Attention! Discontinuation Notice**
- Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

UBR: UDDI Business Registry

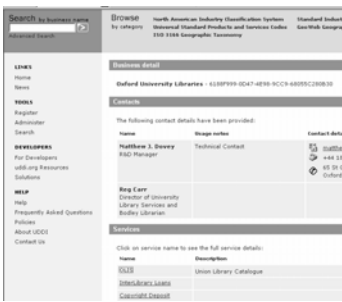
Attention! Discontinuation Notice

UDDI von Microsoft (<http://uddi.microsoft.com>) und IBM (<http://uddi.microsoft.com>)

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

UDDI provides information...

- Who – Business Information
- What – Find the right Type of Business
- Where – To Access a Service
- How – Describes how a given Interface functions




Information provided at <http://uddi.microsoft.com>

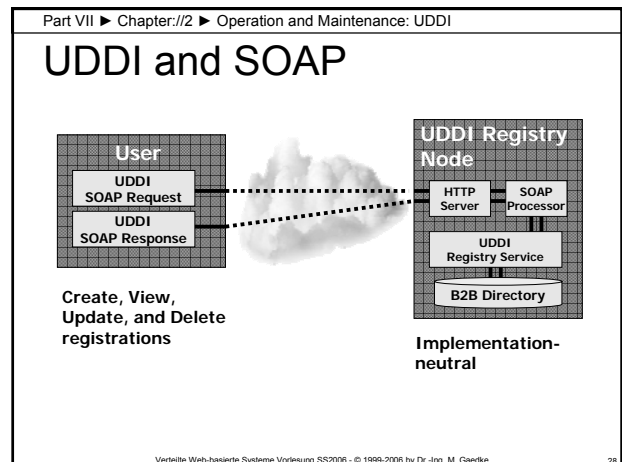
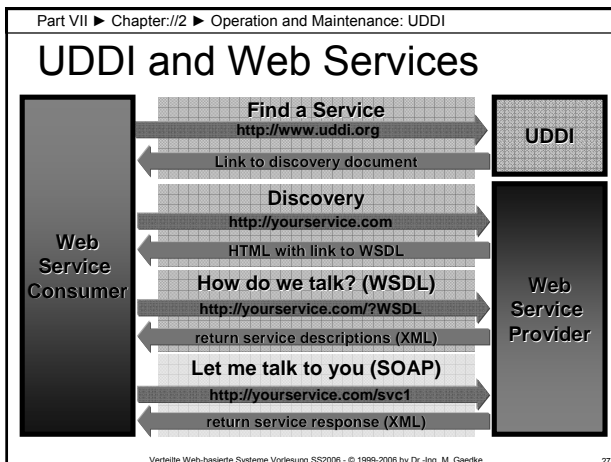
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

UDDI – A Publisher View



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke



Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

Registry APIs (SOAP)

- Inquiry API
 - Find things
 - find_business
 - find_service
 - find_binding
 - find_tModel
 - Get Details about things
 - get_businessDetail
 - get_serviceDetail
 - get_bindingDetail
 - get_tModelDetail
- Publishers API
 - Save things
 - save_business
 - save_service
 - save_binding
 - save_tModel
 - Delete things
 - delete_business
 - delete_service
 - delete_binding
 - delete_tModel
 - security...
 - get_authToken
 - discard_authToken

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

Service Type Registration

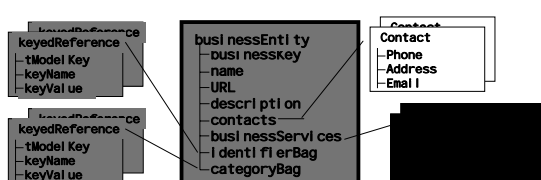
- Pointer to the specifications describing the service type
 - May be any combination of documents or XML resources (schemas, WSDL, etc.)
- Identifier – who published the Service
- Identifier – for the Service Type itself
 - Called a tModelKey
 - Used as a unique signature by Web sites that implement those services

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

Business Registration


- businessEntity – Holds all of the information in UDDI related to a single business or business unit
 - It is the “master container”
 - Organization hierarchy not defined in V1



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

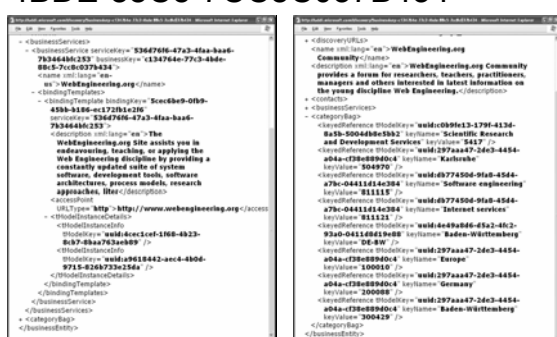
businesskey=C134764E-77C3-4BDE-88C5-7CC8C037B434



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: UDDI

businesskey=C134764E-77C3-4BDE-88C5-7CC8C037B434



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Verteilte Web-basierte Systeme – SS 2006

Section://2 Promotion

Part VII ► Chapter://2 ► Operation and Maintenance: Promotion

Introduction

- Activities after site development:
 - Promotion
 - Maintenance
- This may imply new development cycles: moving through all stages again... Promotion and Maintenance ...
- Process vs. Product View
 - Process-based: **Life Cycle of a Web-Application**
 - Product-based: **Evolution of a Web-Application**

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: Promotion

Promotion

- Question: How to find the Web application?
- Promotion for intranets and extranets
 - Do promote internal Web-Applications!
 - Pre-set browser home, add link on the internal company's homepage, email, trainings, ...
- Promotion for public Web applications
 - Cf. marketing approaches for a product
 - E.g. advertisements in print media, business cards, cross-link with existing customers, etc. → Careful due to access behavior
 - Register with search engines and public portals

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: Promotion

Promotion: Search Engines

- ☉ People get better in using search engines
 - Term of the year 2002: "Google it"
 - Cf. IEEE Technical Speeking
- ☉ Search Engines
 - Content is not understandable by Machines
- ☉ Facilitate Process by Providing Meta-Information
 - In HTML: META-Tag
 - <META NAME="description" CONTENT="WebE Homepage">
 - <META NAME="keywords" CONTENT="WebE, Vorlesung">
 - And there are more secrets... to influence search engine algorithms (Goal: become a top-5 entry)

37

Part VII ► Chapter://2 ► Operation and Maintenance: Promotion

Promotion: Search Engines II

- ☉ Check **Referer** header of an HTTP-Request
- ☉ Log-File Analysis for <http://www.webengineering.org>

```

http://www.google.com/
-----
Total Requests      Total 304's (Method Req)      Bytes sent | Referer URL
-----
2  0.50%      0  0.00%      14700  0.50% | /search q=-commerce applications
2  0.50%      0  0.00%      1798  0.04% | /search q=WebE is web engineering hl=en srarc=10 sa=W
2  0.50%      0  0.00%      2800  0.20% | /search q=-commerce applications btnG=Google Search
2  0.50%      1  2.62%      1692  0.04% | /search q=web hl=en srarc=110 sa=W
1  0.17%      0  0.00%      4900  0.10% | /search q=web engineering hl=en srarc=50 sa=W
1  0.17%      0  0.00%      1150  0.02% | /search q=www10 srarc=120 sa=W
1  0.17%      0  0.00%      4900  0.10% | /search q=Web-based distributed system api... hl=en sr
1  0.17%      0  0.00%      4900  0.10% | /search q=Web Engineering hl=de srarc=50 sa=W
    
```

38

Verteilte Web-basierte Systeme – SS 2006

Section://3

Availability

Part VII ► Chapter://2 ► Operation and Maintenance: Availability

What Is High Availability?

- ☉ How much downtime can my organization afford without losing productivity, profits, sales, etc.?
- ☉ The solution to High Availability is a combination of people, process, AND technology
 - Beware of 99.99% myth - The nines model does not take timing into account

Availability:

- ☉ Simple Example:
 - 24/7 Web Site with two failure a week and each requires 1 hour
- ☉ On a year's time:

$$\frac{(52 \cdot 7 \cdot 24 / 52 \cdot 2)}{(52 \cdot 7 \cdot 24 / 2) + 1/2} \cdot 100 = 99.941\%$$

40

Part VII ► Chapter://2 ► Operation and Maintenance: Availability

Achieve High Availability?

- ☉ It's deceptively simple ...
 - Plan and prepare
- ☉ Key to high availability
 - Deploy systems to create **redundancy** – the key from a technology standpoint, e.g. replicate Web server application logic (scale out, DNS-round robin, Network Load Balancer)
 - Define processes for people to solve conflicts
 - Test, test, test
 - Monitor on a continuous basis

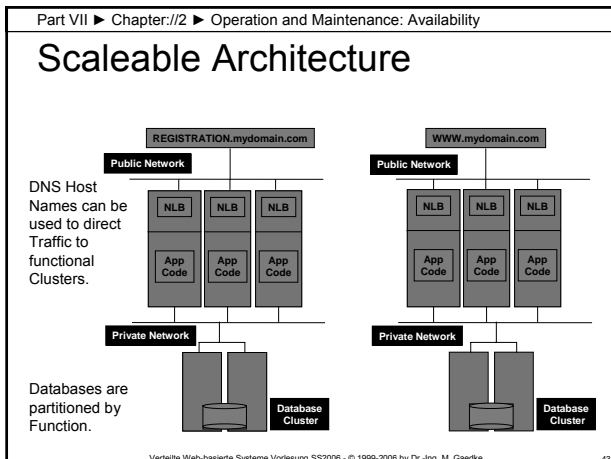
41

Part VII ► Chapter://2 ► Operation and Maintenance: Availability

Technical Approach: NLB

- ☉ NLB = Network Load Balancing
 - E.g. NLB-Service or Windows Load Balancing Service (WLBS)
- ☉ Generally used for scalability
- ☉ Can be used with databases
 - Front end switch for log shipping role change
 - Warm standby server
 - Protect analysis services

42



Verteilte Web-basierte Systeme – SS 2006

Section://4

Maintenance

Part VII ► Chapter://2 ► Operation and Maintenance: Maintenance

Maintenance

- ☉ Web applications are like “living entities”
 - Like a Garden: Must be maintained to look nice.
- ☉ **Maintenance is any event that yields to a new iteration of the life cycle** (of a feature or the application as a whole)
 - **Often you will find: Maintenance** – Any development activity performed to modify or fix the Web application after it has been completed or reached some final milestone. Be aware of ad-hoc maintenance (code-and-fix approach)!
- ☉ Reasons for Maintenance
 - CR: Content, delivery / access, functionality
 - Maintenance is good – it is the beginning of evolution!

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: Maintenance

Aspects of Maintenance - I

- ☉ Content Maintenance (Main Activities)
 - Note: Never work on a live site! Use a staging approach
 - E.g. Content stored in Database and is easily manipulated using dedicated Editors. Reviewed Content is updated on Production Server
- ☉ Delivery Maintenance (Success Disaster)
 - “Perfect” Web application fails – if traffic increases dramatically
 - Prepare for scalability and availability

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: Maintenance

Aspects of Maintenance - II

- ☉ Functionality Maintenance (Crisis Management)
 - **Corrective** – Activities to fix application bugs and design flaws
 - **Adaptive** – Activities to make application work for a “Problem-Browser” configuration
 - **Perfective** – Activities to increase functionality (feature additions)
- ☉ Note:
 - If too many Bugs hinder the functionality and can not be solved in a few minutes:
 - **Provide a currently under maintenance page**

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part VII ► Chapter://2 ► Operation and Maintenance: Maintenance

Tools for Maintenance

- ☉ Monitoring
 - Web application statistics terminology
 - Logs of server, router, etc.
 - Use: Log Analysis Software
 - Your own unit tests
- ☉ Feedback Channels
 - E.g. contact information, forms for user feedback etc.

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Verteilte Web-basierte Systeme – SS 2006

Chapter://3

Further Readings

Part VII ► Chapter://3 ► Further Readings

Literature

- ⊗ Chapter 8: Thomas A. Powell, *Web Site Engineering*, Prentice Hall PTR
- ⊗ Chapter 19-24: Ian Sommerville, *Software Engineering*, Addison-Wesley
- ⊗ Chapter 11: I. Jacobson, G. Booch, J. Rumbaugh, *The Unified Software Development Process*, Addison-Wesley, 1999

=====

Further information available at **Lecture Web Site**

=====

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

50

Part VII ► Chapter://3 ► Further Readings

Important Links

- ⊗ IBM's Ease of Use
 - = http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/558
 - = http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/609
- ⊗ Jakob Nielsen's web usability website
 - = <http://www.useit.com/alertbox/>
- ⊗ Microsoft Usability Home Page
 - = <http://www.microsoft.com/usability/>
- ⊗ Sun's Usability Testing of Web Concepts:
 - = <http://www.sun.com/980113/sunonnet/concepts.html>
- ⊗ User Interface Engineering
 - = <http://world.std.com/~uieweb>

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

51