

Middleware and Web Services

Motivation and Course Overview

doc. Ing. Tomáš Vitvar, Ph.D.

tomas@vitvar.com • @TomasVitvar • <http://vitvar.com>



Czech Technical University in Prague

Faculty of Information Technologies • Software and Web Engineering • <http://vitvar.com/courses/mdw>



Modified: Tue Apr 11 2017, 10:14:38
Humla v0.3

Hellos

- Tomáš Vitvar
 - *Web Engineering curricula chair, CVUT FIT*
 - *Research - Web Intelligence, Semantic Web, LinkedData, Web Services*
 - *Consulting Technical Director at Oracle*
 - *SOA and Oracle Integration Architectures Consulting Projects (IKEA IT, Vodafone UK, MTN ZA)*
- Jaroslav Kuchař, Milan Dojčinovski
 - *research assistants, Ph.D. students*
 - *Web services, data mining, analytics, Web technologies*

Motivation in Brief

- Systems rely on complex infrastructures
 - *A lot of data and many processes, internal and external*
 - *As people communicate, underlying systems must too*
 - *But:*
 - *variety of data formants, technologies, protocols*
 - *variety of architectures, client-server, peer-to-peer, ...*
- Good performance
 - *frequent changes in applications' loads, peak hours*
 - *scalability – effective load balancing*
 - *low costs – cheaper to outsource?*
- Rapid changes in applications' functionality
 - *modular development*
 - *reuse of application functionality*
 - *low costs – do it now and quickly!*

Spaghetti Architecture

- Need for the integration
 - *One-to-one integration*
 - *Hard to maintain, vendor interoperability problem*



SOA Architecture

- Integration organized
 - *Enterprise Service Bus, to be used wisely*



Scope

- Architectural and conceptual basis
 - *What is an architecture – enterprise, processes, data, software*
 - *Service Oriented Architecture, Service Concepts, Middleware, ESB*
- Web Service technologies
 - *Web Service Description Language, SOAP*
 - *Process languages – BPEL/BPMN*
 - *Communication patterns – synchronous, asynchronous, decoupling*
- Middleware
 - *Application server*
 - *Middleware technology for SOA*
 - *Performance, Scalability*

Overview

- Course at a Glance
 - *Motivation and Scope*
 - *Requirements and Organization*
- Assessment
- Communication and Resources

Prerequisites

- Object-oriented programming
 - *Principles*
 - *class, object, inheritance, encapsulation, ...*
 - *basis for service concepts*
- Java
 - *All code examples will be in Java*
 - *All lab work will be in Java*
- Web Architecture
 - *Basics of XML, XPath, HTTP, URI*
- Other
 - *Networking*
 - *Regular expressions, basics of Linux*

Organization of Lectures

- 13 Lectures
 - Czech: Mon 7:15-8:45, T9:155
 - English: TBA
- Plan
 1. 05.10.2015 – Motivation and Course Overview ([html](#))
 2. 05.10.2015 – Introduction to Application Server ([html](#))
 3. 12.10.2015 – Introduction to Architectures ([html](#))
 4. 19.10.2015 – Application Protocols ([html](#))
 5. 26.10.2015 – Application Server Services ([html](#))
 6. 02.11.2015 – Messaging Systems ([html](#))
 7. 09.11.2015 – High Availability and Performance ([html](#))
 8. 16.11.2015 – Service Concepts ([html](#))
 9. 23.11.2015 – SOAP and REST ([html](#))
 10. 30.11.2015 – Web Service Description Language ([html](#))
 11. 07.12.2015 – Enterprise Service Bus ([html](#))
 12. 14.12.2015 – Service Orchestration ([html](#))
 13. 04.01.2016 – Reserve/Exam

Organization of Labs

- Individual work (no teams!)
- Labs every second week
- Number of labs: 5
 1. Introduction, Setup
 2. WebLogic Server, application
 3. Service design, implementation, wsdl, soap, soapui
 4. Oracle Service Bus, Web service integration
 5. Interoperability, transformation

Methodology for Lab Work

- No app development, not directly related assignments
 - *assignment every second week*
 - *be prepared for the lab!*
 - *work alone, ask others for advices*
 - **Results:**
 - *5 completed tasks*
 - *documentation (in the wiki)*
 - *implementation (code in the source tracker)*

Development Platform

- WebLogic Server
 - *JEE development environment*
- Oracle Service Bus
 - *Oracle Middleware platform*
 - *Runs on WebLogic Server*
 - *You use scripts to install it and run it*

Overview

- Course at a Glance
- **Assessment**
- Communication and Resources

Assessment

- Labs
 - Presence is mandatory
 - You can miss up to 1 lab without sending regrets
 - Every task gives you a max. of 6 points
 - $6 \cdot 5 = 30$ points
 - Activity in labs gives you a max. of 10 points
 - Total maximal points: $p_p = 40$, **to pass**: $p_p \geq 20$
- Final exam
 - Written exam: 3 exercises, 1 hour
 - each gives you a max. of 20 points, the total $p_t = 60$ points
 - To pass, you need to have at least 50% from each exercise!
 - Final score:
 - $p_p + p_t = 100$ maximum points

Final Marks

Mark	Points	In words
A	100–90	výborně
B	89–80	velmi dobře
C	79–70	dobře
D	69–60	uspokojivě
E	59–50	dostatečně
F	49–0	nedostatečně

Source: <http://www.cvut.cz/pracoviste/pravni-odbor/dokumenty/studijni-predpisy/studijnirad.pdf>

Overview

- Course at a Glance
- Assessment
- Communication and Resources

Communication

- Language
 - Text: English (slides, tweets, posts, instructions, etc.)
→ choose English/Czech for your contributions to the wiki
 - Voice: Czech and English (English version of the course)
- Direct
 - you can always contact me directly at tomas@vitvar.com or [@TomasVitvar](https://twitter.com/TomasVitvar)

Overview of Resources

- Overview of resources

Item	URL	Feed
Course page	http://vitvar.com/courses/mdw	-
EDUX	http://edux.fit.cvut.cz/courses/MI-MDW	-
Lab project	https://gitlab.fit.cvut.cz	-
Bookshelf	http://vitvar.com/courses/mdw/bookshelf	RSS

- Books
 - Thomas Erl: *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, Aug 2, 2005.

About Slides

- Humla – Open Source HTML5 Presentation Environment
 - *every slide has a unique URL*
 - *all figures linked with Google drawings*
 - *possible to format and print in PDF*
 - *running local, with back-end NodeJS support, and offline*
 - *Fork it at [Humla github repo](#)*
- Keys
 - default browsing mode*
 - slideshow mode (automatically scales to fullscreen)*
 - grid (overview) mode*
 - print mode, 2 slides per page*
 - slide left*
 - slide right*
 - debug mode*
 - toggle last error messages on/off*