SOFTWARE DESIGN AND ARCHITECTURE

LECTURE 15

SlidePlayer 1 / 25

SOFTWARE DESIGN AND ARCHITECTURE LECTURE 15. Review Interaction-Oriented Software Architectures – MVC.

112

Published by Jeffery Elliott

Modified over 7 years ago

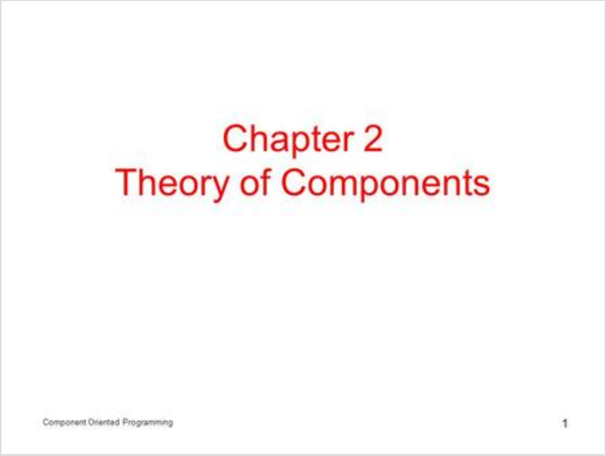
Embed Download presentation

Similar presentations

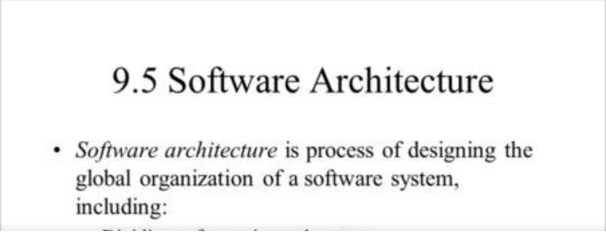
REST Introduction

吴海生 博克软件(杭州)有限公司

COM vs. CORBA



Chapter 2
Theory of Components



9.5 Software Architecture

- *Software architecture* is process of designing the global organization of a software system, including:

Presentation on theme: "SOFTWARE DESIGN AND ARCHITECTURE LECTURE 15. Review Interaction-Oriented Software Architectures – MVC."— Presentation transcript:

1 SOFTWARE DESIGN AND ARCHITECTURE LECTURE 15

2 Review Interaction-Oriented Software Architectures – MVC

3 Outline Component Based Architecture

4 COMPONENT BASED SOFTWARE ARCHITECTURE

5 Component Based Software Architecture Component-based software architecture divides a problem into sub-problems each associated with component partitions. The interfaces of the components play important roles in the component-based design. The main motivation behind component-based design is component reusability.

6 Component Based Software Architecture Designs can make use of existing reusable commercial off-the-shelf (COTS) components or ones developed in-house, and they may produce reusable components for future reuse. This increases overall system reliability since the reliability of each individual component enhances the reliability of the whole system via reuse.

7 Component A component is a deployable software package that can provide services to its clients; – it may also itself require services from other components. A component remains self-contained and substitutable as long as its interface is unchanged.

8 Component Component is a higher level of abstract concept than class. A component may consist of many related classes stored and deployed in the same binary assembly.

9 Component The interface of a component plays the role of contract between its implementation and its clients; hence the approach is also called a contract-oriented software architecture. The first step in component-based software design is to identify firstcut components and specify all necessary connections among these components. The connections are conducted via the interfaces of components.

10 N tier Architecture

12 Components can be made of Source code – Classes – one or more, possibly related Executable code – Object code – Virtual object code Other files – Images, text, indices

13 Connectors Connectors connect components, specifying and ruling their interaction. Component interaction can take the form of – method invocations, – asynchronous invocations such as event listener and registrations, broadcasting, – message-driven

also additionally satisfy global constraints called invariants Non-functional aspects (different constrains, environment requirements, etc.)

18 Component Model A component model is a definition of standards for component implementation, documentation and deployment – Examples of component models EJB model (Enterprise Java Beans).NET model Corba Component Model The component model specifies how interfaces should be defined and the elements that should be included in an interface definition Different application domains have different needs for component-based systems – Different non-functional properties: performance, security, reliability, scalability, etc.

19 Elements of a component model

21 Applicable domains of component- based architecture Applications where the interface contracts between subsystems are clear Applications that require loose coupling between the components and where many reusable components are available Suitable for the class library system organization (.NET class library and Java API are built in component architecture)

22 Benefits Reusability of components. System maintenance and evolution; easy to change and update the implementation without affecting the rest of the system. Independency and flexible connectivity of components.

23 Benefits Independent development of components by different groups in parallel. Productivity for the current and future software development. Many OO design tools can also be used for component-based software development.

24 Limitations It can be difficult to find suitable available components to reuse. Adaptation of components is an issue. Few component-oriented dedicated design tools are available.

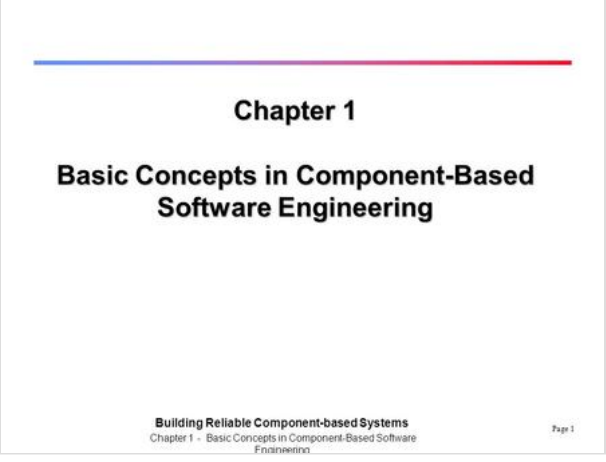
25 Summary Component oriented Software Architecture

Download ppt "SOFTWARE DESIGN AND ARCHITECTURE LECTURE 15. Review Interaction-Oriented Software Architectures – MVC."

Applied Science Private University
جامعة العلوم التطبيقية الخاصة
ASU Open Course System
Component-Based Software Engineering
Basic Concepts

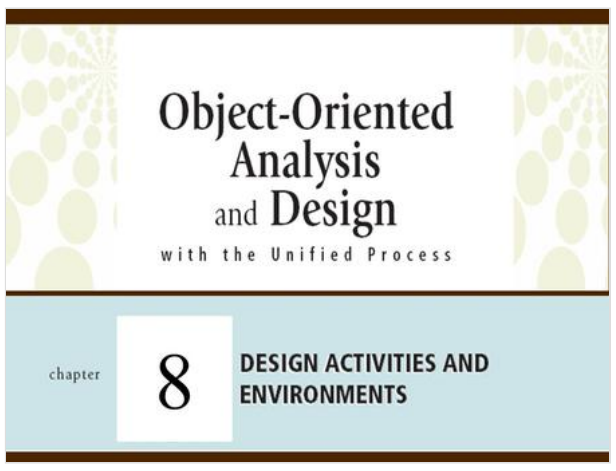
Dr. Eman M. Saleh Al-Maghary
Email: e_saleh@asu.edu.jo – Ext. 1269

SE Department
Faculty of Information Technology
http://fit.asu.edu.jo – http://OpenCourse.asu.edu.jo
Copyright © 2011 Applied Science University. All Rights Reserved
Jordan - Amman Tel:5609999 Fax:5232899 P.O.Box : 166 Email:info@asu.edu.jo

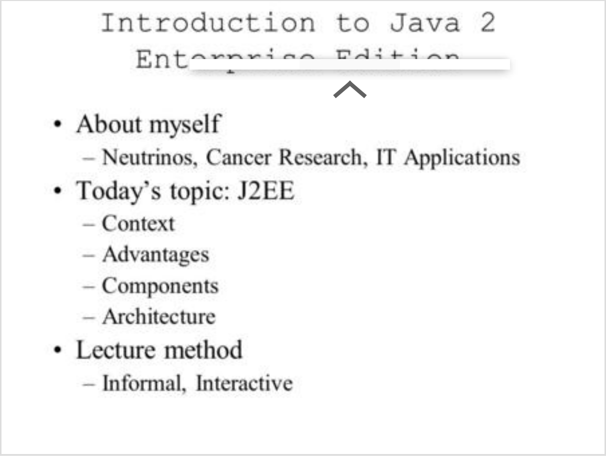

Chapter 1

Basic Concepts in Component-Based Software Engineering

Building Reliable Component-based Systems
Chapter 1 - Basic Concepts in Component-Based Software Engineering
Page 1


Object-Oriented Analysis and Design
with the Unified Process

chapter 8 DESIGN ACTIVITIES AND ENVIRONMENTS

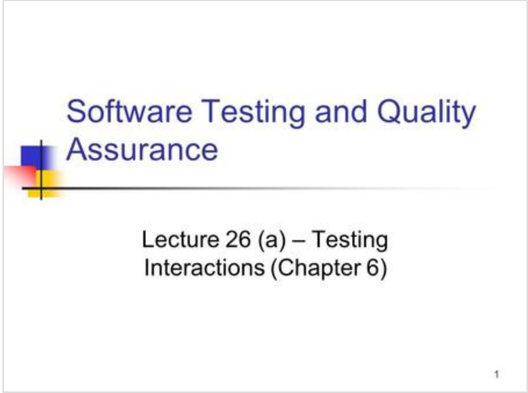

Introduction to Java 2 Enterprise Edition

- About myself
  - Neutrinos, Cancer Research, IT Applications
- Today's topic: J2EE
  - Context
  - Advantages
  - Components
  - Architecture
- Lecture method
  - Informal, Interactive

Similar presentations

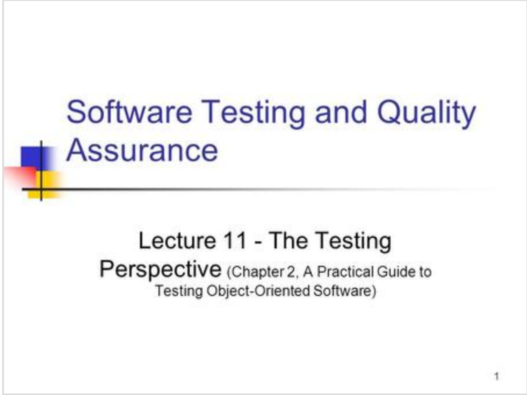
Moving from Analysis to Design

Software Connectors


Software Testing and Quality Assurance

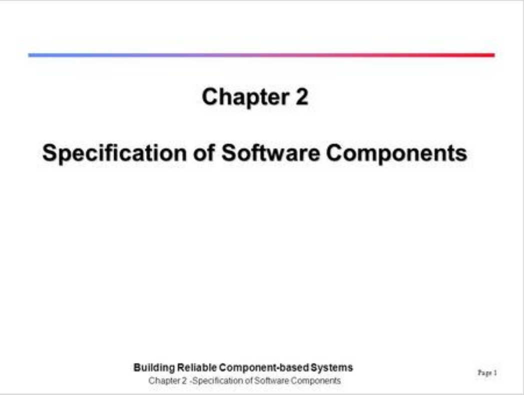
Lecture 26 (a) – Testing Interactions (Chapter 6)

1


Software Testing and Quality Assurance

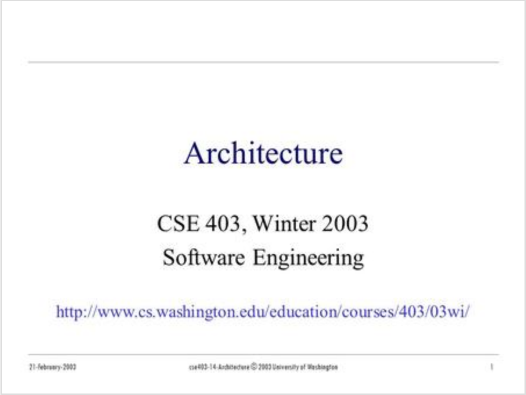
Lecture 11 - The Testing Perspective (Chapter 2, A Practical Guide to Testing Object-Oriented Software)

1


Chapter 2

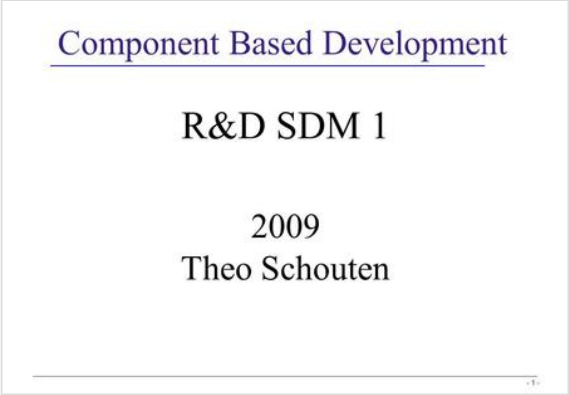
Specification of Software Components

Building Reliable Component-based Systems
Chapter 2 :Specification of Software Components
Page 1


Architecture

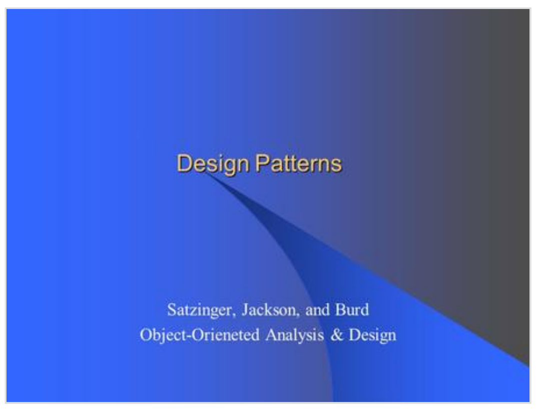
CSE 403, Winter 2003
Software Engineering

http://www.cs.washington.edu/education/courses/403/03wi/

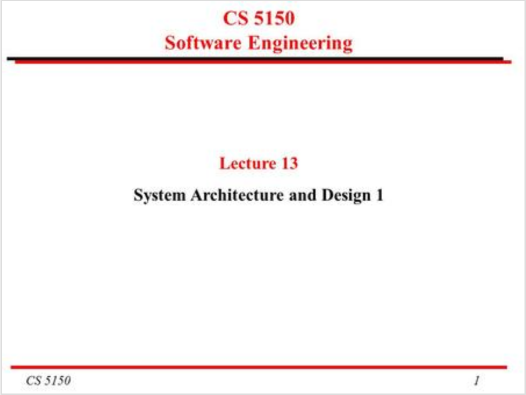

Component Based Development

R&D SDM 1

2009
Theo Schouten

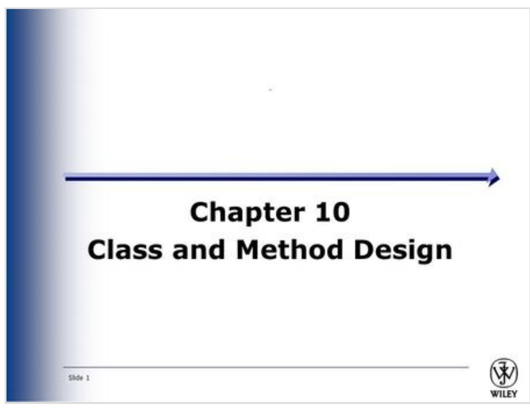

Design Patterns

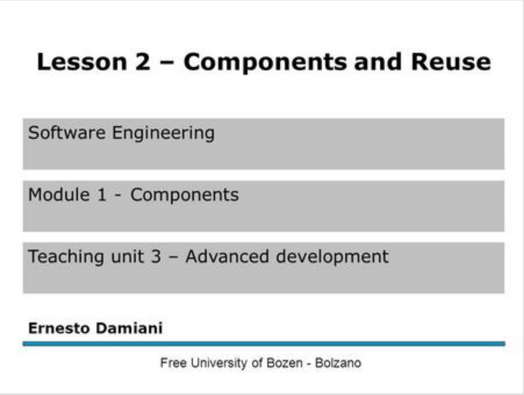
Satzinger, Jackson, and Burd
Object-Orieneted Analysis & Design


CS 5150
Software Engineering

Lecture 13
System Architecture and Design 1

CS 5150 1


Chapter 10
Class and Method Design


Lesson 2 – Components and Reuse

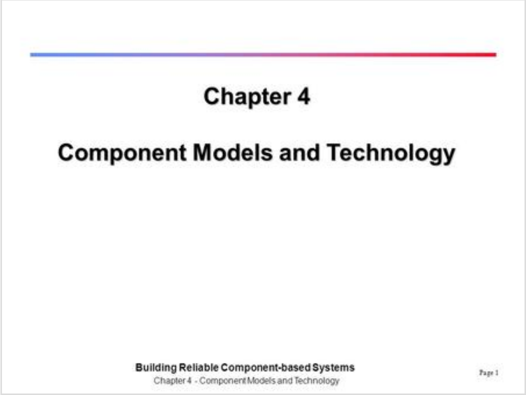
Software Engineering

Module 1 - Components

Teaching unit 3 – Advanced development

Ernesto Damiani

Free University of Bozen - Bolzano

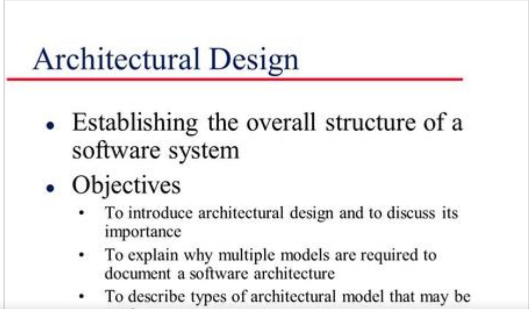

Chapter 4

Component Models and Technology

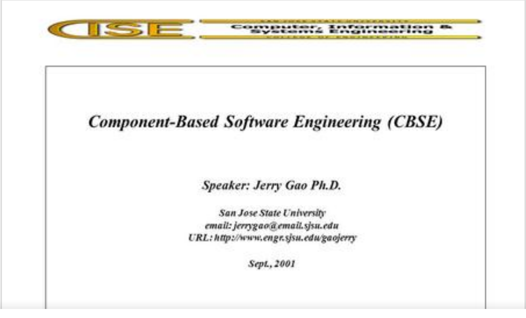
Building Reliable Component-based Systems
Chapter 4 : Component Models and Technology
Page 1

Logical Architecture and UML Package Diagrams

Chapter 13
Applying UML and Patterns


Architectural Design

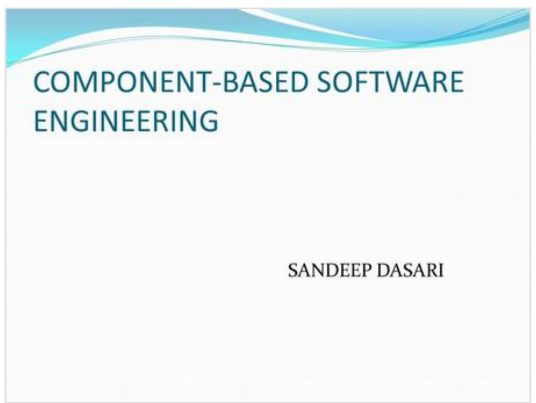
- Establishing the overall structure of a software system
- Objectives
  - To introduce architectural design and to discuss its importance
  - To explain why multiple models are required to document a software architecture
  - To describe types of architectural model that may be


Component-Based Software Engineering (CBSE)

Speaker: Jerry Gao Ph.D.

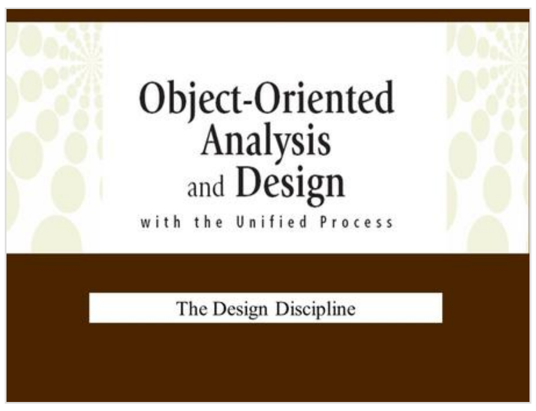
San Jose State University
email: jerrygao@email.sjsu.edu
URL: http://www.engr.sjsu.edu/gaojerry

Sept, 2001



COMPONENT-BASED SOFTWARE ENGINEERING

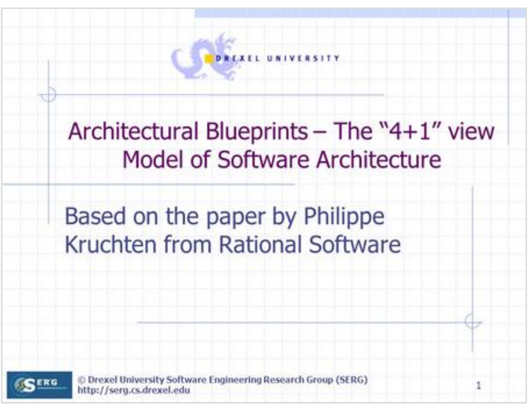
SANDEEP DASARI



Object-Oriented Analysis and Design

with the Unified Process

The Design Discipline



Architectural Blueprints – The "4+1" view Model of Software Architecture

Based on the paper by Philippe Kruchten from Rational Software

© Drexel University Software Engineering Research Group (SERG)
http://serg.cs.drexel.edu

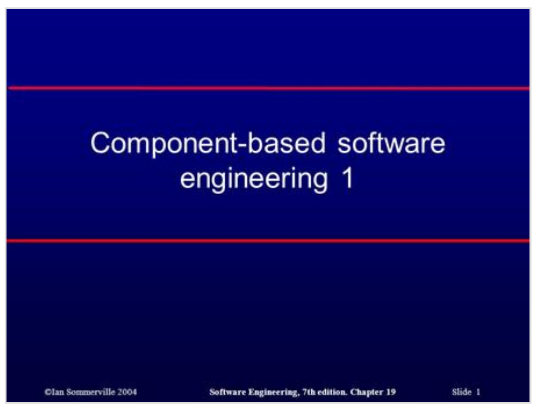
1

Logical Architecture and UML Package Diagrams

徐迎晓
xuyingxiao@126.com
http://javabook.126.com
复旦大学软件学院

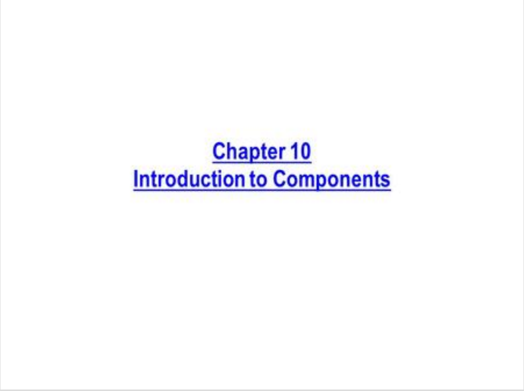
Quality Assurance for Component-Based Software Development

Cai Xia (Mphil Term1)
Supervisor: Prof. Michael R. Lyu
5 May, 2000

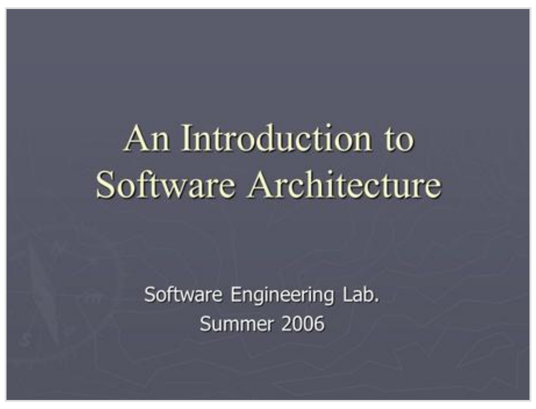


Component-based software engineering 1

©Ian Sommerville 2004 Software Engineering, 7th edition. Chapter 19 Slide 1



Chapter 10
Introduction to Components



An Introduction to Software Architecture

Software Engineering Lab.
Summer 2006

CSE 303 – Software Design and Architecture

LECTURE 03