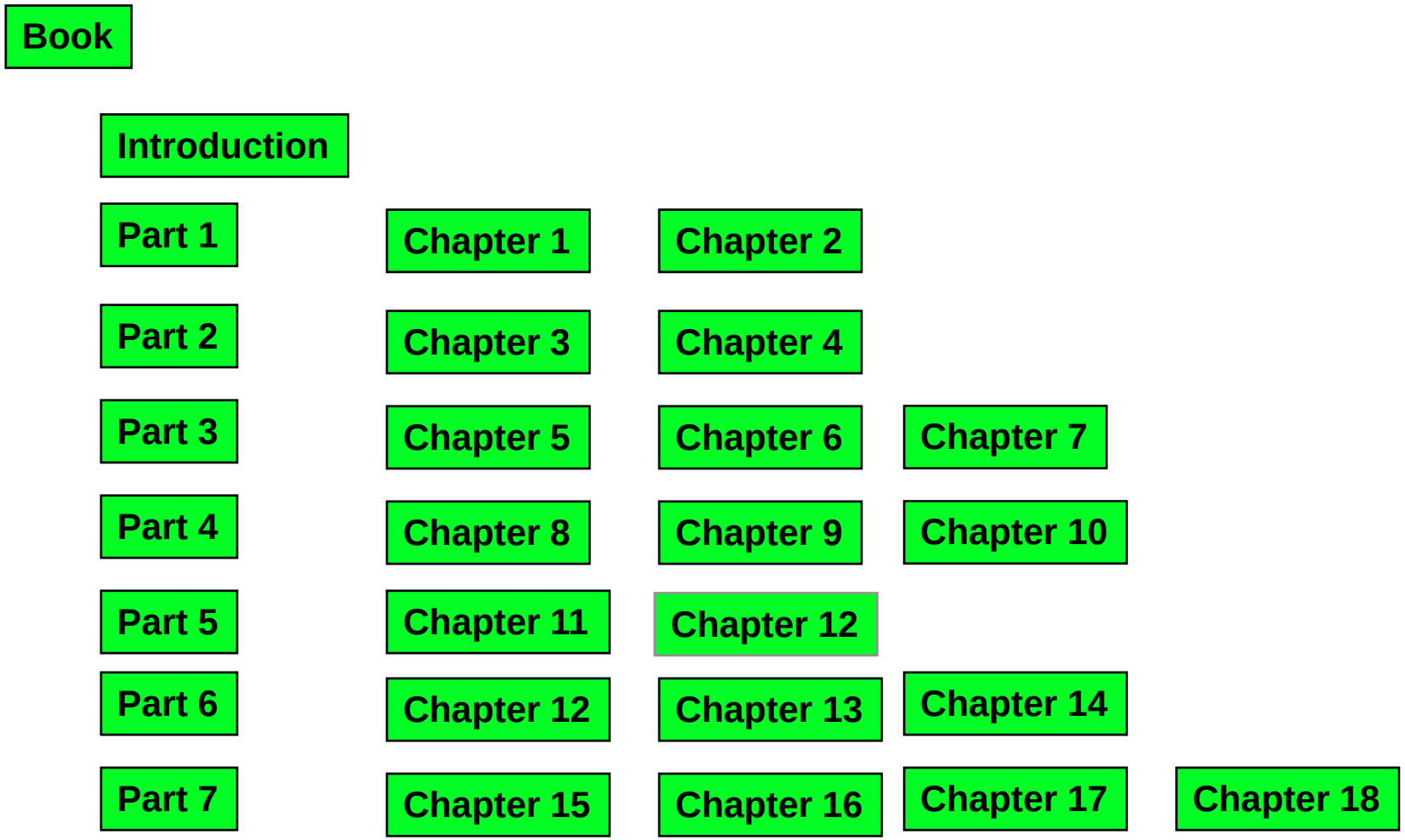
Component-Based Software Engineering
Building reliable component-based systems

Overview

www.idt.mdh.se/cbse-book

The Book Organization



The Book Organization

Part 1	Basic definitions and concepts of component specification
Part 2	Component models and architecture
Part 3	Developing Software Components (development process)
Part 4	Using components (evaluation, test, composition)
Part 5	Software Product-Lines
Part 6	Real-time components, real-time systems and components
Part 7	Case studies – real–time, industrial and office component-based systems

The Book Organization II

Part 1	Component	Theoretical parts (definitions)	General for components
Part 2			
Part 3	Compositions	Processes	dependability/reliability
Part 4			
Part 5	Systems	Domain specific processes	Real-time requirements Safety-critical requirements
Part 6			
Part 7	Complex systems	Case studies	Specific cases

PART 1 CONCEPTS OF CBSE

What is a software component?

Chapter 1

Basic Concepts in CBSE

- How to define a component?
- Component specification
- Interface
- Contracts
- Patterns
- Frameworks

Chapter 2

Specification of Software Components

- Component Specification
- Interface Specification
- UML Component Specification
- Specifying extra functional components

Part 1 - Interesting questions

- **What are the relations between objects and components? How does it work in different technologies?**
- **What are component frameworks in different component models?**
- **Which types of design patterns can be implemented as components. Examples of some patterns and their implementation in a component technology**
- **How are interfaces implemented in different technologies?**
- **UML and component specification (UML components)**
  - In particular UML 2.0

PART 2 - SOFTWARE ARCHITECTURE AND COMPONENTS

Chapter 3
Architecting Component-based Systems

- Relation between Software architecture and CBSE
- Architecture – design issue – identify components
- Redesign system depending of components availability
- Type of components
  - "architectural" components
  - already existing components

Chapter 4
Component Models and Technology

- Different component technologies from architectural point of view
- ADL (architectural definition language)
- JavaBeans, CORBA Component model, .NET Component Model and "Open Service Gateway Initiative" Component Model

Part 2- Interesting questions

- **Dynamic software architecture**
  - Dynamic replacement of components
  - Dynamic restructuring of resources
- **Different ADLs and their relations to components**
- **UML 2.0**
- **Containers and Frameworks in different technologies**

Part 3 - Developing Software Components

Chapter 5

CBD Life-Cycles

- **System & Application development**
- **Separation of processes**
  - Component development
  - Component-based development
- **Different phases and emphasize on parts specific for CBD**

Chapter 6

Semantic Integrity in Component - based Development

- **Importance of semantics**
- **Different levels of semantic specifications**
- **Addressing semantic questions in CBSE literature – a statistic survey**

Part 3 - Developing Software Components

Chapter 7

Role-Based Component Engineering

- Relations class/object – role – framework – components
- Role – parts of interface having a particular “role” in a framework together with other components
- How a role is implemented in OO languages?

Part 3- Interesting questions parts

- **Component-based processes**
  - Component-based databases – problems and examples
  - How and when to test components
  - Component documentation
  - Component certification
- **Semantic integrity**
  - UML, OCL and specification of pre- and post-conditions
- **Role-Based components**
  - Component Frameworks and Roles

Part 4 - USING SOFTWARE COMPONENTS

Chapter 8

Dispelling the Myth of Component Evaluation

- How to evaluate and select components?
- What should we evaluate? Components or component compositions?
- How component properties behave in compositions?

Chapter 9

Component Composition and Integration

- Integration – putting components together (complied to component models)
- Composition – reasoning about compositions attributes
- Predictable assembly from "certificated" components

Part 4 - USING SOFTWARE COMPONENTS (cont)

Chapter 10

Predicting System Trustworthiness from Software Component Trustworthiness

- Predictable assembly
- Can be predict reliability of a composition from reliability from components
- How to test assemblies?
- Fault injection method – Interface Propagation Analysis - send invalid data between connected components

Part 4- Interesting questions

- **Component evaluation**
  - Component repositories
  - Component documentations
  - Automatic test of components
- **Fault injection models**
  - Managing exception handlings in components
- **Component and system properties**
  - Reliability, Safety, Security, etc.
  - Experience from hardware systems and components

Part 5 - SOFTWARE PRODUCT-LINES

Chapter 11

Components in product line architectures

- **What is "Software product lines"**
- **How to make reusable parts in in-house development for different families of products?**

Chapter 12

- **KOALA –component model implemented at Philips**
  - Requirements, model architecture, interface definitions, experience

Part 5- Interesting questions/additional parts

- **Software product lines**
  - Overview
  - Process challenges – how to develop platforms
  - What is a component in PLA
  - Platform-based development
  - Configuration Management and PLA
- **Integration principles**
  - Type of bindings/compositions (functions, libraries, shared libraries, dynamic binding,…)

Part 6- REAL-TIME SOFTWARE COMPONENTS

Chapter 13

Components in Real-Time Systems

- Real-time requirements
- Real-time components (OS)
- Designing real-time component-based systems
- Reusing RT components

Chapter 14

Test of Reusable Software Components in Safety-Critical Real-Time Systems

- Safety-critical systems
- Large costs for testing
- Can we reuse components?
- What is the minimum of tests we must repeat

Part 6- REAL-TIME SOFTWARE COMPONENTS (cont)

Chapter 15

Providing real-time services for COTS components

- Using non-real time system (Windows NT) for real time application
- Can we adjust non-real time systems to use it as a real-time component
- RT characteristics of Windows NT
- Adding a new RT component – what is the behavior of the entire system

Part 6- Interesting questions

- **RT components**
  - Interface required for RT components
  - RT component components
  - RT CORBA
  - Timing aspects of using indirect (or dynamic) bindings
  - Why component models such as COM, CORBA cannot be used in hard RT systems?
  - RT UML - overview
- **Testing safety-critical components**
  - What is a reliability of a component (hardware /software comparison)
  - Dependability of components in relation of dependability of systems

Part 7 - CASE STUDIES

COMPONENT-BASED DEVELOPMENT IN INDUSTRIAL APPLICATIONS

Chapter 16

Component-Based Embedded Systems

- How to use components in small embedded systems?
- Which component model to use?
- Which composition environment?
- Which run-time environment?
- Case study – ABB embedded systems

Chapter 17

Architectural Support for Reuse: A Case Study in Industrial Automation

- ABB's next generation of automation system architecture
- AspectObjects
- Aspect directories
- Flexibility in integration and data mining

Part 7- Interesting questions

- **Embedded systems and component-based systems**
  - Identification of configuration environment/framework and run-time environment
  - OS for embedded systems and possibility of using CBSE for them (example Rubus)
  - OPC overview
  - Interesting services of a RT component-based framework

Part 7 - CASE STUDIES (cont.)

COMPONENT-BASED DEVELOPMENT IN INDUSTRIAL APPLICATIONS

Chapter 18

A Framework for Integrating Business Applications

- Similar to chapter 16, but standards used (Microsoft)
- Office Information systems
- Issue Management Systems
- Integration of large applications
- From different applications to common Interface

Chapter 19

Industrial Experience with the Dassault Système Component Model

- Reuse, dynamic configuration of applications (CAD/CAM)
- Internal component model
- Problems and experiences