



Department of Computer Science Institute for Software and Multimedia Technology, Software Technology Group

Component-Based Software Engineering

Cheesman/Daniels Process

Exercise Friday, 5. DS, APB/E023 Thomas Kühn (thomas.kuehn3@tu-dresden.de)





UML Components: A Simple Process for Specifying Component-based Software (Component Software Series)

John Cheesman, John Daniels Addison-Wesley Longman, Amsterdam, November 2000 ISBN: 978-0201708516





1) Business Concept Model

- Domain model
- Class-diagram (No operations, no data types)

2) Use-Case Model

- Use-Case diagram
- Identify Use-Case steps
- Derive system interfaces





1) Business Type Model

- Class Diagram (derived from Business Concept Model)
- Add Types stereotypes

2) Business Object Interface Model

- From Business Type Model
- Identify business interfaces

3) Component Specification

- Allocation of business object interfaces to components
- Added Lollipop notation

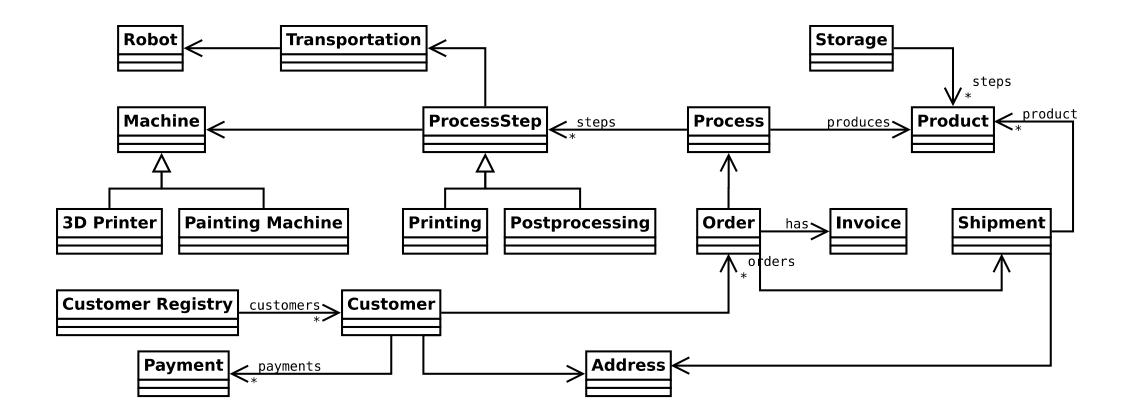
4) Component Architecture

- UML Component diagram (derived from Component Specification)
- Connect provided/required ports





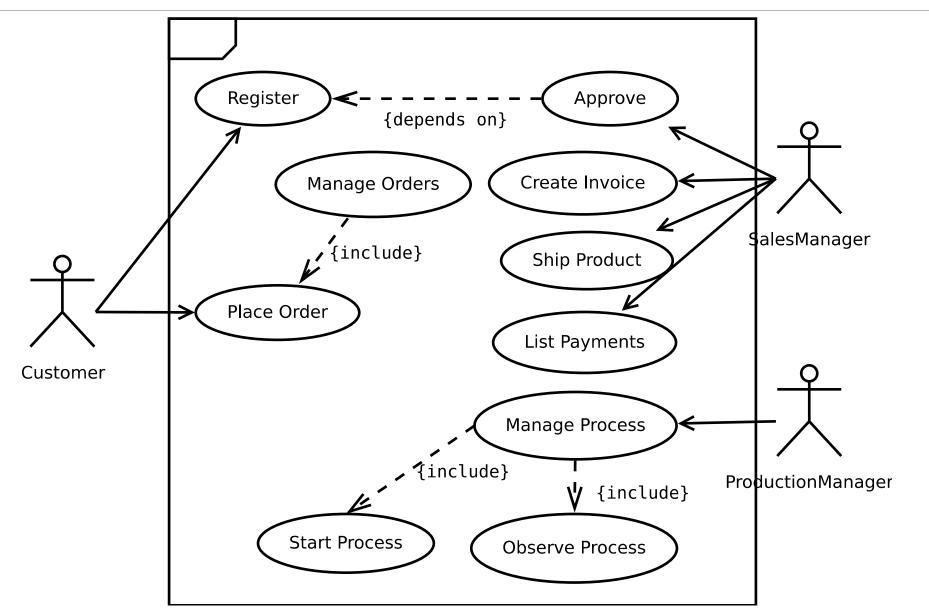
Cheesman/Daniels Process Business Concept Model





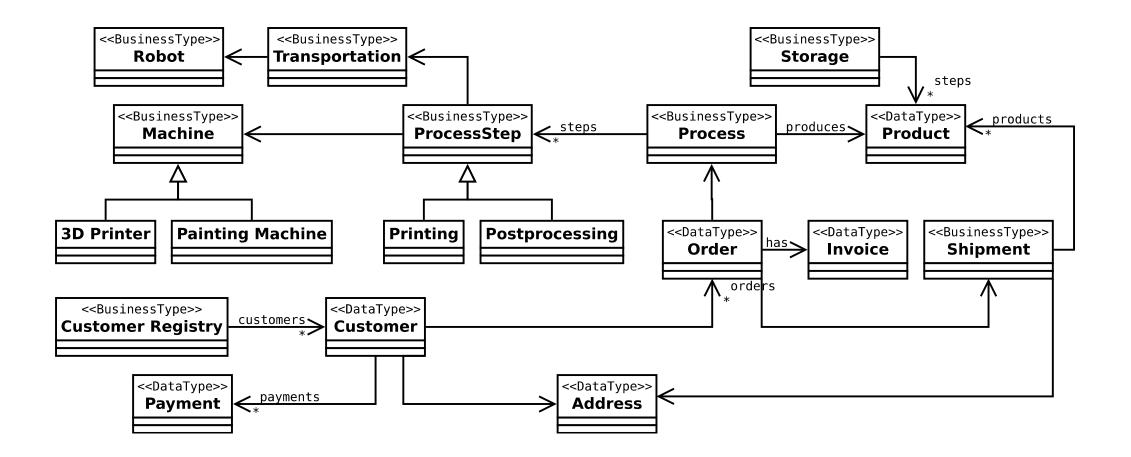


Cheesman/Daniels Process Use Case Diagram





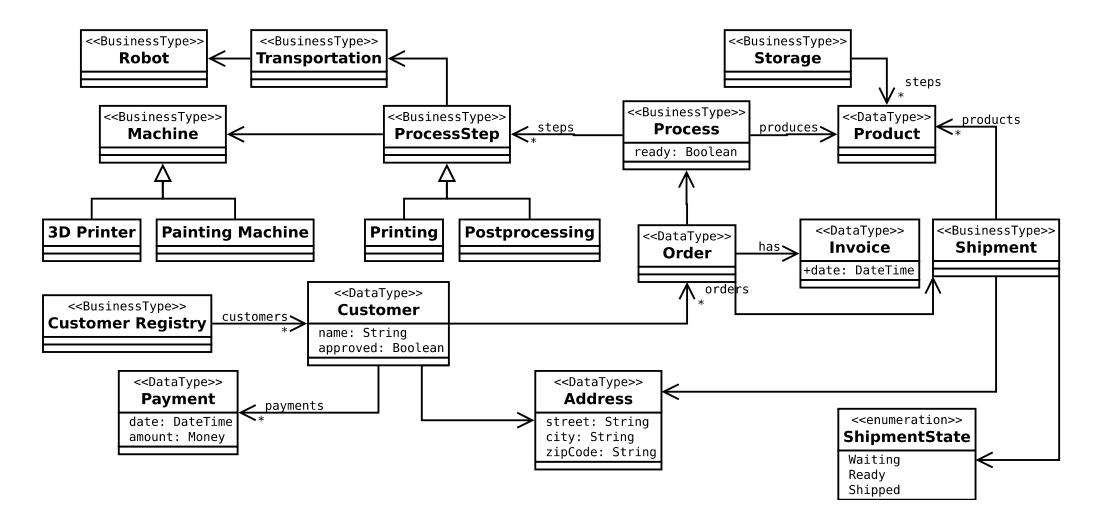
Cheesman/Daniels Process Business Type Model







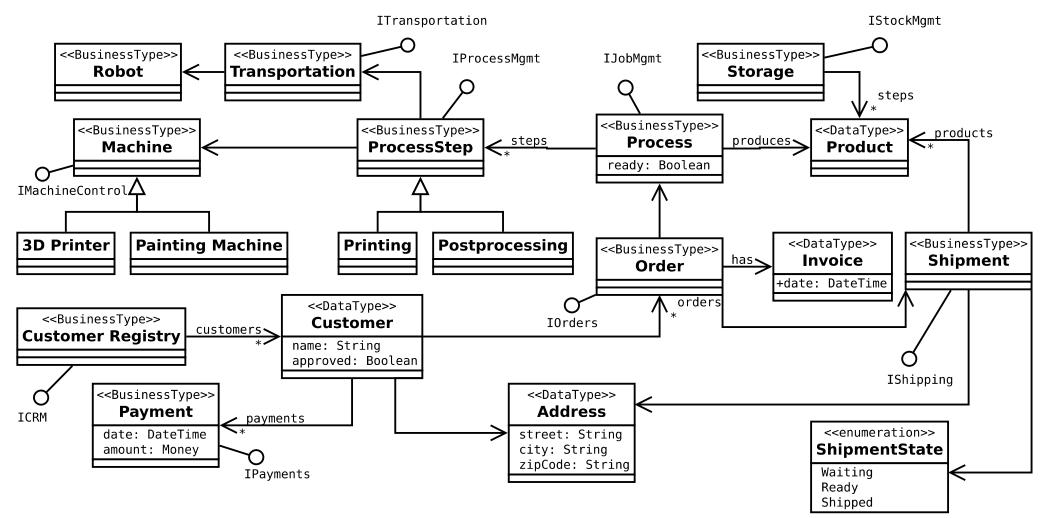
Cheesman/Daniels Process Business Type Model+Attributes







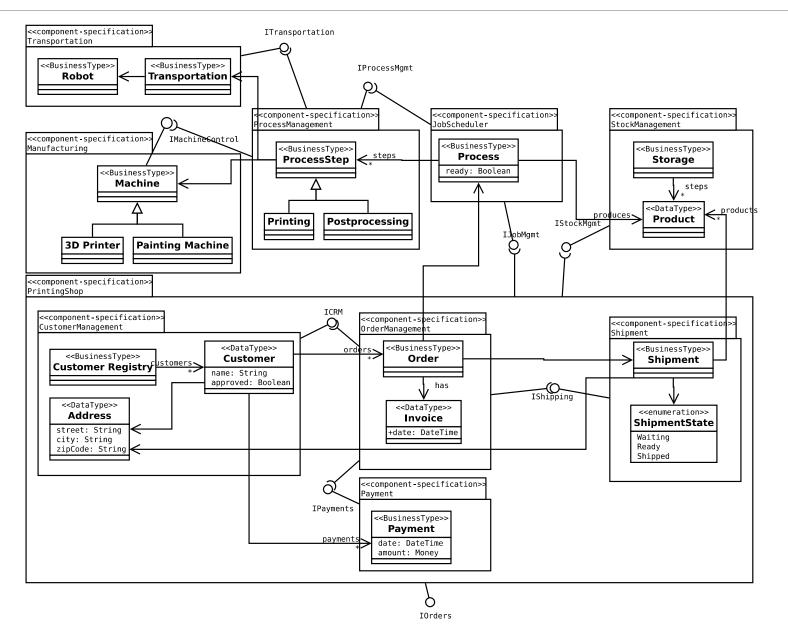
Cheesman/Daniels Process Business Object Interface Model







Cheesman/Daniels Process Component Specifications

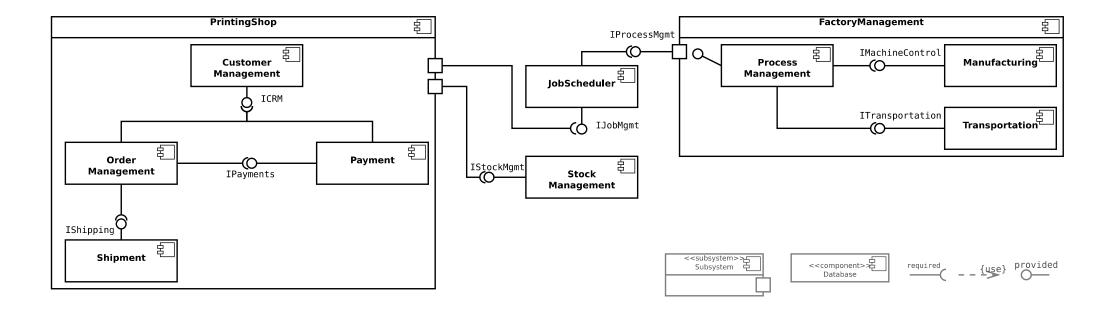


Software Technology Group

10/12



Cheesman/Daniels Process Component Architecture

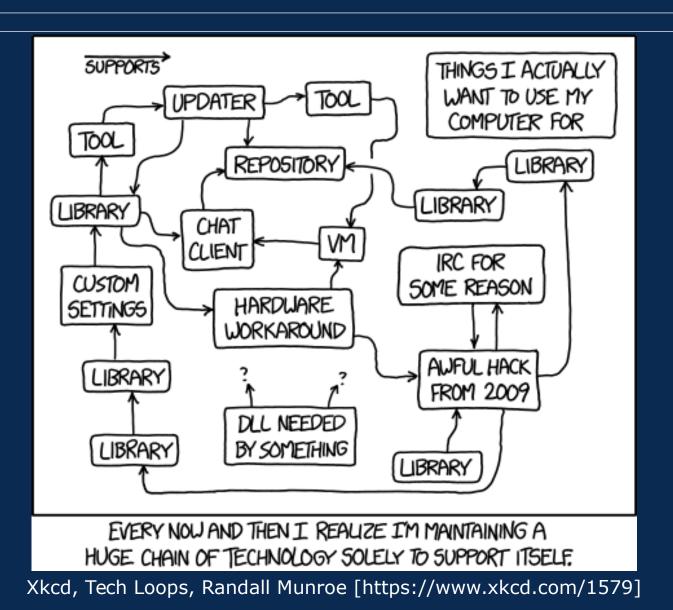






CBSE Cheesman/Daniels Process





12 / 12