Componentware - The Big Picture

Klaus Bergner, Andreas Rausch, Marc Sihling

{bergner|rausch|sihling}@in.tum.de

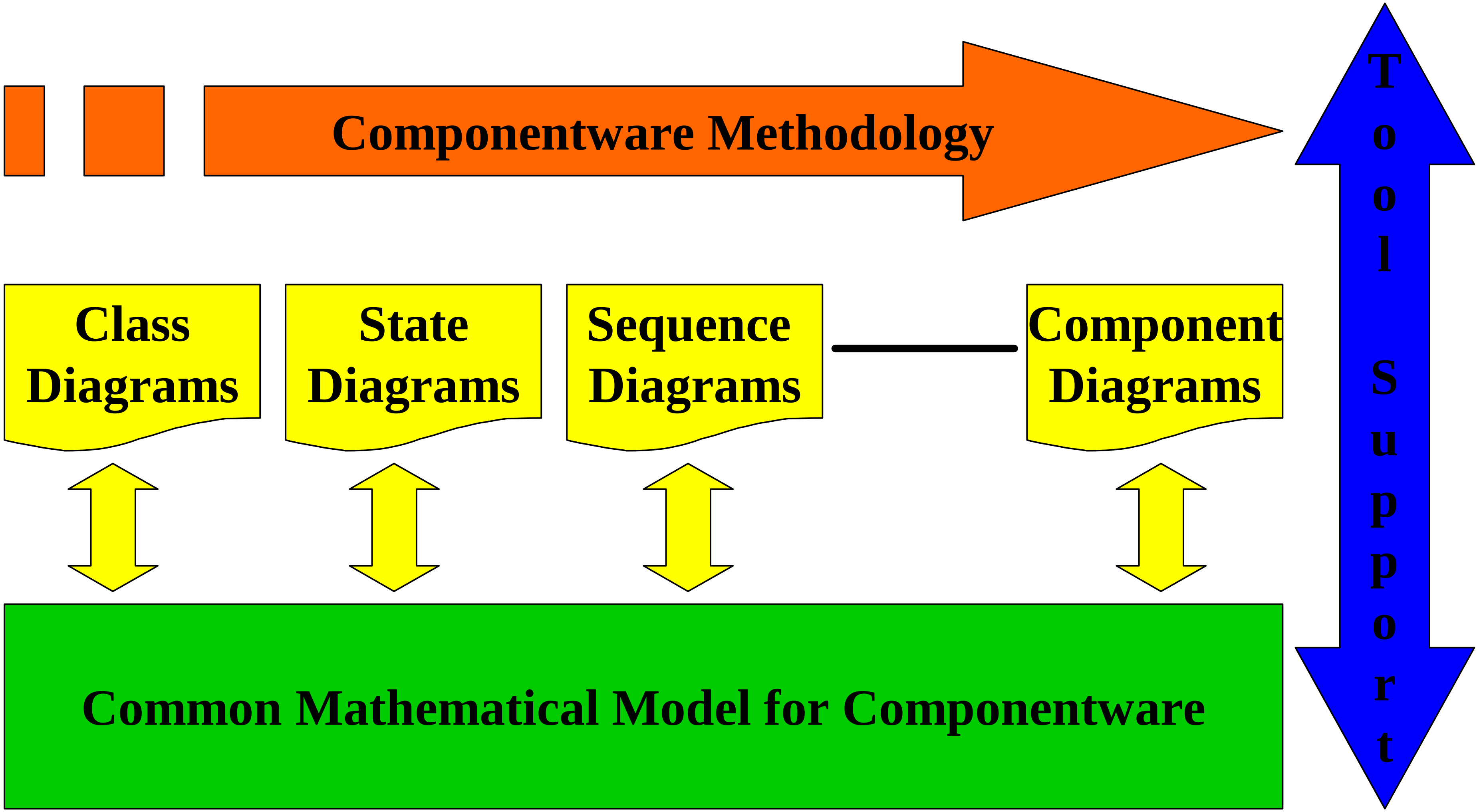
Lehrstuhl Prof. Dr. Manfred Broy

Institut für Informatik
Technische Universität München

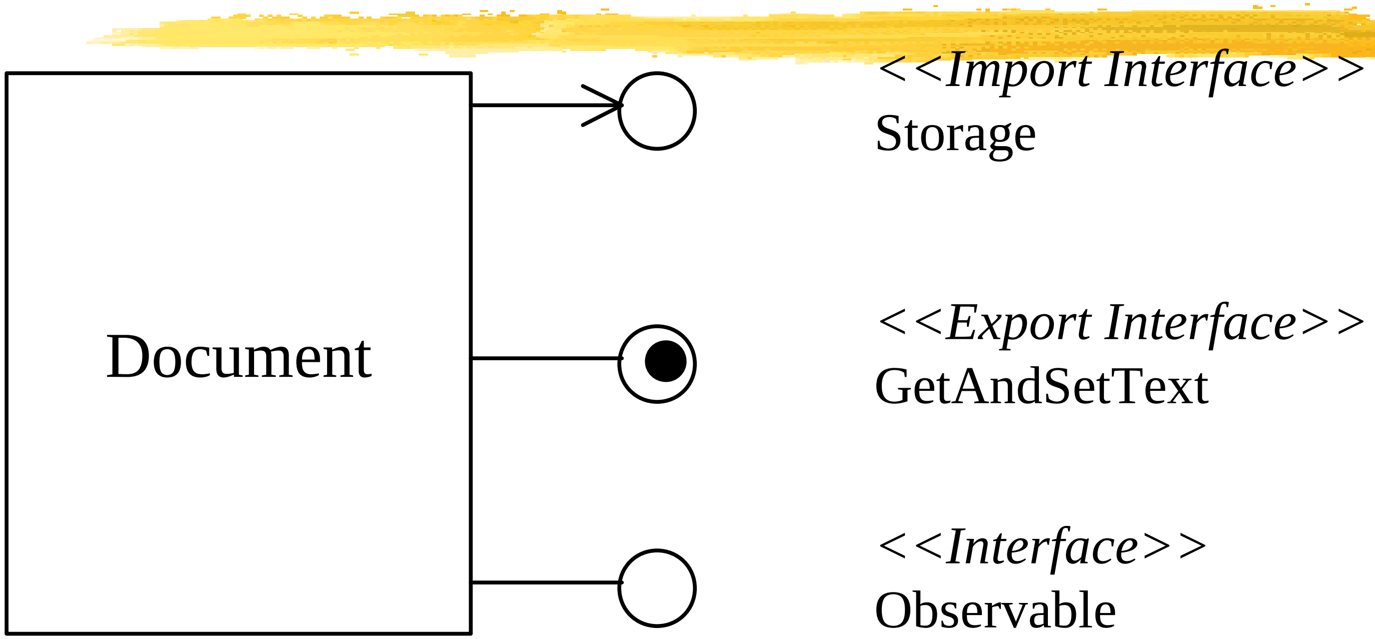
FORSOFT

A1: Methods for Component-Based Software Engineering

The Overall Vision

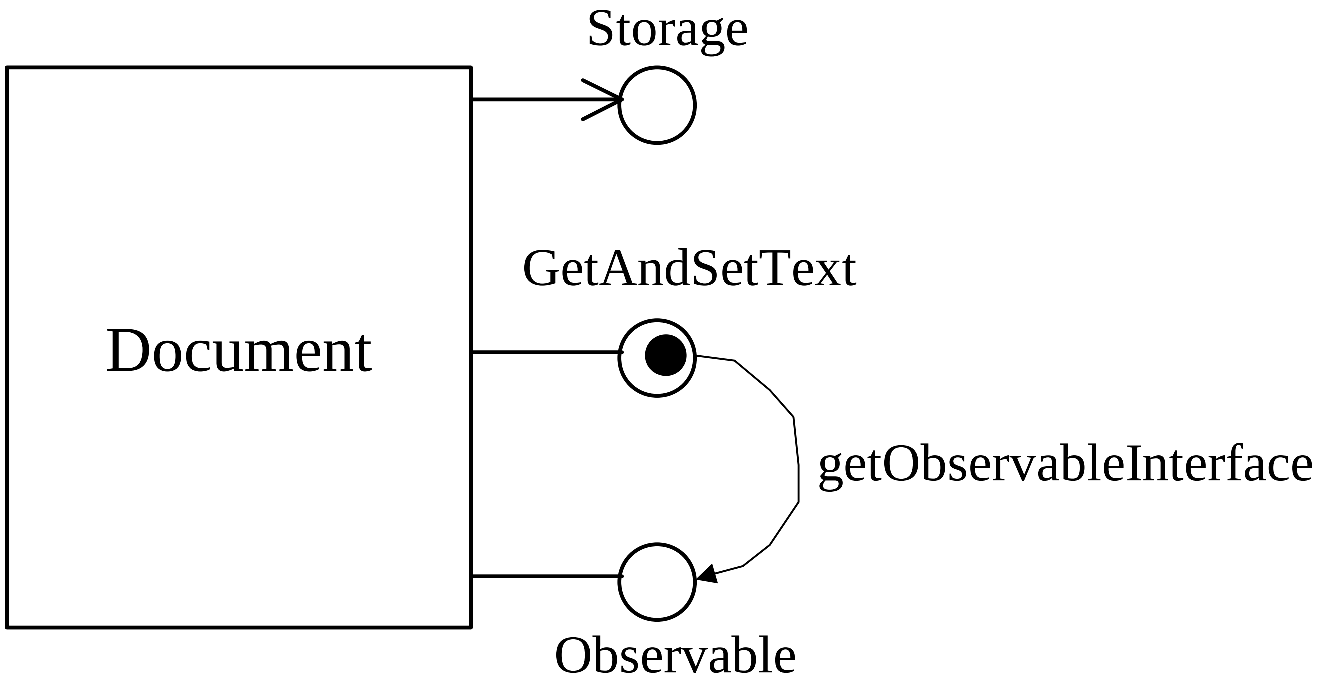


Components and Interfaces



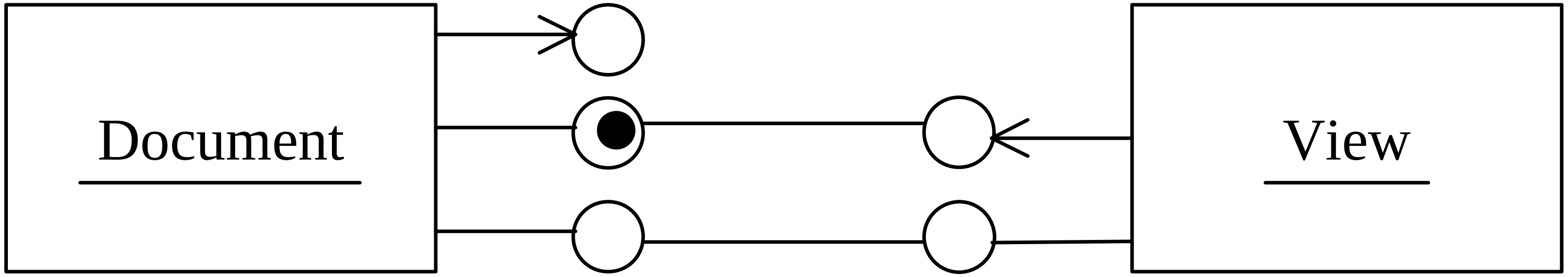
- component type is described by a set of interfaces
- component instance has an own data state and obeys the restrictions of the corresponding type
- interface is described by
  - a signature part
  - a behavior part

Component Interface Diagrams



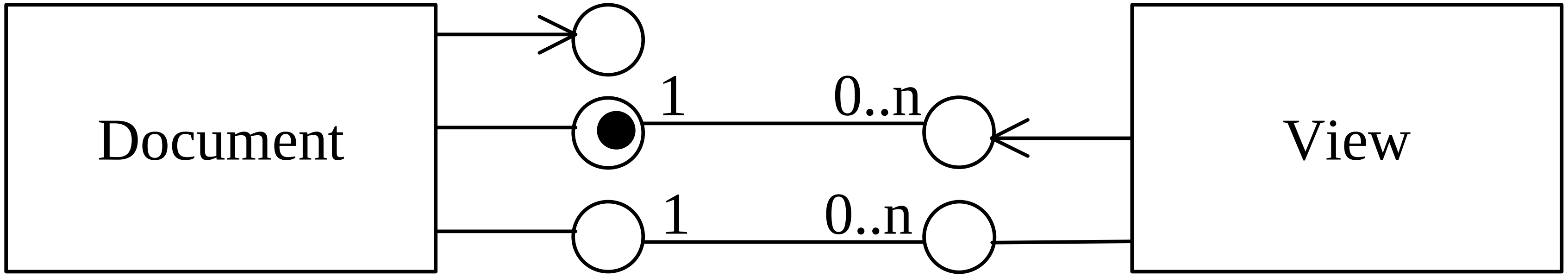
- specifing navigation between a component interfaces
- mapping into several technical component approaches
- static and dynamic consistency checks

Connecting Components



- interfaces of connected components must be compatible
- verification of compatibility enabled by the underlying formal model
- ‚state of the art' component-based tools
  - only syntatic checks
  - instance based
  - static connection structures

Component Connection Diagrams

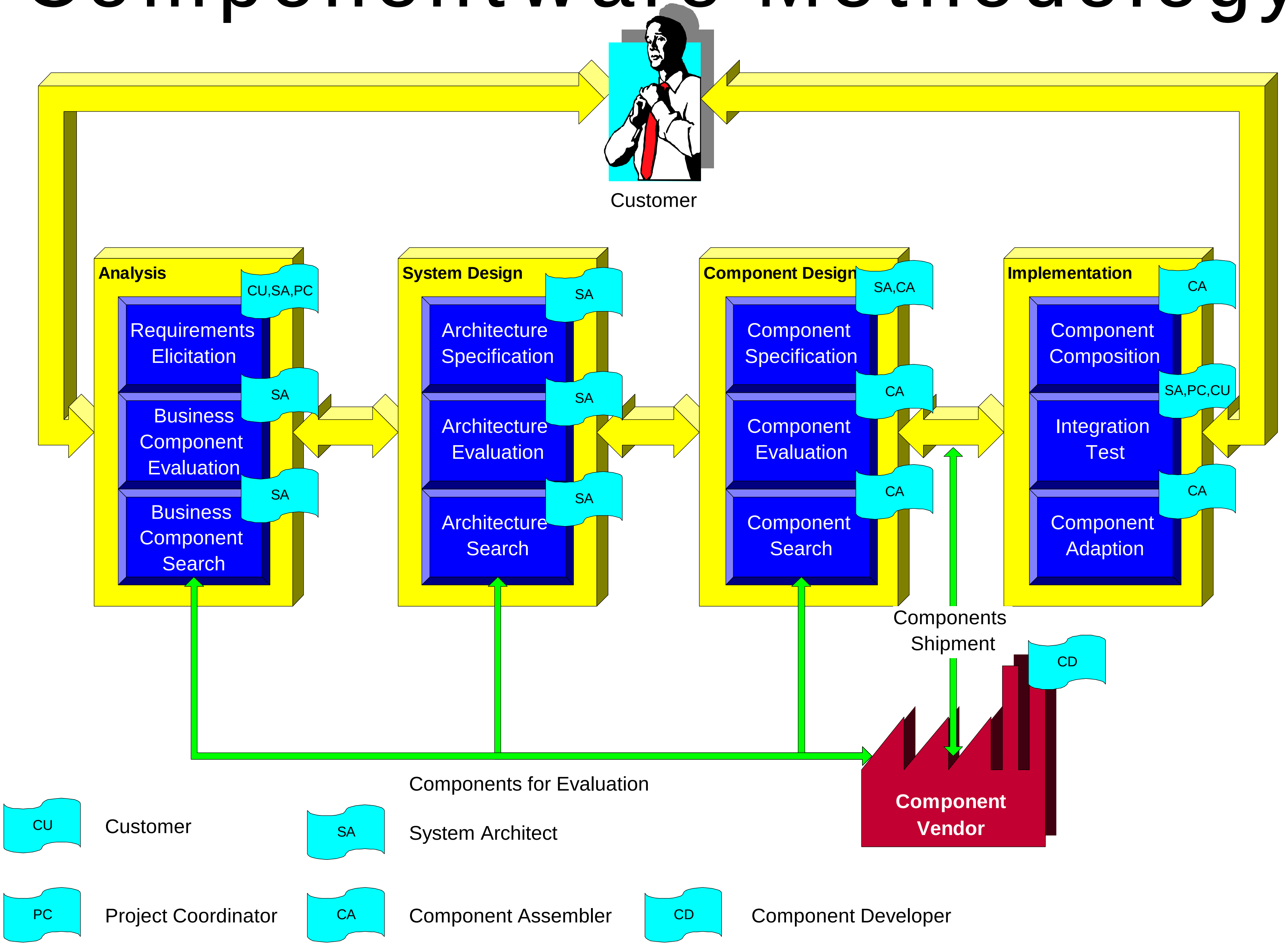


- dynamic connection structures
- creation and destruction of components
- visualization of maximum multiplicity of corresponding component instances
- component instances restricted by component connection specification
- static and dynamic consistency checks

A Componentware Methodology - Requirements

- adaptation of proven methodologies
  - improving methodologies
  - combining methodologies
- combining bottum-up and top-down approaches
- support of reuse and customization
  - new reuse-tasks and process-roles
  - searching for scalable architecture

A Componentware Methodology



Conclusion

- Summary
  - precise notion of component and interface
  - clear understanding of component connection
  - towards embedding the provided concepts into a formal method
  - framework for component-based development process
- Further Work
  - decomposition and refinement of components and interfaces
  - completion of embedding of provided concepts into a formal method
  - refinement of presented component-based development process
  - tool support of description techniques and development process
  - tool supported consitency checks based on the underlying formalism