A Connector Model for Object-Oriented Component Integration

Stefan Tai

stai@cs.tu-berlin.de

Technische Universität Berlin

Computergestützte
Informationssysteme

Integrated Object Systems (IOS)

Component integration

- Software development as long-term endeavor, combined forward and reverse engineering
- Composition of diverse, previously isolated applications in distributed, heterogeneous environments

Integration technologies

- Principle of software bus
  - Plug-in mechanism for component integration
  - Services for component interoperation
- exemplified by *Object Request Brokers*

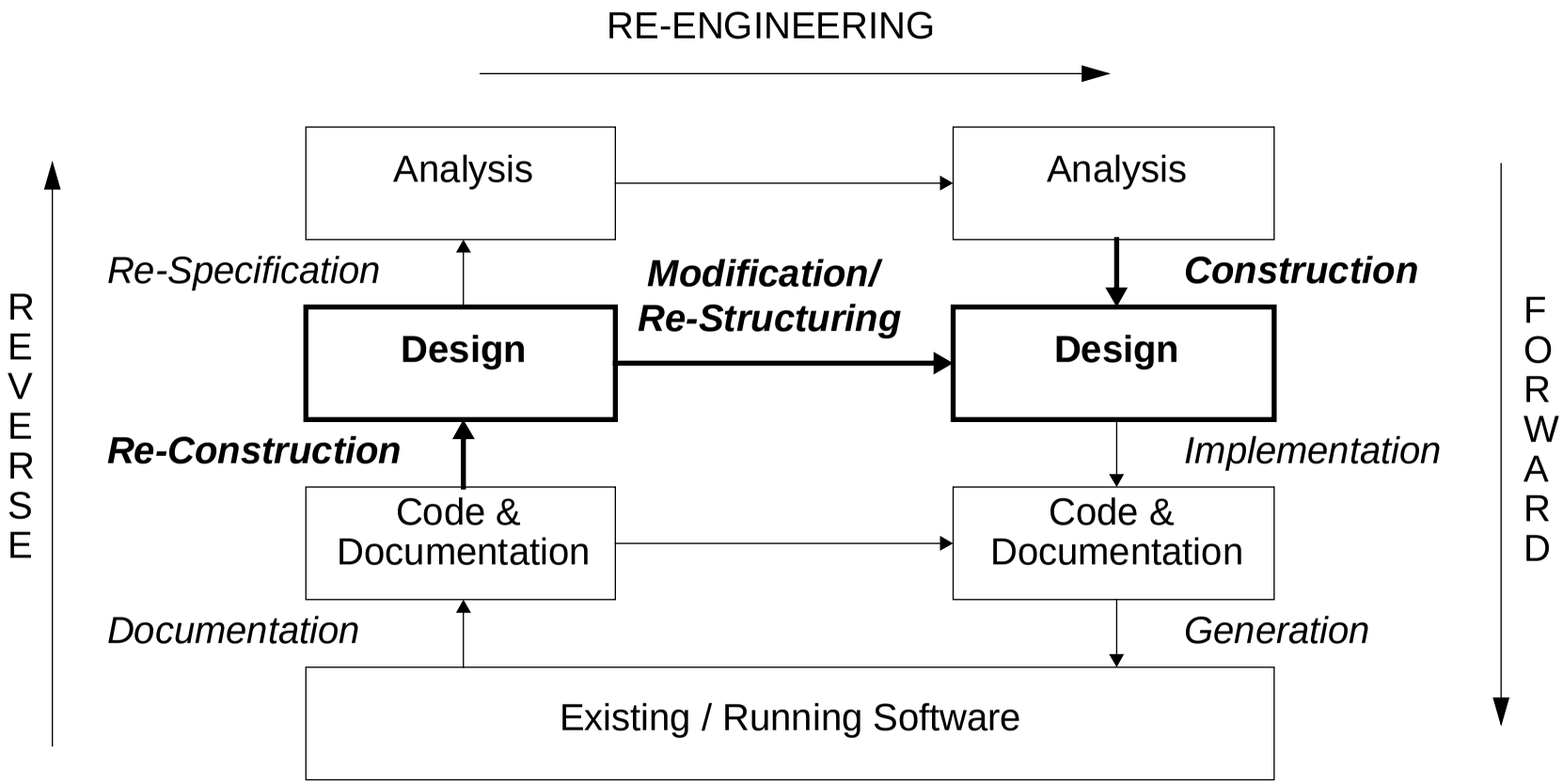
Integrated Object Systems

Software systems that are continuously built using object-oriented integration technology

Technische Universität Berlin

Computergestützte
Informationssysteme

Software Design in Continuous Software Engineering



Technische Universität Berlin

Computergestützte
Informationssysteme

Software Architecture

General Definition

"The description of elements from which systems are built, interconnections among those elements, patterns that guide their composition, and constraints on these patterns" [Shaw & Garlan 96]

Objectives

- Specification
- (Re-)Documentation
- Identification & Provision of transferable, reusable software solutions

Technische Universität Berlin

Computergestützte
Informationssysteme

Creating Software Architectures

Techniques toward an architecture

- Reference models
- Architectural styles/archetypes
- Design patterns

Languages to express an architecture

- Architecture Description Languages
  - Universal *component* and *connector* model
- Object-Oriented Modeling Languages
  - Class-based object model

Architecting IOS

What are suitable techniques and languages to meet the diverse objectives?

Technische Universität Berlin

Computergestützte Informationssysteme

Software Architecture for Integrated Object Systems

Observations

- Mutual influence between IOS architecture and enabling technology used to build IOS
- Conventional architectural description distant from IOS technical realization

- Impact of technology on architecture should be made explicit
- Improving documentation practices by dedicated abstraction concepts for modeling IOS

Technische Universität Berlin

Computergestützte
Informationssysteme

Describing Component Collaborations

Connectors of ADLs	• First-class language constructs symmetrical to components • Def. of *roles* as types, of *interaction protocols* • Universal model
Object collaborations	• Separate modeling languages/notations, rather weakly integrated with object type/class modeling languages • Def. of *sequences of method invocations* among a group of objects • Object-oriented model

Technische Universität Berlin

Computergestützte
Informationssysteme

Issues of Using Connectors/Object collaborations for Modeling IOS

Connectors

- Object-oriented interpretation of connectors
- Technical realization of connectors
- Connector abstractions of
  - technology (engineering viewpoint)
  - business problem domain (enterprise viewpoint)
  - existing software infrastructures

Object collaborations

- Programming level vs. architecture level of description
  - Object/component granularity
  - Intensional and extensional relationships between 'architectural objects'

Technische Universität Berlin

Computergestützte
Informationssysteme

Description of Components and Connectors – A Specialized Perspective –

Description of	Transferable Abstraction	Particular System
Component	Isolated component	Integrated, adapted /wrapped component
Connector	Pattern of component collaboration	Composition of specific integrated components

Def. *Component*:
System of programming objects (incl. technical infrastructure objects), encapsulated by ORB-level interfaces

Technische Universität Berlin

Computergestützte Informationssysteme

Connectors as Pattern-like Transferable Abstractions

	Structured template/recording schema
Roles	• Collaboration participants marking responsibilities
Role Interfaces	• Role behavior expressed in terms of provided and/or required services
Interactions	• Sequences of requests wrt. role interfaces
Modeling Example	• Use and impact of applying a connector to *specific* components
Abstract Architecture	• Component composition using connectors: Distribution of role responsibilities to specific components
Concrete Architecture	• Component composition using component abstractions only: Exhibiting individual component customizations and adaptations of collaboration rationale

Connector Examples

- an Event Notification connector, Relationship connectors

  S. Tai, S. Busse. *Connectors for Modeling Object Relations in CORBA-based Systems.* Proc. TOOLS-24, 1997.

- an OTS connector

  S. Busse, S. Tai. *Software Architectural Modeling of the CORBA Object Transaction Service.* Proc. COMPSAC-22, 1998. *to appear.*

Technische Universität Berlin

Computergestützte
Informationssysteme

Summarizing Position

Integrated Object Systems

- A class of component-based systems characterized by distribution, heterogeneity, continuity of development

Software Architecture for IOS

- Describing transferable abstractions vs describing particular systems, Making impacts of technology on architecture explicit

Specialized connector model

- Documentation template combining concepts of software architectural connectors and object-oriented modeling
  - to exhibit (enterprise or engineering) patterns and assumed contexts guiding component composition
  - can be related to a more technical, implementation-oriented system representation
  - existing, standard modeling languages can be used