
Model-Driven,
Component-Based Development

Contents

- Motivation
- Component Modeling
- Product Line Engineering
- Process Modeling
- Conclusion

Industrial Reuse Drivers

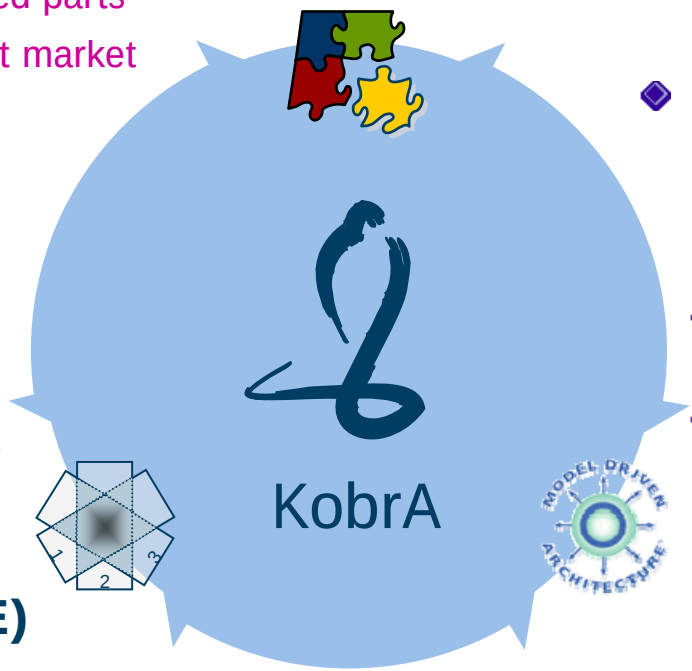
Component-based Development (CBD)

Vision

- Assemble applications from prefabricated parts
- COTS component market
- Web Services

Obstacles

- Current technologies (.NET, J2EE) very implementation oriented
- Little understanding of how to scope components



Vision

- Development activities oriented around product families
- manage commonalities and variabilities

Product-Line Engineering (PLE)

Obstacles

- Larges upfront investment
- Poor connection with regular "single-system" technology

Obstacles

- Lack of systematic methods for creating PIMs
- Fixed and Ad hoc mapping techniques

Model-Driven Architecture (MDA)

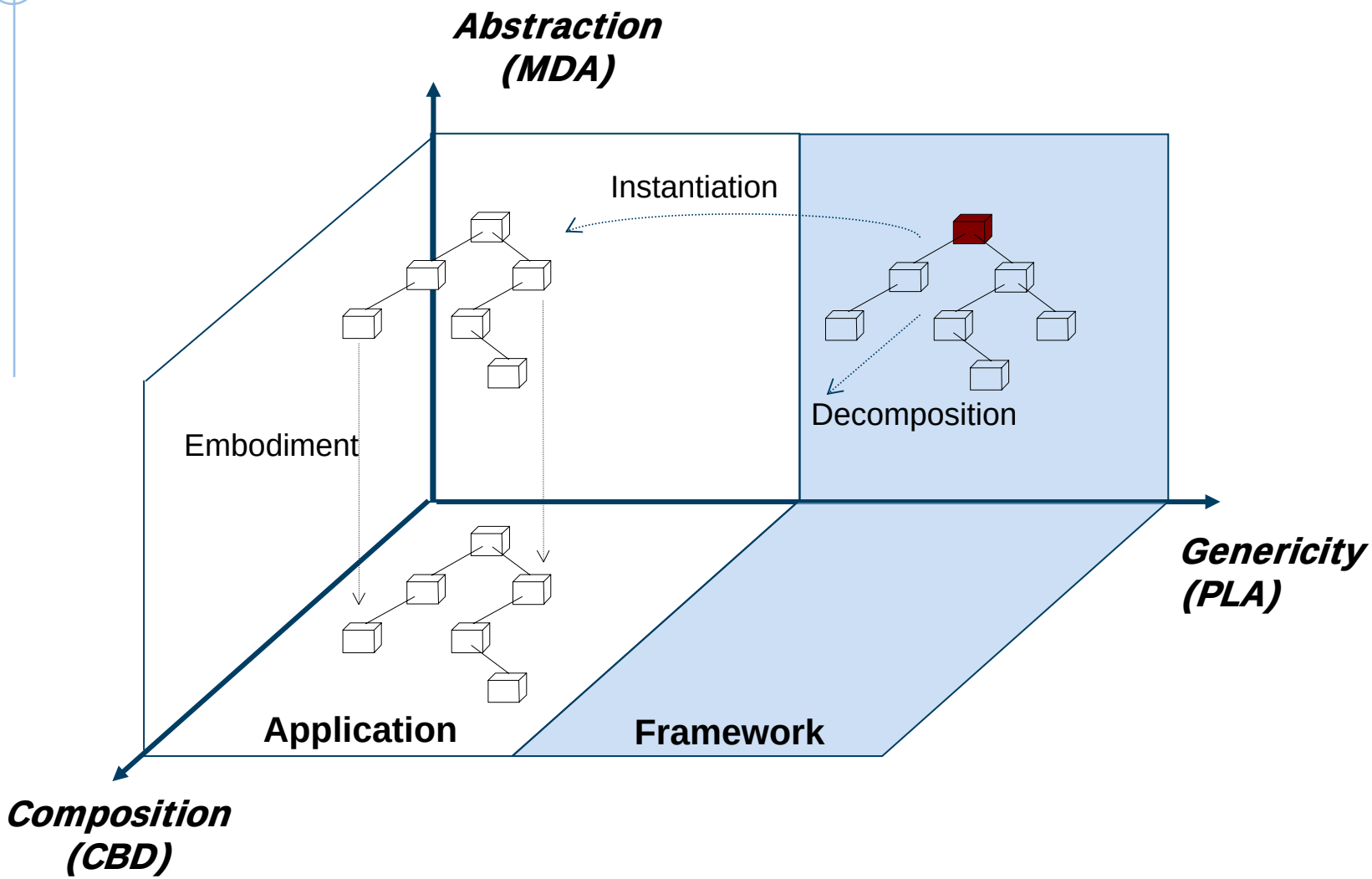
Vision

- Capture core software assets as platform-independent models (PIMs)
- Automatically map PIMS to PSMs

The KobrA Project

- Komponentenbasierte Anwendungsentwicklung
  - Supported by BMBF
  - January 1999 -> December 2001
- Four partners
  - Softlab GMBH, Munich
  - PSIPENTA Software Systems, Berlin
  - Fraunhofer FIRST, Berlin
  - Fraunhofer IESE, Kaiserslautern
- Successfully applied by numerous companies
  - PSIPENTA, Digital Steps, SIEDA, ...
- At IESE, further developed for embedded systems development
  - MARMOT Method

Separation of Concerns

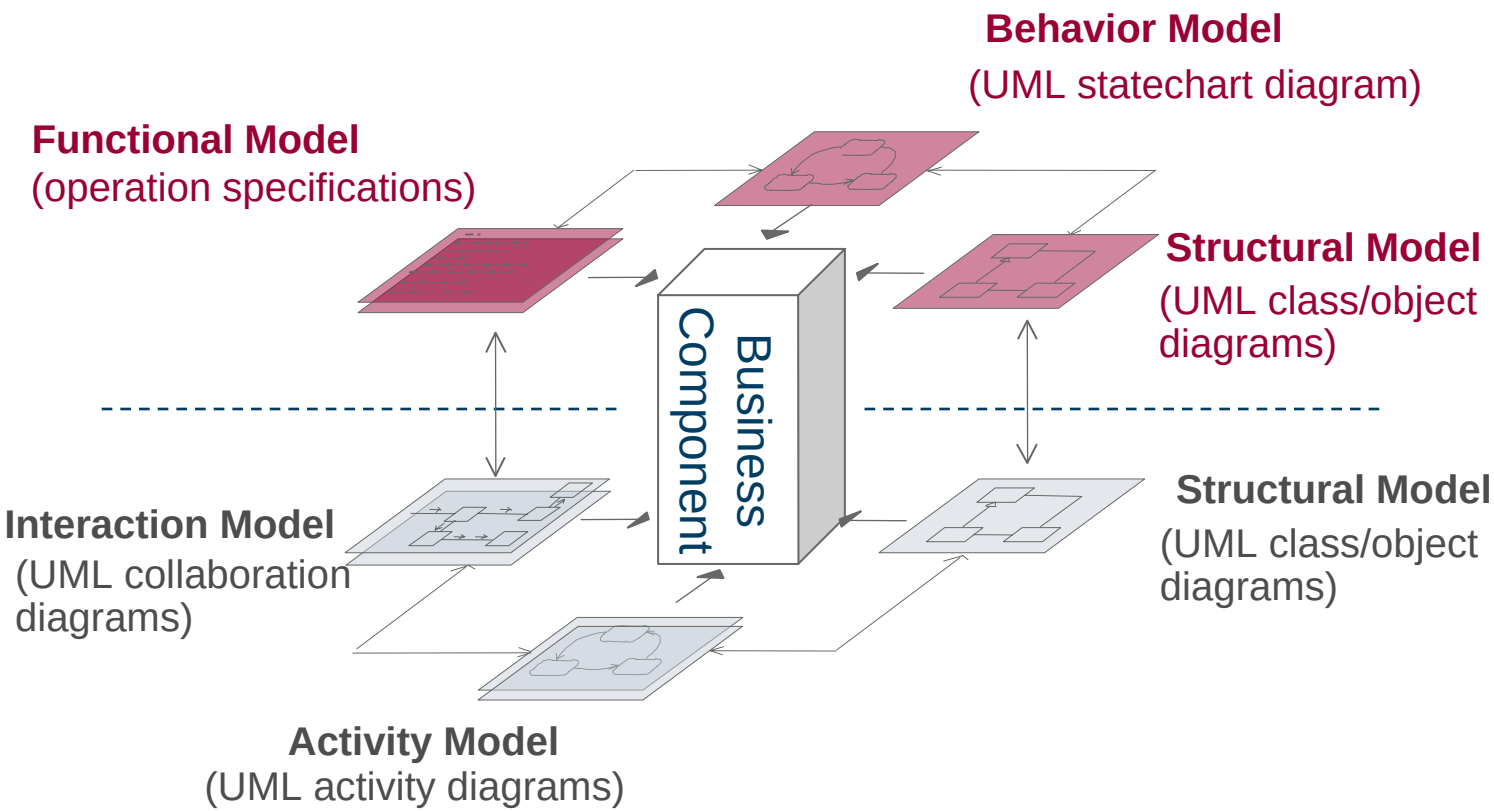


Modeling Principles

- Uniformity
  - **all** behavior rich elements should be viewed as components, including (sub)systems
  - component assembly = component development
- Parsimony
  - minimal set of concepts (no redundancy)
- Locality
  - all models should be local to a component
- Encapsulation
  - component specifications (what) must be separated from component realizations (how)

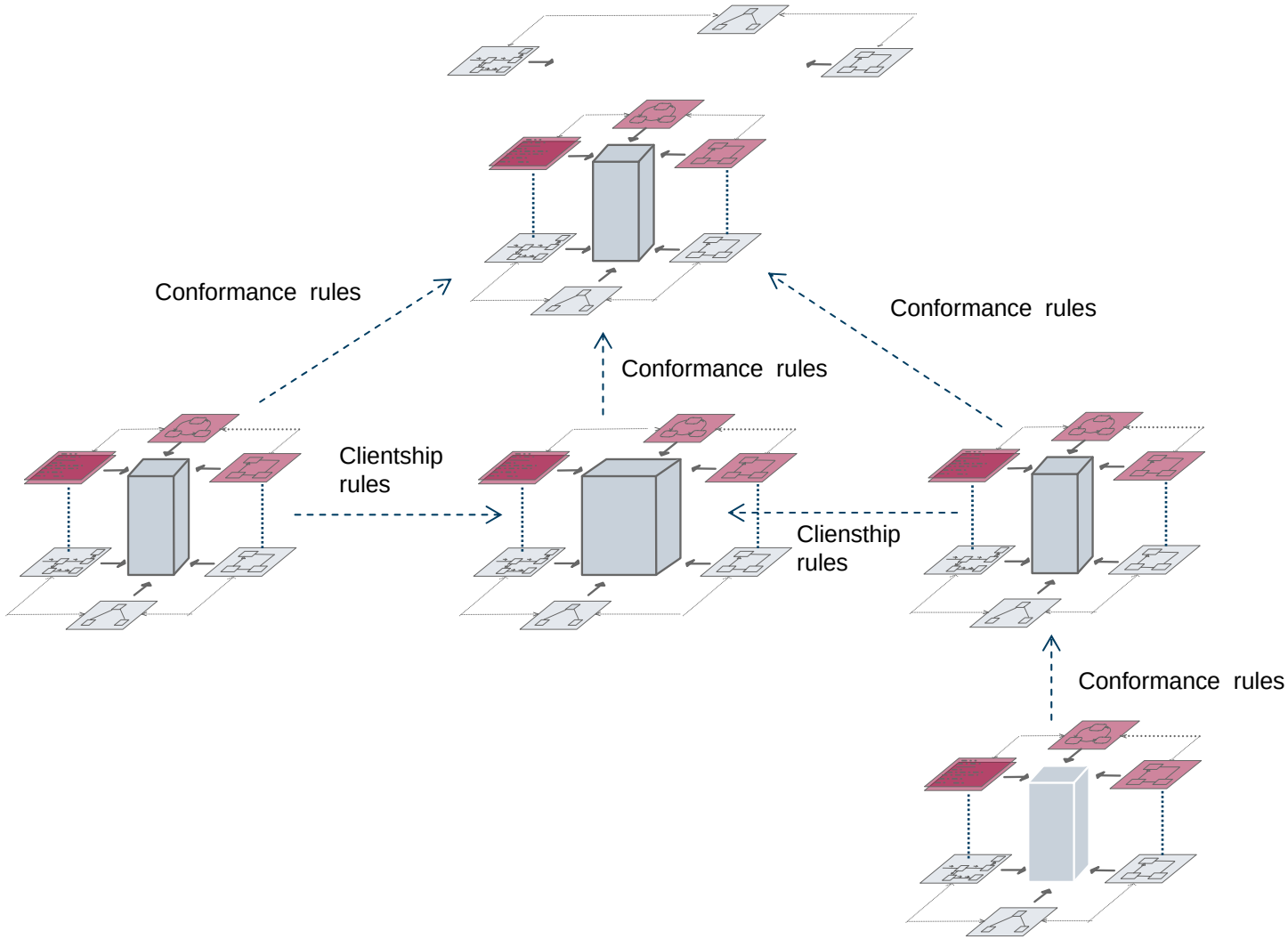
Component Modeling

Specification

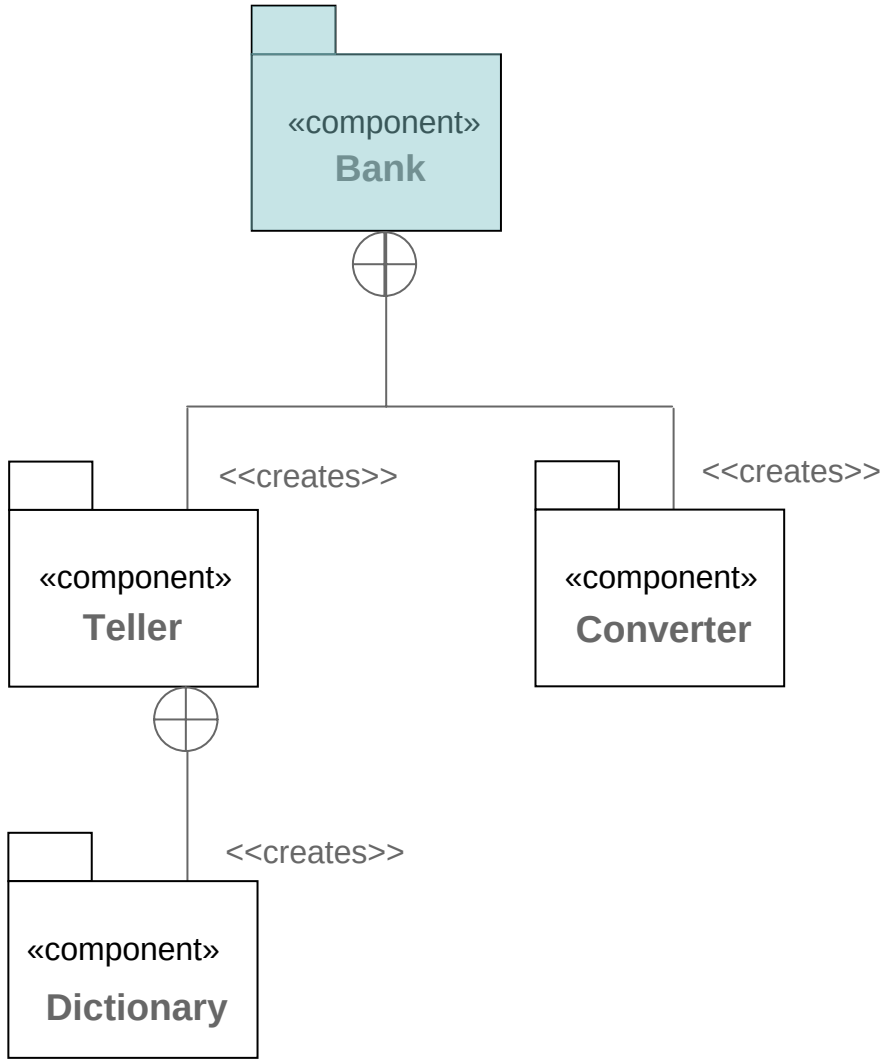


Realization

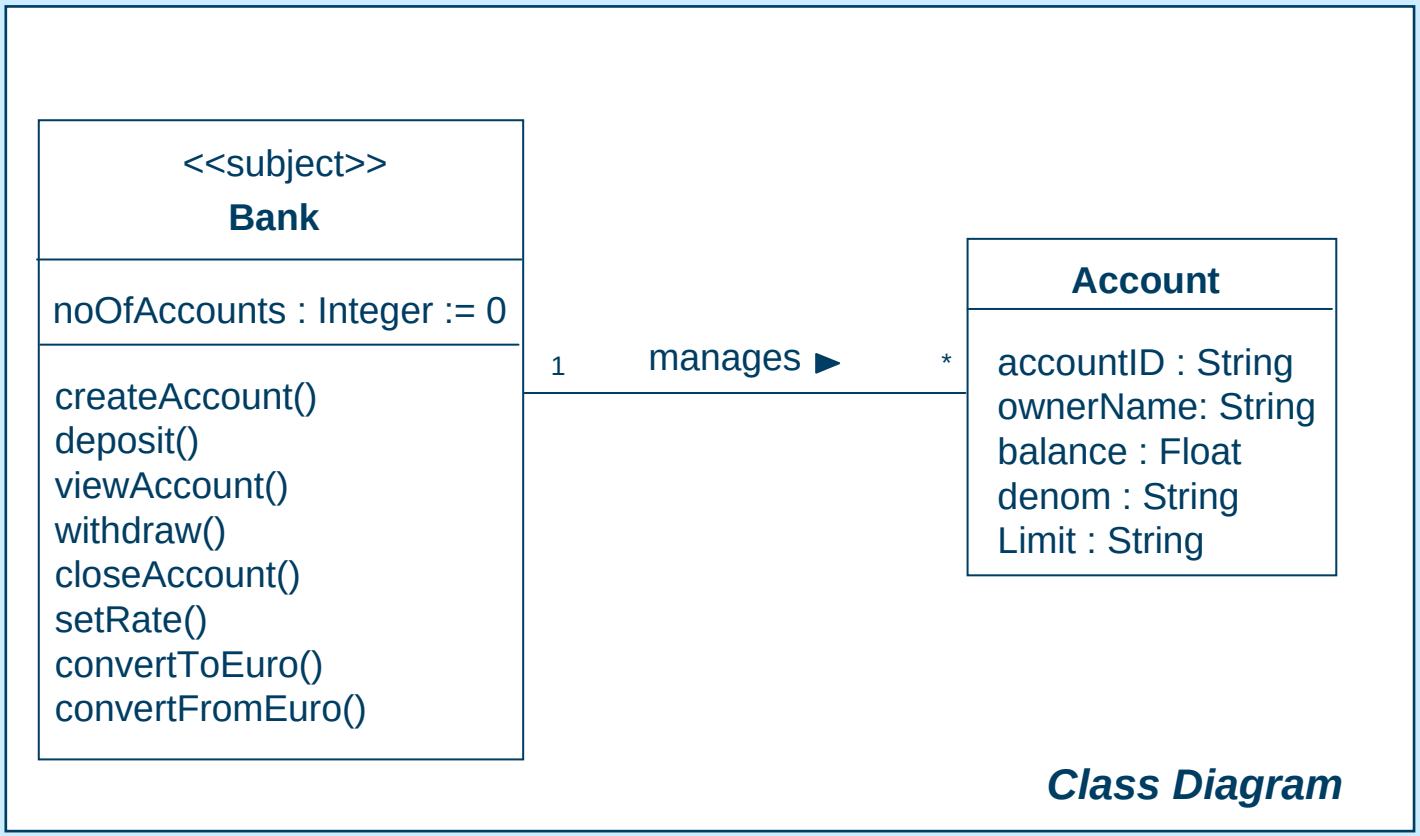
Composition Hierarchy



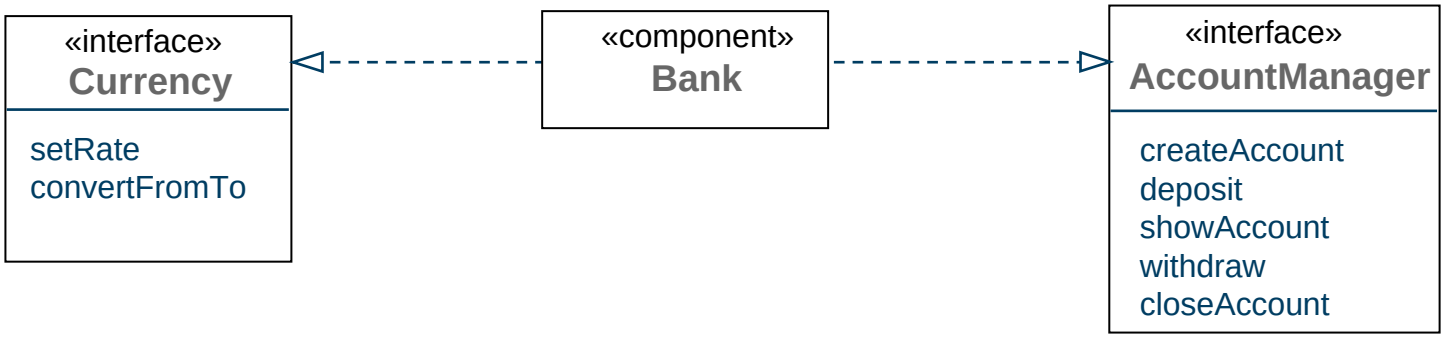
Simple International Bank (SIB) Example

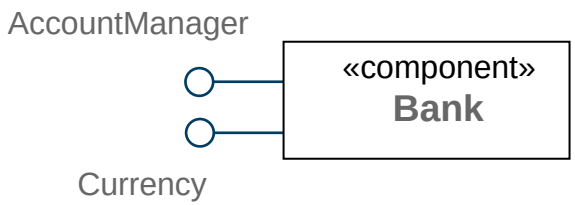


Component Specification (Bank)

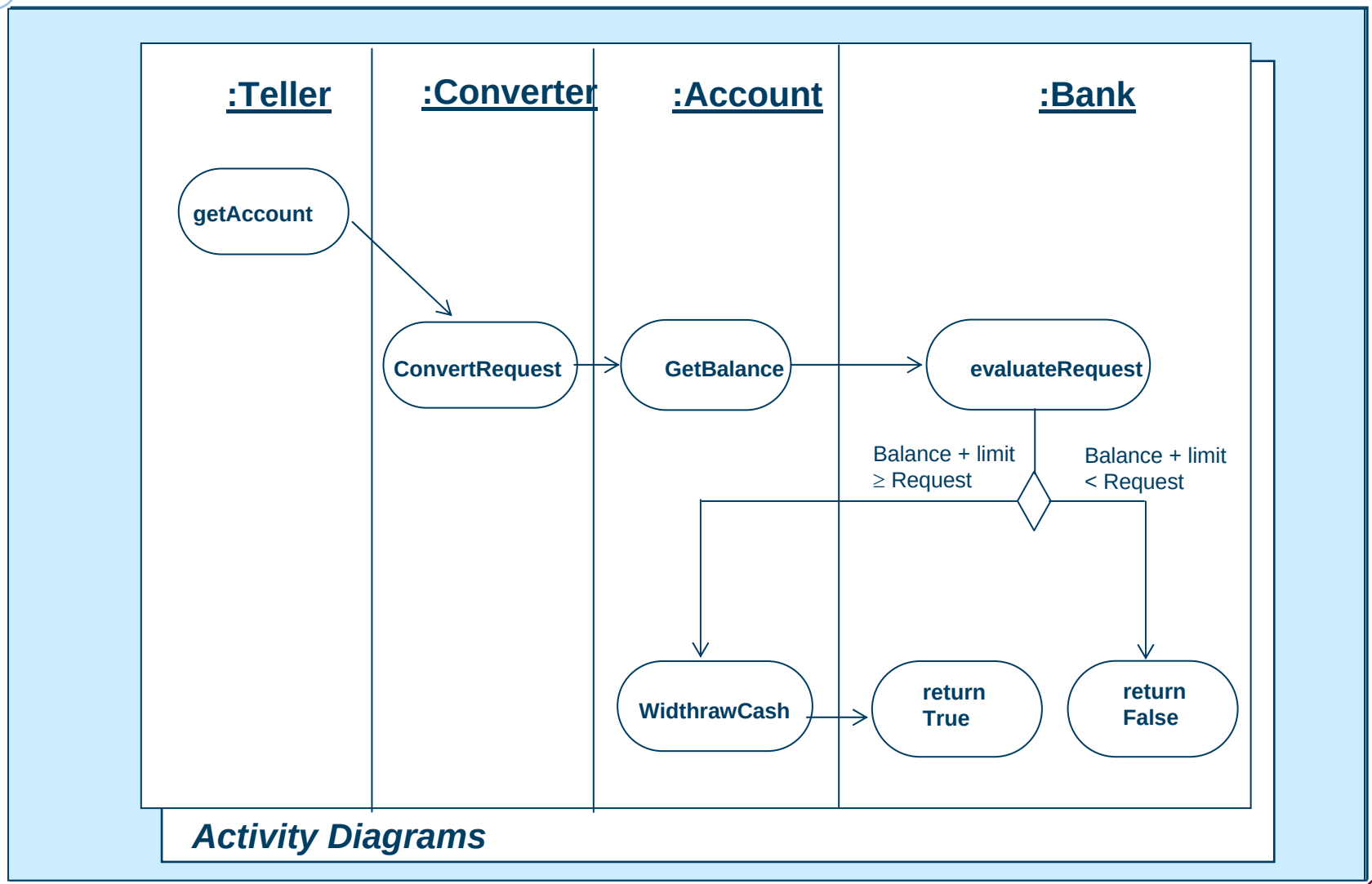


Simple International Bank (SIB) Example

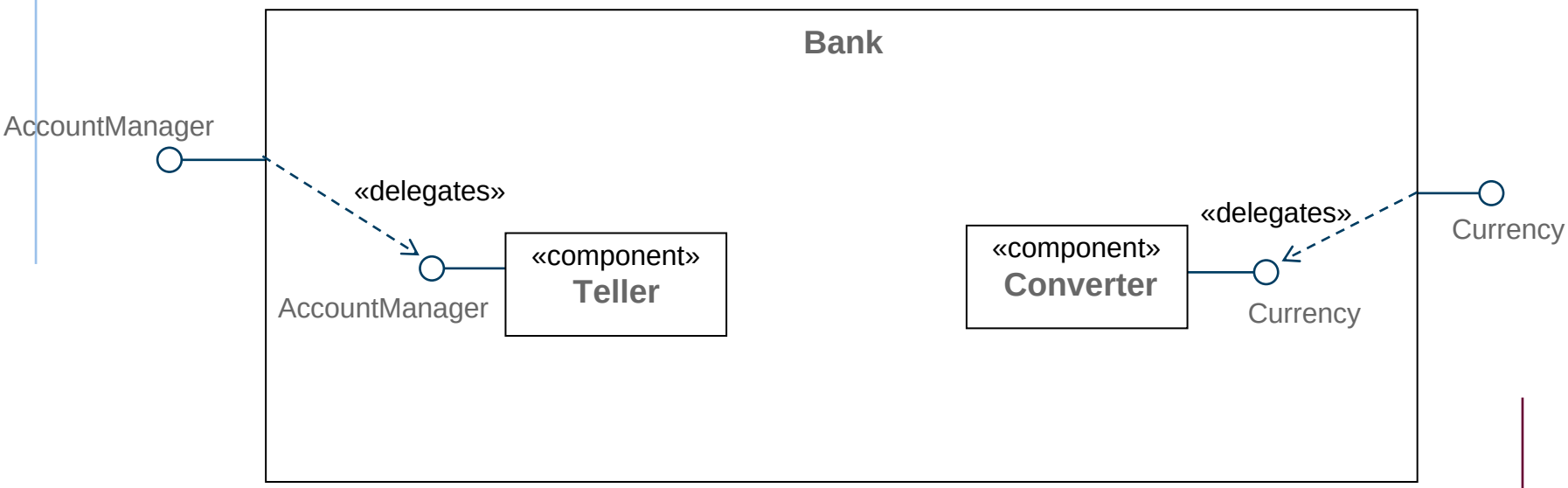




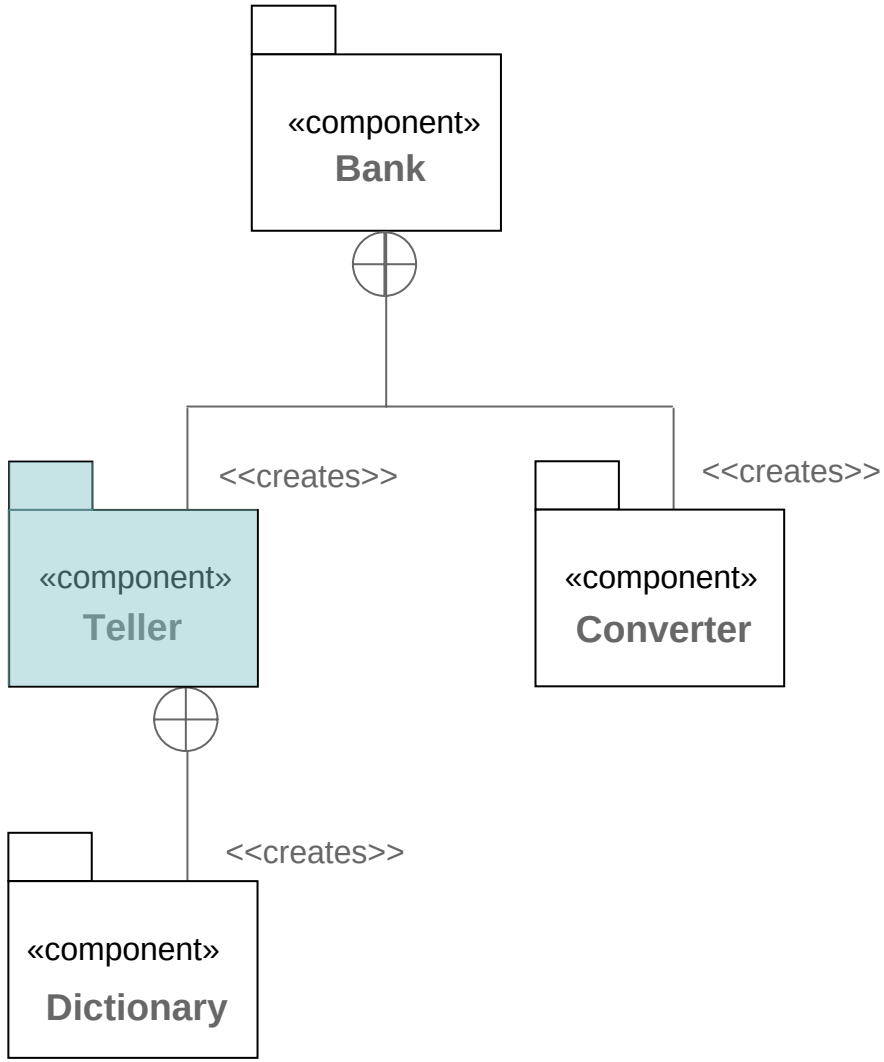
Component Realization (Bank)



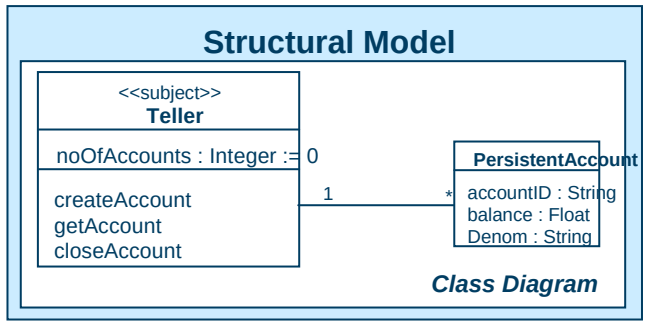
SIB Realization



Simple International Bank (SIB) Example



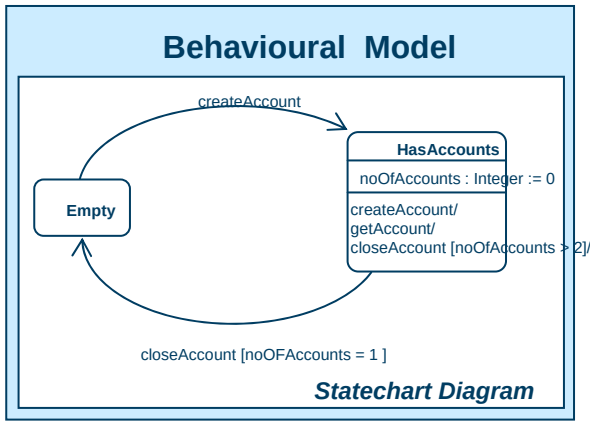
Component Specification (Teller)



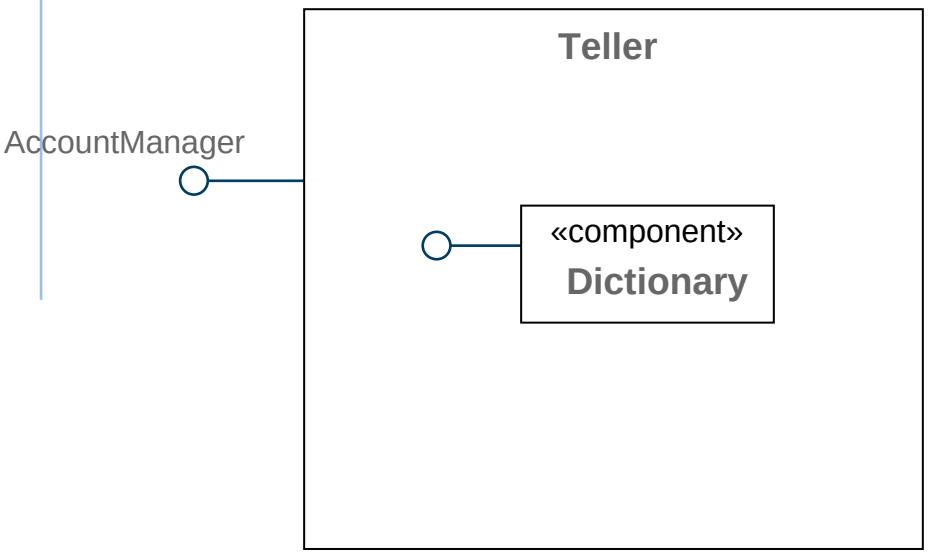
Functional Model

Name	createAccount
Informal Description	An account is opened in a particular currency for a customer with a particular name, and the Account ID is returned
Constraints	--
Receives	name : String currency:String
Returns	A String with the ID of the account
Changes	teller
Assumes	There is an exchange rate for the specified currency
Result	A new account with a unique ID in the denomination, currency, has been generated The name of the customer has been stored in account The account ID has been returned

Operation Specifications

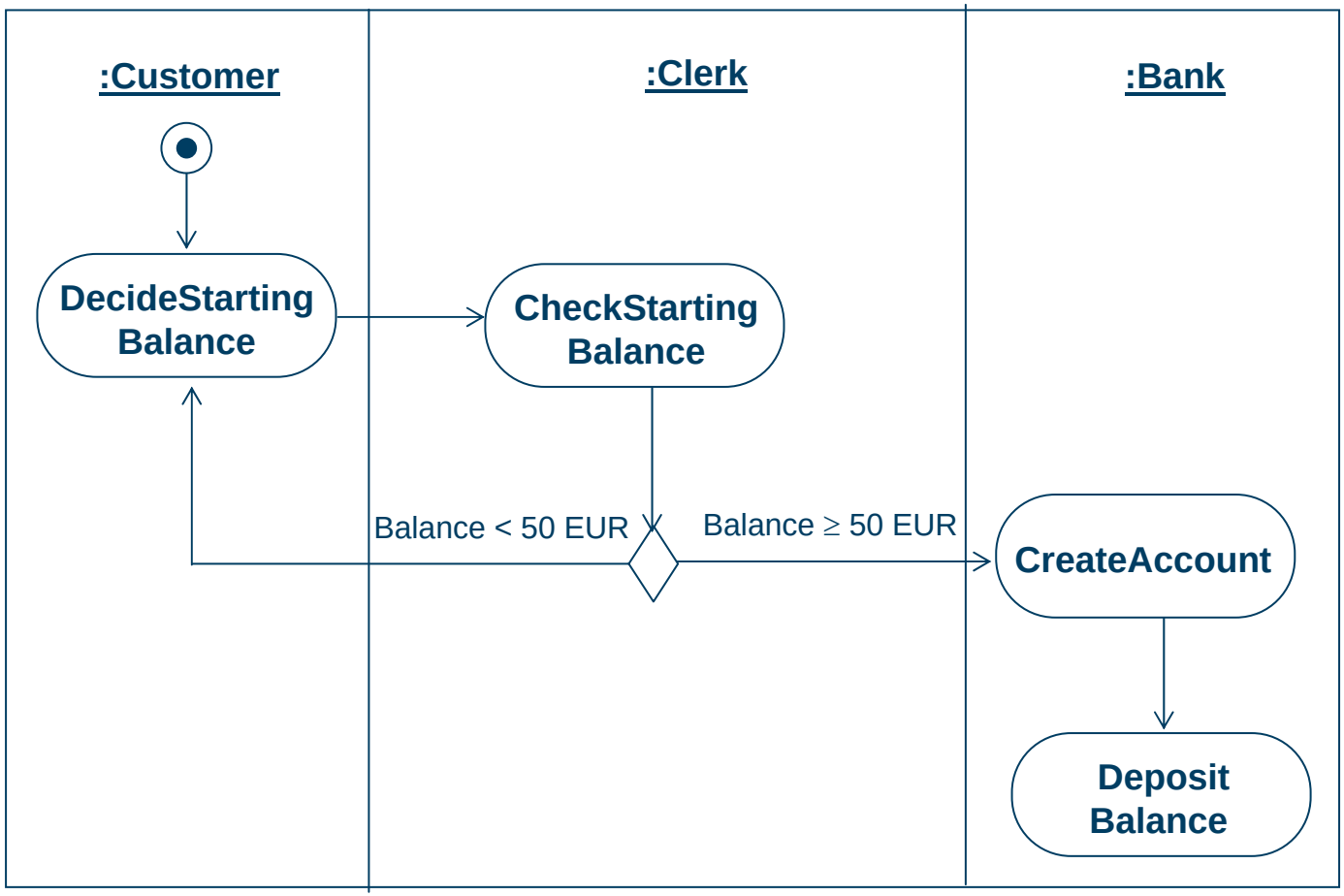


Teller Realization



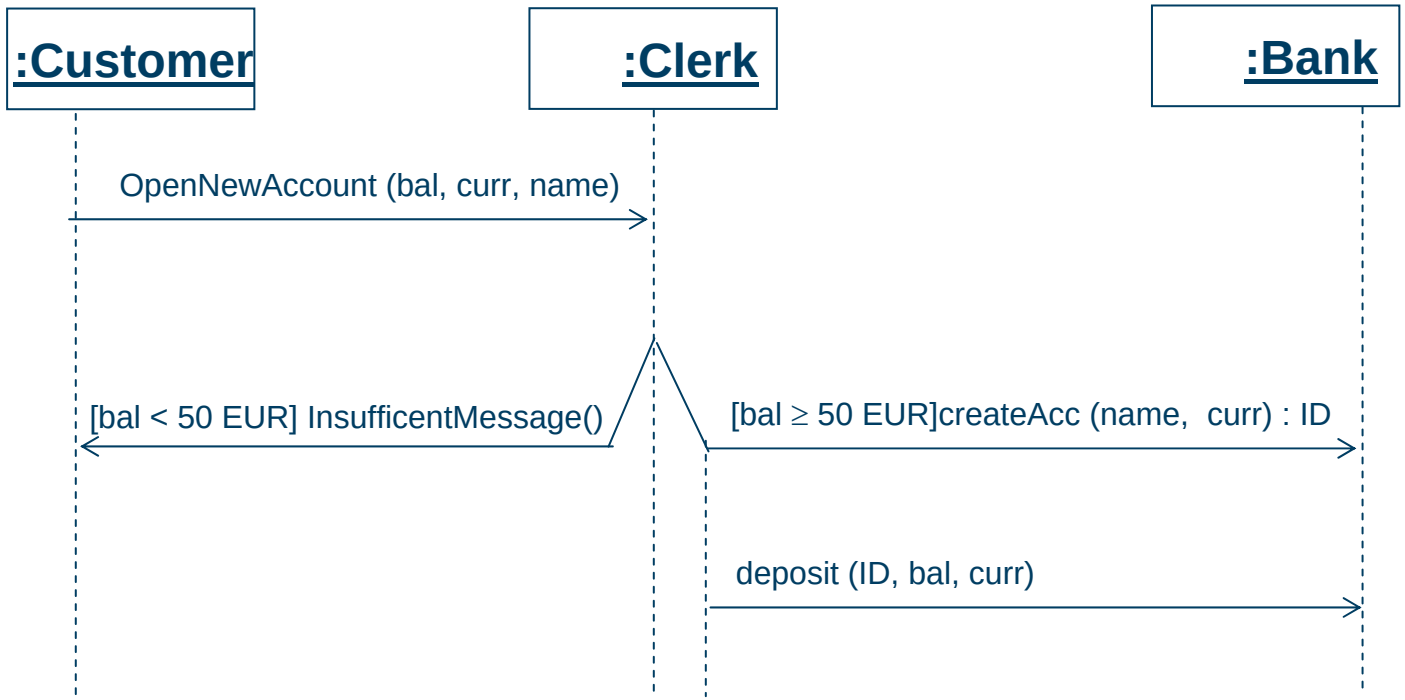
SIB Context Activity Model

- openNewAccount activity diagram



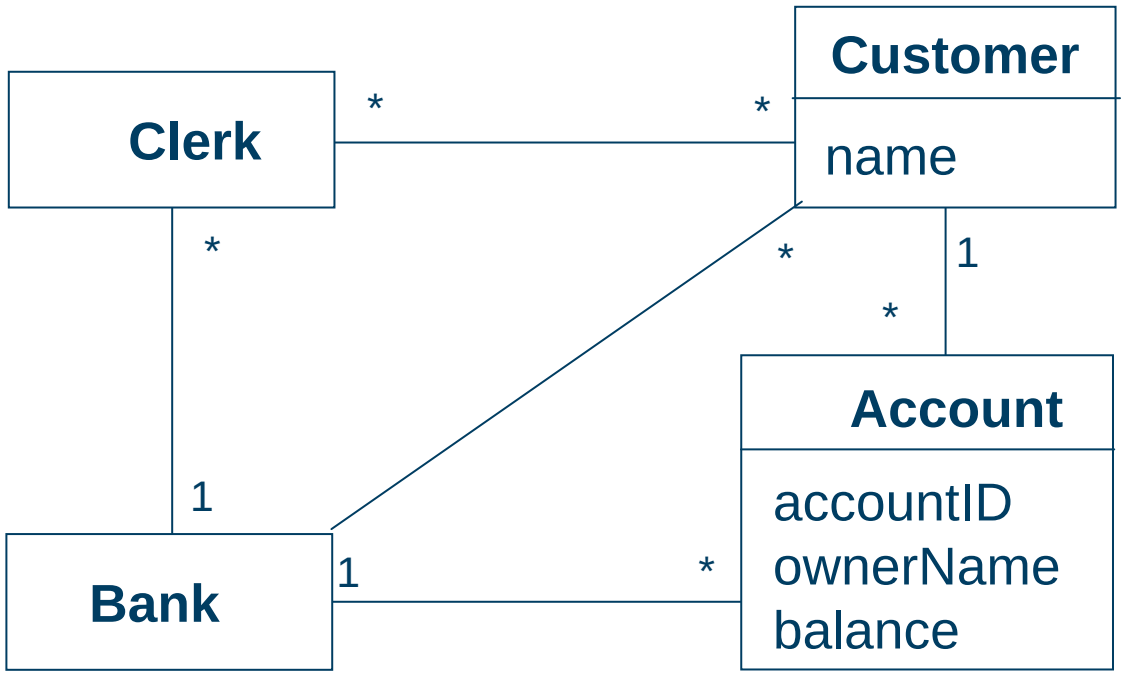
SIB Context Interaction Model

- openNewAccount sequence diagram

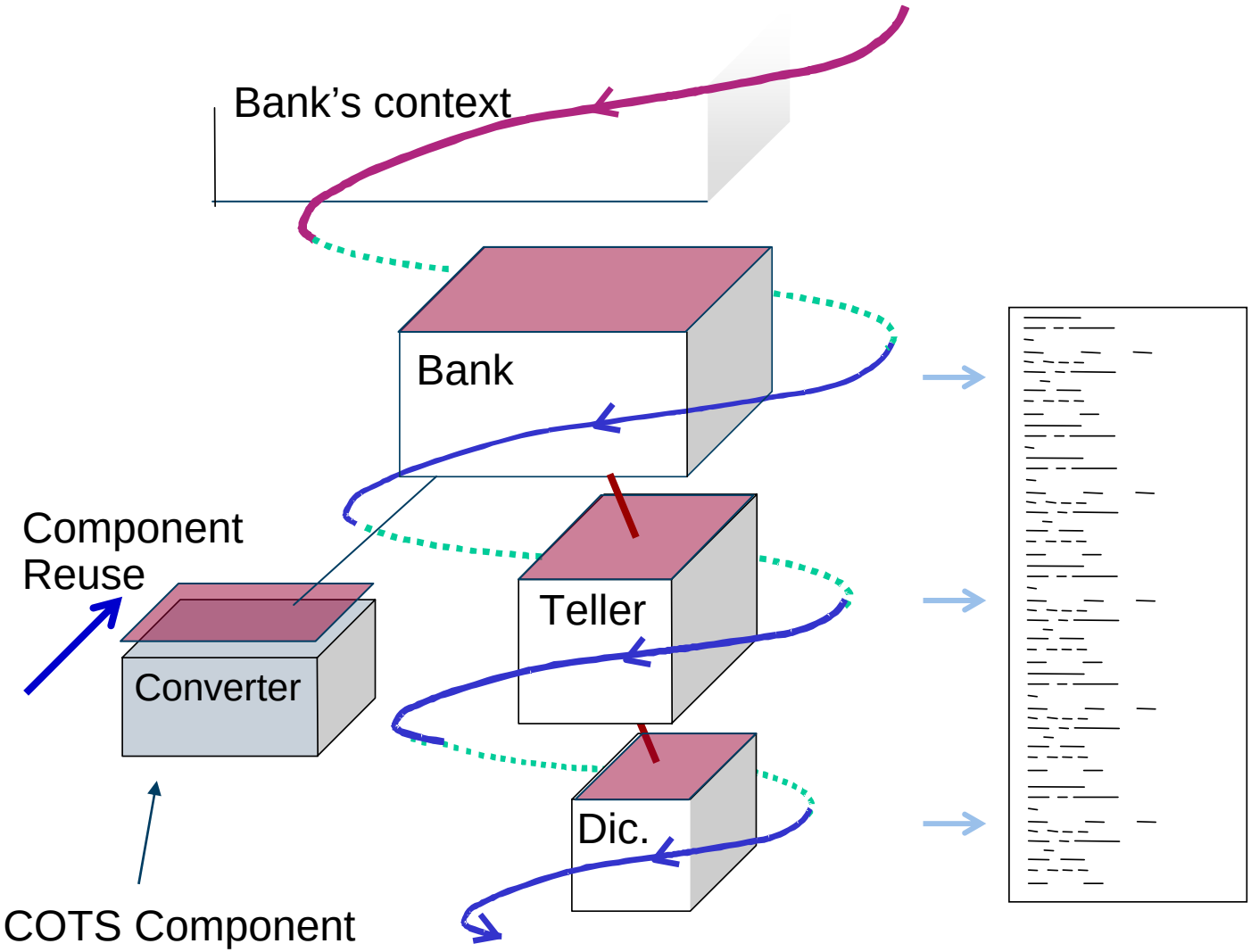


SIB Context Structural Model

- Bank Context (openNewAccount) Class Diagram



Component Engineering Process



Product Line Engineering

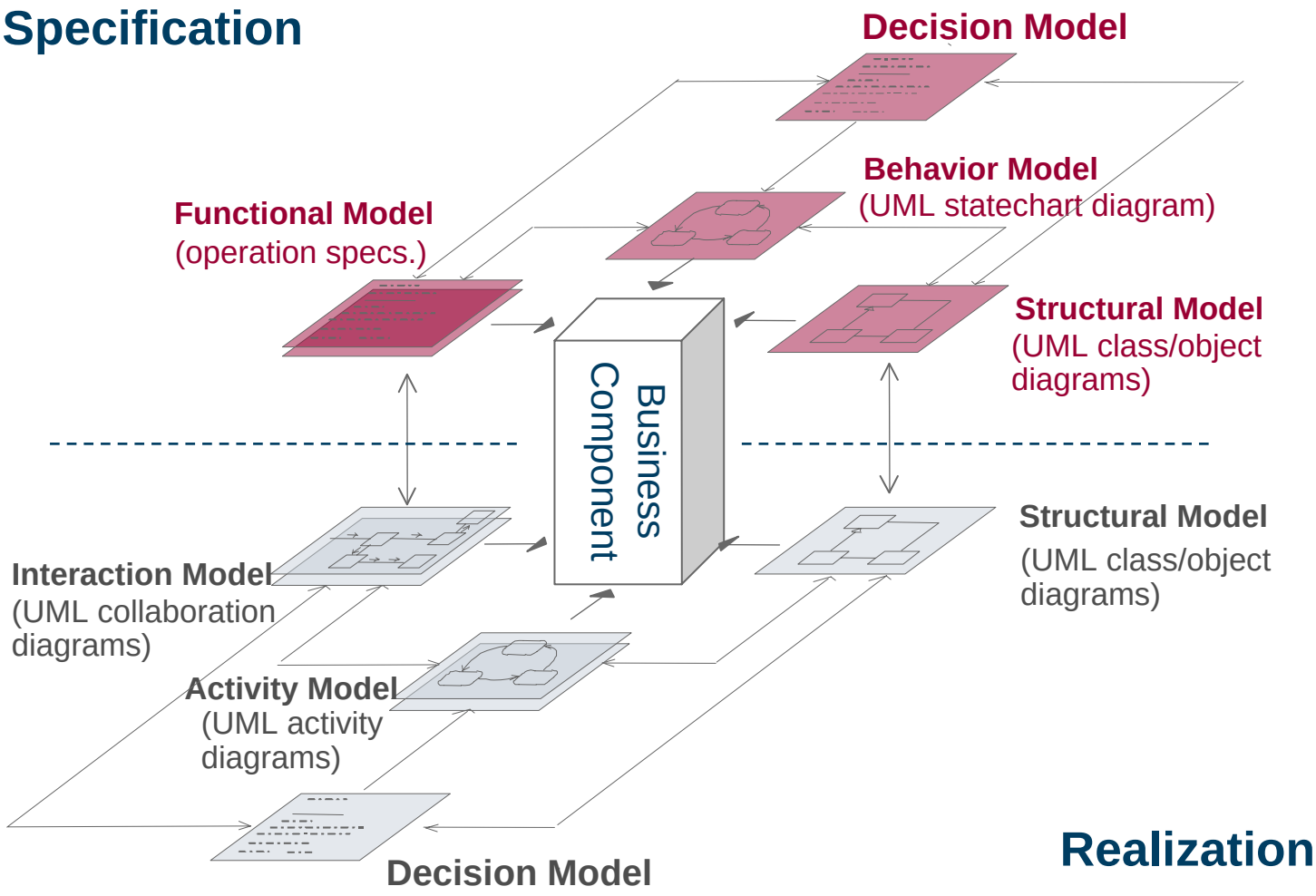
- Aims to systematically exploit the similarities between systems

- Involves the development of a generic infrastructure reusable across a family of target products

- Key activities include
  - analysis of common and variable product characteristics
  - definition of the intended scope of reuse
  - identification of the optimal level of genericity to support variant and optional features

- Traditionally implies major upfront investment

Generic Component Models

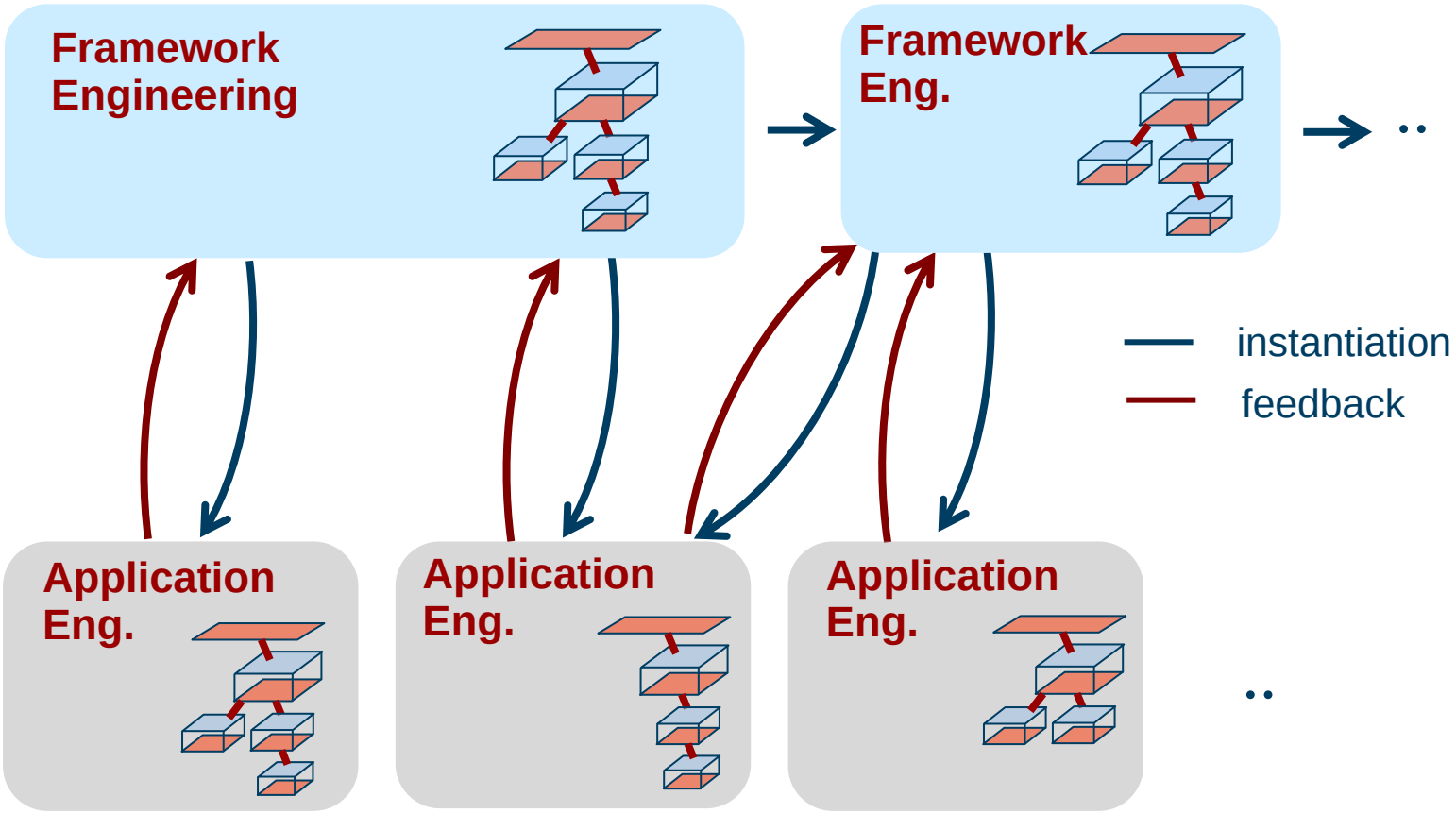


Generic Component Specification (Bank)

	Question		Diagram	Effect
1	Is a customer allowed to overdraw his/her account up to a certain limit?	Y		
		N	Class Diagram	Remove attribute Account.limit
			Operation Schema withdraw	Remove limit from <Result>
2	Is it an international bank that handles different currencies?	Y		
		N	Class Diagram	Remove operation Bank.setRate()
				Remove operation Bank.convertToEuro()
				Remove operation Bank.convertFromEuro()
			Operation specification withdraw	Remove currency from <Description>
				Remove currency from <Receives>
				Remove currency from <Result>

Decision Tables

Product Line Life Cycle



Pros and Cons of KobrA

- Simple and Systematic
  - strict separation of concerns
- Incremental introduction of component and product lines
- Uniform treatment of systems and components
  - component assembly = component creation
  - fractal-like product, recursive process
- Integrated quality assurance
  - Inspections, testing, quality modeling

BUT

- Fairly complex and difficult to apply without a tool

Further Information

- Book
  - Atkinson et. al., *Component-Based Product Line Engineering with the UML*, Addison-Wesley, September 2001
- Web Pages
  - http://www.iese.fhg.de/Kobra_Method/
  - http://swt.informatik.uni-mannheim.de
- Contact
  - atkinson@informatik.uni-mannheim.de