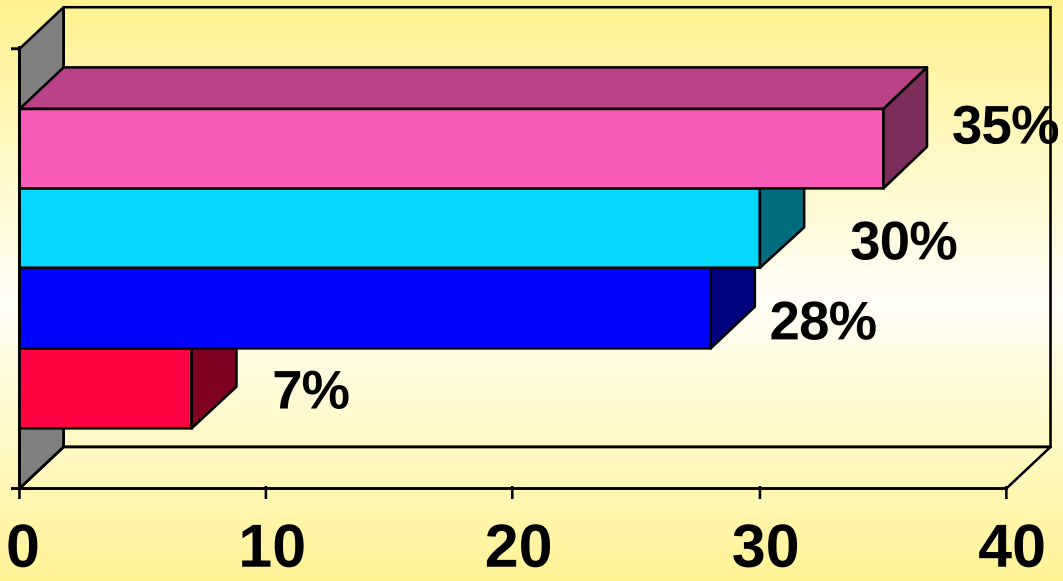
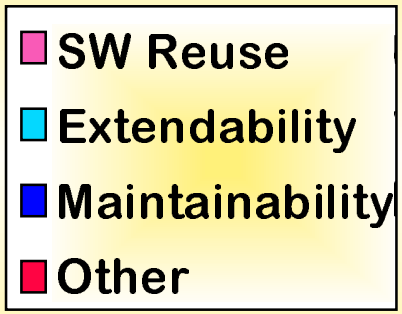
Reuse in Reality
-
The Reuse-Driven Software-Engineering Business

Ivar Jacobson
Rational Software Corporation
2800 San Tomas Expressway
Santa Clara, CA 95051
phone: (408) 496 3600, fax: (408)-496 3973
email: ivar@Rational.com

OO and Reuse

"Reuse is consistently ranked as the top benefit that development managers seek from a migration to object-oriented technology"...





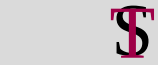
Survey of 100 development managers by Valley Forge Information Services, July 1994.

But Generally speaking: Large Scale Reuse has not yet happened!

- That´s why we still talk so much about it
- We have high expectations from all “OO = Reuse” Hype
- Inheritance ? - Yes, but only low level (fine grain)
- Frameworks and Patterns are both examples of steps in the right direction

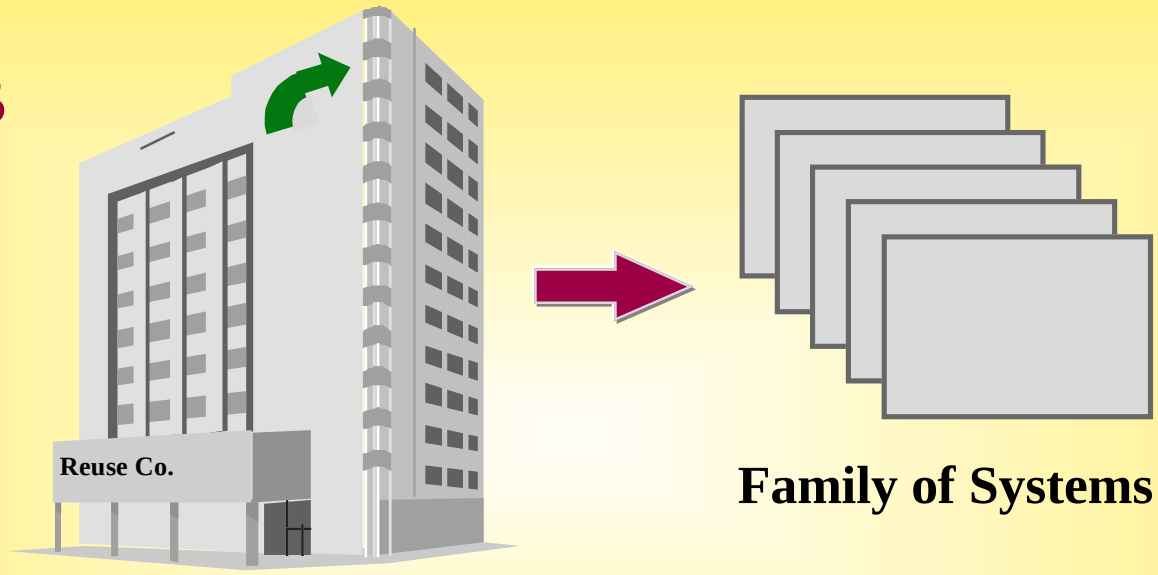
But this is still not enough!

- You have to design your system architecture, your design processes and your organization in harmony to make it happen in reality !



Such an organization is a
"Reuse-Driven Software-Engineering Business"

RSEB



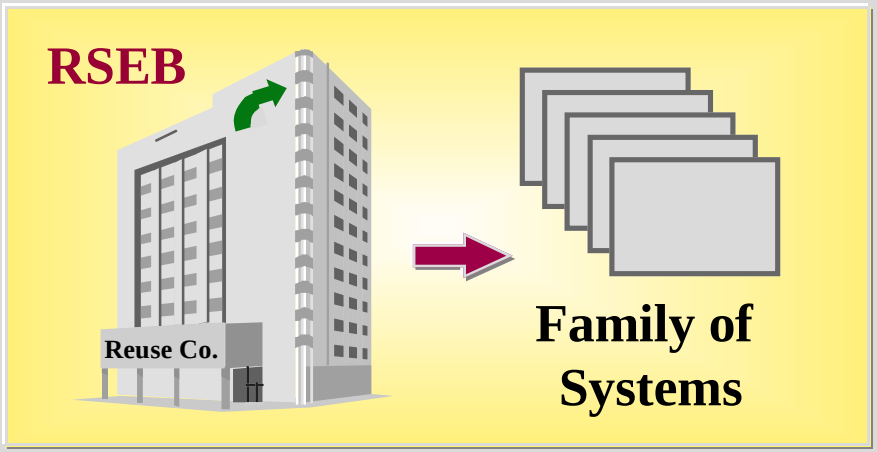
An RSEB, cost-effectively and systematically develops a family of related systems with extensive reuse of components

The Origin of the RSEB

The Ericsson AXE 10 Switching System

A True Success Story on OO and REUSE

The largest commercial success ever in Sweden



OO Software Engineering

> 28 Years of Experiences

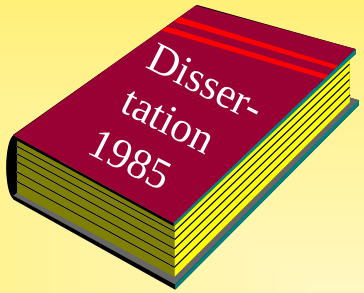
OO Business Engineering

7 Years of Research

> 8 years of R&D within Rational (Objectory)

A >2 year Partnership with experts from HP Labs

The Origin of the Techniques used:

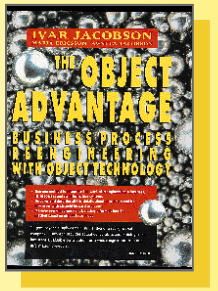


Ivar Jacobson´s approach to Use Case Engineering based on Use Cases and Objects

1992

OOSE - Object-Oriented Software Engineering

OOBE - Object-Oriented Business Engineering



1994

Software Reuse -
Architecture, Process and Organization for Business Success

Software Reuse

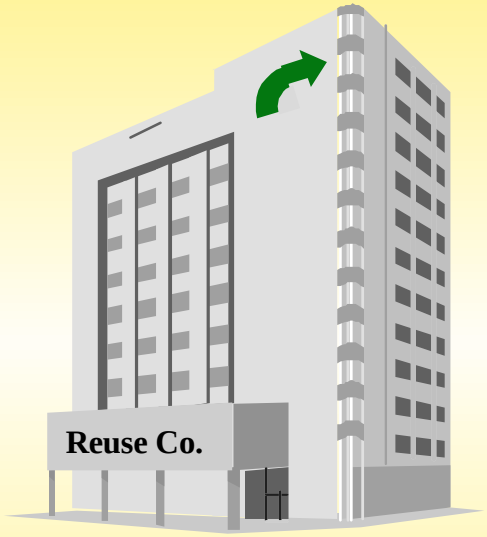
Ivar Jacobson
Martin Griss
Patrik Jonson

Dec 1996

Achieving Substantial Reuse Requires:

- **A**rchitecture
- **P**rocess
- **O**rganization

RSEB - Architecture

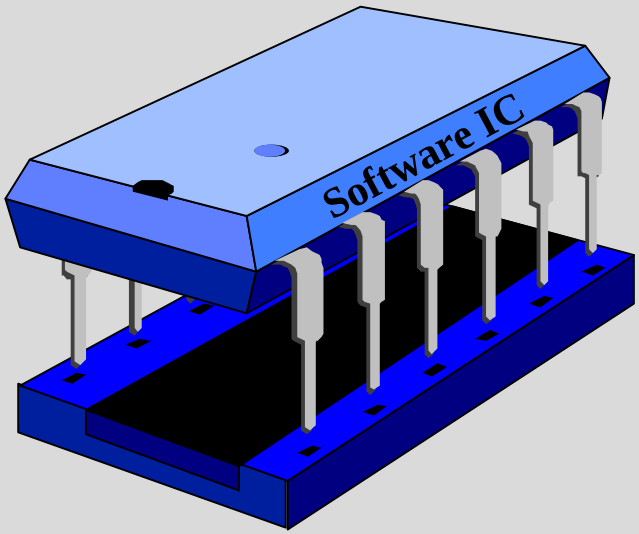


First: We distinguish between Components and Applications

Then: We complement the classic OOSE architectural styles with:

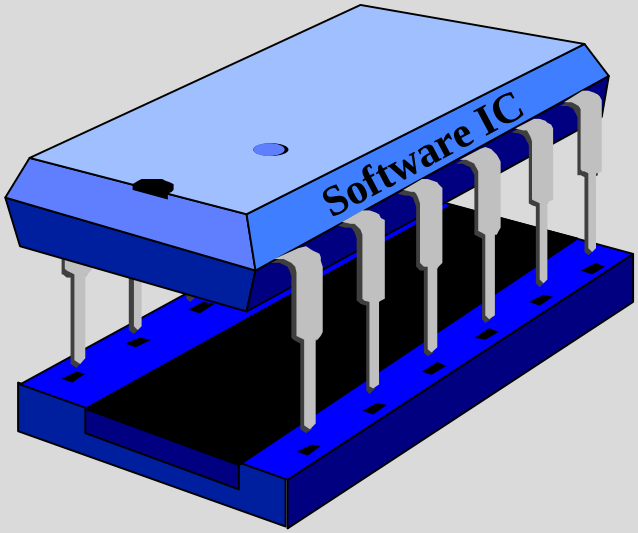
- Interfaces
- Facades
- SIS
- Layered architecture
- Variation Points

Components are Characterized by:



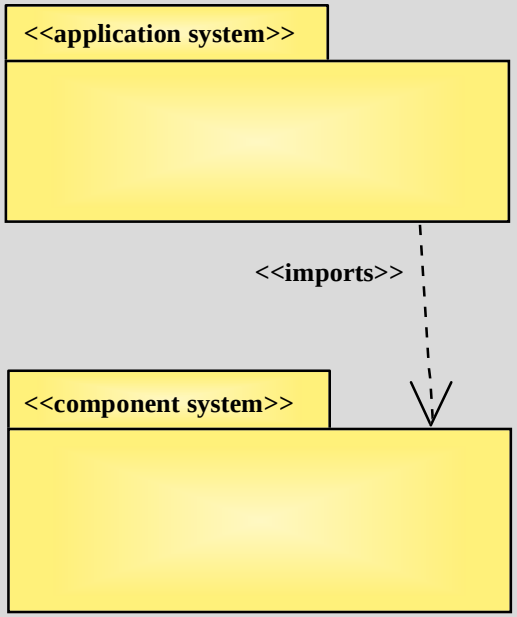
- A good abstraction for higher level design, with access restricted by visibility rules
- Not bound to any specific application
- A high quality product due to careful design and testing
- Packaged for reuse with a well-designed interface, documentation, etc.
- General so that it can be used in several places
- Specialized when used

Our Components are:



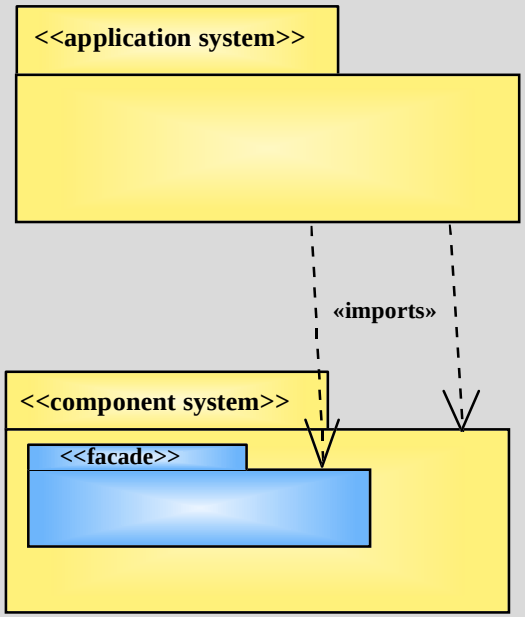
- Actors
- Use Cases
- Classes
- Subsystems
- Frameworks
- Patterns
- Interfaces

Components are packaged into Component Systems



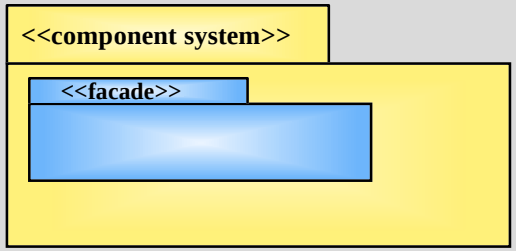
- An **Application System** is a system delivered to customers outside of the reuse business
- A **Component System** provides a set of compatible components used to engineer another application or component system

Component Systems are used through Facades with Visibility Rules



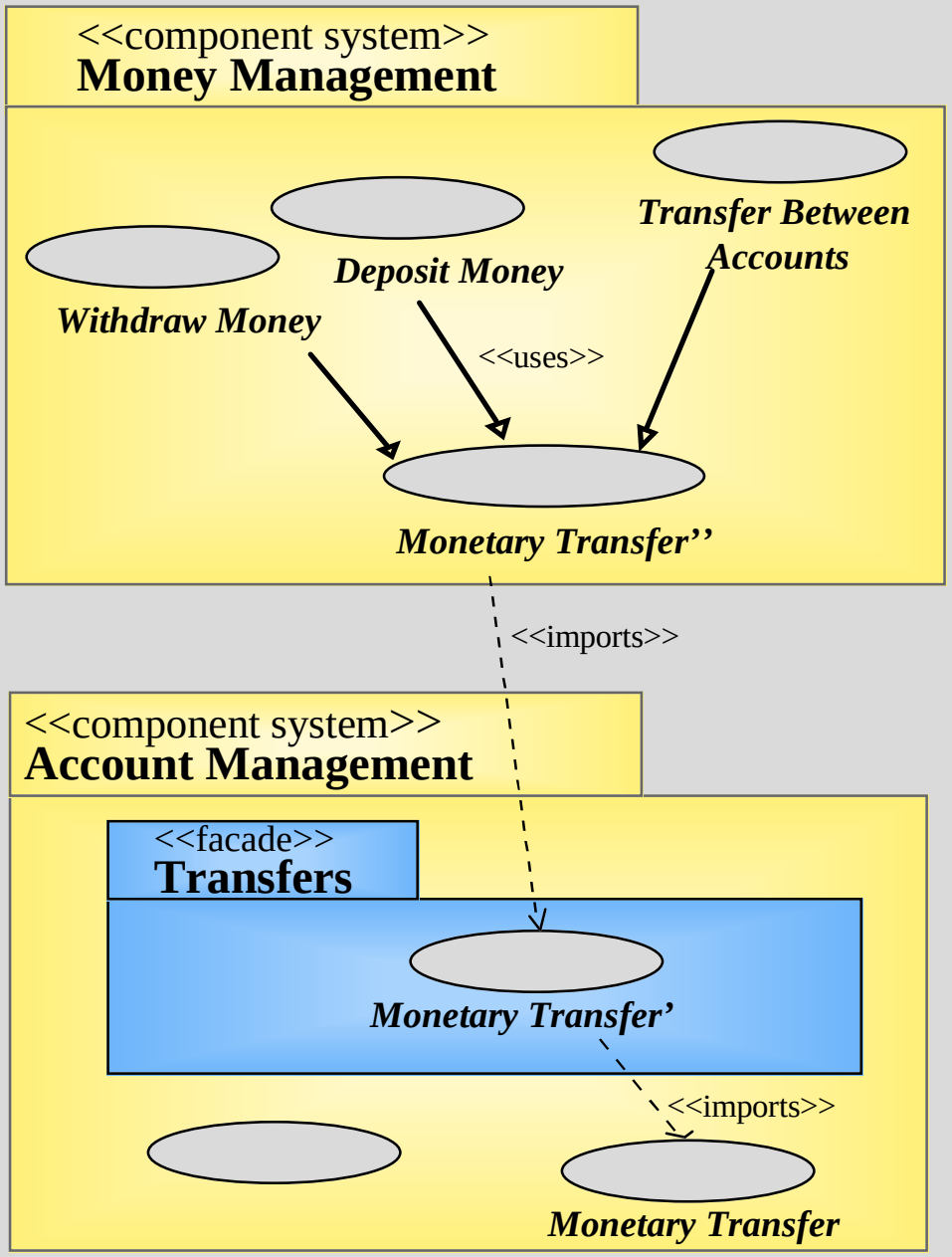
- A system may **import** components from **facades** provided by lower layers
- **Visibility rules** restrict how components in the facade may be used by the importing system
- A **facade** contains (abstract or concrete) components exported from a component system

A Component System Facade

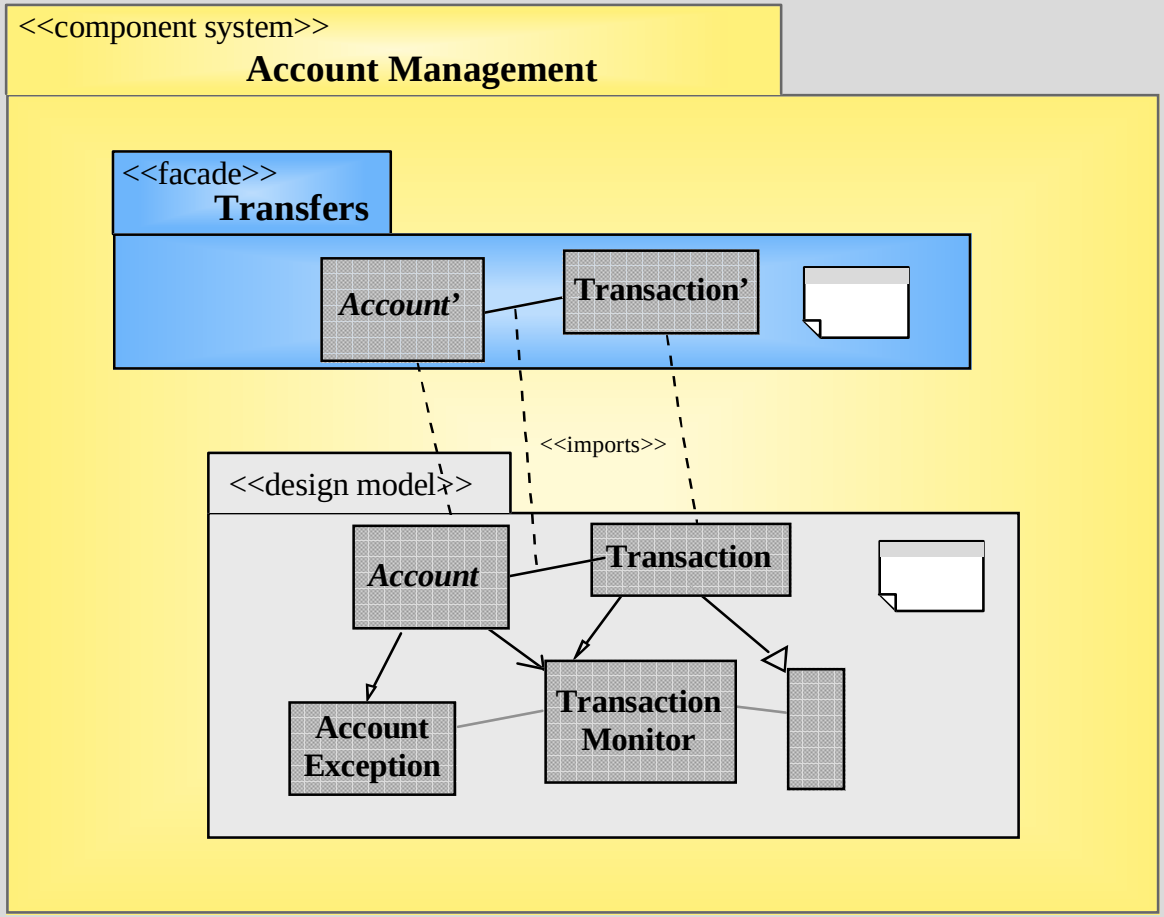


- **A component system facade may contain components that need to be parametrized by the importer**
- **Put each optional component in a separate configuration item**
- **Specify “rules” for what components depend on, or exclude, each other**

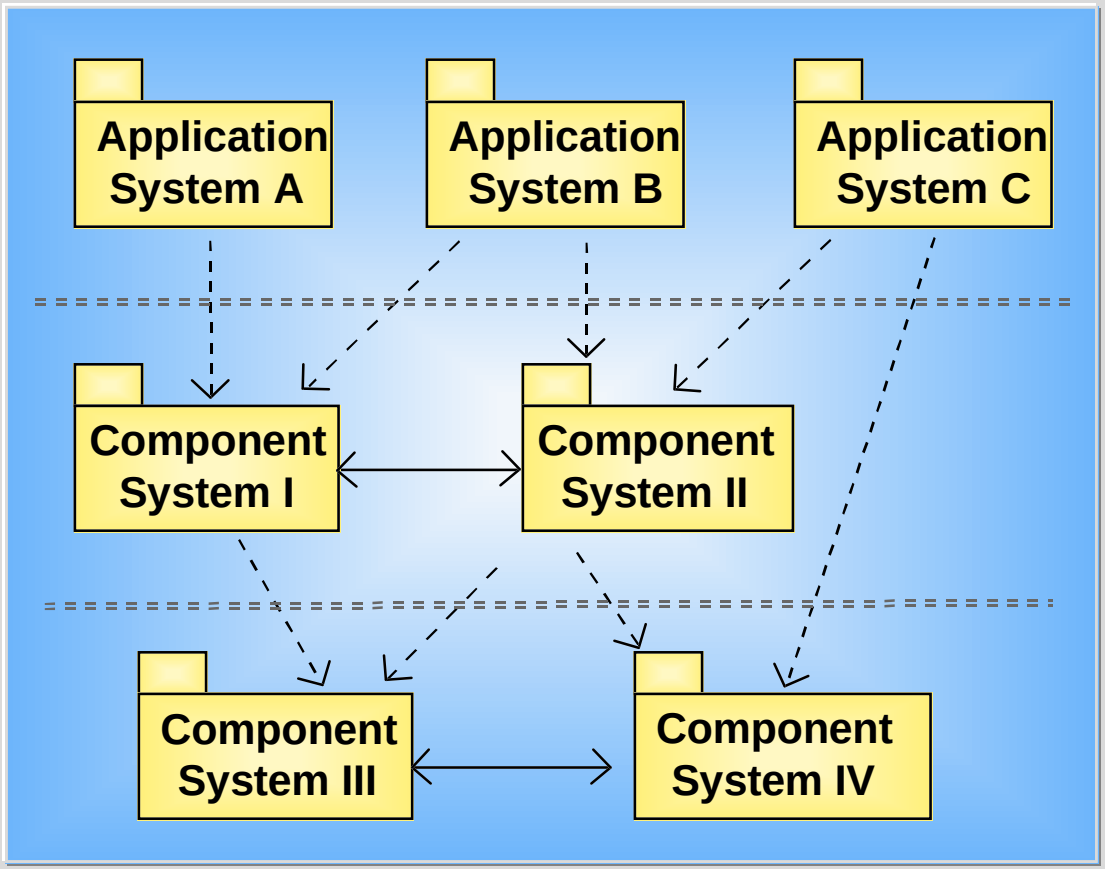
A Use Case Model Facade

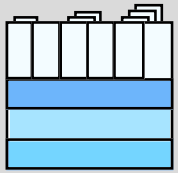


An Object Model Facade



Scaling Up through SIS





A “SIS” is a generalized layered architecture !

- **Systems built from lower-layer component systems**
- **Several interoperable application systems used in a “suite”**
- **Variants of the same application system**
- **Contracts can be used to separate specification from implementation**
- **The pluggability of each component system is defined explicitly**

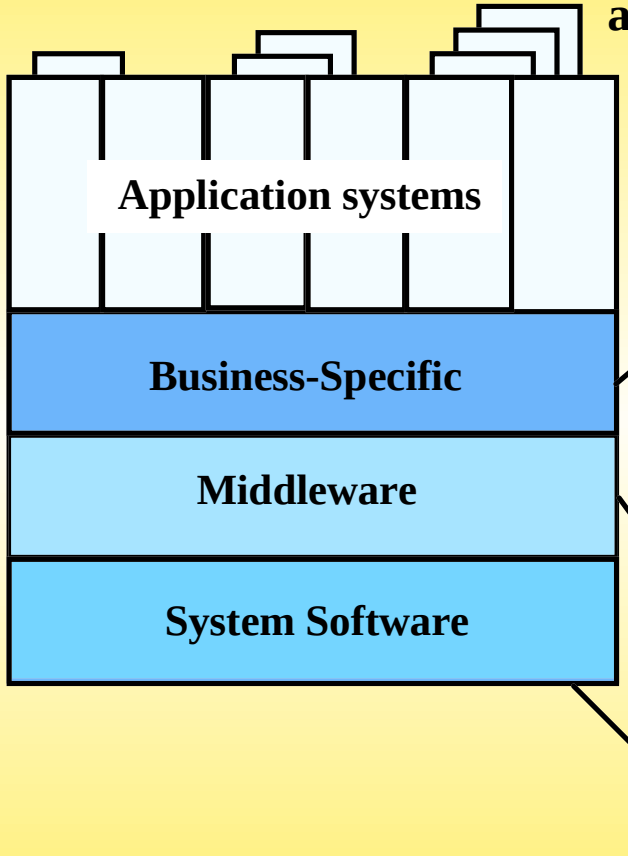
A Layered Architecture

Enables simple and fast development of:

- one or several configurable application systems
- a set of interoperating application systems
- application systems which share common components

Application systems built on top of existing component systems:

- by customization of generic components



Variants of a particular application system

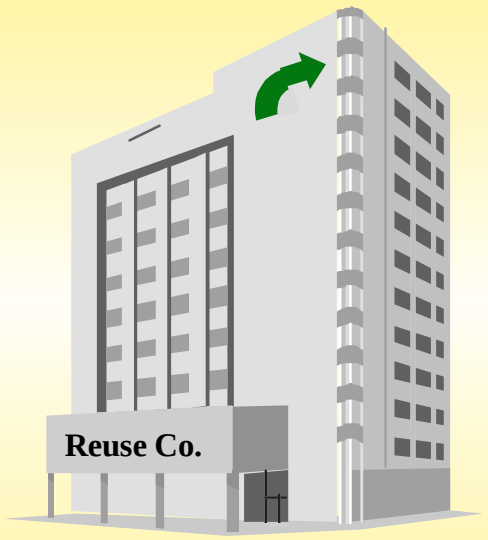
Distinct Application Systems

Domain and org. specific component systems

Encapsulating infrastructure and utility classes, e.g. UI

Platform Specific HW, OS & DBMS´

RSEB - Process



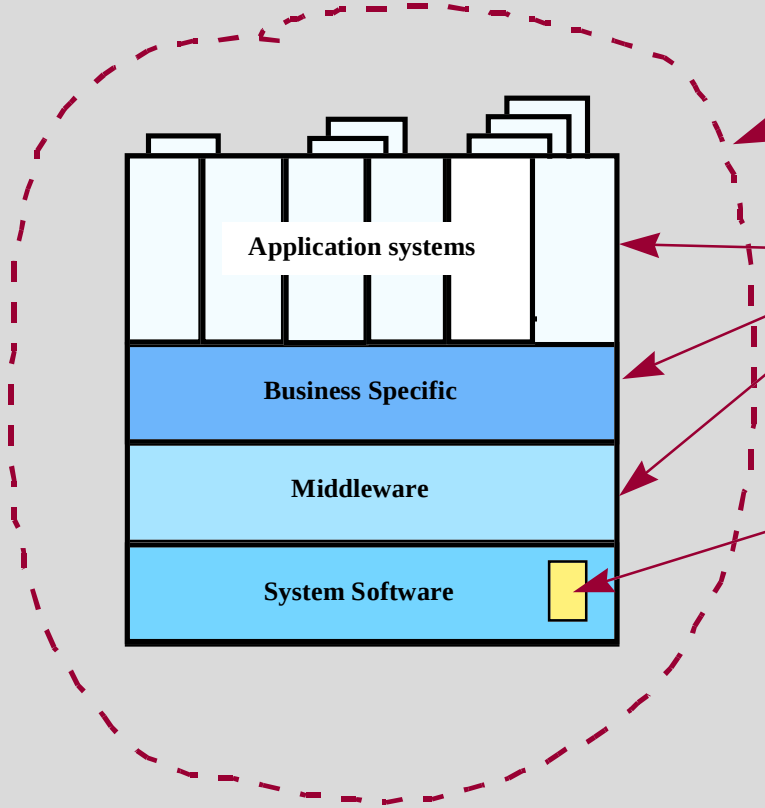
An RSEB runs concurrent processes for

application system development

and

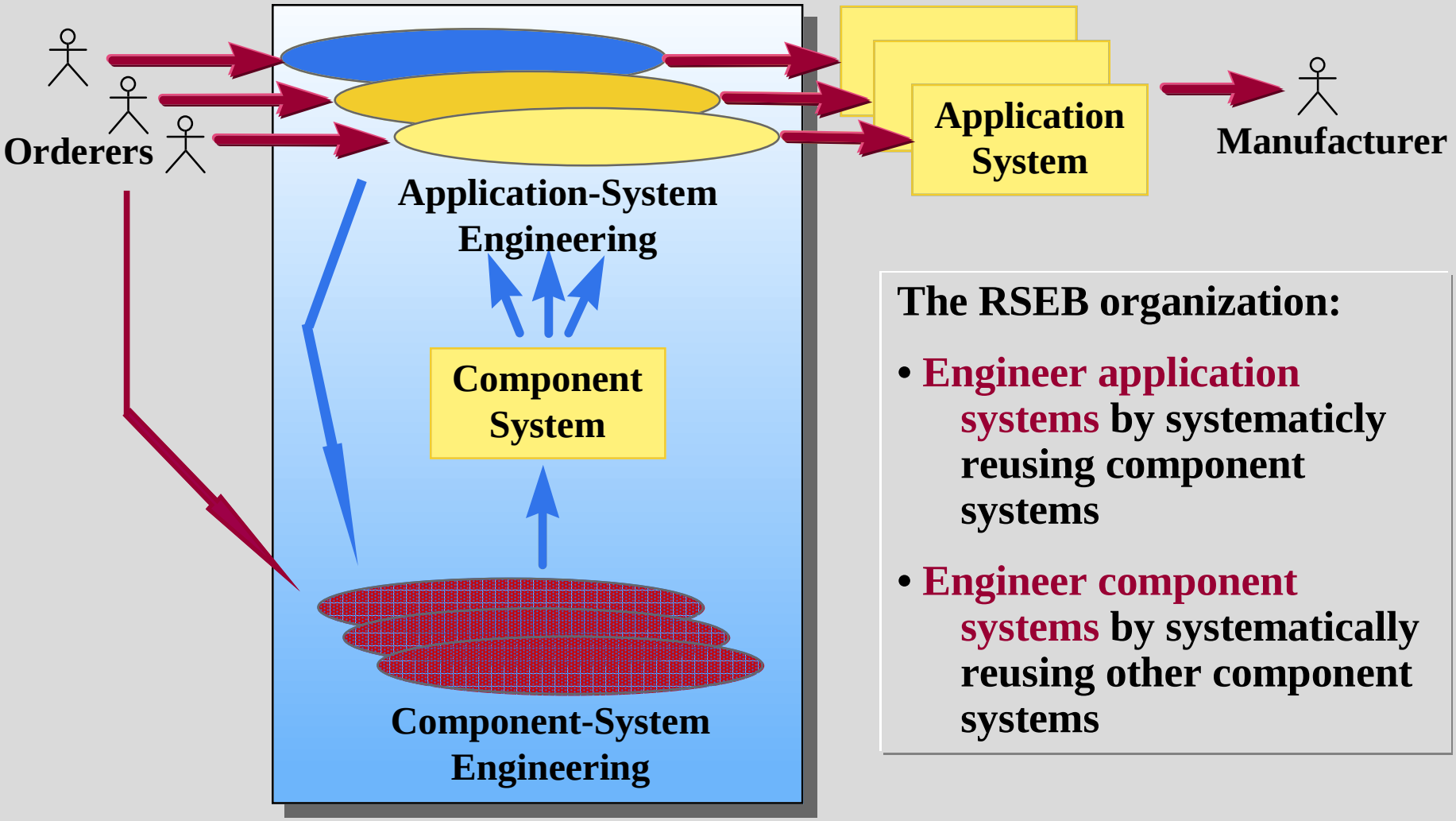
component system development

The different RSEB processes



- One overall process
- One process per application or component system or layer
- Specific processes for specific types of components, e.g. outsourced components or procured frameworks

The RSEB in Operation



The RSEB organization:

- **Engineer application systems** by systematicly reusing component systems
- **Engineer component systems** by systematically reusing other component systems

RSEB - Organization

An RSEB is organized

to match the architecture of the developed systems

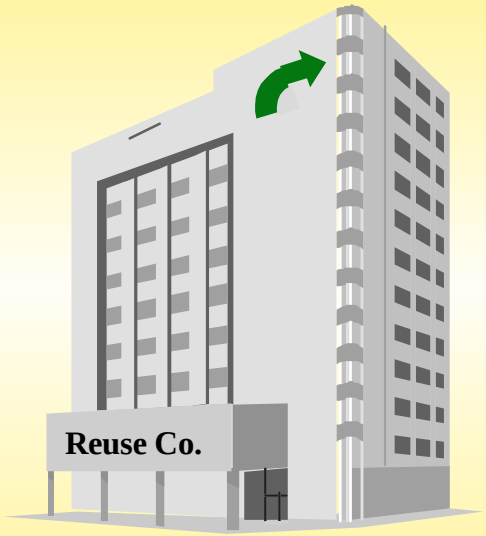
and

to implement the concurrent development processes

Summary of the Core Ideas

	Architecture	Process	Organization
What ?	Applications vs. Components Architectural Styles: • Facades • Interfaces • Layers & SIS • Variation Points	Application Family Engineering (Architecture) Application and Component Systems Engineering	Identify, Design and Implement: • Workers • Work Objects • Competence units
How ?	BE to capture the reqt's on the applications. SE/UML to design the architecture.	BE to define the processes. SE to design the supporting tools.	BE to design and implement the organization.

Reuse in Reality



Architecture

Processes

Organization

} **In Harmony!**

No Use --> Reuse --> Use

So, Will our Reuse-Dreams finally come true ?

- **Well, for some companies, e.g. Ericsson Telecom, it has been a reality, at least to some extent, for more than 25 years !**
- **With an RSEB approach, It can become a reality for others to !**

Making this happen, is probably one of the most important steps to bring our SW Engineering techniques to a state where we can regard it as an industrial strength, engineering discipline