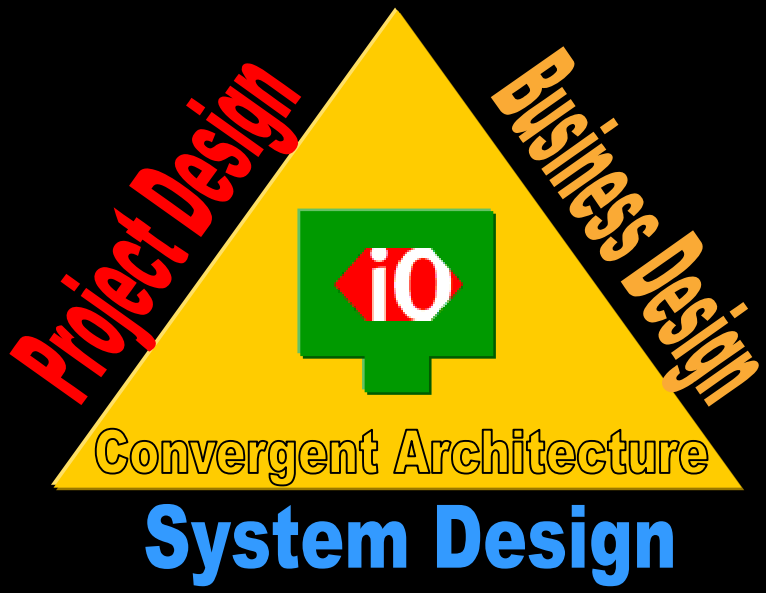
Architectural Style for the Internet Age

Concept, Process and Full Life-Cycle IDE



Dr. Thomas Koch

Interactive Objects Software GmbH
Baslerstrasse 65
D-79100 Freiburg

Web: http://www.io-software.com
Tel. [+49] 0761 / 4 00 73 - 0
Fax [+49] 0761 / 4 00 73 - 73
eMail:koch@io-software.com

© iO Software GmbH

Interactive Objects

Part 1

Introduction of the Convergent Architecture



© iO Software GmbH

Content

- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

What's the point?

A quote from Ken Follett's "Pillars of the Earth" expresses the principle reasons why architecture is important in any design domain.

"'But what's the point of having everything measured by poles? Why not build everything higgedy-piggedy, like a house?'

'First, because it's cheaper this way. All the arches of the arcade are identical, so we can re-use the falsework arches. The fewer different sizes and shapes of stone we need, the fewer templates I have to make. And so on.

Second, it simplifies every aspect of what we're doing, from the original laying-out -- everything is based on a pole square -- to painting the walls -- it's easier to estimate how much whitewash we'll need. And when things are simple, fewer mistakes are made. The most expensive part of building is the mistakes.

Third, when everything is based on a pole measure, the church just looks right. Proportion is the heart of beauty'"

© iO Software GmbH

Classical architecture is a solid basis

IT architecture exists for the same reason classical architecture exists and can thus leverage its principles.

Classical architecture has taught us the importance of, for example:

- Pattern languages
- Named styles
- Engineering as science and an art
- Dealing with complexity
- Project design as part of the architecture
- Industry standard

© iO Software GmbH

IT-architecture adds new dimensions

The importance of architecture as been now recognized by the leaders in the IT industry

IT-Architecture has some aspects not present in classical architecture:

- Dynamics of a running systems
- The nature of software as “soft” (flexible)
- Harder to see than buildings and bridges
  - “If it runs, then the design must be satisfactory?”
  - “What I can’t see won’t hurt me.”
- A relatively new field:
  - Little universally excepted basis of professional knowledge
  - No certification system (Diploma, Meisterbrief)

© iO Software GmbH

Architecture is “design-in-the-large”

*An **architecture** **is** a useful description of how a system is to be built.*

An architecture describes how well defined components are composed into a stylistically harmonious system that consolidates trade-offs.

RHT, iO GmbH 1996

Good architecture is:

- easy to see, hard to define, very real.

Style is important! **“Architecture”:** ***A** style **and method of design and construction. The** art and science **of designing and erecting [systems]***

Heritage Dictionary

© iO Software GmbH

Interactive Objects

IT architecture is high ROI

IT-Architecture is an important strategic factor for companies that want to thrive, or survive, in the Information Age

- Is a means of effective communication
- Supports organizational and process decisions
- Enables a planned, engineering approach
- Enables the selection and controlled leveraging of “best-fit” external technologies
- An Architectural Style helps take a “less is more” approach and focus on adding value
- Enables reusable, “checklist” type template based design and development

© iO Software GmbH

We can learn a lot from nature...

Observing living systems with respect to our IT problem domain indicates that there is still a lot of progress to be made.

- **Living systems have extensive feedback mechanisms.**
- **Vision is our most effective means of perception.**
- **Living systems are very fault tolerant where they should be.**
- **Living systems adapt and improve incrementally.**

© iO Software GmbH


„DIE ERSTEN BLÜTEN:
AM 24. APRIL 12½ UHR.
DIE ERSTEN FRÜCHTE:
HANDELSKLASSE A
AM 18. JULI 17¾ UHR.
IST DAS KLAR?!"

Content

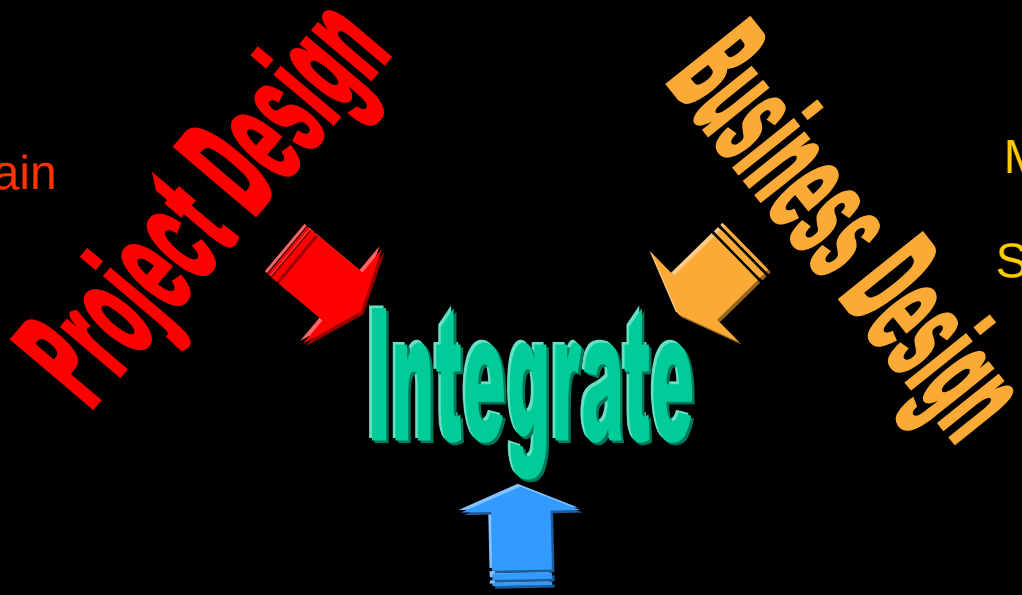
- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

An arch. style must integrate 3 themes

Arch. Styles are only complete when they cover 3 intimately related themes.

Models the domain of IT-System Development



Models the business for which the IT-System is being built.

Rational's Kruchten (RUP): "Focus on architecture first", "The process as a product".

Maps the respective business models "convergently" to available technology.

Bertrand Meyer: "Eiffel, more appropriately characterized as a method..."

© iO Software GmbH

Interactive Objects

An architectural style ...

positions and tightly integrates metamodels, process, tools and technology projections into a holistic approach.

A complete architectural style comprises:

- A clear architectural metamodel (Meta-2)
- A component metamodel (Meta-1)
- A full life cycle development process
- Formal technology projections to open standards
- A full coverage tool suite assisting style conform design and implementation

© iO Software GmbH

Content

- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

Interactive Objects

The CA Architectural Metamodel (M2)

Defines the vision and 7 basic principles of the architectural style.

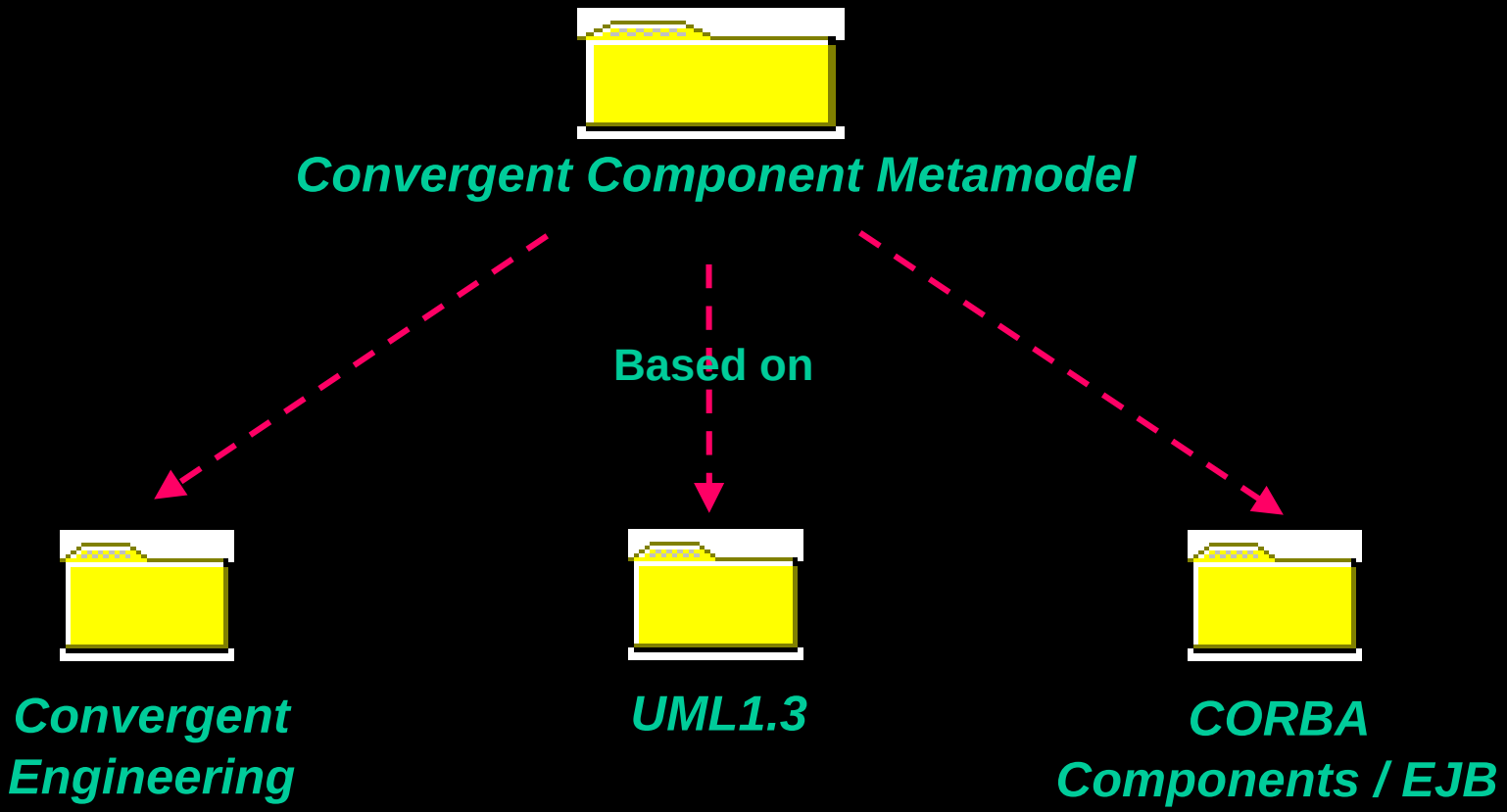
- **The 3 pillars of holistic architecture**
- **Convergence / Convergent Engineering™**
- **Machine Shop Metaphor**
- **Reduced Abstraction Set Computing ™**
- **Conceptual Isomorphism**
- **Component Metamorphosis**
- **Technology Categories and Projections**

See Article in OBJEKTSpektrum Juli/August 1999

© iO Software GmbH

The CA Component Metamodel (M1)

is a convergent RASC metamodel which leverages the non-convergent UML, EJB and CORBA Components metamodels



See articles in OBJEKTSpektrum Nr. 1/97, 3/99, 1/2000

© iO Software GmbH

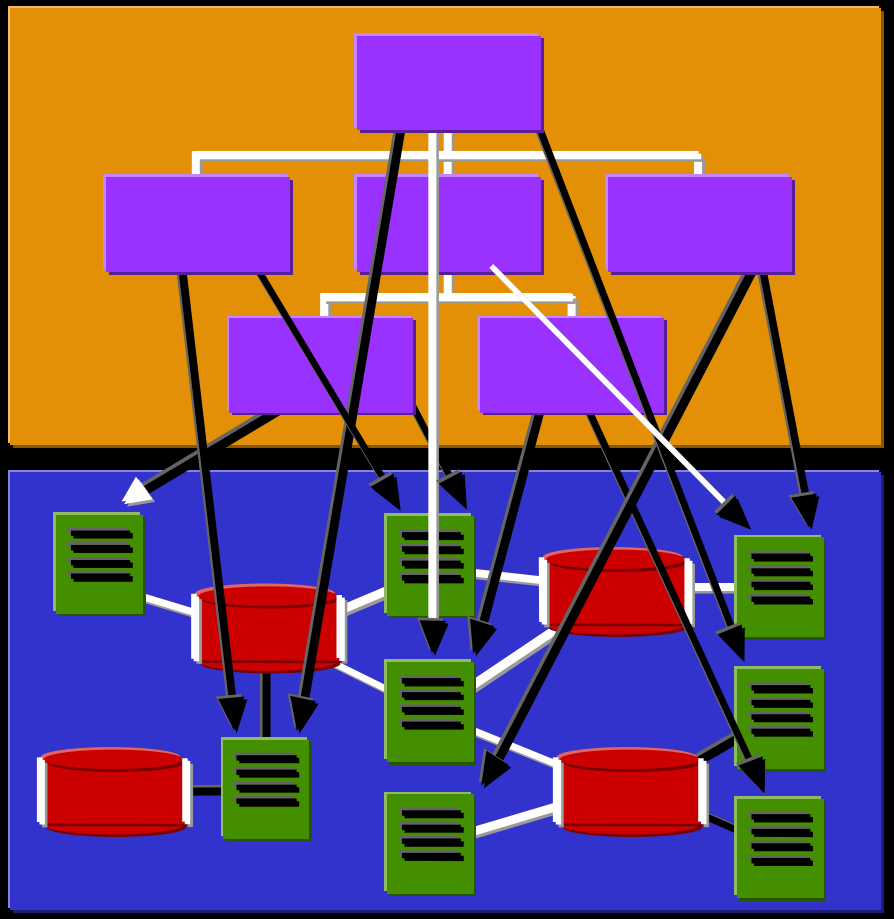
Interactive Objects

A major problem: Conceptual Divergence

The translation loss from a business model (or no business model), to a design, to an implementation and deployment technology is the most significant source of today's IT problems: Divergent engineering.

Business System

Software System

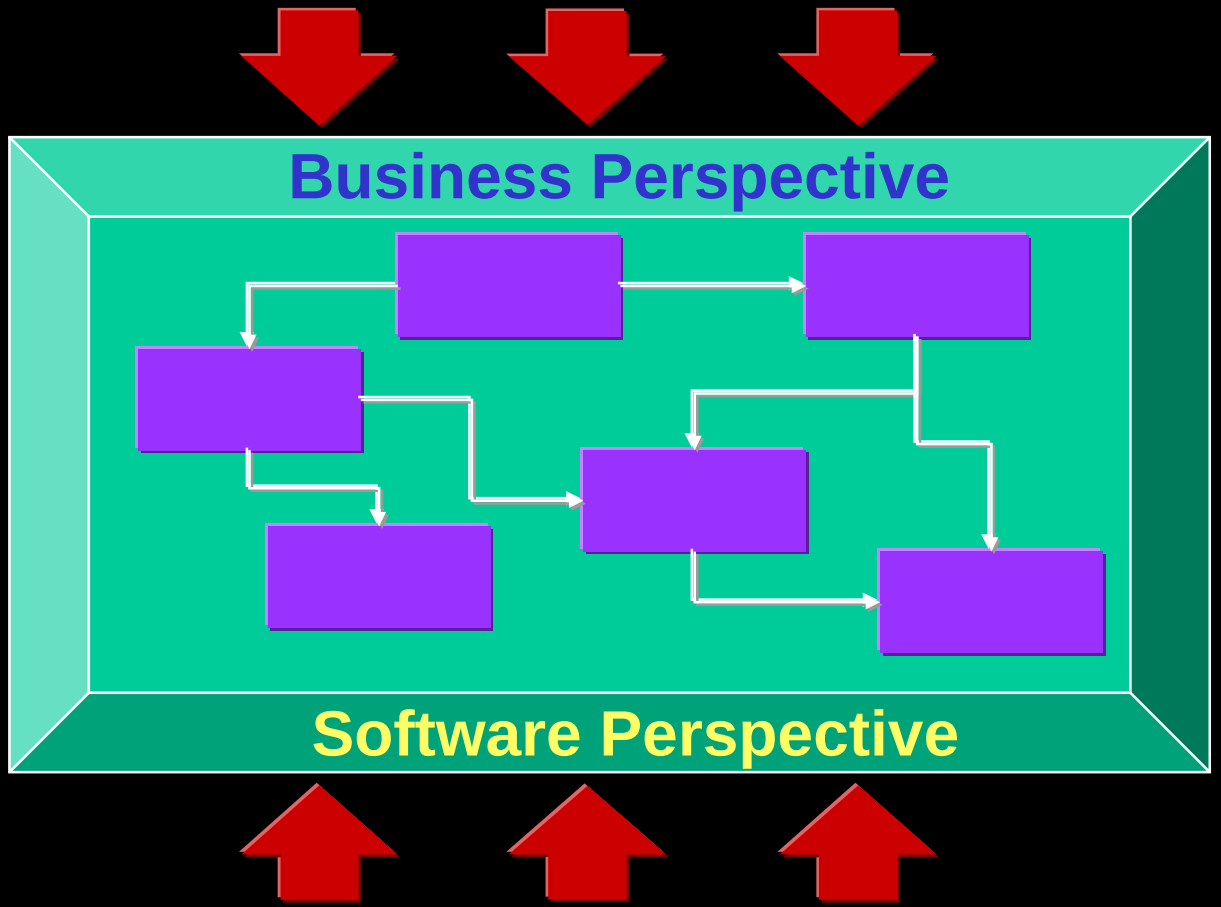


© iO Software GmbH

Interactive Objects

The solution: Convergence

*The Convergent Engineering solution is the **synthesis of business and software engineering** - isomorphic views of one system.*

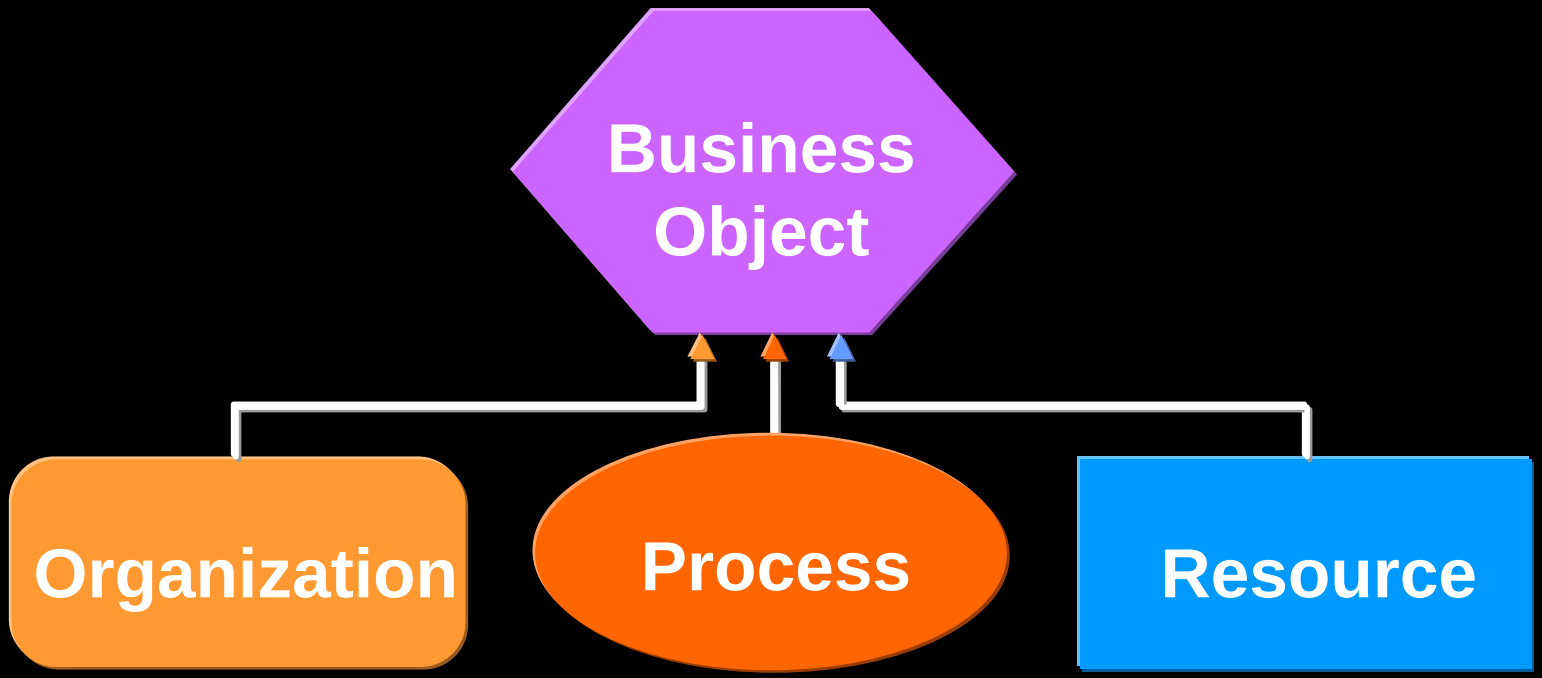


© iO Software GmbH

Interactive Objects

Reduced Abstraction Set Computing

Organizations, Processes and Resources are the most suitable abstractions. The Convergent Component Model maps them to IT.

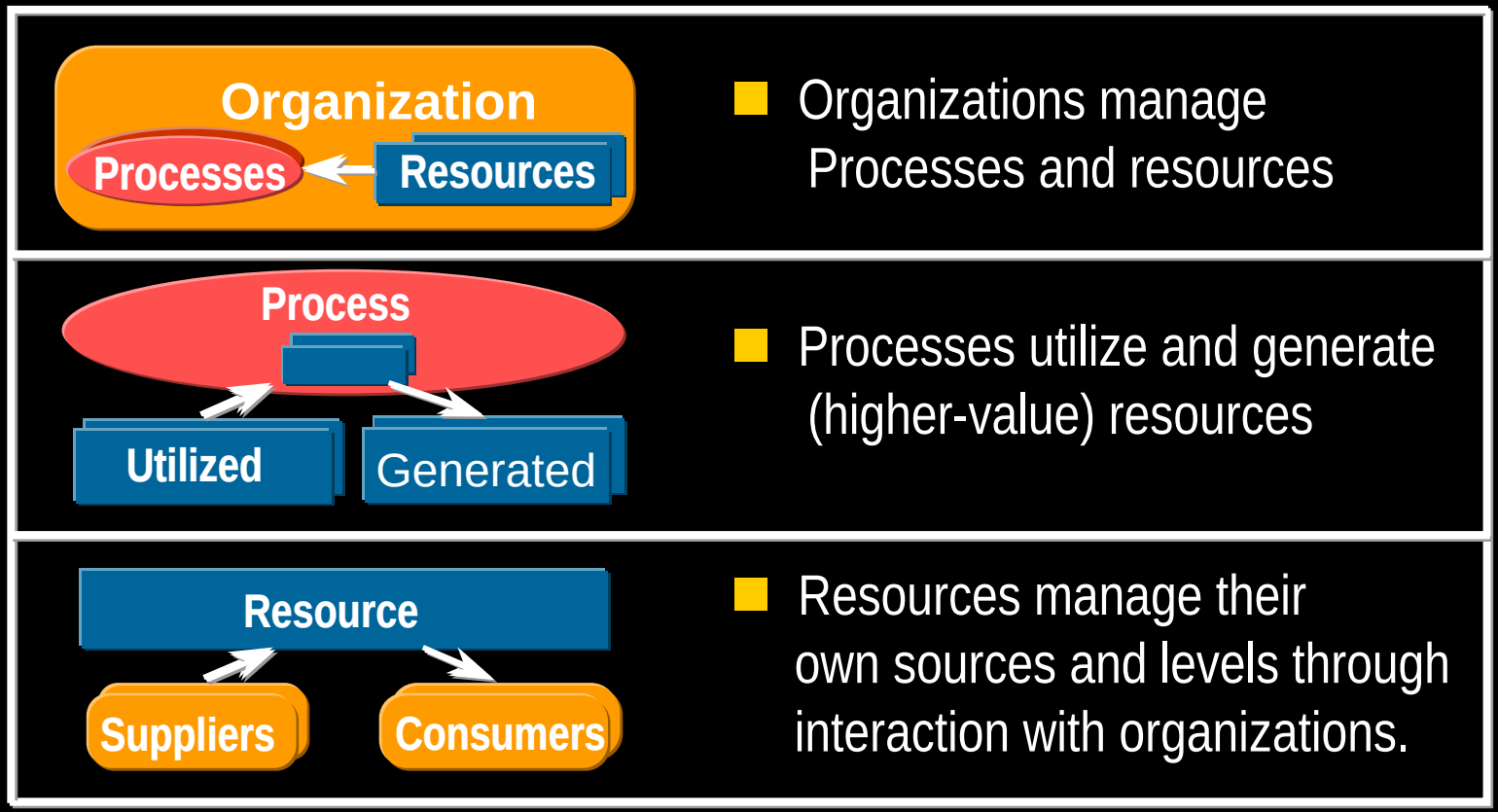


Three subclasses inherit abilities of business object
They interact to implement the business semantics

© iO Software GmbH

Provide universal semantics

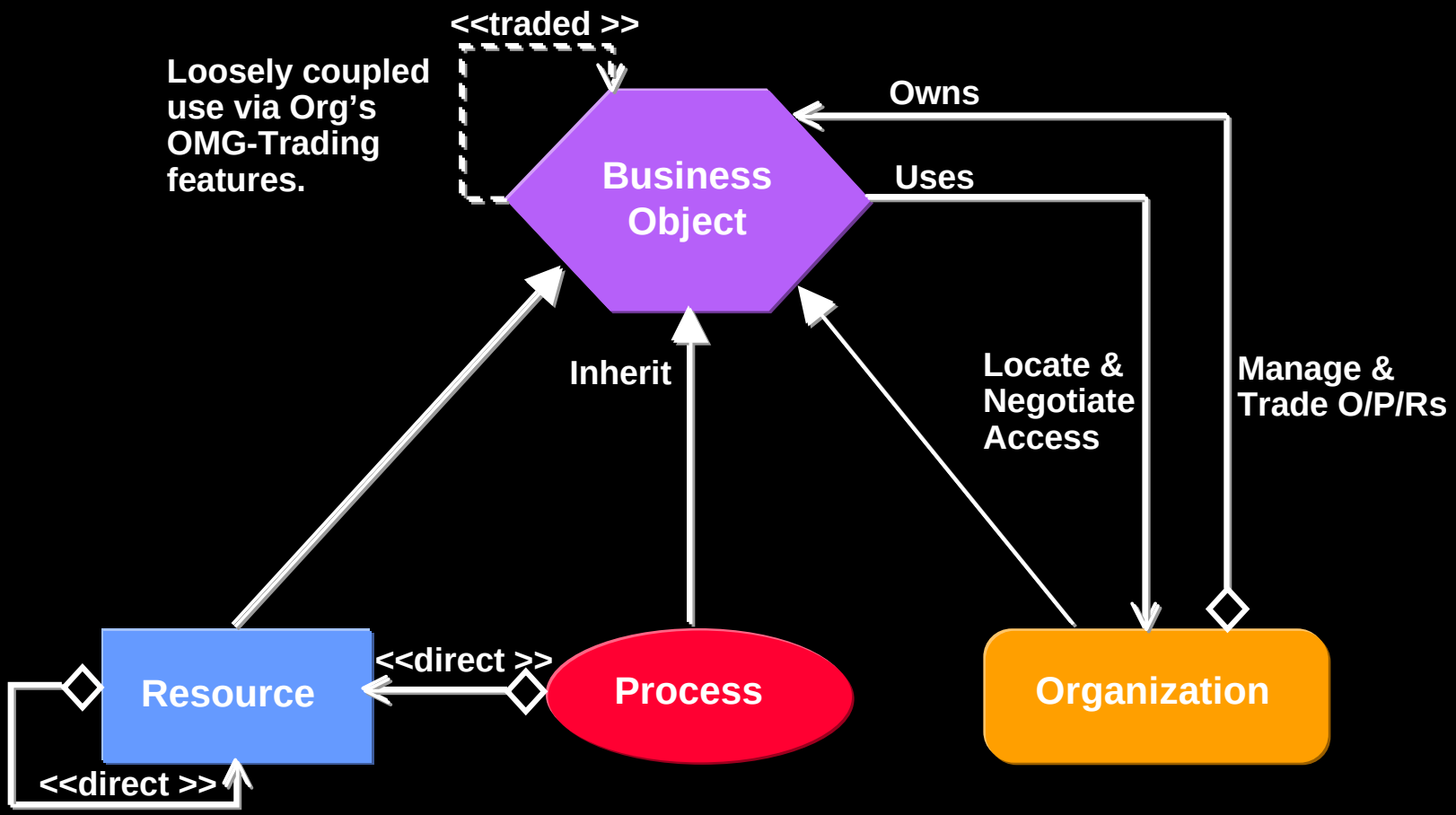
These OPR business abstractions ***permit us to now object-model any part of any business*** *using a simple but expressive pattern language.*



© iO Software GmbH

Convergent Engineering™ Based Metamodel

We start at the level of business design and the represented business instances. These OPRs align the business and IT representations.



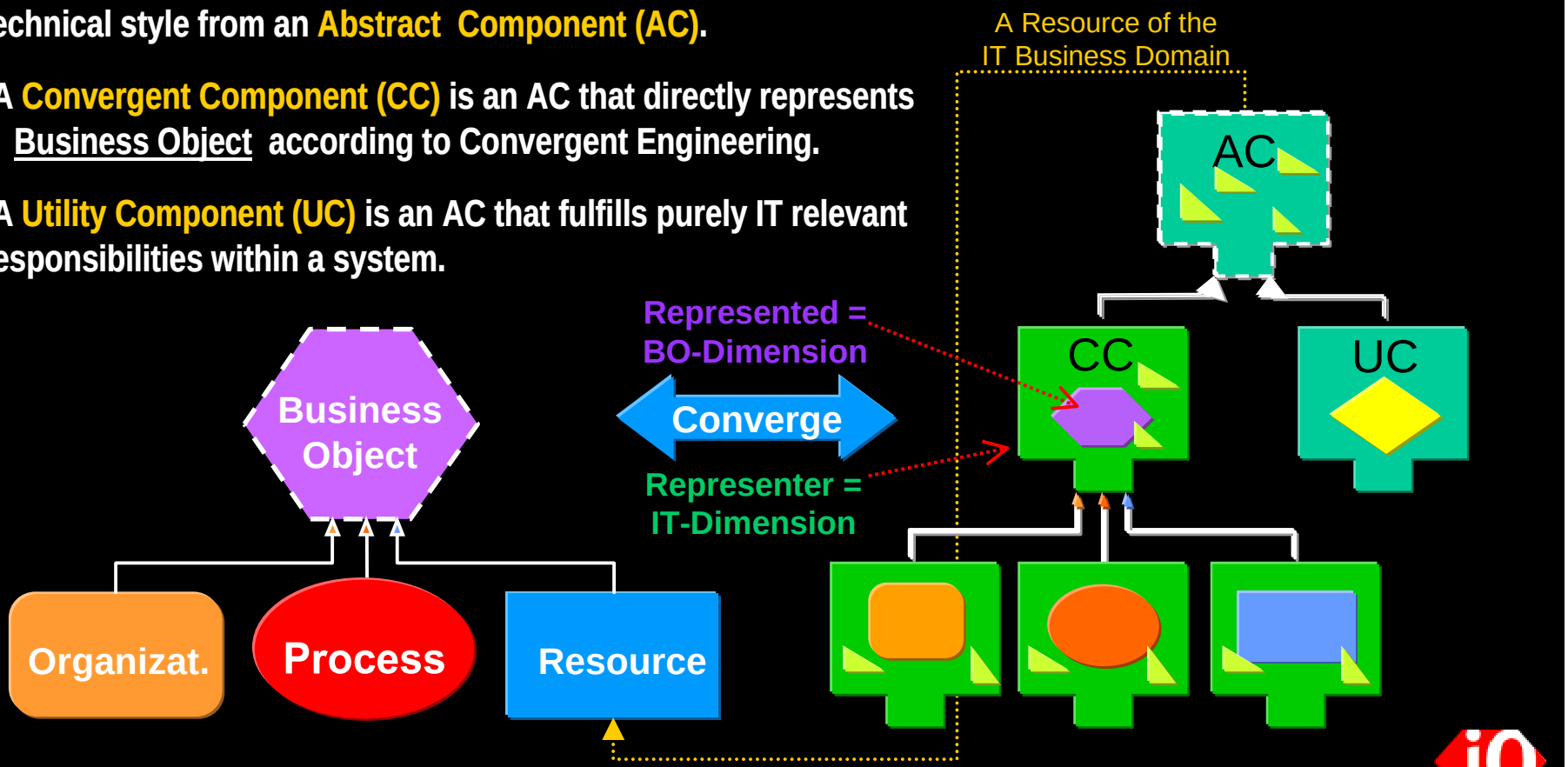
© iO Software GmbH

Interactive Objects

The Top Level Component Hierarchy

Convergent Components are IT Resources that represent business objects. They are the building blocks that enable us to realize our architectural vision.

- In the Convergent Architecture all components inherit abstract technical style from an **Abstract Component (AC)**.
- A **Convergent Component (CC)** is an AC that directly represents a Business Object according to Convergent Engineering.
- A **Utility Component (UC)** is an AC that fulfills purely IT relevant responsibilities within a system.

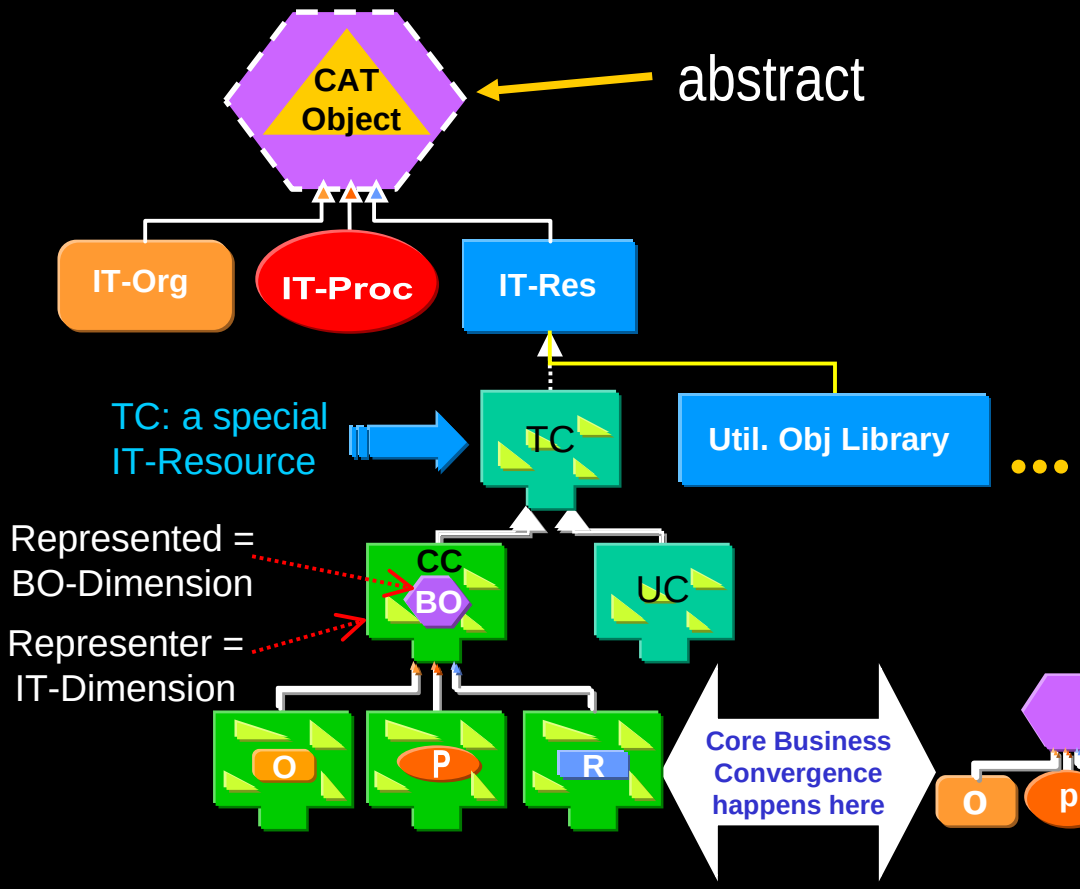


© iO Software GmbH

Interactive Objects

The Convergent Architecture Metamodel

Simplification through conceptual isomorphism.



The IT Domain:
IT OPRs,
the Rs having
two jobs.

Business Domain:
Convergent
Representations of
the core
business OPRs

© iO Software GmbH

Interactive Objects

The Convergent Architecture

builds on the cornerstones of the Convergent Engineering® vision.

The CA is a systematic approach to solving many common problems in most of today's designs by:

- addressing the source of our IT problems and removing them instead of dealing with their symptoms.
- providing an architecture driven template to integrate evolutionary Project Design, Business Design and System Design
- evolving over time to rigorously consolidate monopolized knowledge into accessible (reusable) knowledge.

© iO Software GmbH

Content

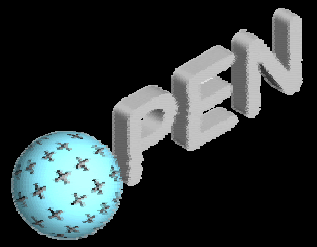
- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

Interactive Objects

An Instance of The OPEN Process

We refine the OPEN full-cycle process architecture which encompasses Taylor's CE principles, an expressive modeling language (UML++) and 3rd generation techniques



- **OPEN advocates a process encompassing:**
  - Methodology for full life cycle OO coverage... (OPEN p. 27)
  - People and organizational influences
  - Technology support for the process
- **OPEN advocates Convergent Engineering:**
  - "The techniques of Convergent Engineering are fully and readily integrated into OPEN." (OPEN p. 23, 86)
  - "We note that we are working within a Convergent Engineering environment." (OPEN p. 49)
- **A Software Engineering Process Architecture (SEPA)**
  - must be configured in to a specific *instance*

© iO Software GmbH

An Instance of The Rational Unified Process

We refine the RUP full-cycle process architecture which is extremely similar to OPEN – may be seen as an instance the OPEN SEPA.



- **Architecture-centric Process**
  - Process as part of the Architectural Style
- **Business Use-Case Driven**
  - Business Design Theme drives System Design
  - Convergence
- **Iterative Development**
  - RUP confirms what iO has practiced for 10 years
    - "iO Hurricane"
  - Effective iterative development via T-Technique.
- **Also a SEPA approach**
  - must be configured in to a specific *instance*

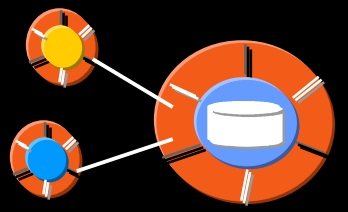
© iO Software GmbH

Incremental Introduction Possible

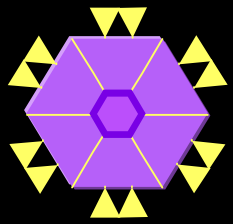
The process may start with the simple leveraging of OO concepts and continues with a goal oriented approach toward optimized information age business. Over time, the process involves the whole enterprise.



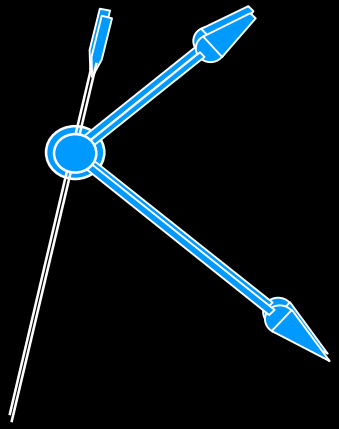
Business Optimization



Added value OO Apps

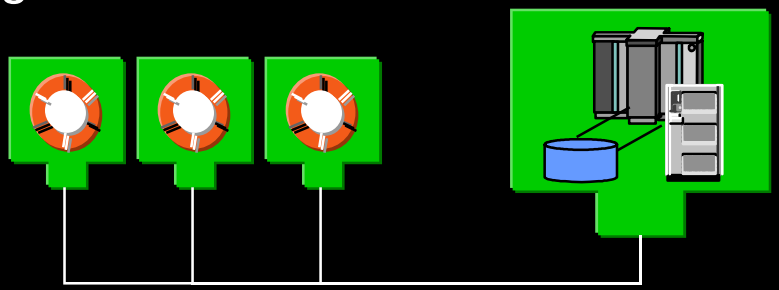


Integrated Modeling





Flexible Integration



Legacy Leveraging

© iO Software GmbH

Interactive Objects

The top level process is simple and effective

Business components evolve incrementally. The OPR model is often the first formal model a company has ever done. It is a real eye-opener, presenting management with many opportunities for improvement regardless of the supporting IT-System.

Phase 1: Discovery

1 Identify the critical organizations of the business

2 Analyze the core processes of each organization

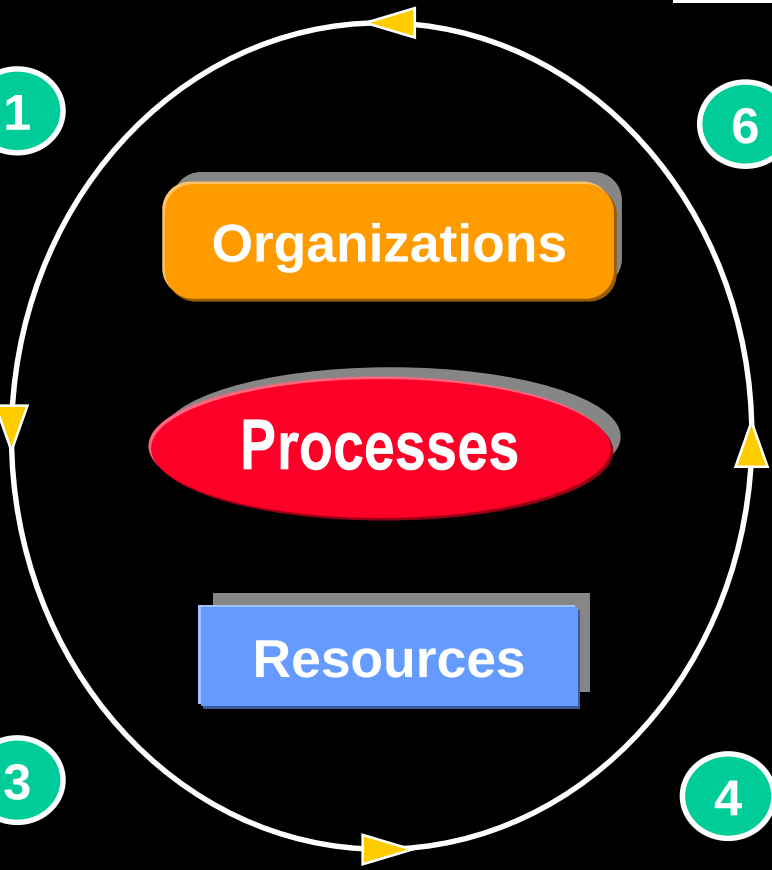
3 Determine the essential resources required by each process

Phase 2: Invention

4 Optimize resources and make them self managing

5 Simplify the processes to reduce time and cost

6 Reorganize around the new business processes

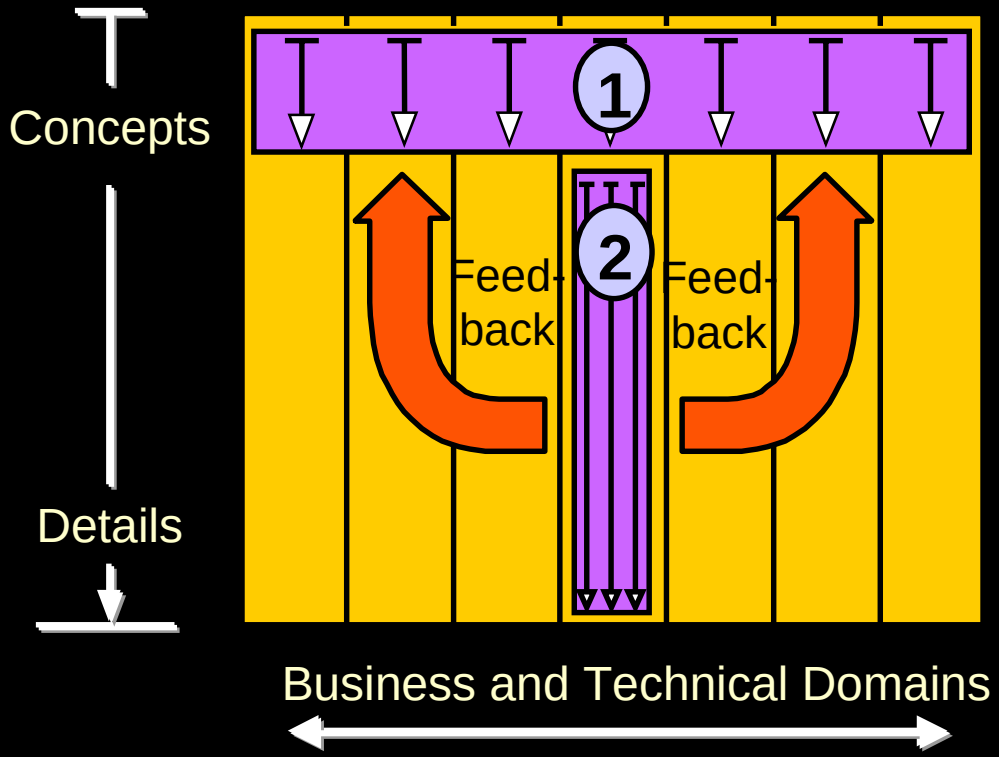


© iO Software GmbH

Interactive Objects

The CA applies the T-Technique

We apply the Taylor T-Technique in an iterative fashion.
Partitioning and feedback to constantly reduce risk.



Horizontal

- contextual requirements
- horizontal system requirements
- architectural style

Vertical

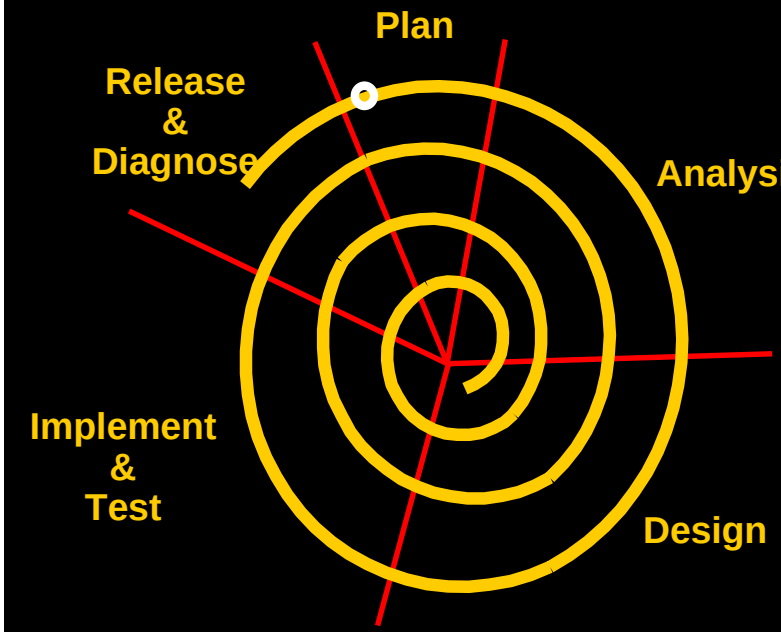
- vertical system requirements
- high risk areas
- “applications”

© iO Software GmbH

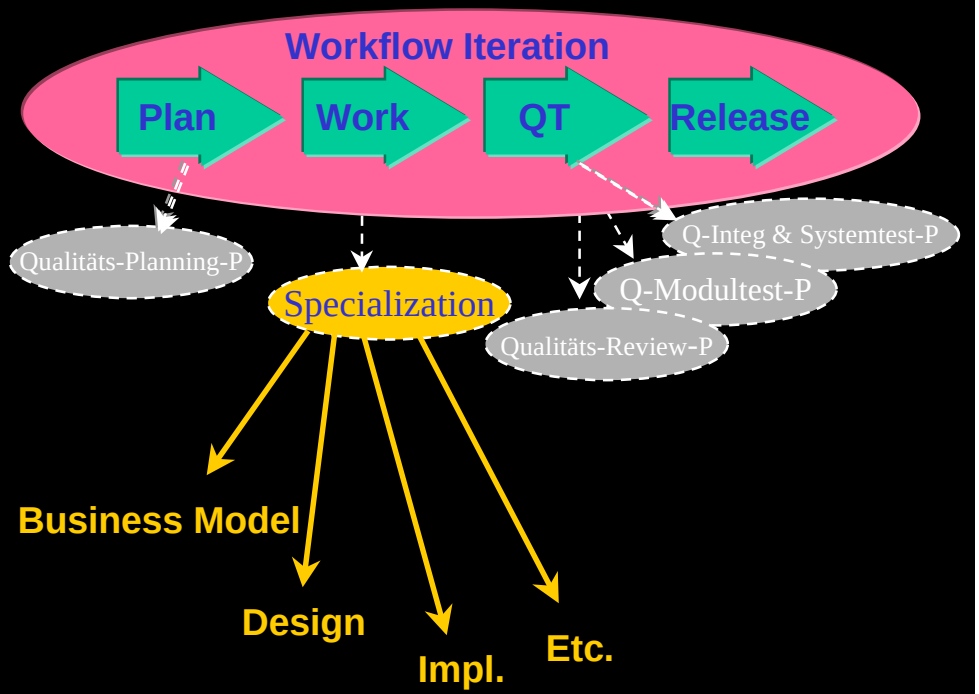
Interactive Objects

Flattening the iteration spiral

We implement our abstract "spiral" in the form of fractally nested Workflow Processes. Each of these inherits from the Base IT Process which enforces the "Plan-Do-Study-Act" paradigm.



Abstract &
Generalized



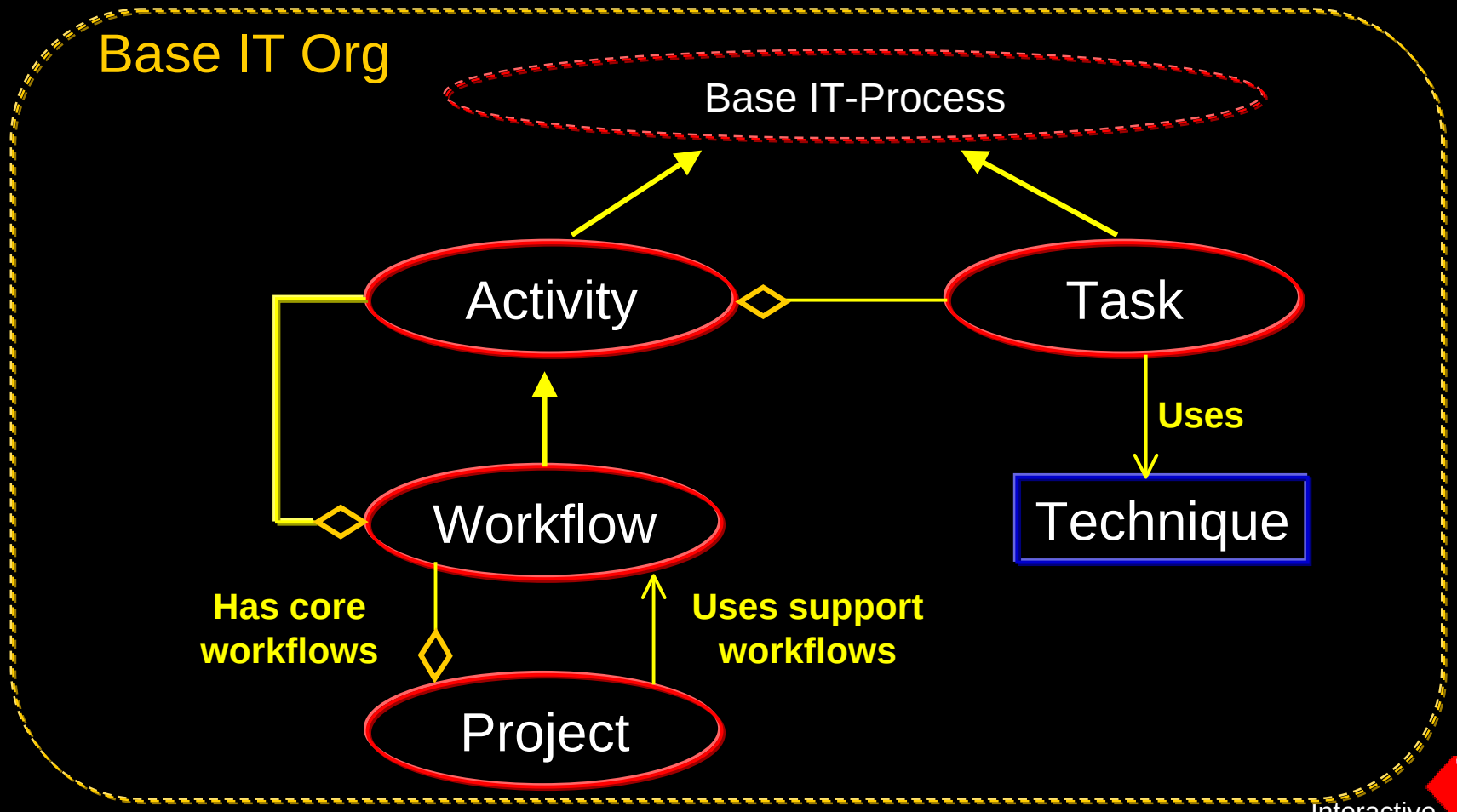
Concrete application via Base
IT-Process patterns.

© iO Software GmbH

Interactive
Objects

The IT-Process Metamodel

The RUP/OPEN IT-Process relationships.

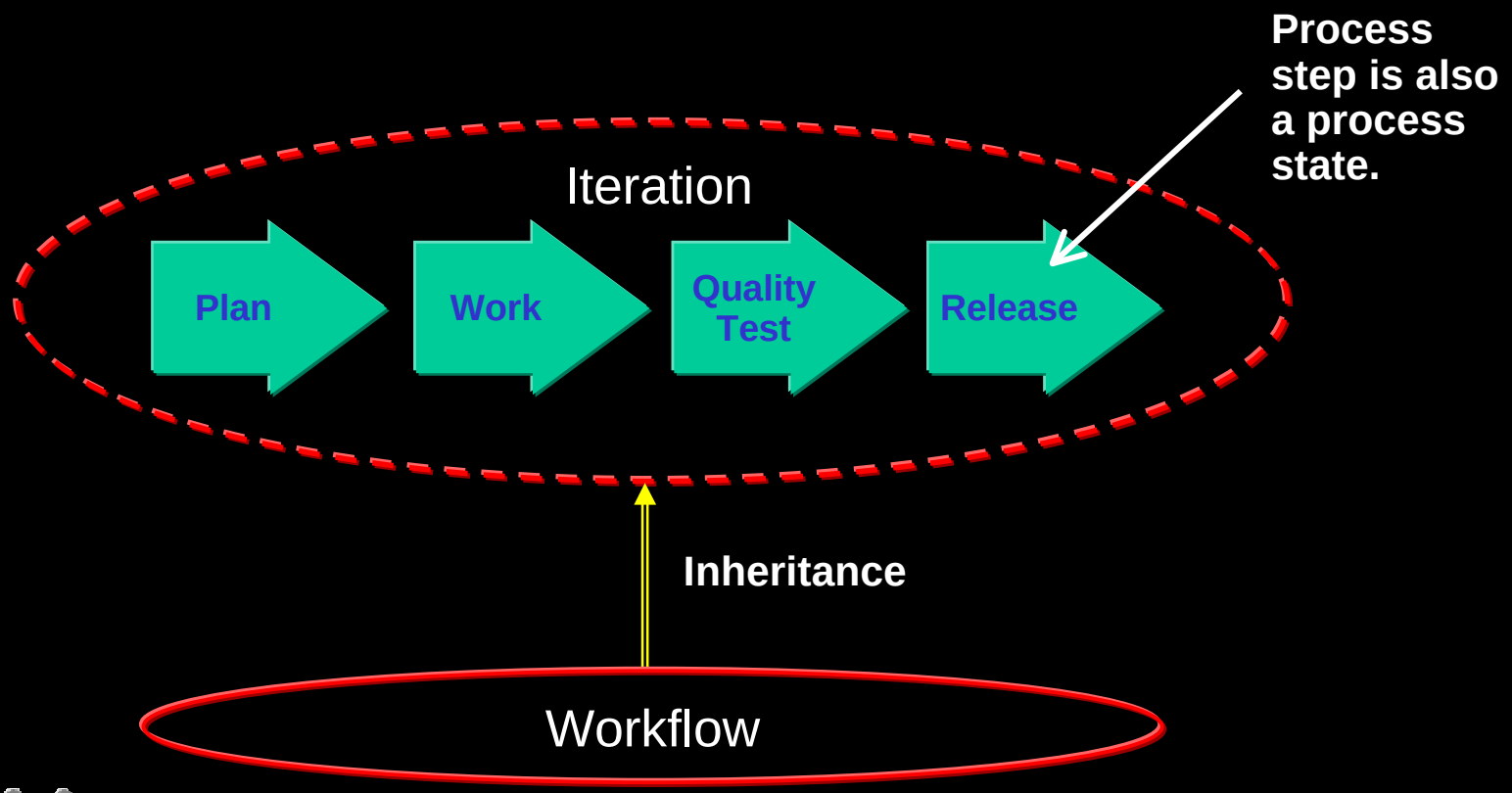


© iO Software GmbH

Interactive Objects

The Base Iteration Process: a pattern

Each Process adheres to, and specializes upon a universally applicable iteration pattern. The Base Iteration Process consists of four Process States: Plan, Work, Quality and Release.



© iO Software GmbH

Interactive Objects

The steps in each iteration

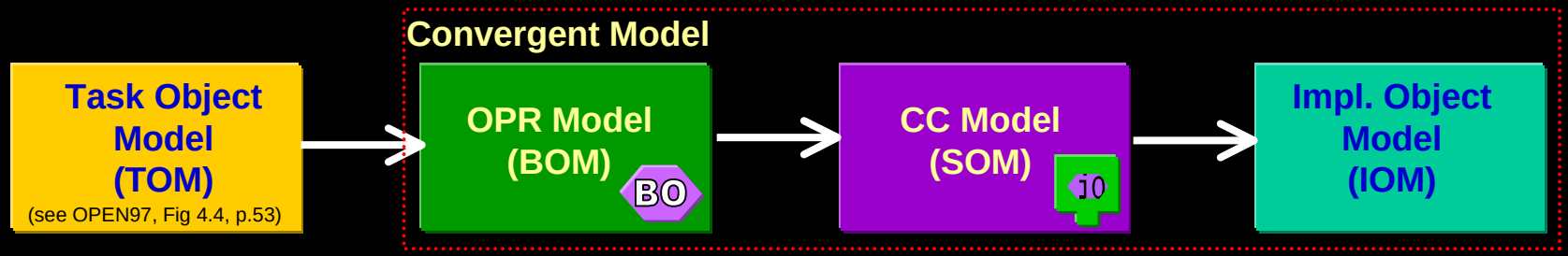
Every iteration is significant and must be planned in some way. However, the required effort for each step is project and context specific.

- **The Plan step: This 4-pass iteration planning step is where QA begins. OPEN/RUP advocate pro-active quality via process and architectural integrity, not retroactive/reactive quality activities.**
- **The Work step: tasks occur fractally in the Work step of an Workflow which consists of Activities (their results)**
- **The QT step: The Quality/Test step: Testing and explicit focus on QA in appropriately small increments during the process assures pro-active quality.**
- **The Release step: Synchronizes, cleans-up, evaluates and cleanly closes a planned development increment. This is important for process metrics, clear planning and transitions.**

© iO Software GmbH

Overview of OPEN Core Workflow/Deliverables

Deliverables from architecture driven analysis by design. (OPEN p 36,53)



- Task Scripts
- Use-UI/Test-Cases
- Requirements Text

- CRC-Run Through
- UI Design
- OPR CIRT Model
- CC-SDT Analysis
- Plans

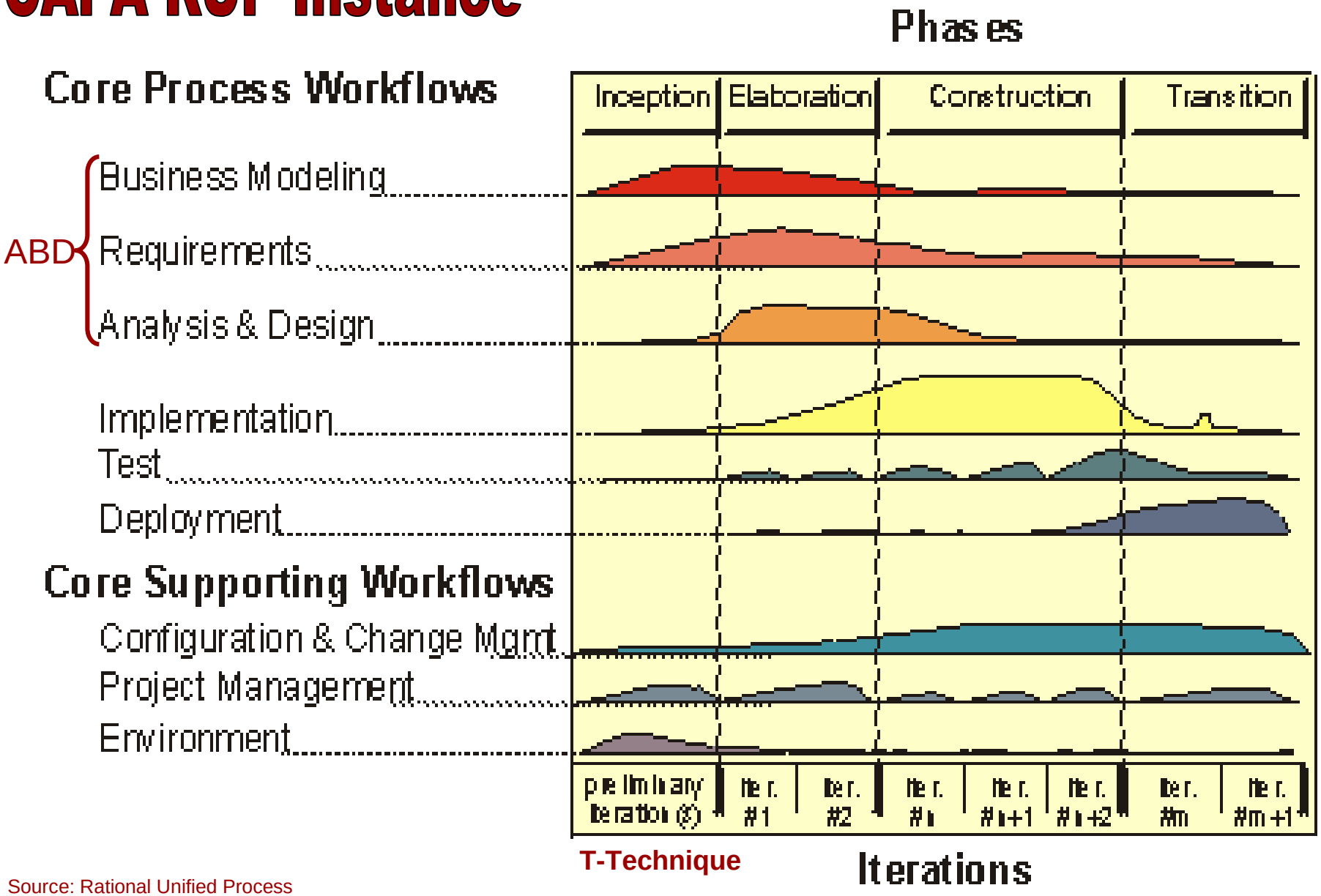
- TProj Analysis
- CCC Mapping
- CCs
- Test Harness
- Deployment Plan
- Operation Plan

- Impl. Projection
- Installation Kit
- Containers

© iO Software GmbH

Interactive Objects

CA: A RUP Instance



Source: Rational Unified Process

© iO Software GmbH

Content

- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

Formal tech. projections to open standards

The current technology projection is to EJB as a subset of CORBA Components

- BO, Organization, Process, Resource, Utility Projections
- Coverage of Critical Design Themes:
  - IT infrastructure vs. business dimension
  - Openness and stability
  - Scalability
  - Managing persistent components: create, lookup, remove; navigation along associations; inheritance and polymorphism
  - Schema mapping for RDBMS
  - Support for distributed transactions
  - Runtime administration
  - Distributed security
  - Nomadic systems
  - Evolution of components

© iO Software GmbH

What is an application server?

- **Anything marketing wants it to be!**
- **Good ones support tightly integrated:**
  - Distributed component development and deployment model with
  - Transactional resource coordination
  - Configurable persistence mechanisms
  - Scalability through distribution across machines
  - Visual monitoring
  - Client access via standard protocols IIOP, RMI, HTTP.
- **Basically 2 types, both have proprietary aspects:**
  - J2EE/EJB based architectures (Future: CORBA Components)
  - Proprietary CORBA or Java based architectures

© iO Software GmbH

EJB reality: The good news.

A major step toward completing the object paradigm...

Good news: progress...

- Component specification with usability features in complex area of distributed IT infrastructure:
- Distributed transaction model.
- Persistence models.
- Class object model.
- (Security model).
- Configurable deployment model.
- Clean separation of concerns.
- “Standard” enables beneficial market forces.

© iO Software GmbH

EJB reality: The bad news.

Many IT-Architectural questions are left unanswered...

The bad news: still a long way to go...

- No formal metamodel - Compare CORBA Components.
- Non-specific on: Associations, Inheritance, Security (improvements in EJB 2.0)
- Skews the object paradigm in favor of Relational DBs.
- The “Standard” is still a moving target.
- Only covers a portion of the component life cycle.
- Implementations are extremely different:
  - highly proprietary but necessary features not covered by EJB
  - cause extreme vendor coupling in many areas
  - lack support of a full-cycle development process

© iO Software GmbH

What else is missing?

A business oriented, complete architectural style is required for success with these technologies in the Information Age.

- **No direct business alignment. We have a better technical infrastructure. That's all.**
- **We still need:**
  - Visible relationship to our business organizations, processes and resources
    - Components must visibly accommodate the business architecture.
  - A clear architectural style to cover the full cycle development of business-relevant components.

© iO Software GmbH

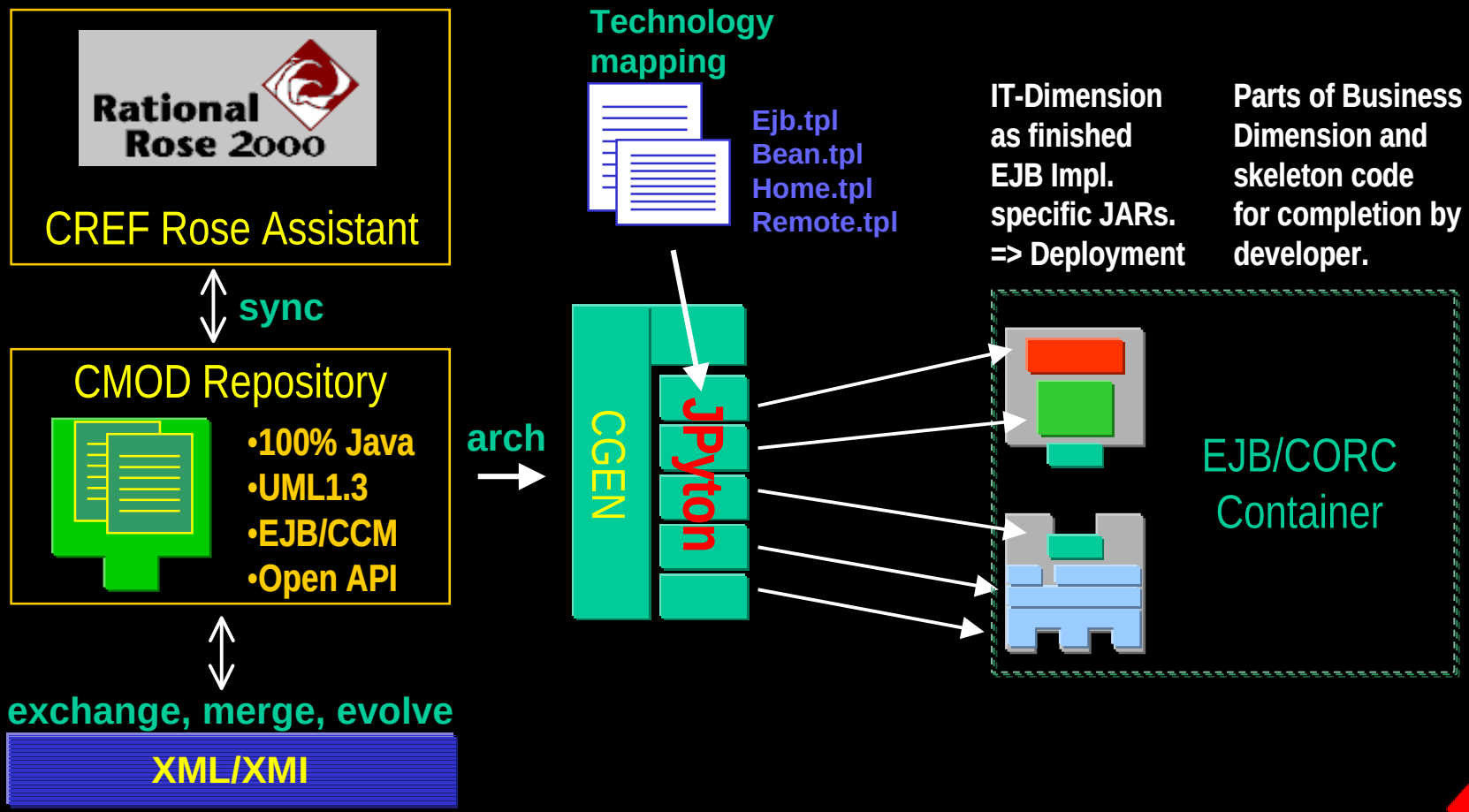
Why the CCs build upon EJB/CCM...

- **Considerable flexibility and return on investment when**
  - An open component model improves on EJB/CCM strengths and mitigates its weaknesses in the context of an architectural style.
  - Architecture driven modeling and code generation builds upon the EJB as a cross-vendor substrate (freedom to change and choose.)
  - Higher level convergent models and tools can be optimized for an infrastructure that eliminates many system-level programming tasks.
    - Even if a lot of it is not standard -> encapsulated in code generation.
  - Developers can focus on the Business Dimension of components and avoid the risk of fast changing technology.

© iO Software GmbH

Generator Assisted Tech. Projection

The IT-Dimension is completely generated based on the CC/UML model.

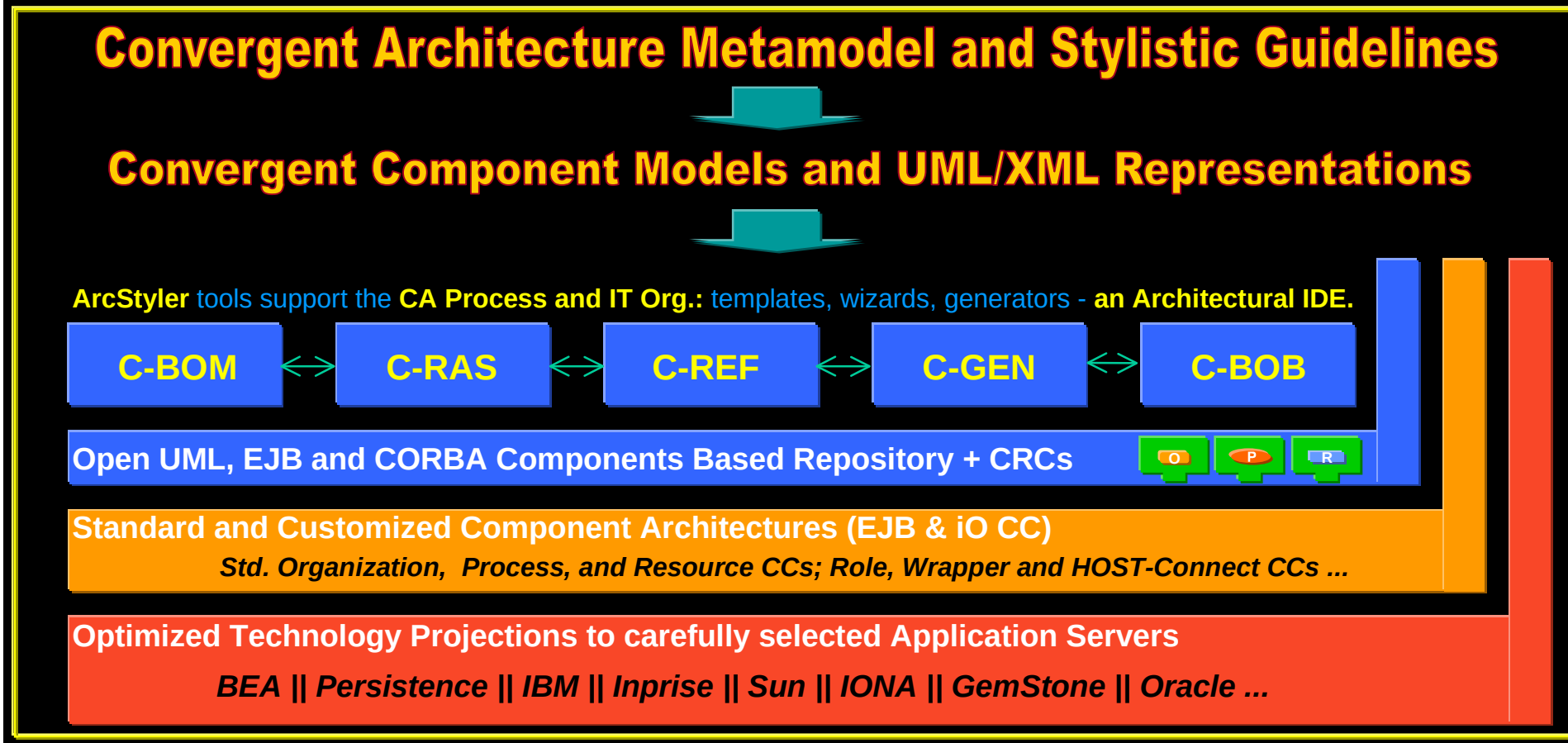


© iO Software GmbH

Interactive Objects

A complete architectural style

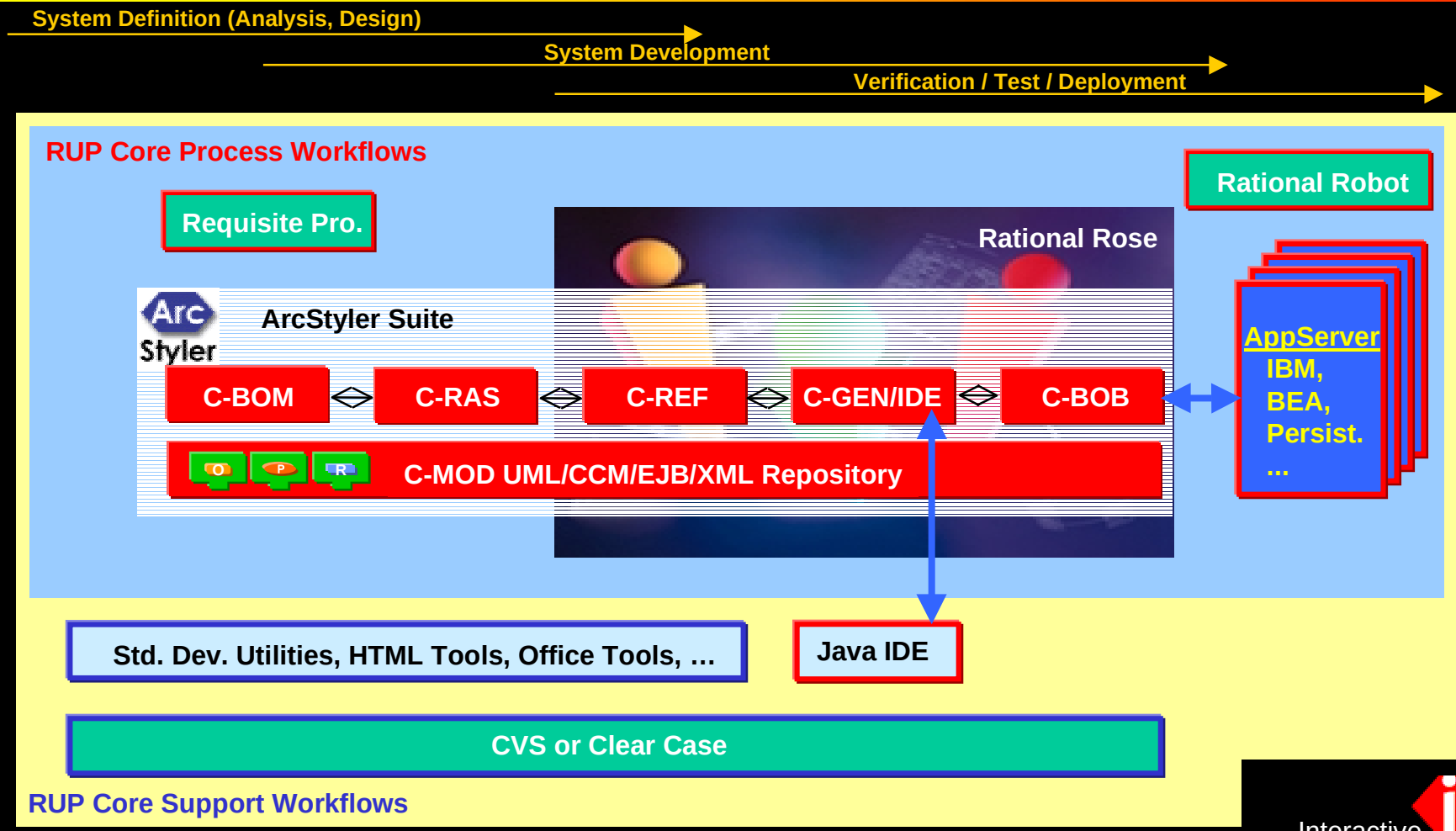
Components are a vehicle of an architectural style. We must start with architecture to define the specific role of components, process, tools...



© iO Software GmbH

ArcStyler Tool Topology

The Project Design is a RUP/OPEN instance. Architecture driven tool integration assures rapid entry, high quality, scalability and re-use…



© iO Software GmbH

Interactive Objects

Content

- **Motivation**
- **Architectural style**
- **Convergent Architecture**
  - Architectural metamodel
  - Component metamodel
  - Development process
  - Projection to open standards
  - tools

© iO Software GmbH

Part 2

Implementation Details and Tools Demonstration



© iO Software GmbH

Architecture Driven EJB/CCM Generation

Standard based cross-EJB-server modeling and automatic generation.



C-BOM->C-RAS: Component architecture assisted creation and refinement of business model.

C-RAS & C-REF: Pattern based, verifier assisted development of EJB/UML component model

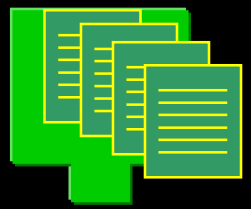
C-GEN: "Smart templates" flexibly handle EJB Impl. Differences

IT Dimension as finished EJB Impl. => HotBuild => HotDeploy

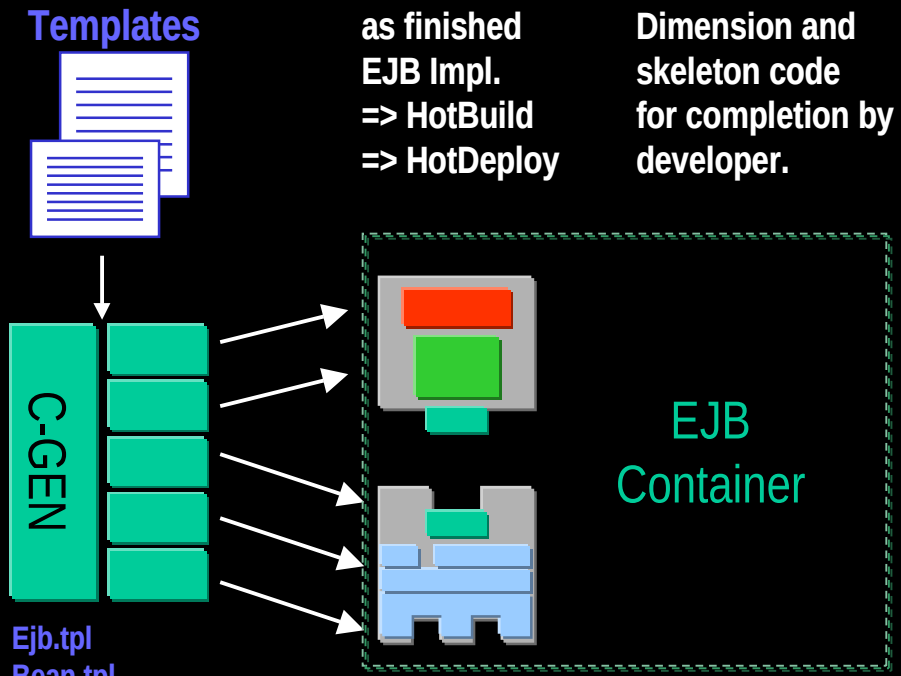
Business Dimension and skeleton code for completion by developer.



Arch. Assisted UML Modeling e.g. Bank.mdl



Standard, product independent UML/EJB/XML model repository



Ejb.tpl
Bean.tpl
Home.tpl
Remote.tpl

Bank CC
Account CC...

© iO Software GmbH

Interactive Objects

ArcStyler - Tool Guides

Overview

- **C-CMD: The ArcStyler Commander**
- **C-BOM: Requirements and Business Modeling**
- **C-RAS: Assisted Refinement**
- **C-REF-ROSE: Enriching the Model**
- **C-GEN: Generating CCM/EJB Code**
- **C-GEN-IDE: Developing Technology Projections**
- **C-BOB: Using the generated GUI Applets, Programming model**

© iO Software GmbH

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- **C-BOM: Requirements and Business Modeling**
- C-RAS: Assisted Refinement
- C-REF-ROSE: Enriching the Model
- C-GEN: Generating CCM/EJB Code
- C-GEN-IDE: Developing Technology Projections
- C-BOB: Using the generated GUI Applets, Programming model

© iO Software GmbH

The Role of the C-BOM Tool

- Formalization of system's requirements and structure
- Support analysis by design via Walk-Through techniques for Use-Case Scenarios
- Generation of project documentation

© iO Software GmbH

Main tasks of business object modeling

- **Analysis by design means iterating over the following tasks:**
  - CRC cards based modeling of business objects and their responsibilities
  - Use-Case scenarios based modeling of collaboration between business objects
  - Walk-Through based model verification and debugging

© iO Software GmbH

Main tasks of business object modeling

CRC cards based modeling

- **CRC cards represent business objects**
- **Types of CRC cards are:**
  -  **Organization** (e.g. a person, a bank)
  -  **Process** (e.g. a transfer)
  -  **Resource** (e.g. an account)
  -  **Accessor** (e.g. graphical user interface)
  -  **Utility component** (to model technical components, e.g. a transaction)

© iO Software GmbH

Each object is described on a specially formatted index card with slots for contract (responsibilities), and the three principle OO relationships. Start by selecting a self-evident business objects from a typical scenario and defining their responsibilities....

CLASS Base_Dev_Proc	SUPERCLASS CA_Process
RESPONSIBILITIES	COLLABORATIONS
K,H Prerequisites/Precond	
D,H Check Preconditions	
K,V Current State	
E,H Process state pattern	
K,V Deliverables/Postcond	
	Components
SDTs:	

© iO Software GmbH

Walk-throughs and run-throughs use role playing. They serve to verify and "debug" the design as well as to achieve consensus with domain experts regarding requirements and priorities.



Investigate and invent

Verify design and record dynamics.

"Run" the design

© iO Software GmbH

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- C-BOM: Requirements and Business Modeling
- **C-RAS: Assisted Refinement**
- C-REF-ROSE: Enriching the Model
- C-GEN: Generating CCM/EJB Code
- C-GEN-IDE: Developing Technology Projections
- C-BOB: Using the generated GUI Applets, Programming model

© iO Software GmbH

The Role of the C-RAS Tool

Convergent Refinement of the Business Model into UML Components.

- **The transfer of a business model into a UML model is traditionally a costly, high-risk translation step.**
- **C-RAS graphically assists the designer in creating the UML component model from the C-BOM design results.**
- **Pattern-based refinement as defined by the OPEN Consortium and the OMG.**

© iO Software GmbH

C-RAS: Project Activities and Tasks

- **Partitioning of the business model responsibilities into UML**
  - interfaces, operations, attributes
  - conditions
  - associations
- **Refining constraints in the form of**
  - preconditions
  - postconditions
  - invariants.

© iO Software GmbH

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- C-BOM: Requirements and Business Modeling
- C-RAS: Assisted Refinement
- **C-REF-ROSE: Enriching the Model**
- C-GEN: Generating CCM/EJB Code
- C-GEN-IDE: Developing Technology Projections
- C-BOB: Using the generated GUI Applets, Programming model

© iO Software GmbH

Motivation (I)

Added value to Rational Rose

- **C-REF-ROSE adds value to Rational Rose by providing ...**
  - a clear style via component metamodels
  - guidance via architectural driven modeling assistants
  - a better quality by configurable design verifiers

© iO Software GmbH

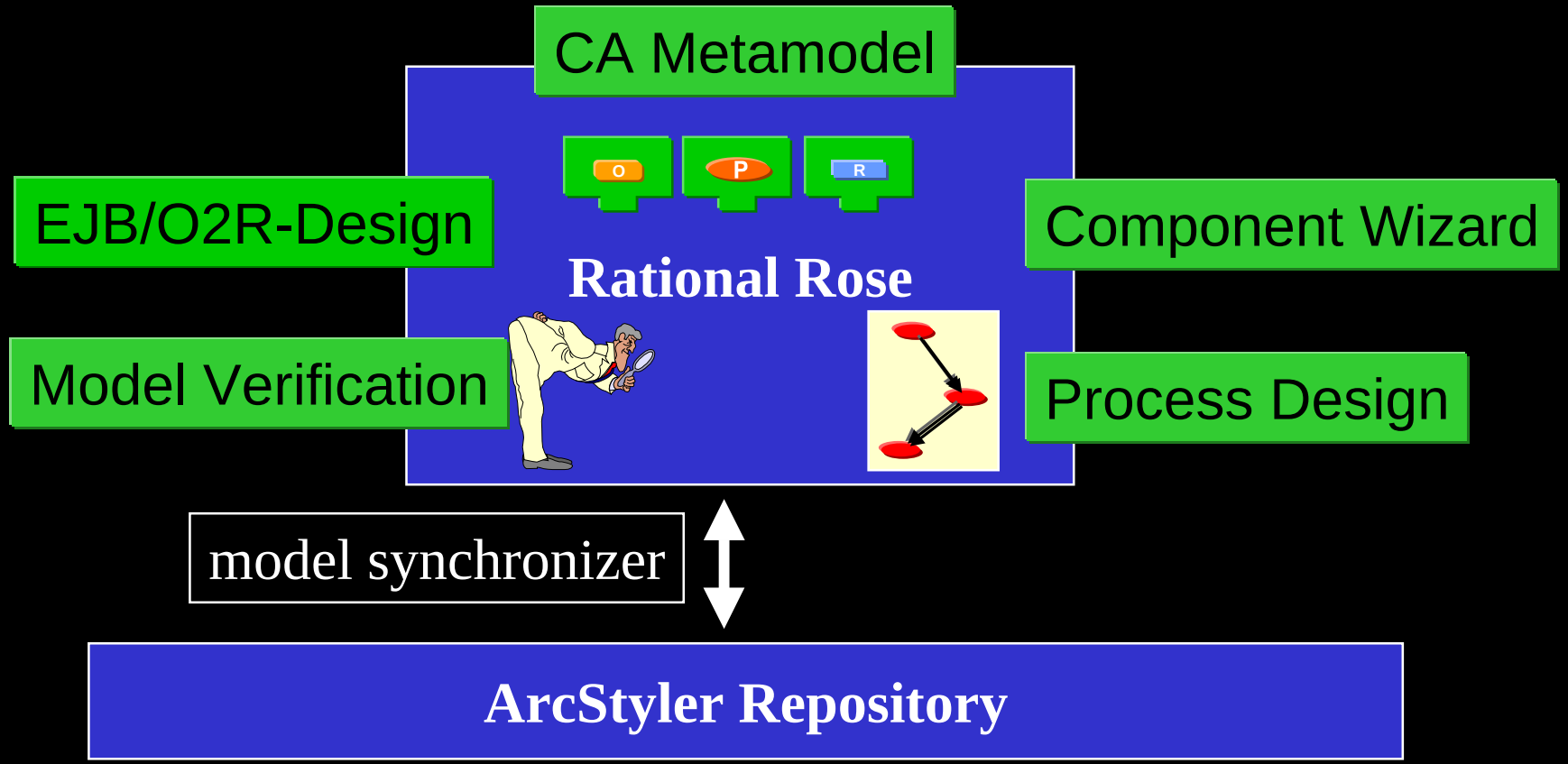
Motivation (II)

The Role of C-REF-ROSE in the Convergent Architecture

- **Use C-REF-ROSE to refine your business model. You can ...**
  - detail your UML entities from C-RAS towards EJB- or CORBA components
  - add new classes, interfaces, signals, methods, attributes, exceptions, ...
  - configure and verify extensions for a specific EJB/CCM implementation
- **After finishing the refinement you generate a projection onto a specific EJB App. Server.**

© iO Software GmbH

Rational Rose Integration



© iO Software GmbH

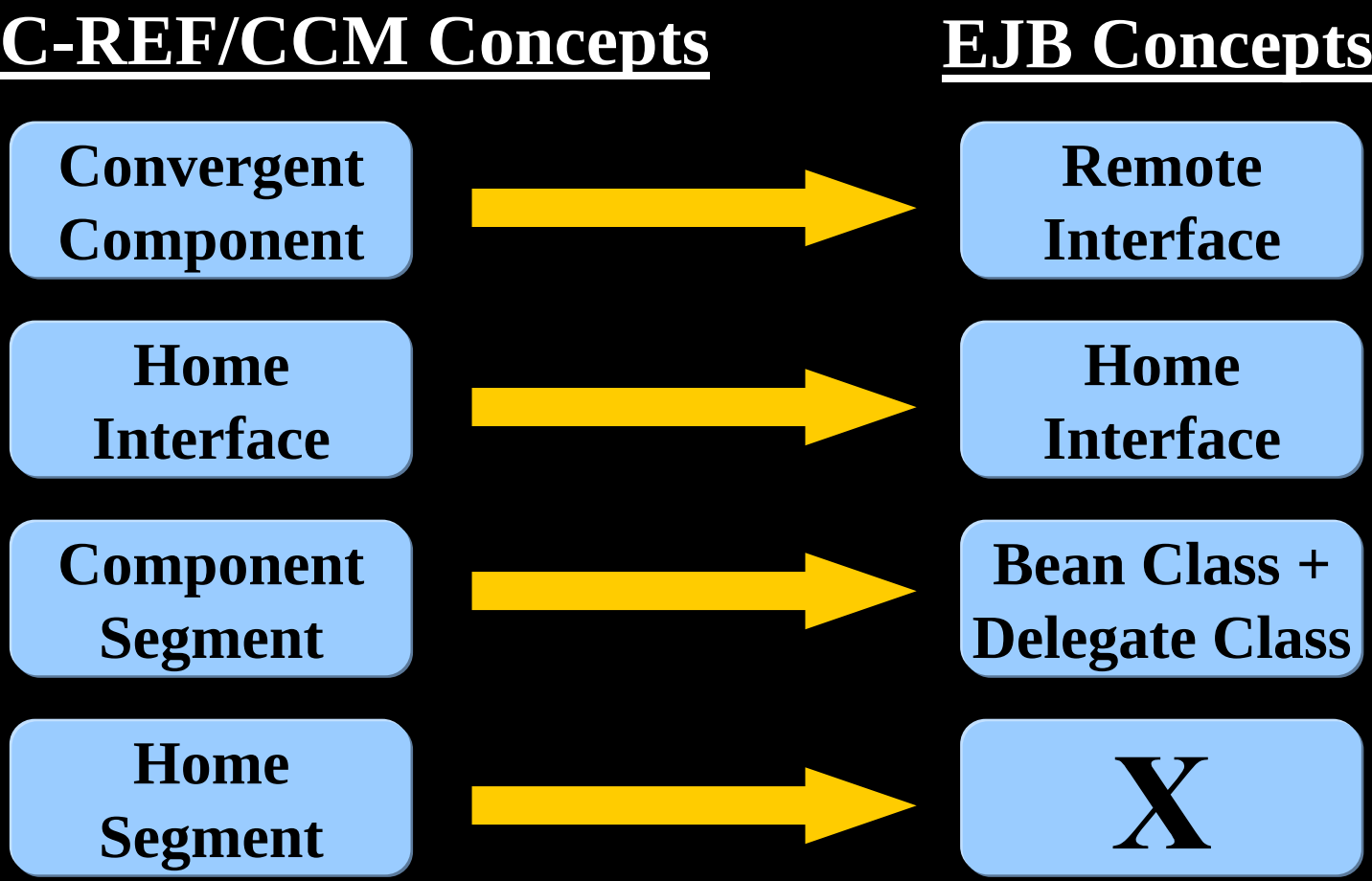
Completing the Components

- **EJB and CORBA components require several IT-Dimension classes**
  - e.g. a home interface for each component
- **C-REF automatically adds the following entities for a Convergent Component:**
  - HomeInterface
  - HomeSegment
  - ComponentSegment

© iO Software GmbH

The Mapping to EJB Concepts

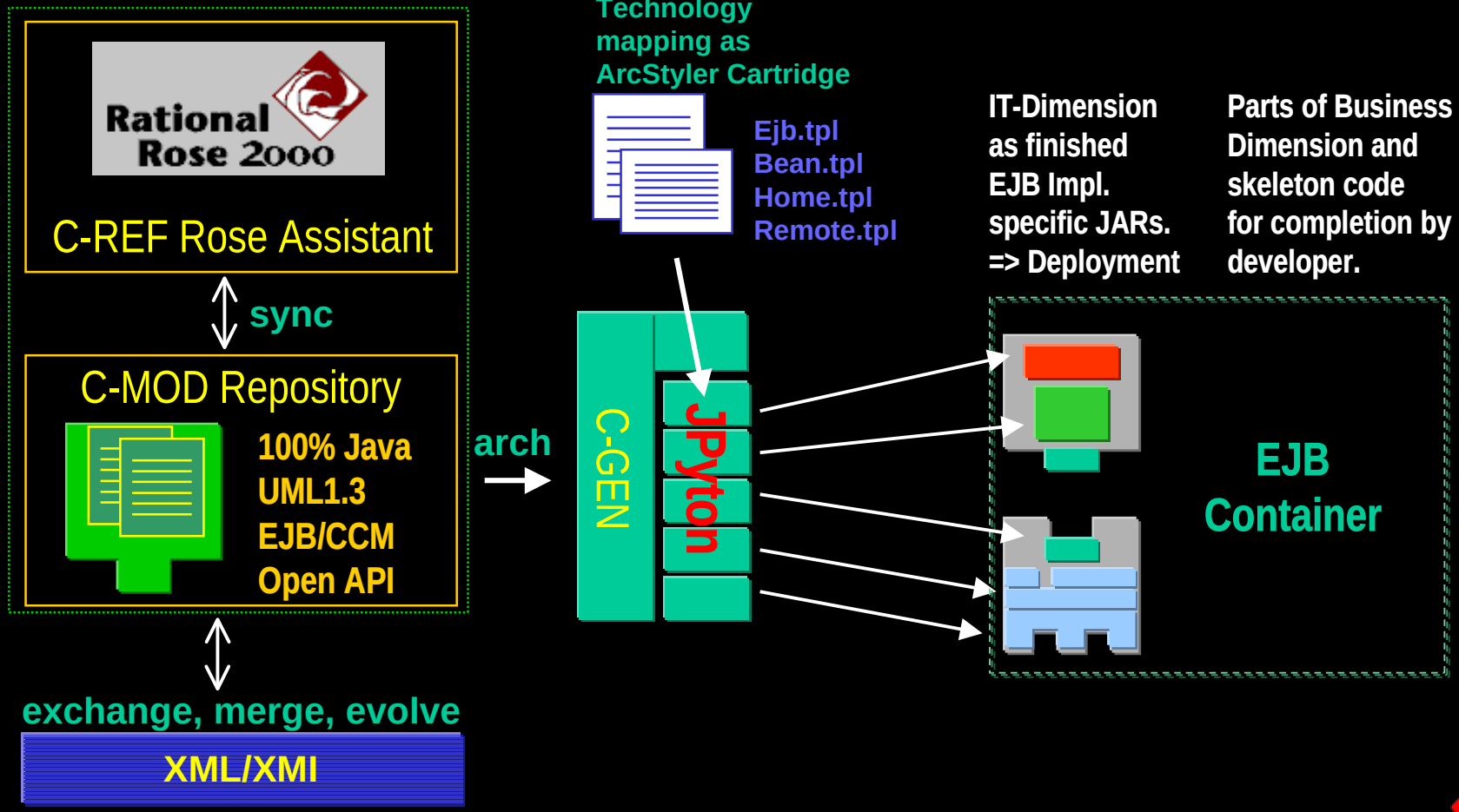
C-REF leverages the CORBA Component Metamodel, a superset of EJB.



© iO Software GmbH

Interactive Objects

C-GEN: Automation of Tech. Projection



© iO Software GmbH

Interactive Objects

Adding Attributes and Operations to the Components

- You can add *attributes* and *operations* to the CCs and the generated entities
- This results in appropriate declarations and implementations (if possible) in the generated code
- In order to allow a mapping to EJB- and CCM-Components, you have to follow some rules



skip details

© iO Software GmbH

Interactive Objects

Adding Attributes to a Convergent Component

- You can add *arbitrary attributes* to a CC
- Attributes result in appropriate *getter- and setter-declarations* in the EJB Remote Interface
- Furthermore, attributes result in appropriate *attribute declarations and getter- and setter-implementations* in the EJB Bean class

© iO Software GmbH

Adding Operations to a Convergent Component

- You can add *arbitrary operations* to a CC
- Operations result in appropriate *method-declarations* in the EJB Remote interface
- Furthermore, operations result in appropriate *method headers and empty bodies* in the EJB Bean class
- You must *add the business logic* after code generation (into so-called “protected areas”)

© iO Software GmbH

Adding create-Operations to a HomeInterface

- You can add *operations with the name "create"* to a HomeInterface
- A create-operation results in an appropriate *declaration* in the EJB Home interface
- An operation create(...) results in a *method header ejbCreate()* in the EJB Bean class
- You must add the semantics after code generation

© iO Software GmbH

Adding find-Operations to a HomeInterface

- You can add *operations with a name starting with "find"* to a HomeInterface
- A find-operation results in an appropriate *declaration* in the EJB Home interface
- You must add the semantics after code generation

© iO Software GmbH

Adding Attributes and Operations to a ComponentSegment

- **You can add arbitrary attributes and operations to a ComponentSegment**
- **At the moment, no code will be generated from the ComponentSegment information**

© iO Software GmbH

Extending the model

Adding Components

- Add new Components via the tool palette, or
- Add a class and choose one of the stereotypes <Organization>, <Process>, <Resource>, <Utility>, <Component>
- Those components are treated as the ones coming from C-RAS

© iO Software GmbH

Extending the model

Adding Classes and Interfaces

- You can add arbitrary classes and interfaces to the model
- You can add attributes and operations to those classes and interfaces
- During generation, declarations with protected areas will be generated for those classes and interfaces

© iO Software GmbH

Configuring the Components

Standard and container specific properties

- **You can configure both ...**
  - standard conform properties for such features like transactions, security, and persistence, and
  - properties for features that are container specific, e.g., schema mapping, caching etc.
- **Configuration is done via ROSE property specification**

© iO Software GmbH

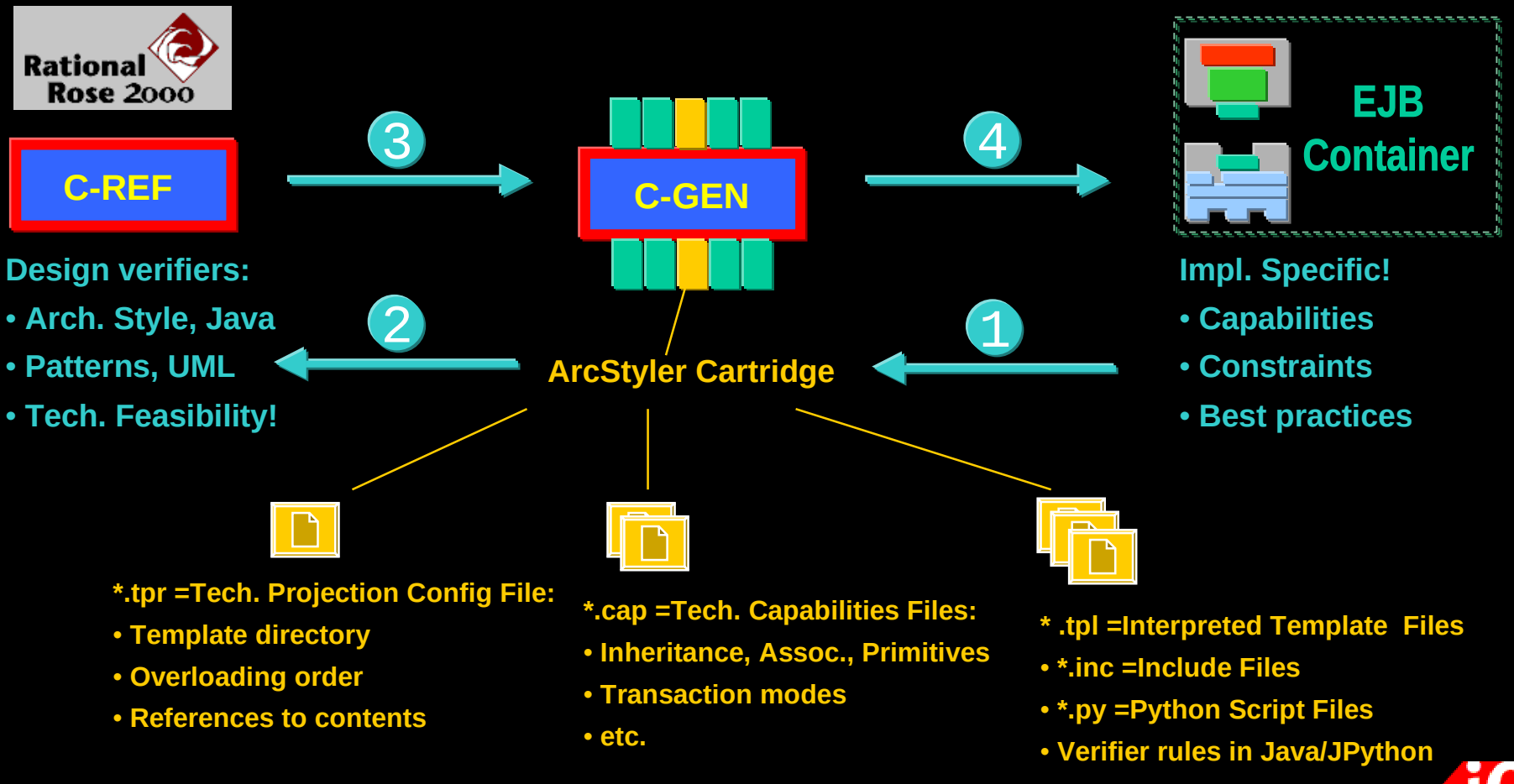
Validating the Model

- **You can automatically verify the model with respect to both**
  - architectural style, and
  - the selected technology projection
- **The validation rules are configured via the selected cartridge and can be extended via the ArcStyler Extensibility Service**

© iO Software GmbH

Example: Tech. projection design and verifiers

The technology projection dynamically propagates technical constraints back to UML/Java/EJB/Rose verifiers in C-REF/Rose.



© iO Software GmbH

See annotated *tpr and *.cap files in examples directory.

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- C-BOM: Requirements and Business Modeling
- C-RAS: Assisted Refinement
- C-REF-ROSE: Enriching the Model
- **C-GEN: Generating CCM/EJB Code**
- C-GEN-IDE: Developing Technology Projections
- C-BOB: Using the generated GUI Applets, Programming model

© iO Software GmbH

The Role of the C-GEN tool

- Generation of source code from your model
- Generation of deployment descriptors
- Generation of support files (JBuilder project, SQL scripts, ...)
- Optimized build process to partially rebuild sources
- Deployment of components to application server

© iO Software GmbH

Using the Build Menu

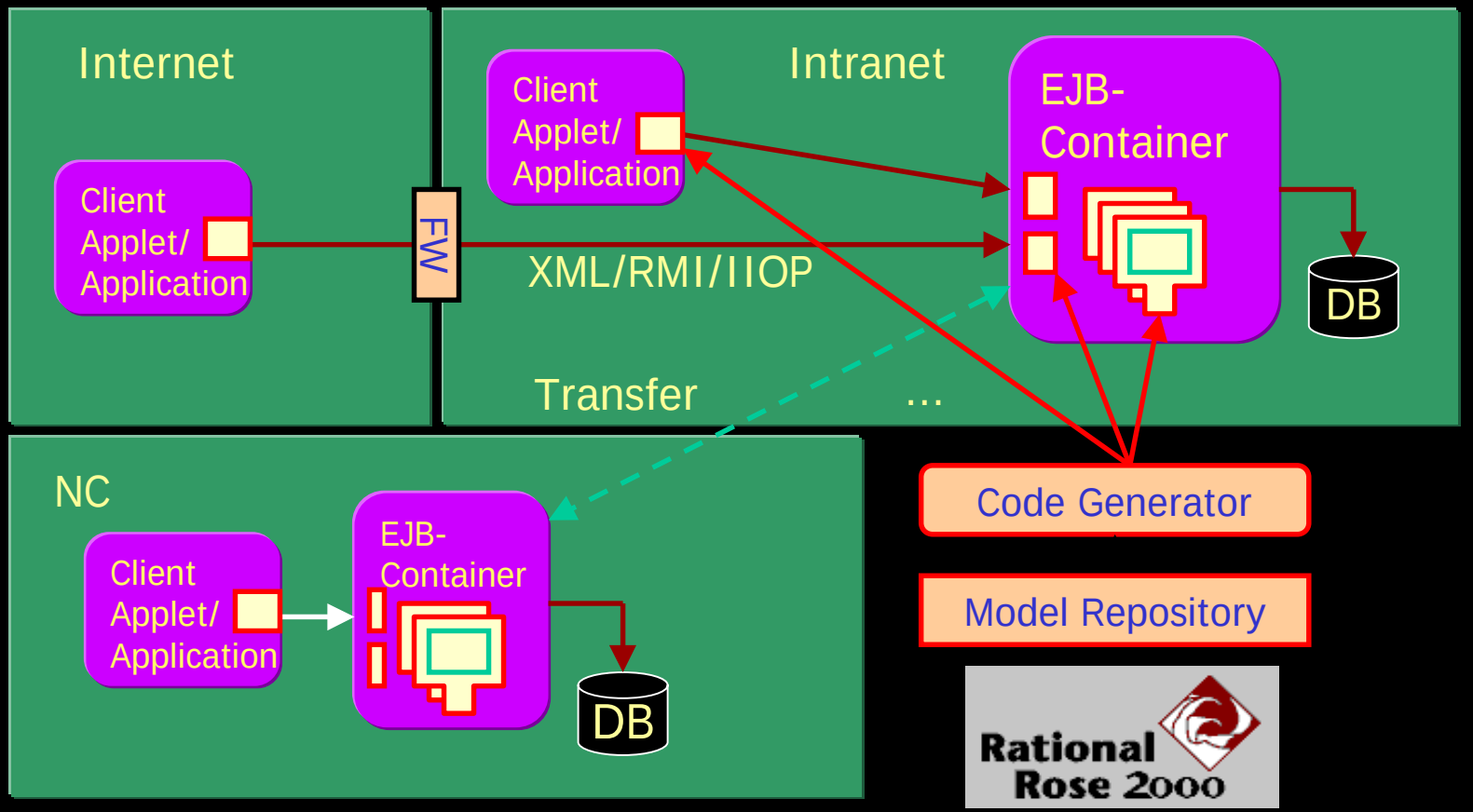
- **From the Build menu you can choose**
  - **Generate Source code** to generate sources for the packages or components of the model that are highlighted in the browser window
  - **Build components** to build or rebuild the sources that have changed since the last build process
  - **Hot deploy** to build helperclasses (stubs, skeletons, ...) and install the components on the application server

© iO Software GmbH

Current EJB-Container projection generates...

Generated vs. programmed elements.

C-BOM C-REF C-ACE C-GEN C-BOB



Legend:
- Protected Business Dimension
- Generated IT Dimension infrastructure

© iO Software GmbH

Interactive Objects

Protected code areas

- **Protected areas in the generated code preserve your manual changes over development iterations:**

```
public void makeWithdrawal(double amount) throws RemoteException
  {
    /* START OF PROTECTED AREA  <<method:makeWithdrawal:double>> */
    // insert custom implementation of method makeWithdrawal here
    m_balance = m_balance - amount;
    /* END OF PROTECTED AREA 6e888ca300000047 */
  }
```

© iO Software GmbH

Using and Extending the Testclients

Overview

- The ArcStyler generates java client code for accessing the generated and deployed components
- There are protected areas in the testclients for adopting the code to your needs
- The ArcStyler generates batch-scripts for starting the testclients

© iO Software GmbH

Projection to partner EJB-Containers

Our experience with Technology Projections to various EJB-Containers in projects and in our ArcStyler development.

- Persistence PowerTier
- BEA WebLogic/TopLink
- IBM WebSphere
- Inprise IAS
- GemStone GemStone/J
- ProSyst EBS
- SUN/Forte SynerJ
- Others: IONA, Silverstream etc.

© iO Software GmbH

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- C-BOM: Requirements and Business Modeling
- C-RAS: Assisted Refinement
- C-REF-ROSE: Enriching the Model
- C-GEN: Generating CCM/EJB Code
- C-GEN-IDE: Developing Technology Projections
- C-BOB: Using the generated GUI Applets, Programming model

© iO Software GmbH

Interactive Objects

The Role of the C-GEN-IDE

- **A Technology Projection is a collection of JPython based templates. The C-GEN-IDE is an Integrated Development Environment for**
  - adjustment of existing templates according to your needs,
  - development of a new cartridge (technology projection) for a special application server,
  - testing of new templates.

© iO Software GmbH

Features of the C-GEN-IDE

- **Core of the C-GEN-IDE is a debugger that visualizes code generation from the templates. The debugger can:**
  - run to specific lines in the template
  - step over or into / out of functions
  - inspect values of variable and expressions
- **In case of an error during code generation the debugger starts automatically, loads the active template, and jumps to the error's position.**

© iO Software GmbH

C-GEN: Design principles

Convergent Architecture driven translative generation.



- **Translative code generation is the most open, modern form of code generation.**
  - **Ref: S. Mellor, "Automatic Code Generation from UML Models", C++ Report, April 1999.**
- **Clean separation of concerns:**
  - 1. Input for architectural driven design model (CORC based UML in XML)
  - 2. Input for technology projection (templates)

© iO Software GmbH

C-GEN-IDE

The graphical Integrated Development Environment for „Meta-level development“ and extension of Cartridges.

- What does the C-GEN-IDE do:
  - **Code-sensitive highlighted editing**
  - **Run to break point, step in, over, out of, pause...**
  - **Trace/monitor view extensible via script expression call-outs.**
  - **Several output levels: info, script, source code level.**
  - **On-demand, dynamic load of Cartridge artifacts**
  - **Configuration of Cartridge and options.**
  - **Accessible directly from Rose and other tools.**

© iO Software GmbH

ArcStyler - Tool Guides

Overview

- C-CMD: The ArcStyler Commander
- C-BOM: Requirements and Business Modeling
- C-RAS: Assisted Refinement
- C-REF-ROSE: Enriching the Model
- C-GEN: Generating CCM/EJB Code
- C-GEN-IDE: Developing Technology Projections
- **C-BOB: Using the generated GUI Applets, Programming model**

© iO Software GmbH

The Role of the C-BOB Tool

- Provide easy access to automatically generated default clients for your components
- Provide a starting point to develop your own client applications
- Provide a framework to easily test your own clients

© iO Software GmbH

C-BOB Features (1)

Support for Plain EJBs

- **Provides a GUI for starting the generated default clients for Plain EJBs:**
  - select an available default client
  - determine a server connection
  - pass parameters
  - start the default client
  - result window
  - history functionality

© iO Software GmbH

C-BOB Features (2)

Support for Convergent Components

- **Clients are pure XML/Java/Swing**
- **Clients can be used in both an Internet browser or as a standalone Java application**
- **Clients can perform many common e-commerce tasks such as:**
  - monitoring
  - logging life-cycle statistics
  - interacting with a standard Web browser

© iO Software GmbH

Simplify your world!



Celebrating 10 years of production objects with an advance in vendor-neutral IT consolidation: Convergent Architecture

http://www.io-software.com

Interactive Objects

iO Software GmbH · Basler Str. 63 · D-79100 Freiburg · +49 (761) 400 73-0