The Future of Component-Based Enterprise Application Development

Alan W. Brown

Director, Research

Application Development Division

Sterling Software

Alan_Brown@Sterling.com

Agenda

- Enterprise Development Challenges
- Why Does CBD Help?
- What Will Make CBD Successful?
- What is the Current State of CBD?
- What is in the Future for CBD?
- Summary

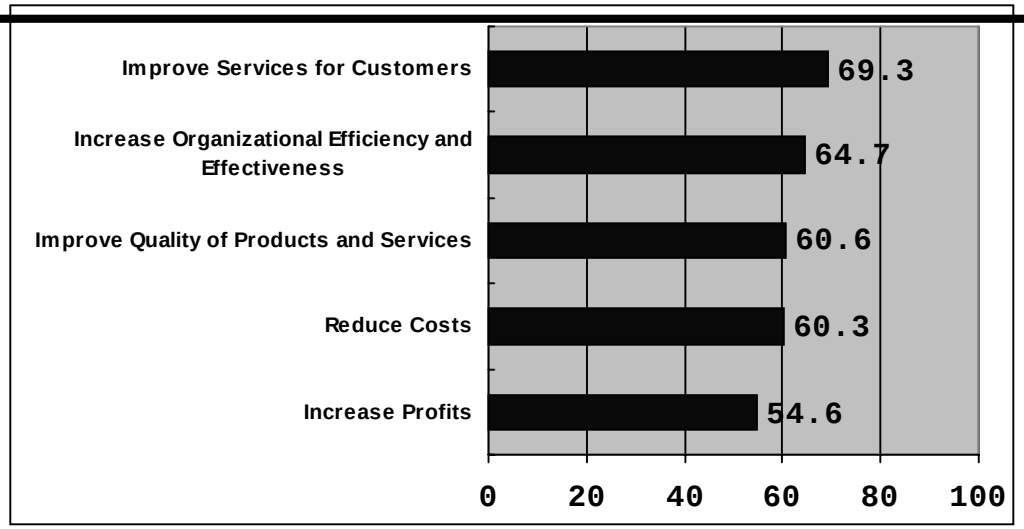
Enterprise Development Challenges

What's Driving EAD?

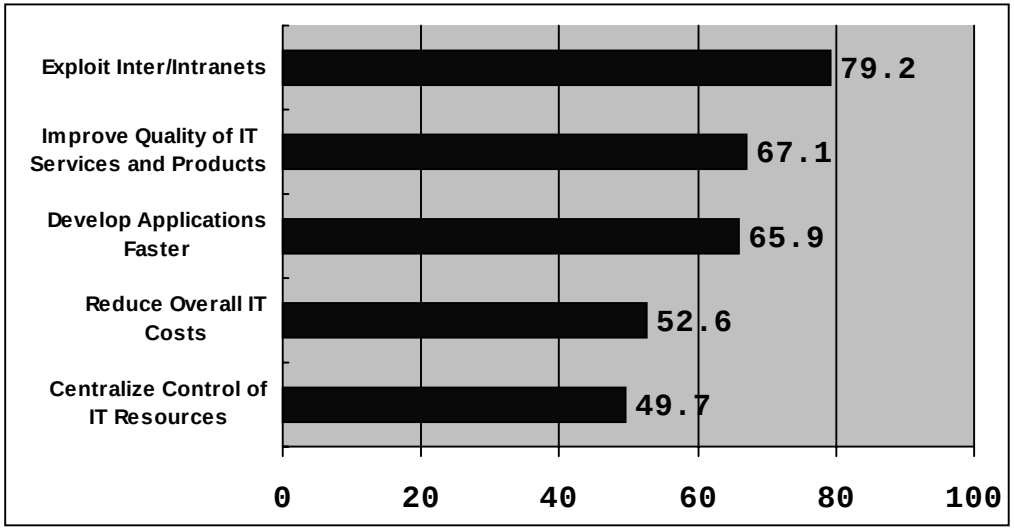
- Evolving customer needs
- Business trends
- Technology trends

Enterprise Development Challenges

Evolving Customer Needs



Top 5 IT Business Drivers
(Source: SPG Analyst Services)



Top 5 IT Initiatives and Mandates
(Source: SPG Analyst Services)

Enterprise Development Challenges

Business Trends

- Outsourcing of IT in whole or in part
- Importance of E-commerce
- Deregulation of key industries
- Mergers and acquisitions
- Impact of Y2K and EMU

Enterprise Development Challenges

Technology Trends

- Pervasive use of Internet/Intranet
  - July 98 estimate: 36,739,000 Internet hosts
  - Barron's projected "business-to-business" trade on the net to be $22B annually by the year 2000
- Distributed systems
  - middleware standards
  - middleware products
- Decreasing hardware cost/performance ratio
- Heterogeneous computing infrastructure

Enterprise Development Challenges

- Managing complexity
  - understanding the business domain
  - defining the systems architecture
  - scoping each system's behavior
  - implementing using appropriate technology
- Rapidly effect change
  - supporting new ways of doing business
  - adjusting to organizational re-alignments
  - finding and reusing existing assets
  - taking advantage of technology evolution

Why Does CBD Help?

- Component Based Development is about:
  - An approach to developing software based on assembly of pre-built parts
  - A philosophy of application development emphasizing cooperation among separately developed parts offering services through interfaces
  - Standard ways to describe and document parts to enable a common way to add parts to a system, to query parts to determine their behavior, and to replace one part with another
  - Supporting technology for these standards offering a range of component management services
  - A growing market of reusable pieces, interoperable infrastructure, and trained people

Why Does CBD Help?

CBD Promises

- Basis for incremental growth and evolution of enterprise solutions
- Allows access to new distributed technologies
- Provides the basis for a modeling approach that recognizes the distributed, heterogeneous nature of current applications
- Offers a way for client-focused developers and host-focused developers to work together
- Encourages and support a culture of reuse

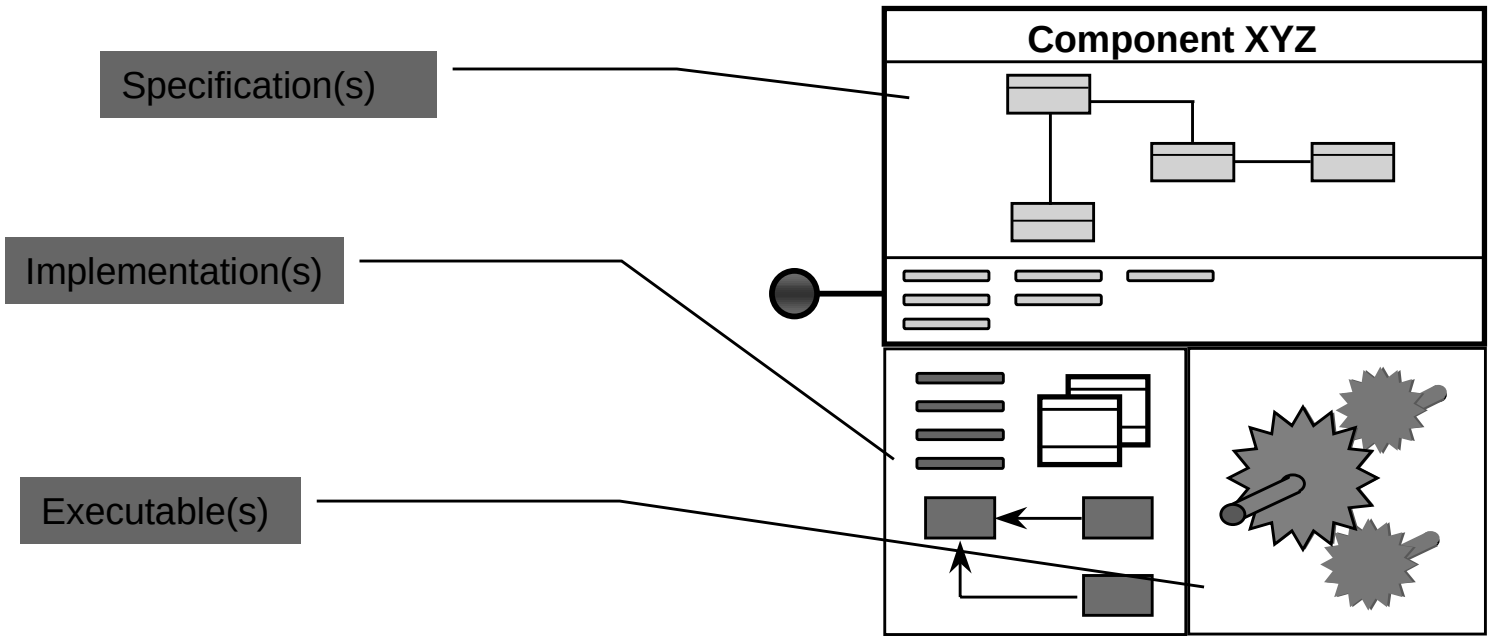
Why Does CBD Help?

Different Component Perspectives

OOP	Visual Programming	OOA/D	Infra-structure	Enterprise Applications
Smalltalk C++ Java Eiffel etc.	Visual Basic Visual Café VisualAge Visual J++ etc.	Rose Select COOL:Jex etc.	CORBA DCOM Java Beans	COOL:Gen SAP ORACLE etc.

Why Does CBD Help?

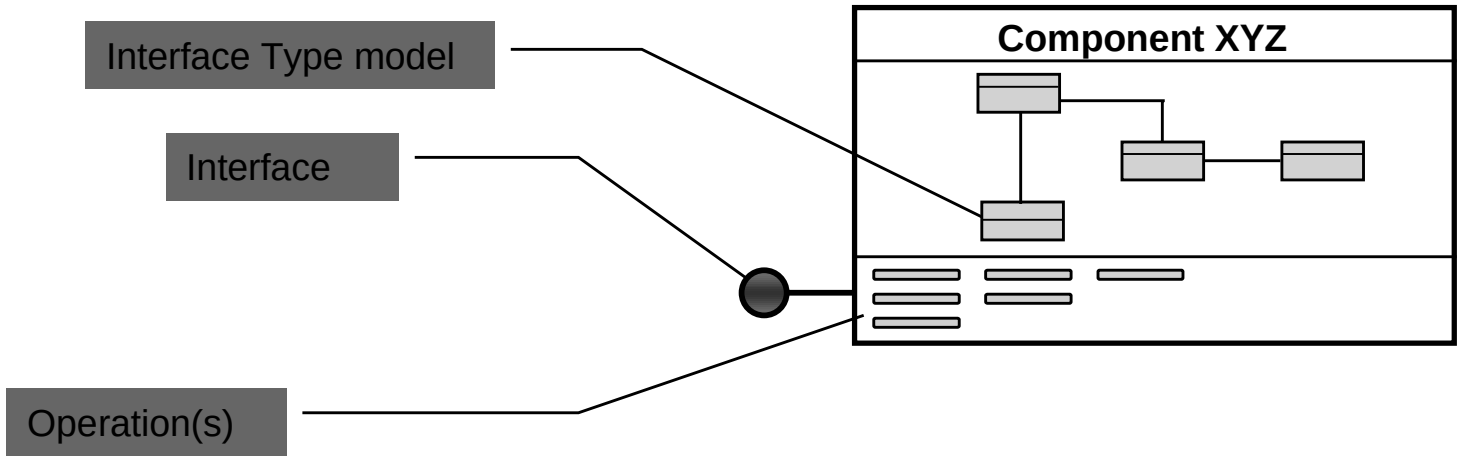
Component facets



- A description of **WHAT** the interface can do
- A description of **HOW** the interface does what it does
- A platform deployable implementation of one or more interfaces

Why Does CBD Help?

Component Specification



- The interface type model defines the “vocabulary” of the interface
- An operation is a service available for consumption by a client of the component
- An interface is a logical grouping of semantically related operations

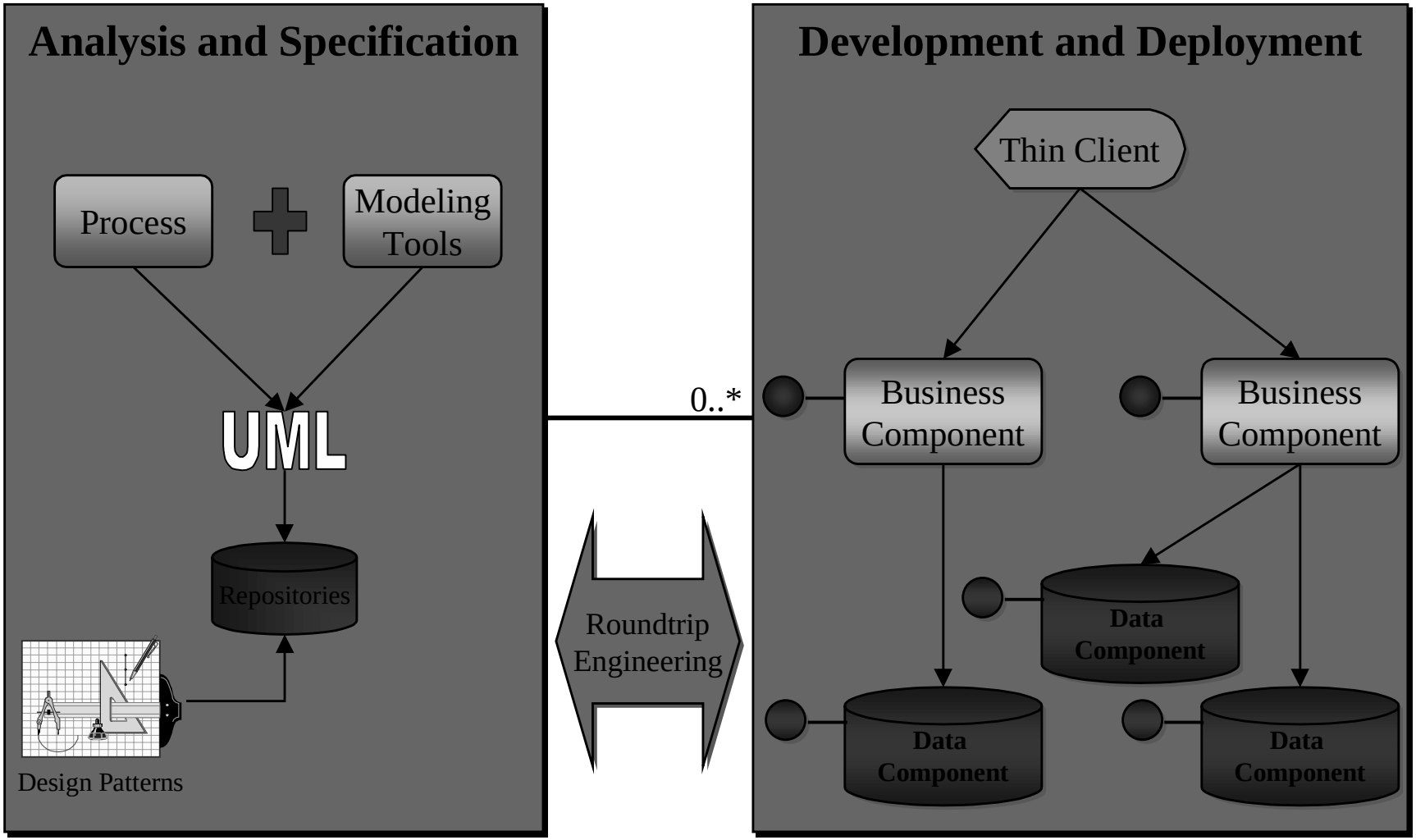
Why Does CBD Help?

Component Characteristics

- Access is only granted via a published interface
- May depend on other components without revealing this fact (via Delegation, Containment, Embedding)
- It understands the contents of its boundaries… it is self-aware
- It communicates state via messages and a consistent exception handling mechanism.
- It's not a granularity thing… large grain, small grain, widget etc… they're all components!

Why Does CBD Help?

Current Component-Based Solutions



Source: Mike Sparling, "Components: Today, Tomorrow, and Beyond…", Casetek, 1998.

Why Does CBD Help?

- Design approach
  - traditional development methods are inappropriate for distributed, reuse-based, multi-thread applications
- Tool support
  - new tool functionality is needed for design, management, and deployment of component-based systems
- Targeted platforms
  - deployed applications must be net-centric, flexible, and scaleable

What Will Make CBD Successful?

5 Key Directions

- We have identified 5 keys to success:

1. *Model-based software engineering*
2. *Generation of technology-specific implementations*
3. *Interface-based design*
4. *Knowledge reuse through patterns*
5. *Flexible, integrated tools across the life-cycle*

What Will Make CBD Successful?

- The attractions of model-based software engineering are more important today that ever!
- Complex systems are understood through
  - abstraction
  - decomposition
  - refinement
- Model-based software engineering directly supports these ideas across the software life-cycle
- This is a major factor in managing the maintenance and evolution of enterprise-scale systems

What Will Make CBD Successful?

- But it requires development organizations to
  - have greater discipline in following repeatable methods
  - make investments in design, recouped during maintenance
  - reuse designs and code more systematically
- These challenges are being addressed through
  - improved method support incorporating the best of IE and OO
  - productivity enhancements through patterns and frameworks
  - reuse-focused methods and tools as we move from development to assembly of applications

What Will Make CBD Successful?

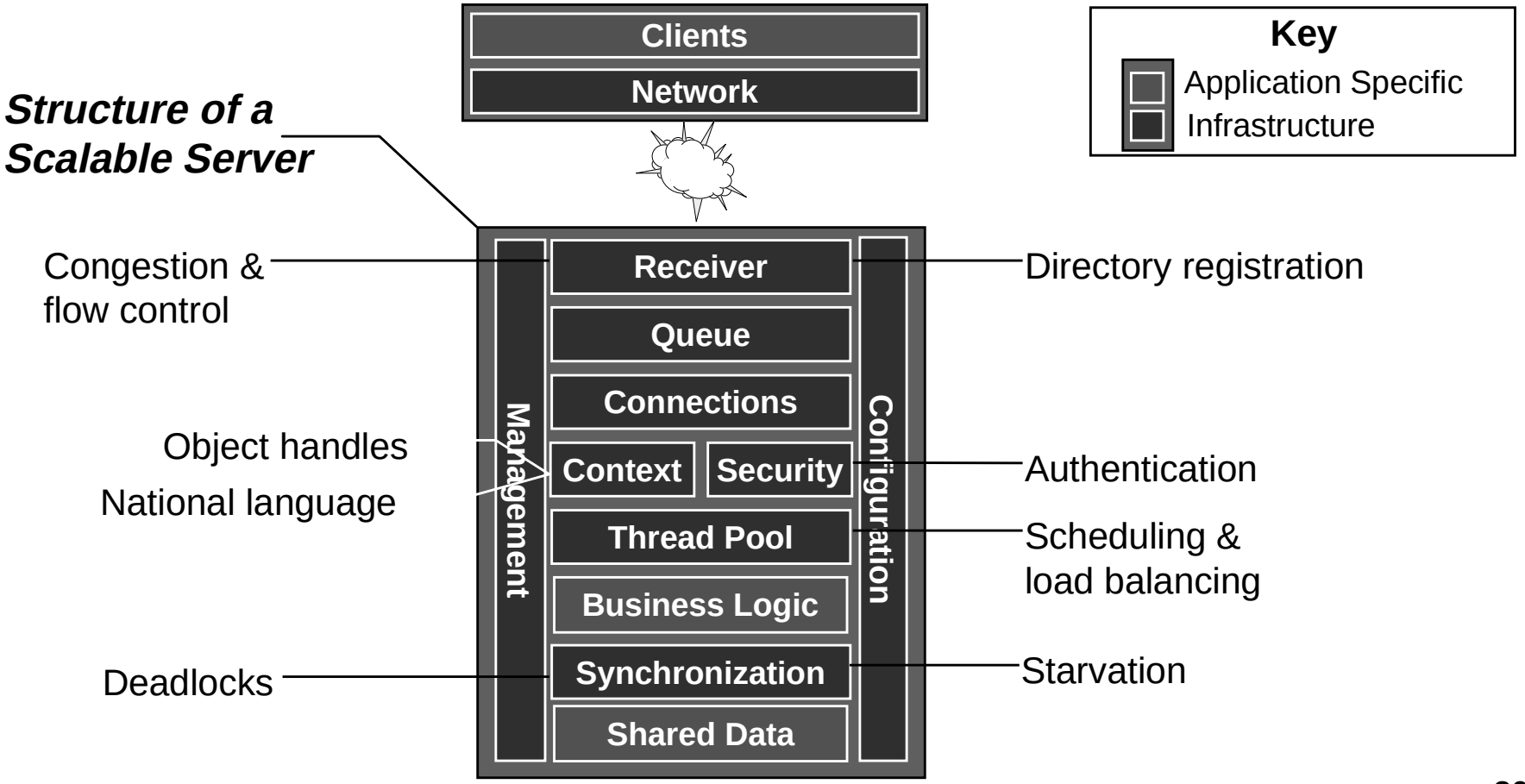
Generation of Implementations

- Developers spend most of their time developing and maintaining the application infrastructure
- This is typically
  - the most difficult part of the application to write
  - requires expensive specialist skills
  - technology-specific and subject to greatest change
- Generating the technology infrastructure aspects of application provides greater stability, quality, and ease of upgrade
- This is most important
  - in times of greatest technological change
  - when computing specialist skills are in high demand

What Will Make CBD Successful?

Application Complexity

- Only a small part of most applications deals with the business logic and shared data



What Will Make CBD Successful?

- Moving to new technologies and methods such as interface-based design is important to many organizations
  - to take advantage of improvements in technologies and techniques
  - to recruit and retain high-technology staff
- Organizations need to take advantage of existing legacy systems while creating more adaptable systems for the future
- A design approach is required based on modeling interactions among service providers

What Will Make CBD Successful?

Interface-Based Design

- Interface-based design concentrates attention on the roles and responsibilities supported within a domain
- Such and approach allows the behavior to be described in terms of interfaces and collaborations
- Components support one or more interfaces
- Implementation of components can be via reuse of existing components and systems, or implementation in tools such as COOL:Gen and COOL:Jex

What Will Make CBD Successful?

Interface-Based Design

Principal Concept

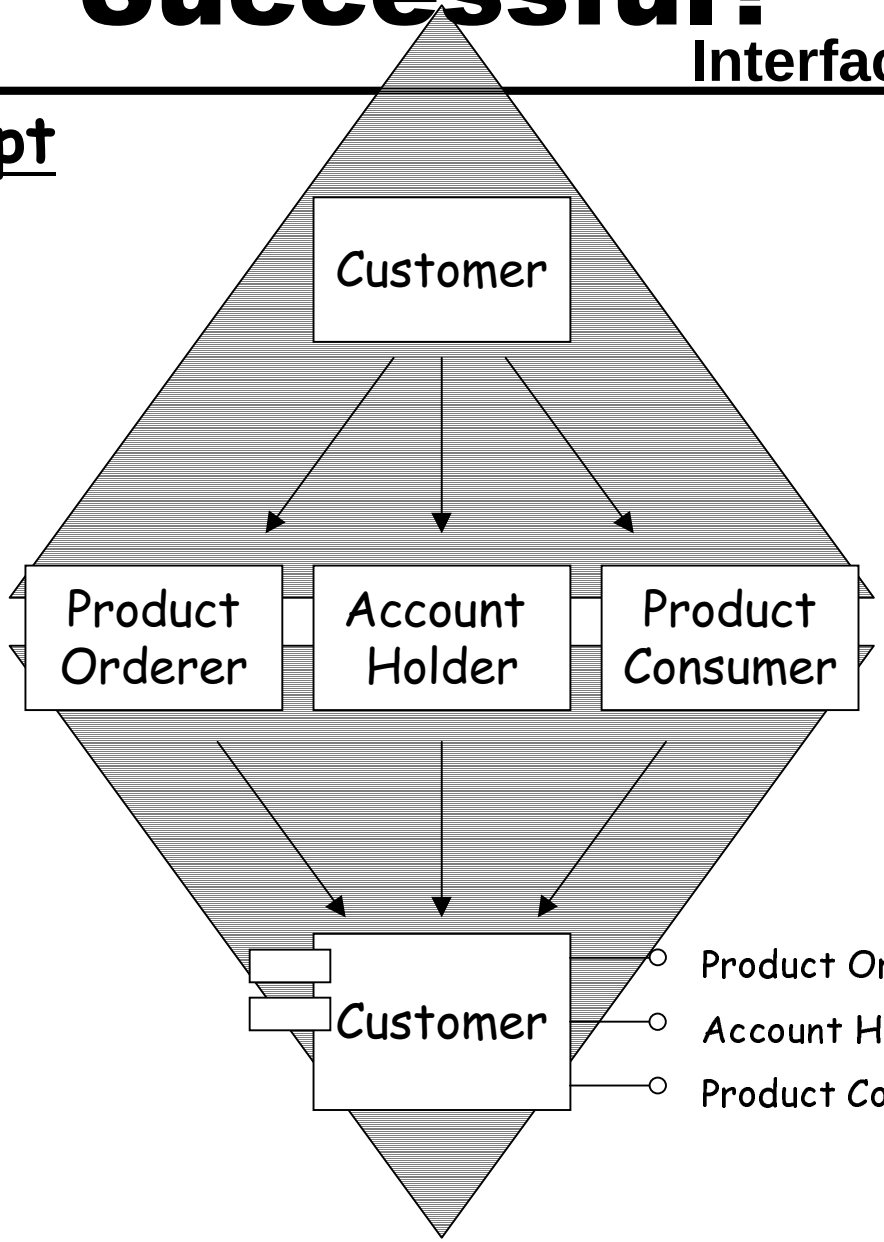
Object

"normalization"

Interface

"de-normalization"

Component



Stage

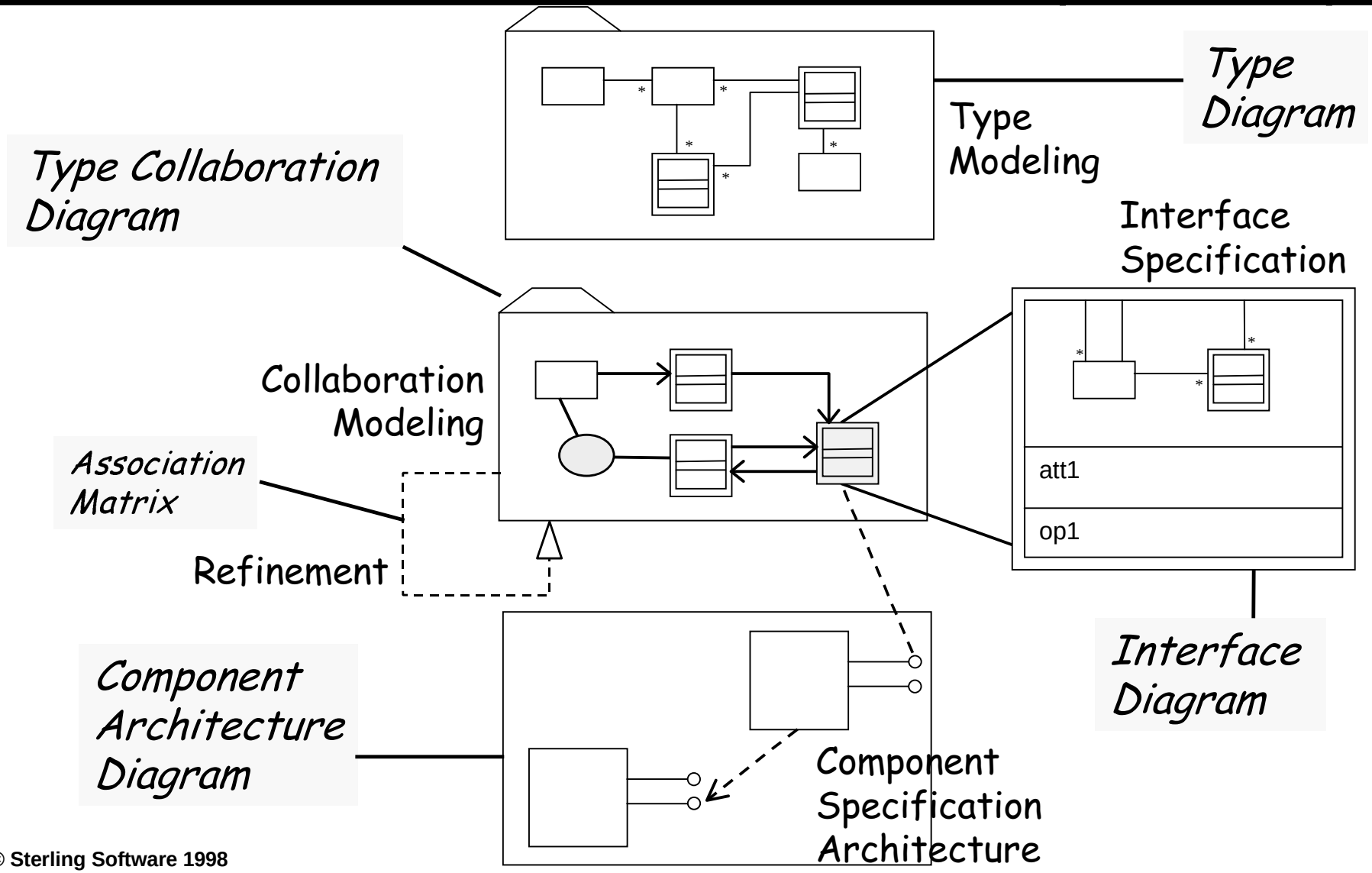
Domain Modeling

Role Modeling

Component Modeling

What Will Make CBD Successful?

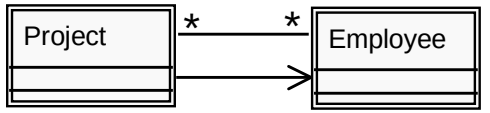
An Example: COOL:Spex

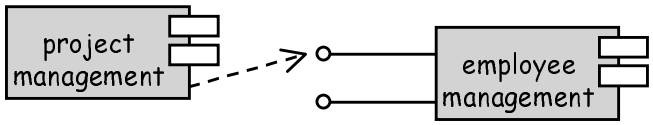


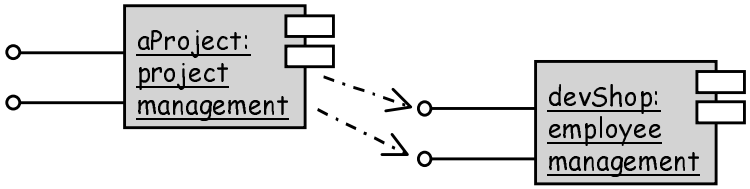
What Will Make CBD Successful?

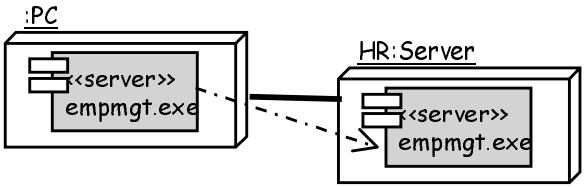
Component Architecture

- Architecture - the parts of a system, and how they are related
- Component Architectures
  - Interface Dependency Models
  - Component Architecture
  - Component Object Architecture
  - Component Deployment Architecture
- Two scopes
  - project component architecture (e.g. one application)
  - enterprise component architecture (for multiple applications)



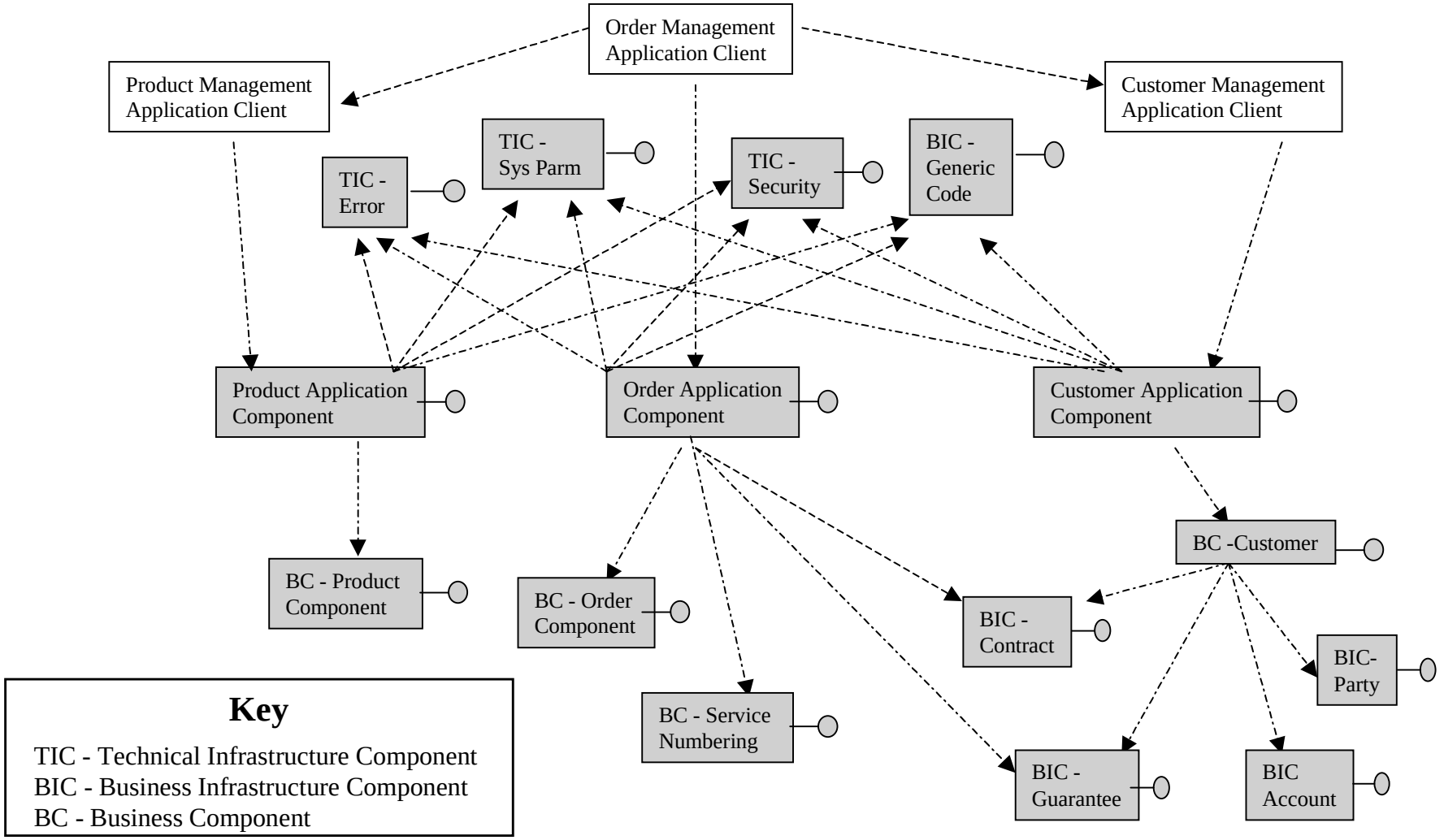






What Will Make CBD Successful?

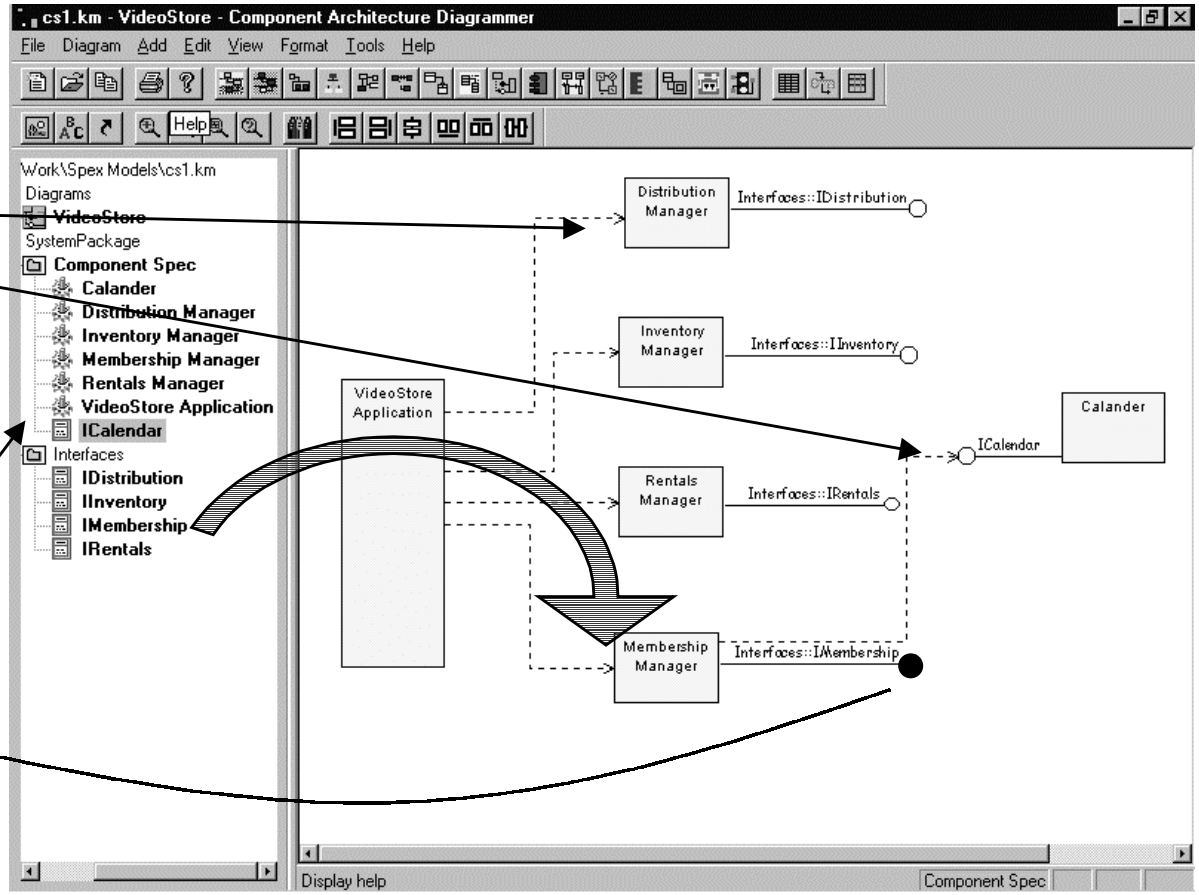
An Example Component Architecture



What Will Make CBD Successful?

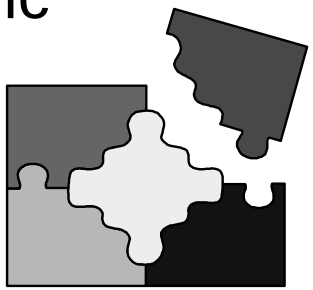
Component Modeling in COOL:Spex

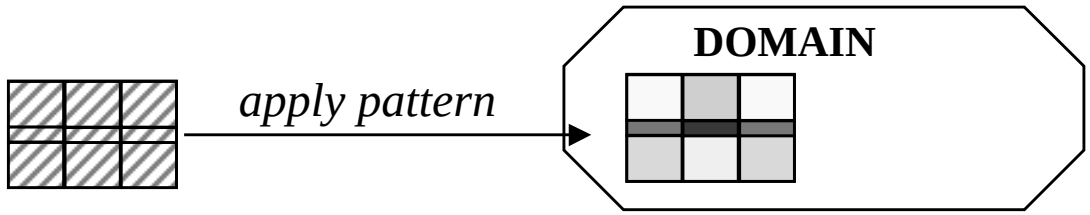
- Component Usage at different levels of abstraction
  - Interface is not known
  - Interface is known
- Drag ‘n’ drop interface assignment to Comp Specs
- Model Tree contents scoped by Diagrammer
- Interfaces Inheritance indication



What Will Make CBD Successful?

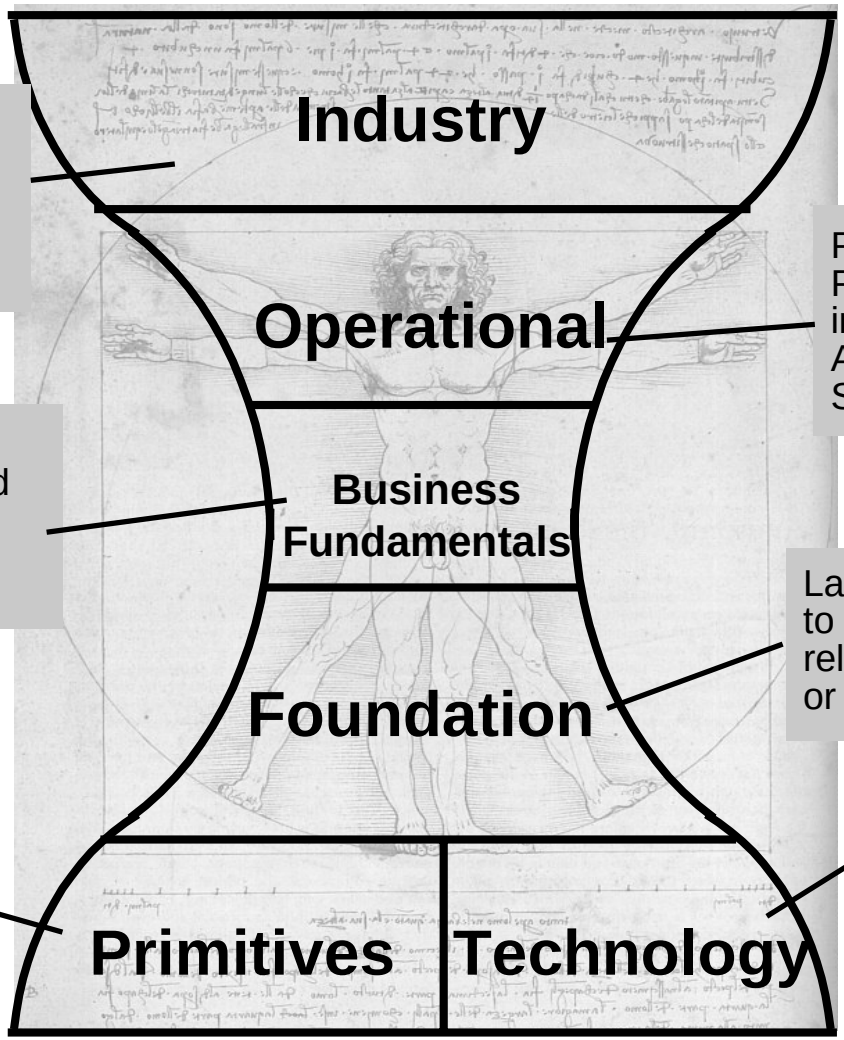
- Pattern captures the “essence” of some behavior
- May be architectures, specifications or designs
- May be expressed as a collaboration with generic types and attributes
- May define generic plug points
- Pattern is applied to a domain
  - generic aspects substituted with domain specific details
  - domain specific behavior is attached at plug points





What Will Make CBD Successful?

Different Kinds of Patterns



Application level patterns that can be applied across majority of business applications. E.g. Business Contact

Patterns (produced by Partners) that cover multiple industries or represent core Applications. E.g. Customer Service, A/P, Inventory

Application level patterns that can be applied across majority of business applications. E.g. Business Contact

Larger patterns that are generic to all applications. Follow data relations, component relations, or UI metaphor.

Small to medium sized basic building blocks of UI and Database Access

Small API wrappers e.g.ActixeX

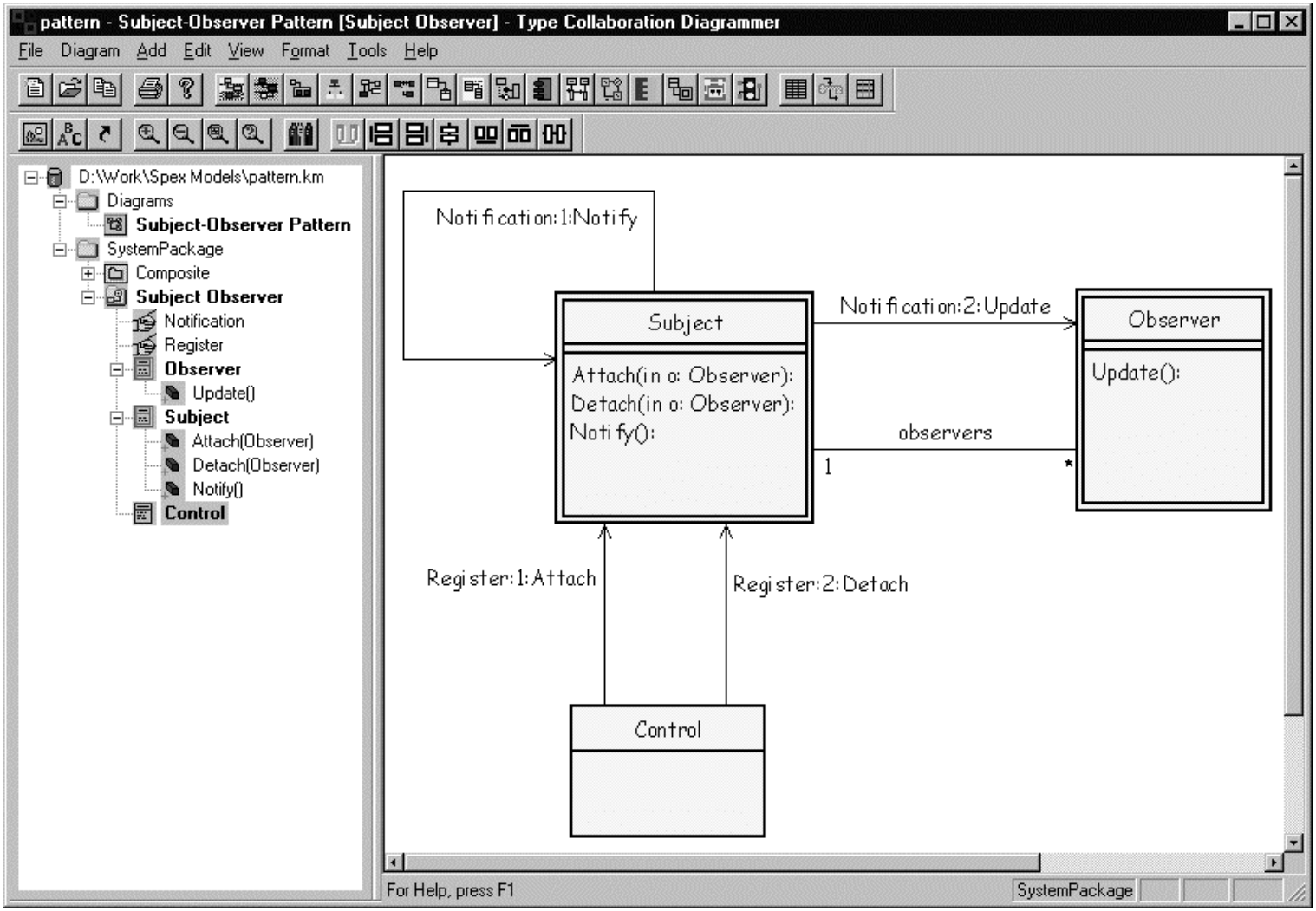
What Will Make CBD Successful?

Patterns and Components

- The combination of patterns and components presents many opportunities, e.g.,
  - components as building blocks to “plug-in” to placeholders in a pattern
  - patterns to provide design knowledge for designing and implementing component behavior
  - patterns capturing common component architectures
- Current efforts are focused on
  - identifying useful patterns
  - representing patterns as type collaborations
  - developing integrated component/pattern methods

What Will Make CBD Successful?

Patterns in COOL:Spex

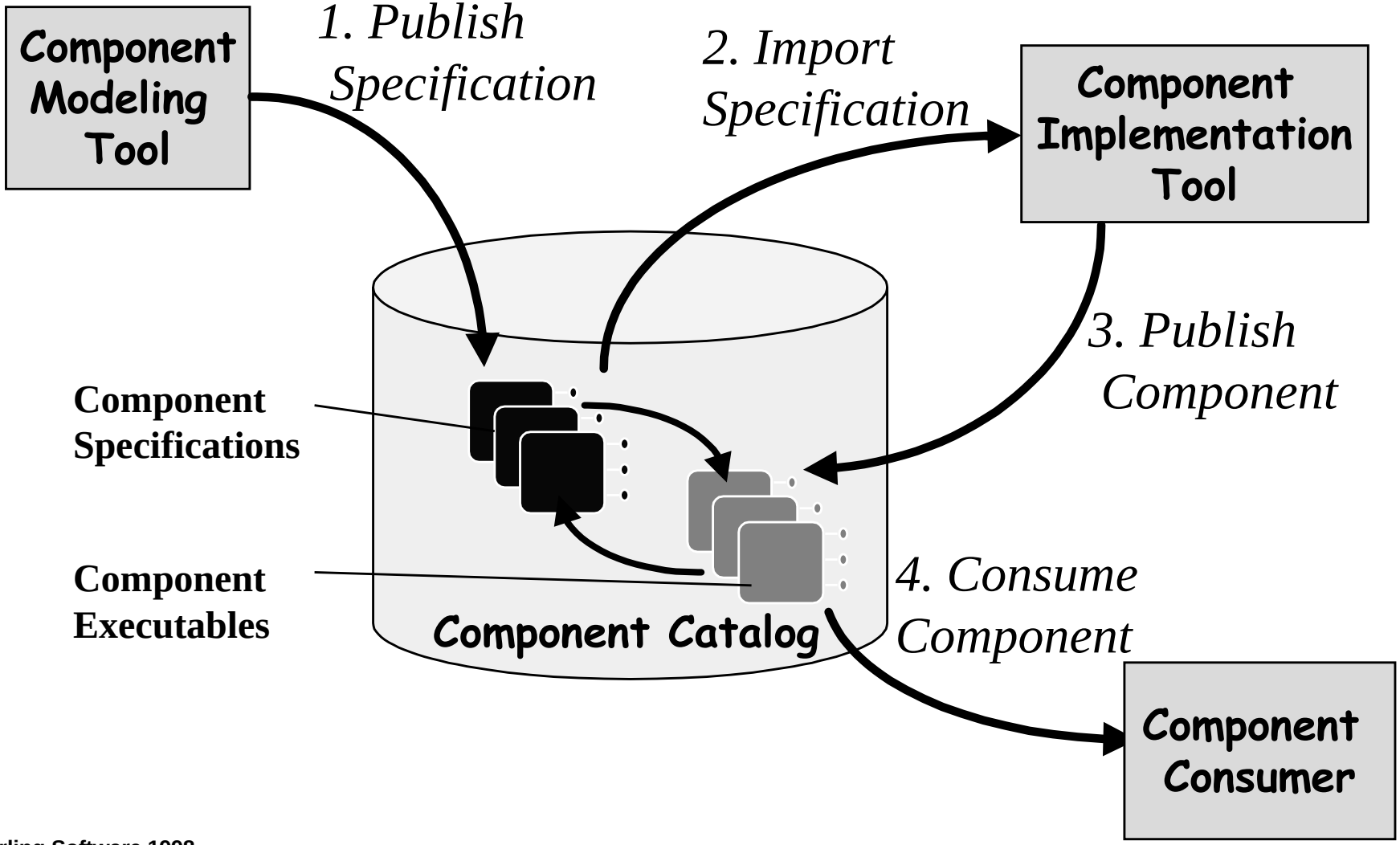


What Will Make CBD Successful?

- An integrated toolset requires
  - strong methods-based support, not just notation support
  - many entry and exit points for different users
  - a choice of implementation techniques to match different users' needs
  - the ability to integrate with external 3rd party tools and users
  - conformance to existing and emerging industry standards and practices

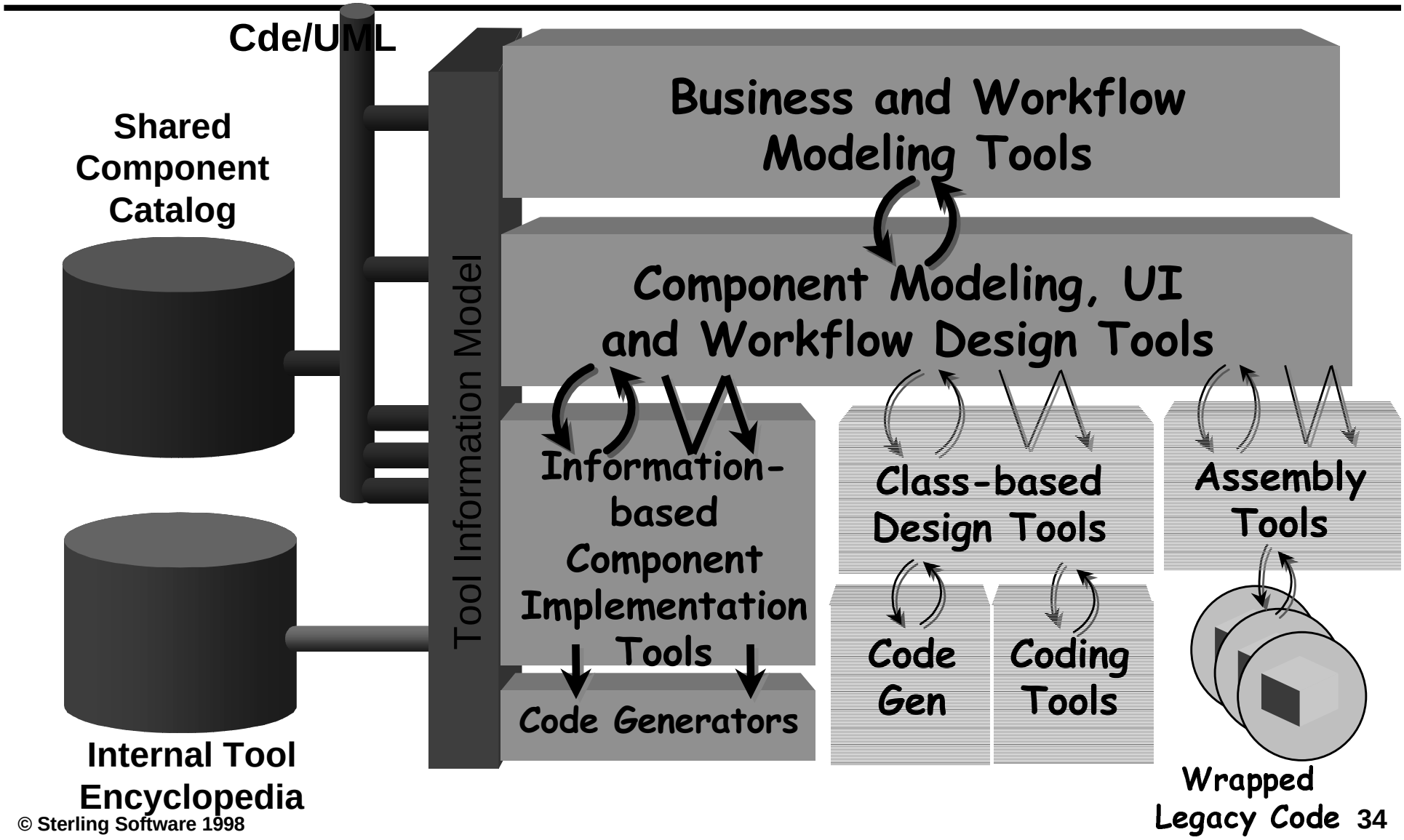
What Will Make CBD Successful?

A CBD Toolset



What Will Make CBD Successful?

A CBD Toolset Architecture



What is the Current State of CBD?

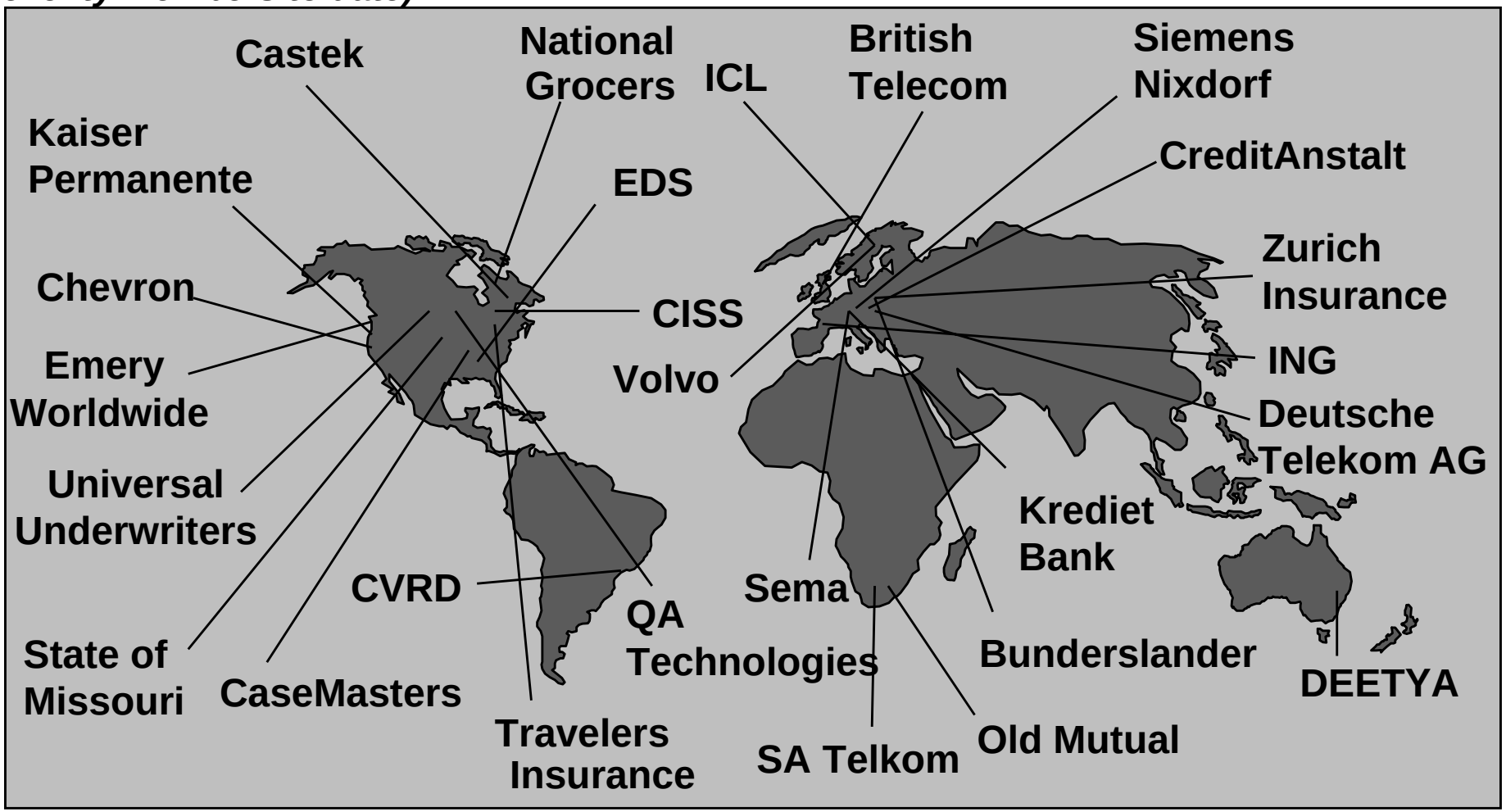
A Sterling Software Perspective

- Component approach added to COOL:Gen via the CBD 96 standard
- A documented set of conventions, naming standards, and best practices
- Changes to COOL:Gen specifically for CBD96
- Customer Advisory Board (CAB) for CBD
- Many documented successes with Enterprise-scale CBD
- A number of 3rd parties offering components and consulting services based on CBD96
- Release of the COOL:Spex tools supporting component architecture, specification, and design

What is the Current State of CBD?

CBD Customer Advisory Board

(A sampling of the more than seventy members to date)



What is the Current State of CBD?

CBD CAB Summary

As Reported at March '98 Conference

- 17 CAB members with 223 components in production
  - Ranging from 33 components by one member to 3 (highly reused) components in another
  - Components focus on delivering business value
  - Some are domain specific, others are infrastructure
- Team set-up
  - Development teams split into 'Provisioners' and 'Assemblers'
  - Others have one development team which develops and assembles components
  - ranging from 200 to 7 developers
- Many are now buying components from 3rd party provisioners

What is the Current State of CBD?

CBD Best Practices Guide

- Overall Framework, Process, Techniques & Definitions
- Re-use Strategies & Implementation
- Business Modeling
- Roles & Responsibilities
- Component Provisioning
- Application Development & Solution Provisioning
- Documenting CBD Projects
- Procurement Management
- Repository & Model Management
- Configuration, Implementation & Change Management
- CBD Training & Support
- Measurement and Evaluation
- Project Management
- Technical Environment for CBD

What is in the Future for CBD?

Critical Areas for the Future

- There are a number of areas critical to future productivity and quality in CBD approaches, e.g.,
  - Taking advantage of new technologies for dynamic system assembly from components and patterns
  - More rigorous component interface descriptions to improve selection, testing, and assembly of components
  - Integrating descriptions of logical component behavior with practical application and platform architecture constraints during component specification and assembly

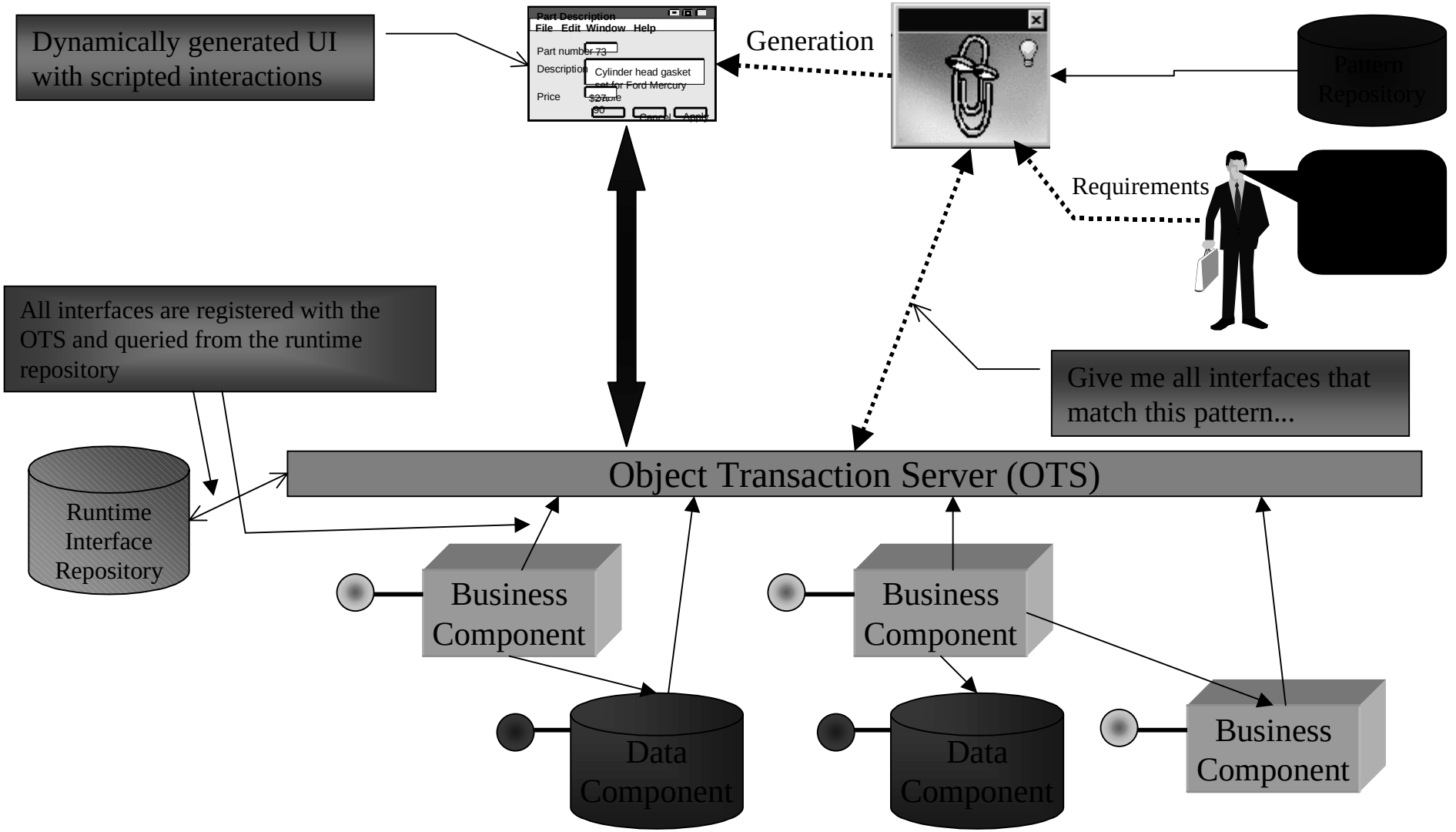
What is in the Future for CBD?

New Technologies

- Several technology advances will radically alter applications development over the next decade
  - increased processor power and network capability
    - provides the infrastructure for greater interaction and collaboration in performing tasks
  - special devices, hand-held displays, and greater use of voice input improve human-computer interaction
    - the flexibility to reuse existing business rules and data in new application domains
  - agent-based systems will apply greater intelligence to information gathering and display
    - applications will be more dynamically assembled to solve new business needs as they arise

What is in the Future for CBD?

Future Application Architecture?



Source: Mike Sparling, "Components: Today, Tomorrow, and Beyond…", Casetek, 1998.

What is in the Future for CBD?

Rigorous Component Specifications

- Component specifications require more precision

  e.g.,

  pre *The member belongs to the membership manager and owns a membership to an organization known to the membership manager and the member has a recorded subscription.*

  (m . belongs_to ≠ NIL) ^

  (o. known_by ≠ NIL)

  (∃ ms:Membership • ms ∈ m . owns ^ o . comprises) ^

  (m. owns . subscription ≠ NIL)

  post *The membership level is updated to the new level number supplied.*

  m . owns . level = new_level

What is in the Future for CBD?

…but rigor with practicality!

- Most users cannot easily use formal languages directly
- We need more innovative approaches based on wizards, English-like concrete syntax, etc.
- Also, we must guide the use to the right situations to use formal notations, and when they are not necessary
- Many practical issues in the use of formal methods remain unaddressed
- If we can make some progress here, we open up many opportunities in the areas of:
  - more intelligent cataloging and searching of components
  - automatic test case generation
  - enhanced completeness and consistency checks

What is in the Future for CBD?

Technology-specific architecture

- Component-based design must allow users to express technology and platform requirements and constraints
- These must be part of the modeling process, and will influence many quality attributes of the resultant systems
- At what stage in the component life-cycle should technology-specific issues become important?
- How can we take advantage of technology aspects to guide users on the most appropriate solutions to their needs?
- How do we isolate technology specific aspects to allow reuse of component specifications in different contexts?

What is in the Future for CBD?

One approach is ADLs

- A great deal of research work has taken place in the area of Architecture Description Languages (ADLs)
- People have considered architectural styles and idioms, collected case studies of successful architectures, considered qualities of software architectures, and developed simple guidelines to avoid common mistakes
- We can build on this work to introduce architectural aspects of target technologies into our tools
- Introducing the idea of “technology containers” as a way to expose technology-specific issues in a practical, usable way

Summary

CBD is the future of EAD

- We are in a very dynamic marketplace with constantly changing:
  - business practices
  - organizational structures
  - technology infrastructures
- The key to the future is being able to manage and control these changes
- Component-based development provides significant benefits for enterprise applications
- As CBD becomes a dominant AD approach, other solutions will become less attractive