


MBASE Design Overview

CS477, Spring 2003

Ed Colbert
USC Center for Software
Engineering

1 3/6/2003



Goal of Presentation

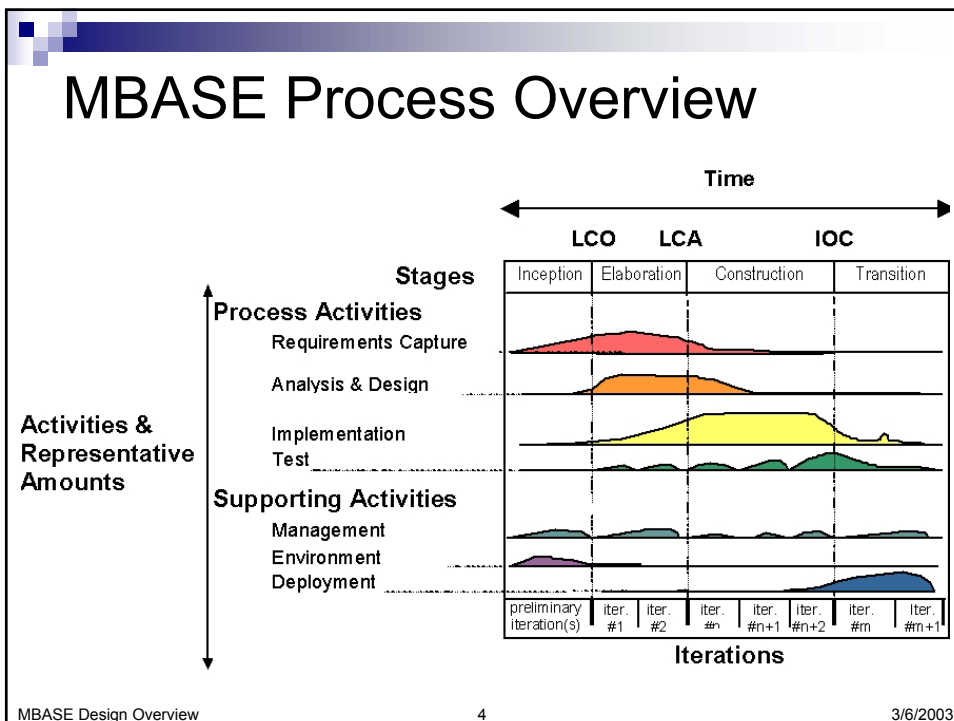
- Understand how to perform Object-Oriented Design
 - Using
 - MBASE
 - Object-oriented techniques
 - RUP
 - Rational Rose
- Understand how to document design

MBASE Design Overview 2 3/6/2003

Outline

- When Last We Met ...
- Design Process Overview
- Design Process by Example
- Summary

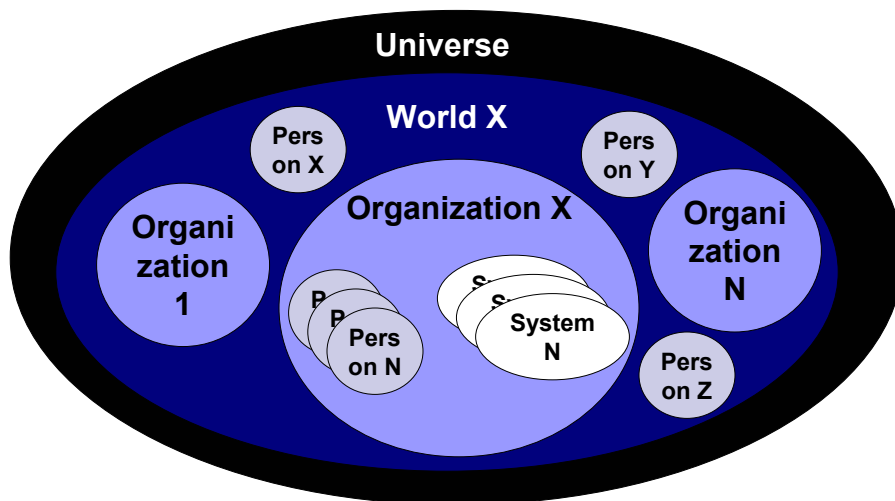
MBASE Design Overview 3 3/6/2003



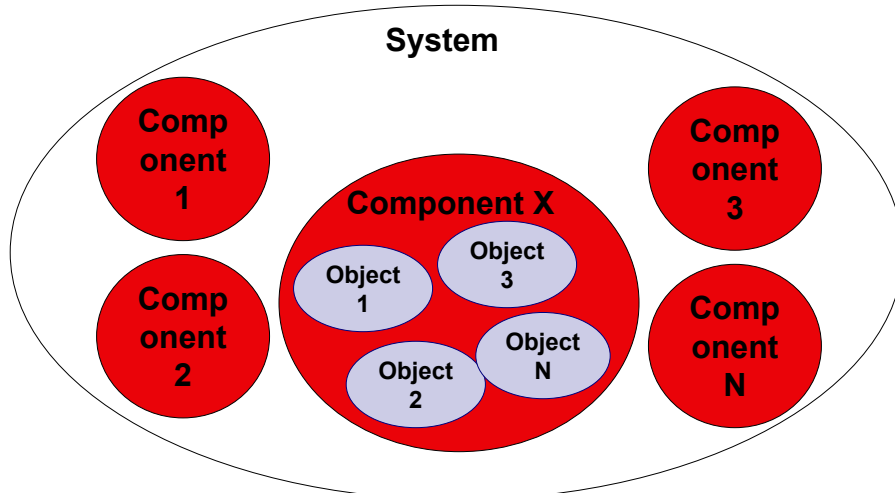
Purposes of Architecture Analysis & Design

- To transform requirements into design of system
- To evolve robust architecture for system
- To adapt design to match implementation environment
 - Designing it for performance
 - RUP 2001

Object, Component, System, People, Organization,



Object, Component, System, People, Organization, (cont.)



MBASE Design Overview

7

3/6/2003

What You Need To Describe At Any Level?

- **Structure**
 - External Perspective (Specification/Interface)
 - Internal Perspective (Implementation)
- **Responsibilities**
 - its state
 - its dynamic behavior
 - the operations it can perform
 - the requests it makes of other objects
 - Each component/object collaborates (works) with other components/objects in performance of its responsibilities
- **Qualities**
 - Quantitative (e.g., size, speed)
 - Qualitative (e.g., "user friendly", re-usable)

MBASE Design Overview

8

3/6/2003

MBASE Describes **System Internal (Architectural) View** in SSAD Section 3

- Structure
 - Sec 3.1
- Responsibilities
 - its state
 - its dynamic behavior
 - the operations it can perform
 - the requests it makes of other objects
 - Sec 3.2
 - Sec 3.3
 - Sec 3.5
- Qualities
 - Quantitative
 - Qualitative
 - Sec 3.4

Independent of implementation details

MBASE Design Overview 9 3/6/2003

Define Architecture Software Component Classifier Model For Full-text Title Database System, Top-Level

```

    graph TD
      LU["<<Component>>  
Library_User"] -.-> S["<<Component>>  
Server"]
      SA["<<Component>>  
System_Administrator"] -.-> S
    
```

- Decision
 - Client-Server Model

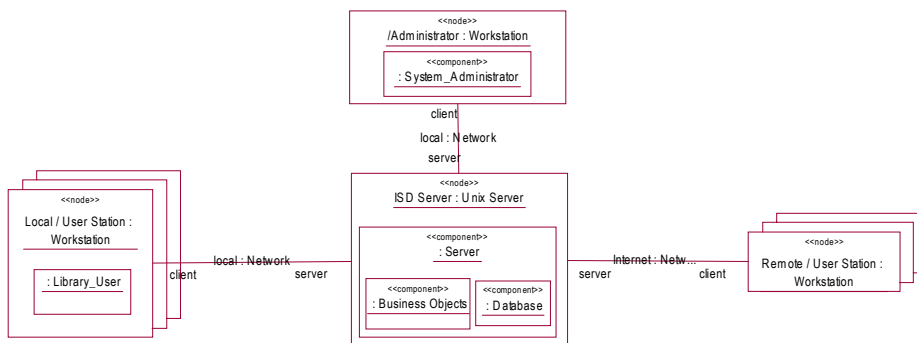
MBASE Design Overview 10 3/6/2003

Define Architecture Software Component Classifier Model For Full-text Title Database System, Server Component

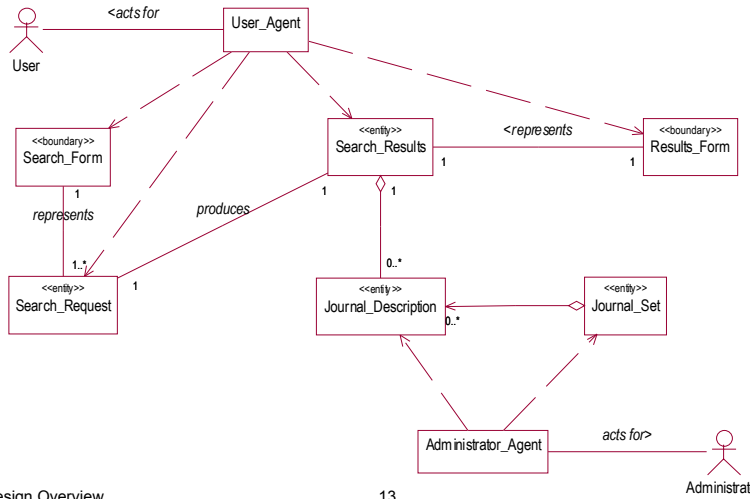


- Note: using a style that allows nested component classifiers

Define Architecture Deployment Diagram for Full-Text Title Database System



Define Analyze Classes Enterprise Classification Model for Full-Text Title Database System

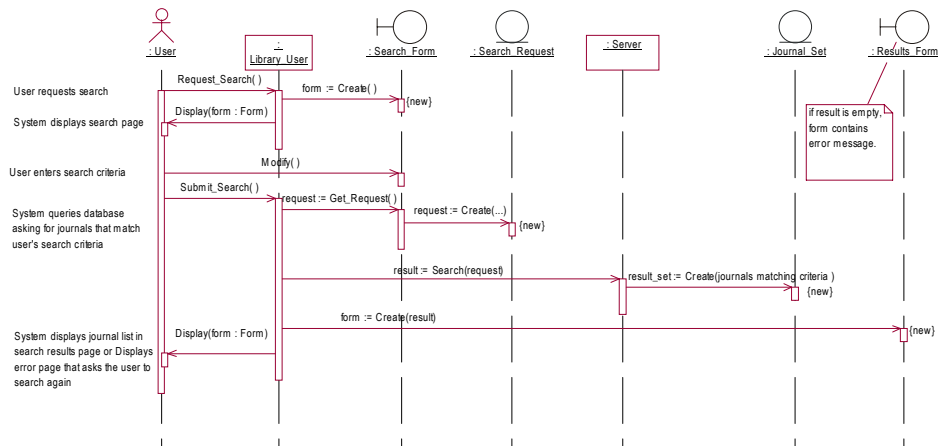


MBASE Design Overview

13

3/6/2003

Analyze Behavior (1st Build) Search and Locate Journals



MBASE Design Overview

14

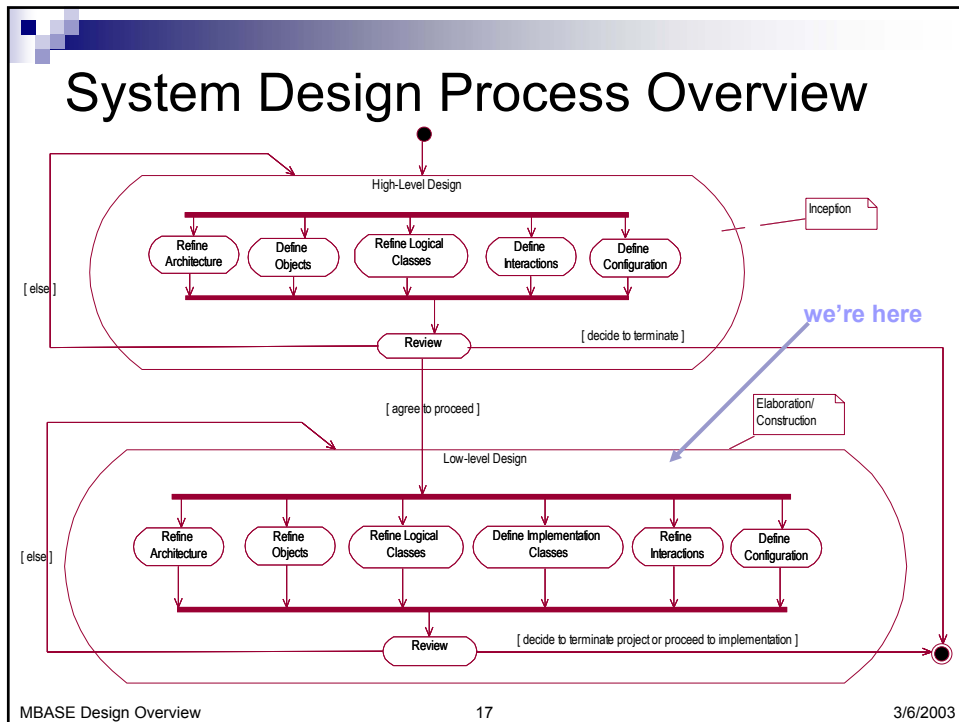
3/6/2003

Outline

- When Last We Met...
- **Design Process Overview**
- Design Process by Example
- Summary

System Design

- Describe how system can be implemented in software
- Describes specific technology solutions that satisfy Project & System requirements
- 2-levels
 - High-level Architectural Design
 - Resolves Analysis issues
 - e.g. how will roles and states be handled, expand bi-directional relationships, break multi-way relationships, handle global and relational attributes, decompose Components into objects, complex dependencies & other constraint
 - **Low-level Implementation Design**
 - **Implementation considerations**
 - e.g. use of databases, web-servers, hardware, critical algorithms, sequence, significant events, GUI's, etc.



Outline

- When Last We Met...
- Design Process Overview
- **Design Process by Example**
- Summary

MBASE Describes **System Internal (Implementation) View** in SSAD Section 4

- Structure
 - Sec 4.1 & 4.5
- Responsibilities
 - its state
 - its dynamic behavior
 - the operations it can perform
 - the requests it makes of other objects
 - Sec 4.2
 - Sec 4.3
- Qualities
 - Quantitative
 - Qualitative
 - Sec 4.4

Independent of implementation details

MBASE Design Overview 19 3/6/2003

Refine Architecture – LCA or IOC

- Purpose:
 - Incorporate implementation decisions into architecture for system
 - Based on experience gained from similar systems or in similar problem domains
 - Identify Components
 - Understand
 - Hardware execution environment
 - Allocation of components to hardware
- Inputs:
 - LCO Architecture
 - LCO Logical Class Model
 - LCO Object Model
 - LCO Interaction Model
 - Architectural Patterns
 - L.O.S Requirements
 - Evolution Requirements

MBASE Design Overview 20 3/6/2003

Refine Architecture – LCO or IOC Artifacts

- **Component Model**
 - Define Implementation Component Model
 - Determine whether components are implemented by COTS or not
 - Determine interfaces of components
 - Define implementation-specific classes, objects, interactions
 - Refine component classes
 - e.g. select specific implementation stereotypes
 - Add implementation-specific components
 - Identify dependencies between components
- **System Topology**
 - Updated as appropriate to reflect changes to component model
- **System Deployment Model**
 - Updated as appropriate to reflect changes to component model

MBASE Design Overview 21 3/6/2003

Refine Architecture – LCO or IOC Implementation Component Model

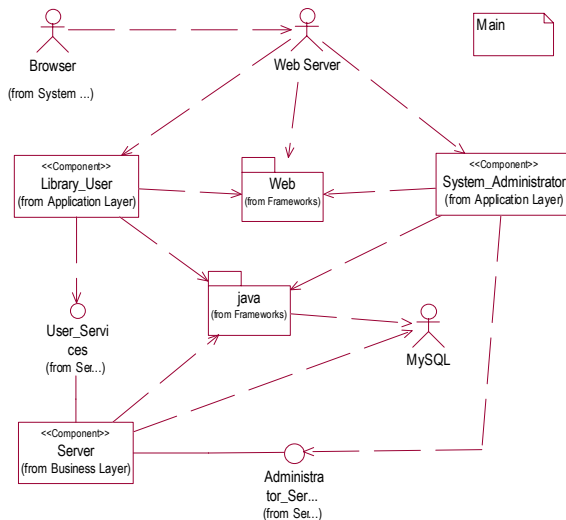
- **Purpose:**
 - Describe how the components will be implemented
 - Define their interfaces
 - Describe COTS products that will be used to implement components, and how they are configured
 - Identify development technologies to be used
 - including database tables, Java, XML/HTML, HTTP servers, APIs, class libraries, design patterns
- **Inputs:**
 - See revised component model
- **Artifacts**
 - Interface Class Diagram(s)
 - Implementation Class Model
 - Description of COTS configuration
 - See *MBASE COTS Integration Supplement*

MBASE Design Overview 22 3/6/2003

Implementation Component Model For Full-text Title Database System

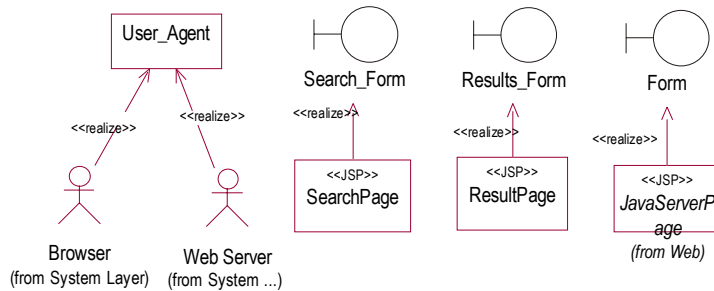
- Document in SSAD 4.1.3
 - Library_User Component
 - Will be implemented with web-browser that access pages generated by Java Server Pages
 - System_Administrator Component
 - Will be implemented with web-browser that access pages generated by Java Server Pages
 - Server Component
 - Server::Business Objects
 - Server::Database
 - Will use MySQL as DBMS
 - Will implement classes in Java
- (See MBASE Guide & *MBASE COTS Integration Supplement* for details of section)

Refine Architecture – LCA or IOC Component Model For Full-text Title Database System



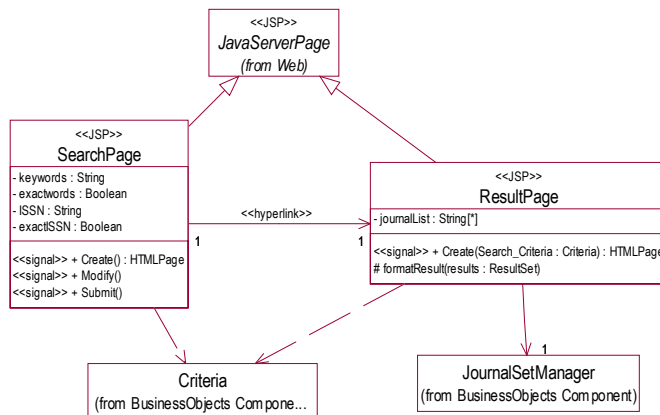
- Revise to show any changes due to implementation decisions
 - Use of COTS
 - Web Browser
 - Web Server
 - MySQL
 - Dependency on
 - Java Framework
 - Web Framework

Define Implementation Class Model – LCO or LCA For Library_User Component



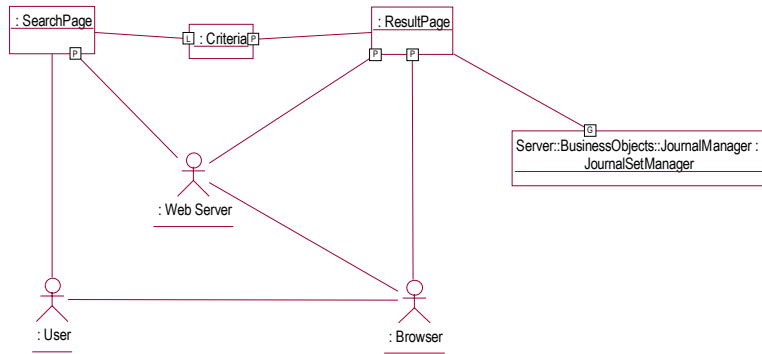
- Define implementation of logical classes

Define Implementation Class Model – LCO or LCA For Library_User Component



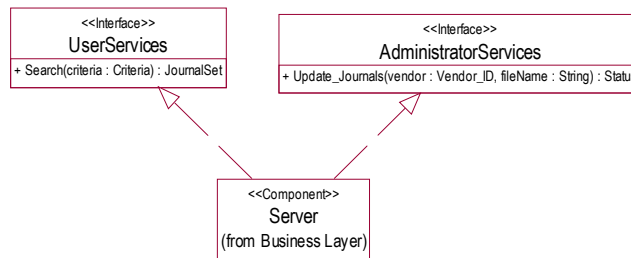
- Define attributes, operations, and relations for implementation classes

Refine Object Model – LCO or LCA For Library_User Component



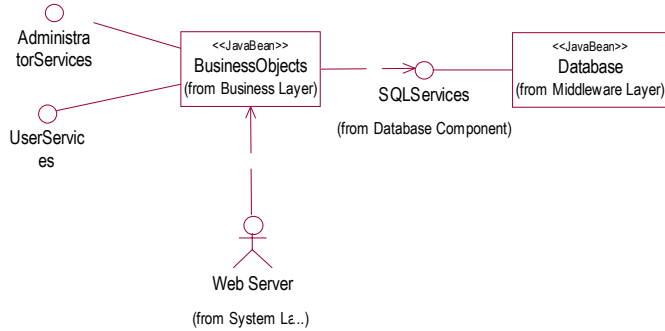
- Define implementation objects that are used to build component
 - May not need if implementation class model is real close to logical class model

Define Component Interfaces – LCO or LCA For **Server** Component



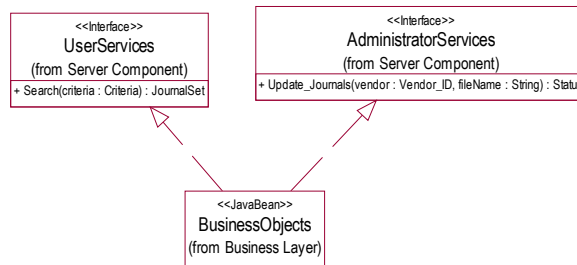
- Define *interfaces* for Component
 - Define operations available in each *interface*
 - Define dependencies on other classes & *interfaces*
- Note: deferring complete AdministratorServices *interface* until next build

Refine Architecture – LCA or IOC Component Model For **Server** Component



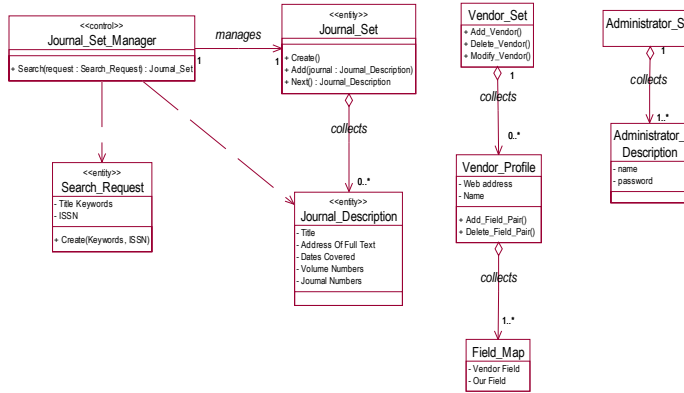
- Revise to show any changes due to implementation decisions
 - Interface named SQLServices supplied by Database component
 - Database component will be Java Bean
 - New BusinessObjects Component (Java Bean)
 - Decouple higher-level components from decisions about Database (e.g. SQL)
 - Realizes AdministratorServices & UserServices interfaces

Define Component Interfaces – LCO or LCA For Server::BusinessObjects Component



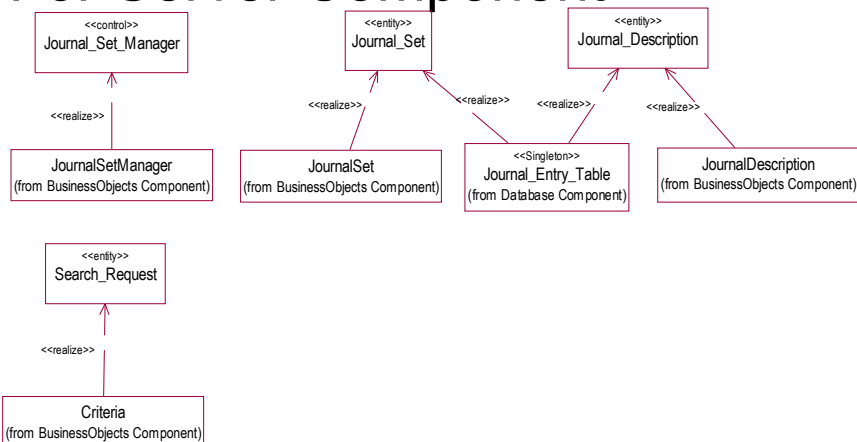
- Define *interfaces* for Component
 - Define operations available in each *interface*
 - Define dependencies on other classes & *interfaces*
- Notes:
 - Deferring complete AdministratorServices *interface* to later build
 - This diagram is shown for completeness
 - Information already resented in Component Diagram for Server

Refined Logical Class Model – LCA or IOC Component Model For **Server** Component



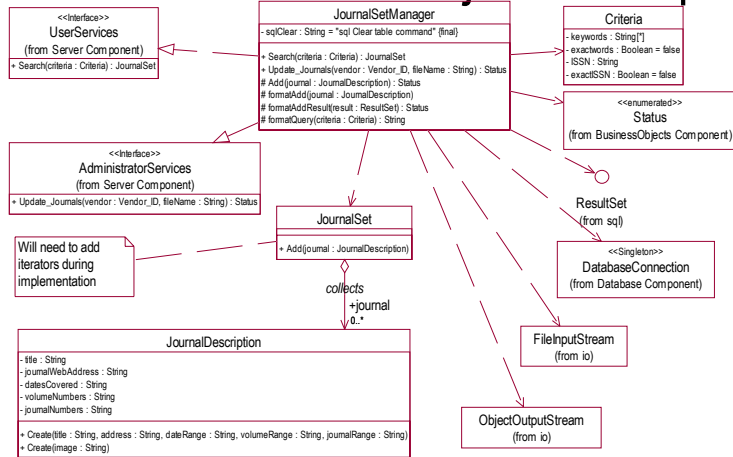
- When add BusinessObjects Component, moved this diagram (and logical Object-Structure Diagram) from Database subcomponent to Server component
 - Realized that this diagram represented classes in both BusinessObject & Database subcomponents
 - Alternative is to duplicate diagram in each component

Define Implementation Class Model – LCA or IOC For **Server** Component



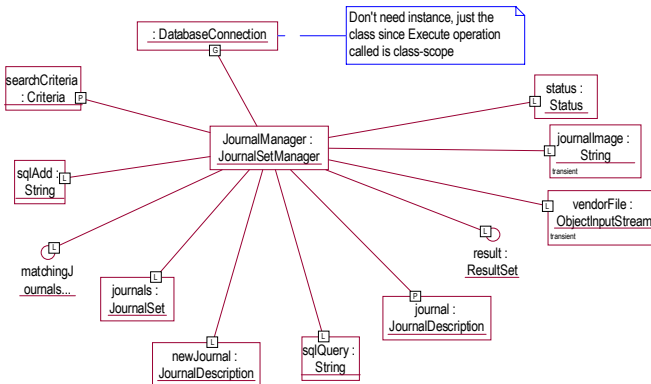
- Define implementation of logical classes
- Note: deferring Vendor_Set & User_Set to later build

Define Implementation Class Model – LCA or IOC For Server::BusinessObjects Component



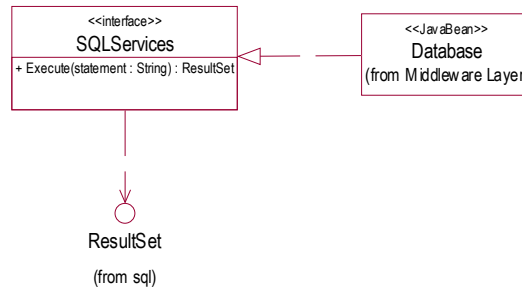
- Define attributes, operations, and relations for implementation classes
- Note: deferring Vendor_Set & User_Set implementations to later build

Refine Object Model – LCA or IOC For Server::BusinessObjects Component



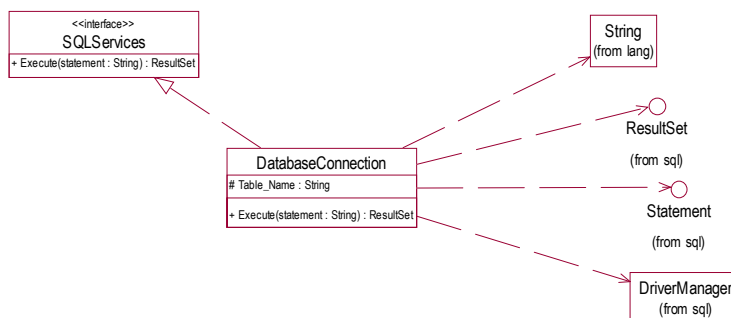
- Define implementation objects that are used to build component
 - May not need if implementation class model is real close to logical class model

Define Component Interfaces – LCA or IOC For Server::Database Component



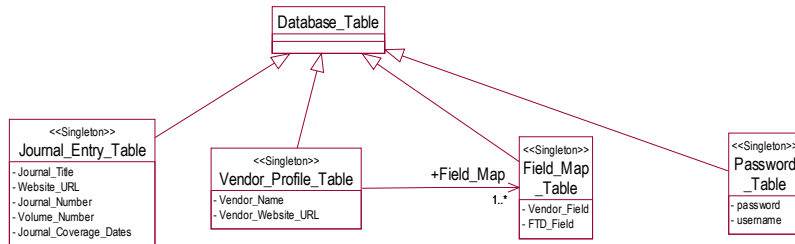
- Define *interfaces* for Component
 - Define operations available in each *interface*
 - Define dependencies on other classes & *interfaces*

Define Implementation Class Model – LCA or IOC For Server::Database Component



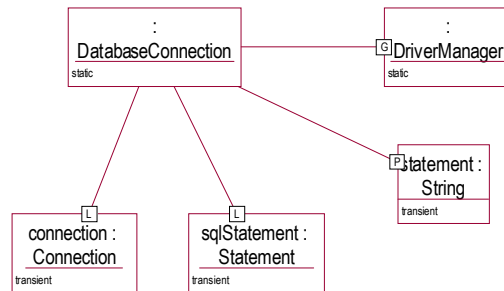
- Define attributes, operations, and relations for implementation classes

Define Implementation Class Model – LCA or IOC For Server::Database Component, Relational Database Schema



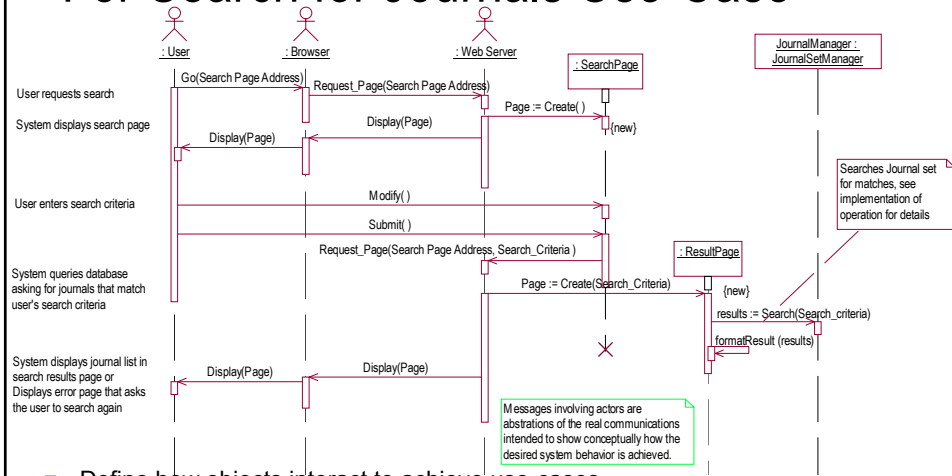
- Define attributes, operations, and relations for implementation classes

Refine Object Model – LCA or IOC For Server::Database Component



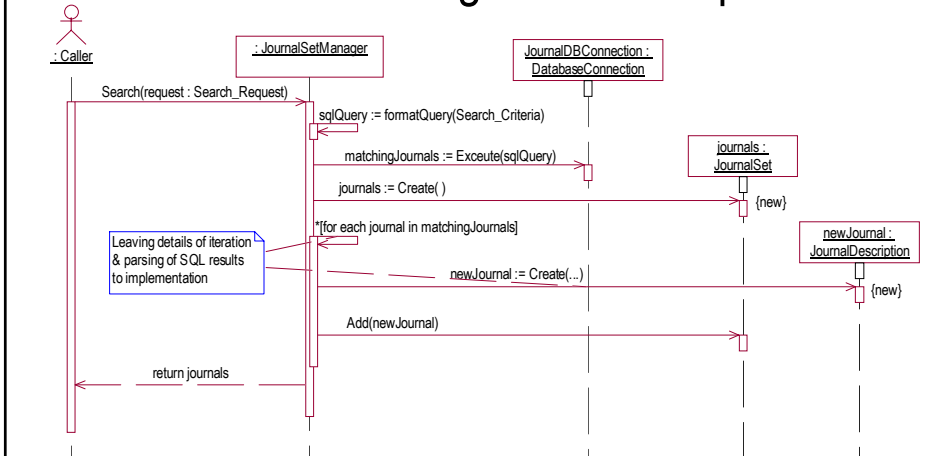
- Define implementation objects that are used to build component
 - May not need if implementation class model is real close to logical class model

Refine Interaction Model – LCA or IOC For Search for Journals Use-Case



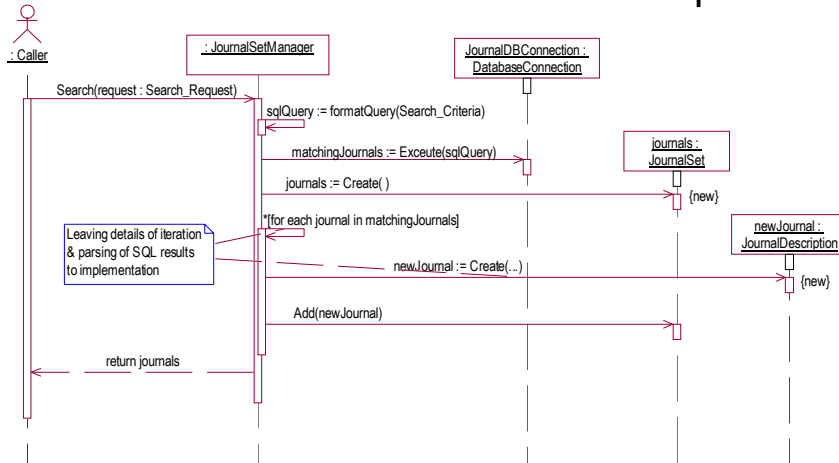
- Define how objects interact to achieve use-cases
 - May not need if implementation class model is real close to logical class model

Refine Interaction Model – LCA or IOC For JournalSetManager.Search Operation



- Define how objects interact to achieve use-cases
 - May not need if implementation class model is real close to logical class model

Refine Interaction Model – LCA or IOC For DatabaseConnection.Execute Operation



- Define how objects interact to achieve use-cases
 - May not need if implementation class model is real close to logical class model

Refine Deployment Model – LCA or IOC Component Configuration for ...

- /Administrator : Workstation



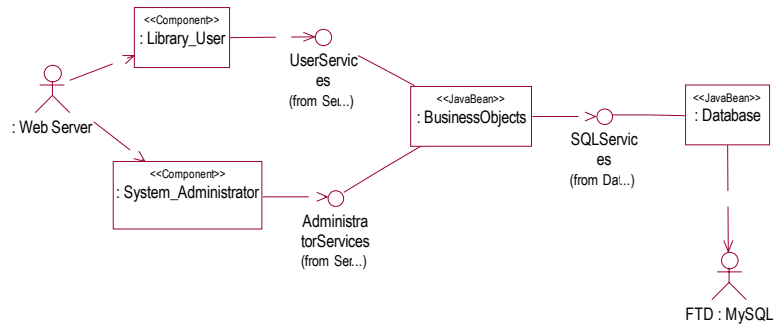
- /User : Workstation



- Define how software components are allocated to hardware nodes
 - May not need if Implementation Deployment Model is same to Logical Deployment Model

Refine Deployment Model – LCA or IOC

Component Configuration for : Unix Server



- Define how software components are allocated to hardware nodes
 - May not need if Implementation Deployment Model is same to Logical Deployment Model

Define Artifact Configuration Model

- Purpose:
 - Define *artifacts* that will be produced during implementation
 - Source files
 - Scripts
 - Binary files
 - Dynamic or static link libraries
 - Database Tables
 - Define what components or classes are in each artifact?
 - Define which artifacts “know” each other
- Inputs:
 - Component Model
 - Implementation Class Model
- Artifacts:
 - Configuration Model
- Typically done at LCO & LCA only if creating executable prototype

Configuration Model – LCA or IOC For Full-text Title Database System

Contents of FTD root directory

Contents of Library_User package\directory

- Define
 - File structure
 - What Components or classes are in each file
 - Directory structure

MBASE Design Overview
45
3/6/2003

Outline

- When Last We Met...
- Design Process Overview
- Design Process by Example
- **Summary**

MBASE Design Overview
46
3/6/2003

What You Need To Describe At Any Level?

- Structure
 - External Perspective (Specification/Interface)
 - Internal Perspective (Implementation)
- Responsibilities
 - its state
 - its dynamic behavior
 - the operations it can perform
 - the requests it makes of other objects
 - Each component/object collaborates (works) with other components/objects in performance of its responsibilities
- Qualities (Levels of Services)
 - Quantitative (e.g., size, speed)
 - Qualitative (e.g., “user friendly”, re-usable)

MBASE Design Overview

47

3/6/2003

MBASE Describes **System Internal (Architectural) View** in SSAD Section 3

- | | |
|--|-----------|
| ■ Structure | ■ Sec 3.1 |
| ■ Responsibilities | ■ Sec 3.2 |
| □ its state | ■ Sec 3.3 |
| □ its dynamic behavior | ■ Sec 3.5 |
| □ the operations it can perform | |
| □ the requests it makes of other objects | |
| ■ Qualities | ■ Sec 3.4 |
| □ Quantitative | |
| □ Qualitative | |

Independent of implementation details

MBASE Design Overview

48

3/6/2003

MBASE Describes **System Internal (Implementation) View** in SSAD Section 4

- Structure ■ Sec 4.1 & 4.5
- Responsibilities ■ Sec 4.2
 - its state ■ Sec 4.3
 - its dynamic behavior
 - the operations it can perform
 - the requests it makes of other objects
- Qualities ■ Sec 4.4
 - Quantitative
 - Qualitative

Independent of
implementation
details