

System Analysis II

CS577a

Fall 2001

Ed Colbert

USC Center for Software
Engineering



Goal of Presentation

- Understand how to perform System Analysis
 - Using
 - MBASE
 - Object-oriented techniques
 - RUP
 - Rational Rose
- Understand how to document analysis
- Presentation is part 2 of 3 on System Analysis



Outline

- **Key Concepts**
- Process Overview
- Example Project Description
- Process by Example



Purposes of Analysis & Design

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



Analysis & Design Approach

- Architecture–centric
 - Define stable architecture early
- Use–case Driven
 - Use–case describe capabilities/usage of system
 - Use–case help identify objects & operations needed
- Iterative & Incremental



Analysis Goals

- Quantify *what* we want to represent
 - Not *how* it is done
- Formalize & refine automated parts of organizations capabilities, entities, activities, & interactions described in domain description
- Model high-level architectural of system



Analysis Audience

- Domain Description is for all constituents of the project
- Analysis is for Domain Experts
 - i.e. high level leaders who
 - Understand domain
 - Know what they want,
 - Have authority to make decisions
- Not for implementers
 - Who prefer design and implementation details (“hows”)



What's A Software Architecture?

- *Software Architectures* [Shaw & Garlan 96] defines for a system
 - Computation components
 - Clients
 - Servers
 - Databases
 - Filters
 - Layers
 - Interactions among components
 - Subprogram calls
 - Shared data
 - Client–server
 - DB–accessing protocols
 - Asynchronous even multicast
 - Piped streams
 - etc.



What's A Software Architecture? (cont.)

- IEEE Working Group on Architecture [IEEE98]
 - Defines as highest-level concept of a system **in its environment**
 - Concern with
 - Fit with system integrity
 - Economic constraints
 - Aesthetic concerns
 - Style



What's A Software Architecture? (cont.)

■ RUP

- Organization or structure of system's significant components
- Interacting through interfaces
- With components composed of successively smaller components & interfaces



What's a Component?

■ Many Definitions

- *Software Architectures* [Shaw & Garlan 96]
 - Loci of computation & state
 - Has an interface specification that defines its properties
- *Component Software: Beyond Object-Oriented* [Szyperski 97]
 - A unit of
 - independent deployment
 - third-party composition
 - Has no persistent state



What's a Component? (cont.)

□ UML

- Modular, deployable, & replaceable part of system that
 - Encapsulates implementation
 - Exposes set of interfaces
 - Specified by one or more classifiers that reside on it
- May be implemented by one or more artifacts
 - e.g., binary, executable, or script files

□ RUP

- A **non-trivial, nearly independent**, & replaceable part of system that fulfills clear function in context of well-defined architecture
- Conforms to & provides physical realization of a set of interfaces



What's a Component? (cont.)

□ MBASE Guide

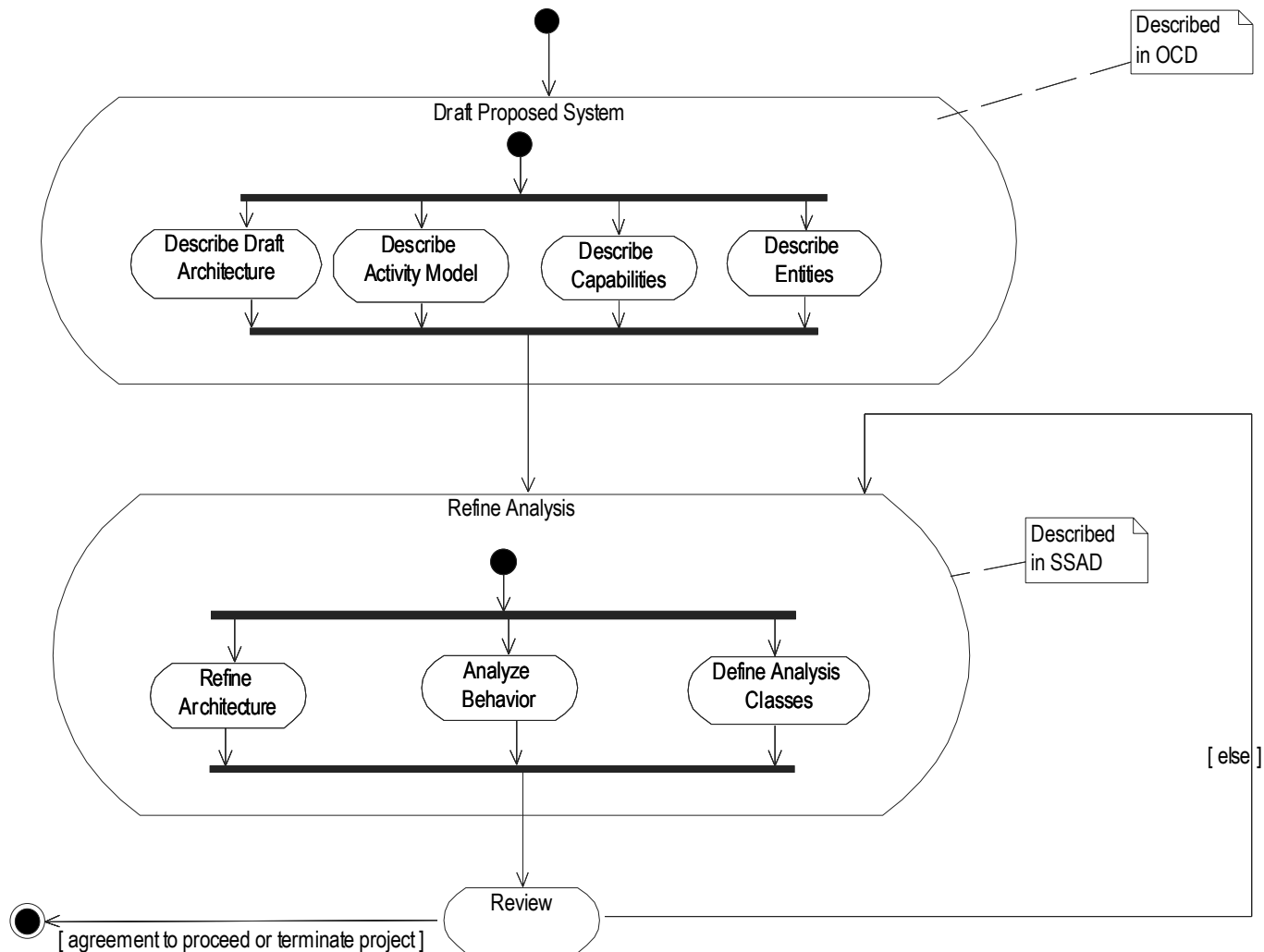
- An abstraction that represents both memory & functionality
- A partition of system
- A part of system
- Implemented by objects
- Detail guides for identifying components are typically used for objects
- We'll use
 - "component" to mean part
 - "*Component*" to refer to UML/RUP definition



Outline

- Key Concepts
- **Process Overview**
- Example Project Description
- Process by Example

System Analysis Process Overview





Outline

- Key Concepts
- Process Overview
- Example Project Description
- Process by Example



OCD 2.1 System Capability Description

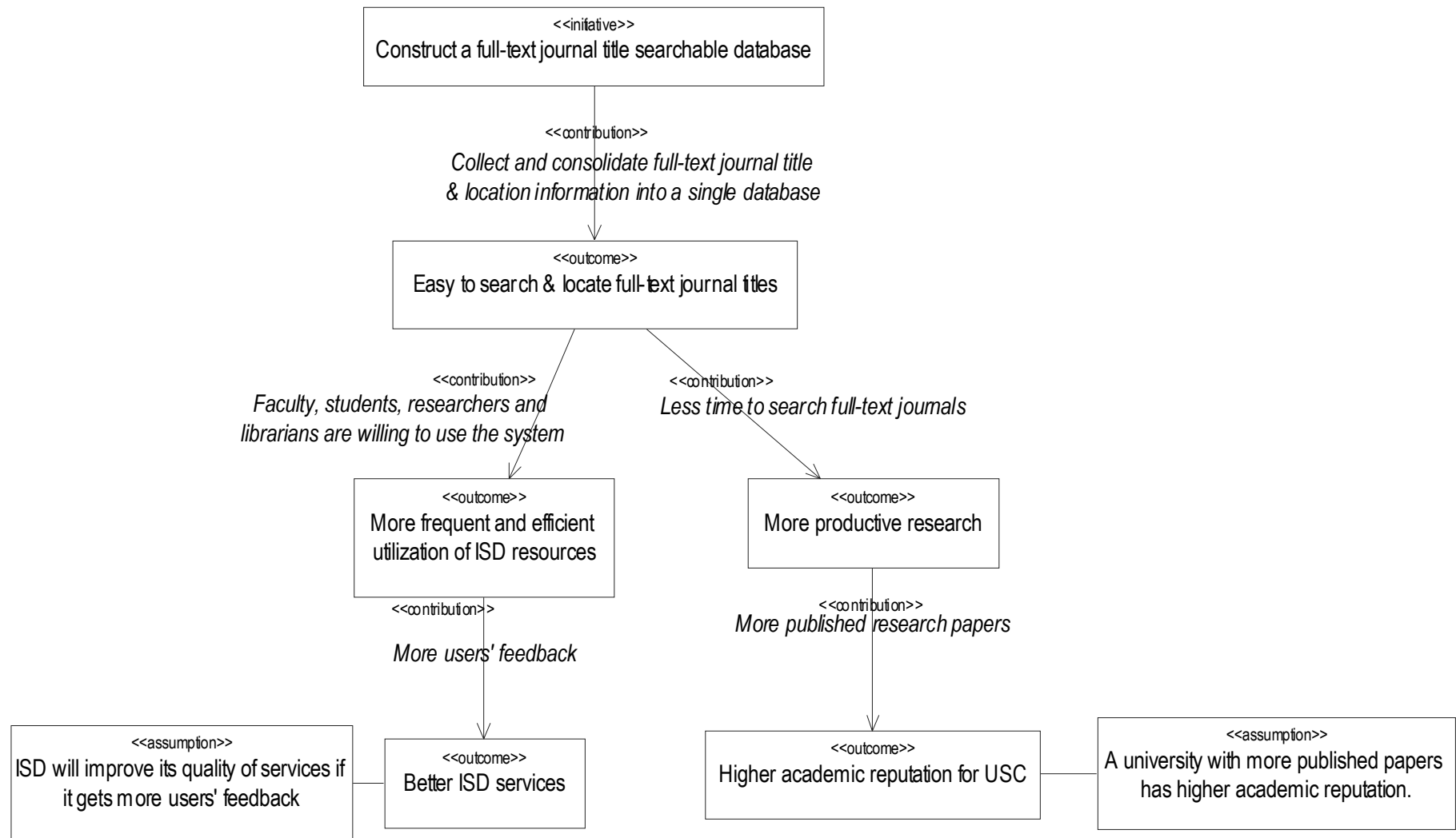
- Target customer: Information Service Division (ISD) in University of Southern California
- The system is proposed and implemented for faculty, students, researchers and librarians at USC who need to identify where a particular periodical title is indexed, the date of coverage and whether it is available in full-text.
- The Full-text Title Database is a web-based full-text journal title searchable database.
- That system retrieves and consolidates journal title information from different vendors and provides it to general public users such as faculty, students, researchers and librarians.
- Unlike the Jake project at Yale University and full-text journal and newspaper search engine at Indiana University, our product is designed dedicated to USC which is accessible globally via World Wide Web and provides full-text journal title information to general public especially USC community. The Jake project of Yale University and the journal and newspaper search engine at Indiana University provide similar functionality. However, the former includes some information in those vendors' databases that USC does not has access to and the later only contains journal title information from databases that Indiana University has access to which is inadequate.



OCD 2.1.1 Benefits Realized

- The database we are going to build will let the general public users (faculty, students, researchers, librarians, etc.) easy to find the full-text journal title information they need. They can do a search in our database instead of searching each vendor's database USC has access to one after another. Thus it will much reduce the full-text journal searching time in the research. Since it's an easy to use and powerful system, more and more users are willing to use our system. It leads to more frequent and efficient utilization of resources of ISD, and more productive research work by faculty, students and researchers at USC. And it will lead to better ISD service and a higher academic reputation for USC.
- The database we are going to build will provide a web-base interface to system administrator, so ISD can easily find a person who has some basic computer knowledge to maintain the whole system, and doesn't have to provide much training to him/her or additional resources. Therefore ISD can save budget and time on a system which is easy to maintain like this proposed system.

OCD 2.1.2 Results Chain

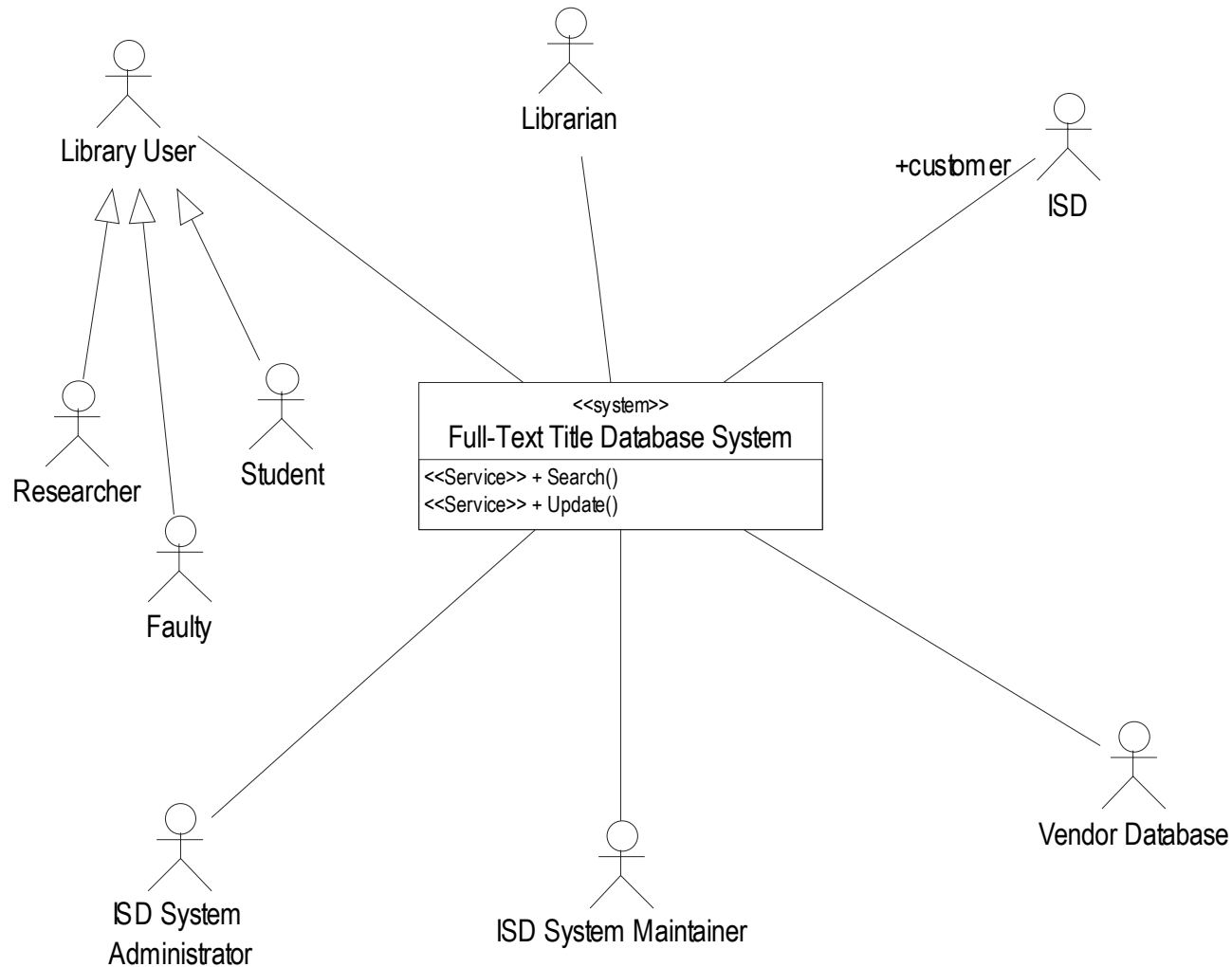




OCD 2.2 Key Stakeholders

- **Software/IT system's users: Faculty, Researchers, Students, and Librarians at USC.**
 - The relationship with result chain: The users will be the ones who are going to interact with our system through user interface. If they are willing to use our system, they will get the benefit which is to "fill a serious gap in the tools available and result in more efficient research for researchers and students".
- **Customers: ISD**
 - The relationship with result chain: Whom we are going to develop the system for. This is also the one who is going to inspect the result chain.
- **Developers: CS577a Team8 & CS577b Team for Full-text Title Database**
 - Developers are responsible for design, construct and implement the software system. They are also responsible for helping the customer and provide some training to the system administrator and software maintainer during the transition phase of the system.
 - Relationship with result chain: We are the people who develop the result chain.
- **System Administrator: Greg Fleming**
 - Greg Fleming will be the one who is assigned by ISD and responsible for updating the system database. He needs to observe the vendors' databases. Once vendors update their databases, our unified database should also be updated.
 - Relationship with result chain: He is responsible for ensuring the system to provide the updated information to users so that they can actually get the expected benefits.
- **Software Maintainer: ISD employee**
 - ISD will assign a person to maintain the system. He/she is responsible for maintaining the software code and crash recovery of the system.
 - Relationship with result chain: He/she is responsible for ensuring the system stability and availability so that the expected benefits can be realized.
- **Data Source: Vendors**
 - Vendors are the commercial organizations which provide databases containing the full-text journals and subscription service to others.
 - Relationship with result chain: The initiative of the result chain. We assume the vendors' databases are searchable. Our unified Fulltext Title Database will be generated from their databases

OCD 2.3 System Boundary & Environment

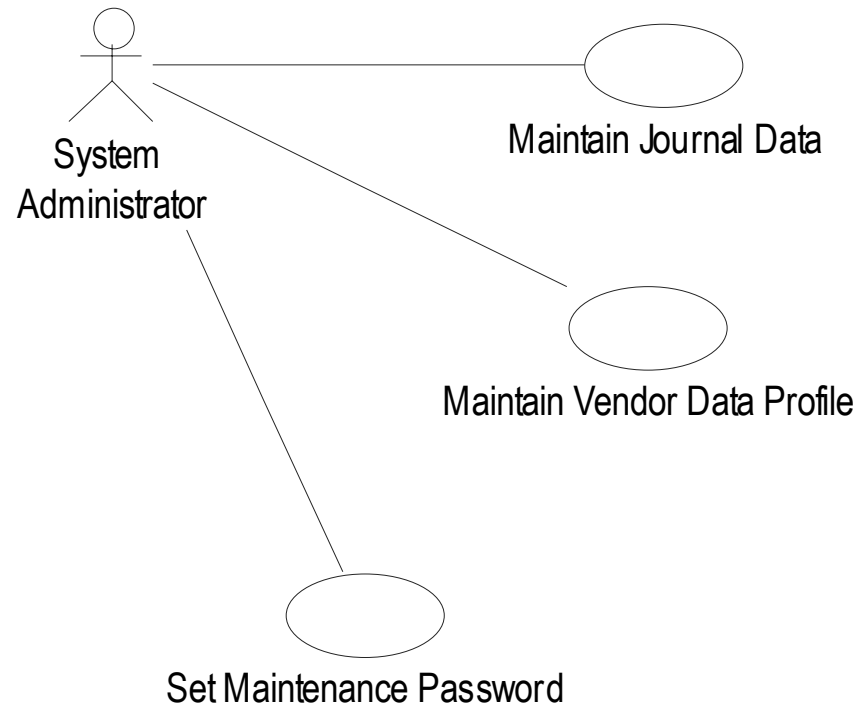
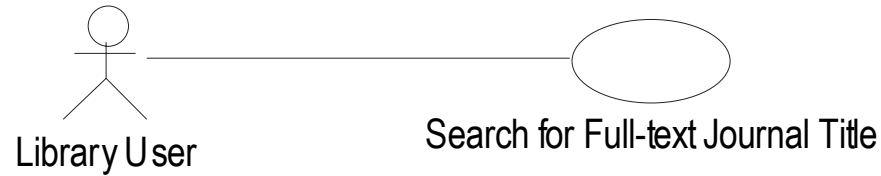




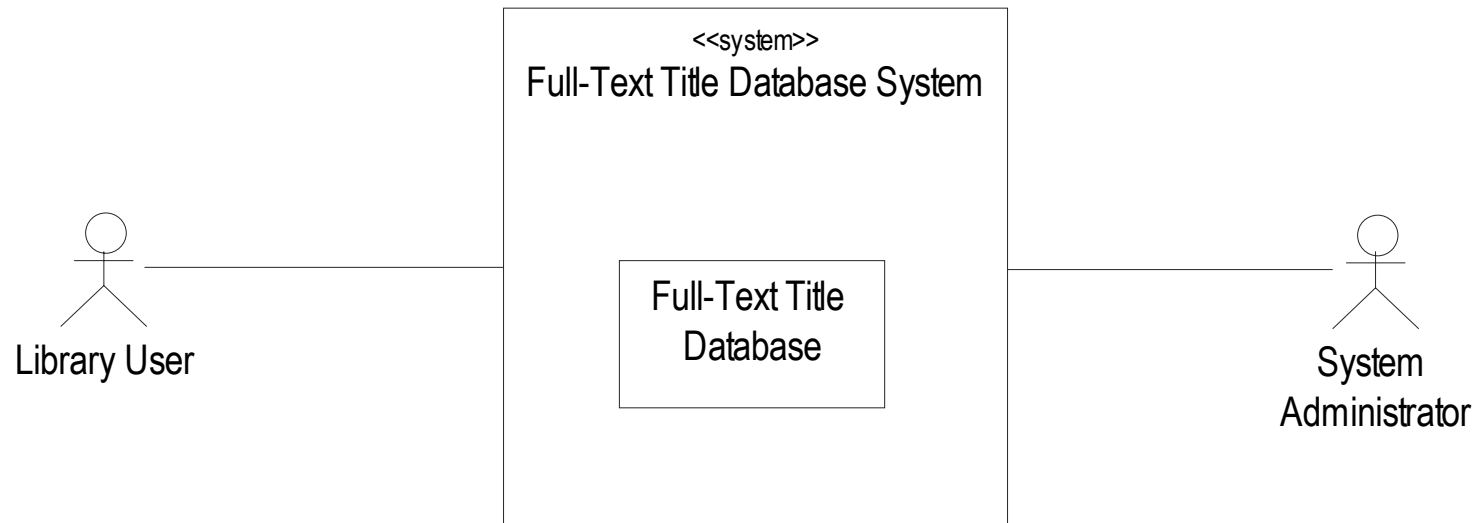
OCD 3.2 Organization Goals

- **OG-1: Fast Information Access**
 - Provide a faster way to access information
- **OG-2: Reduced workload**
 - Allowing the librarians to focus their time & energy on those patrons that really need help
- **OG-3: User friendly environment**
- **OG-4: Enhance the library collection**
 - Enhance the library collection by subscribing to vendors' full-text journal databases
- **OG-5: Ensure the Integrity of Information**
 - Provide a way to ensure the integrity of the information and protect the contents of the system
- **OG-6: Make the Resource Available to Distributed Users**
 - Users may be locate at any place and want to access the resources

OCD 4.3 Capabilities

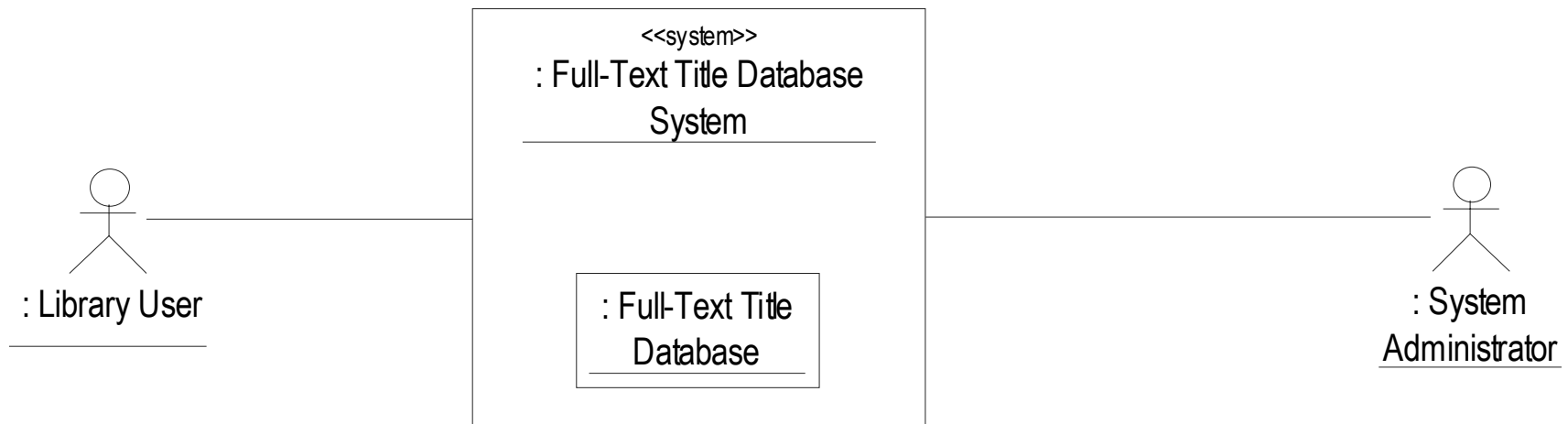


OCD 4.5 Proposed System



OCD 4.5 Proposed System (cont.)

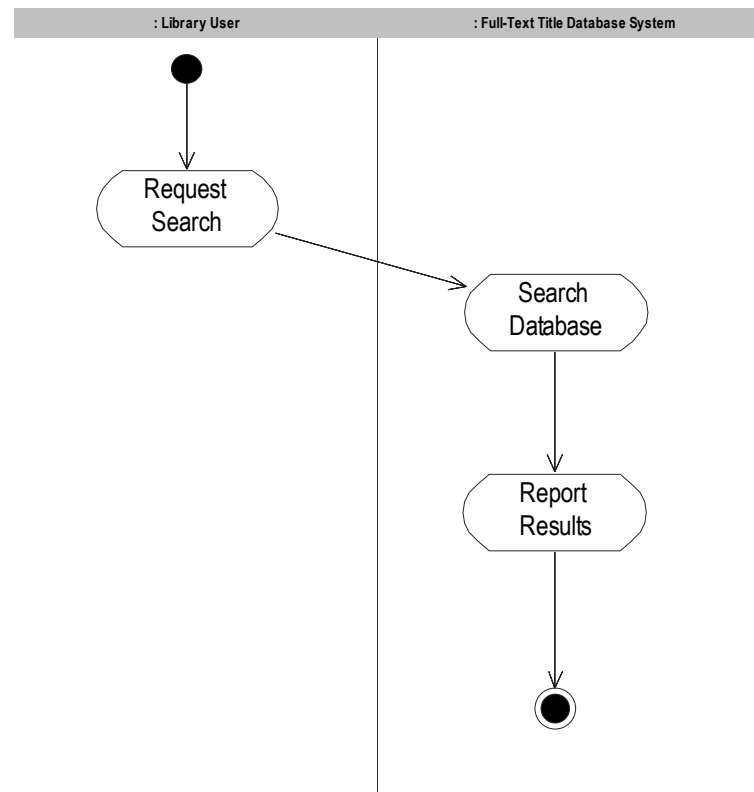
Typical Configuration



- For this system,
 - There's really only 1 configuration
 - Doesn't add much information

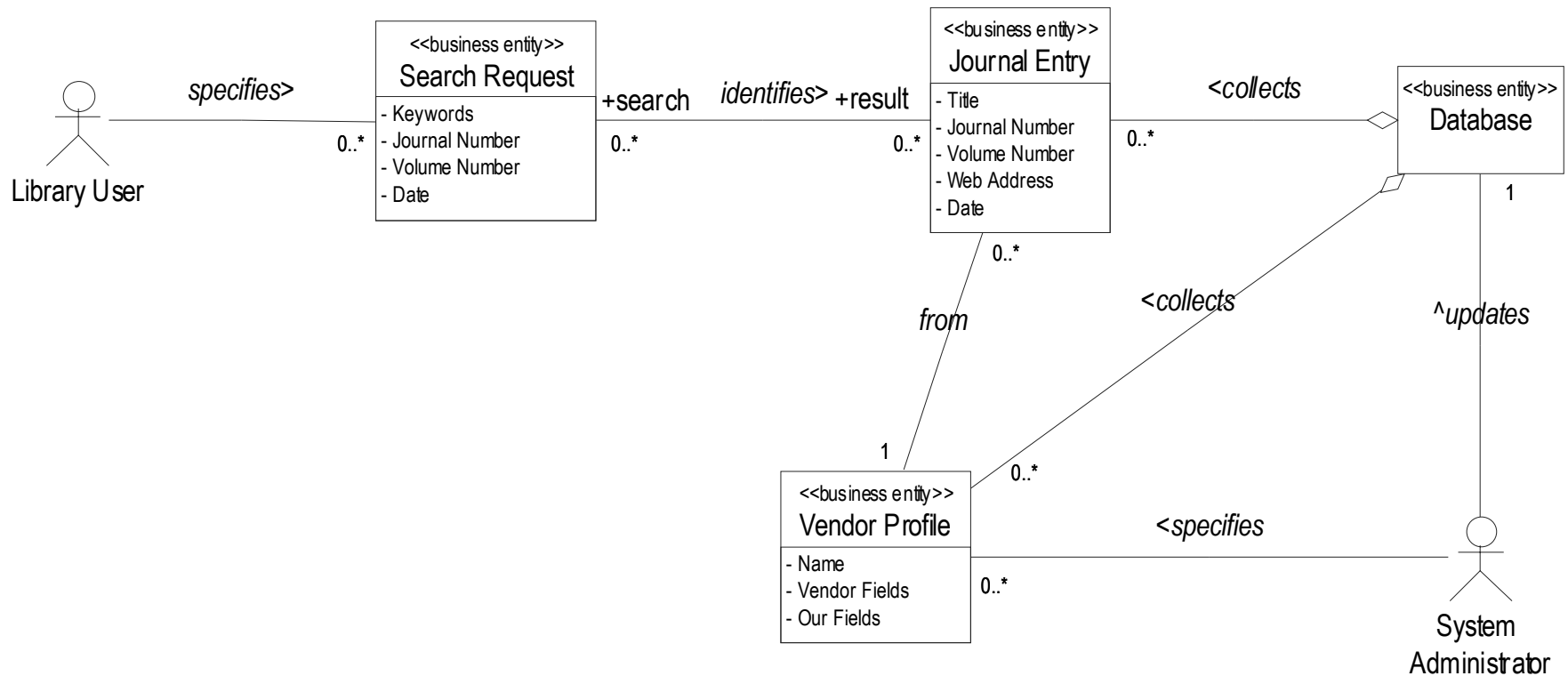
OCD 4.5.1 Proposed Activities

Search for Full-Text Journal Title



- Other activities not shown

OCD 4.5.2 Proposed Entities





Iteration Plan (Informal)

■ 2 Iterations

□ Capabilities in Iteration#1

- Search for Full-Text Journal Title
- Maintain (Enter) Journal Data

□ Additional Capabilities in Iteration#2

- Maintain Vendor Data Profiles
- Maintain (Change) Journal Data
- Set Maintenance Password



Outline

- Key Concepts
- Process Overview
- Example Project Description
- **Process by Example**

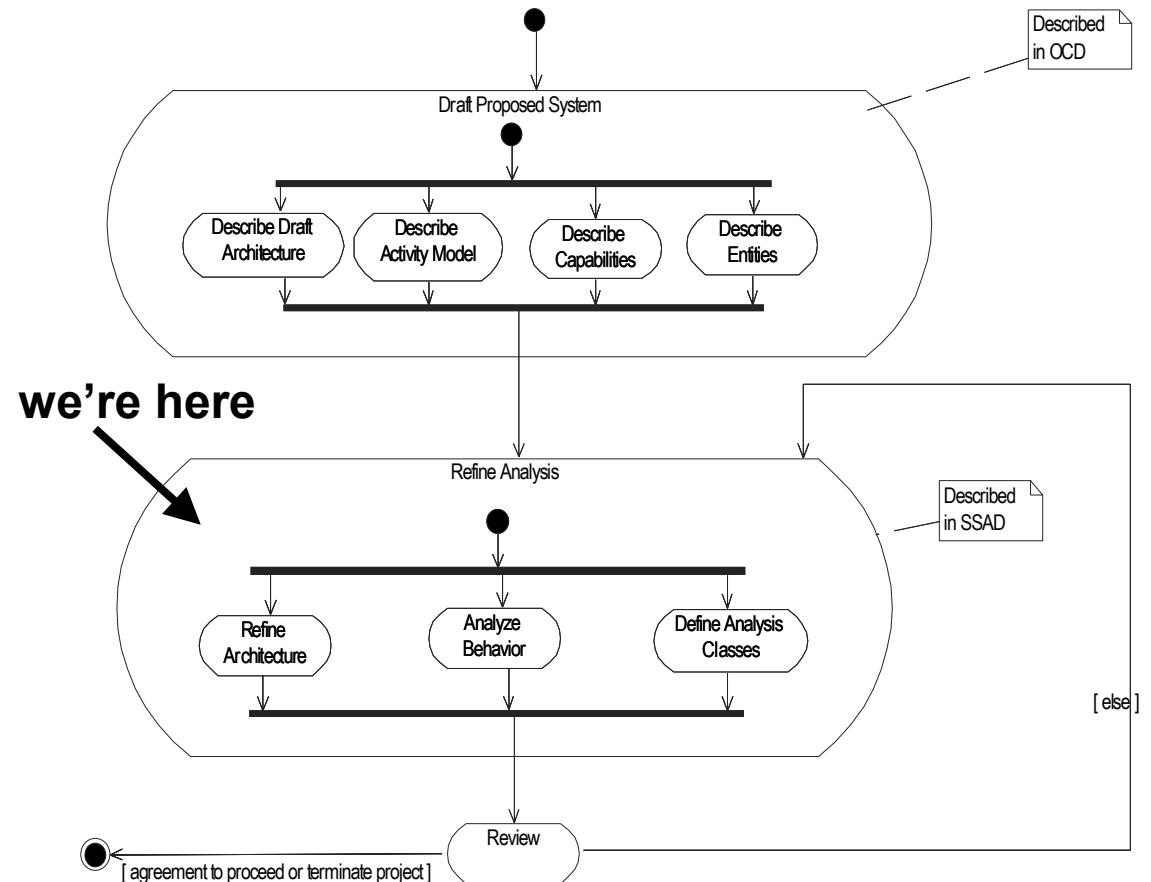
System Analysis Process Overview

- Sub-activities not order

- i.e. can start anywhere

- Why not?

- Iterative Process





Analyze Behavior – LCO

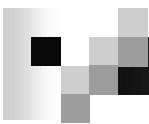
- Purpose:
 - Improve understanding of behavior requirements
- Inputs:
 - Proposed System Context
 - Activity Model
 - Capability Model
- Artifacts:
 - Use–cases Model
 - Use Diagram(s)
 - Descriptions of each use–case
 - See Form



Analyze Behavior – LCO

Guide for Creation of Use–Case Model

- For each capability in Capability Model
 - If abstraction of more specify behavior
 - Create use–case for each specific behavior
 - e.g. Manage Vendor Profile => Add, Edit, Delete Profile?
 - Show generalization relation in Behavior Classification Model
 - Otherwise, create use–case for capability
- All actors that participate in each use–case
- For all actors,
 - Verify that have identified capabilities to be included in current iteration



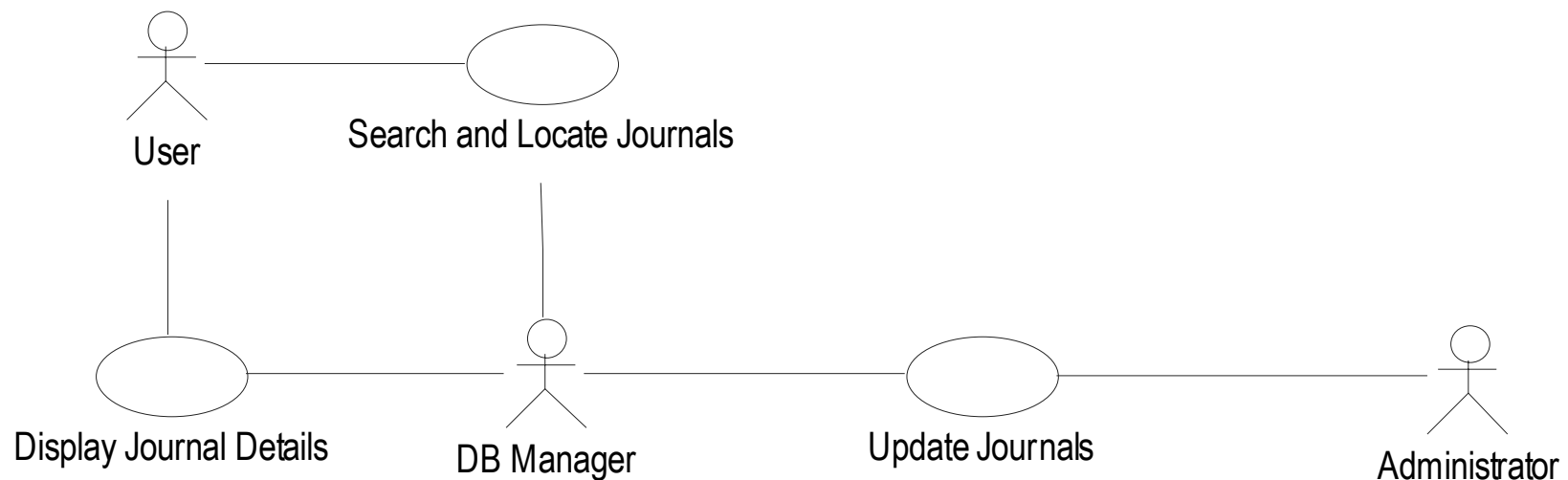
Analyze Behavior – LCO

Guide for Creation of Use–Case Model (cont.)

- Define relations
 - For each use–case that an actor participates in
 - Draw bi-directional association between actor & use–case
 - If actor specializes another actor
 - Draw generalization from specialized actor to general actor
 - If use-case extends other use-case
 - Draw extends relation
- Describe each actor
- Describe each use–case
 - Form
 - Sequence Diagrams (optional at LCO)
- Identify common behavior in different use–cases
 - Create use–case for common behavior
 - Draw include relation from all behaviors that share this common behavior to the use–case for common behavior

Analyze Behavior – LCO

Use-Case Diagram Example





Analyze Behavior – LCO

Use-Case Description Example 1

Use-Case Name	Full-text Journal Title Search
Abstract	
Purpose	To allow a user to search for journals to which USC Libraries subscribes by keyword
Actors	User, DB Manager
Importance	Primary
Requirements	Full-text Journal Title Search
Risks	
High-Risk?	No
Architecturally Significant?	Yes
Development Status	Draft LCO
Overview	User enters search criteria and system returns lists of journals matching criteria
User Interface	
Pre-conditions	Database has been initialized
Post-conditions	Displayed List of all the complete journal titles containing the user's search criteria
Includes	
Extension Points	



Analyze Behavior – LCO

Use-Case Description Example 1 (cont.)

■ Typical Course of Action

Seq. #	Actor Actions	System Response
1.	User requests search	
2.		Displays search page
2	User enters search criteria	
3		Queries the database asking for journal titles that match the user's search criteria
4		Displaying the journal list in search result page




Analyze Behavior – LCO

Use-Case Description Example 1 (cont.)

- Alternate Course of Action: No results match search criterion

Seq. #	Actor Actions	System Response
4.		Display error page that asks user to search again

- Exceptional Course of Action: None



Analyze Behavior

Exit Criteria for Use-Case Descriptions

- LCO
 - High-risk, architecturally significant, or particularly complex use-case
 - Include detailed courses of actions with exception & alternate courses of action identified
 - Other use-cases need only include high-level overview
- LCA
 - High-risk, architecturally significant, or particularly complex use-case
 - Designed using one or more Sequence Diagrams
 - Interaction Model
 - Other use-cases need only include high-level overview
- IOC
 - All use-cases should be designed using one or more Sequence Diagrams