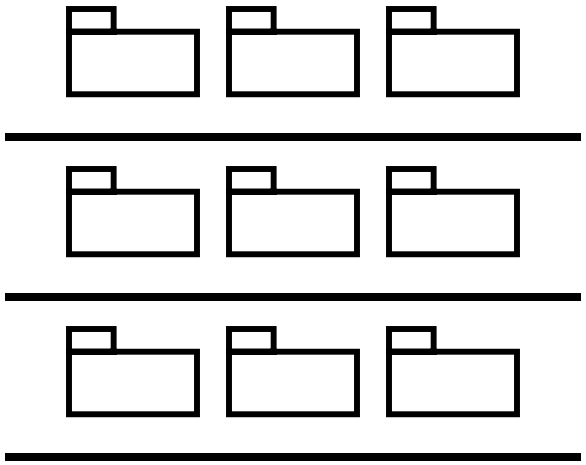
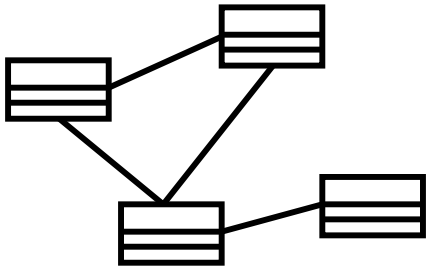
Architectural & Use Case Analysis

Presented by
William Ebenezer
ITC Infotech India Ltd.

Architectural & Use Case Analysis





Architecture Definition

- **The RUP defines architecture as**

“the highest level concept of a system in its environment. The architecture of a software system (at a given point in time) is its organization or structure of significant components interacting through interfaces, those components being composed of successively smaller components and interfaces.”

More Definitions

- **Architectural investigation**
  - Identify functional and non-functional requirements that have significant impact on the system design.

- **Architectural design**
  - Resolve the forces and requirements in the design of the software.

So What is it …..

Architectural Investigation

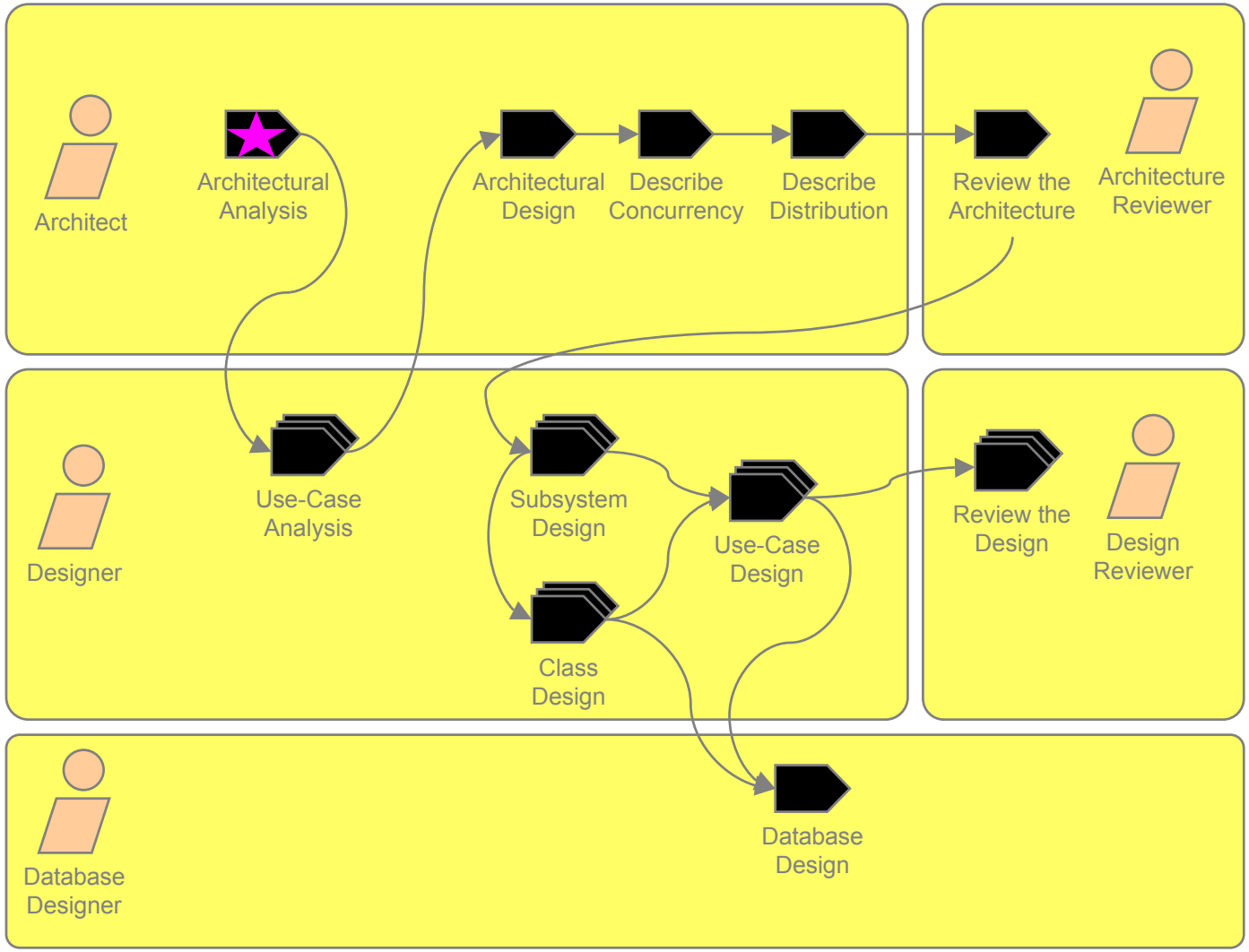
+

Architectural Design

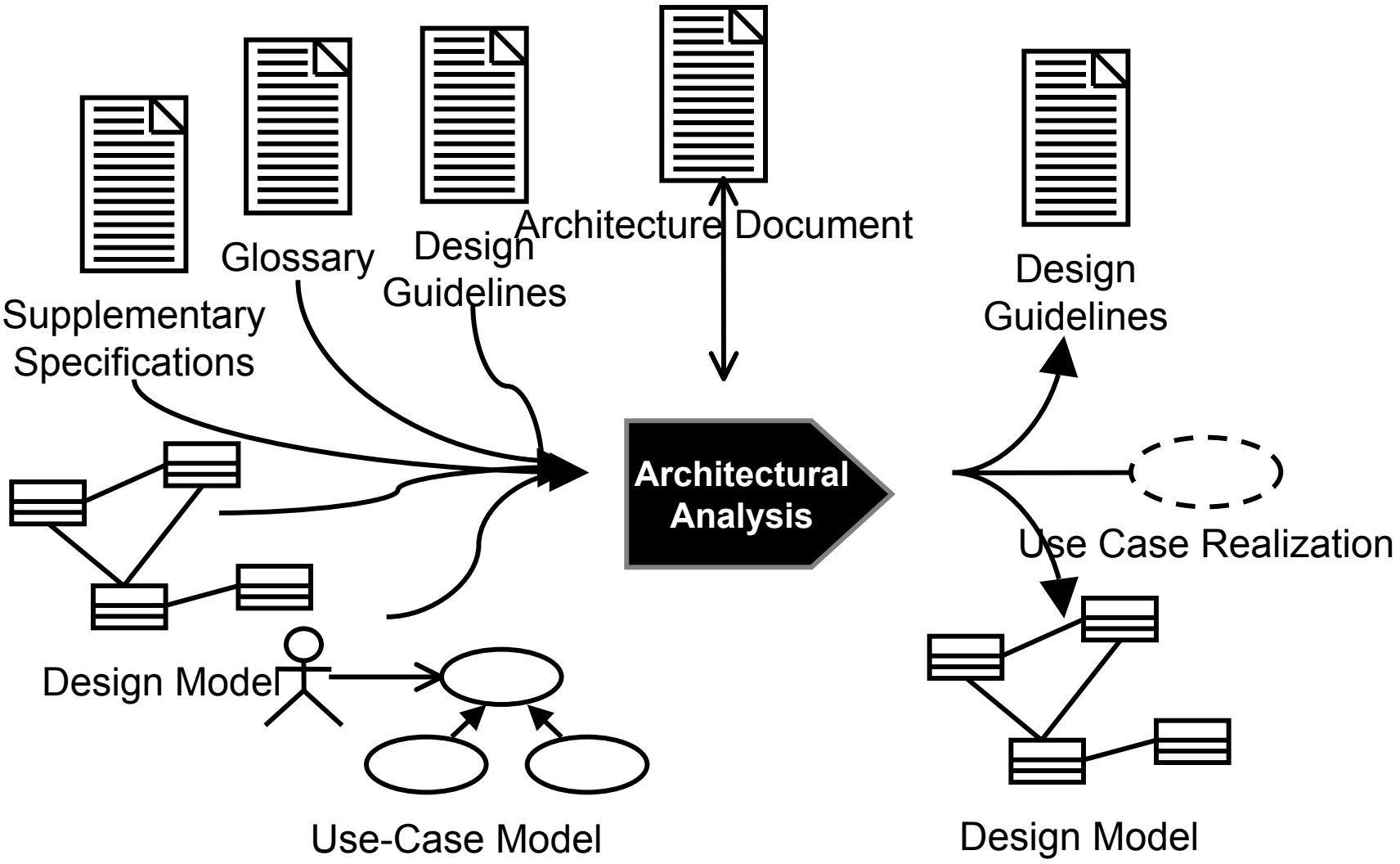
=

Architectural Analysis

So Where Are We?



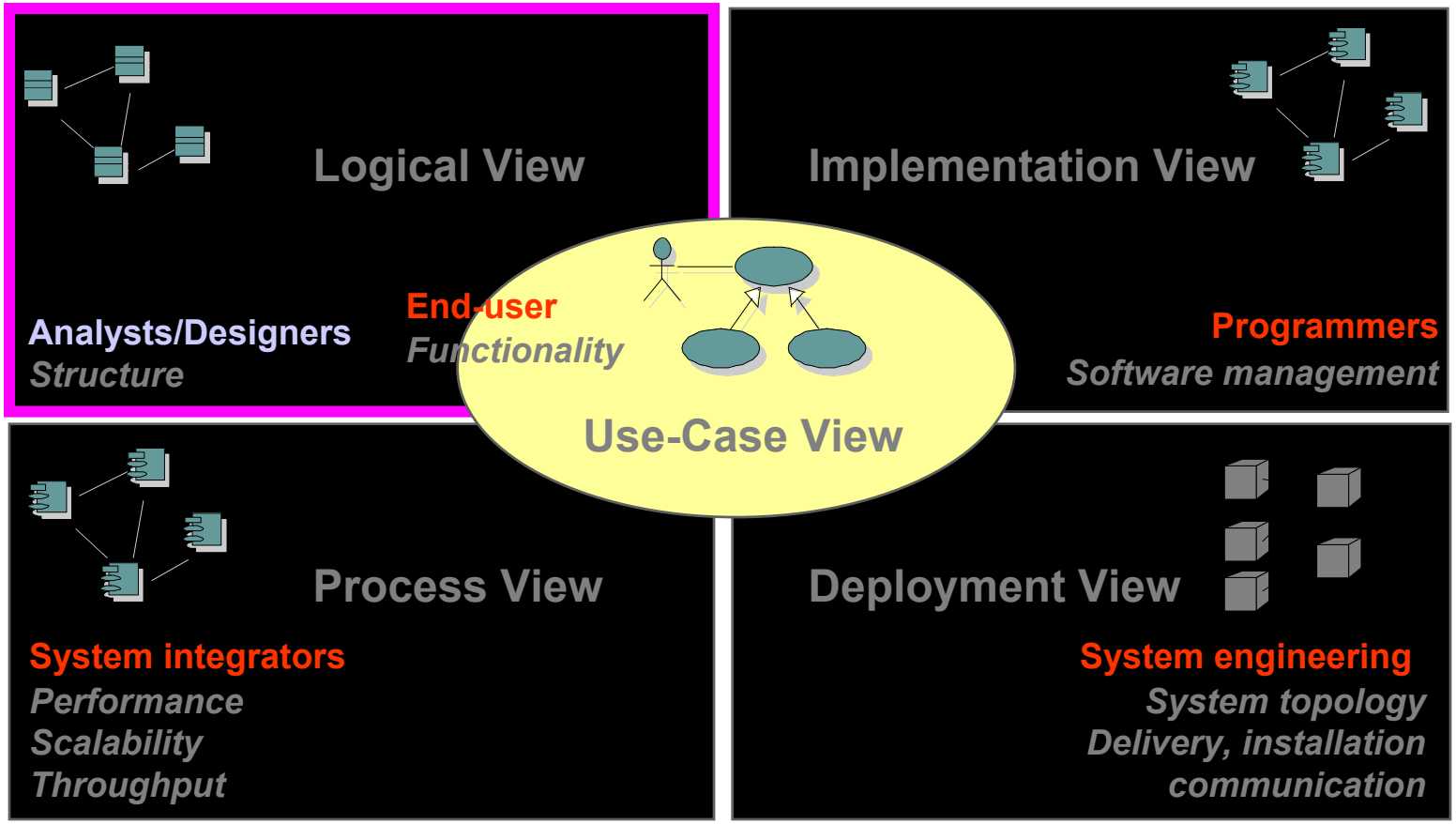
Architectural Analysis Overview



Architectural Analysis Topics

- **Key Architectural Analysis Concepts**
- Modeling Conventions
- Analysis Mechanisms
- Key System Concepts
- Architectural Patterns

Architecture: The “4+1 View” Model



Architectural Analysis Topics

- Key Architectural Analysis Concepts
- **Modeling Conventions**
- Analysis Mechanisms
- Key System Concepts
- Architectural Patterns

Modeling Conventions

- **What Are They?**
  - What diagrams and modeling elements to use
  - Rules for the use of modeling elements and diagrams
  - Naming conventions

Architectural Analysis Topics

- Key Architectural Analysis Concepts
- Modeling Conventions
- **Analysis Mechanisms**
- Key System Concepts
- Architectural Patterns

What Are They?

- **Architectural Mechanisms**
  - Analysis Mechanisms (conceptual)
  - Design Mechanisms (concrete)
  - Implementation Mechanisms (actual)

Analysis Mechanisms	Design Mechanisms	Implementation Mechanisms
Persistence	RDBMS	JDBC to SQL Server
Persistence	OODBMS	Ontos, Versant, ObjectStore
Distribution (of objects)	Remote Method Invocation	Java 1.4 from Sun Microsystems.
Distribution (of objects)	Web services	Microsoft .NET
Legacy Interface	CICS application on MVS mainframe	SNA 3270, LU 6.2 APPC

Sample Analysis Mechanisms

- Persistency
- Communication (IPC and RPC)
- Message routing
- Distribution
- Transaction management
- Process control and synchronization (resource contention)
- Information exchange, format conversion
- Security
- Error detection / handling / reporting
- Redundancy
- Legacy Interface

Analysis Mechanism Characteristics

- **Persistency**
  - Granularity
  - Volume
  - Duration
  - Access mechanism
  - Access frequency (creation/deletion, update, read)
  - Reliability
- **Security**
  - Data granularity
  - User granularity
  - Security rules
  - Privilege types

Describing Analysis Mechanisms

- **Collect all analysis mechanisms in a list**
- **Draw a map of the client classes to the analysis mechanisms**

Analysis Class	Analysis Mechanism(s)

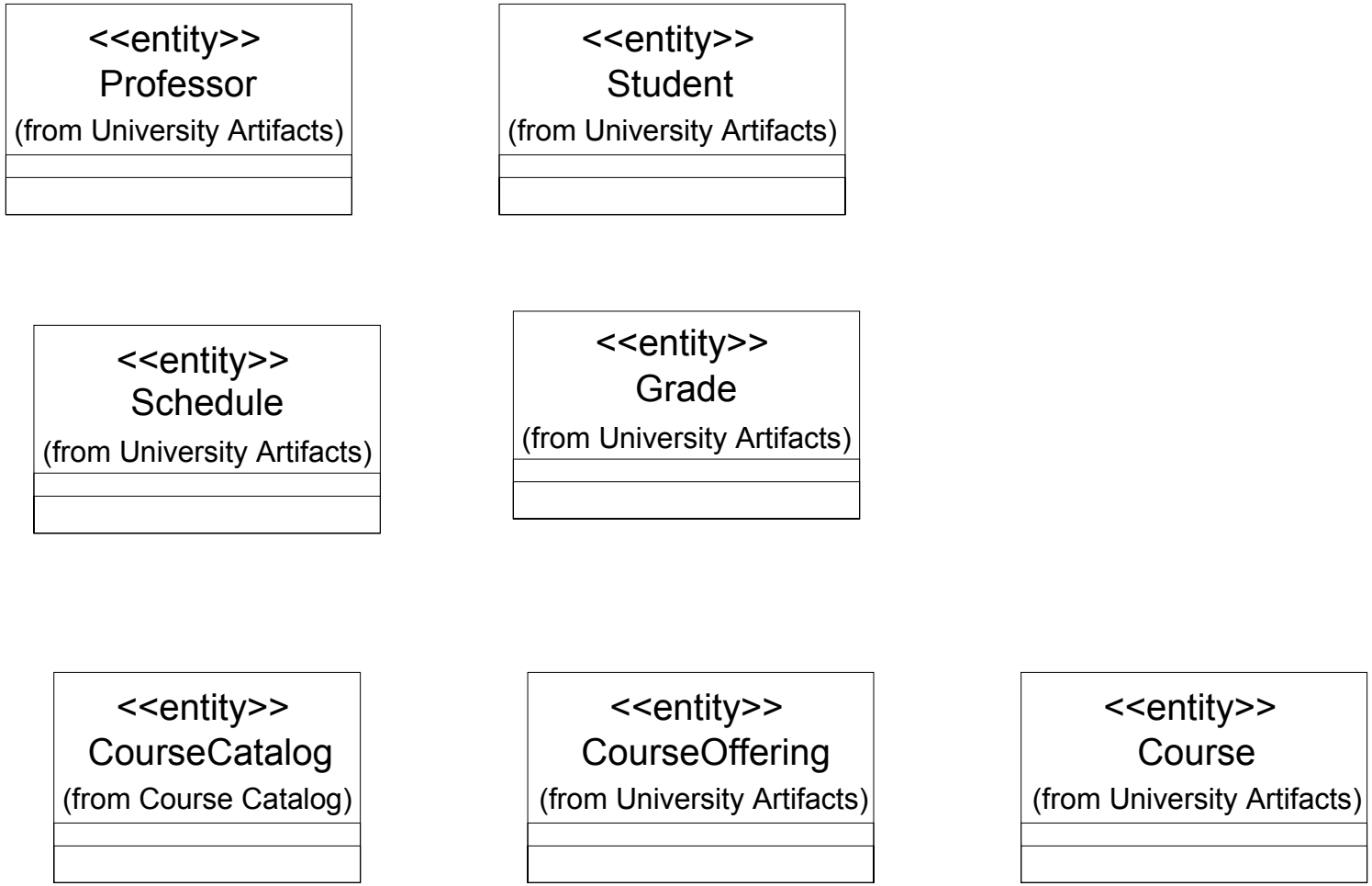
Architectural Analysis Topics

- Key Architectural Analysis Concepts
- Modeling Conventions
- Analysis Mechanisms
- **Key System Concepts**
- Architectural Patterns

Identify Key Concepts

- **Define preliminary entity analysis classes**
  - Domain knowledge
  - Requirements
  - Glossary
  - Domain/Business Model (if exists)

Example: Key Concepts



Architectural Analysis Topics

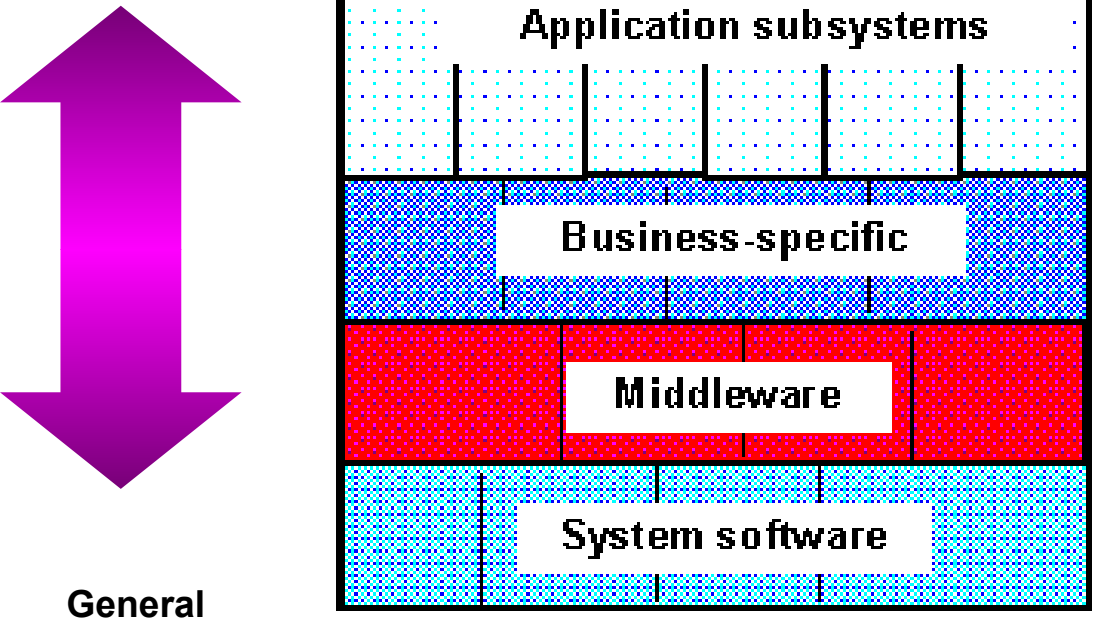
- Key Architectural Analysis Concepts
- Modeling Conventions
- Analysis Mechanisms
- Key System Concepts
- **Architectural Patterns**

Architectural Patterns

- **Layers**
- **Model-view-controller (M-V-C)**
- **Pipes and filters**
- **Blackboard**

Typical Layering Approach

Specific functionality



General functionality

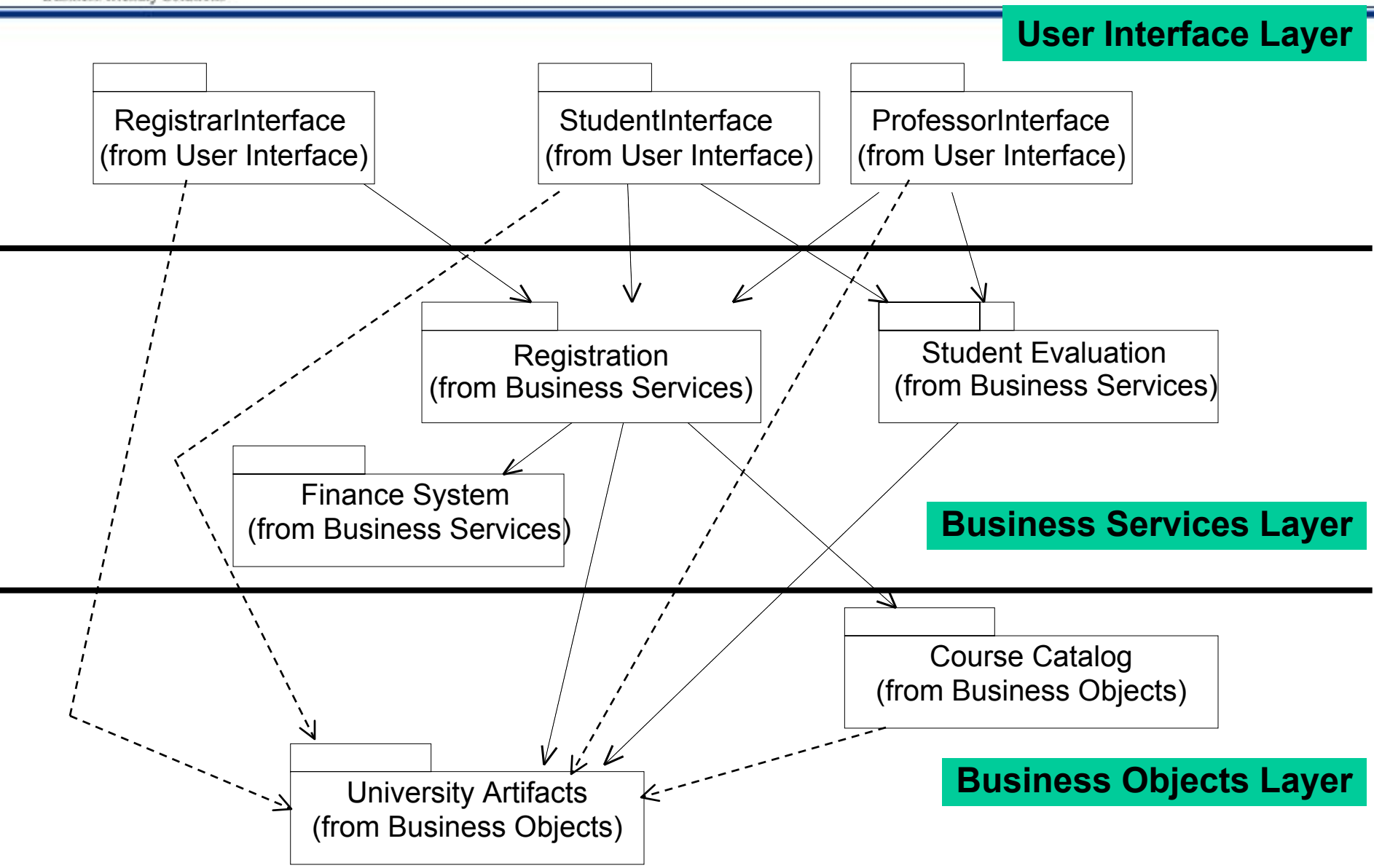
Distinct application subsystem that make up an application - contains the value adding software developed by the organization.

Business specific - contains a number of reusable sybsystems specific to the type of business.

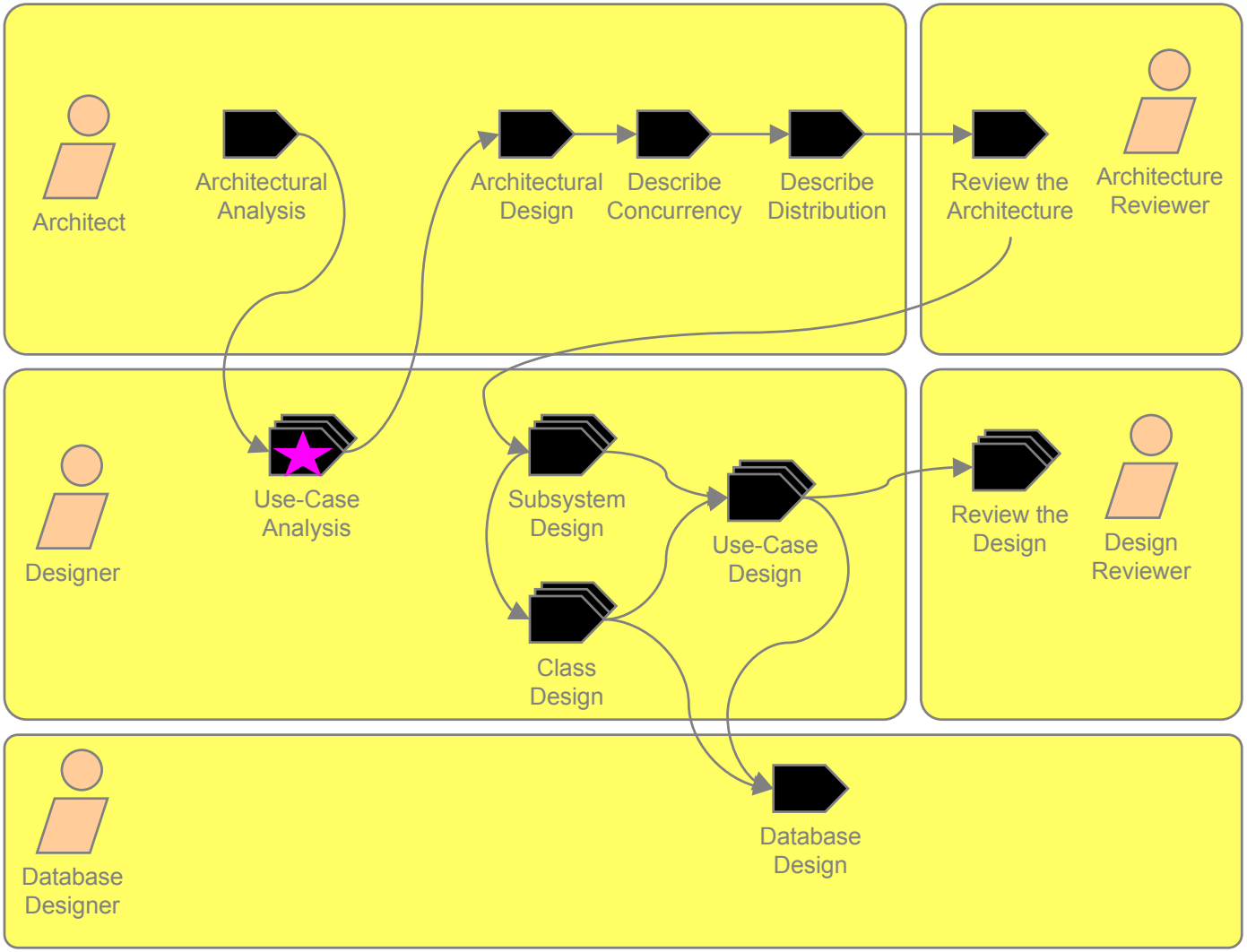
Middleware - offers subsystems for utility classes and platform-independent services for distributed object computing in heterogeneous environments and so on.

System software - contains the software for the actual infrastructure such as operating systems, interfaces to specific hardware, device drivers and so on.

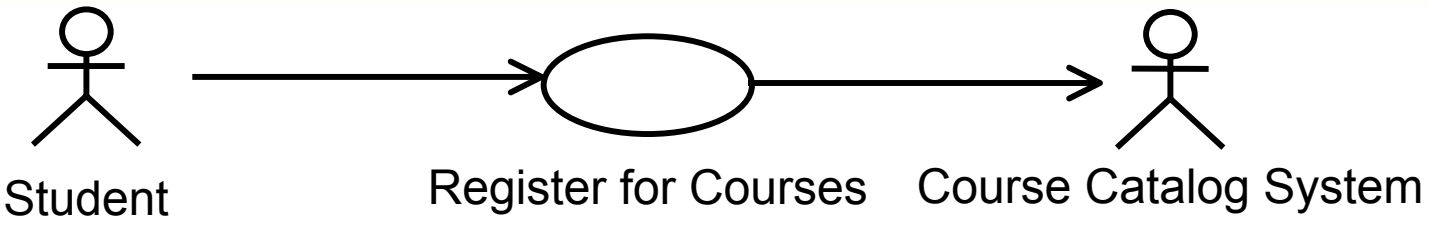
Organization of the Model



Use Case Analysis



Sample Use Case

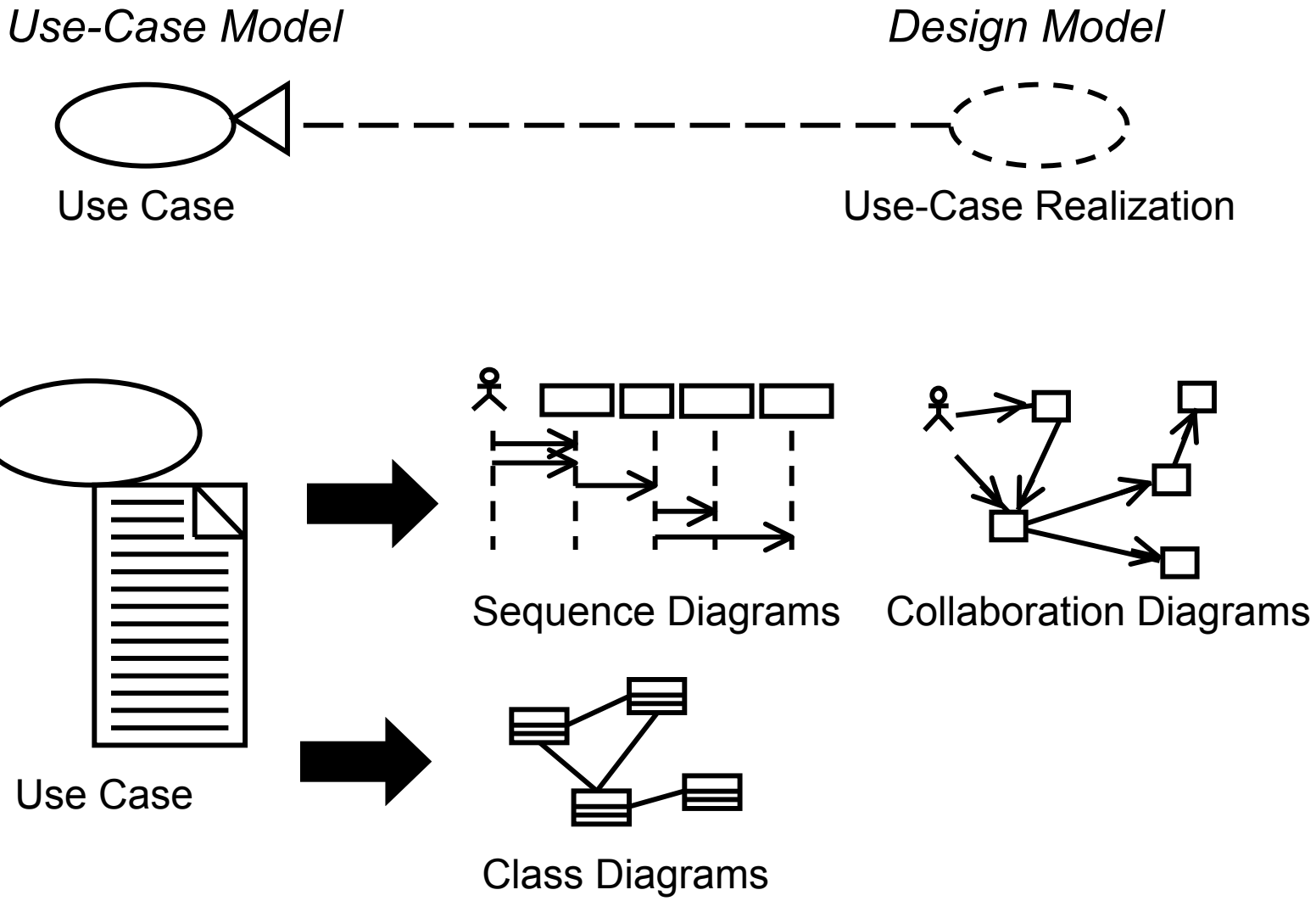


Use-Case Model

Use Case Analysis Activity

- **Create an Use-Case Realization**
- Supplement the Use-Case descriptions
- Find Analysis Classes from Use-Case Behavior
- Distribute Behavior to Analysis Classes

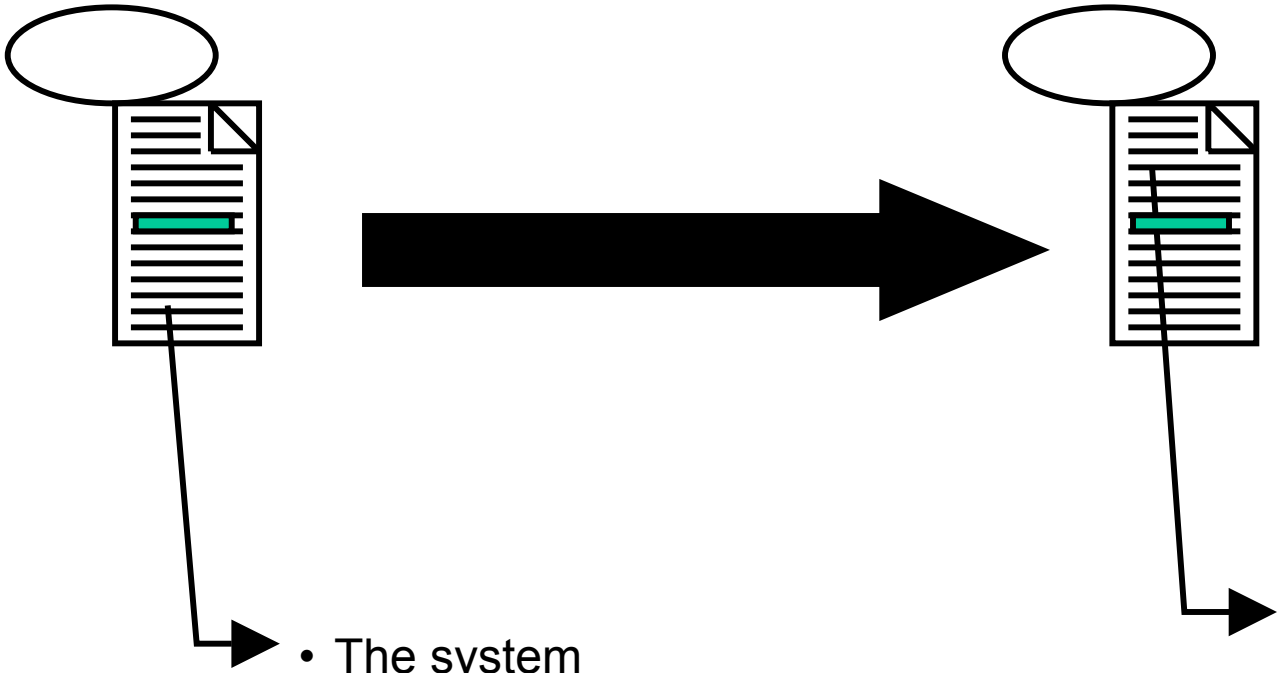
Use-Case Realization



Use Case Analysis Activity

- Create an Use-Case Realization
- **Supplement the Use-Case descriptions**
- Find Analysis Classes from Use-Case Behavior
- Distribute Behavior to Analysis Classes

Supplement Use-Case Description



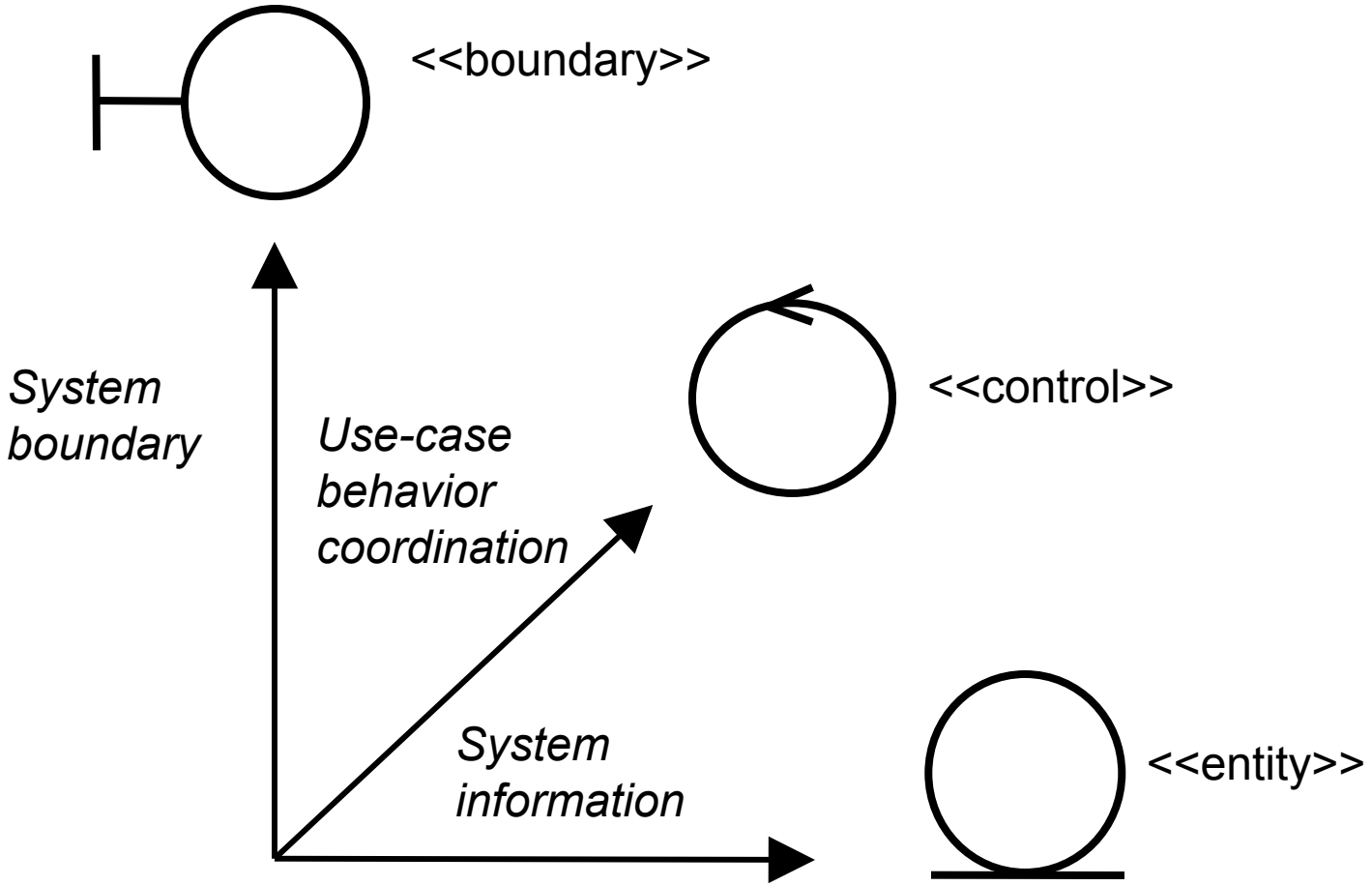
- The system displays a list of course offerings.

- The system retrieves and displays a list of current course offerings from the course catalog legacy database.

Use Case Analysis Activity

- Create an Use-Case Realization
- Supplement the Use-Case descriptions
- **Find Analysis Classes from Use-Case Behavior**
- Distribute Behavior to Analysis Classes

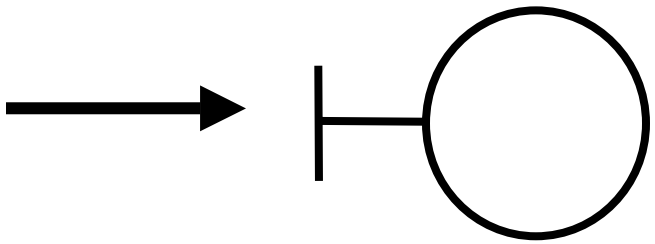
Analysis Classes



Boundary Classes

- **Intermediates between the interface and something outside the system**
- **Several Types**
  - User interface classes
  - System interface classes
  - Device interface classes
- **One boundary class per actor/use case pair**

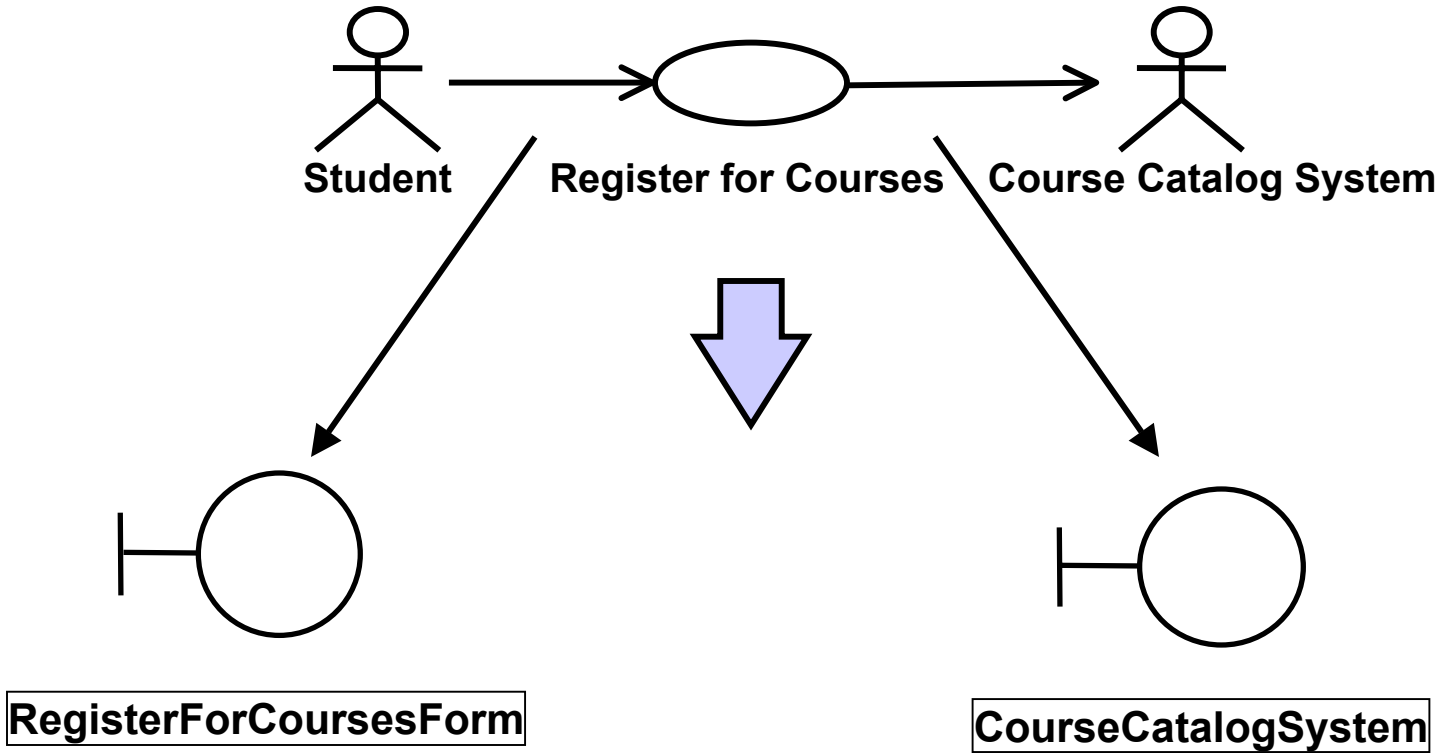
Analysis class stereotype



Environment Dependent

Finding Boundary Classes

- **One boundary class per actor/use case pair**



Entity Classes

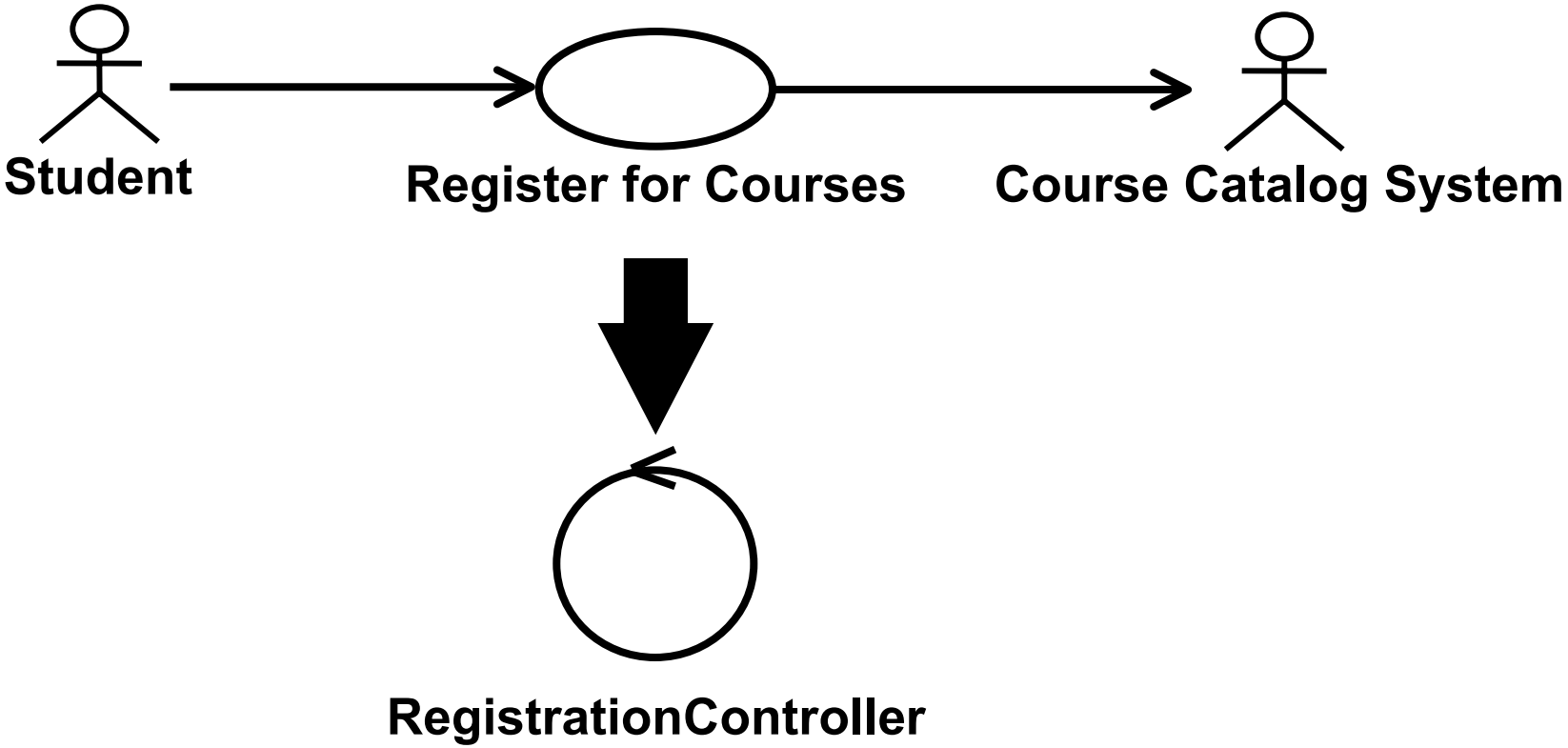
- Typically store + manage data
- Represent key concepts
- Show logical data structure
- Usually not specific to one use case
- May model real world objects

Control Classes

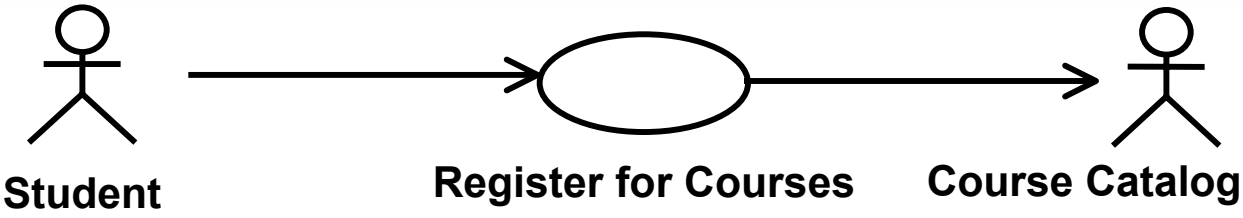
- Use-case behavior coordinator
- Interacts with several entity classes
- Effectively decouples boundary and entity classes
- One control class per use case

Finding Control Classes

- One control class per use case

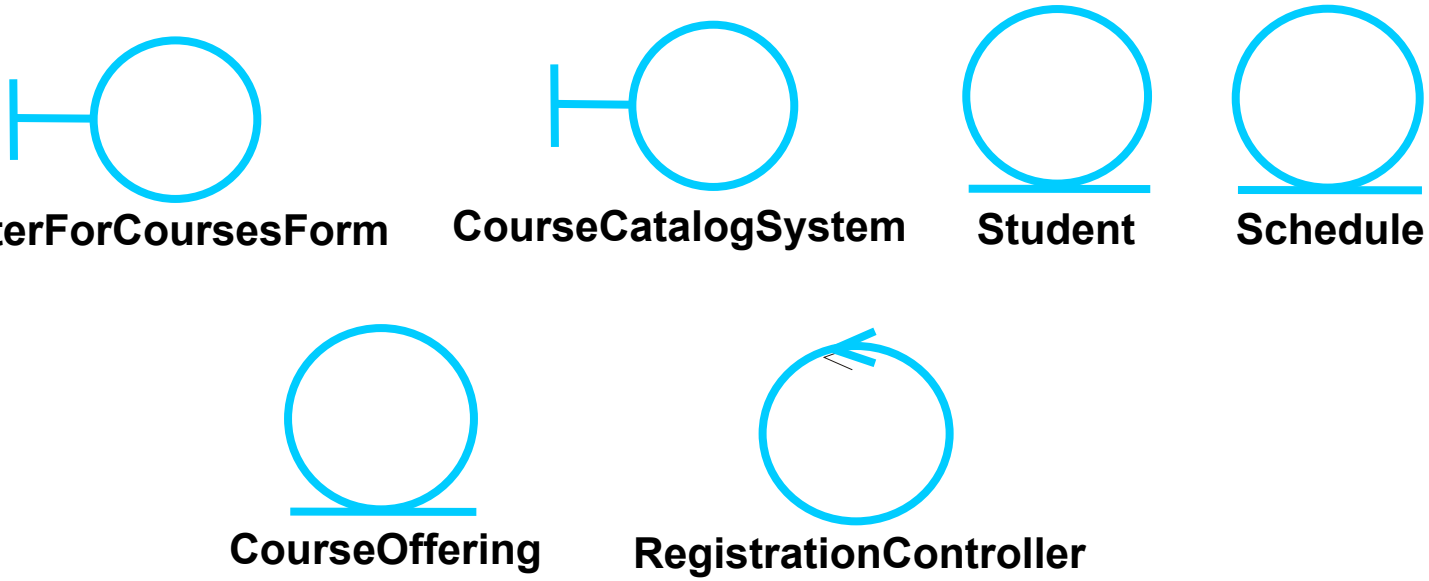


Summary: Analysis Classes



Use-Case Model

Design Model



Map from Use-Case Analysis

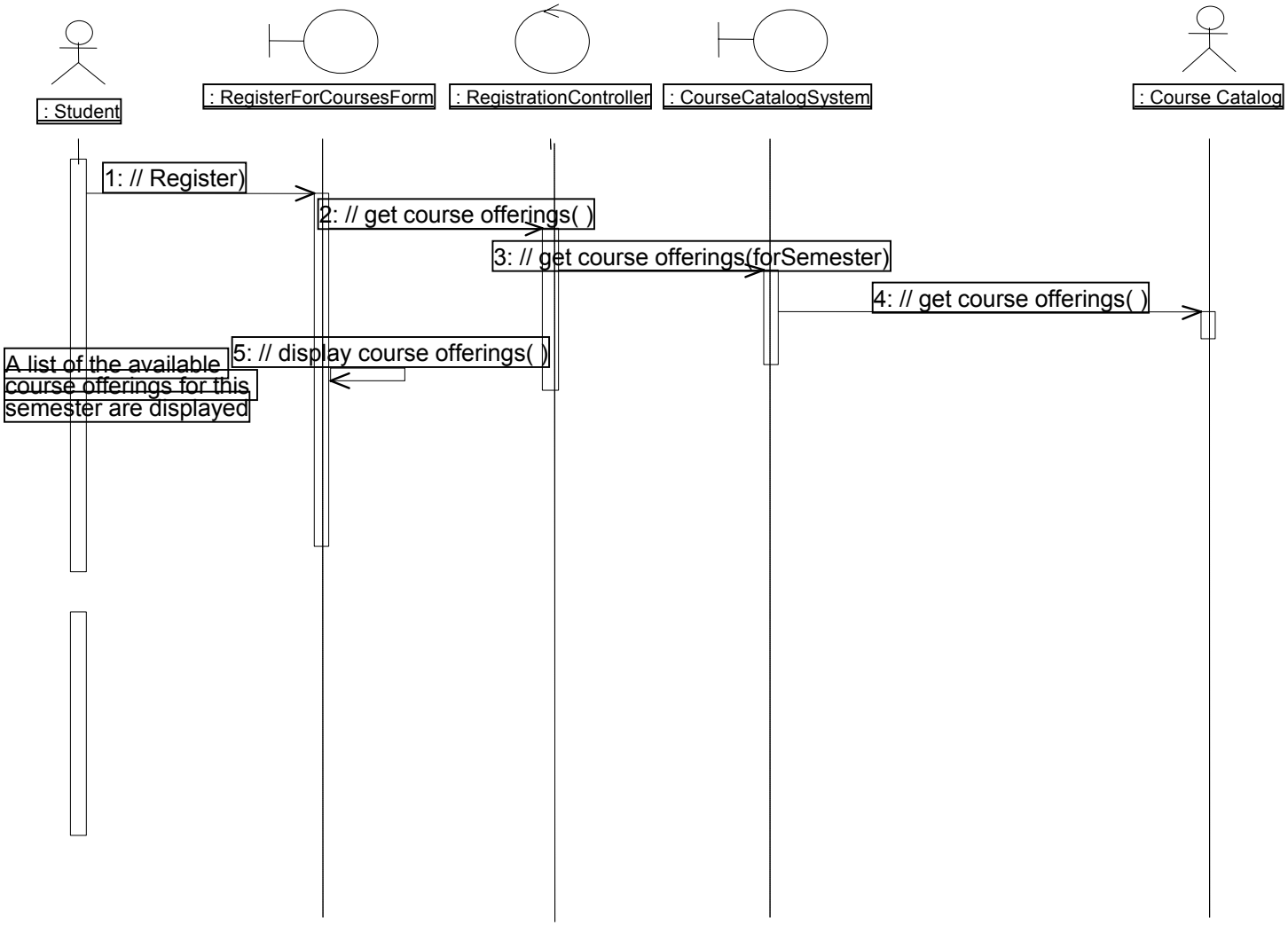
- **Map Analysis Classes to Architectural Mechanisms**

Analysis Class	Analysis Mechanism(s)
Student	Persistency, *Security*
Schedule	Persistency, *Security*
CourseOffering	Persistency, Legacy Interface
Course	Persistency, Legacy Interface
RegistrationController	Distribution

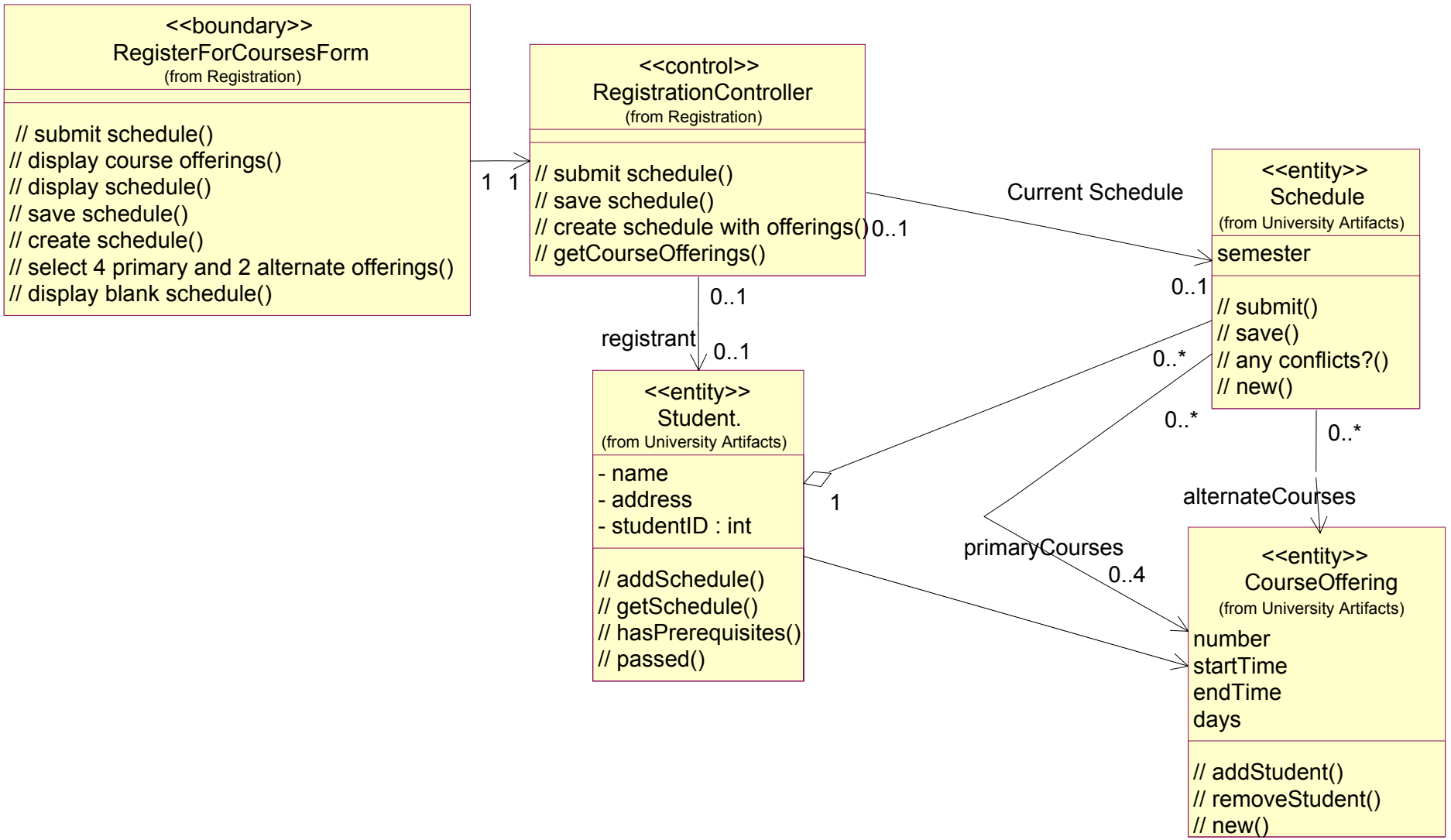
Use Case Analysis Activity

- Create an Use-Case Realization
- Supplement the Use-Case descriptions
- Find Analysis Classes from Use-Case Behavior
- **Distribute Behavior to Analysis Classes**

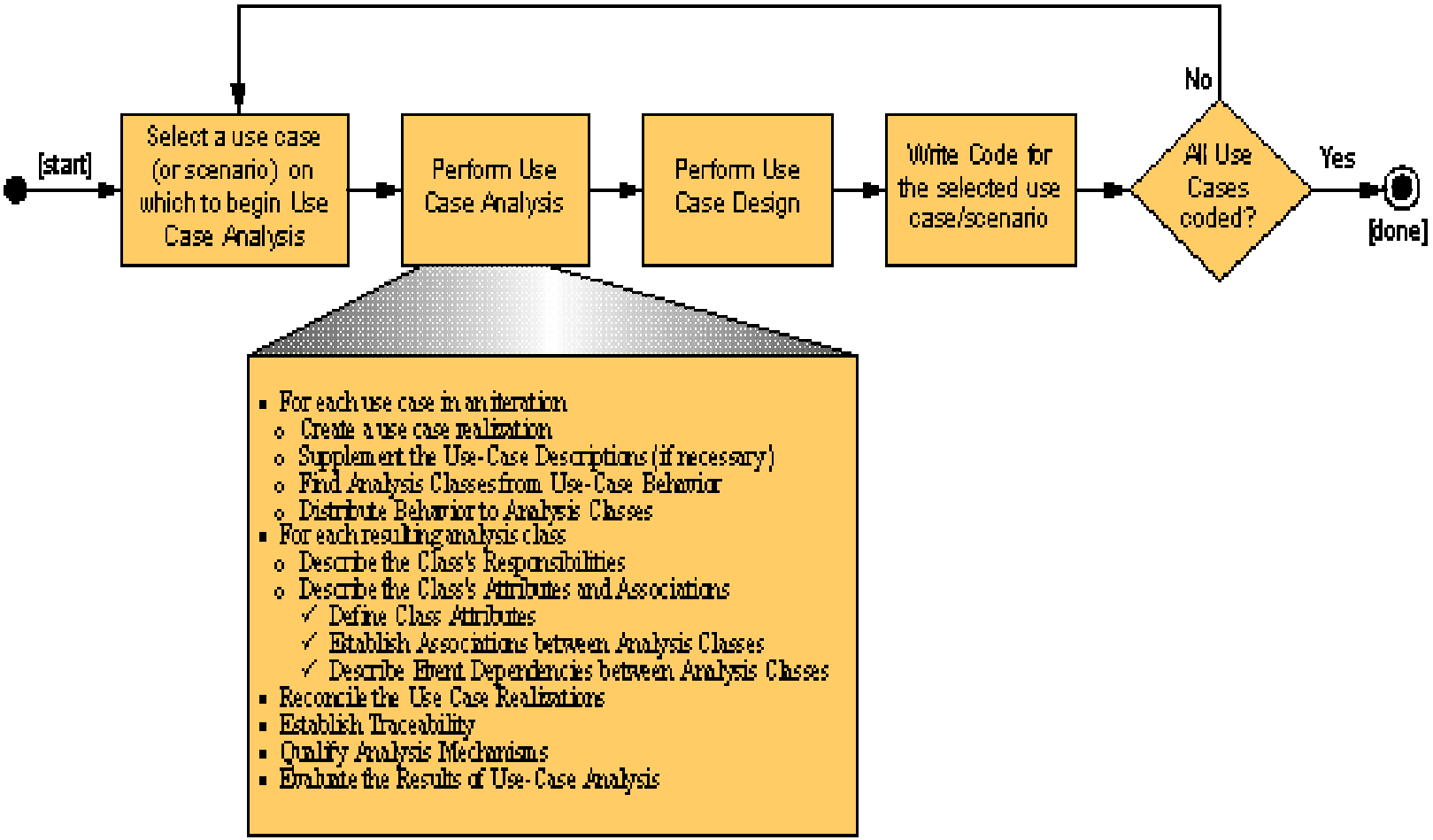
Distribute Behavior



Distribute Behavior



Road Map


[start]
Select a use case (or scenario) on which to begin Use Case Analysis
Perform Use Case Analysis
Perform Use Case Design
Write Code for the selected use case/scenario
All Use Cases coded?
No
Yes
[done]
For each use case in an iteration
Create a use case realization
Supplement the Use-Case Descriptions (if necessary)
Find Analysis Classes from Use-Case Behavior
Distribute Behavior to Analysis Classes
For each resulting analysis class
Describe the Class's Responsibilities
Describe the Class's Attributes and Associations
✓ Define Class Attributes
✓ Establish Associations between Analysis Classes
✓ Describe Event Dependencies between Analysis Classes
Reconcile the Use Case Realizations
Establish Traceability
Qualify Analysis Mechanisms
Evaluate the Results of Use-Case Analysis

Thank You

william.ebenezer@itcinfotech.com