Getting Started

If architects had to work like programmers

Dear Architect,

Please design and build me a house. I am not sure of what I need, so you you should use your discretion.

My house should have between two and forty-five bedrooms. Just make sure the plans are such that the bedrooms can be easily added or deleted.

When you bring the blueprints to me, I will make the final decision of what I want. Also, bring me the cost breakdown for each configuration so that I can arbitrarily pick one.

Keep in mind that the house I ultimately choose must cost less than the one I am currently living in. Make sure, however, that you correct all the deficiencies that exist in my current house (the floor of my kitchen vibrates when I walk across it, and the walls don't have nearly enough insulation in them).

As you design, also keep in mind that I want to keep yearly maintenance costs as low as possible. This should mean the incorporation of extra-cost features like aluminum, vinyl, or composite siding. (If you choose not to specify aluminum, be prepared to explain your decision in detail.)

Please take care that modern design practices and the latest materials are used in construction of the house, as I want it to be a showplace for the most up-to-date ideas and methods. Be alerted, however, that kitchen should be designed to accommodate, among other things, my 1952 Gibson refrigerator.

Getting Started

If architects had to work like programmers

To insure that you are building the correct house for our entire family, make certain that you contact each of our children, and also our in-laws. My mother-in-law will have very strong feelings about how the house should be designed, since she visits us at least once a year. Make sure that you weigh all of these options carefully and come to the right decision. I, however, retain the right to overrule any choices that you make

Please don't bother me with small details right now. Your job is to develop the overall plans for the house: get the big picture. At this time, for example, it is not appropriate to be choosing the color of the carpet. However, keep in mind that my wife likes blue.

Also, do not worry at this time about acquiring the resources to build the house. Your first priority is to develop detailed plans and specifications. Once I approve these plans, however, I would expect the house to be under roof within 48 hours.

While you are designing this house specifically for me, keep in mind that sooner or later I will have to sell it to someone else. It therefore should have appeal to a wide variety of potential buyers. Please make sure before you finalize the plans that there is a consensus of the population in my area that they like the features this house has.

I advise you to run up and look at my neighbor's house he constructed last year. We like it a great deal. It has many features that we would also like in our new home, particularily the 75-foot swimming pool. With careful engineering, I believe that you can design this into our new house without impacting the final cost.

Getting Started

If architects had to work like programmers

Please prepare a complete set of blueprints. It is not necessary at this time to do the real design, since they will be used only for construction bids. Be advised, however, that you will be held accountable for any increase of construction costs as a result of later design changes.

You must be thrilled to be working on as an interesting project as this! To be able to use the latest techniques and materials and to be given such freedom in your designs is something that can't happen very often. Contact me as soon as possible with your complete ideas and plans.

PS: My wife has just told me that she disagrees with many of the instructions I've given you in this letter. As architect, it is your responsibility to resolve these differences. I have tried in the past and have been unable to accomplish this. If you can't handle this responsibility, I will have to find another architect.

PPS: Perhaps what I need is not a house at all, but a travel trailer. Please advise me as soon as possible if this is the case.

Enterprise Modeling

Why?

(Goal-oriented EM in Requirements Elicitation)

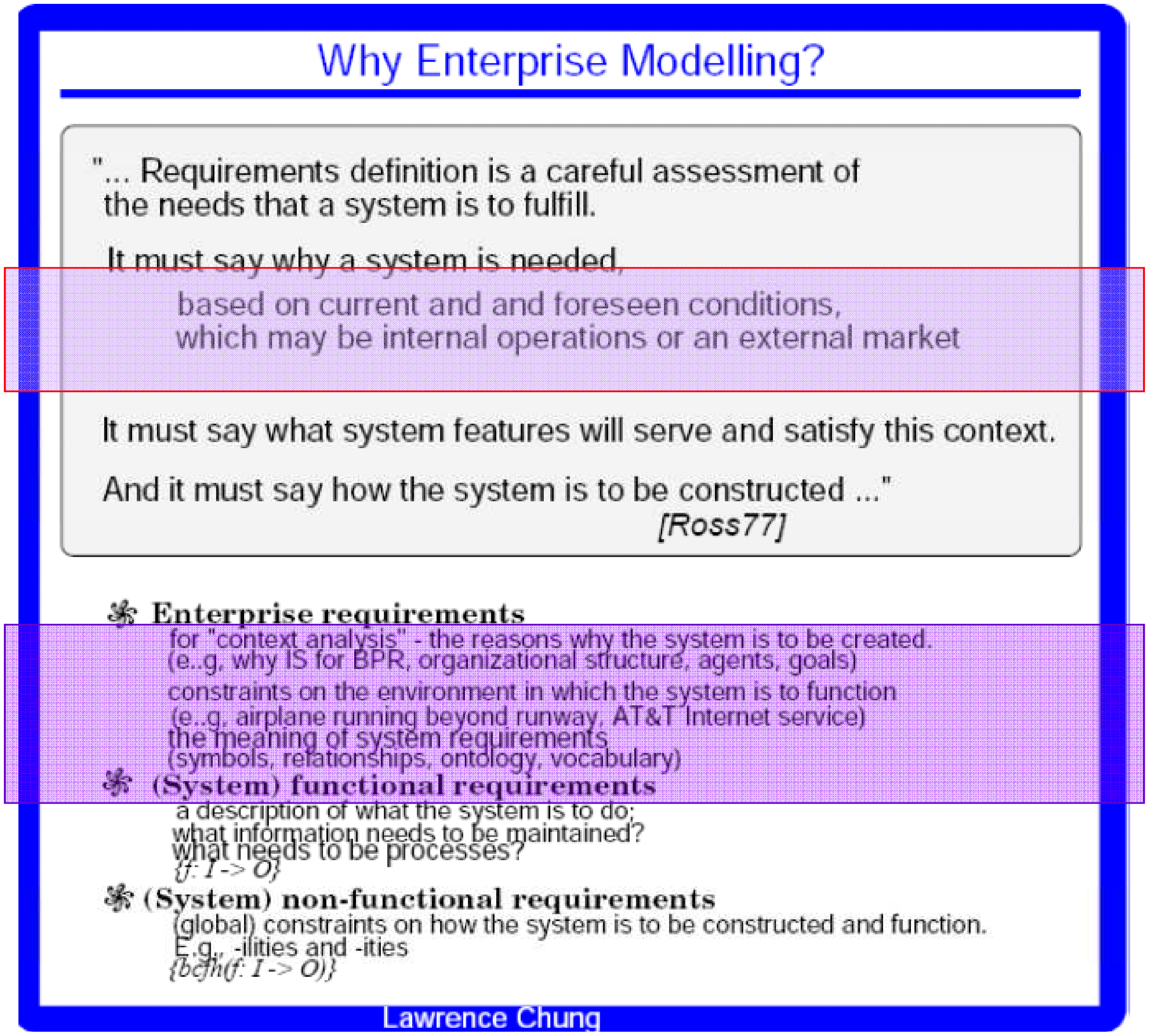
Agent-oriented EM

Business Modeling Using UML

Conventional Techniques

Appendix: More on Enterprise Modeling with the UML

When the only tool that you have is the hammer, every problem will look like a nail.



Why Enterprise Modelling?

"... Requirements definition is a careful assessment of the needs that a system is to fulfill.

It must say why a system is needed,

based on current and and foreseen conditions,
which may be internal operations or an external market

It must say what system features will serve and satisfy this context.

And it must say how the system is to be constructed ..."
[Ross77]

Enterprise requirements

for "context analysis" - the reasons why the system is to be created.
(e..g, why IS for BPR, organizational structure, agents, goals)
constraints on the environment in which the system is to function
(e..g, airplane running beyond runway, AT&T Internet service)
the meaning of system requirements
(symbols, relationships, ontology, vocabulary)

(System) functional requirements

a description of what the system is to do;
what information needs to be maintained?
what needs to be processes?
$\{f: I \to O\}$

(System) non-functional requirements

(global) constraints on how the system is to be constructed and function.
E.g., -ilities and -ities
$\{befn(f: I \to O)\}$

Lawrence Chung

World Knowledge is Essential - the most error prone part of the requirements;
Most problems can be traced to erroneous assumptions about the environment
(e.g., TCAS—transponder assumptions; NY subway—separation not enough; Patriot missile—clock drift)

Enterprise Modeling

Why?

(Goal-oriented EM in Requirements Elicitation)

Agent-oriented EM

Business Modeling Using UML

Conventional Techniques

Appendix: More on Enterprise Modeling with the UML

When the only tool that you have is the hammer, every problem will look like a nail.

Business Modeling
A Business is an Enterprise too

- ***Why Business Modeling ? - Business complexity***
  - Need to understand the structure and dynamics of the organization
  - Common understanding of the organization

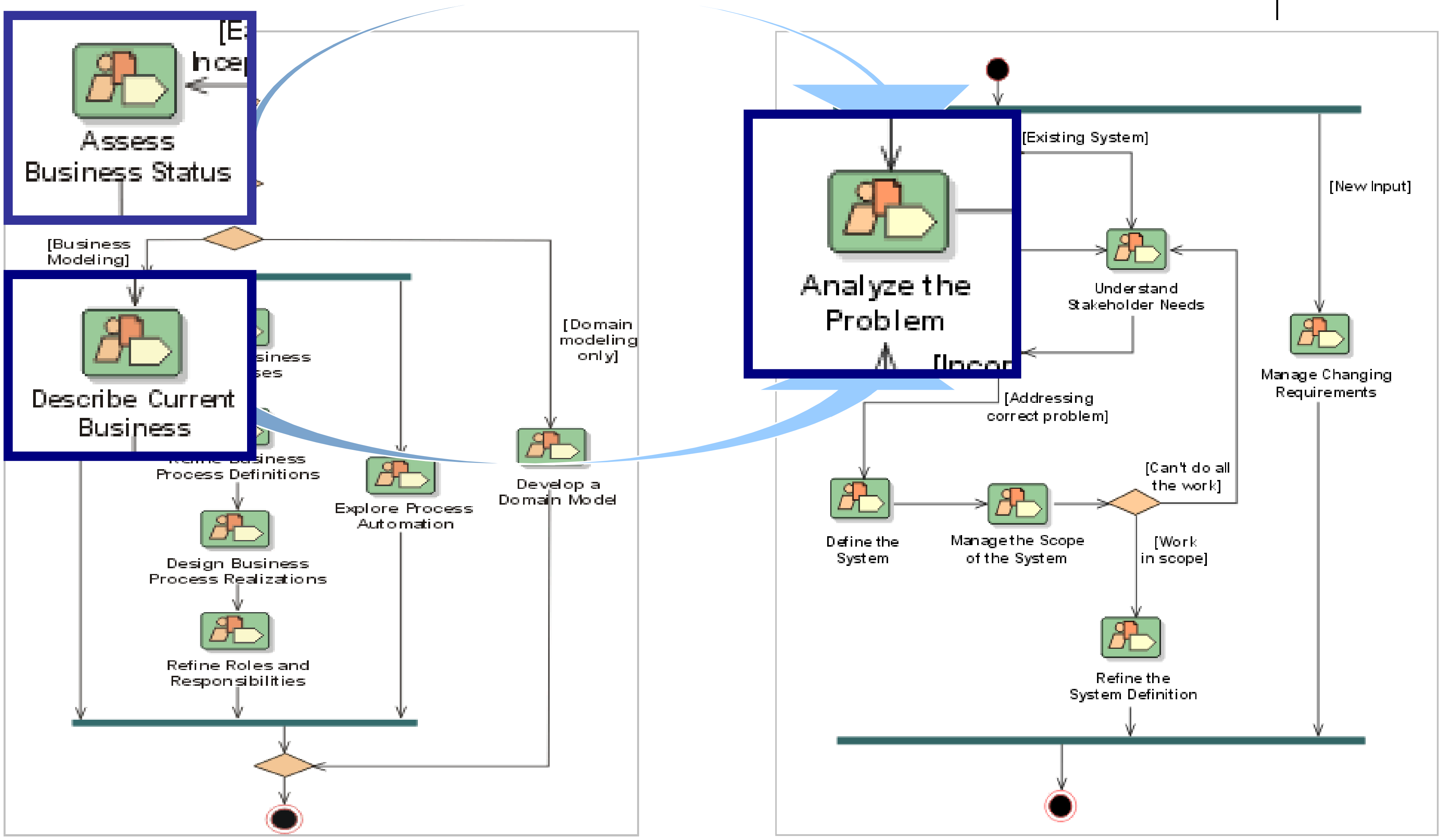
- ***Business modeling is a technique to help answer critical questions, such as:***
  - What do the workers (users) do before using our system?
  - What *business* ***value*** does the system bring?
  - What is the business system (process) this computer system will be supporting?
  - What is the business system information this computer system will be maintaining?
  - How do you know you have identified all system use cases?
  - How do you know you have identified the right use cases?

What are the differences between Enterprise Modeling, Business Modeling and System Modeling?

S achieves **R** to solve **P** in **D**

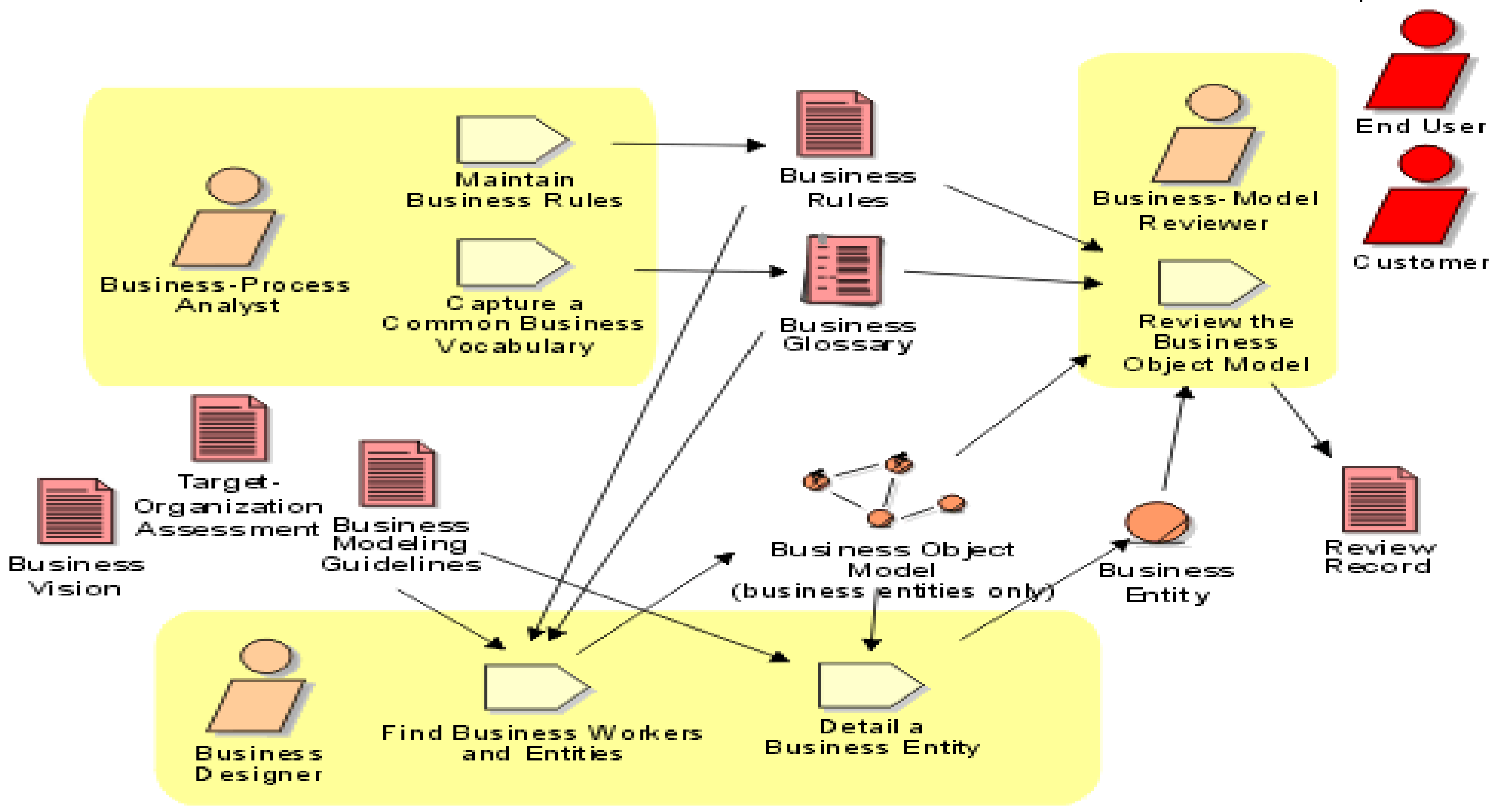
Business Modeling

Requirements



[E
Incep
Assess Business Status
[Business Modeling]
Describe Current Business
siness
ses
Refine Business Process Definitions
Design Business Process Realizations
Refine Roles and Responsibilities
Explore Process Automation
[Domain modeling only]
Develop a Domain Model
Analyze the Problem
[Incep
[Existing System]
[New Input]
Understand Stakeholder Needs
Manage Changing Requirements
[Addressing correct problem]
Define the System
Manage the Scope of the System
[Can't do all the work]
[Work in scope]
Refine the System Definition

Workflow for Business Modeling in the RUP

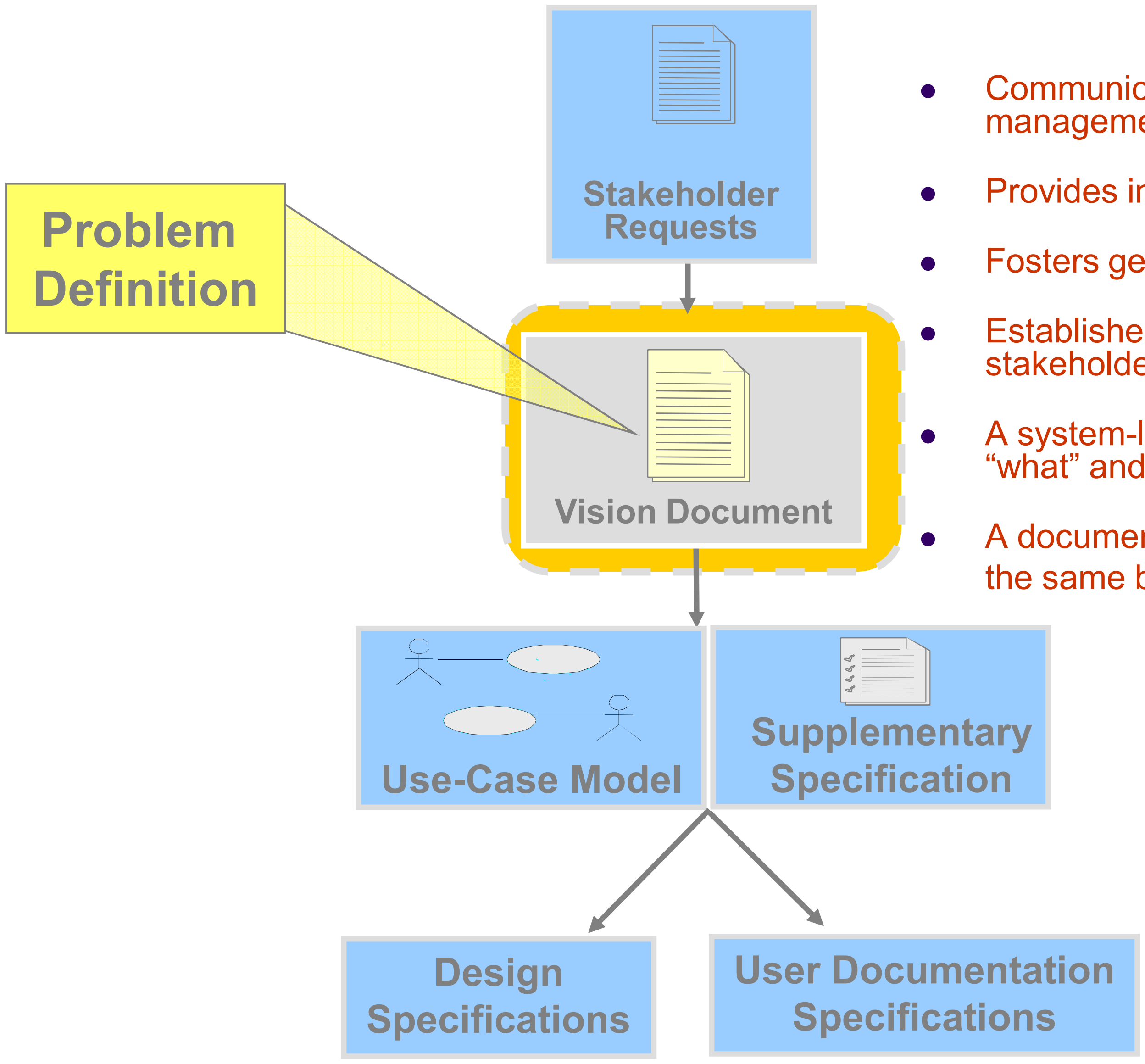

Business-Process Analyst
Maintain Business Rules
Capture a Common Business Vocabulary
Business Rules
Business Glossary
Business-Model Reviewer
Review the Business Object Model
End User
Customer
Business Vision
Target-Organization Assessment
Business Modeling Guidelines
Business Object Model (business entities only)
Business Entity
Review Record
Business Designer
Find Business Workers and Entities
Detail a Business Entity

What Artifacts Are Used to Manage Requirements in RUP?

Where is the problem defined? Where are the stakeholders and users listed? Where are the environments and platforms identified?	**Vision**
Where are the non-functional requirements located?	**Supplementary Spec**
Where are the use cases maintained?	**Use Case Specs**
Where is the common terminology stored?	**Glossary**
Where are the stakeholder Needs/Requests captured?	**Stakeholder Requests**

Business Modeling Using UML

Describe the Problem in the *Vision* Document

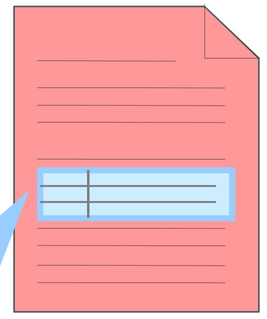


- Communicates information between management, marketing, and the project team.
- Provides initial customer feedback.
- Fosters general understanding of the product.
- Establishes scope and priority of high-level stakeholder requests and features.
- A system-level document that describes the "what" and "why" of the product.
- A document that gets "all parties working from the same book."

Vision Document Outline*

1. Introduction
2. Positioning
3. Stakeholder and User Descriptions
4. Product Overview
5. Product Features
6. Constraints
7. Quality Ranges
8. Precedence and Priority
9. Other Product Requirements
10. Documentation Requirements
11. Appendix 1 - Feature Attributes

Vision



The problem of	(describe the problem)
affects	(the stakeholders affected by the problem)
the impact of which is	(what is the impact of the problem)
a successful solution would	(list some key business benefits of a successful solution)

Problem Statement

* see http://www.utdallas.edu/~chung/SAMPLE/vision_exercise.pdf

Business Modeling Using UML

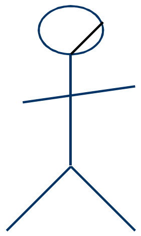
business actors, business use cases, business object models

- define who and what will interact with the business.
- define what services the business are to provide.
- To develop a survey of the business services.
- define the boundaries of the business to be modeled.
- outline the processes in the business.

Domain model – One Notion:

- an "incomplete" *business object model*, i.e., a subset of the business object model
- focusing on explaining products, deliverables, or events that are important to the business domain,
- ***but not*** including the *responsibilities* people carry.

Find Business Actors



- A business actor candidate is any individual, group, organization, company, or machine that interacts with the business:
    - Customers
    - Partners
    - Suppliers
    - Authorities (legal, regulatory, and so forth)
    - Subsidiaries
    - Owners and investors (the board of directors may be part of the business or modeled as an actor)
    - Information systems outside of the business
- If the business to be modelled is part of a large company, these categories may also contain business actors:
    - Other parts of the company
    - Individual roles within other departments
- Name each business actor using its role in the business. Define each business actor briefly, including its responsibility and why it interacts with the business.

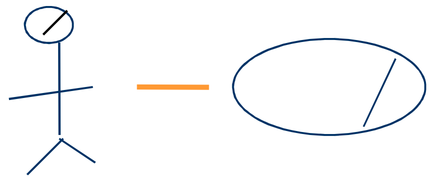
Find Business Use Cases

- For the primary business use cases, consider what value each business actor gets from the business. Start with the primary and most important business actors — the customers:
  - What are the primary services a customer receives from the business? Study the customer's lifecycle:
    - What was the customer's first contact with the business?
    - What stages or states does the customer go through in relation to the business?

- From a perspective of supporting the business, processes can also be represented as business use cases. Activities to:
  - Develop and maintain the staff
  - Develop and maintain the IT within the business
  - Develop and maintain the office
  - Security
  - Legal activities

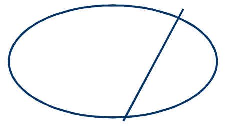
- From the perspective of managing the business, although not as interesting from an IS perspective, consider what the owner actors get from the business. Activities to:
  - Develop and provide information about the business to owners and investors
  - Set up long-term budget goals
  - Coordinate and prioritize between the other use cases in the business
  - Create new processes in the business
  - Monitor the processes in the business

Describe How Business Actors and Business Use Cases Interact



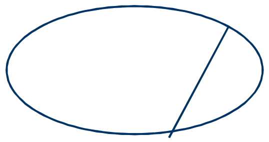
- Establish those business actors who interact with the business use case by defining a *communicates-association* between them.
- If it's important to show who initiated the communication, you can add navigability to the association.

Prioritize Business Use Cases



- Prioritize those business use cases that are of interest and to be described in some detail:
  - Involved in *business (re-)engineering, hence* to find requirements on the intended information system.
  - In need of a step-by-step description before making a decision whether to become business use cases or not, due to their unclear relevance to the intended information-system.

Develop an Outline of the Workflow of Business Use Cases



- **A step-by-step outline of the workflow to understand the purpose of the business use case.**

  Example:

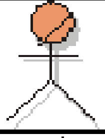
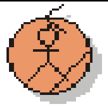
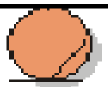
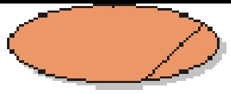
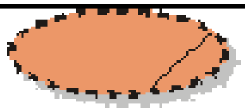
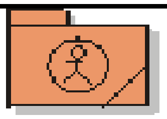
  - The first draft of a step-by-step workflow description of the business use case "Individual Check-in" might look:
    - Passenger enters the queue to the check-in counter.
    - Passenger gives ticket to check-in agent.
    - Check-in agent validates ticket.
    - Check-in agent registers baggage.
    - Check-in agent reserves seat for the passenger.
    - Boarding card is printed.
    - Check-in agent gives passenger boarding card.
    - Passenger leaves the check-in counter.

- **As a first draft, it may very well lack activities that will be discovered later. May also include alternative flows in this first draft.**

Final steps

- **Package Business Use Cases and Actors**
- **Present the Business Use-Case Model in Use-Case Diagrams**
  - Combination of business actors, business use cases, and their relationships:
    - a business actor and all the business use cases with which he or she interacts
    - business use cases that interact with the same business actors
    - business use cases that are usually performed in a sequence
    - business use cases that belong to the same use case package
    - the most important business use cases, as a summary of the complete business use-case model and for reviewing the model
- **Develop a Survey of the Business Use-Case Model**
  - The Survey Description of the business use-case model conveys:
    - the purpose of the business being described
    - the typical sequences in which the business use cases are employed
    - the parts of the business that are not included in the business use-case model
- **Evaluate Your Results**
  - all necessary business use cases are identified
  - any unnecessary business use cases are identified
  - the behavior of each business use case is described in the right order
  - each business use case's workflow is as complete as it could be at this stage
  - the Survey Description of the business use-case model makes it understandable

Business Modeling with the UML

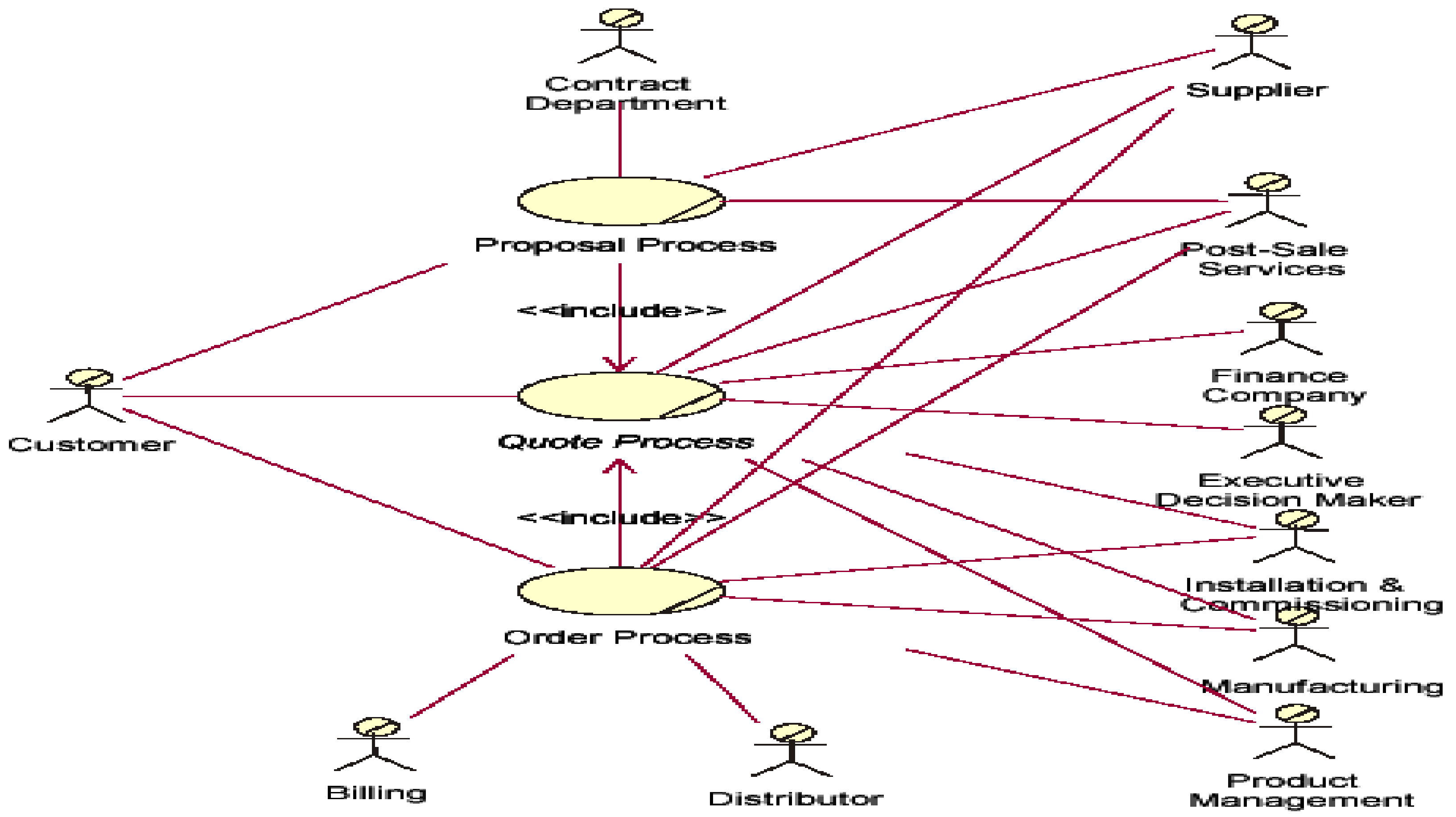
Modeling icon	Name	UML Definition
	Business actor	Someone or something, outside the business that interacts with the business.
	Business worker	Role or set of roles inside the business. A business worker interacts with other business workers and manipulates business entities.
	Business entity	A "thing" handled or used by business workers.
	Business use case	A sequence of actions a business performs that yields an observable result of value to a particular business actor. (In this paper, synonymous with business process)
	Business use case realization	A collection of diagrams to show how the organization elements (workers and entities) are deployed to support a business process.
	Organizational unit	A collection of business workers, business entities, relationships, business use-case realizations, diagrams, and other organization units. Used to structure the business (object) model by dividing it into smaller parts.

The UML provides different diagrams

Each UML diagram provides a different view of the business:

- *use case* diagrams describe the business ***context***.
- *activity* diagrams describe ***behaviors*** in the business, or business workflows.
- *class* diagrams describe the static ***structure*** in the business.
- *interactions* diagrams (*sequence* diagrams and *collaboration* diagrams) describe the dynamic ***interactions*** between employees and things that they manipulate. Thus they indicate how the behaviors described in activity diagrams are realized.
- *state transition* diagrams,
- *deployment* diagrams,
- *object* diagrams,
- etc.

(Business) Use-Case Diagram

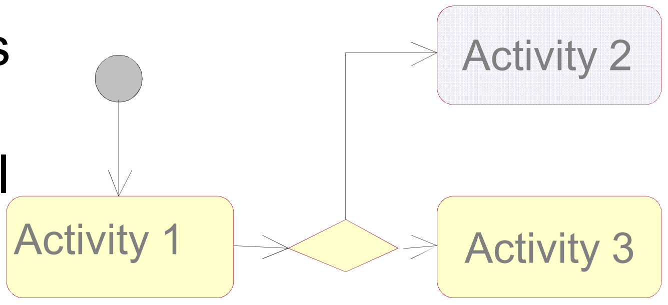


(Business) Activity Diagrams

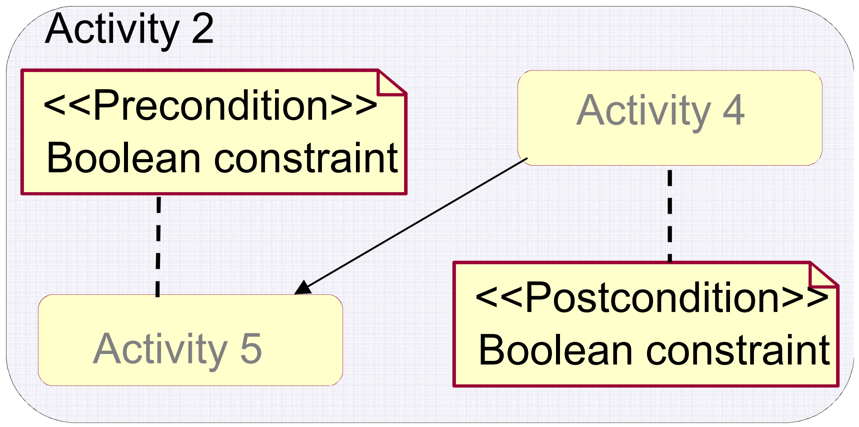
To document a business workflow:

- what happens in a workflow,
- what activities can be done in parallel,
- whether there are alternative paths through a workflow.

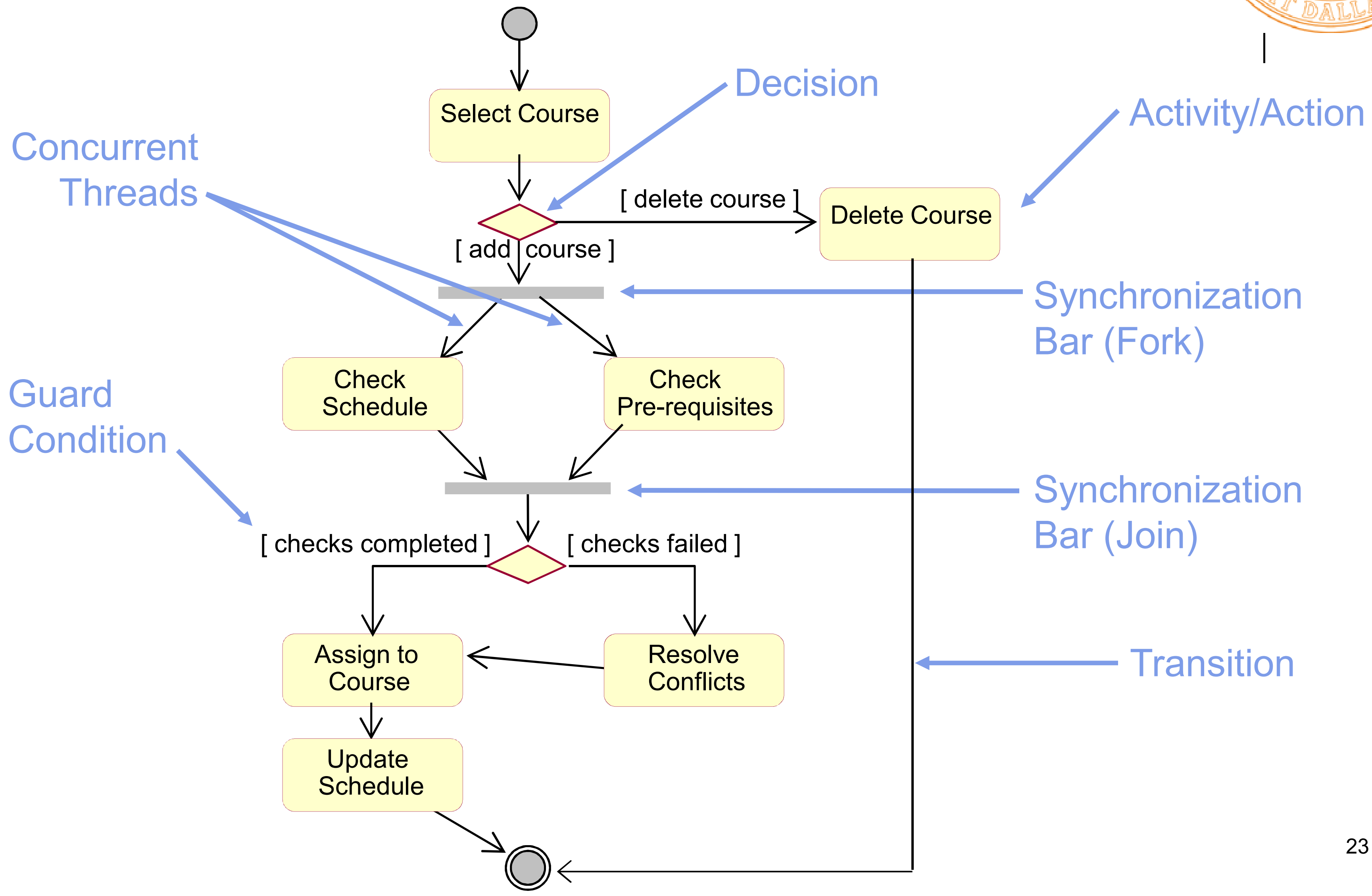
- In the use-case model: to capture the activities and actions;
- Essentially a flow chart, showing flow of control from one activity or action to another.



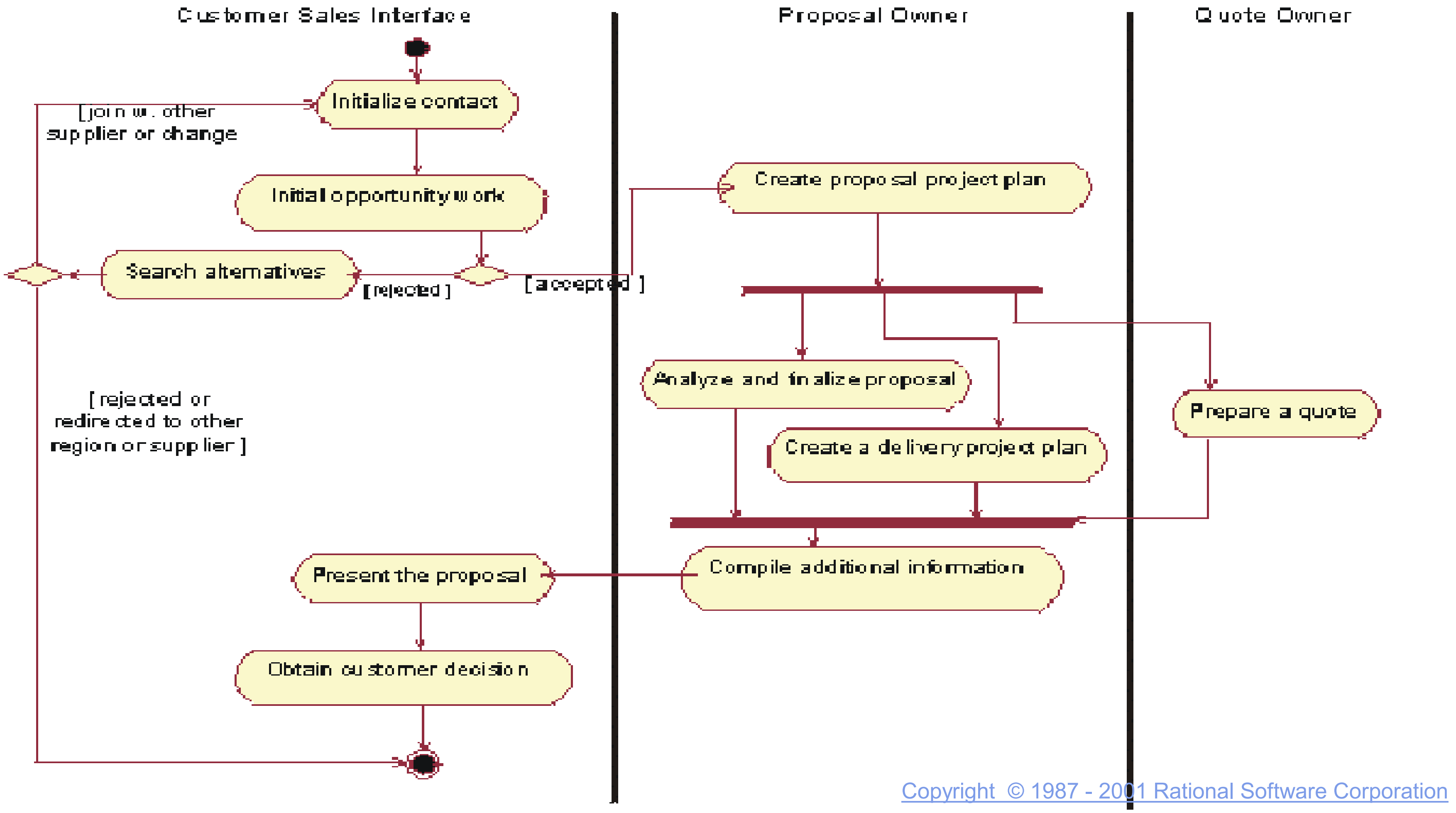
- An activity specifies the behavior expressed as a flow of execution via sequencing of subordinate units.
  - Subordinate units include nested activities and ultimately individual actions.
- May contain boolean expression constraints when the activity is entered or exited.



Example1: Activity Diagram

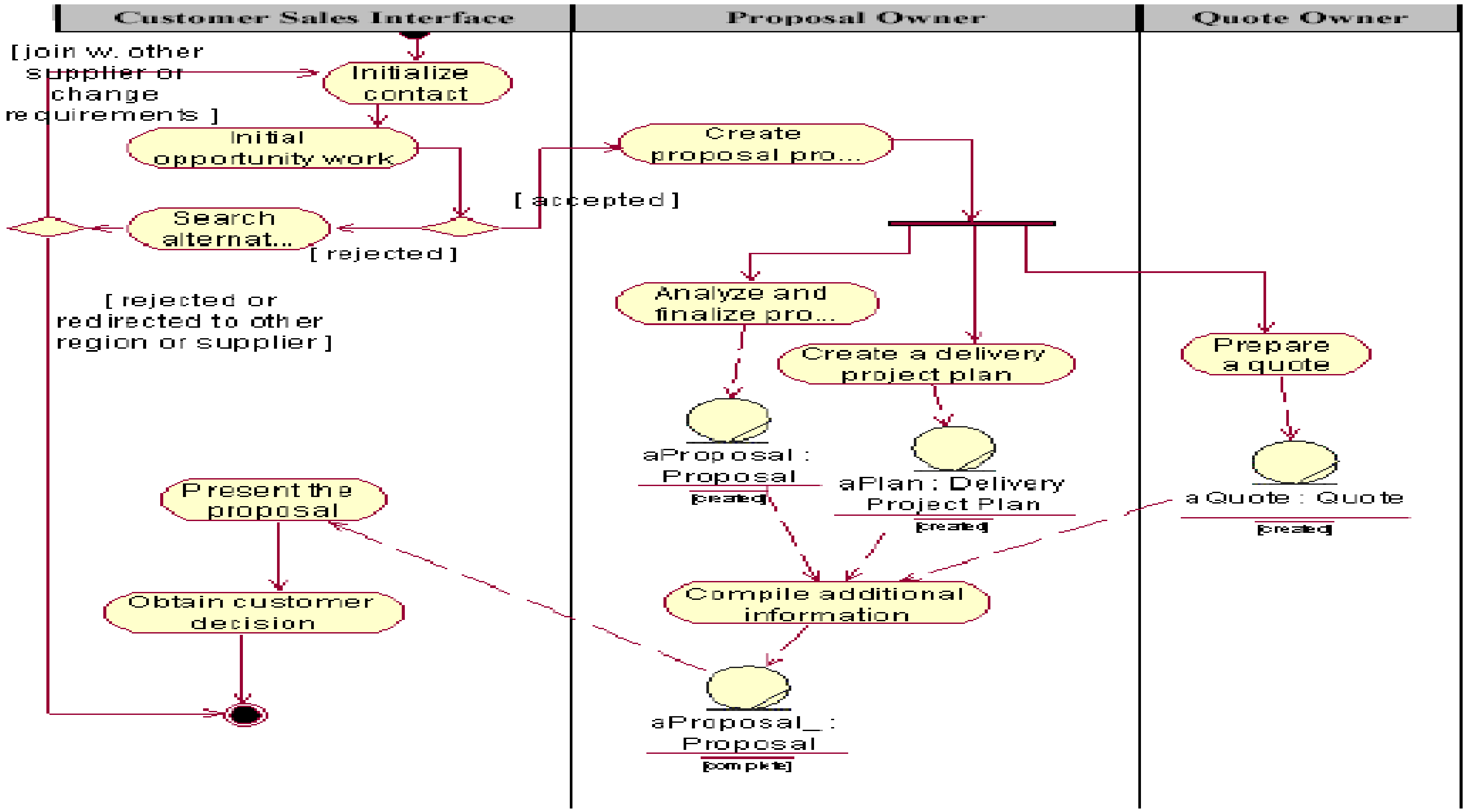


Example2: business activity diagram


Customer Sales Interface
Proposal Owner
Quote Owner
Initialize contact
[join w. other supplier or change
Initial opportunity work
Search alternatives
[rejected]
[accepted]
Create proposal project plan
Analyze and finalize proposal
Create a delivery project plan
Prepare a quote
[rejected or redirected to other region or supplier]
Present the proposal
Compile additional information
Obtain customer decision

An Activity Diagram documenting how the business performs a *Proposal* process, with three areas of responsibilities: Customer Sales Interface, Proposal Owner, and Quote Owner.

Show business entities being manipulated in the activities

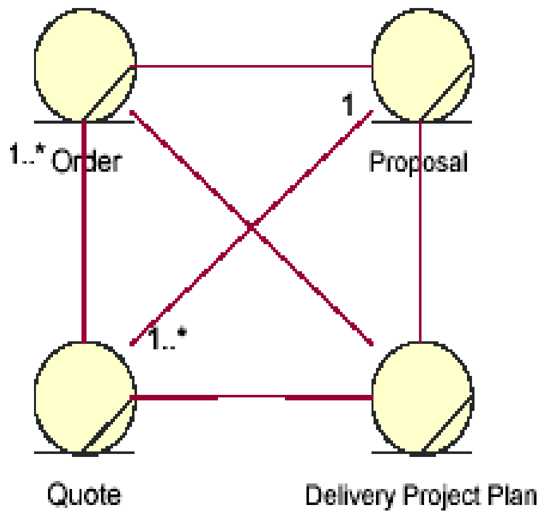


An activity diagram showing business entities (a Proposal, a Quote, a Plan) and their states (created/complete).

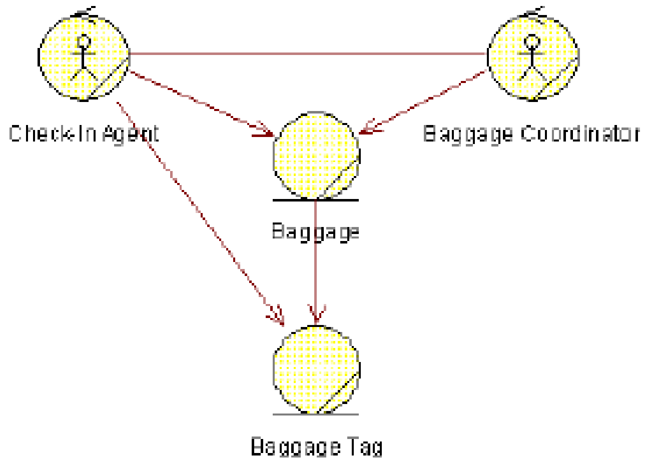
Business class diagrams

?

- To show **which** ***business workers*** and ***business entities*** are collaborating to implement a business process.
- To show **static structure** and relationships among ***business entities***. A class diagram would be used to represent the org chart of a business (using organization units and business workers).



A class diagram showing relationships among ***business entities***.



A class diagram showing relationships between ***business workers*** (*Check-in Agent*, *Baggage Coordinator*) and ***business entities*** (*Baggage*, *Baggage Tag*), showing that the Check-In Agent has the knowledge of a Baggage Tag, but the Baggage Coordinator does not.

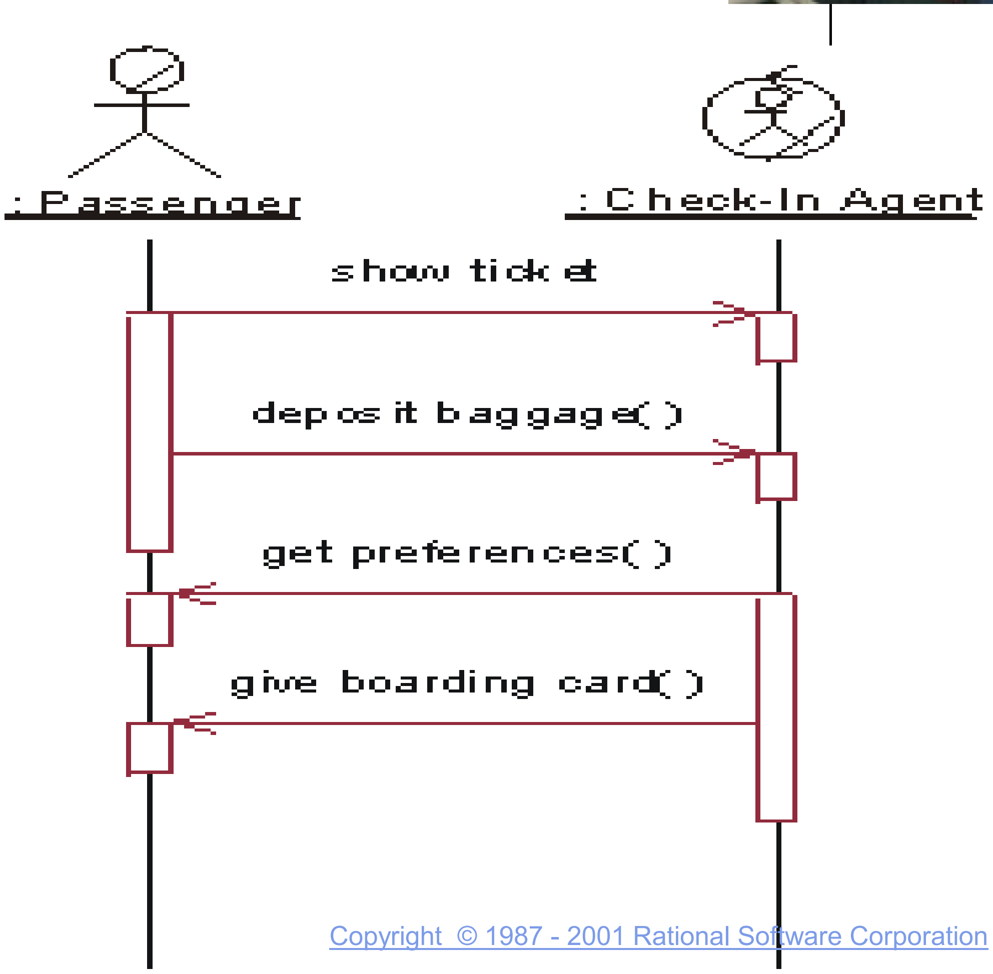
Business Sequence Diagram

The passenger shows the ticket.

The passenger deposits the baggage.

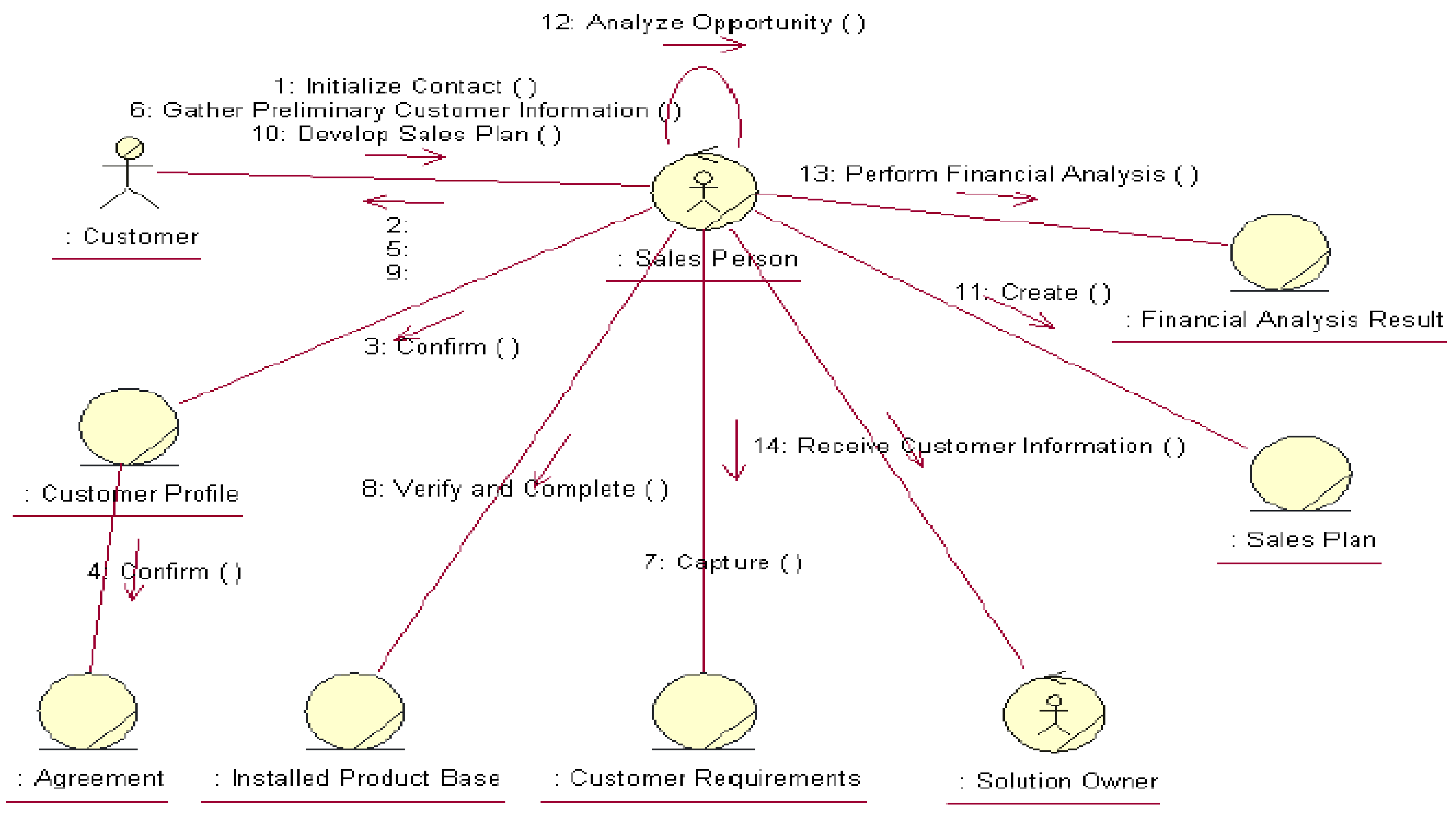
The passenger is asked for preferences.

The passenger receives a boarding card.



Can you come up with a collaboration diagram from this?

A business collaboration diagram


12: Analyze Opportunity ( )
1: Initialize Contact ( )
6: Gather Preliminary Customer Information ( )
10: Develop Sales Plan ( )
2:
5:
9:
: Customer
: Sales Person
13: Perform Financial Analysis ( )
11: Create ( )
: Financial Analysis Result
3: Confirm ( )
: Customer Profile
8: Verify and Complete ( )
14: Receive Customer Information ( )
: Sales Plan
4: Confirm ( )
7: Capture ( )
: Agreement
: Installed Product Base
: Customer Requirements
: Solution Owner

A collaboration diagram showing a view of participating business workers (Sales Person, Solution Owner) and business entities (Customer Profile, Sales Plan, etc) in a *Proposal* process.

Can you come up with a sequence diagram from this?

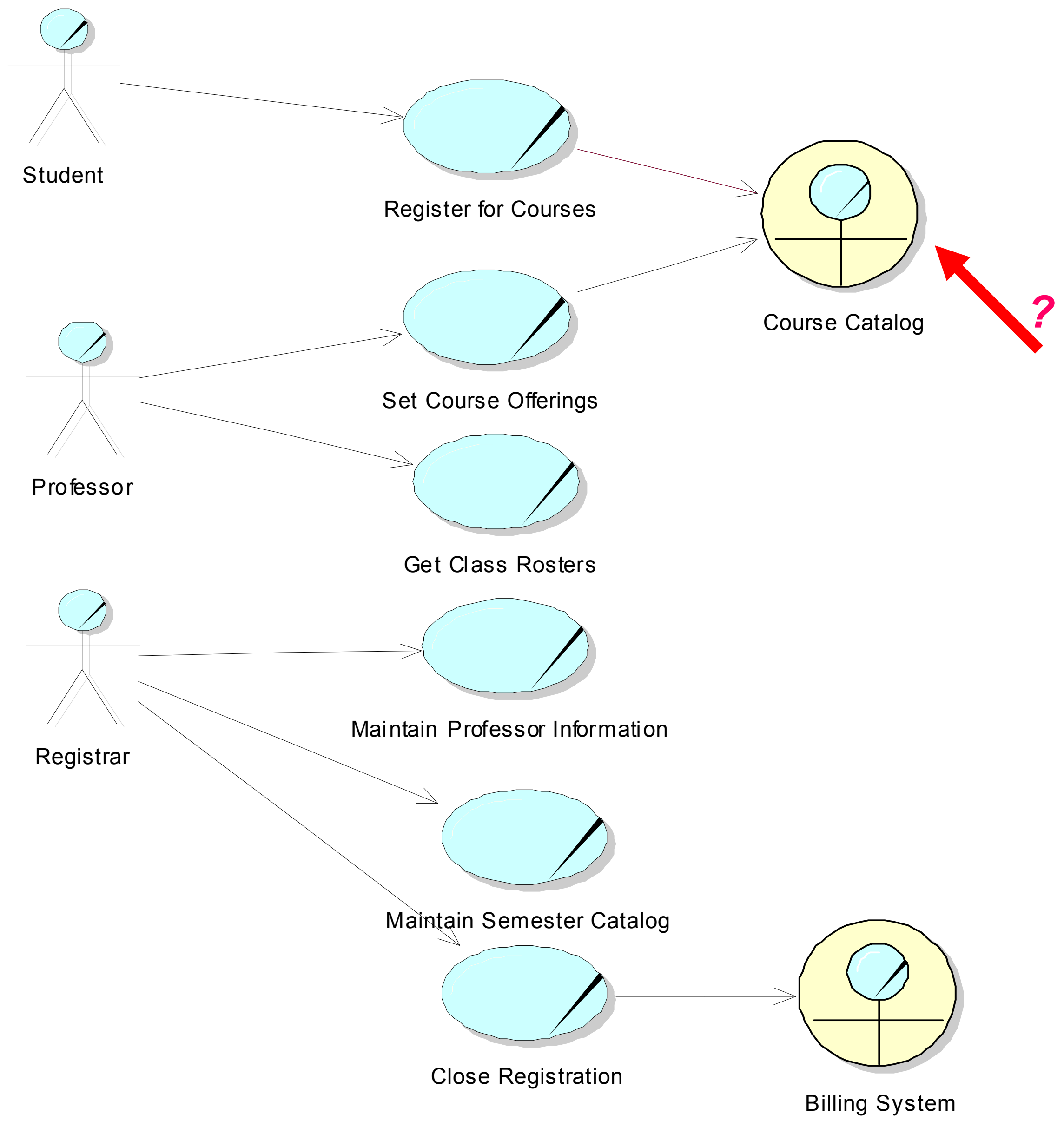
Practice draw:
(1) Use Case Diagram,
(2) Activity Diagram and
(3) Business Class Diagram

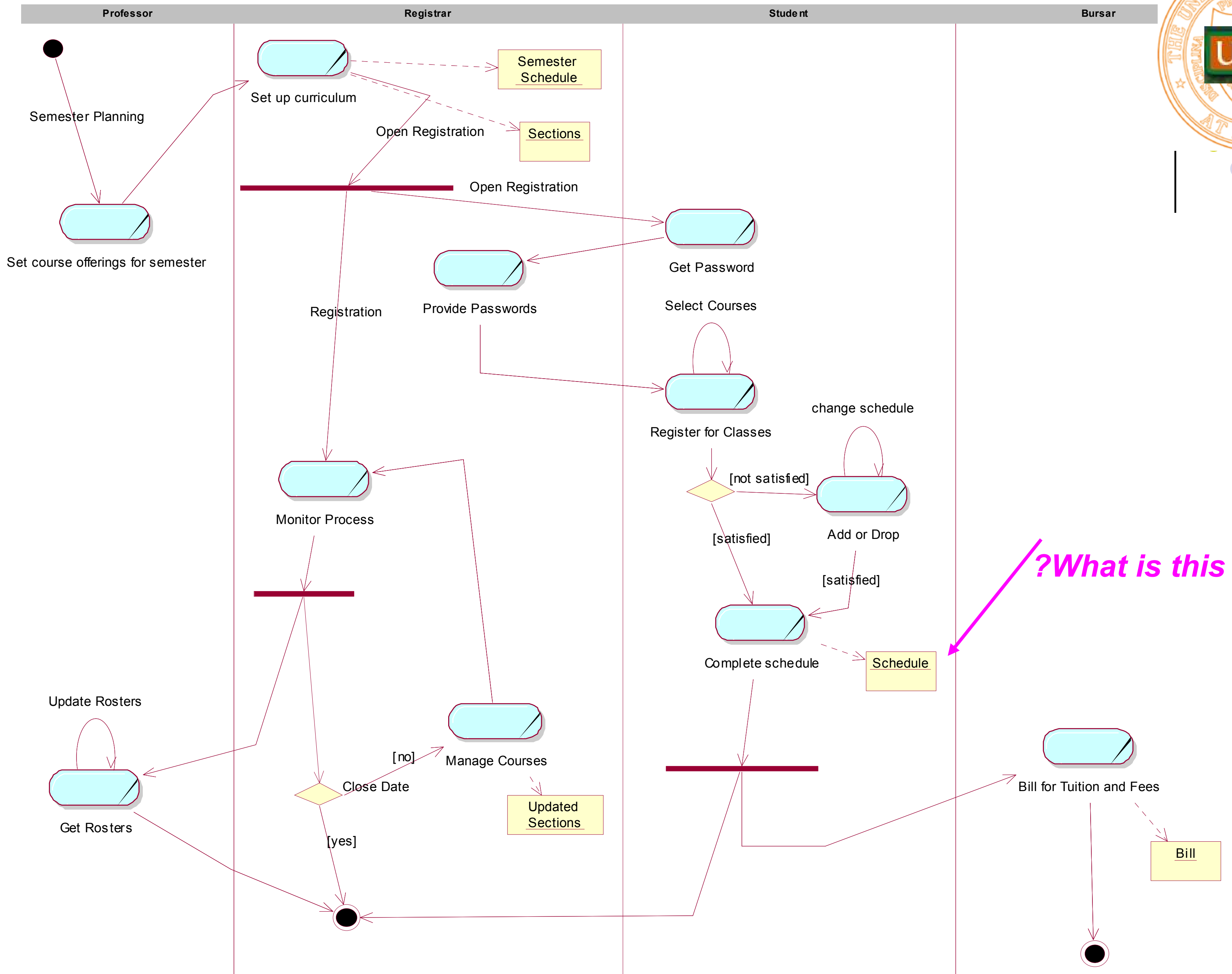
Business Process to be implemented

UTD wants to computerize its registration system

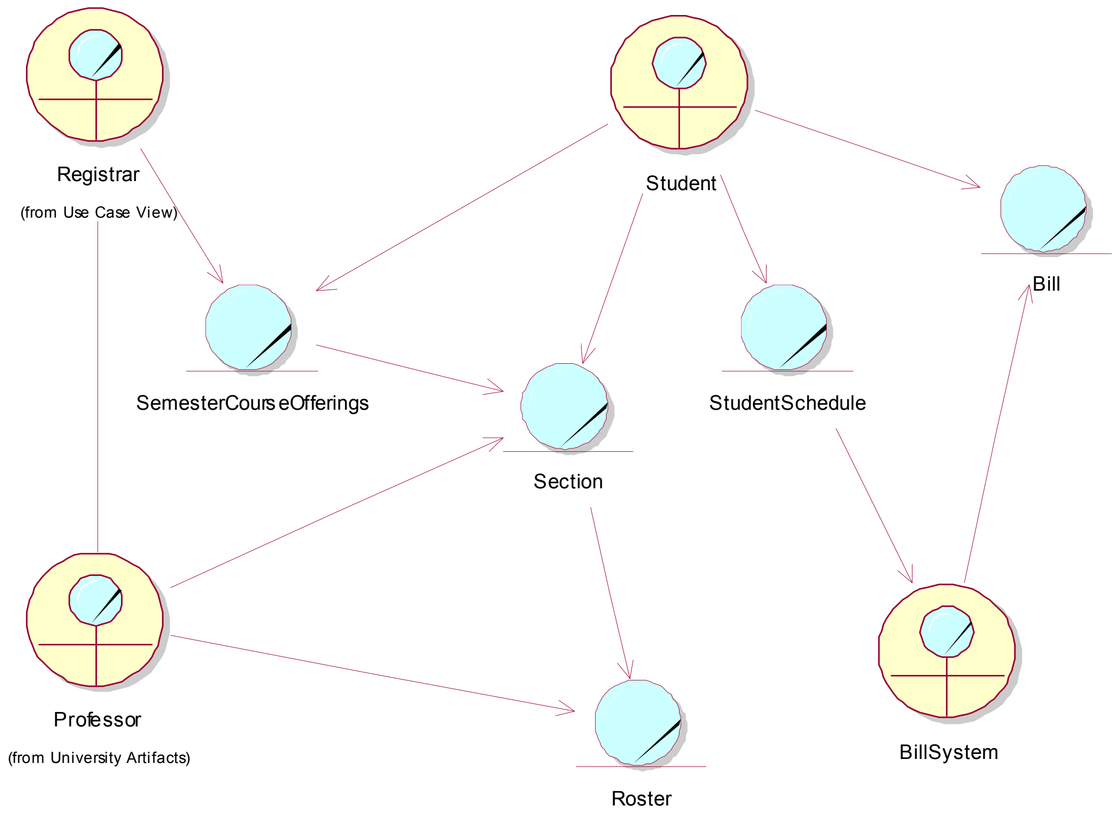
- The Registrar sets up the curriculum for a semester
  - One course may have multiple course offerings
- Students select four (4) primary courses and two (2) alternate courses
- Once a student registers for a semester, the billing system is notified so the student may be billed for the semester
- Students may use the system to add/drop courses for a period of time after registration
- Professors use the system to set their preferred course offerings and receive their course offering rosters after students register
- Users of the registration system are assigned passwords which are used at logon validation

What kind of diagram is this?




Professor
Registrar
Student
Bursar
Semester Planning
Set up curriculum
Semester Schedule
Open Registration
Sections
Open Registration
Set course offerings for semester
Get Password
Provide Passwords
Registration
Select Courses
Register for Classes
change schedule
Monitor Process
[not satisfied]
Add or Drop
[satisfied]
[satisfied]
?What is this
Complete schedule
Schedule
Update Rosters
Manage Courses
[no]
Close Date
Get Rosters
Updated Sections
[yes]
Bill for Tuition and Fees
Bill

What kind of diagram is this?



Where are actors?

Some Critical Review Questions

Can you show how business actors, business workers, and (software system) actors work together?

Can you show how business use cases are related to (software system) use cases?

Can you show how business class diagrams are related to (software system) use diagrams?

Can you show how business use cases are related to business sequence diagrams?

Can you show how use case diagrams are related to state transition diagrams?

True of False?
Actors can be associated only with use cases, classes or components, but *not* with other actors.

Enterprise Modeling

Why?

(Goal-oriented EM in Requirements Elicitation)

Agent-oriented EM

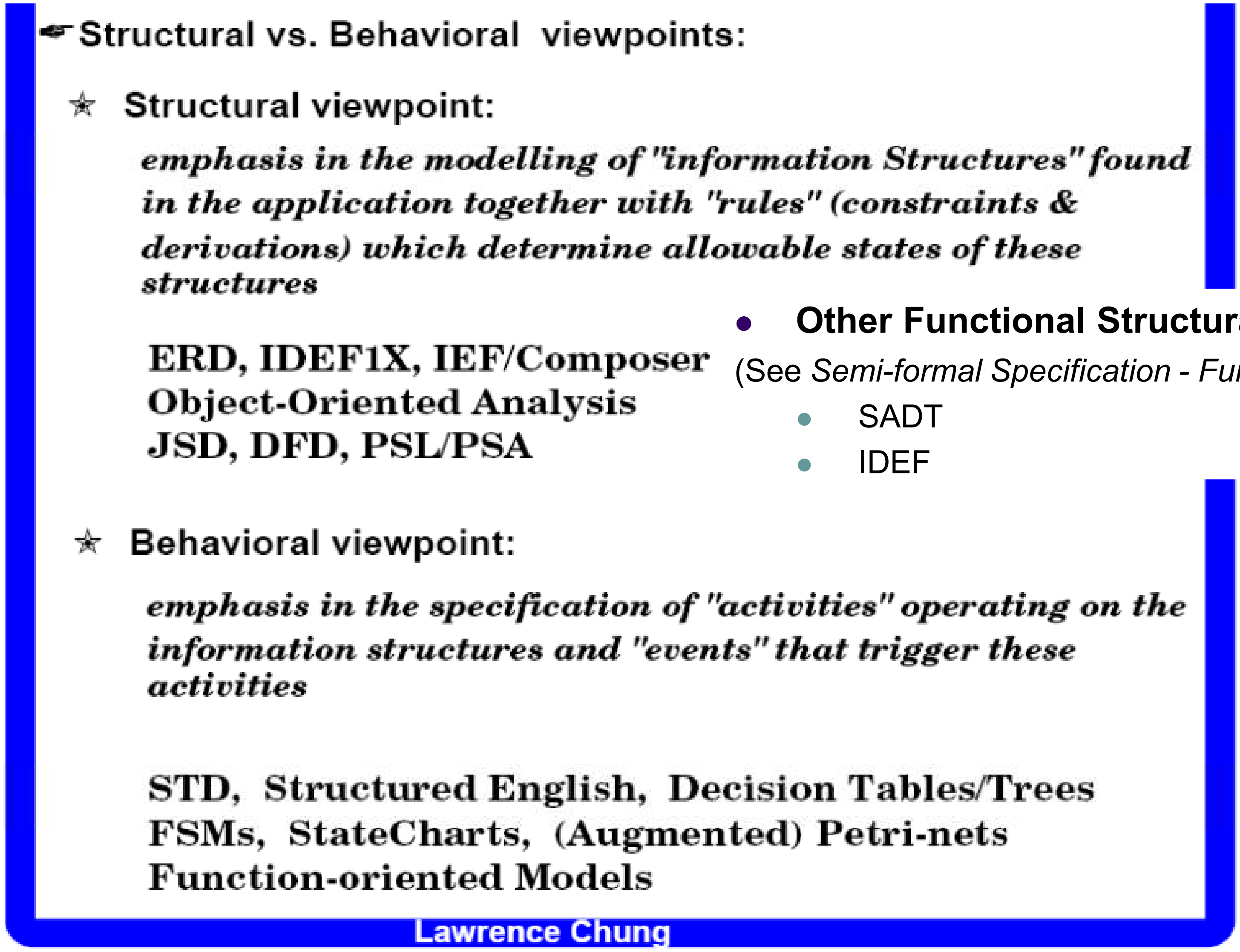
Business Modeling Using UML

Conventional Techniques

Appendix: More on Enterprise Modeling with the UML

When the only tool that you have is the hammer, every problem will look like a nail.

Conventional Techniques



☞ Structural vs. Behavioral viewpoints:

☆ Structural viewpoint:

emphasis in the modelling of "information Structures" found in the application together with "rules" (constraints & derivations) which determine allowable states of these structures

ERD, IDEF1X, IEF/Composer
Object-Oriented Analysis
JSD, DFD, PSL/PSA

☆ Behavioral viewpoint:

emphasis in the specification of "activities" operating on the information structures and "events" that trigger these activities

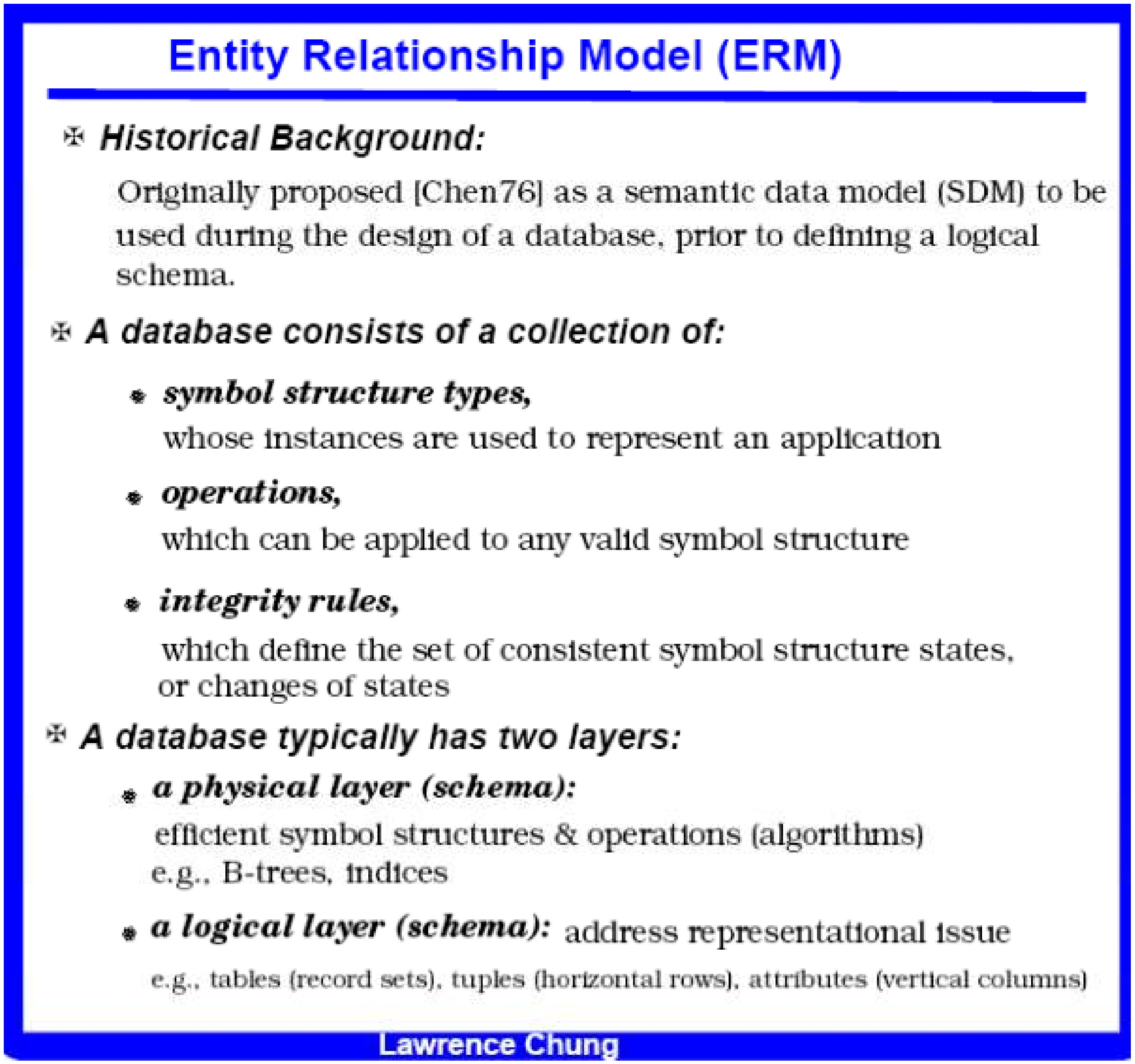
STD, Structured English, Decision Tables/Trees
FSMs, StateCharts, (Augmented) Petri-nets
Function-oriented Models

Lawrence Chung

- **Other Functional Structural RM techniques**

(See *Semi-formal Specification - Functional Structural RM)*:

  - SADT
  - IDEF



Entity Relationship Model (ERM)

- ***Historical Background:***

  Originally proposed [Chen76] as a semantic data model (SDM) to be used during the design of a database, prior to defining a logical schema.

- ***A database consists of a collection of:***
  - ***symbol structure types,***
    whose instances are used to represent an application
  - ***operations,***
    which can be applied to any valid symbol structure
  - ***integrity rules,***
    which define the set of consistent symbol structure states, or changes of states

- ***A database typically has two layers:***
  - ***a physical layer (schema):***
    efficient symbol structures & operations (algorithms)
    e.g., B-trees, indices
  - ***a logical layer (schema):*** address representational issue
    e.g., tables (record sets), tuples (horizontal rows), attributes (vertical columns)

Lawrence Chung

Relational Database

Students

ID#	Name	Address
1234	Alice	Wonderland
2345	Robinhood	Nottingham
3456	Pinocchio	Anonymous St., Italy

Courses

Number	Name
CS6361	Requirements Engineering
CS6362	Software Architecture & Design
CS6367	Sw Testing, Validation/Verification

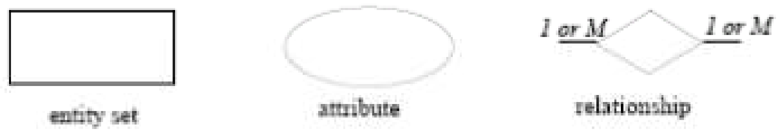
- Operations: add/delete/modify tuple
- Integrity rule: NO 2 tuples in the same table can have the same key.

BUT, tables are "flat", can't describe complex assemblies

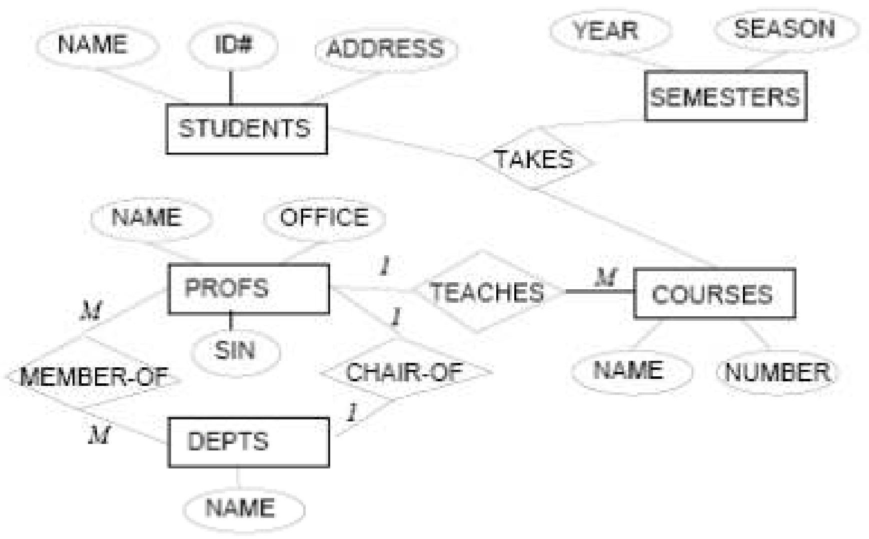
☛ SO, use a SDM like ERD

ERD

Symbos used in ERDs:



Example: ERD for a university database



```
Note: multiple relationships between same entity sets
        (e.g., member-of, chair-of)
```

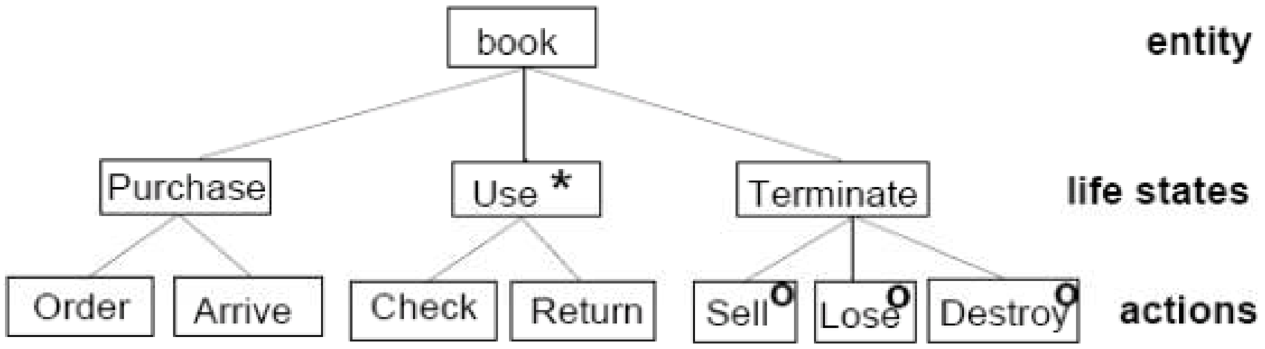
- Appropriate for modelling static parts of an application, but not dynamic parts
- Generalization is not supported

Lawrence Chung

Jackson System Development (JSD)

- First step in the development process is the construction of a model of the (relevant) real world, consisting of entities & actions.
- Entities have roles characterized by responses to sets of actions. They are defined in terms of their life states and the actions that cause these to change.
- Actions are discrete events, which take place in the real world external to the system.

E.g., The life of a book entity



⋆ *iteration*
o *selection*

⌘ *For subsequent development phases, see: M. Jackson, System Development, Prentice Hall, 1983*

Appendix I: More on Enterprise/Domain/Business Modeling with the UML

The Unified Process Another Perspective

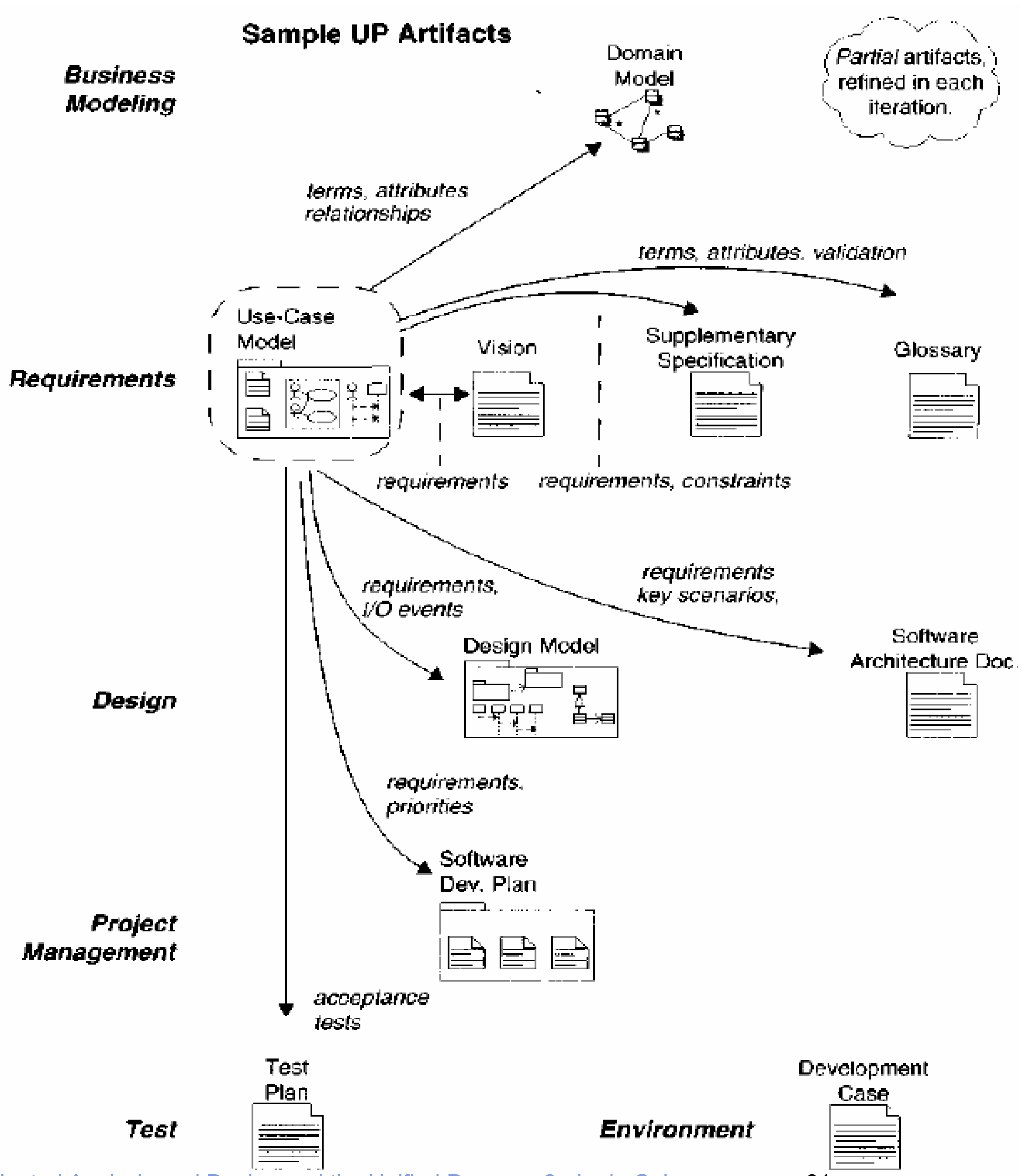
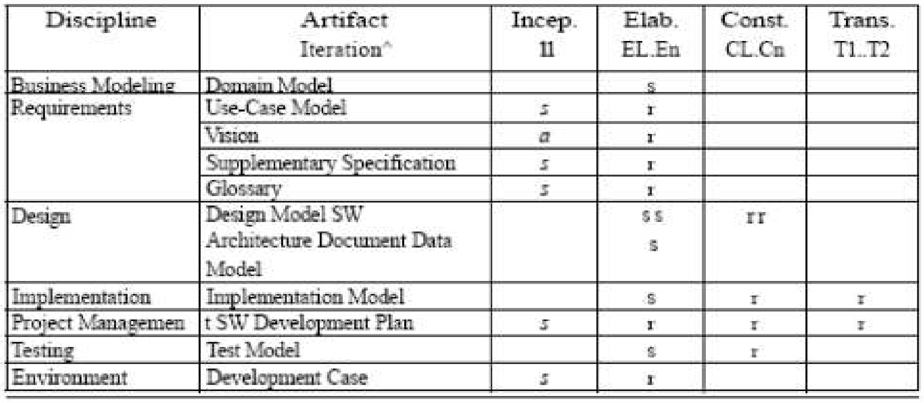


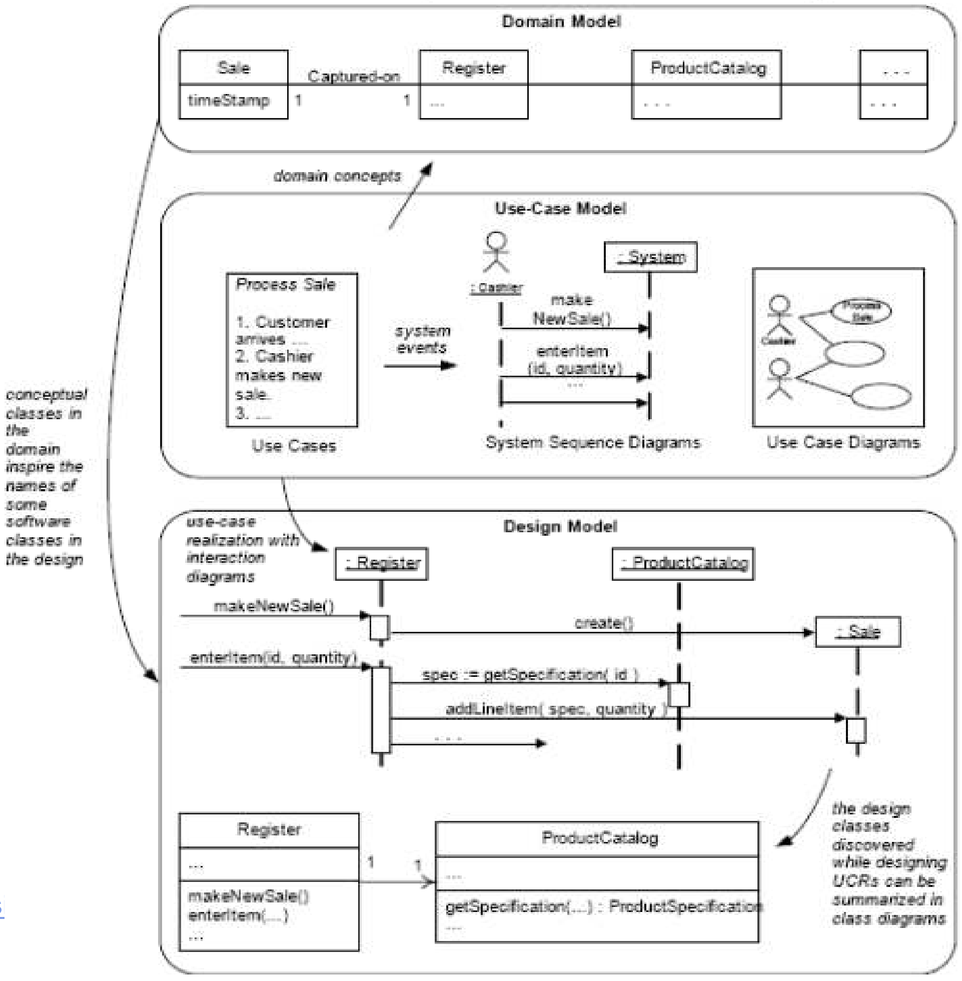
Figure 6.5 Sample UP artifact influence.

The Unified Process: Another Perspective



Discipline	Artifact Iteration^	Incep. I1	Elab. EL.En	Const. CL.Cn	Trans. T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	a	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model SW Architecture Document Data Model		s s s	r r	
Implementation	Implementation Model		s	r	r
Project Managemen	t SW Development Plan	s	r	r	r
Testing	Test Model		s	r	
Environment	Development Case	s	r		

Sample Unified Process Artifact Relationships



Applying UML and Patterns: An Introduction to Object-Oriented Analys

The Unified Process: Another Perspective

Analysis emphasizes an *investigation* of the problem and requirements, rather than a solution. For example, if a new computerized library information system is desired, how will it be used?

"Analysis" is a broad term, best qualified, as in *requirements analysis* (an investigation of the requirements) or *object analysis* (an investigation of the domain objects).

Design emphasizes a *conceptual solution* that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Ultimately, designs can be implemented

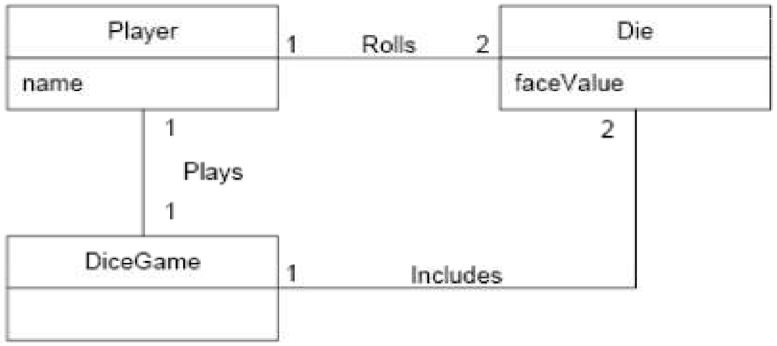
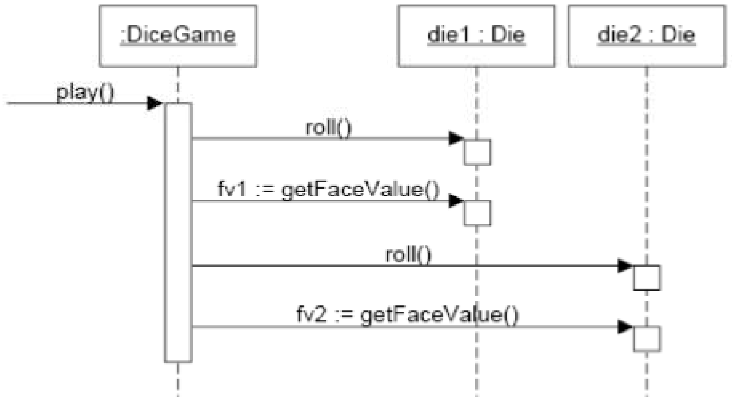
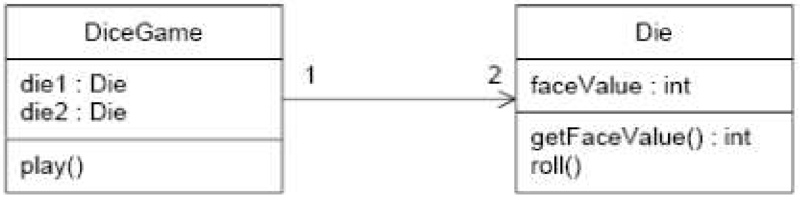


Figure 1.3 Partial domain model of the dice game.



1.4 Interaction diagram illustrating messages between software objects.



Applying UML and Patterns: An Introduction to Object-Oriented Analysis

Figure 1.5 Partial design class diagram.

Conveyor Line Sorting System (CLSS)

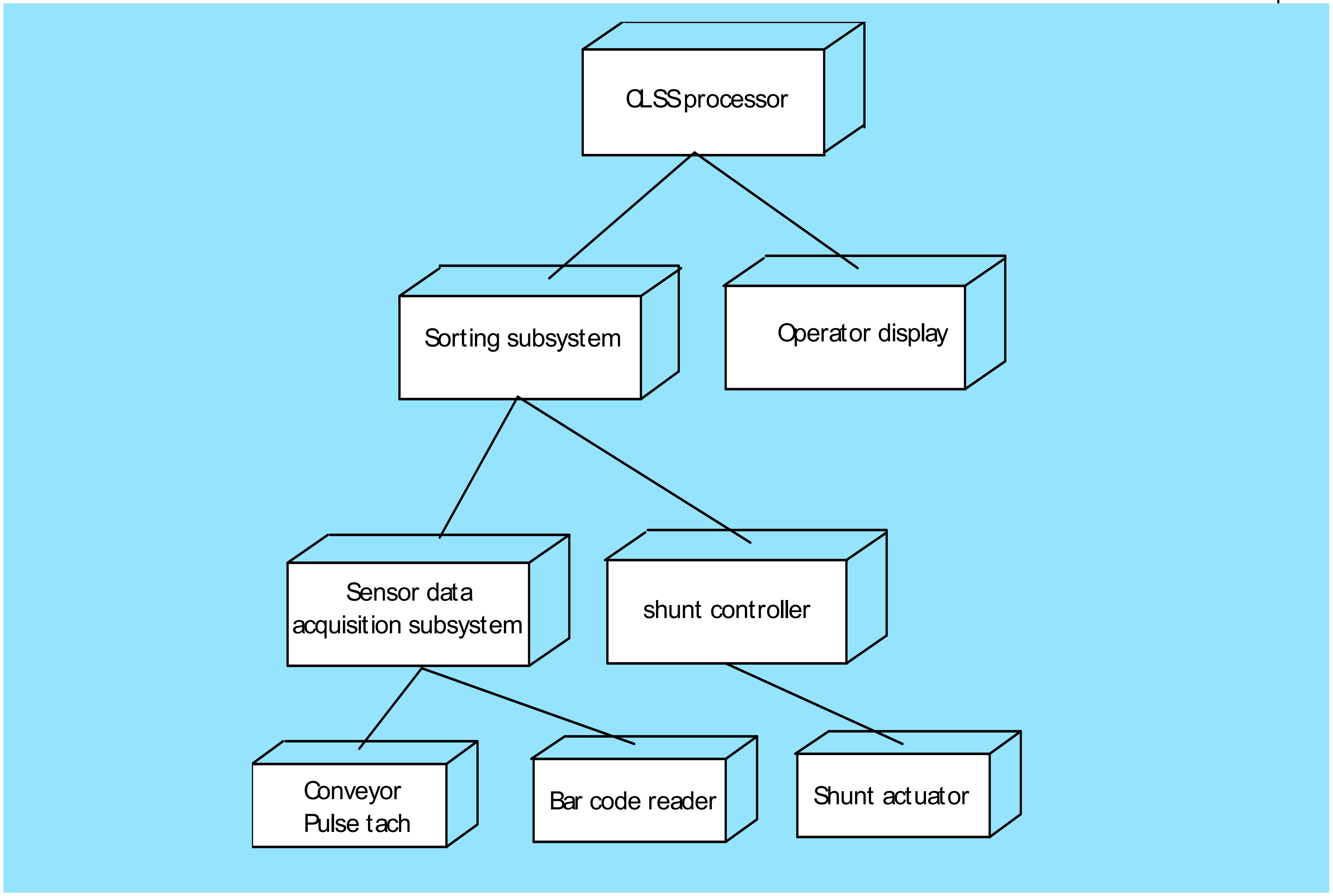
CLSS must be developed such that boxes moving along a conveyor line will be identified and sorted into one of six bins at the end of the line. The boxes will pass by a sorting station where they will be identified. Based on an identification number printed on the side of the box and a bar code, the boxes will be shunted into the appropriate bins. Boxes pass in random order and are evenly spaced. The line is moving slowly.

A desk-top computer located at the sorting station executes all CLSS software, interacts with the bar-code reader to read part numbers on each box, interacts with the conveyor line monitoring equipment to acquire conveyor line speed, stores all part numbers sorted, interacts with a sorting station operator to produce a variety of reports and diagnostics, sends control signals to the shunting hardware to sort the boxes, and communicates with a central factory automation system.

Is this D, R or S?

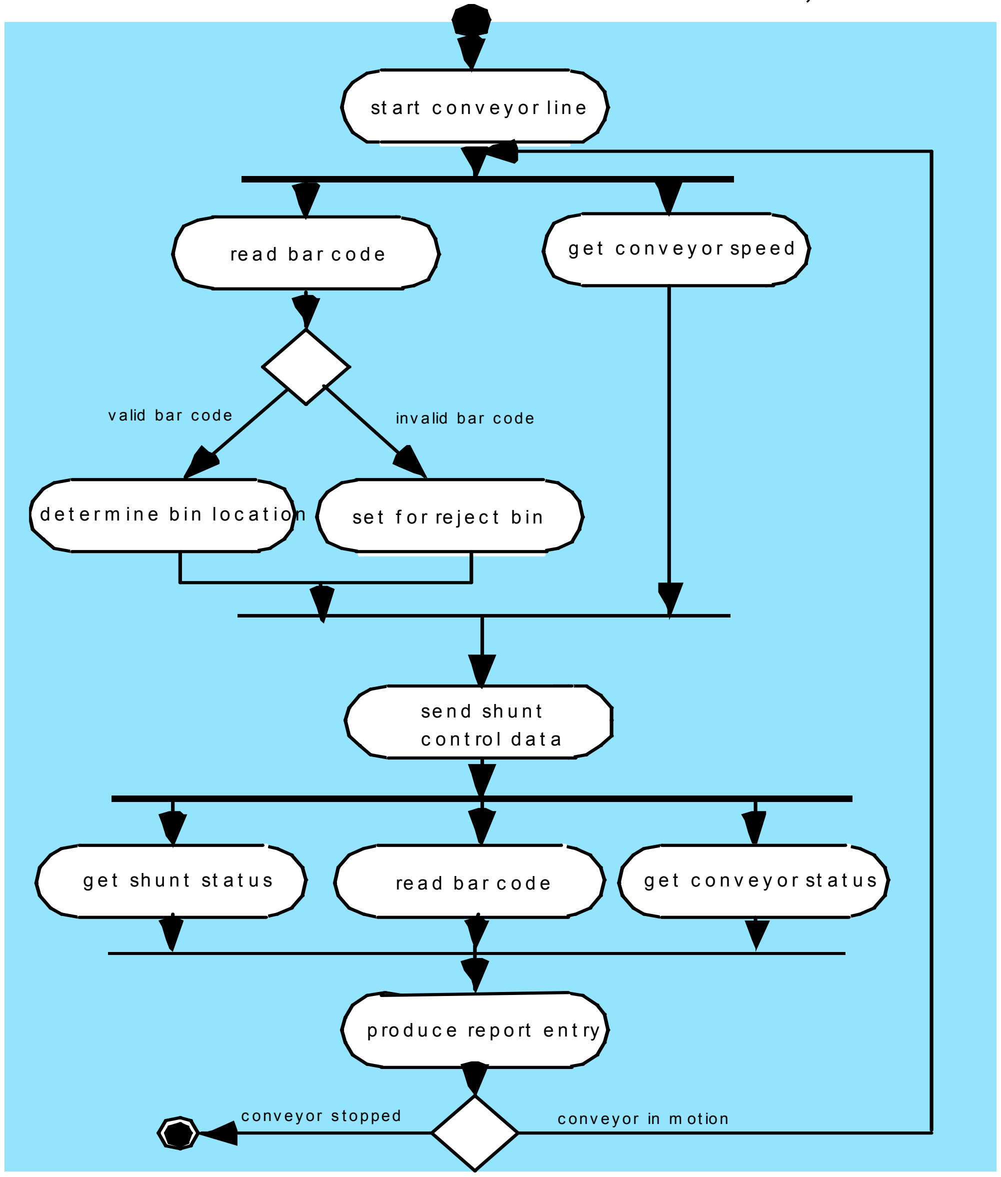
Deployment Diagram

Software Engineering: A Practitioner's Approach, 6/e

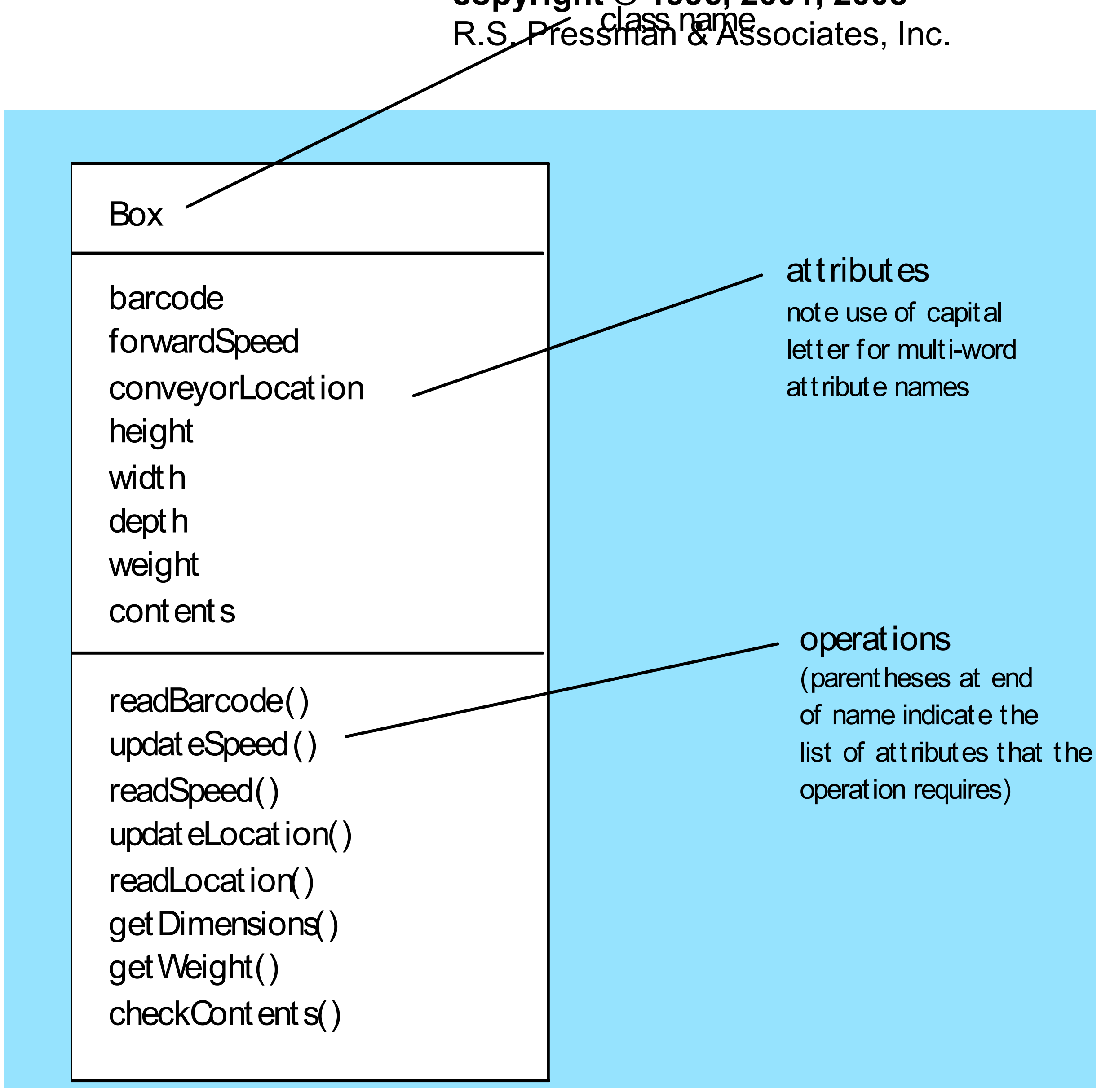


Activity Diagram

Software Engineering: A Practitioner's Approach, 6/e

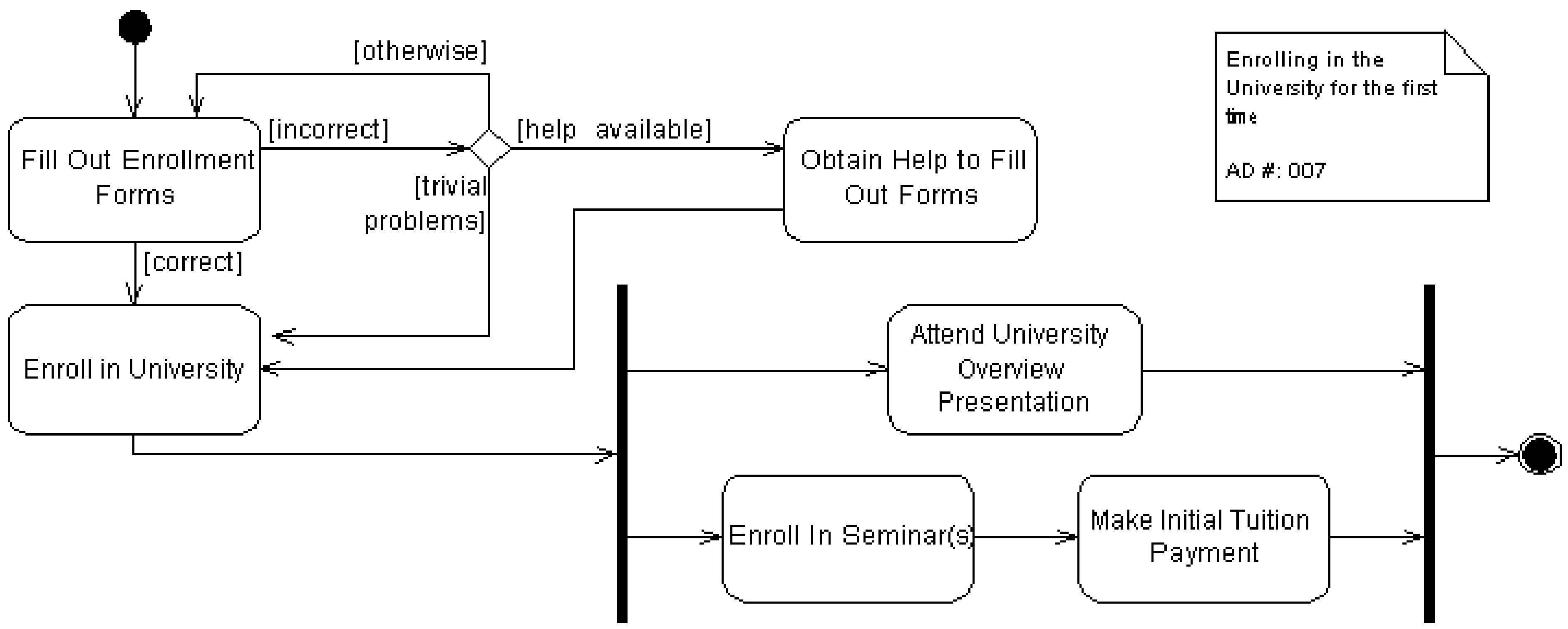


Class Diagram



Modeling a business process with a UML Activity Diagram

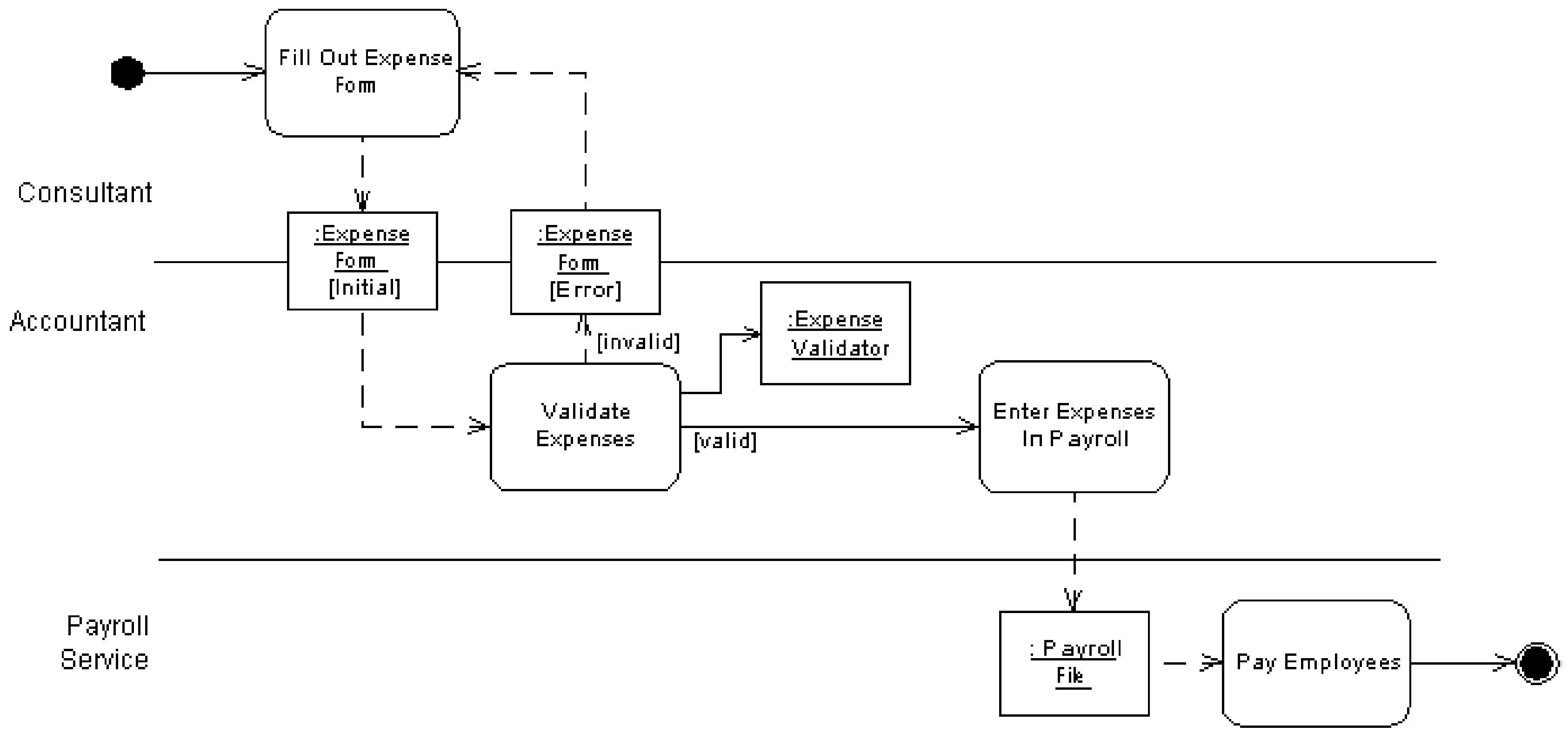
http://www.agilemodeling.com/style/activityDiagram.htm


[otherwise]
Fill Out Enrollment Forms
[incorrect]
[help available]
Obtain Help to Fill Out Forms
[trivial problems]
[correct]
Enroll in University
Attend University Overview Presentation
Enroll In Seminar(s)
Make Initial Tuition Payment
Enrolling in the University for the first time
AD #: 007

Modeling a business process with a UML Activity Diagram

http://www.agilemodeling.com/style/activityDiagram.htm

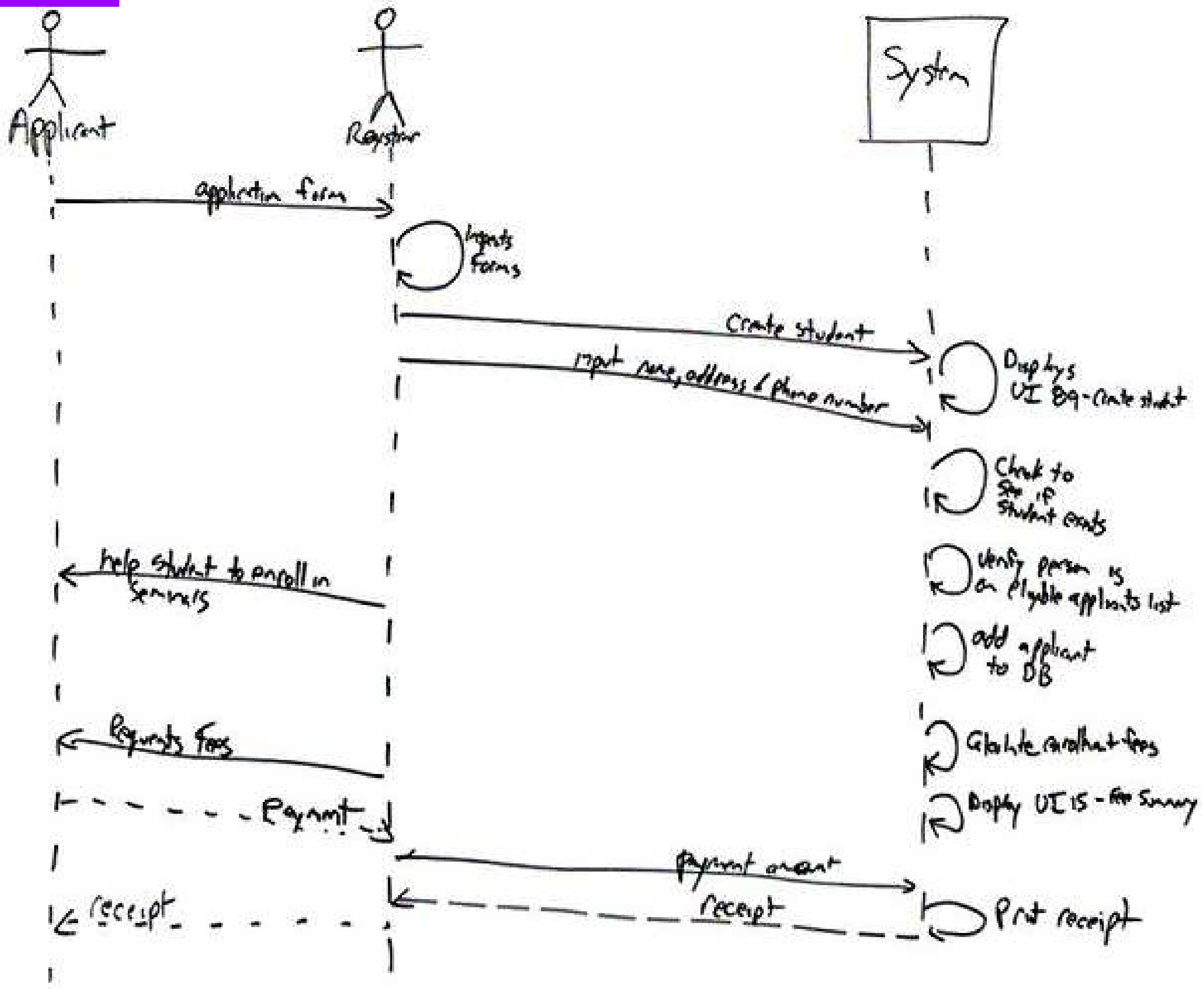
Submitting expenses



Modeling system-level interactions with a UML Sequence Diagram

http://www.agilemodeling.com/artifacts/sequenceDiagram.htm

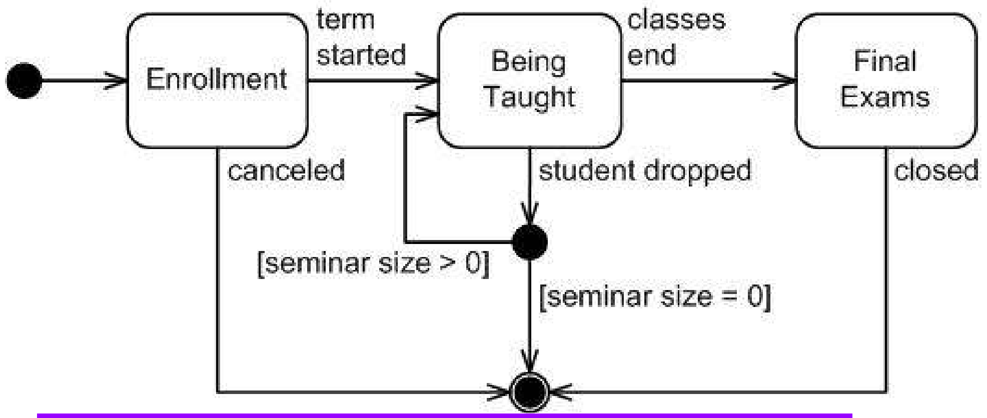
Student application


Applicant
Registrar
System
application form
Inspects forms
Create student
Displays UI 89 - Create student
input name, address & phone number
Check to see if Student exists
verify person is on eligible applicants list
add applicant to DB
help student to enroll in seminars
Requests fees
Calculate enrollment fees
Display UI 15 - Fee Summary
Payment
payment amount
receipt
receipt
Print receipt

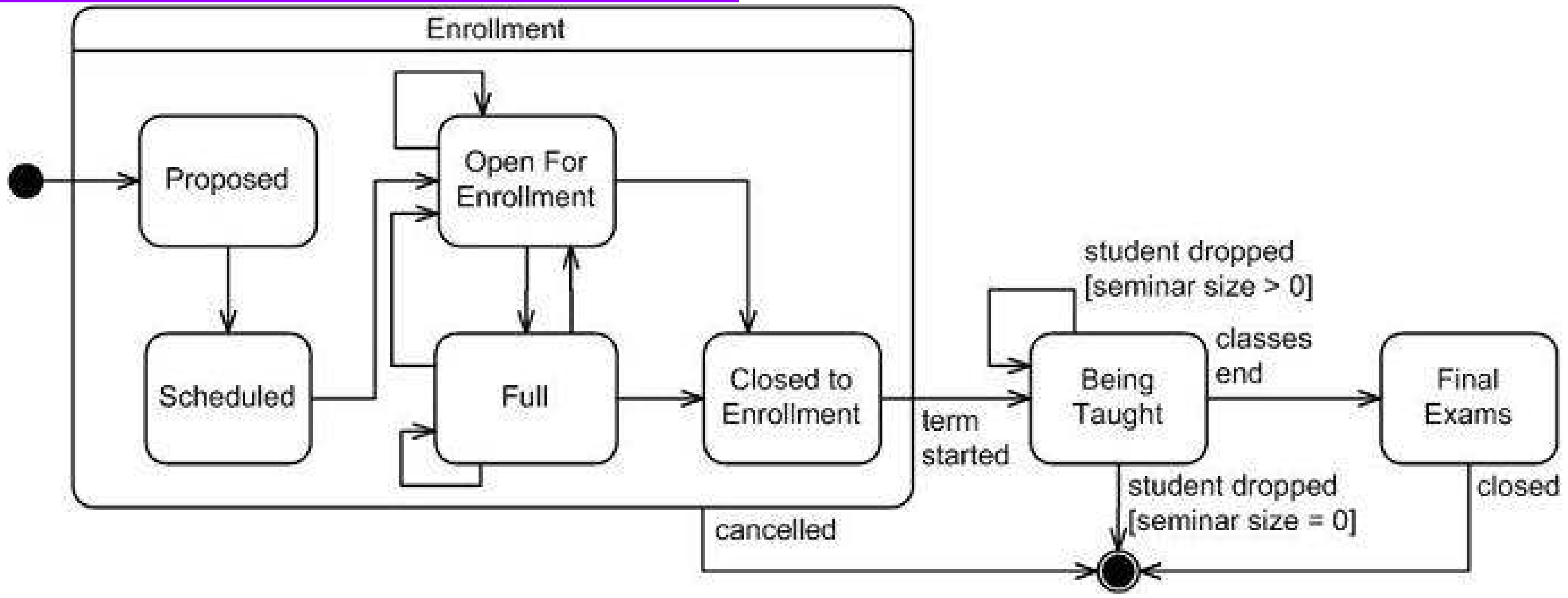
Modeling enterprise-level behavior with a UML State Transition Diagram

http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm

Top-level seminar lifecycle



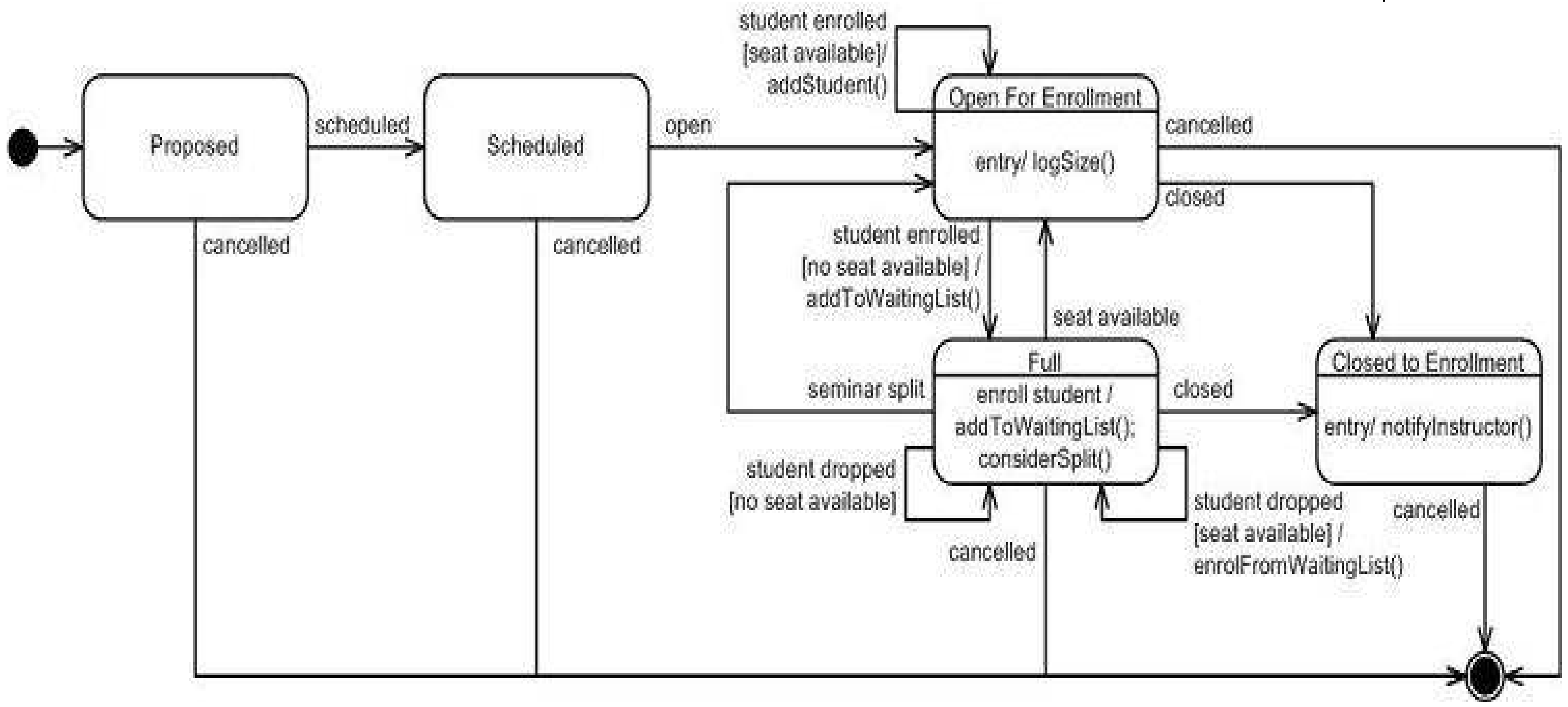
Intermediate-level seminar lifecycle



Modeling enterprise-level behavior with a UML State Transition Diagram

http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm

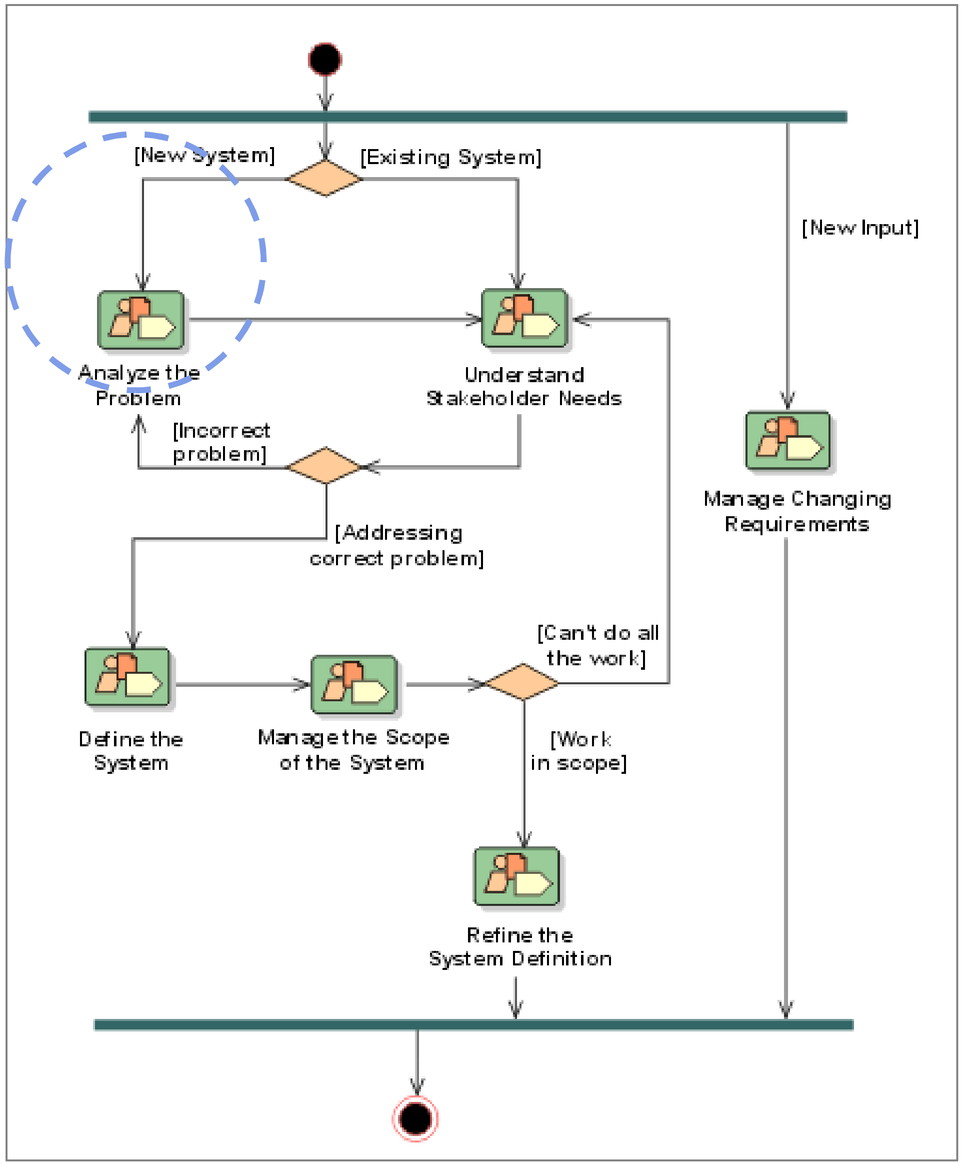
Lower-level seminar lifecycle during registration



student enrolled [seat available]/ addStudent()
Proposed
scheduled
Scheduled
open
Open For Enrollment
entry/ logSize()
cancelled
closed
cancelled
cancelled
student enrolled [no seat available] / addToWaitingList()
seat available
seminar split
Full
enroll student / addToWaitingList(); considerSplit()
closed
Closed to Enrollment
entry/ notifyInstructor()
student dropped [no seat available]
cancelled
student dropped [seat available] / enrolFromWaitingList()
cancelled

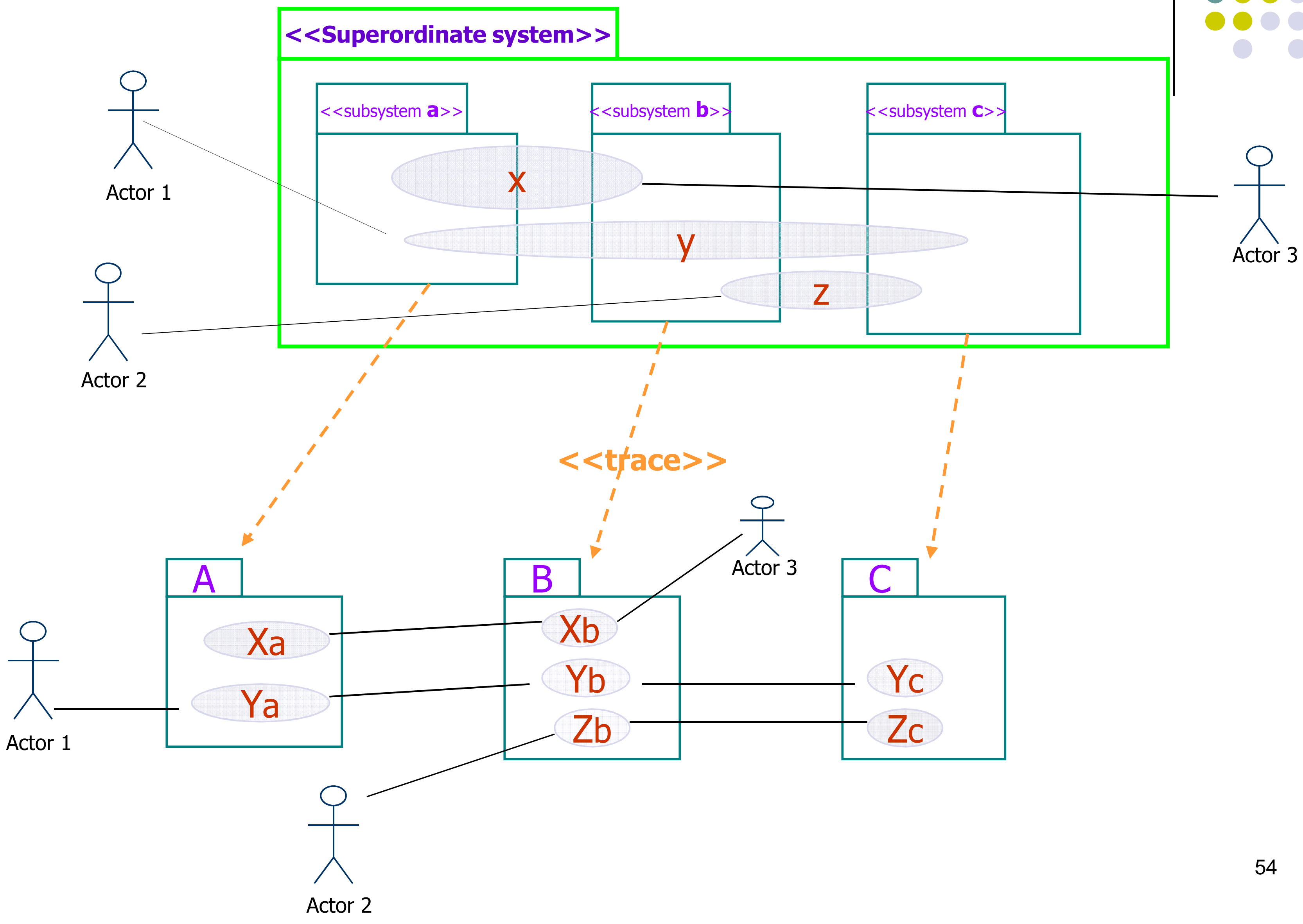
What Kind of Diagram is This?

For Process or Product?



[New System]
[Existing System]
[New Input]
Analyze the Problem
Understand Stakeholder Needs
[Incorrect problem]
[Addressing correct problem]
Manage Changing Requirements
[Can't do all the work]
Define the System
Manage the Scope of the System
[Work in scope]
Refine the System Definition

Use cases are allocated to design subsystems



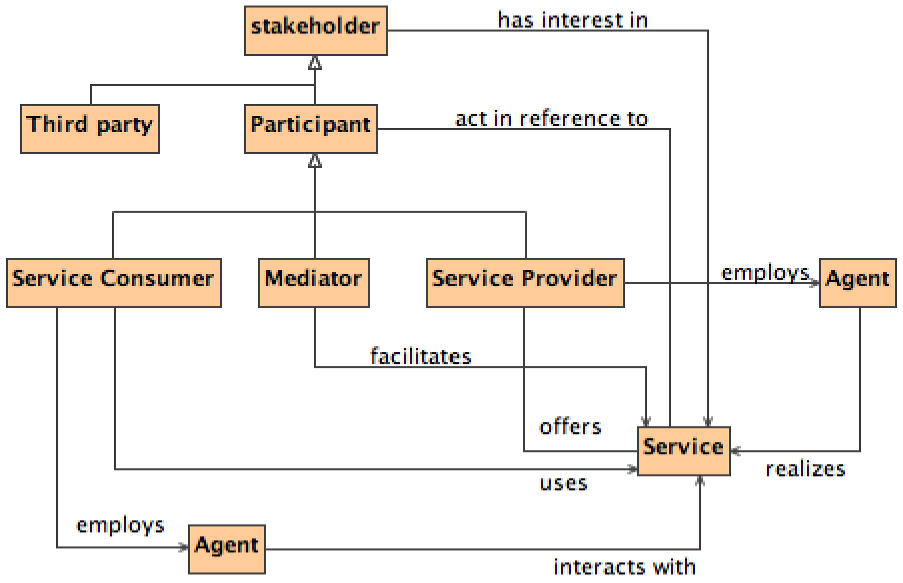
Service Oriented Architecture

Architecture Work::Reference Architectures::Service As Business View

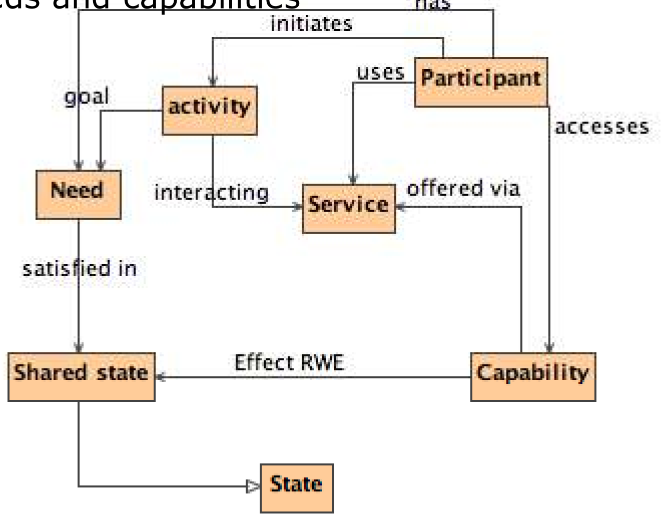
http://wiki.oasis-open.org/soa-rm/TheArchitecture/ServiceAsBusinessView

1.1. Viewpoint

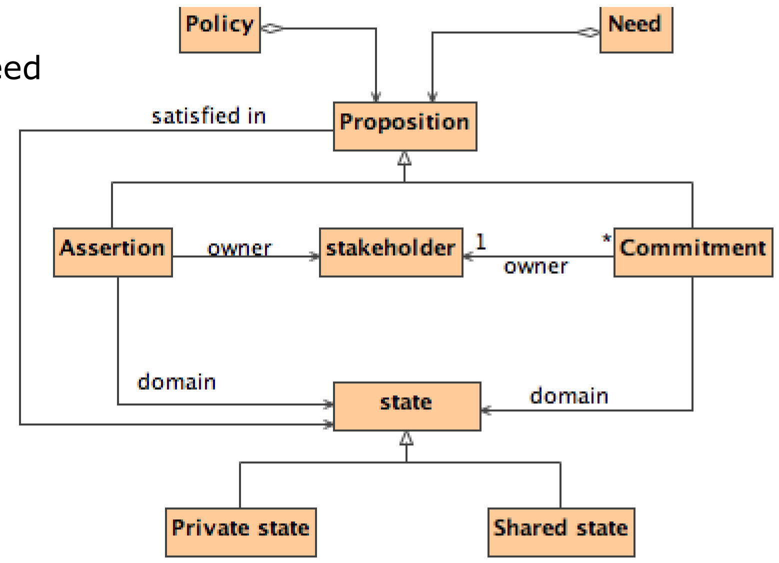
1.2. Stakeholders and Participants



1.3. Needs and capabilities



Need

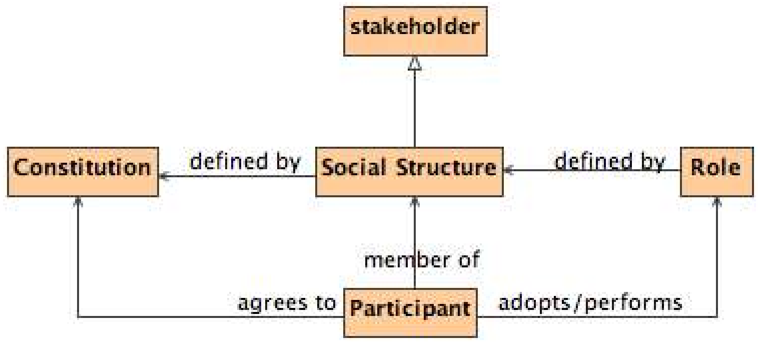


Service Oriented Architecture

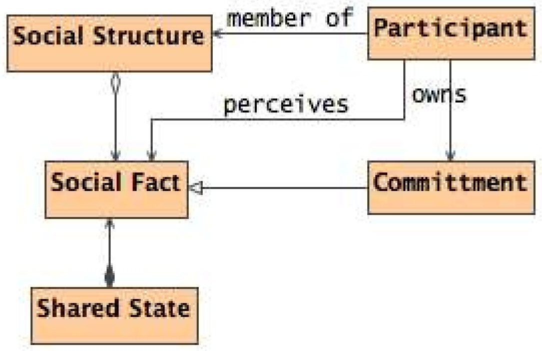
Architecture Work::Reference Architectures::Service As Business View

http://wiki.oasis-open.org/soa-rm/TheArchitecture/ServiceAsBusinessView

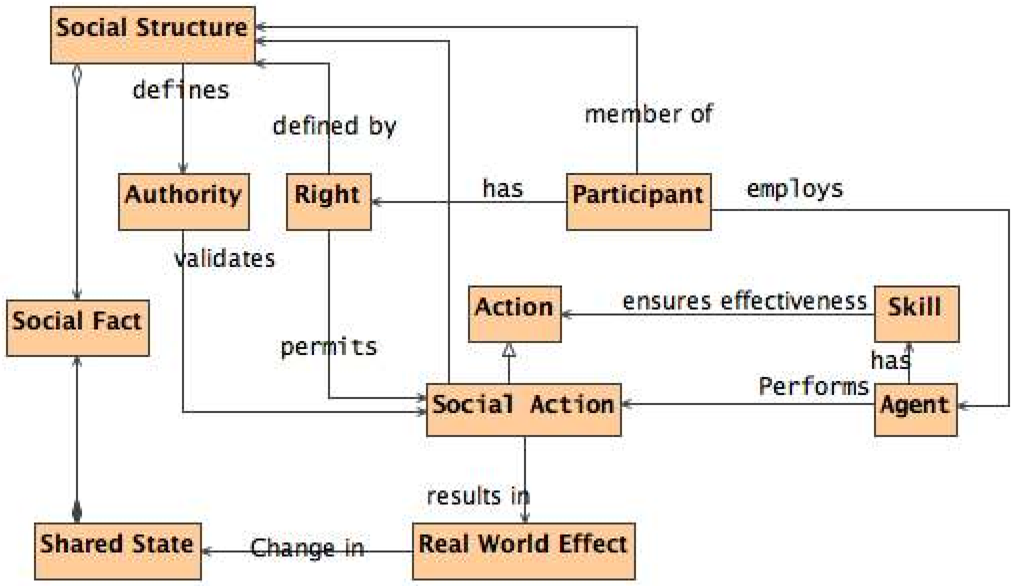
1.4. Social Structure



1.4.1. Shared state and social facts



1.5. Acting in a social context



1.6. Roles in Social Structures

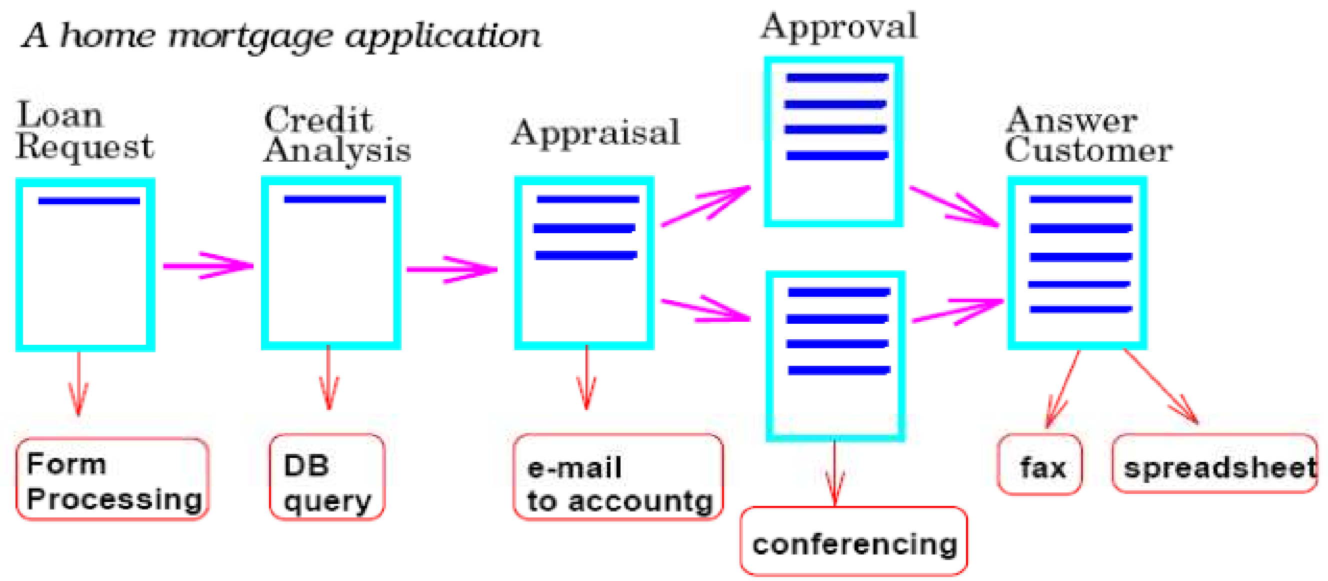
1.7. Governance and social structures

1.8. Tail piece

How would you model a workflow?

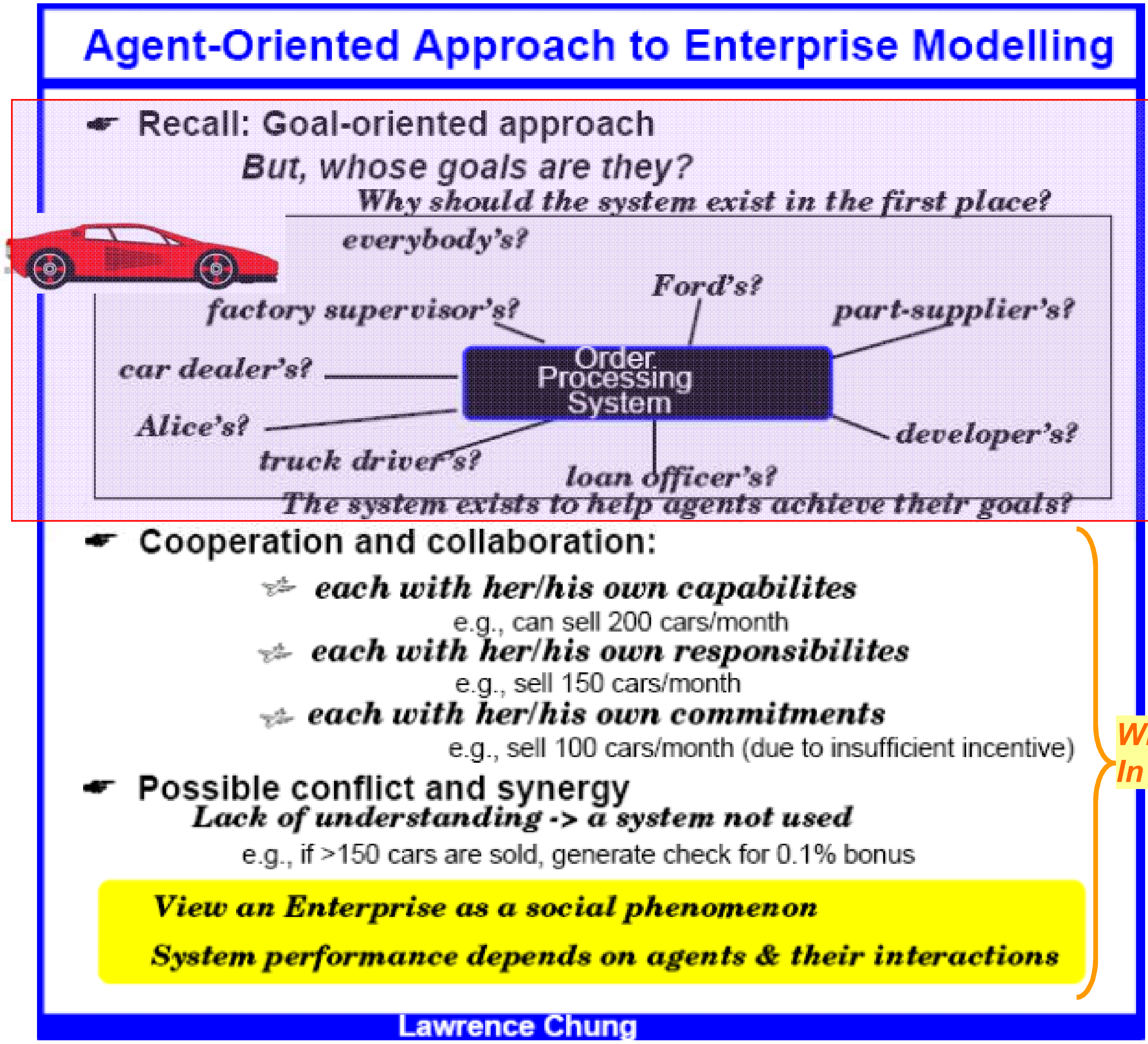
- Workflow

A home mortgage application



- Many office workers spend a high percentage of their time processing documents
- The "workflow river" carries the flow of work from port to port, value being added along the way
- The workflow definition must take into consideration:
  - *Routes* along which the object moves
  - *Rules* about what information is routed and when
    E.g., "If the loan is over $100, 000, send it to the supervisor within the next hour or else send it to the next hop"
  - *Roles* define job functions independently of the people who do it
    E.g., the "supervisor" role can be handled by users "Adam" and "Eve"

Lawrence Chung



Agent-Oriented Approach to Enterprise Modelling

- Recall: Goal-oriented approach

  But, whose goals are they?

  Why should the system exist in the first place?

  everybody's?

  factory supervisor's? ***Ford's?*** ***part-supplier's?***

  car dealer's? Order Processing System

  Alice's? ***developer's?***

  truck driver's? ***loan officer's?***

  The system exists to help agents achieve their goals?

- Cooperation and collaboration:
  - ***each with her/his own capabilites***

    e.g., can sell 200 cars/month
  - ***each with her/his own responsibilites***

    e.g., sell 150 cars/month
  - ***each with her/his own commitments***

    e.g., sell 100 cars/month (due to insufficient incentive)

- Possible conflict and synergy

  Lack of understanding -> a system not used

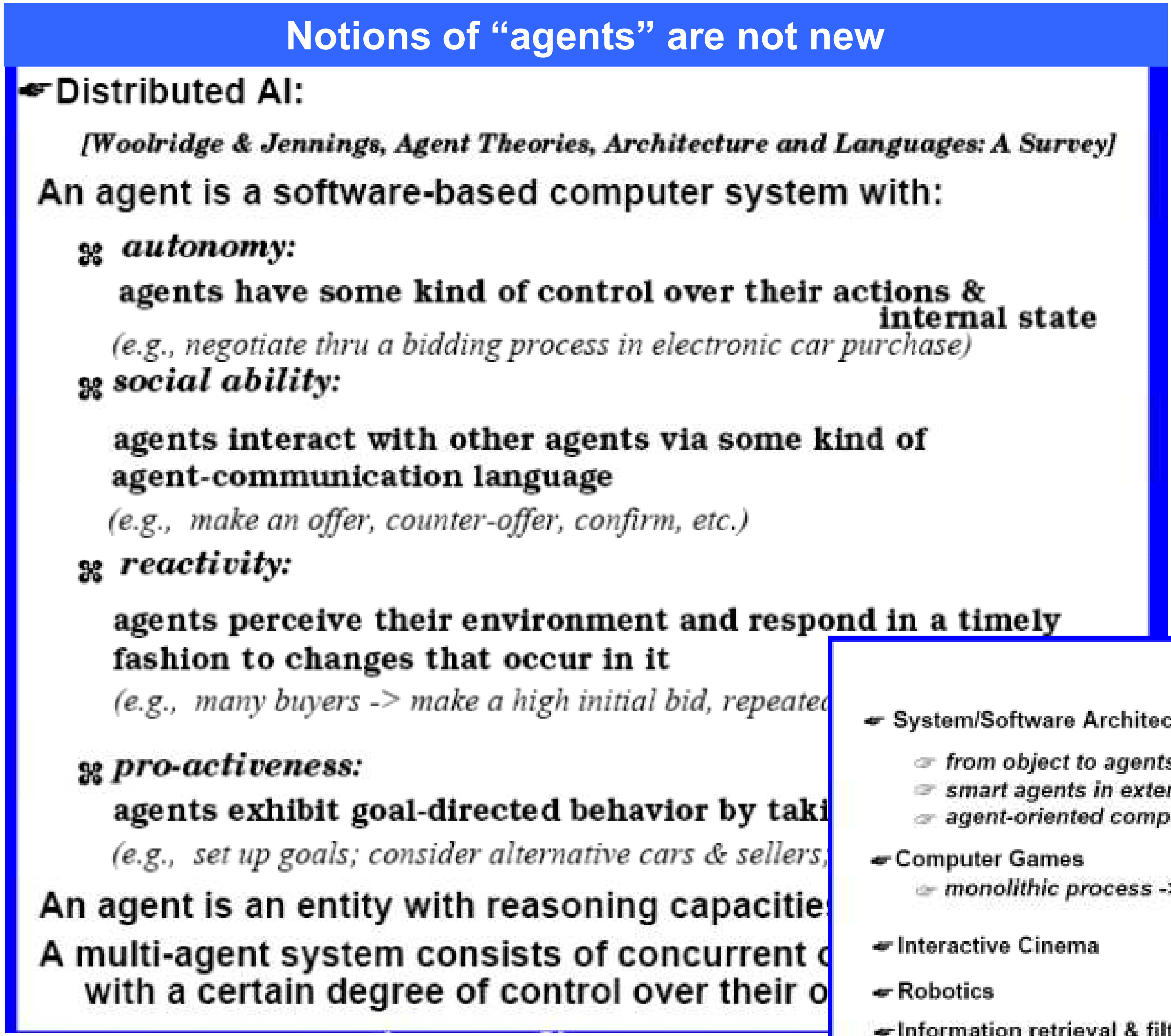
  e.g., if >150 cars are sold, generate check for 0.1% bonus

View an Enterprise as a social phenomenon

System performance depends on agents & their interactions

Lawrence Chung

What's the role of the machine
In the social phenomenon?



Notions of “agents” are not new

☛Distributed AI:

[Woolridge & Jennings, Agent Theories, Architecture and Languages: A Survey]

An agent is a software-based computer system with:

⌘ ***autonomy:***

agents have some kind of control over their actions & internal state

(e.g., negotiate thru a bidding process in electronic car purchase)

⌘ ***social ability:***

agents interact with other agents via some kind of agent-communication language

(e.g., make an offer, counter-offer, confirm, etc.)

⌘ ***reactivity:***

agents perceive their environment and respond in a timely fashion to changes that occur in it

(e.g., many buyers -> make a high initial bid, repeated

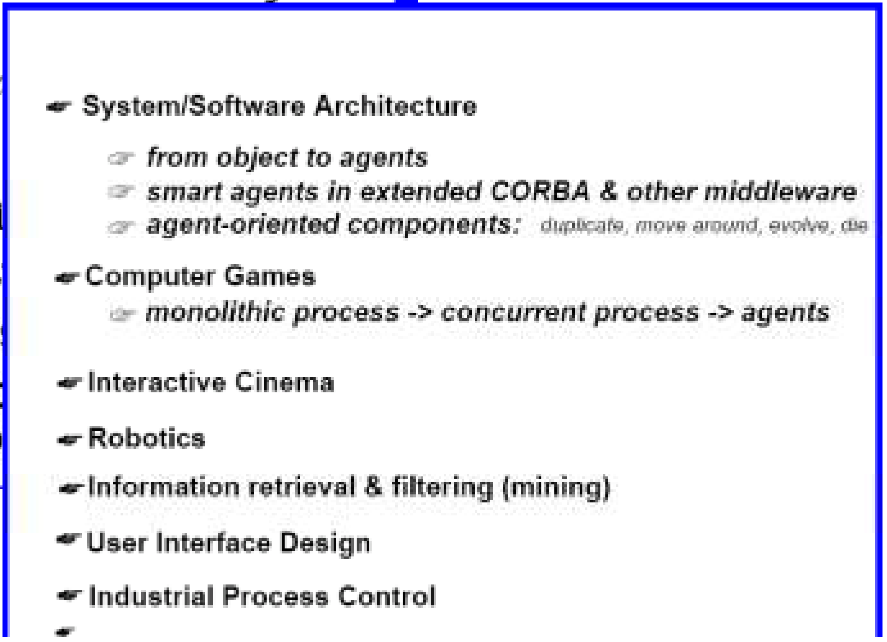
⌘ ***pro-activeness:***

agents exhibit goal-directed behavior by taki

(e.g., set up goals; consider alternative cars & sellers,

An agent is an entity with reasoning capacitie

A multi-agent system consists of concurrent c
with a certain degree of control over their o



- System/Software Architecture
  - *from object to agents*
  - *smart agents in extended CORBA & other middleware*
  - *agent-oriented components:* duplicate, move around, evolve, die
- Computer Games
  - *monolithic process -> concurrent process -> agents*
- Interactive Cinema
- Robotics
- Information retrieval & filtering (mining)
- User Interface Design
- Industrial Process Control
-

Notions of “agents” in SE and RE

☛ Jacobson:

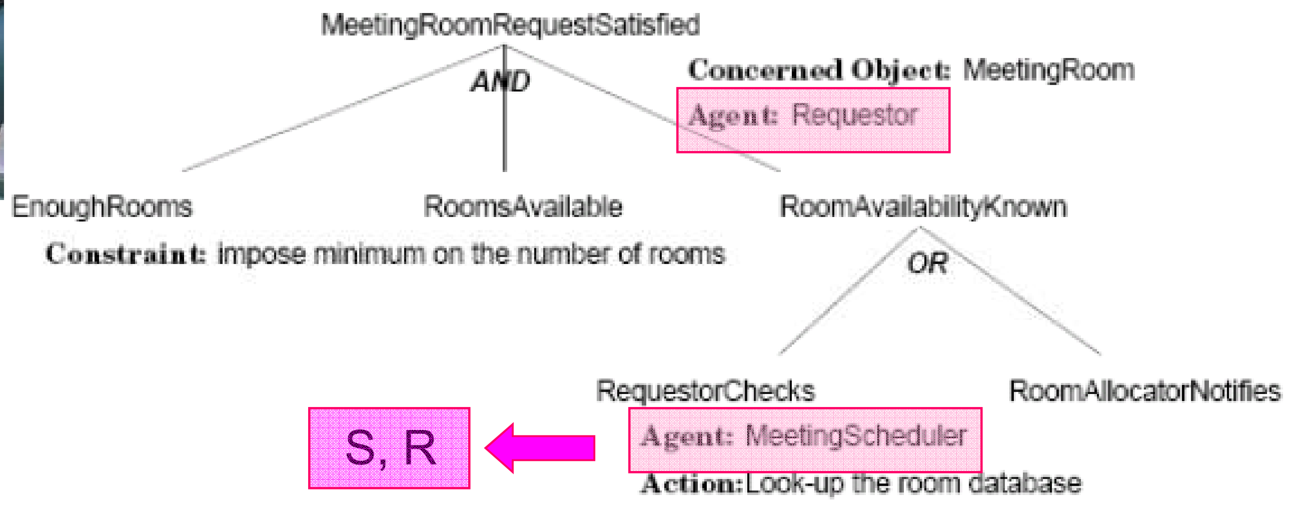
Actors are something "external" to the system.
An actor is a way of modelling "users" of the system (human or non-human), not the system itself. Actors are used to find and describe use cases primarily, and not for making system models.

In UML,
- *business actors,*
- *business workers,*
- *(system) actors*

☛ Recent RE community (some):

Agents can be a part of a system, or the whole system itself.
The distinction between system external and internal is a matter of choice. Agents are active entities/objects in the sense that they communicate with other agents possibly within the system.

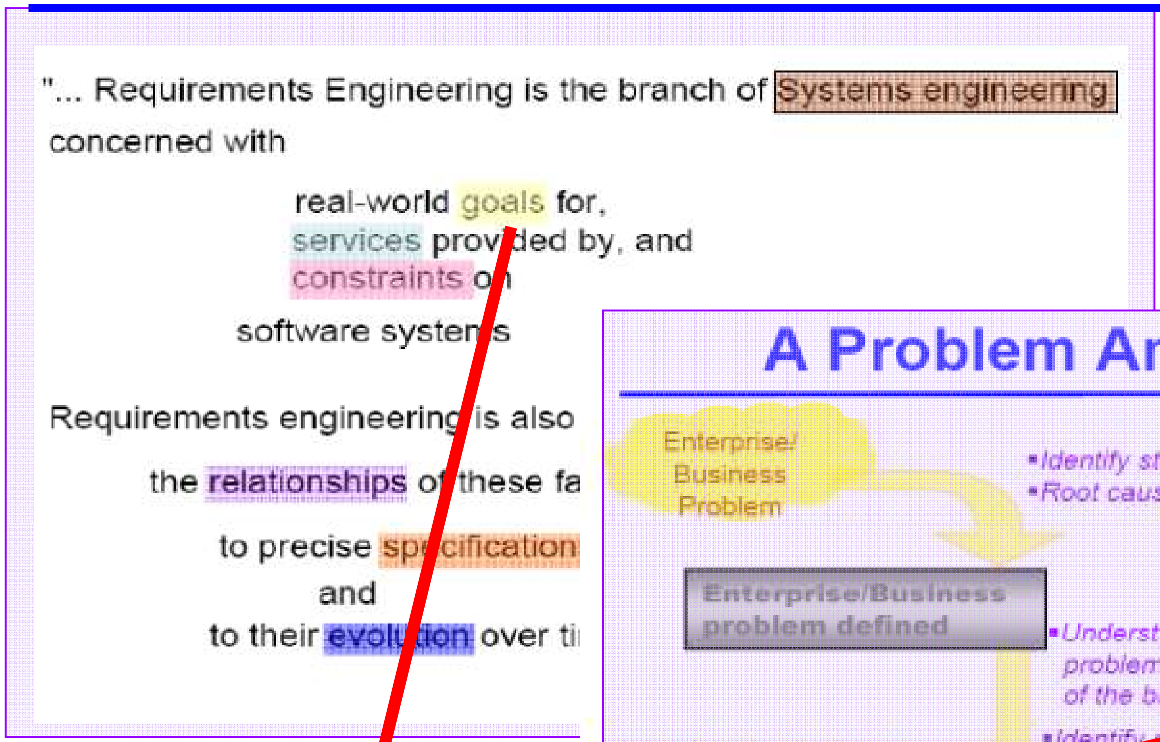
e.g., KAOS: *explicit representation of agents, roles, goals, tasks and resources*


MeetingRoomRequestSatisfied
AND
Concerned Object: MeetingRoom
Agent: Requestor
EnoughRooms
RoomsAvailable
RoomAvailabilityKnown
Constraint: impose minimum on the number of rooms
OR
RequestorChecks
RoomAllocatorNotifies
S, R
Agent: MeetingScheduler
Action: Look-up the room database

A goal-oriented approach

Refresher) Elicit?

Goal-oriented Requirements Elicitation



"... Requirements Engineering is the branch of Systems engineering concerned with

real-world goals for,
services provided by, and
constraints on

software systems

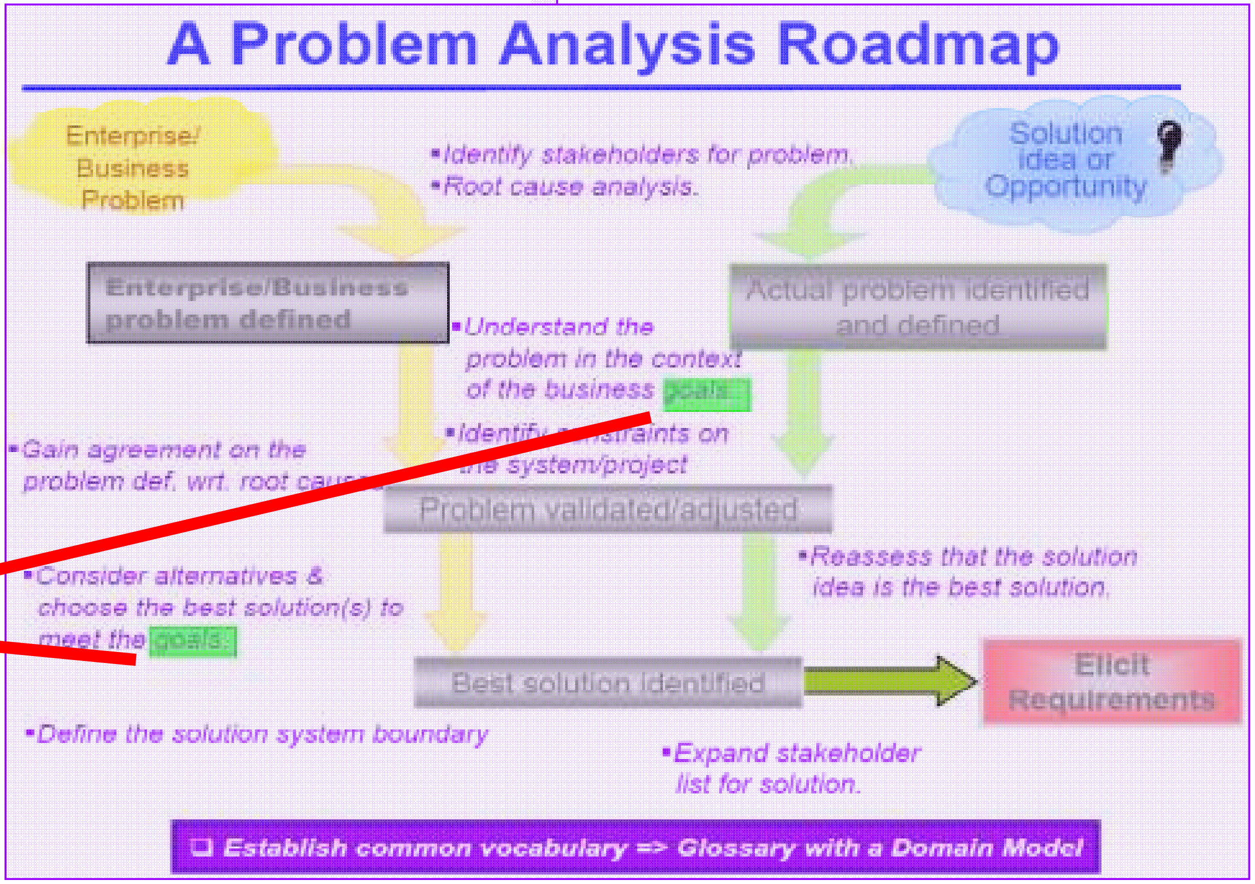
Requirements engineering is also

the relationships of these fa

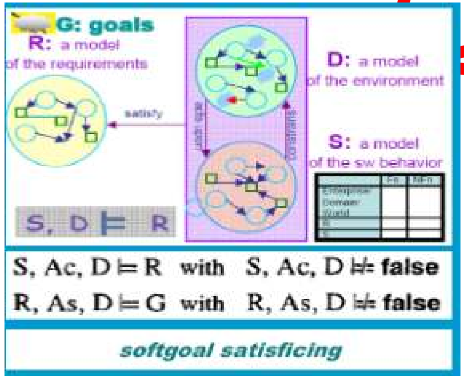
to precise specification

and

to their evolution over ti



A Problem Analysis Roadmap
Enterprise/ Business Problem
•Identify stakeholders for problem.
•Root cause analysis.
Solution idea or Opportunity
Enterprise/Business problem defined
Actual problem identified and defined
•Understand the problem in the context of the business goals
•Identify constraints on the system/project
•Gain agreement on the problem def. wrt. root cause
Problem validated/adjusted
•Consider alternatives & choose the best solution(s) to meet the goals.
•Reassess that the solution idea is the best solution.
Best solution identified
Elicit Requirements
•Define the solution system boundary
•Expand stakeholder list for solution.
Establish common vocabulary => Glossary with a Domain Model



G: goals
R: a model of the requirements
satisfy
D: a model of the environment
S: a model of the sw behavior
S, D ⊨ R

$S, Ac, D \models R$ with $S, Ac, D \not\models$ **false**

$R, As, D \models G$ with $R, As, D \not\models$ **false**

softgoal satisficing

Appendix II: SysML

Brief History of SysML

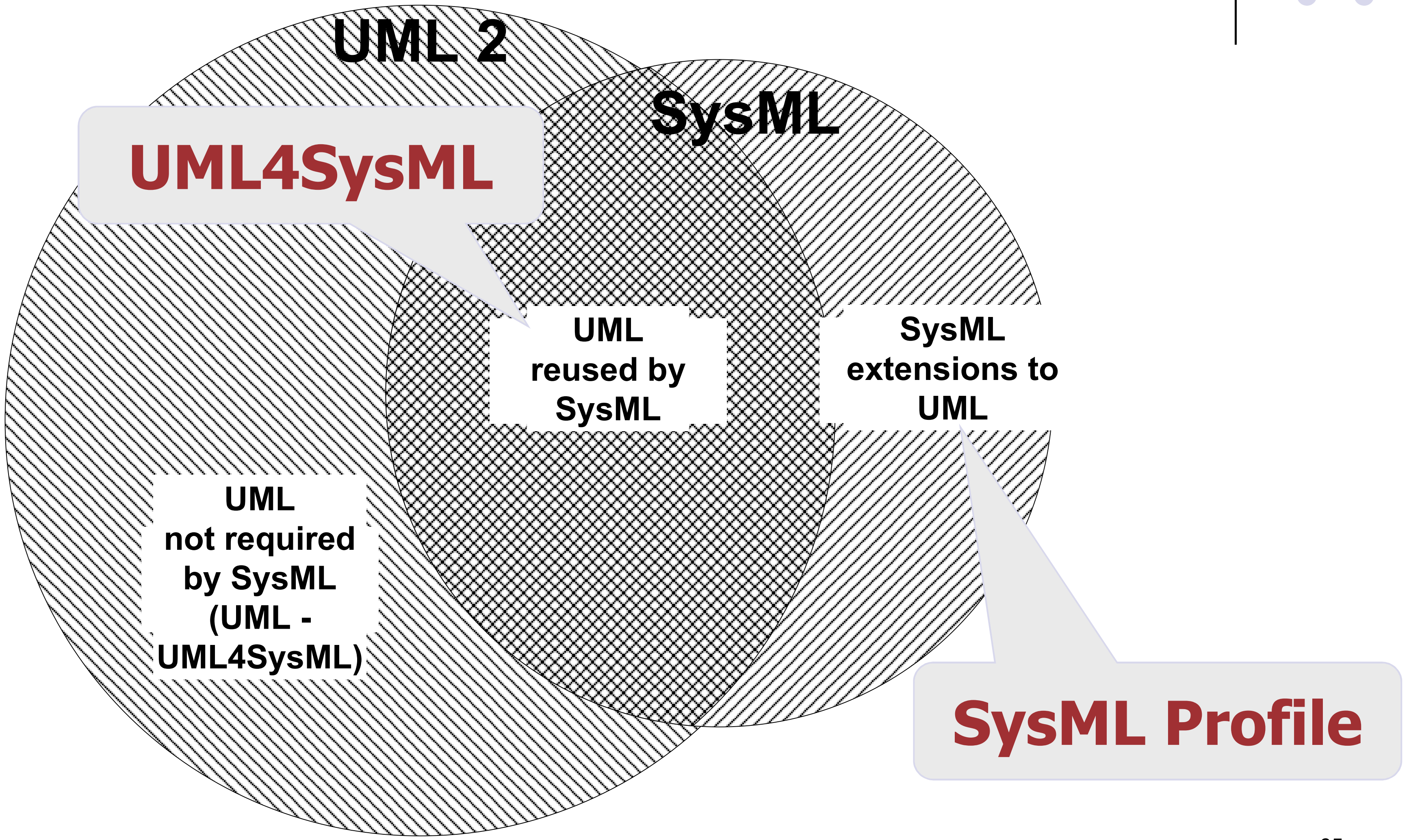
- UML for System Engineering RFP: March 2003, with INCOSE
  Initial draft, January 2004
- SysML Specification v1.0: July 2006, adopted by OMG
- SysML Specification v1.1: June 2008, adopted by OMG
- SysML Specification v1.2: June 2010, adopted by OMG
  http://www.omg.org/spec/SysML/1.2/

A Brief Synopsis of SysML

- A graphical modeling language for UML for Systems Engineering
- A UML Profile that represents a subset of UML 2 with extensions
- Supports the specification, analysis, design, verification and validation of systems that include hardware, software, data, personnel, procedures, and facilities
- Supports model and data interchange via XMI and the evolving AP233 standard (in-process)

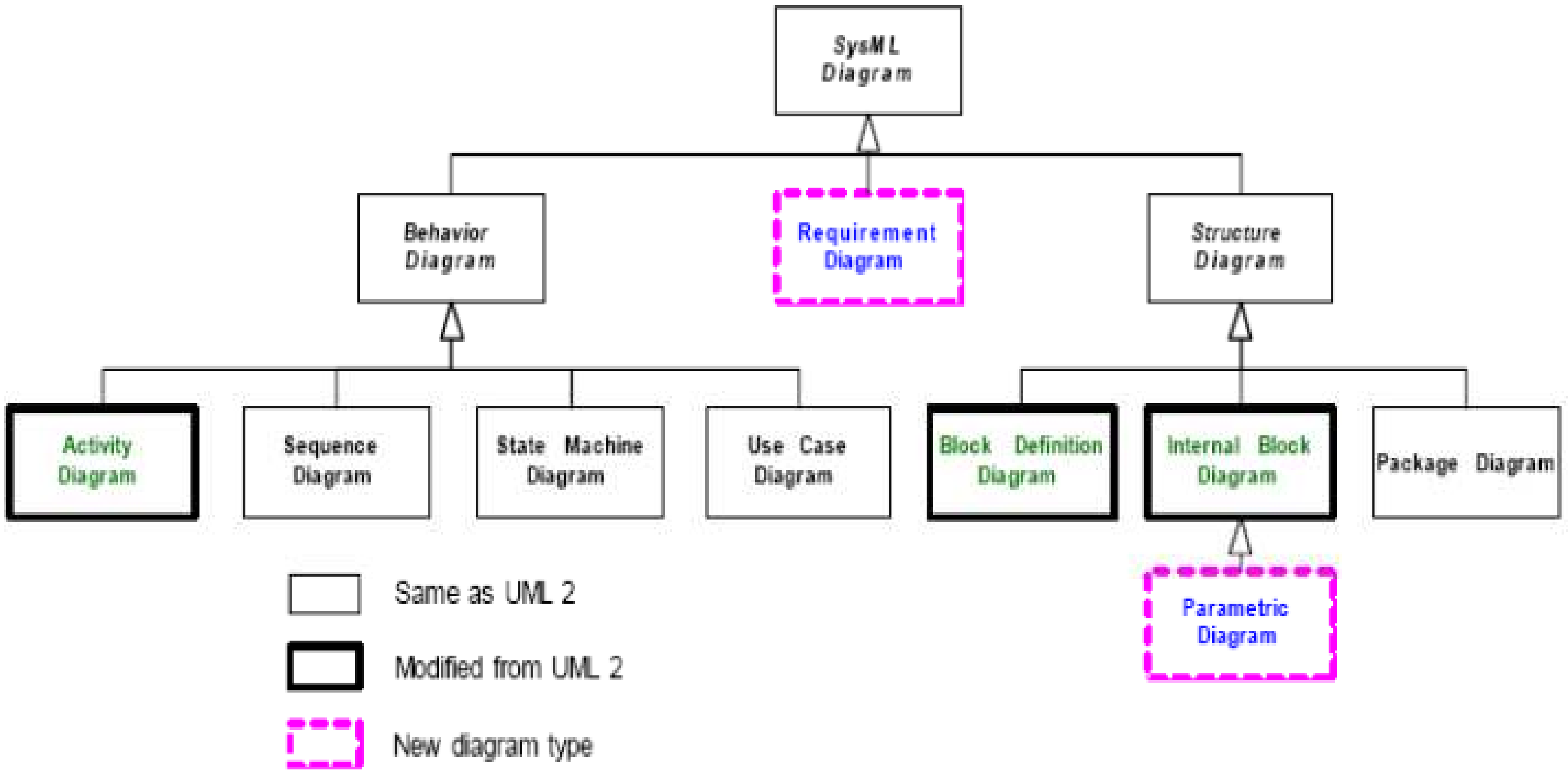
So, SysML is also for Model Driven SE

Relationship Between SysML and UML



ihase.wikispaces.com/file/view/**SysML**+Introduction3.**ppt**

SysML Diagram Taxonomy



Major Extensions to UML 2

- **New Diagram Types**
  - **Requirement Diagram (visual modeling of requirements)**
  - **Parametric Diagram (showing relations between parameters)**
- Structure Diagram
  - Block Definition Diagram (based on UML class diagram with blocks instead of classes)
  - Internal Block Diagram (based on UML composite structure diagram with restrictions and extensions)
- Activity Diagram
  - extensions for continuous flow modeling
  - extensions to support disabling control and control operators.
  - accommodate needs of Extended Functional Flow Block Diagrams (EFFBDs)