CSCE 491 –Lectures 3-4

Introduction to UML by Example

2002/8/28

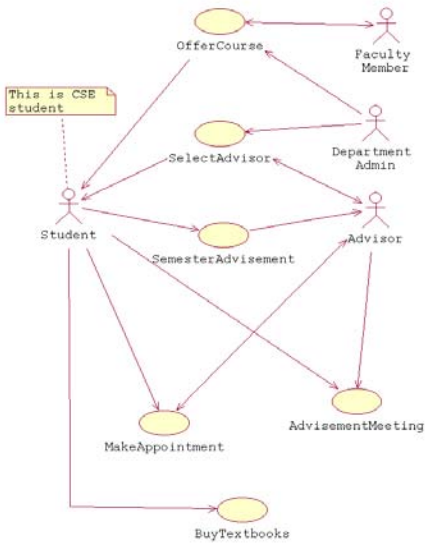
Department of Computer Science & Engineering
Fall 2002

Lectures 3-5 - Outline

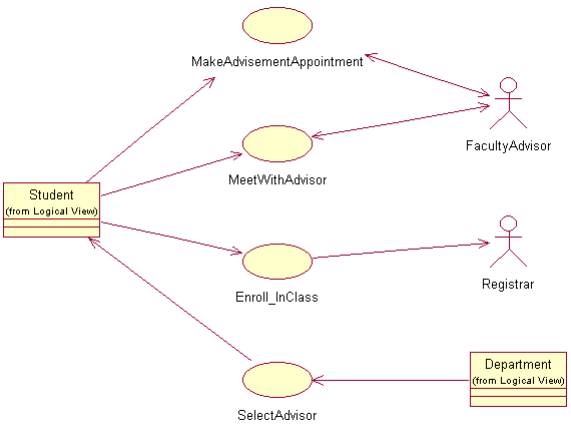
- Introduction to Unified Modeling Language.
  - We will use UML as the means to communicate domain specifications, and to discuss the concepts and architecture for an arbitrary domain of interest.
  - We will start with a description of the various notations, their symbol sets and syntax, and the underlying semantics communicated via conceptual modeling.
- Discussion of the UML Example (attached).
  - Use Case diagrams for creating inventory of operations and transactions in the domain of interest, identifying the key "actors" and their roles.
  - Class diagrams for modeling individual concepts and their "associations" in the domain.
  - State diagrams for modeling domain behaviors associated with a given concept (class). The position we take is that all classes of interest will have the notion of a "lifecycle"—a series of states through which each occurrence (i.e., instance) of a conceptual class will pass through between its "birth" and "death".
  - Sequence diagrams for key interactions between the various conceptual components of the system being created to realize the use cases in the Use Case diagram.
- Discussion of the Student-Course-Advisor example.
  - We want to develop a model of something close and familiar to us all—it is a process, with specific tasks, involving "transactions" between "actors".
  - We'll model the key conceptual entities, their associations with one another, and the core state sequences through which they pass.
  - Two separate, yet related, aspects of the domain: Course Offering development, Student Enrollment and Advisement.

Student Advisement – Core Tasks

- Use Case Model
  - Inventory of basic functions, or tasks, in the domain.
  - Actors: Student, Faculty Member, Advisor, Department Admin. Note, these are mostly roles assumed by people in the domain.
  - Each Use Case will be detailed using the Sequence diagram notation, or as a text description of the sequence of actions that are part of the Use Case.

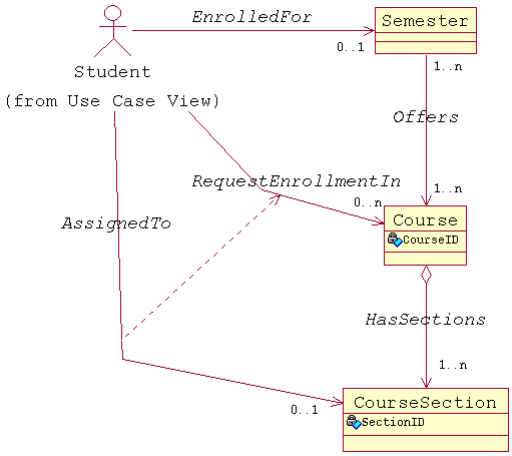


Student Advisement – Use Cases

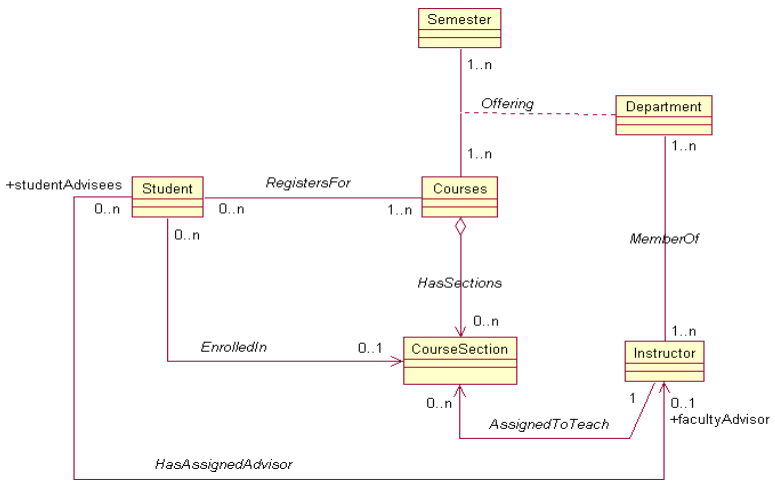


Student Advisement – Primary Concepts-1

- Class Model.
  - Model the key concepts and associations between them.
  - Actors also can serve as concepts in the domain (Rose uses same symbol if declared as an Actor first).
  - The "implication" exists between the associations *AssignedTo* and *RequestEnrollementIn*, meaning that a Student cannot participate in the former relationship without first participating in the latter one.
  - Multiplicities the set cardinalities between instances in one entity set and the other one (read by looking at the direction of the arrow).
  - Note that associations can be unidirectional or bi-directional.

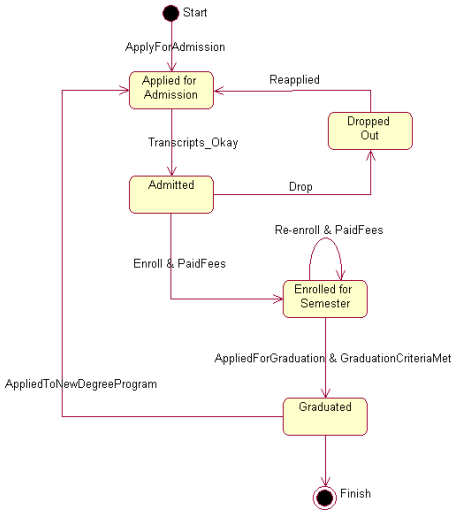


Student Advisement – Primary Concepts-2



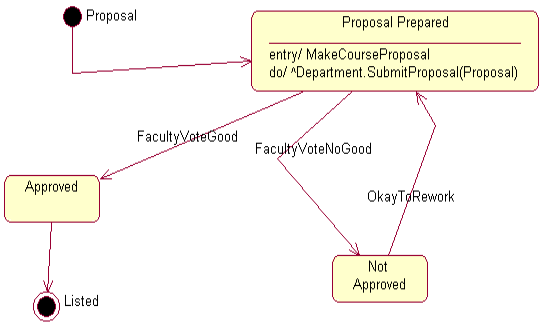
Course Offering – Student Lifecycle

- A concept's relevant lifecycle can be abstracted so as to control complexity:
  - Conceptual lifecycle of an entity may be inherently complex, with many states and many possible transitions.
  - Abstraction, or the hiding or deferral of details, is the means to manage complexity of a model.
  - We can apply abstraction in a state machine model through (1) abstracting away details that are not germane to the problem at hand, or (2) push details of lifecycle complexity into the model through the use of hierarchy or concurrency of FSM models.
  - Example: for the Student class, we are interested in the main sequence of states associated with progression through a degree program. There are many other details omitted, or abstracted away. These could be added through other means.



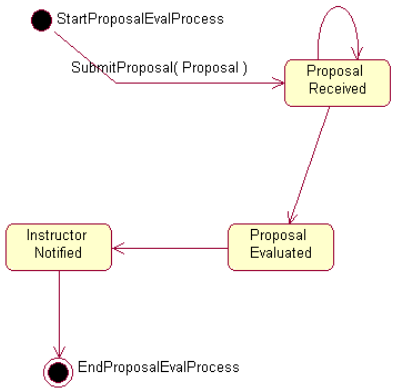
Course Offering – Proposal Lifecycle

- A concept/class may have a lifecycle for each instance.
  - Lifecycle modeled as a sequence of states, represented in the UML State Chart diagram.
  - Start state – where the FSM starts its behavior ("birth" state).
  - End state – double circle for where the FSM ends its behavior during its execution lifetime ("death" state).
  - Each state corresponds to a discrete and stable place where the concept can reside (modeled as the value of one or more state variables).
  - Example: a course proposal as an artifact with an independent lifecycle.



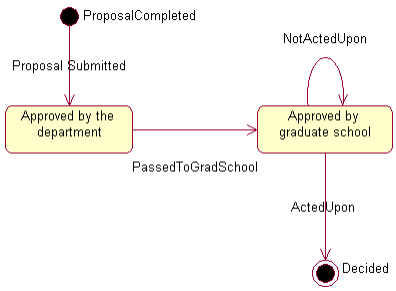
Course Offering – Department Lifecycle

- A state chart may model the specific behaviors associated with a particular view of the concept under consideration:
  - The state-based lifecycle doesn't model all aspects of the concept's behavior, only that which may be relevant.
  - There could be multiple lifecycles, in which case, the class should be broken apart and each lifecycle of interest modeled separately with the class.
  - At the conceptual level, we don't model the specific actions associated with being in a state.
  - However, we may represent the specific actions and events that occur on transition from one state to another (abstractly).

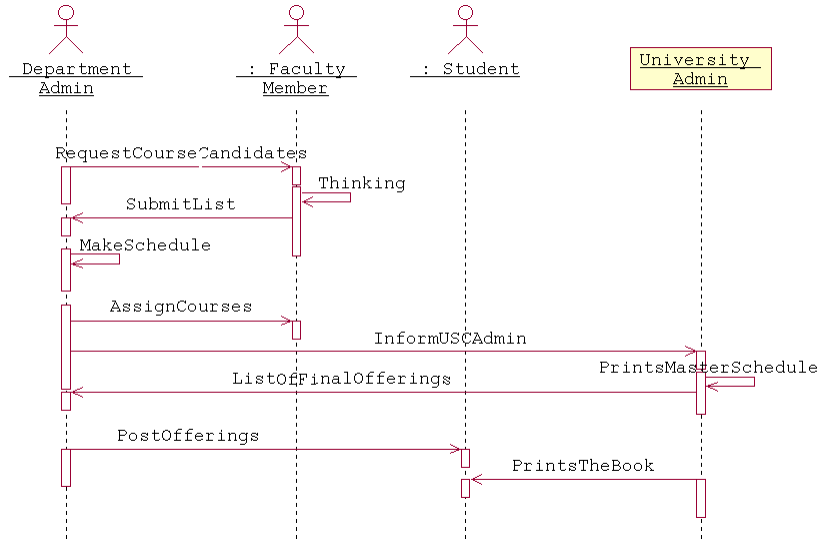


Course Offering – Instructor Lifecycle

- A concept's relevant lifecycle may be complex or may be simple:
  - Depending on the point of view for modeling a lifecycle of interest, there may be a complex model, or there may be a simple one.
  - Complexity depends on the context of the modeling activity.
  - In this scenario, we have a very simple lifecycle for the instructor relative to course proposals. A more involved lifecycle might exist for other activities/behaviors for the Instructor: such as Teaching, Advising, etc.
  - Each of these lifecycles would be associated with different "roles" being played by the concept in the context of associations in the domain (in this case, and Instructor playing the role of Proposal Submitter, as opposed to Advisor, etc.



Course Offering – Primary Sequence



Course Offering – Alternate Sequence