

Domain Modeling

Domains and Domain Models

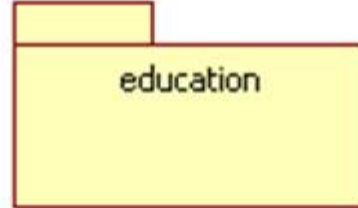
A domain is a collection of related concepts, relationships, and workflows.

Examples of domains include:

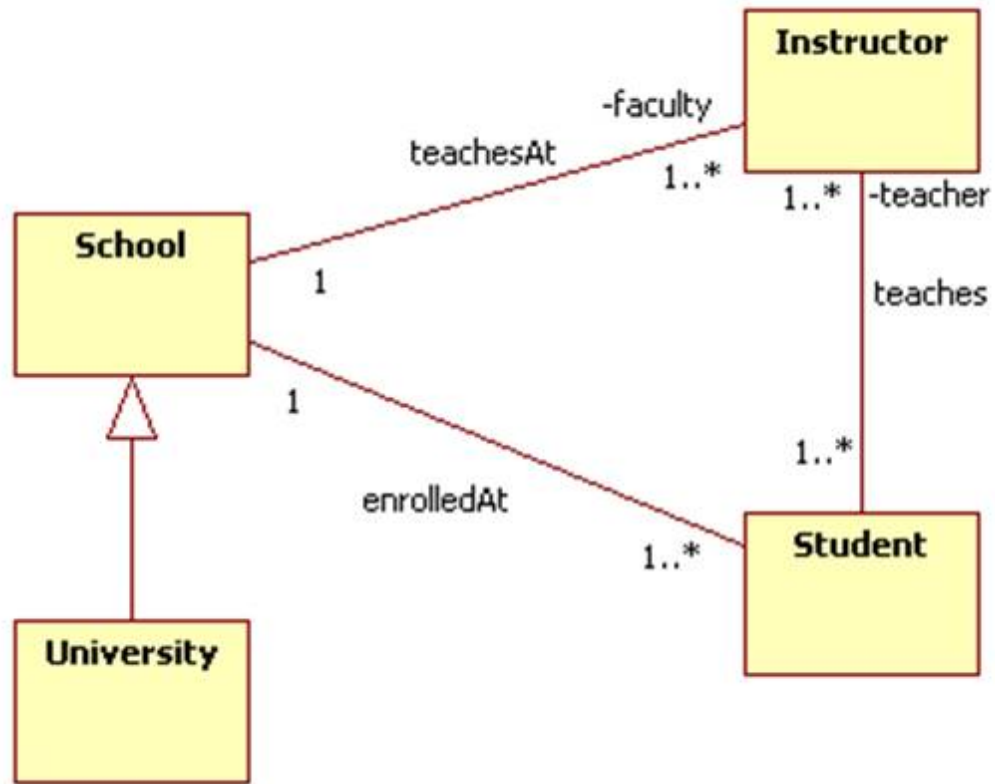
science, engineering, medicine, business, government, military

A domain model is a package containing class and activity diagrams.

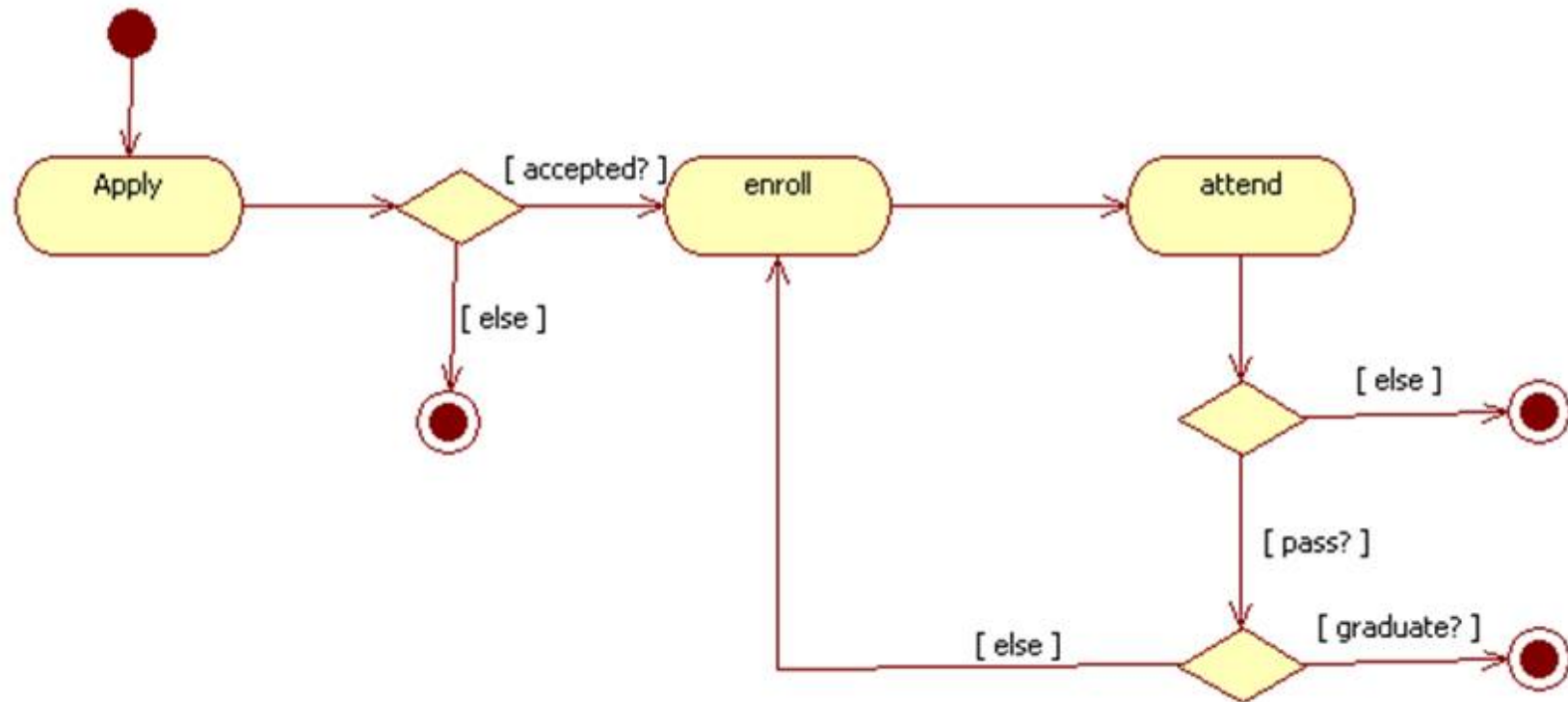
For example, the education domain can be modeled by a package:



This package contains the following class diagram which models the concepts of school, university, instructor, and student as well as the relationships teaches-at, teaches, and enrolled-at:



It also contains the following activity diagram that represents the student workflow:



Other types of domain models

There is a spectrum of domain models. In order of increasing complexity we have:

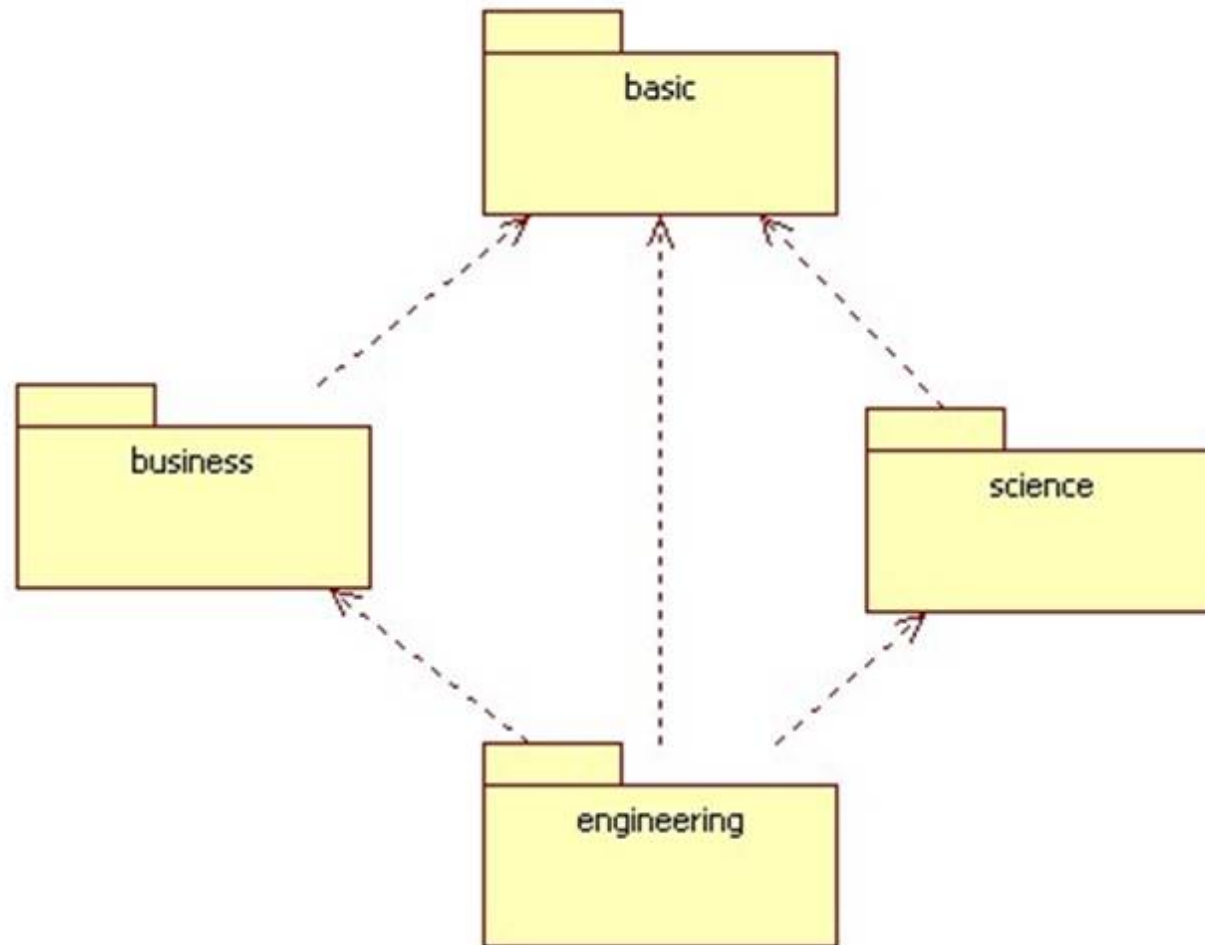
- Glossaries
- Taxonomies
- Thesauri
- Entity-Relationship Models
- Object Models (UML)
- Ontologies (description logic)
- Domain Theories (first-order logic)

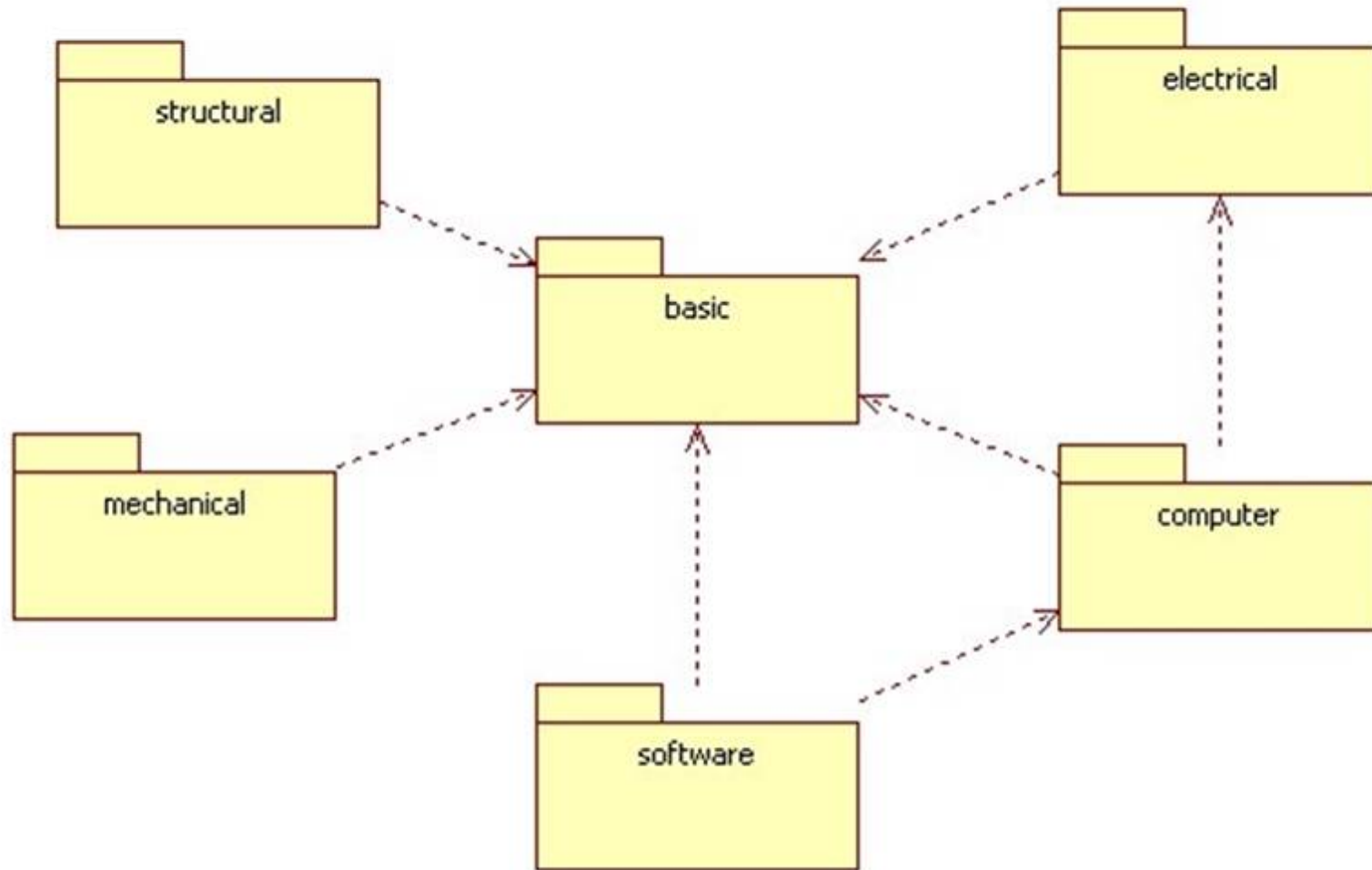
Dependencies between Domains

Domains can import elements from other domains. This can be shown by a package diagram. A package diagram is a class diagram that shows packages and the dependencies between them.

A dependency between two packages is represented by a dashed arrow. More specifically, dashed arrow from package A to package B indicates that package A uses some of the items contained in package B. We say that A imports from B or that B exports to A.

Here are a few examples:





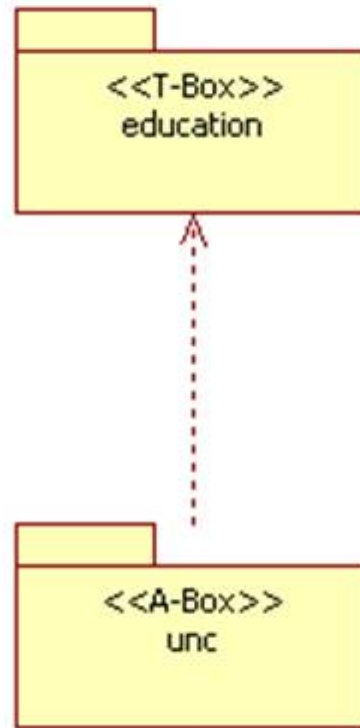
A-Box (M0) and T-Box (M1) Models

A domain model can be conceptual or particular. While a conceptual model captures general concepts and relationships of some domain, a particular model captures objects and links between objects in a specific instance of the conceptual domain. Conceptual models are called T-Box models (T = terminology) while particular models are called A-Box models (A = assertion).

For example, here's a description of a particular university consisting of particular students and instructors:

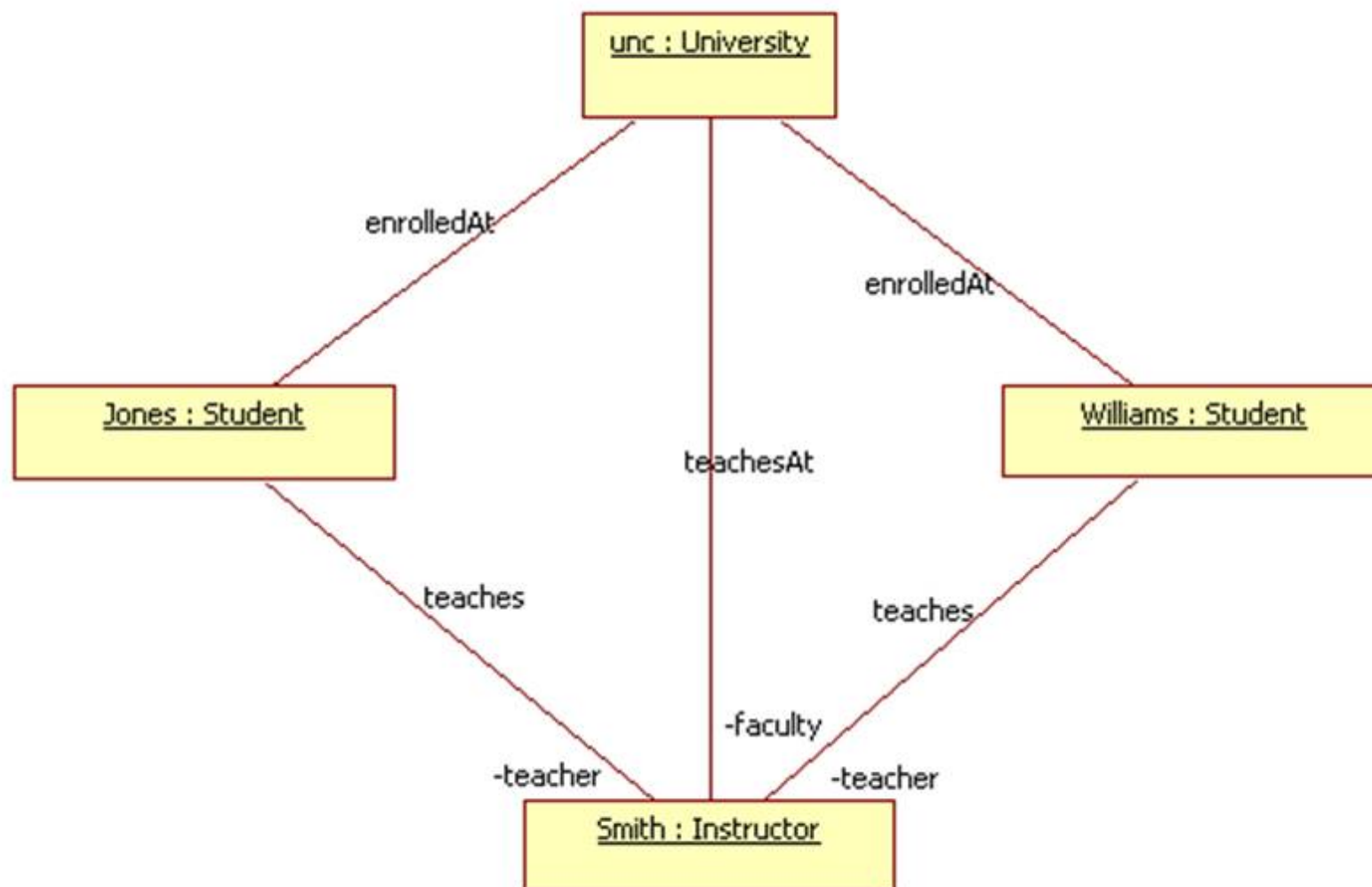
Smith teaches Jones and Williams at UNC (the University of Northern California).

We can represent UNC as a package that depends on elements imported from the education domain:



We may infer that Smith is an instructor and that Jones and Williams are students. In other words, Smith is an instance of the education.Instructor class while Jones and Williams are instances of the education.Student class. This means that the unc package contains an object diagram that instantiates the class diagram in the university package. (An object diagram is a class diagram containing objects and links. An object diagram, A, instantiates a class diagram, B, if all of the objects and links in A instantiate respective classes and associations in B.)

Here it is:



Domain Modeling

Domain modeling is the activity of translating an informal description of a domain into a domain model.

Domain models are used during the requirements gathering phase to clarify important domain-specific terms.

Domain models are useful during the requirements analysis phase to identify the entity classes in the Entity-Control-Boundary architecture.

Domain models are closely related to the [Entity-Relationship models](#) used by database designers.

Domain Stereotypes

Typical domain class stereotypes include:

`<<thing>>` = an entity that has mass and volume

`<<event>>` = an entity that has a start time and duration

`<<role>>` = an entity that executes tasks. For example: plumber, clerk, nurse

`<<type>>` = an entity that describes other entities: film genre (SciFi, comedy, etc.), film rating (P, PG, etc.)

Examples of `<<thing>>` classes:

Person, Report,

Examples of `<<role>>` classes:

Customer, Doctor, Student

Examples of `<<event>>` classes:

Transaction,

Examples of `<<type>>` classes:

Genre, ProductType, Rank

Text Analysis

Text Analysis is a fancy term that simply means identifying important sounding phrases in a textual description of a domain. Important noun phrases—user, owner, controller, buffer, etc.-- are often modeled by classes, important verb phrases—uses, has, owns, controls, etc.-- are often modeled by associations. Phrases suggesting classifications—kinds, types, is one of, etc.—are often modeled using generalization. Phrases suggesting assemblies— is part of, is made from, etc.—are often modeled using composition. Phrases suggesting collections—contains, holds, etc.—are often modeled using aggregation.

Analysis Patterns

Formally, a design/analysis pattern is a [recurring parameterized collaboration](#):

An analysis pattern is a class diagram template that is instantiated in many analysis object models.

[Analysis Patterns](#)

Examples

[Analysis Model Labs](#)