Modeling Concepts

In this Lecture you will Learn:

- What is meant by a model
- The distinction between a model and a diagram
- The UML concept of a model

What is a Model

- Like a map, a model represents something else
- A useful model has the right level of detail and represents only what is important for the task in hand
- Many things can be modelled: bridges, traffic flow, buildings, economic policy

Why use a Model?

- A model is quicker and easier to build
- A model can be used in a simulation
- A model can evolve as we learn
- We can choose which details to include in a model
- A model can represent real or imaginary things from any domain

Modelling Organizations

- Organizations are human activity systems
  - ◆ The situation is complex
  - ◆ Stakeholders have different views
  - ◆ We have to model requirements accurately, completely and unambiguously
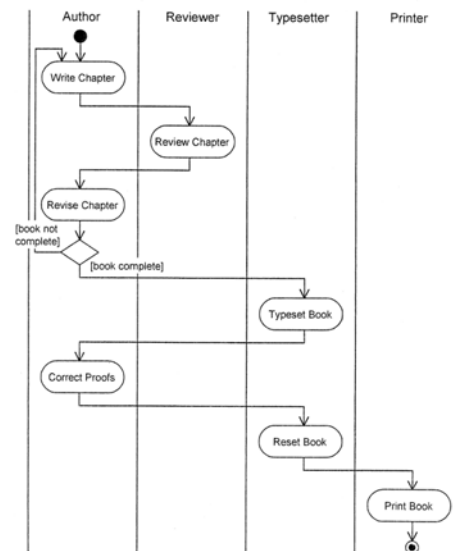 - ◆ The model must not prejudge the solution

What is a Diagram?

- Abstract shapes are used to represent things or actions from the real world
- Diagrams follow rules or standards
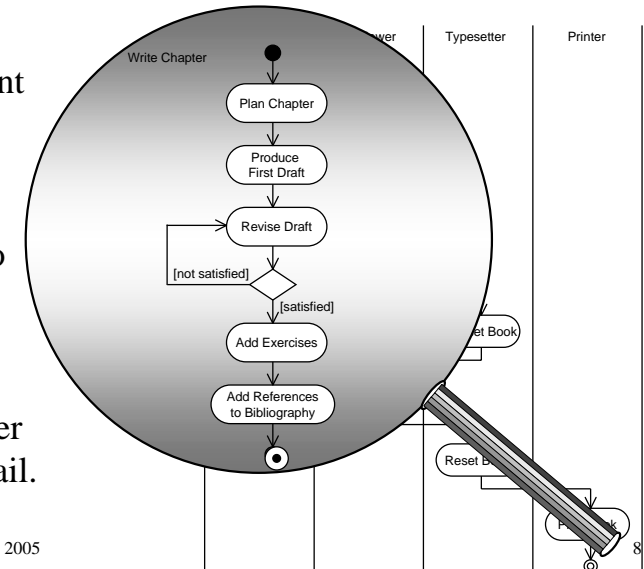- The standards make sure that different people will interpret the diagram in the same way

An Example of a Diagram

- An activity diagram of the tasks involved in producing a book.
- It is highly simplified!

Hidden Detail

- This is meant to be a magnifying glass! It is looking into the Write Chapter activity to show a lower level of detail.
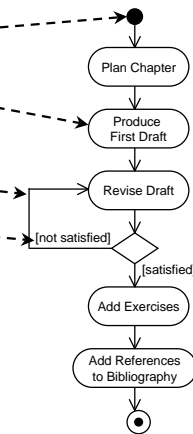
Unified Modeling Language

- The Unified Modeling Language (UML) - is OMG's most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.
- UML, along with the Meta Object Facility (MOF), also provides a key foundation for OMG's Model-Driven Architecture, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution.

Diagrams in UML

- UML diagrams consist of:
  - ◆ icons
  - ◆ two-dimensional symbols
  - ◆ paths
  - ◆ Strings
- UML diagrams are defined in the UML specification.

Plan Chapter

Produce First Draft

Revise Draft

[not satisfied]

[satisfied]

Add Exercises

Add References to Bibliography

Class Exercise

- What are the four elements of a UML diagram?

Diagrams vs. Models

- A diagram illustrates some aspect of a system.
- A model provides a complete view of a system at a particular stage and from a particular perspective.
- A model may consist of a single diagram, but most consist of many related diagrams and supporting data and documentation.

Class Exercise

■ What is the different between a diagram and a model?

Examples of Models

■ Requirements Model
  ◆ complete view of requirements
  ◆ may include other models, such as a Use Case Model
  ◆ includes textual description as well as sets of diagrams

Examples of Models

■ Behavioural Model
  ◆ shows how the system responds to events in the outside world and the passage of time
  ◆ an initial model may just use Collaboration Diagrams
  ◆ a later model will include Sequence Diagrams and Statecharts

Class Exercise

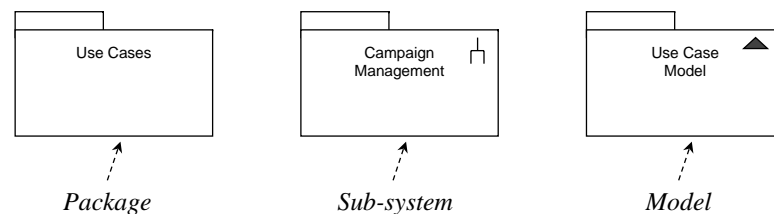■ Why do we need standards for the graphical elements of diagrams?

Models in UML

- A system is the overall thing that is being modelled
- A sub-system is a part of a system consisting of related elements
- A model is an abstraction of a system or sub-system from a particular perspective
- A model is complete and consistent at the chosen level of abstraction

Models in UML

- Different models present different views of the system, for example:
  - ◆ use case view
  - ◆ design view
  - ◆ process view
  - ◆ implementation view
  - ◆ deployment view

Packages, Sub-systems and Models

- UML has notation for showing sub-systems and models, and also for packages, which are a mechanism for organising models (e.g. in CASE tools)

Use Cases	Campaign Management	Use Case Model
Package	*Sub-system*	*Model*

Class Exercise

- In what two ways can we show in UML that something is contained within something else, for example a sub-system within another sub-system?

Developing Models

- During the life of a project using an iterative life cycle, models change along the dimensions of:
  - ◆ Abstraction ─ they become more concrete
  - ◆ Formality ─ they become more formally specified
  - ◆ Level of detail ─ additional detail is added as understanding improves

Development of the Use Case Model

- In a system development project that uses an iterative life cycle, different models that represent the same view may be developed at different levels of detail as the project progresses.
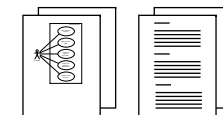
Example

- The first use case model of a system may show only the obvious use cases that are apparent from the first iteration of requirements capture.
- After a second iteration, the use case model may be elaborated with more detail and additional use cases that emerge from discussion of the requirements. Some prototypes may be added to try out ideas about how users will interact with the system.
- After a third iteration, the model will be extended to include more structured descriptions of how the users will interact with the use cases and with relationships among use cases. The number of iterations is not set at three. Any phase in a project will consist of a number of iterations, and that number will depend on the complexity of the system being developed.

Development of the Use Case Model

Iteration 1
Obvious use cases.
Simple use case descriptions.

Iteration 2
Additional use cases.
Simple use case descriptions.
Prototypes.

Iteration 3
Structured use cases.
Structured use case descriptions.
Prototypes.

Class Exercise

- Why do we use models in developing computerized information systems and other artifacts?