Domain Understanding and Modeling

DUM 01

Course organization

- Attendence at lectures is not mandatory.
- Attendence at seminars is mandatory although there is tolerance of 3 absences. There is a 5 points penalty for every absence above three.
- Seminar projects are delivered by teams of 4 members (default).
- Finishing the project is prerequisite for attending final test. Seminar supervisors are allowed to penalize projects.
- Final test is written, consists of 5 assignments / questions for 50 points total.
- 25 points are requested for passing the course.

Aim of this course

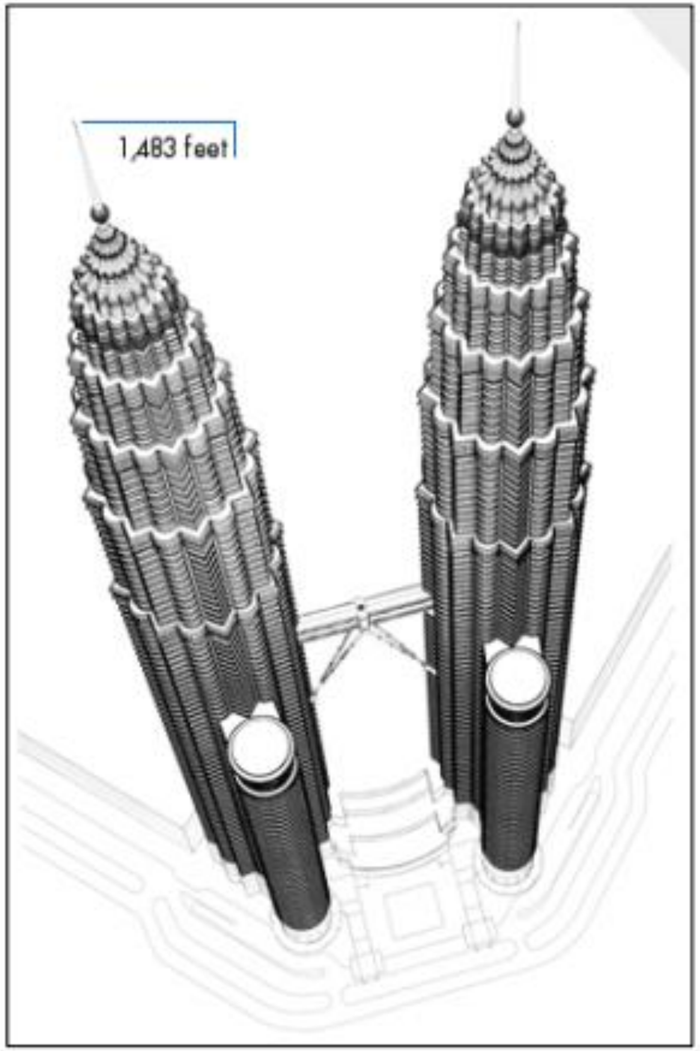
- To understand models in IT, to learn how to use and combine them in real project execution.

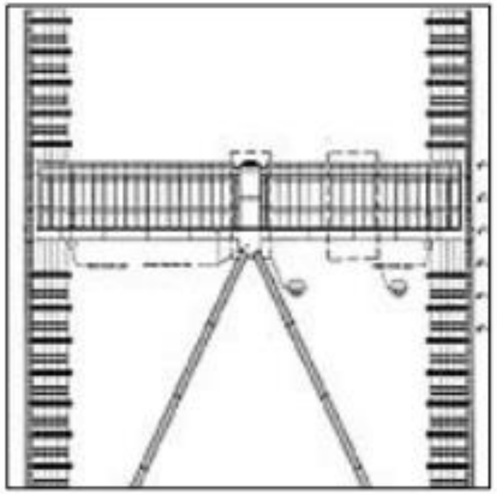
How it will be done

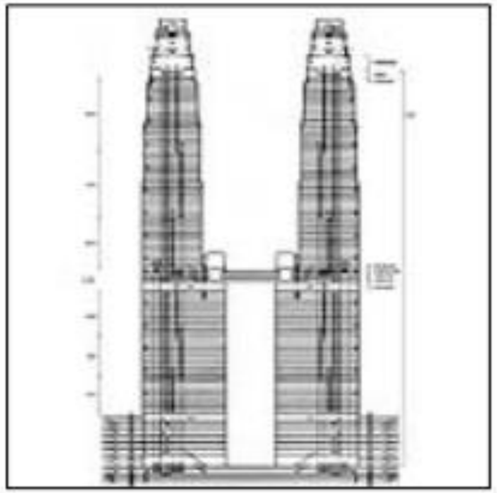
- Aim of lectures is to:
  - refresh knowledge about domain modelling from former studies
  - fill-in possible knowledge gaps
  - put everything in context

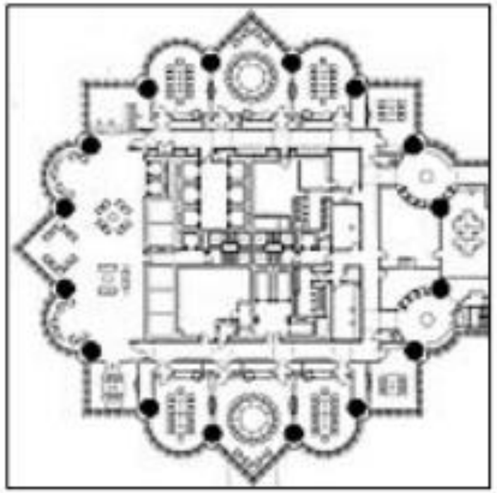
- Aim of seminars is to:
  - get practical experience with creating and balancing models

Plans and models are indispensable...









Purpose of models

- To emphasise important traits and hide unimportant traits.
- To discuss changes of specification originating from users' request with minimal cost and minimal risk.
- To check up whether we understand the background of our IS (real-life environment supported by the IS).
- To check up whether the knowledge was captured accordingly so the developers can build up the system.
- Use as simple and comprehensive system as possible with regard to the style of work and solution of situation.

Attributes of models

- Graphical form with appropriate amount of documentation
- Ability to look at the system hierarchically - ’top-down’ approach
- Minimal redundancy
- Transparency
  - Reader must have an impression of a system, not of an abstract model.

Example - Models of SW request specification

- List of functional/nonfunctional requests
    - Simple list of identified requests
    - Classification into groups according to character and relevance
- Event-reaction list
    - Model of system’s external behavior
    - For instance a table of event-reaction doublets
- Requested functionality
    - User’s vision about
        - requested functionality
        - form of output (screens, sets of screens)
        - outputs recorded/ specified by prototypes

Decomposition

- Decomposition helps system’s analyst and other team members to disassemble the system into smaller components that are easier to understand and to handle
- Allows focusing attention on certain subsystem without interference with other subsystems in given time
- Allows focusing attention on certain subsystem that is important for certain audience without confusing them with unrelevant details
- Allows delivering different parts of system in independent time and by various (teams of) people

Which models will we use?

- Mind maps and WBS
- Static and dynamic UML models
- Process models
- Data models
- Web and UI models
- Project models