What is Modelling?
What is Modelling?

- Observes
- Makes Comparisons
- How the system's properties evolve over time
- How the system is controlled
- System Theory

Diagram: A thought process involving observation, making comparisons, and understanding system dynamics and properties.

Image: A person observing a medical scenario, indicating the application of modelling principles in a real-world context.
Modelling...

• **Modelling can guide elicitation:**
  – It can help you figure out what questions to ask
  – It can help to surface hidden requirements
    • i.e. does it help you ask the right questions?

• **Modelling can provide a measure of progress:**
  – Completeness of the models -> completeness of the elicitation (?)
    • i.e. if we’ve filled in all the pieces of the models, are we done?

• **Modelling can help to uncover problems**
  – Inconsistency in the models can reveal interesting things...
    • e.g. conflicting or infeasible requirements
    • e.g. confusion over terminology, scope, etc
    • e.g. disagreements between stakeholders

• **Modelling can help us check our understanding**
  – Reason over the model to understand its consequences
    • Does it have the properties we expect?
  – Animate the model to help us visualise/validate the requirements
Systems involves a lot of modelling

• A model is more than just a description
  – it has its own phenomena, and its own relationships among those phenomena.
  • The model is only useful if the model’s phenomena correspond in a systematic way to the phenomena of the domain being modelled.

Source: Adapted from Jackson, 1995, p120-122
“It’s only a model”

- There will always be:
  - phenomena in the model that are not present in the application domain
  - phenomena in the application domain that are not in the model

- A model is never perfect
  - “If the map and the terrain disagree, believe the terrain”
  - Perfecting the model is not always a good use of your time...

Source: Adapted from Jackson, 1995, p124-5
Don’t forget what we’re modelling

• **During analysis**
  – we want to know about the application domain and the requirements
  – …so we develop a course-grained model to show where responsibilities are, and how objects interact
    • Our models show a message being passed, but we don’t worry too much about the contents of each message
    • To keep things clear, use icons to represent external objects and actors, and boxes to represent system objects

• **During design**
  – we want to say how the software should work
  – … so we develop fine-grained models to show exactly what will happen when the system runs
    • e.g. show the precise details of each method call