The Role of Modelling

Definition: Model

A model is something that provides an abstraction of a solution, capturing its essential characteristics and omitting unnecessary detail

- We use models to help produce solutions for both WSPs and also ISPs

1 Why do we need to model?

- Complexity and size of software products means we need abstractions (models) to help predict how desired characteristics will be realised in the actual product
- So we use models to support such activities as:
  - Requirements elicitation and specification
  - Design of systems and applications
  - Costing and planning
  - Risk assessment
- An important role for models in designing is that they help manage the cognitive load for large applications

2 How do we develop a model

- Ideas about architecture can help with deciding what form of model we want to create
- Frameworks such as design patterns can help with organising the detailed structure of a model

Important: Models

Creating models is an iterative process, we rarely get it right first time. It's a but like programming

3 Perspectives and Viewpoints

Definition: Perspective

Relates to a software development role. Each has their own set of interests and needs

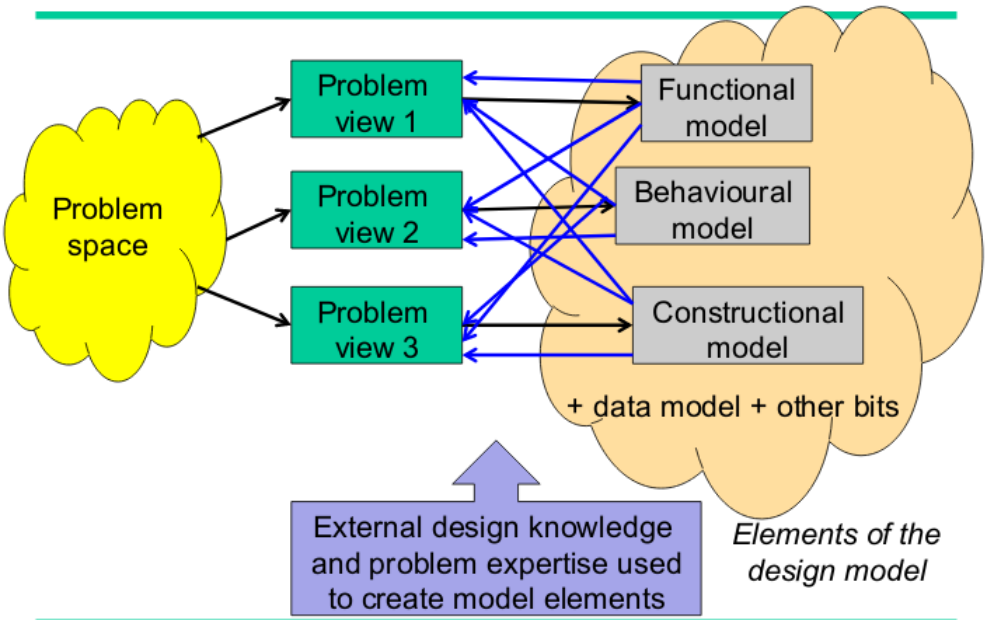
Definition: Viewpoint

A set of particular characteristics or attributes of a model, which in turn embody some specific aspect of software. A specific viewpoint is usually described by using one or more particular representations

3.1 Design viewpoints

- An abstraction essentially "omits properties other than those of immediate interest"
- A viewpoint therefore focuses on a set of design attributes that relate to a particular abstraction
- The viewpoints we care about are:
  - **Constructional** - Describing static properties and constructional details
  - **Behavioural** - Describing the casual links between events and system responses
  - **Functional** - describing the operations performed
  - **Data modelling** - describing the forms of data elements and the relations between them

4 Software as an ISP



5 Interconnectedness

The viewpoints can be considered to be ”projections” of the model, and are not independent

6 Representations

- Provide the abstract descriptions we use in our models. A representation usually describes the attributes of the model that are related to a particular viewpoint
- Representations may use such forms as
  - Textual
  - Mathematical
  - Diagrammatical

They have many roles, such as:

- Describing the characteristics and properties of the problem (usually for requirements analysis)
- Documenting the designer’s ideas about the form of solution being proposed (for architectural design and construction)
- Explaining design ideas to others (customer, design team, implementers)
- Negotiating design ideas between team members (and possibly with the customer)
- Checking the degree of both consistency and completeness in a solution (V and V)

6.1 Forms of representation

Diagrams usually have a ’box and line’ for and can be very informal. Established forms incorporate both:

- Explicit semantics, provided by the symbols
- Implicit semantics, reflected by the ’conventions’ adopted by users, such as positioning, orientation etc

The invisible nature of software also mean that there is no visual correspondence between a notation and the properties it intends to describe

7 Abstraction

- A representation provides a particular abstraction of a system, related to the needs of a specific activity
- An abstraction omits the information that is not relevant to the task in hand, and emphasises the essential properties of interest

8 Sketching

Experts tend to sketch their designs, and often, they only formalise the diagrams if there is a specific requirement to do so

In particular, they:

- Rarely observe rules of diagram syntax very closely when sketching (if at all)
- Make notes, particularly when they recognise something which they know how to handle
- Make lists, often to help with checking that their model addresses necessary features
- Use the sketches to explore ideas, check consistency etc