Modeling Introduction

**Purpose**
- This is only a “brief” introduction not a full course
- Modeling is a very deep subject matter
- I will only cover very few concepts
Modeling Introduction

- What are Models?
Modeling Introduction

- What are Models?
  - A compromise with reality
Modeling Introduction

- What are Models?
  - A compromise with reality
  - A caricature of reality
What are Models?
- A compromise with reality
- A caricature of reality
- A replica of reality
Modeling Introduction

- What are Models?
  - A compromise with reality
  - A caricature of reality
  - A replica of reality
  - A reflection of reality
Modeling Introduction

What is this?
A compromise with reality

- Consider the map of the world – it is a 2-D chart which is rectangular in shape with literally 4 corners to it – But we know the world to be (almost) a perfect sphere. The compromise here is a 3-D to 2-D transformation. We compromise on fidelity (shape).

- The detail we are interested in deriving from the chart (international borders, continental shapes, oceanic dispositions, direction, location, latitude, longitude, time zone, distance etc.) are not diminished or obscured by model shape (in this instance the 2-D chart).

- Even though some fidelity is lost there is no question that we can still recognize our world from the chart. So it is important to understand the attributes of fidelity and how to control/manipulate them to ensure that we can derive the information we seek without losing sight of the reality we seek to represent.
Modeling Introduction

Who is This?
A caricature of reality

The caricature has the distinct attribute of **distorting fidelity** by **exaggerating certain attributes** and **diminishing other attributes** to arrive at a result which generally is satirical. However, in a practical sense the exaggeration or necessary diminishing serves to **focus** on or **highlight key attributes** which are of interest above and beyond other aspects but without obscuring the identity of the subject of the model. Political satirists use this technique quite effectively as do system modelers.

The model must always be **identifiable** with the subject
Modeling Introduction

What is this?

HMS Victory
A Replica of Reality

- Consider a scale model, this is in fact a true facsimile or replica
  - Absolute in fidelity (with respect to certain attributes)
  - Does not exaggerate
  - Does not Obscure
  - Only diminishes in dimension (scale)
- The model bears makes an exact representation of most aspects of the actual – the difference are purely scale related
- Every attribute of the model can be scrutinized
- Every attribute can be studied in extreme detail and extrapolation upon its behavior made and scaled up or down
- The notable omissions, for example in this case are:
  - Physical size
  - Weight
  - Actual displacement
  - Material
  - May be internal (hidden) fittings
  - Working parts may or may not be actually working but representative of reality
What is this?
Modeling Introduction

- A Reflection of Reality
  - Consider a Hollywood set – it is a model
  - The set is a full size reflection (impression) of “reality”
  - From one perspective it is sufficiently complete that it looks deceptively like the real thing, the difference is the infrastructure – the scaffolding here replaces the actual structure of a real construction
  - From a different perspective the set is a sham, it is nothing like the real thing – then again it didn’t need to be and that is one of the great advantages of modeling – you only model to satisfy a specific objective – you are not trying nor should you be trying to build the real thing – that is what we have developers for.
  - The latter is also a clear advantage, working to the same specifications – we build so that we have the form (function) – not the substance – the actual code
All these were “models”
In each case they represented and highlighted the attributes of interest
None is absolutely complete
Yet all are useful to one degree or another
They serve a purpose
Modeling Introduction

- The P & C’s of Modeling
  - Provocation
    - Models should be provocative (think about it, ;-) )
    - Models should employ a non ambiguous notation that helps us understand and "discover" aspects of the system that we hadn’t yet understood or considered.
    - Models should provoke us to question
  - Completeness
    - Understand first that **no model is “complete”** – a model by its nature omits, compromises on or distorts certain characteristics whilst emphasizing or highlighting others
    - The “incompleteness” is called *abstraction* – never the less the model needs to be “complete” at the level of abstraction selected with no key attributes omitted
  - Consistency
    - Insure the model is predictable (*deterministic*) – i.e. given the same input it *repeatedly* produces the same output
  - Clarity
    - Avoid unnecessarily complicated or convoluted composition (it is a sure sign something is not right)
    - Strive for *brevity* and *simplicity*
Modeling Introduction

Software Modeling Space

- The 3 dimensional space is the one in which requirements are generated
- The axes represent the software modelling space with each providing a unique perspective

Our interest is in the **behavioral** attributes of the system
P²E of Modeling

- Prescribed Behavior
  - Prescribed behavior is generally drawn from the system requirements
  - Behavioral models draw from requirements and strive to express Prescribed or intended (required) behavior
  - Models invariably end up becoming a de facto formalism of the originating requirements

- Proscribed Behavior
  - It is difficult to specify Proscribed or unwanted behavior with any degree of completeness, if at all
  - Some (most) requirements don’t talk about Proscribed behavior at all
  - In general Proscribed behavior is arrived at by consensus of key stakeholders and unique models looking to express that are built for that purpose – consider credentialing and security matters as an example

- Executability
  - Models, in order to achieve our goals of speed and efficiency, really should be executable
  - Executable models are in effect an “executable” specification – thus models need robust reviews and must be “validated”
Modeling Introduction

- Kinds of Behavioral Models
  - Black Box Models
  - Glass Box Modes
Modeling Introduction

- Black Box Models
  - Similar (to a point) in concept to Black Box Tests
    - Black Box Tests are built only at the functional level without knowledge of the underlying (structural) code within the AUT
    - In terms of modelling, Black Box Models are functional and built without knowledge of the actual structure of the AUT
    - Black Box Models are built essentially by discovery
Modeling Introduction

- Black Box Approach – You see what you see

  Process of Discovery – What happens when?
  1. I press A – Box changes color to Yellow
  2. I press A again – Box returns to green
  3. I press B – Box changes to Blue
  4. I press B again – Box remains Blue
  5. I press C – Box changes color to red
  6. I press C again – Box returns to green
Modeling Introduction

Black Box Model

Diagram: A flowchart illustrating transitions between states labeled Blue, Green, Yellow, and Red, with transitions labeled Press_B, Press_A, and Press_C.
Modeling Introduction

Black Box Model Approach

What I have Physically

Behavior I observe

But is What I observe that which was Intended?
Modeling Introduction

- Glass Box – Structural and Function
  - With glass box we are able to deduce structure
  - We have requirements to describe function
Modeling Introduction

- Glass Box – Structure & function - Requirements

  - Structure
    - The application under test is a Box
    - The application shall have 3 controls
      - A
      - B
      - C

  - Function
    - Selecting A shall result in the box changing color to Yellow
    - Whilst the box is Yellow, selecting A shall cause the box to change color to Green
    - Selecting B while the Box is Green shall cause the box to change color to Blue
    - Selecting B whilst the Box is Blue shall cause no change color change
    - Selecting C whilst the Box is Blue shall cause the Box to change color to Red
    - Selecting C whilst the Box is Red shall cause the Box to change color to Green
Modeling Introduction

Glass Box Model

A

B

C
Modeling Introduction

- Glass Box Models
  - Provide the capability to confirm structure
  - Test functionality
  - Validate AUT against expectations formally captured in the model
Modeling Introduction

- Glass Box Approach

What I physically have

What I have modelled based on Requirements

I have an objective means of comparison
How To Model – Where to start

Of all the possible modeling approaches we will use the Extended Finite State Machine approach and notation (EFSM) – Because

- TestOptimal (our modeling environment of choice) is designed around the EFSM notation
- Our products are reactive systems
- Reactive systems:
  - Express:
    - States (outward observable expression of condition)
  - Consume
    - Events (triggers or transitions)
Modeling Introduction

- How To Model – Where to start
  - Start with your requirements (if you have them)
  - If you have no requirements consider the workflows and ask subject matter experts for guidance
  - Review the model progressively and often with knowledgeable individuals
  - Capture the behavior as a series of States and Transitions
    - Define your system as a series of discrete States connected by directed actions (transitions)
    - Consider your states – can their condition or behavior be viewed as being the outcome of other internal behavior at a higher level of resolution – Sub-states?
    - Consider your states – can their condition or behavior be viewed as the outcome of internal behavior that is separately and autonomously exercisable – Sub-models?
    - Capture the triggers that cause your system to change States or Sub-states – Transitions
Modeling Introduction

Consider:
Modeling Introduction

Consider:

There is an aspect of required behavior that can be exercised autonomously or in cooperation with this model – Logging in.
Modeling Introduction

Consider – Login model:

This is an autonomously existing and executing model that achieves real results individually.
Instead of rebuilding a Login functionality we can re-use an existing functionality captured in an existing model.
Modeling Introduction

Sub-model Expanded

Existing login model used as a sub-model embedded in a state "BusinessAccountLogin"
Modeling Introduction

Additionally, we can add sub-states to the Search:
Modeling Introduction

Model complete with Sub-models and Sub-states
Guided Model Execution

- EFSM’s have the attribute of executing an “if” like method on transitions – this provides for guarded conditional execution
- Global and State Variables are available for use in EFSM models
- Binary operators such as NOT, NAND, NOR, AND, OR, XOR are supported in conditional execution
- In testing we are interested in validation and so Assertions are primary to this in automated EFSM
Data Driven Models

- Consider the **Data** you are using to drive your model. Your data needs to be modeled with the best approaches possible, consider as a minimum the:
  - the input domains
  - the equivalence classes that fully cover the input
  - an exhaustive boundary value analysis for each equivalence class
  - generating combinatorial arrangements for every equivalence class, every field etc.
  - For each combination propose an expectation, an assertion
Modeling Introduction

- Questions?