

Model-Based Testing in C# (Goldilocks)



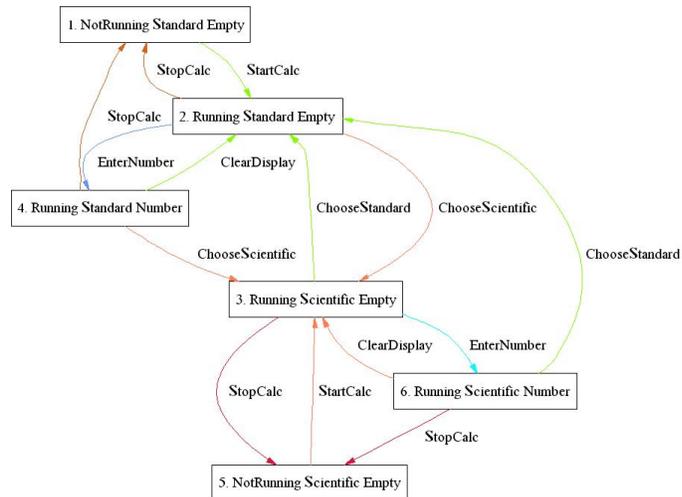
Harry Robinson (harryr)
Michael Corning (mcorning)
Enterprise Management Division
30 April 2004

What is a Model?

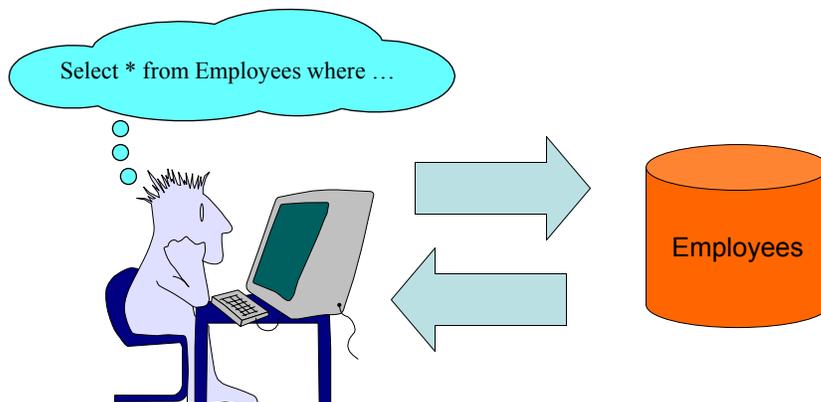


- A model is a description of a system's behavior.
- Models are simpler than the systems they describe.
- Models help us understand and predict the system's behavior.

State Modeling Works Well for State Models



But not Every Model is a Matter of State ...

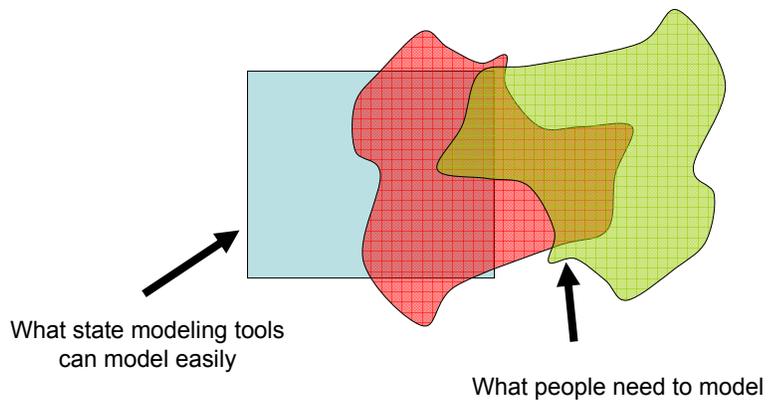


The tool should fit the problem

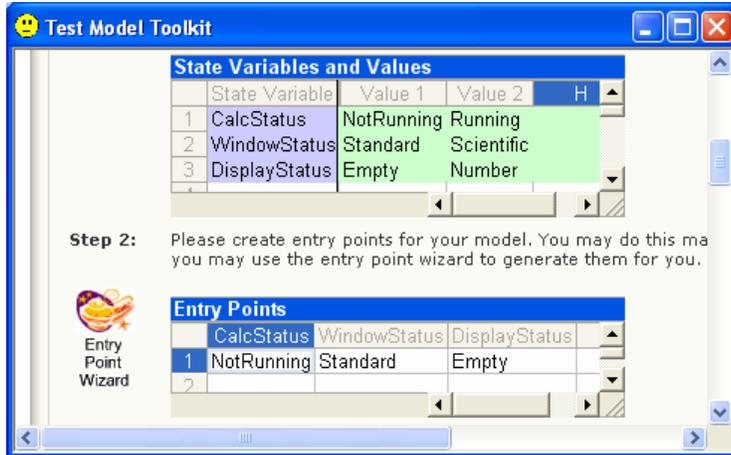


... not the other way around...

The States vs. the People



TMT State Variables & Initial State



C# State Variables & Initial State

```
// state variables
public enum CalcStatusValues {NotRunning, Running }
public enum ViewStatusValues {Standard, Scientific}
public enum DisplayStatusValues {Empty, Number};

// initial state
CalcStatusValues CalcStatus = CalcStatusValues.NotRunning;
ViewStatusValues ViewStatus = ViewStatusValues.Standard;
DisplayStatusValues DisplayStatus = DisplayStatusValues.Empty;
```

TMT Rules for Enabled Actions and State Changes

Transition Rules				
	Action	CalcStatus	WindowStatus	DisplayStatus
8				
9	StartState	Running	Standard	
10	ChooseStandard			
11	EndState			
12				
13	StartState	Running	Scientific	
14	ChooseStandard			
15	EndState		Standard	Empty
16				

C# Rules for Enabled Actions

```
ArrayList a = new ArrayList();  
  
if (CalcStatus == CalcStatusValues.NotRunning)    a.Add("StartCalc");  
  
if (CalcStatus == CalcStatusValues.Running)      a.Add("StopCalc");  
  
if (CalcStatus == CalcStatusValues.Running)      a.Add("ChooseStandard");  
  
if (CalcStatus == CalcStatusValues.Running)      a.Add("ChooseScientific");  
  
if (CalcStatus == CalcStatusValues.Running)      a.Add("ClearDisplay");  
  
if (CalcStatus == CalcStatusValues.Running  
    && DisplayStatus == DisplayStatusValues.Empty) a.Add("EnterNumber");
```

C# Rules for State Changes

```
// Rules for State Changes
void StartCalc(){ CalcStatus = CalcStatusValues.Running; }

void StopCalc() { CalcStatus = CalcStatusValues.NotRunning; }

void ClearDisplay() {DisplayStatus = DisplayStatusValues.Empty; }

void EnterNumber() {DisplayStatus = DisplayStatusValues.Number;}

void ChooseStandard()
{
    if (ViewStatus == ViewStatusValues.Scientific )
        DisplayStatus = DisplayStatusValues.Empty;
    ViewStatus = ViewStatusValues.Standard;
}

void ChooseScientific()
{
    if (ViewStatus == ViewStatusValues.Standard)
        DisplayStatus = DisplayStatusValues.Empty;
    ViewStatus = ViewStatusValues.Scientific;
}
```

Making MBT Available to Partners

From: Chris Shelley
Sent: Wed 2/18/2004 9:12 AM
To: Model-Based Testing
Subject: Public availability of tools

I assume that TMT is a Microsoft internal tool. Does anyone know of any similar tools that are available to companies outside Microsoft? I'm in communication with a test team at Dell, and they are asking me if I know of any products available similar to TMT.

Proliferating MBT via C#

From: William Rollison
Sent: Thursday, February 19, 2004 12:27 PM
Subject: RE: C# modeling

Model based testing has proven very useful in several case studies throughout MS. The ability to implement MBT in C# without the reliance on tools is essential to the proliferation of this technique throughout the company and to also be able to introduce this externally (colleges, universities).

... MBT is a significant stride forward in advanced testing techniques. We should not restrict ourselves to a tool, but we should develop a process in which an easy implemented solution to drive the adoption of MBT is made available and can easily be incorporated into any testing harness either internally or externally.

Interest in using C#

From: Mike Gallacher
Sent: Wednesday, February 18, 2004 10:29 AM
Subject: RE: Public availability of tools

I'm very interested in what you are doing with C#. Our PUM wants dev and test doing only C#, so he nixed A# and was wary of modeling tools in general. We decided not to push ITE harder with him, since ITE's timeframe and stability didn't line up with our schedule requirements, so we have been moving in a pure C# direction ...

Shared Sourcing the Work

From: John Lambert (WEB SERVICES)
Sent: Wednesday, February 18, 2004 1:20 PM
Subject: RE: Public availability of tools

I'm very interested in this; do you have more details?

Is there any way I can get involved?

[I work on Indigo and wrote the PICT / pairwise C# wrapper]

Take-aways

- Models should accommodate the system under test
- Not everything we want to model is a state machine
- More flexible than some approaches
- Easier learning curve than others
- C# modeling means MBT not reliant on a tool
- MBT can be shared with partners